Politecnico Di Torino

Dipartimento Energia "Galileo Ferraris"

Corso di Laurea Magistrale in Ingegneria Elettrica

Algoritmo di classificazione degli elettrodomestici tramite firma armonica e foresta decisionale

Relatore:
Prof. Gianfranco Chicco

Candidato:
Gabriele Sedino

Anno Accademico 2020-2021

Perseverare, in alcuni casi,
non è diabolico.

Indice

Introduzione

Gli ultimi decenni sono stati scenario di un crescente sviluppo dei sistemi elettrici con una conseguente espansione della rete di distribuzione. Se tale ampliamento della rete di distribuzione era stato una sufficiente soluzione per servire il numero crescente di utenze, con il trascorrere del tempo tale implementazione ha portato ad un sistema di gestione complesso che non soddisfa le attuali esigenze di affidabilità e flessibilità. Ne consegue una ulteriore conferma dell'importanza dell'energia elettrica nel consentire lo sviluppo dei nuovi mezzi della società moderna nonché, alla luce di questo irrefrenabile progresso tecnologico, l'esigenza di un corrispondente aggiornamento strutturale del sistema elettrico.

L'esigenza di costante modernizzazione del sistema elettrico non è dovuta solo al progresso tecnologico citato, ma rileva anche alla luce del corrispondente impatto ambientale. La produzione di energia elettrica, infatti, costituisce uno tra i processi più gravosi per quanto concerne l'emissione di CO_2. E' possibile affermare, pertanto, che il processo di modernizzazione strutturale del sistema elettrico dovrà coinvolgere anche settori come ad esempio il trasporto ed il riscaldamento in cui viene utilizzato il combustibile fossile, al fine di sostituirsi a quest'ultimo e ridurre le emissioni inquinanti. Tale processo di graduale sostituzione dell'energia elettrica al combustibile fossile prende il nome di elettrificazione (dall'inglese *electrification*) conferma ulteriormente l'esigenza di uno sviluppo strutturale del sistema elettrico, che dovrà far fronte al crescente numero di ambiti in cui sarà applicato.

In questa direzione va il "Quadro 2030 per il clima e l'energia" [8] inserito nell'ambito del Green Deal europeo [7], il quale presenta gli obiettivi strategici ed i traguardi imposti dall'unione europea per il periodo 2021-2030. Gli obiettivi che l'Unione Europea si è posta, mirando anche ad un aumento dell'efficienza energetica e dell'energia da fonti rinnovabili, sono i seguenti:

- una riduzione almeno del 40% delle emissioni di gas a effetto serra (rispetto ai livelli del 1990);
- una quota almeno del 32% di energia rinnovabile; [1]

[1] Per ulteriori approfondimenti si rinvia al sito della Commissione Europea al seguente

- un miglioramento almeno del 32.5% dell'efficienza energetica. [2]

Il panorama appena delineato ha contribuito alla formazione di una nuova definizione di rete elettrica: Smart Grid. Con tale espressione si indica una rete o un insieme di reti "intelligenti", capaci di gestire e monitorare la distribuzione di energia elettrica da tutte le fonti di produzione e soddisfare le diverse richieste di elettricità degli utenti collegati, produttori e consumatori in maniera più efficiente, razionale e sicura. L'utilizzo di tali reti elettriche porta diversi benefici: l'integrazione della generazione distribuita, l'apporto di energia elettrica necessario al soddisfacimento dei nuovi utilizzi sopracitati, nonché il miglioramento della continuità del servizio grazie alle funzioni di riconfigurazione automatica della rete e delle protezioni calibrate sulla tipologia di rete in questione. In particolare, il miglioramento dell'attuale sistema elettrico viene ottenuto attraverso punti di monitoraggio ed elaborazione distribuiti che rendono possibile una più precisa conoscenza dello stato della rete e un conseguente controllo degli impianti. Tali strategie mirano a:

- rendere il più omogeneo possibile il flusso di energia tra i punti di produzione e quelli di assorbimento per limitare gli stress alla rete dovuti ai picchi di consumo tramite classificazione e monitoraggio dei carichi;

- gestire in maniera più efficiente la generazione distribuita, anche in luoghi remoti, dell'energia, in particolare di quella originata da fonti di natura intermittente, come le turbine eoliche ed i pannelli fotovoltaici;

- aumentare l'affidabilità delle strutture e la qualità dell'energia fornita attraverso il monitoraggio dello stato della rete;

- fornire servizi aggiuntivi ai consumatori affinché abbiano la possibilità di scegliere il fornitore più adatto alle loro esigenze, siano agevolati nell'assumere comportamenti conformi al risparmio e nell'evitare eccessivi carichi della rete. Tempistica appropriata nell'utilizzo degli elettrodomestici, l'ottimizzazione del loro utilizzo ed eliminazione di attività indesiderate sono concetti chiave in tal senso.

Un passo in avanti verso le smart grid è sicuramente il recente aumento delle installazioni di contatori intelligenti nelle famiglie e nelle piccole imprese da parte delle aziende elettriche. Tale aumento ha evidenziato ulteriormente l'importanza del monitoraggio, al fine di fornire un servizio migliore ed ottenere informazioni utili sull'utilizzo degli apparecchi da parte degli utenti.

link: https://ec.europa.eu/energy/topics/renewable-energy_en

[2] Per ulteriori approfondimenti si rinvia al sito della Commissione Europea al seguente link: https://ec.europa.eu/energy/topics/energy-efficiency_en

Dunque, le reali condizioni operative degli elettrodomestici, e in generale delle apparecchiature elettriche, utilizzati in ambiti l'industriali e in contesti familiari, non possono essere determinate senza un adeguato sistema di classificazione e monitoraggio. La possibilità di riconoscere l'uso dei singoli elettrodomestici e il loro consumo energetico a partire da un solo dispositivo di misura è l'obiettivo delle tecniche di monitoraggio non intrusivo. A partire dalla metà degli anni Ottanta sono stati proposti diversi metodi e algoritmi ma non è ancora disponibile una soluzione completa e performante.

Questa tesi è strutturata come segue.
Parte I: propone le definizioni generali e presenta lo stato dell'arte della ricerca in campo NILM. Per una maggiore completezza delle informazioni saranno introdotti i concetti chiave della Cluster Analysis al fine di fornire un quadro più completo delle moderne tecniche di analisi del carico. Dunque si descriveranno i processi principali e i dataset di riferimento presenti in letteratura e i metodi seguiti al giorno d'oggi.
Parte II: sarà focalizzata sull'algoritmo di data mining sviluppato: classificazione degli elettrodomestici tramite firma armonica e foresta decisionale. All'interno di questa parte, pertanto, saranno presentati i concetti teorici necessari ai fini dello sviluppo e della comprensione di tale algoritmo, con successivo sviluppo dello stesso.
A seguire saranno esposte le conclusioni ricavate dall'analisi effettuata con un'esposizione dei possibili lavori futuri.

Parte I

Stato dell'arte

Capitolo 1

La Cluster Analysis

1.1 Introduzione

Con il termine analisi dei gruppi, o **Clustering**, ci riferiamo ad un insieme di metodi di analisi multivariata dei dati volta ad individuare unità tra loro simili rispetto ad un insieme di caratteri presi in considerazione e secondo uno specifico criterio. L'obiettivo è quello di raggruppare oggetti fisici o astratti tra loro eterogenei in classi omogenee ed esaustive. Si definisce cluster un insieme di oggetti che presentano tra loro delle similarità, ma che, contemporaneamente, presentano dissimilarità con oggetti posti in altri insiemi. L'input di questi algoritmi è costituito da un campione di elementi chiamato dataset, l'output è un certo numero di cluster in cui tali elementi vengono suddivisi in base ad una misura di similarità. Tali tecniche si differenziano quindi in base a come viene concepita la metrica, analizzata nel capitolo 1.2.

La cluster analysis consente di raggiungere i seguenti risultati [10]:

- la generazione di ipotesi di ricerca, infatti per effettuare una analisi di raggruppamento non è necessario avere in mente alcun modello interpretativo;
- la riduzione dei dati in forma, anche grafica, tale da rendere facile la lettura delle informazioni rilevate e parsimoniosa la presentazione dei risultati;
- ricerca tipologica per individuare gruppi di unità statistiche con caratteristiche distintive che facciano risaltare la fisionomia del sistema osservato;
- la costruzione di sistemi di classificazione automatica [15];
- la ricerca di classi omogenee, dentro le quali si può supporre che i membri siano mutuamente surrogabili [12].

Da notare quindi che i metodi di clustering, a differenza di altre tecniche statistiche, non compiono alcuna assunzione preventiva sulle diverse tipologie fondamentali esistenti che possono caratterizzare l'insieme studiato.

Osservando le caratteristiche dei diversi cluster in output si potranno successivamente prendere decisioni strategiche sulle azioni da compiere verso tali gruppi. Nel caso di sistemi elettrici, una buona pratica è quella di assumere come dataset i profili di carico al quarto d'ora ed applicare i problemi di clustering per raggruppare gli utenti in determinate categorie sulla base delle similarità di consumo.

1.2 Metriche

La **metrica** permette di identificare quanto simili siano due oggetti fra loro, questi potranno essere visti come valori che a loro volta potranno essere raggruppati in modo da formare dei vettori che rappresentino punti in uno spazio euclideo. La scelta di una metrica appropriata influenza la forma dei cluster, poiché alcuni elementi possono essere più "vicini" utilizzando una distanza e più "lontani" utilizzandone un'altra.

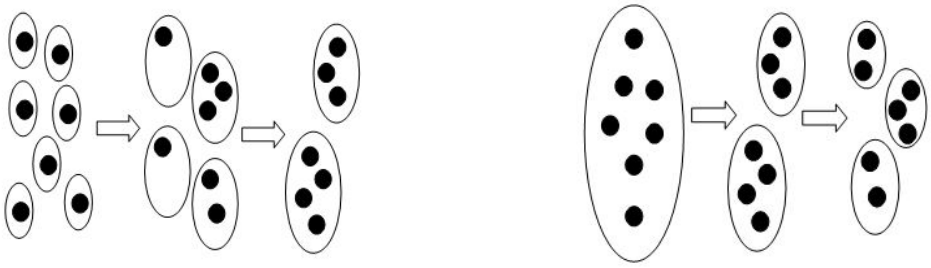
Le metriche maggiormente utilizzate sono:

- Distanza euclidea, ossia misura del segmento avente per estremi i due punti:
$$d(x, y) = \sqrt{\Sigma_{i=1}^{D}(x_i - y_i)^2}$$
- Distanza Manhattan, ossia la somma del valore assoluto delle differenze tra le coordinate dei due punti:
$$d(x, y) = |x_i - y_i|$$
- Distanza di Hamming, ovvero numero di bit differenti tra due stringhe:
$$d(x, y) = \Sigma_{i=1}^{D}(x_i - y_i)^2$$

dove $d(x, y)$ è la distanza tra i punti x e y D-dimensionali, con $i \in D$

1.3 Classificazione

Esistono più processi che permettono di clusterizzare un insieme di oggetti. La più importante suddivisione dipende dalla tecnica di generazione dei cluster stessi, che divide gli algoritmi in famiglie come mostrato in figura 1.1:



(a) Metodi bottom-up

(b) Metodi top-down

Figura 1.1: Metodi di classificazione

- Metodi **bottom-up** o *aggregativi*: inizialmente l'algoritmo considera ogni elemento come un cluster a sé per poi raggrupparli iterativamente in base al criterio di similarità scelto fino al raggiungimento di un numero scelto a priori di cluster, oppure fin quando la distanza tra i diversi gruppi supera un prefissato valore. È possibile anche scegliere un criterio statistico per determinare lo stop dell'algoritmo;
- Metodi **top-down** o *divisivi*: inizialmente l'algoritmo raggruppa tutti gli elementi in un unico cluster per poi, tramite il criterio di scelta, separarli iterativamente in cluster sempre più piccoli e omogenei. Il processo termina quando verrà raggiunto il numero di cluster desiderato.

Esistono varie classificazioni. Una prima categorizzazione dipende dalla possibilità che un elemento possa o meno essere assegnato a più cluster:

- Cluster **esclusivo** o *hard clustering*: ogni elemento può appartenere ad un unico cluster. Non esistono quindi elementi in comune tra cluster differenti;
- Cluster **non esclusivo** o *fuzzy clustering*: un elemento può appartenere a più cluster con gradi di appartenenza (*membership* o *livelli di sfumatura*) diversi. I cluster quindi possono contenere elementi in comune.

Un'altra classificazione tiene conto del tipo di algoritmo utilizzato per dividere lo spazio:

- Algoritmo di cluster **gerarchico**: i dati vengono organizzati in gruppi ordinabili secondo livelli crescenti rappresentabili in una struttura ad albero (dendrogramma);
- Algoritmo di cluster **partizionale**: forniscono un'unica partizione degli n oggetti in un dato valore di gruppi specificato a priori. Questi metodi sono derivazioni del più noto algoritmo di clustering detto delle k-means, da cui prendono anche il nome alternativo di k-clustering.

1.4 Rappresentazione dei dati

1.4.1 Dendrogramma

La rappresentazione grafica più utilizzata nella pratica è, il già citato, **dendrogramma**, figura 1.2a, in quanto restituisce immediatamente informazioni sui cluster ottenuti in seguito all'analisi. Questo è un grafo che presenta:

- nell'asse delle ascisse, la distanza logica dei clusters secondo la metrica definita;
- nell'asse delle ordinate, il livello gerarchico di aggregazione. La scelta del livello gerarchico (del valore che compare sull'asse verticale) definisce la partizione rappresentativa del processo di aggregazione.

1.4.2 Coefficiente di silhouette

Un'altra rappresentazione molto utile è quella basata sul **coefficiente di Silhouette**, figura 1.2b, questo tiene conto sia della separazione sia della coesione e viene calcolato per ogni osservazione del dataset, secondo il seguente algoritmo:

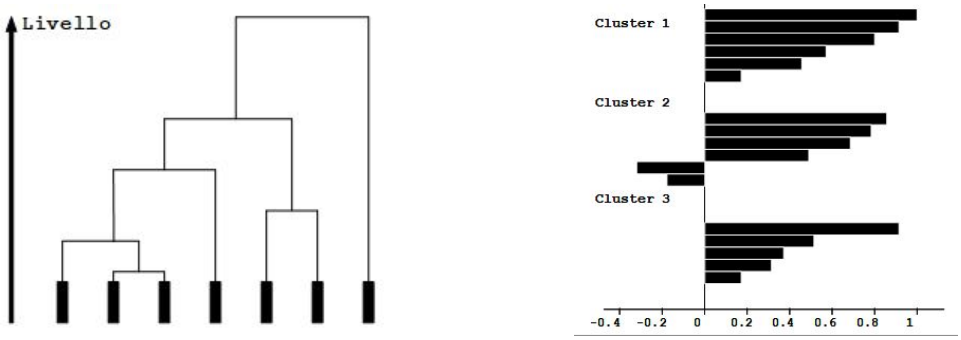
1. Si calcola la distanza media a_i tra l'i-esimo elemento e i restanti elementi dello stesso cluster, per ogni cluster;
2. Si calcola la distanza media tra l'i-esimo elemento del cluster considerato dagli elementi di altri cluster e se ne determina il minimo b_i;
3. Si calcola il coefficiente di Silhouette tramite la seguente formula:

$$silh(x_i) = \frac{b_i - a_i}{max(a_i, b_i)}$$

Il coefficiente di silhouette varia tra -1 e 1. Avere alti valori del coefficiente implica una buona appartenenza al cluster, mentre un basso valore ($silh < 0$) mostra un'errata adesione al cluster, ovvero il clustering finale per il dato elemento è errato. La rappresentazione grafica pone sulle ascisse i valori di silhouette tra -1 e 1, e sulle ordinate gli elementi suddivisi per cluster di appartenenza.

1.5 Metodi ed implementazione

Nell'analisi dei diversi metodi si è scelto di utilizzare Matlab come ambiente di calcolo. I metodi verranno implementati con la medesima matrice in input in modo tale da poter valutare pregi e difetti. Qualora necessario specificare a priori il numero dei cluster, per semplicità di rappresentazione negli esempi



(a) Dendrogramma (b) Silhouette-plot

Figura 1.2: Rappresentazione dei dati

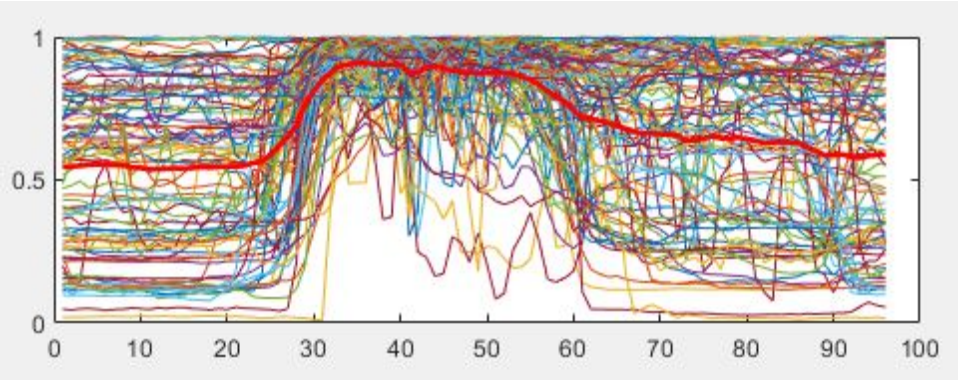


Figura 1.3: DataSet

presentati viene scelto un numero di cluster pari a 6. Il dataset, figura 1.3, è una matrice contenente le curve di carico di 100 utenti nell'arco di un giorno, suddivisi in 96 quarti d'ora.

Il codice Matlab è il seguente:

Codice Clustering 1: Caricare dati da matrice

```
1 load data_matrix.txt;
```

1.5.1 Metodi di classificazione gerarchica

I **metodi gerarchici** si affiancano ad una situazione in cui si hanno n grappoli di una sola unità per giungere, attraverso successive fusioni dei grappoli meno distanti tra di loro, ad una situazione in cui si ha un solo grappolo che contiene tutte le n unità.

L'output di questo tipo di clusterizzazione non è un numero predefinito di gruppi, ma un dendrogramma che contiene al suo interno l'intera serie di fusioni dei vari grappoli. Sull'asse delle ordinate viene riportato il livello di distanza e sull'asse delle ascisse le unità. Ogni ramo del diagramma corrisponde ad un grappolo, la linea di congiunzione tra due o più rami individua il livello di distanza al quale i grappoli si fondono [16] [9].

Gli algoritmi gerarchici proposti in letteratura si differenziano unicamente per il diverso criterio che regola la valutazione delle distanze tra i gruppi ai fini delle aggregazioni in serie.

L'implementazione avviene tramite il codice:

Codice Clustering 2: Eseguire metodi gerarchici

```
1 Y = pdist(data_matrix, 'euclidean');
2 Z = linkage(Y, 'average');
3 vettore_hierarchical = cluster(Z, 'mazclust', numero_clusters);
```

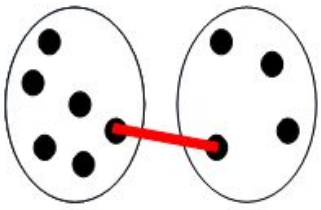
Da notare che la seconda riga definisce il tipo di legame con cui il metodo procederà: *'single'* per legame singolo, *'complete'* per legame completo, *'average'* per legame medio, *'centroid'* per metodo del centroide.

Come esempio verrà solamente implementato il metodo del legame singolo.

Metodo del legame singolo o delle unità più vicine

Il **metodo del legame singolo**, chiamato anche *single linkage* o del *vicino più vicino*, definisce la distanza tra due gruppi come il minimo delle distanze tra ciascuna delle unità di un gruppo e ciascuna delle unità dell'altro gruppo.

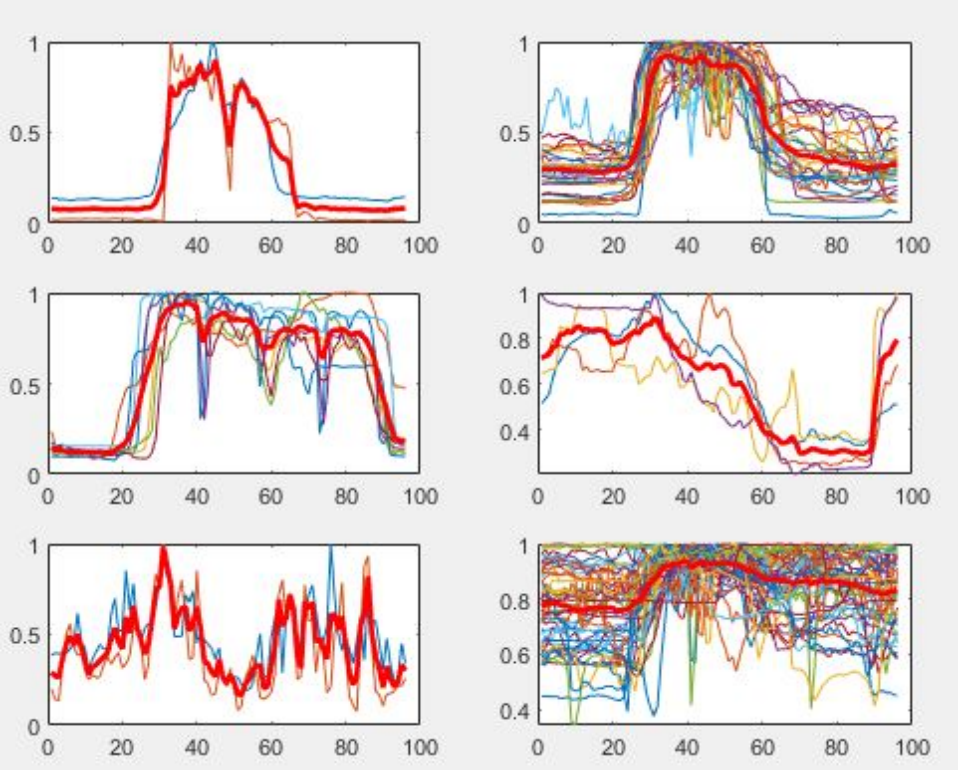
Definito C_1 il cluster numero 1, C_2 il cluster numero 2, d_{ij} la distanza tra l'elemento i-esimo $\in C_1$ e l'elemento j-esimo $\in C_2$, allora la distanza tra i gruppi è $d(C_1, C_2) = min(d_{ij})$.



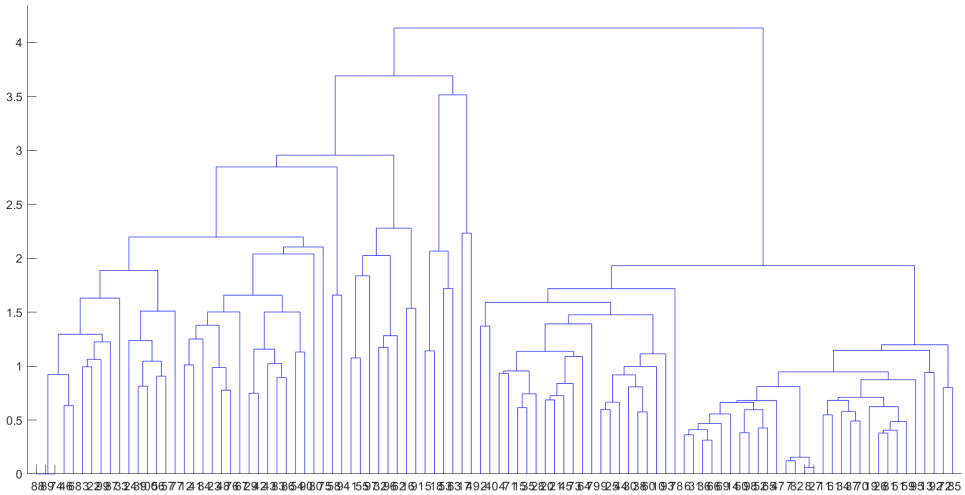
Questo algoritmo evidenzia le similitudini tra gli elementi, privilegia la differenza tra i gruppi piuttosto che l'omogeneità degli elementi di ogni cluster. Il dendrogramma (figura 1.4b) avrà rami corti e sarà più compatto in quanto vengono valorizzate le somiglianze.

Metodo del legame completo

Il **metodo del legame completo**, detto anche *complete linkage* o del *vicino più lontano*, definisce la distanza tra due gruppi come il massimo delle m1



(a) Classificazione

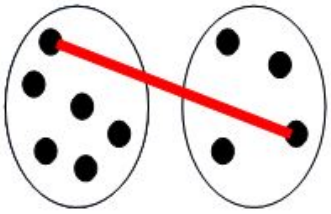


(b) Dendrogramma

Figura 1.4: Output matlab per metodo del legame singolo

m2 distanze tra ciascuna della unità di un gruppo e ciascuna delle unità dell'altro gruppo.

Definito C_1 il cluster numero 1, C_2 il cluster numero 2, d_{ij} la distanza tra l'elemento i-esimo $\in C_1$ e l'elemento j-esimo $\in C_2$, allora la distanza tra i gruppi è $d(C_1, C_2) = max(d_{ij})$.

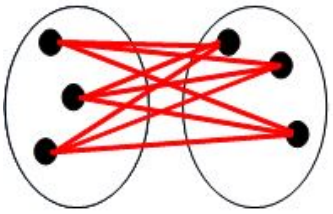


Concettualmente, questo algoritmo, si pone all'opposto di quello precedente: privilegia l'omogeneità degli elementi di ogni gruppo a scapito della differenza netta tra i cluster. Il dendrogramma avrà rami lunghi con distanze maggiori tra essi.

Metodo del legame medio

Il **metodo del legame medio**, o *average linkage*, definisce la distanza tra due gruppi come la media aritmetica delle m_1 m_2 distanze tra ciascuna delle unità di un gruppo e ciascuna delle unità dell'altro.

Definito C_1 il cluster numero 1, C_2 il cluster numero 2, m_1 il numero di elementi apparteneti a C_1 e m_2 il numero di elementi appartenenti a C_2. Se d_{ij} è la distanza tra l'elemento i-esimo $\in C_1$ e l'elemento j-esimo $\in C_2$, allora la distanza tra i gruppi è $d(C_1, C_2) = \frac{1}{m_1 m_2} \sum_i \sum_j d_{ij}$.

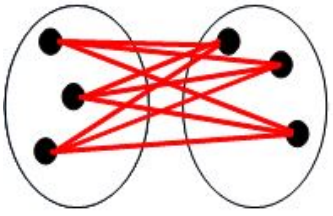


Il metodo del legame medio costituisce un compromesso ragionevole tra una discreta coesione interna e separazione esterna, infatti i gruppi risulteranno più omogenei e ben differenziati tra loro.

Metodo del centroide

Si definisce **centroide** un punto nello spazio corrispondente al punto medio degli elementi appartenenti al cluster. Il metodo considera come distanza tra i gruppi la distanza tra i detti centroidi. Da notare che questi possono non coincidere con uno degli elementi del cluster.

Definito C_1 il cluster numero 1, C_2 il cluster numero 2, $\overline{x_{C1}}$ centroide di C_1 e $\overline{x_{C2}}$ centroide di C_2, allora la distanza tra i gruppi è $d(C_1, C_2) = d(\overline{x_{C1}}, \overline{x_{C1}})$.



1.5.2 Metodi di classificazione non gerarchica

Questi metodi puntano a ripartire direttamente le n unità in r grappoli, fornendo come prodotto finale una sola partizione delle n unità [2] [20], quindi sono metodi che forniscono una partizione dei dati in un numero di gruppi definito a priori.

Gli algoritmi non gerarchici si articolano sostanzialmente in due fasi:

- Inizializzazione dell'algoritmo, indicando G cluster di partenza intorno a cui partizionare le n unità;
- spostamento successivo delle unità tra i G gruppi, in modo da ottenere la partizione che meglio risponde ai concetti di omogeneità interna ai gruppi e di eterogeneità tra gli stessi.

A differenza dei metodi gerarchici, l'assegnazione di un oggetto ad un cluster non è irrevocabile. Ovvero le unità vengono riassegnate ad un diverso cluster se l'allocazione iniziale risulta inappropriata.

L'individuazione della partizione ottimale per ogni singolo elemento comporterebbe l'esaminazione di tutte le possibili assegnazioni delle n unità ai G gruppi, ma un'operazione di questo tipo implicherebbe una grande mole di calcoli. Le procedure non gerarchiche propongono, dunque, la risoluzione del problema tramite strategie di raggruppamento che valutano un numero accettabile di possibili partizioni alternative. In pratica, una volta scelta la partizione iniziale, si procede a riallocare le unità in esame tra i diversi gruppi in modo da ottimizzare la prefissata funzione obiettivo. Per questi motivi il raggiungimento dell'ottimo globale non è garantito.

K-means

L'**algoritmo K-means** è un algoritmo partizionale che permette di suddividere un insieme di oggetti in k gruppi sulla base delle loro similarità. Il metodo segue una procedura iterativa il cui scopo è minimizzare la varianza totale intra-cluster. Ogni gruppo viene identificato mediante il centroide. La scelta dei centroidi iniziali è un fattore che influenza molto il risultato finale, in quanto questo metodo non garantisce il raggiungimento dell'ottimo, come detto precedentemente, ma può bloccarsi in un punto di ottimo locale.

Il procedimento può essere sintetizzato in 4 punti:

1. Inizializzazione: si definiscono i parametri di input. L'algoritmo crea casualmente, o usando alcune informazioni euristiche, k centroidi ai quali corrispondono k partizioni con ampiezze scelte a priori o casuali.
2. Assegnazione del cluster: l'algoritmo calcola la distanza di ogni elemento da ogni centroide: se la distanza minima non è ottenuta in corrispondenza del centrode del gruppo di appartenenza, allora l'unità è riallocata al gruppo che corrisponde al centroide più vicino;

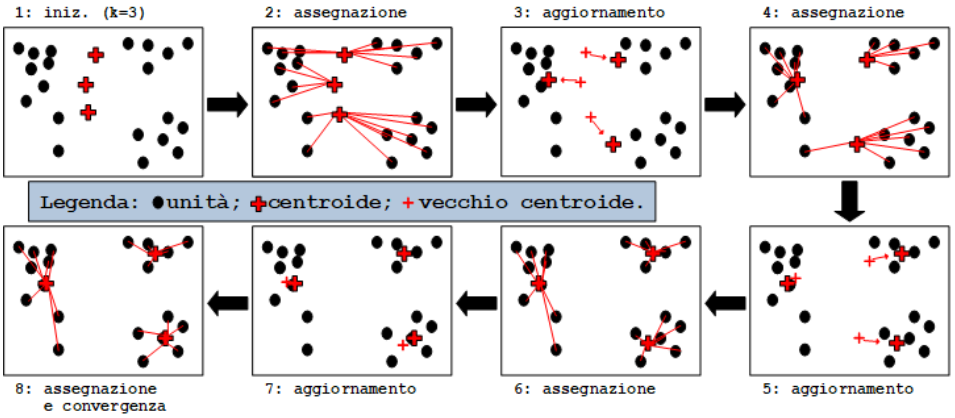


Figura 1.5: esempio di K-means convergente in 8 step

3. Aggiornamento del centroide: calcola il centroide di ogni nuovo gruppo;
4. Ripete i passaggi 2 e 3 finché l'algoritmo non converge.

Come misura di distanza tra l'unità i ed il centroide viene normalmente utilizzata la distanza euclidea in quanto garantisce la convergenza della procedura iterativa.

La condizione di stop, solitamente, è scelta tra le seguenti opzioni:

- nessun dato cambia cluster per un determinato numero di iterazioni;
- la somma delle distanze è minore o uguale ad un valore prestabilito;
- viene raggiunto un numero massimo di iterazioni.

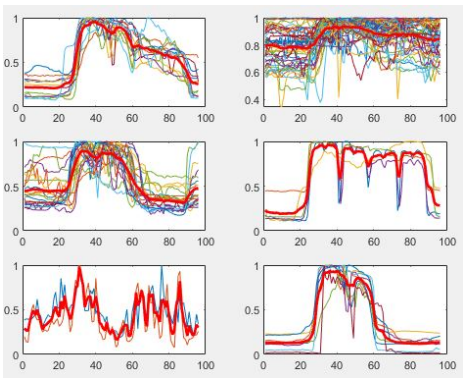
L'implementazione avviene tramite il codice:

Codice Matlab 3: Eseguire K-means

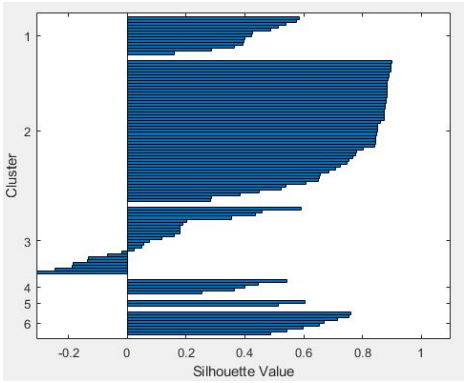
```
1 vettore_kmeans = kmeans(data_matrix, numero_clusters);
```

Il k-means ha il vantaggio di essere abbastanza veloce, dati i pochi calcoli necessari, con conseguente riduzione anche del tempo di elaborazione da parte del computer che svolgerà tali calcoli; si è osservato infatti che generalmente il numero d'iterazioni è minore del numero delle unità. Gli svantaggi nell'adottare questo metodo sono la casualità nello step di inizializzazione, ma principalmente il dover conoscere, a priori, la quantità di cluster in cui raggruppare i dati.

I risultati ottenuti sono mostrati in figura 1.6.



(a) Classificazione



(b) Silhouette

Figura 1.6: Output matlab per K-means

Fuzzy clustering

Il **clustering fuzzy**, noto anche come *soft clustering* o *soft k-means*, è un algoritmo di cluster analysis non esclusivo, ovvero in cui le unità possono appartenere contemporaneamente a più cluster.

Dati n punti, l'obiettivo è suddividerli in partizioni distinte. La particolarità rispetto al clustering standard è che questo approccio consente di determinare quali sono i gradi di appartenenza dell'i-esimo elemento rispetto a ciascuno dei cluster considerati.

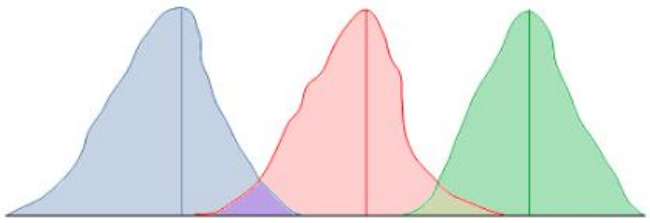


Figura 1.7: esempio Fuzzy Clustering con 3 cluster

L'algoritmo opera assegnando in maniera casuale un grado di **membership** μ_{ij} a ciascuna unità rispetto ad ognuno dei cluster e ricavando di conseguenza una certa distribuzione spaziale iniziale dei centri di massa delle partizioni che si vogliono ricavare. In maniera simile al k-means, tramite un procedimento iterativo, la funzione muove dinamicamente i baricentri verso la localizzazione ottimale andando a minimizzare una funzione obiettivo tramite la somma delle distanze, solitamente quella euclidea, di ciascun elemento da ciascun centro di massa, opportunamente pesate col corretto grado di membership.

Definito x_i come l'i-esimo elemento di un cluster e $\overline{x_j}$ il centro di massa del medesimo cluster, allora procedimento può essere sintetizzato in 5 punti:

1. Inizializzazione random di tutti i valori di membership μ_{ij};
2. Calcolo dei centri di massa dei cluster:

$$\overline{x_j} = \frac{\Sigma_{i=1}^{D} \mu_{ij} x_i}{\Sigma_{i=1}^{D} \mu_{ij}}$$

3. Aggiornamento di tutti i nuovi μ_{ij}:

$$\mu_{ij} = \frac{1}{\Sigma_{k=1}^{N} (\frac{|x_i - \overline{x_j}|}{|x_i - \overline{x_j}|})^2}$$

4. Calcolo della funzione obiettivo J;
5. Si ripetono i passi 2, 3 e 4 fino a che non si raggiunge un miglioramento pressoché nullo della funzione J oppure si raggiunge il numero massimo di iterazioni scelto a priori.

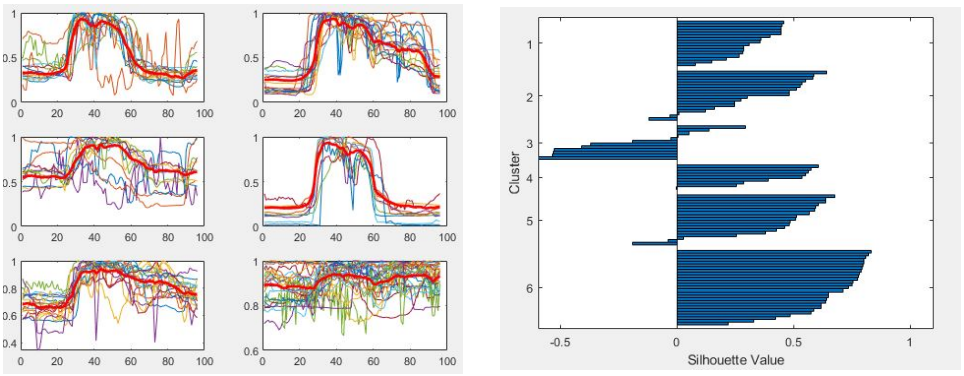
In Matlab è possibile eseguire questo metodo usando la riga:

Codice Matlab 4: Eseguire Fuzzy clustering

```
[centri, membership] = fcm(data_matrix, numero_clusters);
```

Questo metodo ha il vantaggio di essere abbastanza veloce ma mantiene lo svantaggio, già presente nel k-means, di dover conoscere, a priori, la quantità di cluster in cui raggruppare i dati.

I risultati sono mostrati in figura 1.8.



(a) Classificazione (b) Silhouette

Figura 1.8: Output matlab per Fuzzy clustering

Density-based clustering

Il **density-based clustering**, chiamato anche *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN), è un metodo basato sulla nozione di *density-reachability* (raggiungibilità per densità) all'interno di un cluster, questo infatti raggruppa regioni di punti con densità sufficientemente alta. L'algoritmo stima la densità attorno a ciascun punto contando il numero di elementi in un intorno ε specificato dall'utente, ed applica delle soglie chiamate *minPts* per identificare i punti *core*, *border* e *noise*.
Le regole per l'identificazione sono:

- Un punto p è un punto *core* se almeno *minPts* punti sono entro la distanza ε da esso. Questi punti sono density-reachable da p, ovvero direttamente raggiungibili;
- Un punto q è density-reachable da p se il punto q è entro la distanza ε dal punto p, con p che deve essere un punto *core*;
- Un punto q è density-reachable da p se esiste un percorso $p_1, \ldots, p_n$ con $p_1 = p$ ed $p_n = q$, dove ogni p_{i+1} è direttamente raggiungibile da p_i (tutti i punti nel percorso devono essere *core*, con la sola possibile eccezione di q che in questo caso diventa *border*);
- Tutti i punti non raggiungibili da altri punti sono punti *noise*.

Da notare che la raggiungibilità non è una relazione simmetrica poiché, per definizione, un punto non può essere raggiungibile da un punto *non-core*. Pertanto, è necessaria un'ulteriore nozione di connessione per definire formalmente l'estensione dei cluster trovati da DBSCAN. Due punti p e q sono *density-connected*, questa volta simmetrica, se esiste un punto r tale che sia p che q siano raggiungibili da r.

Un cluster nel density-based clustering quindi soddisfa due proprietà: tutti i punti del cluster sono mutuamente density-connected; se un punto è density-raggiungibile da un qualunque punto del cluster, allora è parte del cluster stesso.

Nell'esempio in figura 1.9, è stato posto $minPts = 4$. I punti rossi sono punti *core* poiché l'area che li circonda, con raggio di ampiezza *varepsilon*, contiene almeno 4 punti (incluso il punto stesso). Poiché i punti rossi sono tutti density-reachable tra loro, questi formano un singolo cluster. I punti blu non sono punti *core*, ma sono raggiungibili dai punti *core* (attraverso altri punti *core*) e quindi appartengono a loro volta al cluster. Il punto verde è un punto di rumore (*noise*) poiché non è un punto *core* e non è nemmeno un punto density-reachable.

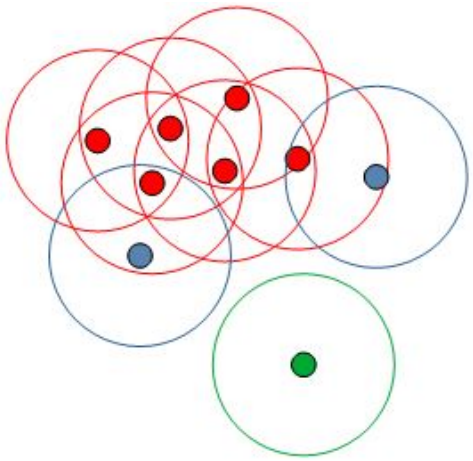


Figura 1.9: esempio schematico di DBSCAN

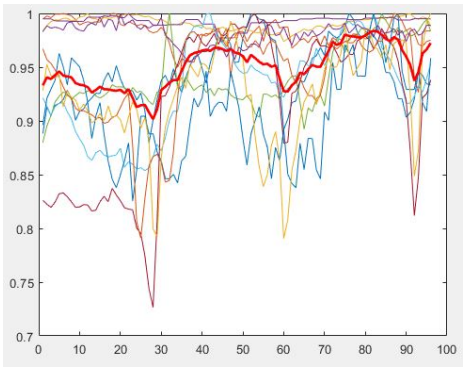
Il codice matlab è composto da poche righe:

Codice Matlab 5: Eseguire DBSCAN

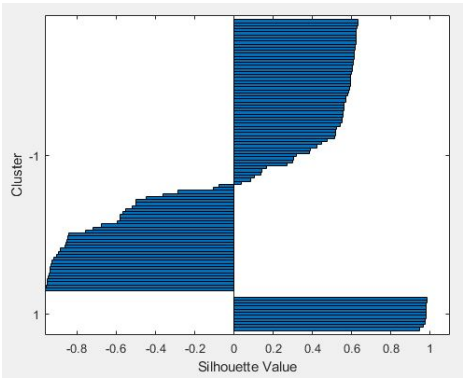
```
1 minPts = 6;
2 epsilon = 0.5;
3 [idx, corepts] = dbscan(data_matrix, epsilon, minPts);
```

Usare questo algoritmo offre innumerevoli vantaggi, tra cui non richiede la conoscenza del numero di cluster a priori, a differenza di quanto avviene con il K-means; può trovare cluster di forme arbitrarie e riesce anche ad individuare cluster completamente circondati da altri cluster differenti, a cui non è connesso ed inoltre possiede la nozione di rumore.

In questo caso è stato trovato solamente un cluster ma con un alto coefficiente di silhouette (figura 1.10).



(a) Classificazione



(b) Silhouette

Figura 1.10: Output matlab per DBSCAN

Capitolo 2

Non-intrusive load monitoring

Questo capitolo riporterà le definizioni generali che riguardano il monitoraggio intrusivo e non intrusivo, al fine di fornire al lettore una panoramica di questo campo di ricerca. Vengono presentate le categorie generali e la struttura degli algoritmi, oltre alle funzionalità proposte in letteratura per rilevare i carichi. Verranno anche presentate le metriche di valutazione più comuni e descritti alcuni set di dati disponibili pubblicamente per la ricerca NILM e nell'ultima parte viene fornita una descrizione di alcuni algoritmi significativi.

2.1 Tecniche di monitoraggio

Si definisce **monitoraggio del carico** un processo di identificazione e acquisizione della misura del carico in un sistema di alimentazione [1], con l'obiettivo di determinare il consumo e lo stato degli apparecchi, al fine di comprendere il comportamento dei singoli carichi all'intero dell'intero sistema.

Il monitoraggio del carico può trovare applicazione anche per conseguire un risparmio energetico, ad esempio intervenendo sulla tempistica dell'utilizzo degli elettrodomestici, l'ottimizzazione del loro funzionamento e l'eliminazione delle attività non necessarie. Potrebbe rilevarsi utile, inoltre, illustrare all'interno della fatturazione non solo l'importo che va pagato, ma anche il consumo di ogni elettrodomestico e quanto ciascun apparecchio ha contribuito al raggiungimento di tale cifra. Tale indicazione non solo contribuirebbe a rilevare eventuali malfunzionamenti degli elettrodomestici, ma sensibilizzerebbe il consumatore alle tematiche ambientali rendendolo più consapevole del proprio impatto [5].

Con l'aumento delle micro reti e con la continua crescita delle installazioni ad energia rinnovabile è quindi necessario raccogliere quante più misure energetiche per monitorare, automatizzare e gestire il sistema elettrico.

A seconda del metodo utilizzato, il monitoraggio può essere ILM (monitoraggio del carico intrusivo) o $NILM$ (monitoraggio del carico non intrusivo).

2.1.1 Intrusive Load Monitoring

Con l'espressione **Intrusive Load Monitoring** (ILM) si intendono tutti quegli approcci che propongono di distribuire un dispositivo di misurazione per ogni elettrodomestico o carico di interesse. La necessità di diversi dispositivi di misurazione nell'ecosistema ILM rende costosa e difficile la manutenzione, l'installazione e l'espansione. Il termine 'intrusive' indica che il dispositivo di misurazione si trova nell'abitazione, vicino all'apparecchio da monitorare. In base al livello di intrusione, si definiscono 3 sottoclassi ILM [28]:

1. **ILM 1** si basa su sub-contatori che misurano tipicamente il consumo, di una zona della casa, posizionandolo a livello dell'interruttore.
2. **ILM 2** utilizza dispositivi di misurazione posizionati a livello di presa, in modo che un dispositivo possa monitorare uno o più dispositivi contemporaneamente.
3. **ILM 3** utilizza dispositivi di misurazione posti a livello dell'elettrodomestico.

Le ragioni sopra spiegate, hanno portato all'introduzione di una variante non intrusiva del metodo con una consistente riduzione dei costi annessi.

2.2 NILM

Il **monitoraggio non intrusivo del carico**, in inglese *Non-intrusive load monitoring* (NILM), è un processo di stima del consumo energetico dei singoli apparecchi a partire da misurazioni di potenza elettrica, un solo dispositivo di misurazione per ogni ingresso di energia in casa. Tale tipologia di monitoraggio si definisce "non intrusiva" poiché non necessita di misurazioni sui singoli carichi, ma mira a ricavarle partendo dalle misurazioni di aggregati effettuate in un numero ridotto di punti di un impianto elettrico di un edificio, trattasi infatti di un solo dispositivo di misurazione per ogni ingresso di energia in casa.

Il monitoraggio non intrusivo del carico viene anche definito con l'espressione *disaggregazione dell'energia*. Le tecniche NILM, inoltre, sono da prediligere rispetto a quelle ILM grazie alla loro installazione che risulta più facile e più economica.

Da notare che, elettricamente, l'unicità dei transitori di avvio è la chiave per la disaggregazione mentre lo stato stazionario delle apparecchiature può

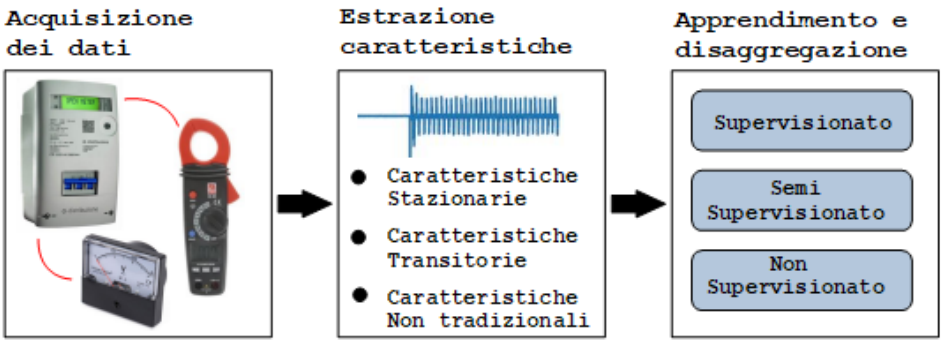


Figura 2.1: Diagramma di flusso generale dei processi di NILM

essere utilizzato per determinare quando e quali carichi sono attivi. I dati disaggregati trovano applicazione anche nella diagnostica e quindi al fine di individuare e riconoscere i guasti.

2.3 Processi

Un processo NILM si presenta come un problema di classificazione di serie temporali in cui si deve rilevare quali apparecchi sono attivi in un determinato istante di tempo e quanto ciascuno di essi contribuisce sul consumo totale. La figura 2.1 mostra un diagramma di flusso generale che descrive il processo NILM. In questa sezione verrà descritta ogni parte di questo processo.

2.3.1 Acquisizione dei Dati

Le misure basilari da acquisire sono:

- tensione ΔV, misurata in volt (V);
- corrente I, misurata in ampere (A);
- potenza apparente S, misurata in voltampere (VA), prodotto della corrente per la tensione.

La misura della **potenza attiva** P, ovvero il trasferimento di energia nella rete indipendente dalla direzione, misurata in Watt (W), si chiama anche *potenza media*, perché viene praticamente determinata misurando l'energia in un intervallo di tempo e dividendo l'energia per la durata dell'intervallo di tempo. Altre misure di interesse derivate dalle precedenti sono: il **fattore di potenza** $cos(\Theta)$, dove Θ è l'angolo tra tensione e corrente; la **potenza reattiva** Q, misurata in volt-ampere reattivi (var). Infine il consumo di energia è la quantità di energia consumata nel tempo, kilowattora (kWh), ovvero la misura che compare all'interno delle bollette elettriche. In una visione

più semplicistica delle tecniche NILM, la disaggregazione serve a suddividere quest'ultima quantità totale per ogni elettrodomestico.

Affiancate a queste quantità elettriche, un dato necessario è la **frequenza di campionamento** f_c, misurata in Hz, dei dati raccolti che determina il tipo di informazione che potrebbe essere estratta dai segnali elettrici. È possibile campionare in due modi, con:

- **Alta frequenza di campionamento**: i dati vengono raccolti ad una frequenza di campionamento di 1 Hz o superiore. Questo tipo di campionamento permette di estrarre alcune caratteristiche nel consumo che sono visibili solo a queste frequenze di campionamento.
- **Bassa frequenza di campionamento**: campionamento con frequenze inferiori a 1 Hz. Le misure più comuni al giorno d'oggi per questioni di costi.

I dati raccolti vengono archiviati in database. In letteratura ci sono diversi database di riferimento per testare diversi algoritmi. Alcuni di loro sono PLAID e WHITED che verranno introdotti nel capitolo 3 di questa tesi.

2.3.2 Estrazione delle informazioni e delle caratteristiche

Dopo aver raccolto i dati, il passaggio successivo è l'estrazione di informazioni sulle serie temporali elettriche al fine di ottenere caratteristiche che consentono di rilevare eventi come le transizioni di stato. Le caratteristiche possono essere classificate come:

- **Caratteristiche di stato stazionario**: derivano dal funzionamento a stato stazionario dell'elettrodomestico. Le variazioni di potenza attiva P e reattiva Q sono comunemente usate per rilevare i cambiamenti nello stato di funzionamento degli apparecchi. Le caratteristiche relative alla sola potenza attiva possono essere estratte a bassa frequenza di campionamento e utilizzate per rilevare apparecchiature con caratteristiche di assorbimento di corrente molto diverse. Altre caratteristiche, come le armoniche di corrente, che verranno utilizzate nel metodo proposto nella parte II, sono più efficienti ma richiedono un campionamento ad alta frequenza per essere osservate.
- **Caratteristiche di stato transitorio**: queste derivano dal funzionamento in stato transitorio e sono molto variabili da elettrodomestico ad elettrodomestico. Necessitano tuttavia una frequenza di campionamento elevata. Le principali sono picchi di corrente, tempo di risposta ai transitori, profili di potenza nei transitori, inviluppi spettrali.

- **Caratteristiche non tradizionali**: solitamente non di natura elettrica, come ora del giorno, frequenza di utilizzo di un elettrodomestico e la correlazione di utilizzo tra più apparecchi.

2.3.3 Apprendimento e disaggregazione

Ottenute le caratteristiche di interesse, si dovrà determinare quali elettrodomestici sono accesi in un dato momento. Queste tecniche possono essere classificate come:

- **Metodi supervisionati**, tutti quei metodi in cui è nota la soluzione nell'insieme di dati di addestramento ed il supervisore, una persona, fornisce alla macchina esempi noti dove, in ognuno di essi, sono indicate le variabili di input e l'output corretto dando modo a questa di imparare ed elaborare un modello predittivo. Gli alberi decisionali, su cui è incentrato il metodo proposto nella parte II, sono una di queste tecniche.
- **Metodi semi-supervisionati**, mirano a utilizzare una piccola quantità di dati di allenamento etichettati insieme ad una grande quantità di dati di allenamento senza etichetta. Ciò si verifica spesso in situazioni reali in cui l'etichettatura, o soluzione, dei dati è molto costosa e/o si dispone di un flusso costante di dati.
- **Metodi non supervisionati**, in questi metodi i dati non sono etichettati, per imparare la macchina deve estrarre le informazioni rilevanti dai dati disponibili. La cluster analysis, nel capitolo 1 di questo documento, è una delle principali tecniche facente parte di questa tipologia.

2.4 Tecniche e algoritmi

Questa sezione è una rassegna delle tecniche e degli algoritmi maggiormente utilizzati per il NILM.

2.4.1 Deep Learning

> *Il Deep Learning, in italiano* ***apprendimento approfondito****, è il ramo più avanzato del Machine Learning. Si tratta di un insieme di tecniche basate su reti neurali artificiali organizzate in diversi strati:* [1] *ogni strato calcola i valori per quello successivo, in modo da elaborare l'informazione in maniera sempre più completa.*

[1] Osservatorio Artificial Intelligence del Politecnico di Milano, https://blog.osservatori.net/it_it/deep-learning-significato-esempi-applicazioni

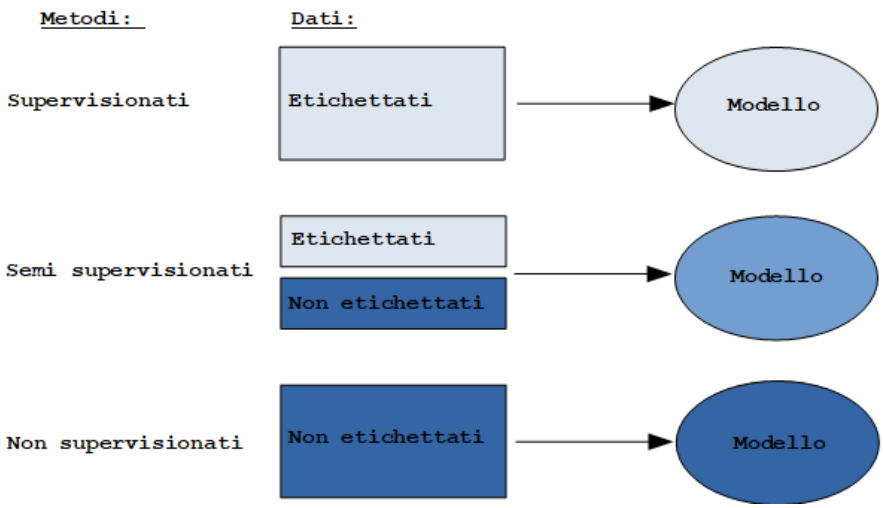


Figura 2.2: Esempio grafico della convoluzione

L'obiettivo dell'utilizzo di questo tipo di architetture è conoscere una gerarchia di funzionalità. Ogni livello elabora un qualche tipo di input, elabora e apprende da esso, per fornire una migliore rappresentazione dei dati all'input del livello successivo, aumentando esponenzialmente il numero di possibili rappresentazioni di stato [24]. Questo metodo computazionale è preso in prestito dalla capacità del cervello umano di osservare, analizzare, apprendere e prendere decisioni, soprattutto per problemi estremamente complessi. Uno dei principali vantaggi di queste rappresentazioni è che possono essere invarianti rispetto alle modifiche locali avvenute nei dati di input. Imparare dalle caratteristiche invarianti è uno degli obiettivi principali nelle attività di riconoscimento dei modelli come quelle necessarie nel campo NILM.

Queste tecniche hanno avuto un gran successo e sviluppo negli ultimi anni grazie al recente superamento di molti problemi di natura tecnica. Inoltre, la crescita esponenziale della potenza di elaborazione e la riduzione significativa del costo delle *graphics processing unit* (GPU), ovverò l'unità di elaborazione grafica, in gergo scheda video, progettata appositamente per la creazione di grafica digitale, rende questi dispositivi più disponibili e utilizzabili per addestrare questo tipo di architetture in tempi più brevi.

Il processo di disaggregazione avviene tramite l'utilizzo di una finestra temporale scorrevole lungo la sequenza di input. Pertanto, la prima sequenza di input per la rete sarà zero. La finestra di input verrà spostata di K campioni, dove $K \geq 0$. Se K è inferiore alla lunghezza della dimensione del livello, o strato, di input della rete, vedrà sequenze di input sovrapposte. Questo comportamento consente alla rete di elaborare gli stessi valori in più tentativi e di rilevare in modo migliore l'accensione dell'elettrodomestico.

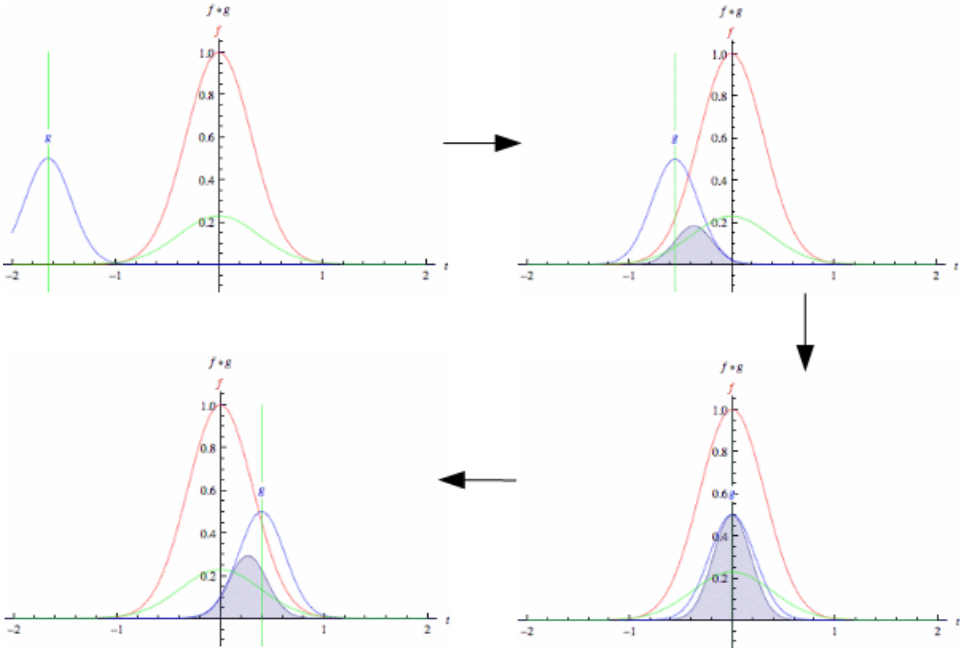


Figura 2.3: Esempio grafico di convoluzione, in rosso: funzione f, in blu: funzione g, in verde: prodotto di convoluzione $f*g$

Rete Neurale Convoluzionale

Le **Reti Neurali Convoluzionali** (CNN), chiamate anche *ConvNet* [23], sono uno degli algoritmi di Deep Learning più utilizzati oggi. Questi trovano applicazione in molteplici campi: dalla computer vision alle automobili autonome, dalle diagnosi mediche al supporto e trattamento per gli ipovedenti. Queste funzionano come tutte le reti neurali: un layer di input, uno o più layer nascosti, che effettuano calcoli tramite funzioni di attivazione, e un layer di output con il risultato ma differiscono dalle classiche reti neurali in quanto mettono al loro servizio il concetto di convoluzione.

Nell'analisi funzionale, la **convoluzione** è un'operazione tra due funzioni di una stessa variabile che consiste nell'integrare il prodotto tra la prima e la seconda traslata di un certo valore. Un esempio grafico di convoluzione è mostrato in figura 2.3, dove la funzione g, di colore blu, 'scorre' sulla funzione f, in rosso, 'mescolandosi'ed ottenendo il loro prodotto per convoluzione $f*g$, in verde.

Nelle ConvNet ogni livello ospita quella che viene chiamata **feature map**, ovvero la specifica caratteristica che i neuroni si preoccupano di cercare, questa è l'output di un filtro applicato allo strato precedente. Questo filtro non è altro che la funzione g vista precedentemente, l'input è la funzione f e la feature map è il prodotto di convoluzione.

Esistono diversi tipi di neuroni di convoluzione a seconda della dimen-

sione. Per il NILM vengono applicati neuroni monodimensionali in quanto i dati di input sono anch'essi di natura unidimensionale.

Reti Neurali Ricorrenti

Il caratteristica propria delle **reti neurali ricorrenti** (RNN) [29] è quella di seguire la sequenza delle informazioni. Nelle reti neurali tradizionali, non si è stati in grado di considerare i diversi input e output collettivamente, anche quando un'informazione è connessa ad altre, la si considera singolare. Le reti neurali ricorrenti si possono considerare come una memoria che raccoglie e memorizza informazioni su ciò che il sistema ha calcolato fino a quel momento. Il termine ricorrente specifica il fatto che la rete può elaborare mantenendo i dati in sequenza.

L'algoritmo più eseguito per questa tipologia di reti si chiama **Long short-term memory** (LSTM), ossia rete di memoria a lungo termine. Questa è stata progettata per superare il problema della dipendenza a lungo termine affrontato dalle RNN. Gli LSTM hanno connessioni di feed back che li rendono diversi dalle reti neurali più tradizionali, questa proprietà consente agli LSTM di elaborare intere sequenze di dati senza trattare ogni punto della sequenza in modo indipendente, ma piuttosto conservando informazioni utili sui dati precedenti nella sequenza per aiutare con l'elaborazione di nuovi punti di dati. DI conseguenza tale tipologia di algoritmo si rivela particolarmente efficace se applicata nell'elaborazione di sequenze di dati come testo, parlato e serie temporali generali.

Nel campo NILM, le architetture basate su LSTM sono state applicate con successo nella disaggregazione energetica raggiungendo punteggi di precisione fino all'80% ma con prestazioni leggermente peggiori delle reti convoluzionali.

2.4.2 Hidden Markov Models

L'**Hidden Markov Models** (HMM) [18] è un approccio statistico che può modellare serie temporali e rappresentare gli stati non osservabili di queste. In un approccio HMM lo stato del modello è nascosto, non è direttamente visibile all'osservatore, mentre l'output è visibile e dipendente da quello nascosto.

In NILM lo stato nascosto è lo stato di tutti gli elettrodomestici, ogni possibile combinazione dei loro possibili stati, l'output è il consumo aggregato. Ogni stato nascosto ha una distribuzione di probabilità relativa a tutti i possibili output, dunque la sequenza di output fornisce informazioni sulla sequenza degli stati nascosti. Un HMM comune può essere definito come [19]:

$$\gamma = \{S, O, P_0, \mathbf{A}, \mathbf{B}\}$$

dove:

- S è l'insieme dei possibili stati;
- O è l'insieme delle osservazioni;
- P_0 le probabilità iniziali;
- **A** la matrice di transizione;
- **B** la matrice di emissione.

in cui:

- il numero totale di stati e osservazioni sono rispettivamente le cardinalità $K = |S|$ e $N = |O|$;
- **A** definisce la probabilità di transizione da uno stato iniziale a quello successivo. È una matrice $K \times K$, dove $\sum_i \mathbf{A}[i, j] = 1$;
- **B** definisce la probabilità di rilevare una particolare osservazione allo stato successivo. È una matrice $K \times N$, dove $\sum_i \mathbf{B}[i, j] = 1$.

2.4.3 Decision Tree

Gli **alberi decisionali**, in inglese *Decision tree* (DT), costituiscono un metodo di apprendimento supervisionato non parametrico utilizzato per la classificazione e la regressione . L'obiettivo è creare un modello che preveda il valore di una variabile obiettivo apprendendo semplici regole decisionali dedotte dalle caratteristiche dei dati. Un albero può essere visto come un'approssimazione costante a tratti. Questo argomento, ai fini di questa tesi, verrà ampliamente spiegato nel capitolo 4.3 a pag 47.

Capitolo 3

Dataset

Al fine di rendere possibile la valutazione delle tecniche NILM, sono stati messi a disposizione dei ricercatori di tutto il mondo alcuni dataset. Questo capitolo riporta le principali caratteristiche generali di questi set di dati pubblici presenti in letteratura. Ai fini del metodo sviluppato nella parte II, verranno approfonditi 2 dataset: PLAID e WHITED. Questi, insieme al COOLL [25], sono i più utilizzati per quanto riguarda tecniche NILM basate su eventi e non, grazie alla loro versatilità. Contengono dati dei transitori di avvio e le tracce spettrali di diversi dispositivi sia individuali che aggregati, con un'alta frequenza di campionamento.

3.1 Definizione e generalità

> ***Dataset*** *In informatica, un insieme di dati organizzati in forma relazionale.* [1]

Nell'ambito della clusterizzazione e della disaggregazione dell'energia, si definisce **Dataset** una raccolta di misurazioni di energia elettrica prese da scenari del mondo reale, senza interrompere la routine quotidiana nello spazio monitorato, ovvero cercando di mantenere i dati il più vicino possibile alla realtà. Solitamente contengono misurazioni di consumo aggregato e/o di singoli carichi, che si ottengono misurando ogni carico a livello di spina (*plug*) o misurando il singolo circuito a cui il carico è connesso. In uno scenario reale, in genere più carichi sono collegati allo stesso circuito. Pertanto, la misurazione a livello di *plug* non garantisce sempre la disponibilità dei dati di consumo individuali per ogni carico.

I dataset finalizzati alle tecniche NILM attualmente disponibili in letteratura sono 26. Questi possono anche essere classificati come set di dati

[1] Dizionario di Economia e Finanza Treccani, https://www.treccani.it/enciclopedia/data-set_%28Dizionario-di-Economia-e-Finanza%29/

basati su eventi o senza eventi. Generalmente i dataset campionati ad alta frequenza vengono considerati con eventi in quanto danno la possibilità di osservare, con buona risoluzione, gli istanti in cui avvengono i transitori. [2]

La maggior parte di tali set, per l'esattezza 21, sono adatti per valutare approcci privi di eventi, mentre solo 5 possono essere utilizzati per valutare approcci basati su eventi questo perché la raccolta di dati per approcci privi di eventi è più semplice e richiede meno tempo rispetto alla sua concorrente. La tabella 3.1 riassume questi set di dati fornendo le seguenti caratteristiche per ognuno di essi:

- anno e Paese di rilascio;
- eventuale utilizzabilità in approcci basati su eventi (CE) o senza eventi (SE);
- tipologia di dati contenuti, ovvero aggregati (A), circuiti individuali (CI) o dispositivi individuali (DI) e se gli eventi sono etichettati (E);
- grandezze elettriche disponibili, cioè corrente (I), tensione (V), potenza attiva (P), potenza reattiva (Q), potenza apparente (S).

3.2 Plug-Load Appliance Identification Dataset

Il **PLAID**, Plug-Load Appliance Identification Dataset, è un dataset pubblico contenente il campionamento ad alta frequenza (30 kHz) dei valori di tensione e corrente elettrica di diversi elettrodomestici. L'obiettivo del PLAID è fornire una biblioteca pubblica di misurazioni, con un'alta risoluzione, di dispositivi elettrici, che può essere impiegata in algoritmi di identificazione di dispositivi esistenti o nuovi. Tutti gli apparecchi all'interno del database, sono stati monitorati singolarmente e aggregati, e le misurazioni includono accensione e spegnimento di questi.

In linea generale, tale set può essere utilizzato in due diversi contesti:

- per etichettare, ovvero per classificare gli apparecchi solamente in base alle misurazioni di tensione e corrente. Utile ai fini del controllo automatico dei carichi e per la pianificazione, altresì utile ai fini economici, per fornire agli utenti finali un piano tariffario più conveniente in base ai propri consumi;
- per disaggregare: tramite processi NILM [6], da una curva di carico aggregata è possibile ricavare le singole curve per i diversi elettrodomestici ottenendo quindi il consumo energetico individuale.

[2] Per un maggiore approfondimento si rinvia al capitolo 2.3.1 a pag 26

Dataset	Rilascio	Approcci		Tipologia				Grandezze				
		CE	SE	A	CI	DI	E	I	V	P	Q	S
REDD	USA 2011		√	√	√	√		√	√	√		
Smart	USA 2012		√	√	√					√		√
BLUED	USA 2011	√	√	√		√	√	√	√	√	√	
HES	UK 2012		√			√						
Tracebase	DE 2021		√			√				√		√
Dataport	USA 2013		√	√	√	√				√		
AMPds	CA 2013		√	√	√	√		√	√	√	√	√
iAWE	IN 2013		√	√	√	√		√	√	√	√	√
IHEPCDS	FR 2013		√	√	√			√	√	√	√	
ACS-Fx	CH 2013		√			√		√	√	√	√	
UK-DALE	UK 2014		√	√	√	√		√	√	√	√	√
ECO	CH 2014		√	√		√		√	√	√		
REFIT	UK 2014		√	√		√				√		
GREEND	A, IT 2014		√			√				√		
PLAID	USA 2014	√	√		√	√	√	√	√			
RBSA	USA 2014		√		√	√						
COMBED	IN 2014		√	√	√			√	√			
DRED	NL 2015		√	√		√				√		
HFED	IN 2015	√				√						
WHITED	DE, A, ID 2016	√				√		√	√			
COOLL	FR 2016	√				√		√	√			
SustDataED	PT 2016	√	√	√		√	√	√	√	√	√	
EEUD	CA 2017		√	√	√	√				√		
ESHL	DE 2017		√	√				√	√	√		
RAE	CA 2018		√	√	√			√	√	√	√	√
BLOND	DE 2018		√	√		√		√	√			

Tabella 3.1: Panoramica dei set di dati di monitoraggio e disaggregazione dell'energia disponibili al pubblico

DataSet	Frequenza di campionamento				Regime di funzionamento	# di abitazioni
	Singoli		Aggregati			
	<1 Hz	≥ 1 Hz	<1 Hz	≥ 1 Hz		
PLAID		√		√	multipli	65
WHITED		√			On, Off	
COOLL					On, Off	
ACS-F2	√				On, Off	
Tracebase	√				On, Off	
REDD	√			√	On, Off	2
DRED	√		√		On, Off	
UK-DALE	√			√	On, Off	6
Dataport	√		√		On, Off	1200
REFIT	√		√		On, Off	20
AMPds2	√		√		On, Off	
HELD1				√	On, Off	

Tabella 3.2: panoramica: PLAID e set simili in termini di frequenza di campionamento, diverse modalità di funzionamento e il numero di edifici considerati

3.2.1 Il dataset

Il dataset è stato aggiornato negli anni a partire da una raccolta primordiale di dati [11] svolta nel 2014, ma la sua ideazione si ebbe già nel 2009. Nell'ultimo aggiornamento il set contiene 1876 curve di carico, misurate su 17 diversi tipi di elettrodomestici con 330 marche e modelli diversi, contiene anche 1314 curve di carico aggregate. Le misurazioni sono state svolte in 65 abitazioni nella città di Pittsburgh in Pennsylvania (USA).

Rispetto alle precedenti pubblicazioni, nella versione considerata [21], alcune delle misurazioni iniziali sono state scartate, in particolare sono state rimosse le misurazioni che non hanno soddisfatto le seguenti richieste:

$$110V \leq Vrms \leq 130V$$

$$maxI \leq 20A$$

In letteratura esistono diversi tipi di banche dati simili al PLAID che si differenziano da esso principalmente per la diversa frequenza di campionamento dei valori misurati e per la quantità di abitazioni considerata nello studio. Il vero punto di forza del PLAID è il fatto che gli apparecchi sono stati campionati per diversi regimi di funzionamento e ad alta frequenza. Nella tabella 3.2 [22] vengono sintetizzate le principali differenze tra i dataset.

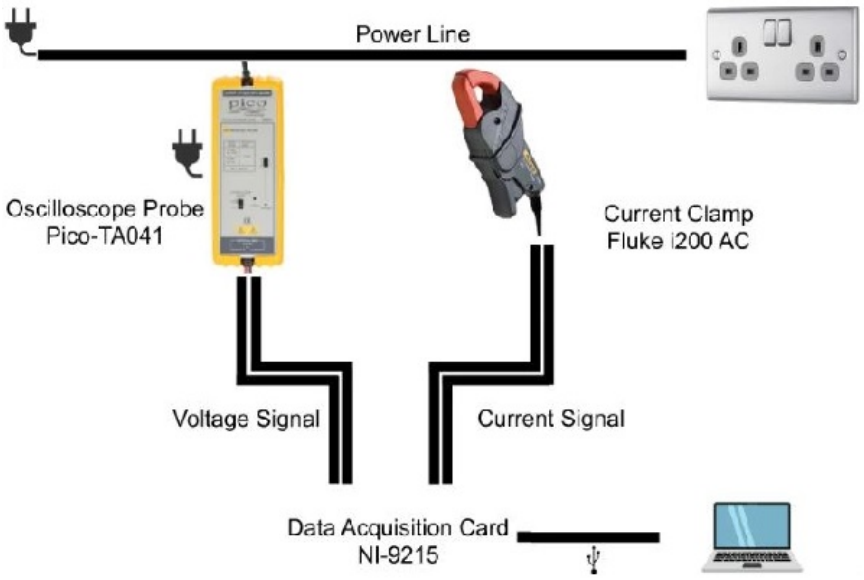


Figura 3.1: Set-up aquisizione dati

3.2.2 Misurazioni

Tutte le misurazioni elettriche sono state effettuate utilizzando sempre lo stesso metodo e le medesime strumentazioni. In particolare sono stati utilizzati:

- una scheda di acquisizione con 4 ingressi analogici accoppiata ad un convertitore analogico-digitale a 16bit;
- un computer per la raccolta dei dati;
- una pinza amperometrica per la misura della corrente;
- un oscilloscopio per la misura della tensione;
- una presa multipla con cui alimentare gli elettrodomestici campione.

La tabella 3.3 fornisce una panoramica dei 17 tipi di elettrodomestici campionati nello studio. Ad esempio, sono state eseguite 204 misurazioni (# mis) su 27 condizionatori diversi (# app) per diverse modalità di funzionamento, inoltre sono state svolte 160 misurazioni su aggregati in cui era presente un condizionatore.

Per rendere più fruibile la raccolta di dati, i carichi sono stati divisi in più famiglie:

- **carichi lineari**: se esiste una relazione lineare tra la sua corrente assorbita e la tensione fornita, questi possono essere resistivi (R), capacitivi (C) o induttivi (I);

Elettrodomestico	Tipologia carico	Singoli		Aggregati	Regime di funzionamento
		# app	# mis	# mis	
Condizionatore	NL	27	204	160	multiplo
Mixer	I	1	2	51	On, Off
Macchina per il caffè	R	1	10	106	On, Off
Lampada fluorescente	NL	45	230	104	On, Off
Ventilatore	I	31	220	102	multiplo
Frigorifero	I	28	108	167	On, Off
Asciugacapelli	R	36	246	0	multiplo
Piastra per capelli	NL	1	10	98	On, Off
Stufa elettrica	R	15	85	0	multiplo
Lampada ad incand.	R	33	157	11	On, Off
Laptop	NL	46	216	90	On, Off
Forno a microonde	NL	32	200	0	multiplo
Saldatore a stagno	NL	1	20	218	On, Off
Aspira polvere	I	15	83	98	On, Off
Lavatrice	NL	16	75	0	On, Off
Bollitore elettrico	R	1	10	109	On, Off
TOTALE			1876	1314	

Tabella 3.3: Riepilogo dei diversi dispositivi

- **carichi non lineari** (NL).

Da notare che, in generale, non sono disponibili carichi puramente capacitivi, quindi i gruppi effettivi sono 3: R, I e NL.

Misurazioni sui singoli elettrodomestici

Ogni volta che un carico viene alimentato dalla rete, si verificherà una transizione di stato tramite un transitorio. La misura cattura dunque l'avviamento transitorio contenente le informazioni dei componenti elettrici presenti e l'eventuale inerzia presente, insieme ad alcuni istanti di regime successivo, da 1 a 20 secondi. Lo switch off degli apparecchi non è stato misurato. Sono inoltre memorizzati, quando disponibili, i seguenti dati:

- Dati dell'elettrodomestico: marca, anno di produzione, numero di modello, tipo di apparecchio, tipo di carico, valori nominali di corrente e tensione,consumo energetico;
- Informazioni relative al processo di acquisizione: il tempo di raccolta dei dati espresso in mese e anno, la frequenza di campionamento, la durata totale della misurazione e la specifica modalità operativa misurata;
- la posizione, salvata all'interno di una stringa.

Le misurazioni di ogni apparecchio sono state memorizzate in file separati, ognuno composto da due colonne rappresentanti rispettivamente corrente e tensione campionata, i dati numerici sono espressi con tre cifre decimali. Non è stato necessario conservare il tempo di misura in quanto la frequenza di campionamento è stata mantenuta costante a 30 kHz, dunque è possibile ottenerlo indirettamente.

Misurazioni sugli aggregati

L'accensione dei singoli apparecchi è stata eseguita in modo consecutivo, diversamente dal caso precedente, è stato registrato anche lo spegnimento. In ogni misura è presente almeno un istante in cui i diversi elettrodomestici aggregati sono a regime. Con riferimento alla divisione fatta nel capitolo 3.2.2 a pag 36, sono state ottenute le seguenti combinazioni:

- due diversi apparecchi appartenenti allo stesso gruppo, combinati in tutti i modi possibili;
- due diversi apparecchi appartenenti a due gruppi diversi, combinati in tutti i modi possibili;
- Tre diversi apparecchi appartenenti a tre gruppi diversi, combinati in tutti i modi possibili.

3.3 Worldwide Household and Industry Transient Energy Data Set

Il **Worldwide Household and Industry Transient Energy Data Set** (WHITED) [17] è un set di dati gratuito contenente il campionamento ad alta frequenza dei transitori di avviamento di tensione e corrente elettrica di diversi elettrodomestici di uso comune. Tutti gli apparecchi all'interno del database sono stati monitorati singolarmente e sono state svolte misurazioni in diverse regioni del mondo (quattro regioni in Germania, una in Austria e due in Indonesia) generalizzandolo in termini di tensione. Il punto di forza del WHITED è la semplicità del set-up e il basso costo degli strumenti utilizzati, permettendo a chiunque di ampliarlo.

3.3.1 Il dataset

Il set comprende 1100 registrazioni, eseguite su 110 diversi elettrodomestici, 10 per ognuno, raggruppati in 47 classi o tipologie, svolte in 6 diverse regioni nel mondo. Al fine di conservare il maggior numero di informazioni, per ogni elettrodomestico è stata scattata una foto alle specifiche elettriche sull'etichetta. Nella tabella 3.4 sono mostrate le diverse classi studiate e il relativo numero di apparecchi misurati (#).

Classe	#	Classe	#	Classe	#
AC	2	Compressore ad aria	1	Bench Grinder	1
Lampada Fluorescente	2	Caricatori Smartphone	7	Macchina per caffè	1
Friggitrice	1	Pc Desktop	1	Dissaldatore	1
Foratrice	2	Ventola	6	Ventola ad aria calda	1
Flat Iron	2	Game Console	4	Amplificatore per chitarra	1
Asciugacapelli	6	Lampada alogena	1	Stufa	1
Rack HiFi	1	Ferro da stiro	3	Seghetto alternativo	1
Spremitore	1	Bollitore	6	Laptop	1
Stampante laser	1	Lampada a led	9	Lampada a bulbo	6
Strumento per massaggi	3	Forno a Microonde	2	Miscelatore	4
Monitor	2	Repellente Zanzare	1	Multitool	1
Alimentatore	4	Proiettore	1	Macchina da cucire	1
Scalda scarpe	2	Shredder	2	Saldatore a stagno	2
Tostapane	4	Tapis roulant	1	TV	1
Aspirapolvere	4	Lavatrice	1	Scaldabagno	4
Pompa ad acqua	1				

Tabella 3.4: Riepilogo dei diversi dispositivi

3.3.2 Misurazioni

Il sistema di misurazione utilizzato nel WHITED, mostrato in figura 3.2, è composto da:

- una scheda audio USB esterna con un chipset Cmedia CM6206;
- una multipresa a 3 porte modificata;
- una pinza amperometrica YHDC con resistenza di carico incorporata. Questa produce un segnale di 1 V con una corrente primaria di 30 A;
- un trasformatore AC-AC. Per le misure di tensione è necessario trasformare da 230 V a 11 V;
- un partitore di tensione. Per avere un segnale di tensione corrispondente che si trova nell'intervallo di ingresso della scheda audio, è stato ridotto con un partitore di tensione a 0.47 V.

L'idea di base è quella di sfruttare una scheda audio come convertitore A/D in quanto queste hanno un ottimo rapporto prezzo/prestazioni se utilizzate da convertitori. Con questo sistema è stato possibile campionare i segnali con una risoluzione temporale di 44.1 kHz e un'ampiezza di 16 bit.

L'implementazione è stata fatta con una routine Matlab, utilizzando il pacchetto DSP interno per monitorare il segnale di ingresso della scheda audio. L'avviamento degli elettrodomestici è manuale, mentre l'avvio della registrazione è automatizzato: se l'energia del segnale di corrente supera

Figura 3.2: Sistema di Misurazione WHITED

una soglia definita, la routine avvia la registrazione della durata di 5 secondi, aggiungendo una finestra di preavvio di 100 ms. Questa scelta è stata presa per facilitare l'utilizzo del WHITED in algoritmi basati sulle differenze. I files sono salvati come flac, formato audio senza perdita. I nomi dei file contengono le informazioni degli elettrodomestici, seguendo la logica: $[Class]\_[Name]\_[Region]\_[\#Kit]\_[TimeStamp].flac$

3.3.3 Qualità dei dati

Per valutare la bontà del sistema di misurazione sono stati confrontati un segnale di 10 secondi vuoto con un segnale sinusoidale di ampiezza massima. Da questo test è stato possibile stimare un SNR medio (rapporto segnale/-rumore) di 4,8 mA dove 30 A corrisponde al massimo RMS, quindi l'SNR effettivo è:

$$SNR = 20 \cdot log_{10} \frac{RMS_{max}}{RMS_{noise}} = 20 \cdot log_{10} \frac{30A}{0.0048A} 75.91dB$$

La corrente massima misurabile picco a picco è:

$$I_{pp} = 30A_{RMS} \cdot 2\sqrt{2} = 84.4A$$

dunque è possibile valutare la risoluzione effettiva di corrente come:

$$I_{step} = \frac{I_{pp}}{I_{maxRMS}} \cdot I_{noiseRMS} = \frac{84.4A}{30A} \cdot 0.0048A = 0.0135A$$

La risoluzione e il rumore della scheda audio consentono un gradino di corrente di 0.0135 A che si traduce in un gradino di potenza misurabile di circa

3.1 W basato su 230 V:

$$I_{step} = 230 \cdot 0.0135 = 3.1$$

W

Parte II

Classificazione degli elettrodomestici tramite firma armonica e foresta decisionale

Capitolo 4

Concetti teorici

All'interno del presente capitolo saranno illustrati i principi su cui si è basata l'analisi oggetto della tesi: la Classificazione degli elettrodomestici tramite Firma armonica e Foresta decisionale. Presentato il concetto di armonica e firma armonica, al paragrafo 4.1, e la trasformata di Fourier, al paragrafo 4.2, saranno approfonditi gli alberi decisionali già introdotti nel capitolo 2.4.3. Seguirà una attenta analisi delle tecniche utilizzate per la validazione dell'esperimento, ovvero la matrice di confusione (capitolo 4.4) e la cross validation (capitolo 4.5).

4.1 Le armoniche

In un sistema elettrico a corrente alternata, la tensione di alimentazione viene generata e fornita sotto forma di un'onda sinusoidale. Quest'ultima in alcuni Paesi, tra cui l'Italia, ha una frequenza di 50 Hz, mentre in altri, come USA, ha una frequenza di 60 Hz. Per quanto concerne invece il carico elettrico, si rileva come la sua forma d'onda periodica nel tempo sia costituita dalla combinazione di più onde a diverse frequenze, chiamate armoniche. Queste sono generate dall'elettronica di potenza presente all'interno degli elettrodomestici.

Le **Armoniche**, quindi, sono componenti sinusoidali delle correnti o delle tensioni, che hanno una frequenza pari ad un multiplo intero della frequenza del sistema di distribuzione, denominata frequenza fondamentale. Esse, sovrapponendosi rispettivamente alla corrente, o tensione, fondamentale provocano la distorsione della forma d'onda, come mostrato in figura 4.1 con la somma tra un'onda fondamentale ed una 3° e 5° armonica.

L'ampiezza della forma d'onda alla frequenza fondamentale rimane comunque predominante.

Il contenuto armonico è una considerazione fondamentale per analizzare l'elaborazione del segnale. Poiché qualsiasi segnale periodico può essere

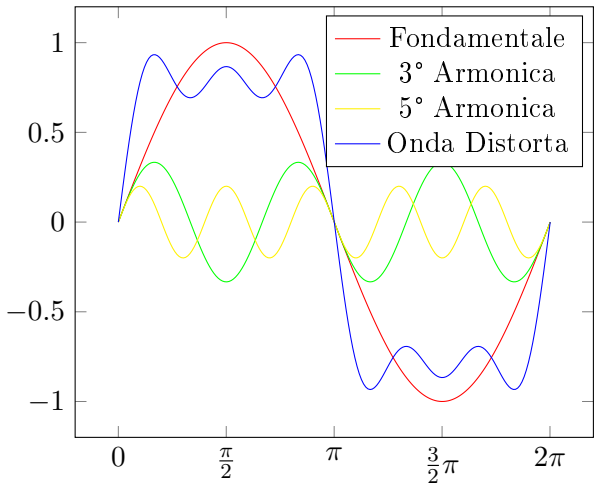


Figura 4.1: Armoniche e forma d'onda distorta

espresso tramite una combinazione lineare di funzioni sinusoidali pure, è possibile utilizzare l'analisi armonica chiamata **serie di Fourier**.

> ***Teorema di Fourier*** *Qualunque funzione periodica di periodo T_0 o di frequenza $f_0 = 1/T_0$, continua e limitata può essere rappresentata mediante una somma di funzioni sinusoidali pure di opportuna ampiezza e di frequenza multipla della frequenza fondamentale f_0.*

Matematicamente, una forma d'onda sinusoidale pura di periodo T può essere espressa come:

$$y(t) = A \cdot sin(\omega k + \vartheta)$$

dove:

- A è l'ampiezza del segnale;
- $\omega = \frac{2 \cdot \pi}{T}$ è la frequenza angolare;
- ϑ è l'angolo di fase iniziale.

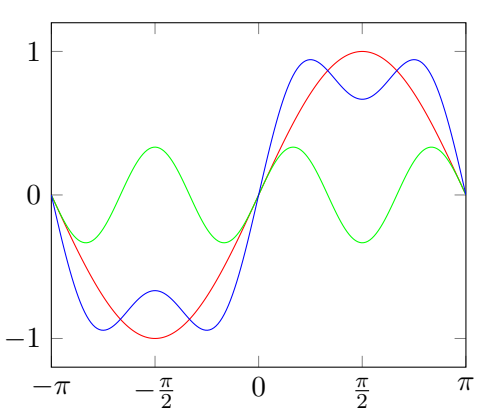
In un sistema reale, una forma d'onda di potenza contiene armoniche rumore, esprimibile secondo la formula:

$$y(t) = a_0 + a_1 \cdot sin(\omega k + \vartheta_1) + \sum_{n=2}^{N} \big(a_n \cdot sin(\omega k + \vartheta_n) + \eta(k)\big)$$

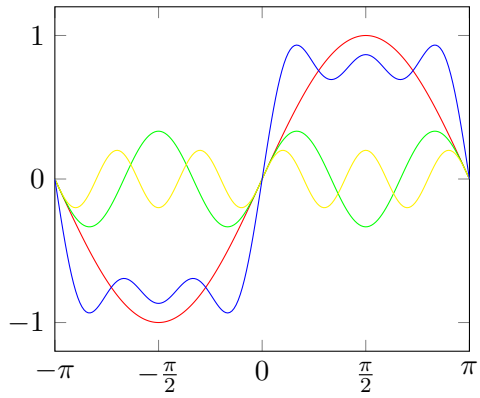
in cui i termini con pedice 0 indicano la componente continua, con pedice 1 la prima armonica e con pedice n l'armonica $n - esima$.

Tramite Fourier è possibile riscrivere la formula precedente come:

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cdot cos(n\omega k) + \sum_{n=1}^{\infty} b_n \cdot sin(n\omega k)$$



(a) Armonica fondamentale in Rosso, 3° armonica in Verde, Somma in Blu



(b) Armonica fondamentale in Rosso, 3° armonica in Verde, 5° armonica in Giallo, Somma in Blu

Figura 4.2: Armoniche dispari di ordine inferiore

La parte pari del segnale è la somma dei termini $a_n \cdot cos(n\omega k)$, mentre la parte dispari è la somma dei termini $b_n \cdot sin(n\omega k)$. Riorganizzando si ottiene:

$$f(t) = c_0 + \sum_{n=1}^{\infty} c_n \cdot cos(n\omega k - \vartheta_n)$$

ovvero una riscrittura in termini di armoniche del segnale originale, dove:

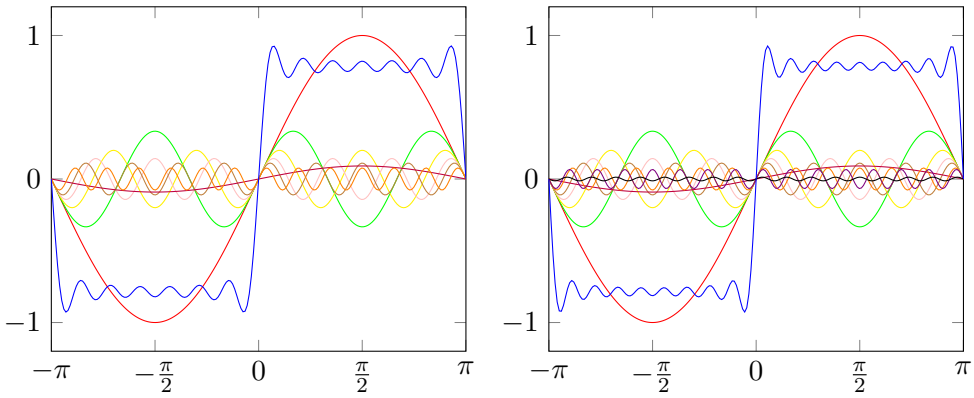
- la componente continua è $c_0 = a_0$;
- l'ampiezza dell'armonica n-esima vale $c_n = \sqrt{a_n^2 + b_n^2}$;
- gli angoli di fase $\vartheta_n = tan^{-1}\big(b_n/a_n\big)$.

Le frequenze armoniche pari non influenzano in modo rilevante la distorsione forma d'onda. Il semiperiodo positivo e quello negativo dell'onda iniziale hanno lo stesso andamento nel tempo, a parte il segno, pertanto le armoniche pari sono trascurabili.

Per quanto riguarda le armoniche dispari, le figure 4.2 e 4.3 mostrano rispettivamente l'influenza delle armoniche di ordine inferiore e di quelle di ordine superiore sulla forma d'onda. La forma d'onda distorta viene modificata grandemente dalle prime, mentre il contributo delle seconde è esiguo dunque trascurarle non implicherebbe una perdita di informazione. Un altro vantaggio nel trascurarle sarà la rimozione del rumore.

4.2 Trasformata veloce di Fourier

La **trasformata discreta di Fourier** [13], in inglese Discrete Fourier Transform (DFT), è un particolare tipo di trasformata di Fourier che converte



(a) Prime 13 armoniche dispari (b) Prime 17 armoniche dispari

Figura 4.3: Armoniche dispari di ordine superiore

una collezione finita di campioni equispaziati di una funzione in una collezione di coefficienti di una combinazione lineare di sinusoidi complesse, ordinate al crescere della frequenza. Data una $N - pla$ di numeri complessi $x_0, x_1, ..., x_{N-1}$, la DFT è definita dalla formula:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{2\pi i}{N} nk}$$

con $k = 0, 1, ..., N-1$.

È un modo, dunque, per rappresentare una funzione nel tempo nel dominio delle frequenze. Le frequenze delle sinusoidi della combinazione lineare (periodica) prodotta dalla trasformata sono multipli interi di una frequenza fondamentale, il cui periodo è la lunghezza dell'intero intervallo di campionamento, la durata del segnale.

La **trasformata di Fourier veloce**, in inglese Fast Fourier Transform (FFT), è un algoritmo le cui origini porterebbero ai primi dell'Ottocento, ottimizzato per calcolare la trasformata discreta di Fourier.

4.2.1 Algoritmo Cooley-Tukey

L'**algoritmo di fattorizzazione di Cooley-Tukey** [4] è l'algoritmo FFT più diffuso. Basato sul principio di *divide et impera*, spezza ricorsivamente una DFT di qualsiasi dimensione N in due DFT più piccole di dimensioni N_1 e N_2, tale che $N = N_1 N_2$. È buona pratica dividere e trasformare in due pezzi di dimensione $N/2$ ad ogni passo, poichè questo algoritmo è ottimizzato per dimensioni che siano potenze di due.

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} \cdot e^{-\frac{2\pi i}{N/2} mk} + e^{-\frac{2\pi i}{N} k} \cdot \sum_{m=0}^{N/2-1} x_{2m+1} \cdot e^{-\frac{2\pi i}{N/2} mk} =$$

$$= X_k^{pari} + e^{-\frac{2\pi i}{N}k} \cdot X_k^{dispari}$$

in cui x_n viene spezzettato in due termini rispettivamente con $n = 2m$ sugli indici pari e $n = 2m + 1$ sugli indici dispari, ottenendo alla fine due DFT X_k^{pari} e $X_k^{dispari}$.

Il Cooley-Tukey ha una complessità temporale [1] di $O(NlogN)$, molto più veloce rispetto ad una classica trasformata discreta di Fourier la cui complessità temporale è di $O(N^2)$.

In sintesi, l'energia del segnale viene individuata su una gamma di frequenze attraverso la rappresentazione nel dominio della frequenza, includendo anche le informazioni sullo spostamento di fase. Sulla base delle informazioni di fase, è possibile recuperare il segnale originale combinando tutte le singole componenti di frequenza. Quindi, l'ampiezza e la fase a ciascuna frequenza verranno rappresentate dal dominio della frequenza, ottenendo come output della FFT un numero complesso.

4.3 Alberi decisionali

Gli **Alberi di decisione** costituiscono un modo semplice per classificare oggetti in un numero finito di classi e appartengono alla categoria di algoritmi di Data Mining.

> *Il **Data mining** è l'insieme di tecniche e metodologie che hanno per oggetto l'estrazione di informazioni utili da grandi quantità di dati (es. banche dati, datawarehouse, ecc.), attraverso metodi automatici* [2] *o semi-automatici (es. apprendimento automatico) e l'utilizzo scientifico, aziendale, industriale o operativo delle stesse.*

In natura, un albero si compone da una serie di rami, ognuno dei quali derivante da un nodo, punto in cui avviene la ramificazione dell'albero. Le estremità finali degli alberi sono composte dalle foglie. Analogamente, un albero decisionale suddivide il set di dati in regioni omogenee e non sovrapposte. La categorizzazione è svolta tramite le foglie ovvero nodi finali che non generano ulteriori ramificazioni. Le componenti strutturali di un albero decisionale sono:

- I **nodi** costituiscono una macro-classe, suddivisa successivamente in classi sempre più piccole nei nodi figli. Ogni nodo viene etichettato con il nome dell'attributo di riferimento. In sintesi, su ogni nodo interno verrà effettuato un test su un attributo;

[1] Definizioni e approfondimento al sito https://it.wikipedia.org/wiki/Complessit%C3%A0_temporale

[2] Wikipedia, https://it.wikipedia.org/wiki/Data_mining#:~:text=Il%20data%20mining%20(letteralmente%20dall,o%20semi%2Dautomatici%20(es.

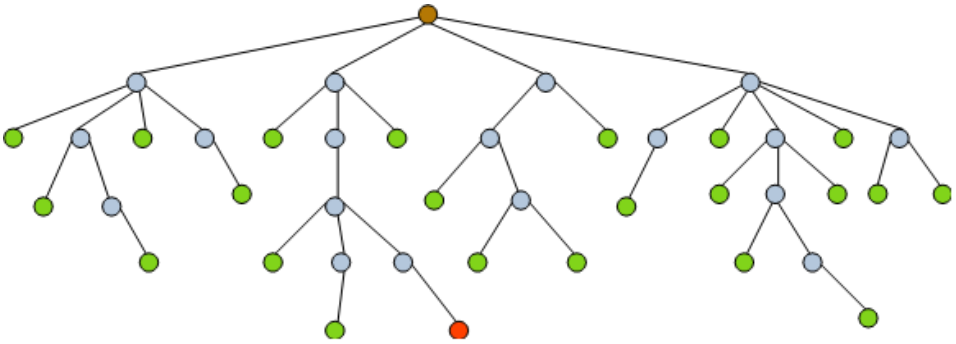


Figura 4.4: Esempio grafico di Albero decisionale dove in marrone: Nodo Radice, in azzurro: Nodo generico, in verde: Nodo Foglia, in rosso: Predizione

- I **rami**, o archi, costituiscono l'insieme delle proprietà, o regole di splitting, che determinano un percorso all'interno dell'albero e, in ultimo, le classificazioni. Tali regole sono definite in relazione a specifici valori assunti dall'attributo con cui è identificato il nodo padre. Ogni ramo corrisponde ad uno dei possibili valori che l'attributo può assumere;
- Le **foglie** sono una particolare categoria di nodi. Rientrano in questa categoria, infatti, tutti i nodi finali dell'albero, che quindi non generano a loro volta delle sotto-classi. Le foglie rappresentano il risultato della classificazione dell'albero. Le sotto-classi costituite dalle foglie rappresentano il migliore livello di informazione generabile dall'albero.

La costruzione di un albero può essere sintetizzata in due fasi: la fase di **Building** in cui avviene la vera e propria formazione dell'albero, e la fase di **Pruning**, in italiano "potatura", in cui si punta alla riduzione delle dimensioni dell'albero affinchè questo sia il più compatto possibile senza però limitare le sue capacità predittive. Quest'ultima fase nasce dall'esigenza di evitare problemi di *overfitting* che si verificano quando il dataset è composto da attributi irrilevanti che rendono difficile l'induzione dell'albero decisionale. Un'altra definizione di overfitting è: rischio di sovradattamento durante il processo di apprendimento induttivo ovvero quando l'algoritmo si adatta (fitting) troppo bene (over) ai dati di training perdendo la generalità.

4.3.1 Metodi di split e induzione

I test sugli attributi, effettuati nei nodi interni, differiscono a seconda della natura dei dati di input e dalla scelta delle regole di split.

L'algoritmo di induzione di alberi decisionali costruisce un albero selezionando ad ogni nodo l'attributo che permette di ottenere lo split migliore ed

inoltre specifica quale test deve essere effettuato su di esso. Il primo algoritmo di induzione si deve a John Ross Quinlan che nel 1986 teorizzò l'*Iterative Dichotomiser 3*, più comunemente noto come ID3[26], che successivamente fu ampliato e ottimizzato su specifici problemi dando vita ai più noti ed efficaci induttori: C4.5[27], C5.0, CART[3], CHAID, MARS. Dato un qualsiasi insieme di record R, uno split è una partizione indotta da un test su un attributo A, ovvero:

$$R = R_1 \cup R_2 \cup ... \cup R_n$$

La bontà dello split è legata alla differenza di purezza tra R e $R_1, R_2, ..., R_n$, dove la purezza indica quantitativamente se una classe è predominante sulle altre. Lo split deve far in modo che gli insiemi $R_1, R_2, ..., R_n$ siano globalmente più puri di R.

Per capire quanto sia efficace uno split è necessario mettere a confronto il grado di impurità del nodo padre e il grado di impurità dei nodi figli dopo lo split.

C4.5

> ***C4.5*** *a landmark decision tree program that is probably the machine learning workhorse most widely used in practice to date* [3]

Il C4.5 [27] costruisce alberi decisionali da un insieme di dati di addestramento allo stesso modo dell'ID3, ovvero utilizzando il concetto di entropia dell'informazione. Il grado di impurità è dunque misurato tramite l'**Entropia di Shannon**:
Sia C il numero di classi e $p(A, j)$ è la probabilità che l'attributo A sia assegnato alla j-esima classe. L'entropia dell'attributo A è calcolata come:

$$H(A) = -\sum_{j=1}^{C} p(A, j) \cdot log_2(P(A, j))$$

Maggiore è la differenza di impurità tra i nodi, migliore sarà la condizione di test usata per lo split. Tale differenza è chiamata **Guadagno dell'informazione**, o in inglese Information Gain:

$$IG(A) = H(A) - \sum_{v \in A} \frac{|A_v|}{|A|} \cdot H(A_v)$$

dove A_v è il sotto insieme dei valori di A in cui l'attributo è uguale a v.

Purtroppo le misure di impurità appena introdotte tendono a favorire split che partizionano l'insieme di record iniziale in molte partizioni. Questo porta il modello ad adattarsi troppo bene al training set per poi non essere

[3] Weka machine learning, https://www.cs.waikato.ac.nz/~ml/weka/book.html

più in grado di rappresentare un altro data set sullo stesso insieme di attributi. Una particolare strategia per ovviare questa problematica, utilizzata appunto all'interno dell'induttore C4.5 è quella di sostituire il Guadagno dell'informazione con il **Rapporto del guadagno dell'informazione**, o in inglese Information gain ratio, definito come:

$$IGR(A) = \frac{IG(A)}{SplitInfo(A)}$$

in cui:

$$SplitInfo(A) = -\sum_{v \in A} \frac{|A_v|}{|A|} \cdot log_2(\frac{|A_v|}{|A|})$$

Questo algoritmo ha alcuni casi base:

- Tutti i dati campionati appartengono alla stessa classe. Quando ciò accade, crea semplicemente un nodo foglia per l'albero decisionale che dice di scegliere quella classe.
- Nessuna delle caratteristiche fornisce un guadagno dell'informazione. In questo caso, C4.5 crea un nodo decisionale in un livello più alto utilizzando il valore atteso della classe.
- Incontrata una classe mai vista in precedenza, di nuovo, C4.5 crea un nodo decisionale in un livello più alto utilizzando il valore atteso della classe.

4.3.2 Random forest

La **Random forest** è un modello di classificazione che sfrutta numerosi alberi decisionali per ottenere un risultato complessivamente migliore rispetto a quello che si otterrebbe con ognuno degli alberi preso singolarmente. In generale la metodologia di utilizzare un insieme di modelli che concorrono ad ottenere un risultato migliore è detta *apprendimento ensemble*, di cui le foreste casuali sono uno dei casi più famosi.

Costruiti gli alberi che formano la foresta, l'output del modello verrà ottenuto confrontando le singole previsioni di ogni singolo albero. Fra tutte le metodologie di confronto, la più pratica è la *Majority vote*, utilizzata nelle tecniche chiamate **Bagging decision forest** (BDT):

$$P_{BDT} = argmax(P_1, P_2, ..., P_n)$$

dove P_{BDT} rappresenta la previsione generale della foresta e $P_1, P_2, ..., P_n$ le previsioni dei singoli alberi.

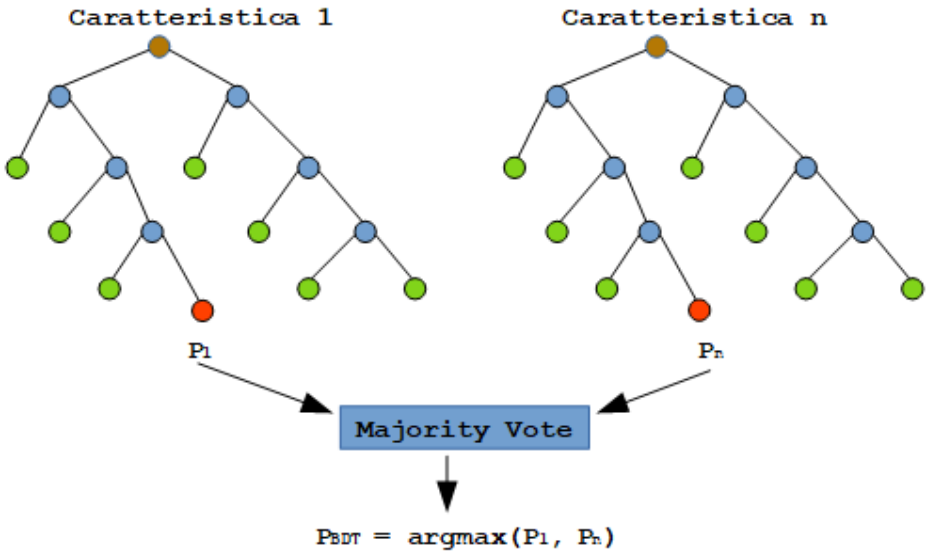


Figura 4.5: Esempio grafico di Majority vote

4.4 Matrice di Confusione

Nel campo del machine learning e nella classificazione statistica, una **matrice di confusione**, o *confusion matrix* in inglese, è una tabella in cui le previsioni sono rappresentate nelle colonne e lo stato effettivo è rappresentato nelle righe. Da questa tabella, quindi, è possibile comprendere le prestazioni di un modello predittivo di classificazione in modo da determinare quanto questo modello sia accurato ed efficace. Esistono due tipologie di matrice di confusione:

- Binaria: tenta di prevedere le prestazioni di un modello predittivo a due classi del tipo Vero/Falso;
- Multiclasse: tenta di prevedere le prestazioni di un modello predittivo formato da più classi.

Prendendo in esempio la matrice binaria in figura 4.6, possono verificarsi quattro casi:

- **Vero positivo**, in inglese *True positive* (TP): Se la classe prevista è SI ed è uguale alla classe effettiva. Il modello ha classificato correttamente.
- **Vero negativo**, in inglese *True negative* (TN): Se la classe prevista è NO ed è uguale alla classe effettiva. Il modello ha classificato correttamente.

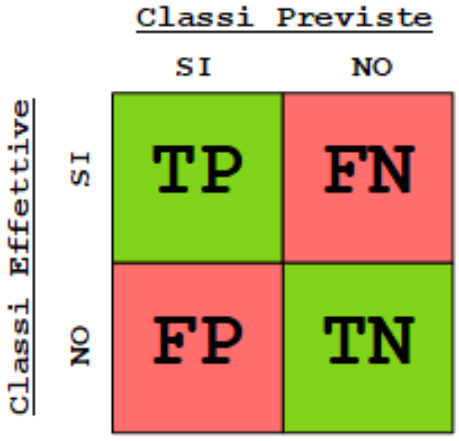


Figura 4.6: Matrice di confusione

- **Falso positivo**, in inglese *False positive* (FP): Se la classe prevista è SI ma è diversa dalla classe effettiva. Il modello ha sbagliato la classificazione.
- **Falso negativo**, in inglese *False negative* (FN): Se la classe prevista è NO ma è diversa dalla classe effettiva. Il modello ha sbagliato la classificazione.

Dalla matrice di confusione è possibile ottenere diverse metriche di valutazione:

- **Precisione**, in inglese *precision*: abilità di non etichettare un'istanza positiva che è in realtà negativa.

$$Precision = \frac{TP}{TP + FP}$$

- **Richiamo**, in inglese *recall*: capacità di un classificatore di trovare tutte le istanze positive. Varia da 0 (peggiore) a 1 (migliore).

$$Recall = \frac{TP}{TP + FN}$$

- **Specificità**, in inglese *specificity*: numero di previsioni negative corrette diviso il numero totale di negativi. Varia da 0 (peggiore) a 1 (migliore).

$$SP = \frac{TN}{TN + FP}$$

- **Tasso dei falsi positivi**, in inglese *false positive rate*: percentuale delle previsioni positive errate sul totale delle istanze negative. Varia

da 0 (migliore) a 1 (peggiore).

$$FPR = \frac{FP}{TN + FP}$$

- **Accuratezza**, in inglese *accuracy*: percentuale delle previsioni esatte sul totale delle istanze, varia da 0 (peggiore) a 1 (migliore).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Tasso di errore**, in inglese *error rate*:

$$EER = 1 - ACC$$

- **F1 Score**: media armonica ponderata delle metriche Precisione e Richiamo in modo tale che il punteggio migliore sia 1 e il peggiore 0.

$$F1 = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$$

4.5 La Cross validation

La **validazione incrociata**, in inglese *Cross validation*, detta anche *k-fold validation*, è una tecnica statistica usata nel machine learning per eliminare il problema dell'overfitting nei training-set.

Consiste nella suddivisione dell'insieme di dati totale in k partizioni di uguale numerosità e, ad ogni passo, la k-esima parte dell'insieme di dati viene a essere quella di convalida, mentre la restante parte costituisce sempre l'insieme di addestramento. Si allena il modello per ognuna delle k partizioni, evitando i problemi di sovradattamento, ma anche di campionamento asimmetrico, tipico della suddivisione dei dati in due sole parti ovvero dataset di test e di training.

In sintesi:

1. si suddivide il training set in k parti di uguale dimensione. Generalmente $k = 5$ oppure 10;
2. per ogni partizione, si seleziona una parte $1/k$ per utilizzarla come *validation set*;
3. le restanti $(k-1)/k$ parti invece continuano a comporre il training dataset.

Capitolo 5

Il metodo proposto

L'algoritmo proposto è stato sviluppato in ambiente Matlab, costituito da due macro in cui si richiamano le varie funzioni elaborate ai fini del risultato finale. Le due macro svolgono rispettivamente l'elaborazione delle curve e il vero e proprio esperimento. L'obiettivo del metodo è quello di riuscire a catalogare tramite la firma armonica ed alberi decisionali, le curve di corrente di elettrodomestici di uso quotidiano. Lo schema a blocchi seguito per costruire tale algoritmo è mostrato in figura 5.1

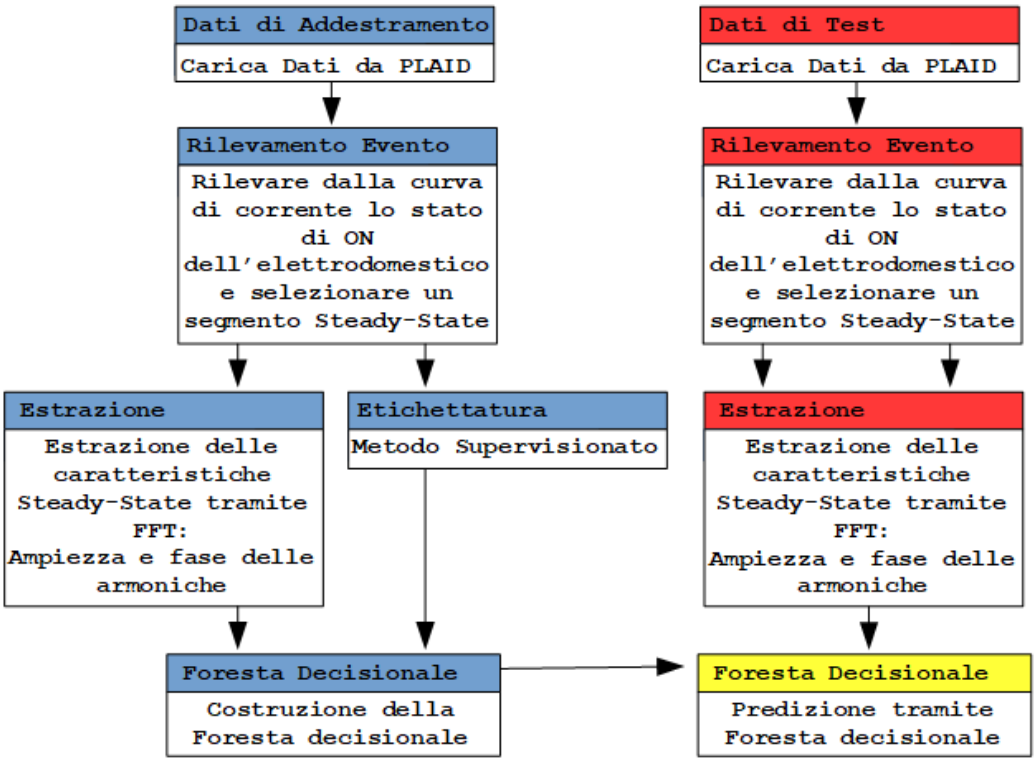


Figura 5.1: Schema a blocchi

L'esperimento è svolto in due fasi:

- Fase di Training, in cui l'obiettivo ultimo è istruire la foresta decisionale;
- Fase di Test, in cui si utilizzerà la foresta già istruita per classificare gli elettrodomestici di test.

In sintesi in ognuna delle due fasi si avrà:

- L'input dell'algoritmo sono segnali di correnti estratti dal dataset PLAID[1]
- Come già discusso nel capitolo 4.1, i circuiti elettronici presenti all'interno dei dispositivi elettrici generano armoniche di frequenza all'interno dei segnali di corrente che sono componenti spettrali uniche. Le frequenze armoniche dispari, principalmente quelle inferiori, hanno maggiore influenza sul cambiamento della forma d'onda della corrente. Sulla base di questa osservazione, si prenderanno in considerazione solamente le seguenti armoniche: la frequenza fondamentale, 3a armonica, 5a armonica e 7a armonica. Considerando che il PLAID è stato campionato in Paesi in cui la distribuzione avviene a 60 Hz, queste armoniche corrispondono rispettivamente a 60 Hz (fondamentale), 180 Hz, 300 Hz e 420 Hz.
- La foresta decisionale proposta, tramite l'induttore C4.5, esamina caratteristiche di stato stazionario [2] quali ampiezza e la fase delle armoniche di corrente, elaborate precedentemente tramite FFT.

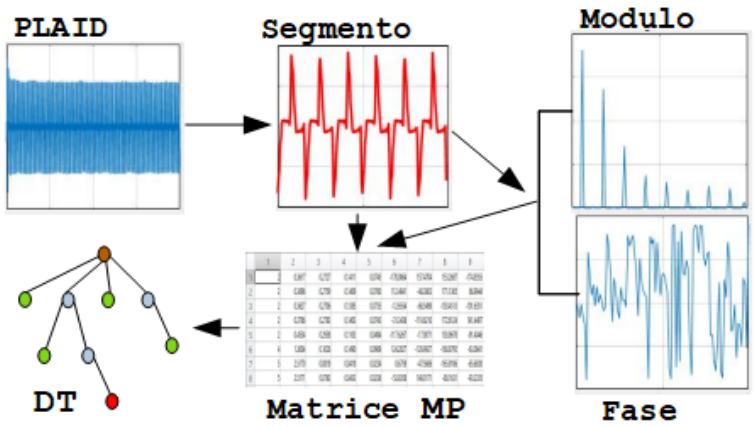


Figura 5.2: Procedimento in forma grafica

[1] Per un maggiore approfondimento si rinvia al capitolo 3.2 a pag 33
[2] Per un maggiore approfondimento si rinvia al capitolo 2.3.2 a pag 26

5.1 Elaborazione delle curve PLAID

Il dataset PLAID fornisce le curve di corrente campionate a 30 kHz mentre gli alberi decisionali prendono in input le caratteristiche MP delle armoniche di corrente, ovvero il modulo e la fase. L'obiettivo dell'elaborazione sarà dunque ottenere queste caratteristiche MP. La macro utilizzata in Matlab richiama due funzioni principali costruite ad hoc per il problema, chiamate $Pre\_Elaboro\_PLAID$ e $FFTtoMP$. È stata scritta anche una terza funzione, chiamata Non_Li_Voglio, utilizzata per escludere dall'analisi le curve che presentano problemi di campionamento o per scartare in toto tutti gli elettrodomestici che non soddisfacevano il numero minimo di curve richiesto, posto uguale a 10, per portare a termine l'esperimento. Ultimata l'elaborazione, i dati ricavati vengono salvati in un file *.txt* pronti per essere processati dalla foresta. Tale modalità di operare consente di svolgere il procedimento in modo più rapido ed efficace: possedere la caratteristica MP in formato file eviterà di dover processare ogni volta le curve PLAID e quindi risparmiare tempo, molto importante quando la quantità dei dati da processare è molto rilevante.

5.1.1 Funzione *Pre_ Elaboro_ PLAID*

La funzione **Pre_Elaboro_PLAID** estrae le curve di corrente dal dataset PLAID, le ricampiona a 2 kHz e ne estrae un segmento Steady-State di 200 campioni.

Durante tutto il processo di pre-elaborazione è possibile visualizzare graficamente, figura 5.3, lo stato dell'algoritmo inserendo il valore 1 nell'input *Grafici*, molto utile ai fini diagnostici.

Il codice è sviluppato tramite un grande ciclo *for* sugli indici di riferimento delle curve del dataset chiamato *Importi*. Il primo passo estrae le curve di corrente scartando quelle di tensione. Il secondo passo esegue il ricampionamento, o *resample*, a 2 kHz. Da diverse prove emerge che il valore di 2 kHz è un ottimo compromesso tra qualità dei dati e capacità computazionale in termini di tempo.

Il terzo passo è il cuore della funzione. In ordine esegue:

1. Calcola il valore RMS della curva totale I_{2kHz}e ad ogni passo riscrive la curva all'interno di un nuovo vettore X. Per ogni periodo calcola il valore RMS del vettore X e se quest'ultimo soddisfa $RMS(X) \geq 0.9 \cdot RMS(I_{2kHz})$ allora si considera l'elettrodomestico in ON.

2. Aspetta 5 periodi affinchè tutti i transitori di avvio siano estinti.

3. Inizia ad estrarre il segmento Steady-State. Per ogni periodo n confronta il picco massimo con il rispettivo nel periodo $n-1$, se i due non differiscono per un massimo del 10% allora il codice continua ad

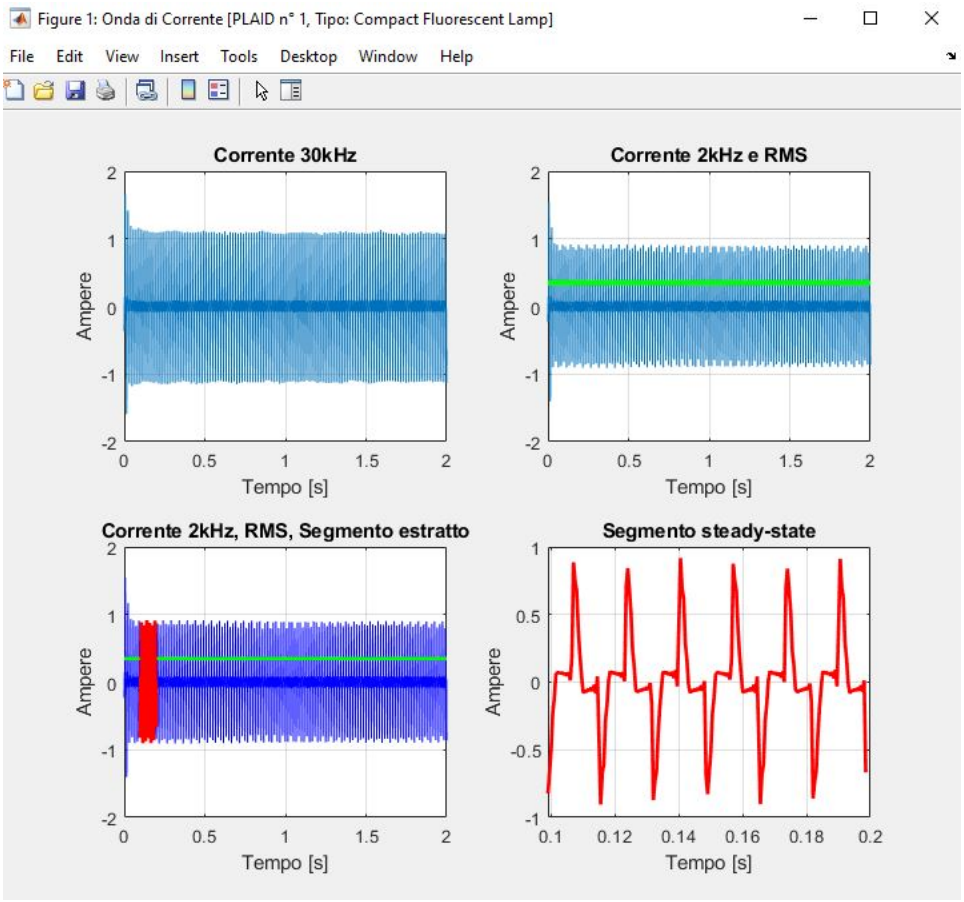


Figura 5.3: Processo funzione Pre_Elaboro_PLAID

estrarre il segmento fino ad ottenere 200 campioni. Se la citata condizione non è stata rispettata allora cancella il segmento e riprocede con il punto 3. Il punto di forza di questa funzione è che è molto 'elastica' rispetto ai dati in input grazie a quest'ultimo punto.

Da notare anche che la funzione conserva i metadati, ovvero nome del dispositivo, wattaggio ed eventuali dati di targa (se riportati in PLAID), molto utile in fase di esperimento per stampare efficacemente i risultati.

5.1.2 Funzione *FFTtoMP*

Per ottenere una caratteristica MP, si dovrà trasformare il segnale corrente dal dominio del tempo al dominio della frequenza usando la trasformata veloce di Fourier. La soluzione proposta è sviluppata nella funzione **FFTtoMP**, che prende come input il segmento steady-state e ne restituisce i valori MP.

Come per la funzione precedente, il codice è sviluppato tramite un grande ciclo *for* in cui si evidenziano i seguenti passaggi:

1. Prende in input il segmento Steady-State e ne calcola la FFT ottenendo spettro e fase a due lati come in figura 5.4a. Da notare che i dati del segmento steady-state sono presentati nel dominio del tempo con frequenza di campionamento di $f_s = 2$ kHz, quindi l'intervallo di campionamento Δt è dato dall'equazione:

$$\Delta t = \frac{1}{f_s} = \frac{1}{2000} = 0.0005s$$

Come già accennato precedentemente, il segmento steady-state è composto da un numero di campioni pari a $N = 200$ pertanto la durata in secondi vale:

$$T = N \cdot \Delta t = 200 \cdot 0.0005 = 0.1s$$

inoltre la frequenza massima in Hz, indicata con F_{max} corrisponde a:

$$F_{max} = \frac{f_s}{2} = \frac{2000}{2} = 1000Hz$$

Le linee spettrali sono il numero di *bin* indicato con SL:

$$SL = \frac{N}{2} = \frac{200}{2} = 100$$

allora la risoluzione in frequenza, ovvero la distanza tra ciascuna linea spettrale, vale:

$$\Delta t = \frac{F_{max}}{SL} = \frac{200}{2} = 100Hz$$

2. Viene scartata la metà dei valori di ampiezza e fase ottenendo i reali valori di MP, mostrati in figura 5.4b.

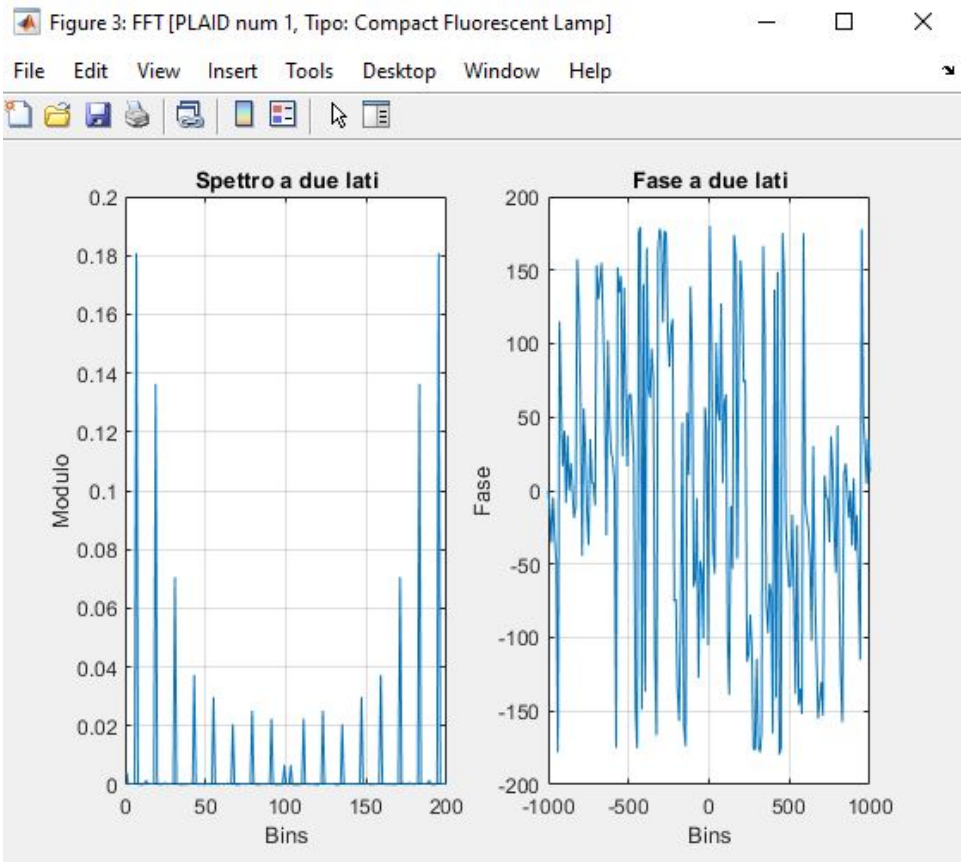
3. Costruisce una tabella MP selezionando i valori alla fondamentale e alle armoniche di ordine 3, 5 e 7, rispettivamente con frequenze 60 Hz, 180 Hz, 300 Hz e 420 Hz.

Un esempio di matrice MP è riportato in figura 5.5, in cui si evidenziano sulla prima colonna le etichette, o classi, degli elettrodomestici; nelle colonne 2, 3, 4 e 5 le ampiezze delle armoniche rispettivamente dalla 1° alla 7°; nelle colonne 6, 7, 8 e 9 i valori di fase.

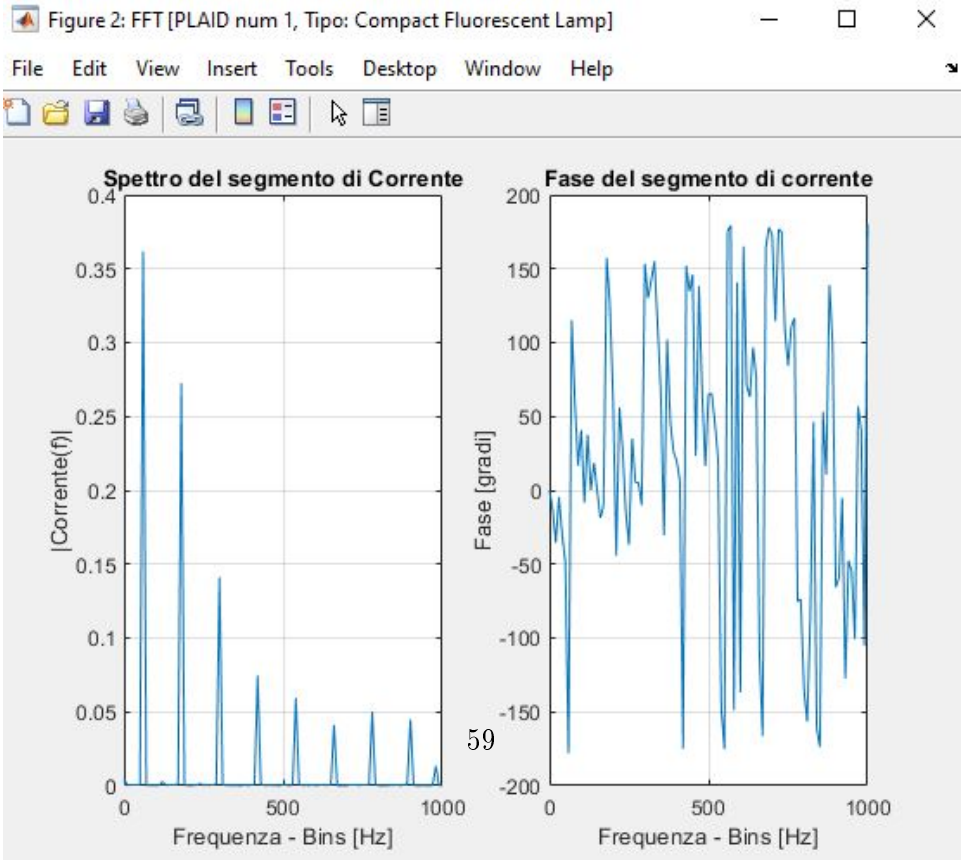
Anche in questa funzione è stata implementata la possibilità di seguire il processo graficamente tramite l'input *Grafici*.

5.2 Costruzione della Foresta decisionale e classificazione

Il vero e proprio esperimento è svolto tramite la macro *Esperimento.m*, in cui in ordine:



(a) Spettro e fase a due lati



(b) Modulo e fase

Figura 5.4: Output funzione FFTtoMP

	1	2	3	4	5	6	7	8	9
1	2	0.3617	0.2727	0.1411	0.0745	-178.0064	157.4704	153.2687	-174.9355
2	2	0.3696	0.2759	0.1409	0.0760	112.4941	-48.3803	171.1365	66.9944
3	2	0.3627	0.2706	0.1385	0.0735	-12.6554	-68.5499	-103.4513	-101.6351
4	2	0.3766	0.2783	0.1403	0.0743	-31.3436	-119.6210	172.6124	141.4497
5	2	0.4534	0.2958	0.1103	0.0494	-117.6267	-17.9071	103.9678	-91.4046
6	4	1.3004	0.1026	0.1490	0.0969	124.2827	-129.9927	-160.8795	-63.0841
7	5	2.5170	0.0819	0.0418	0.0234	0.6759	-47.5686	-165.9186	-65.6838
8	5	2.5171	0.0763	0.0433	0.0236	-53.8358	149.0171	-80.1631	-93.2335
9	1	2.3972	0.0404	0.0143	0.0084	93.0906	-172.6292	76.8053	-123.0725
10	1	2.3961	0.0424	0.0141	0.0083	-12.4529	-128.2705	-98.1659	-139.4813

Classi
Fondamentale, Ampiezza
3° Armonica, Ampiezza
5° Armonica, Ampiezza
7° Armonica, Ampiezza
Fondamentale, Fase
3° Armonica, Fase
5° Armonica, Fase
7° Armonica, Fase

Figura 5.5: Matrice MP

1. carica la matrice MP contenente etichette, ampiezza e fase delle armoniche di corrente;
2. effettua la divisione della matrice MP in $k+1$ subset, k subset di training ed uno di test. La costruzione di questi set è svolta selezionando random gli indici delle curve iniziali PLAID in maniera tale che ogni subset abbia lo stesso numero di apparecchi appartenenti alla stessa classe senza ripetizioni. Ai fini della ripetibilità dell'esperimento è stata inserita la funzione *rng*[3] che blocca il seme del generatore dei numeri casuali;
3. costruisce la foresta decisionale, addestrando ogni albero su ogni k subset di training;
4. richiama la foresta decisionale per classificare gli elettrodomestici del set di test;
5. elabora i risultati sottoponendoli al Majority Vote;
6. calcola la matrice di confusione e gli indicatori di prestazione (*performance metrics*);
7. infine, stampa in video i risultati.

All'interno di tale macro, inoltre, sono state addestrati con i medesimi set di training altre due foreste decisionali ed un singolo albero con lo scopo di determinare, mediante confronto, quante più informazioni circa la

[3]Per ulteriori approfondimenti si rinvia all'help Matlab al link: https://it.mathworks.com/help/matlab/ref/rng.html

bontà della foresta decisionale indotta da C4.5 con IGR. Vengono proposti dunque: una foresta decisionale indotta dall'algoritmo CART per valutare quanto C4.5 sia effettivamente competitivo in ambiente NILM; una foresta decisionale indotta dall'algoritmo C4.5 con IG al fine di dimostrare la qualità del Rapporto del guadagno informatico nei problemi con attributi non binari; un singolo albero decisionale indotto da C4.5 con IGR, utile per osservare la dipendenza del metodo foresta rispetto alla numerosità dei dati di training. I risultati e le conclusioni riguardanti questi confronti sono disponibili al capitolo 6.2 di questo elaborato.

5.2.1 Funzione *Groot*

La funzione **Groot** costruisce in maniera ricorsiva ed automatica un albero decisionale indotto tramite l'algoritmo C4.5 utilizzante il Rapporto del guadagno dell'informazione calcolato con l'entropia di Shannon.

La struttura dell'albero (figura 5.6), e della foresta, è memorizzata tramite una matrice di struttura matlab, *struct*[4], in cui per ogni nodo conserva:

- il numero dell'attributo su cui è avvenuto lo split;
- tutti i valori appartenenti all'attributo, organizzati in ordine crescente senza ripetizioni;
- l'autovalore dell'attributo, ovvero il singolo valore che assume l'attributo all'interno del record su cui si è scelto di effettuare la divisione
- una nuova struct contenente i nodi figli.

Per ogni iterazione e dunque livello dell'albero, il codice ideato effettua in ordine, le seguenti operazioni:

1. calcola l'entropia delle informazioni del nodo corrente;
2. calcola il guadagno informatico per ogni attributo;
3. ricerca il possibile split dividendo il set di dati in due rispetto l'autovalore che massimizza l'IGR;
4. esegue lo split richiamando ricorsivamente la funzione Groot, incrementando di una unità la variabile input *livello*.

Il processo iterativo di costruzione termina quando si compie uno dei tre criteri di stop, ovvero quando:

- il numero dei record è inferiore alla potatura;

[4]Per ulteriori approfondimenti si rinvia all'help Matlab al link: https://it.mathworks.com/help/matlab/ref/struct.html

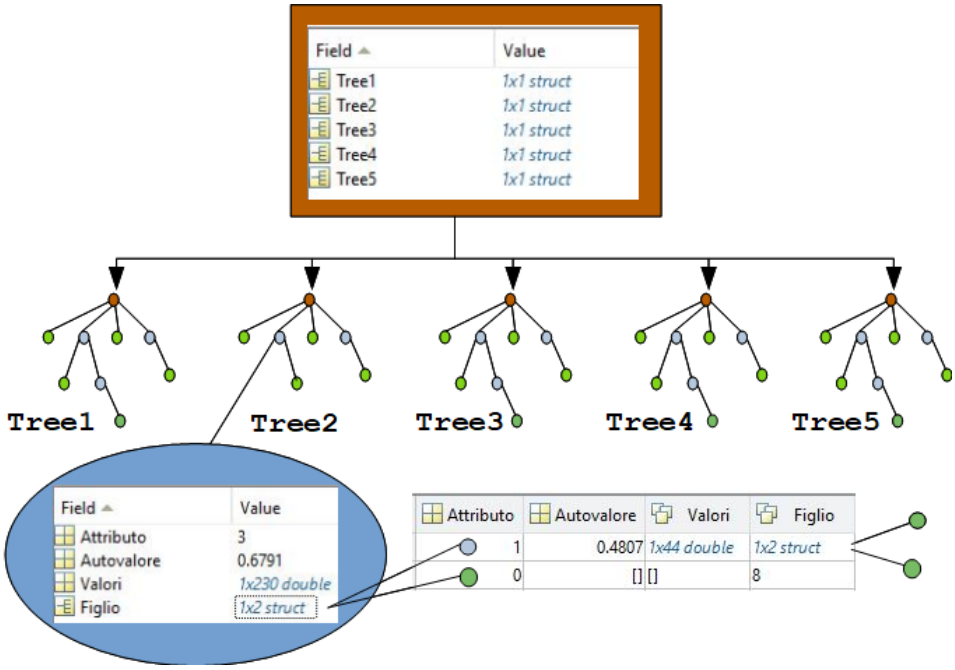


Figura 5.6: Struttura della foresta in matlab

- rimane solamente un record;
- i record rimanenti appartengono tutti alla stessa classe;
- l'attributo contiene valori tutti uguali.

5.2.2 Funzione *Usa_ Groot*

Usa_ Groot classifica i dati all'interno del set di test partendo dalla struttura foresta costruita precedentemente. Gli input della funzione sono il dataset di test; gli indici dei record sotto forma di vettore, ovvero un vettore contenente numeri naturali in ordine crescente rappresentanti i record; l'albero da utilizzare, da già istruito dalla funzione precedente; le classi del dataset di training, utilizzate per ricavare in maniera univoca in quante e quali classi l'albero è in grado di classificare.

L'algoritmo inizia leggendo dalla struttura albero il numero dell'attributo su cui è avvenuto lo *split*. Considerando che è un processo ricorsivo, per ogni passo, ovvero per ogni livello dell'albero, esegue sistematicamente quanto riportato successivamente. Considerando l'n-esimo livello dell'albero, suddivide il set di partenza in due subset, ognuno dei quali diventerà set di partenza per i livelli $n+1$-esimi. Si noti come l'utilizzo del plurale nella frase precedente non sia erroneo poiché l'algoritmo svilupperà più livelli $n+1$. Nel dettaglio, l'algoritmo svilupperà tanti livelli quanti subset, ossia 2.

Il primo subset comprende tutti quei record che posseggono il valore dell'attributo da suddividere minore del o uguale all'autovalore del nodo.
Il secondo subset comprende tutti quei record che posseggono il valore dell'attributo da suddividere maggiore dell'autovalore del nodo. Questa divisione è finalizzata a replicare il processo di costruzione dell'albero, ma nel verso opposto.

Per ogni livello vengono tenuti in memoria gli indici dei record come correlazione univoca tra le classi da predire e i record da classificare. Dal confronto tra l'autovalore tenuto in memoria all'interno dell'albero e i valori dell'attributo in questione, riferito ai subset, l'algoritmo non appena incontrerà un nodo foglia, classificherà i record giunti al nodo con la classe rispettiva della suddetta foglia.

Avvenuta la classificazione, tramite il controllo dell'invocazione *return*[5], ritornerà al livello precedente per proseguire ricorsivamente.

5.2.3 Majority Voting

Ottenute le predizioni dei singoli alberi decisionali appartenenti alla foresta, viene applicata la legge Majority Voting per ottenere la classificazione finale. Il codice comprende poche righe in cui l'algoritmo seleziona, per ogni singolo elettrodomestico, la classe che ha ottenuto un punteggio maggiore tramite la funzione matlab *max*[6], contrassegnandola come decisione finale.

5.3 Valutazione

Per valutare la qualità dell'algoritmo e della classificazione sono stati scritte due funzioni: *TheRISULTATI* e *MConfusione*, i cui output verranno presentati e discussi nel capitolo 6.2.

La scelta di scrivere funzioni indipendenti è stata presa per rendere il codice più generale possibile, sfruttando questa abilità anche per gli ulteriori alberi e foreste generate, rendendolo il più compatto possibile.

5.3.1 Funzione *TheRISULTATI*

L'obiettivo di *TheRISULTATI*, come indica il suo nome, è quello di mostrare in video i risultati ottenuti, elettrodomestico per elettrodomestico, confrontando la classe effettiva con la classe ottenuta dall'esperimento. Dunque, la funzione riceve in input le classi effettive, le classi ricavate dal Majority Voting e i metadati riferiti alle curve di corrente. L'output (figura 5.7) è

[5] Per ulteriori approfondimenti si rinvia all'help Matlab al link: https://it.mathworks.com/help/matlab/ref/return.html

[6] Per ulteriori approfondimenti si rinvia all'help Matlab al link: https://it.mathworks.com/help/matlab/ref/max.html

una tabella in cui viene mostrato, tramite un match, se la classificazione è stata corretta seguita da una stringa contenente la percentuale di correttezza calcolata come segue:

$$PerCentErrore = 100 - \frac{N_{Errori} \cdot 100}{N_{Record}}$$

dove N_{Errori} è il numero degli errori ricavato dal match e N_{Record} è il numero dei record, ovvero il numero delle curve di corrente del set di test.

Predette	Effettive	Match	PLAID
"Air Conditioner"	"Air Conditioner"	"OK!"	"9"
"Air Conditioner"	"Air Conditioner"	"OK!"	"354"
"Microwave"	"Air Conditioner"	"ERRORE"	"404"
"Fan"	"Air Conditioner"	"ERRORE"	"1089"
"Air Conditioner"	"Air Conditioner"	"OK!"	"1065"
"Incandescent Light Bulb"	"Air Conditioner"	"ERRORE"	"1517"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"699"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"1"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"78"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"179"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"140"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"510"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"947"

Figura 5.7: Output funzione *TheRISULTATI*

5.3.2 Funzione *MConfusione*

La funzione *MConfusione* calcola e stampa la matrice di confusione e tutti gli indicatori di prestazione (*performance metrics*) utilizzate nel capitolo 6.2. L'obiettivo è quello di ricavare le metriche per ogni tipologia di elettrodomestici in maniera tale da poter effettuare una valutazione per singole classi.

L'algoritmo è strutturato tramite un ciclo *for* sulle classi. Come primo step calcola i veri positivi, i veri negativi, i falsi positivi e i veri negativi secondo la regola mostrata in figura 5.8. Sulla base di quest'ultimi calcola le metriche tramite le formule riportate nel capitolo 4.4.

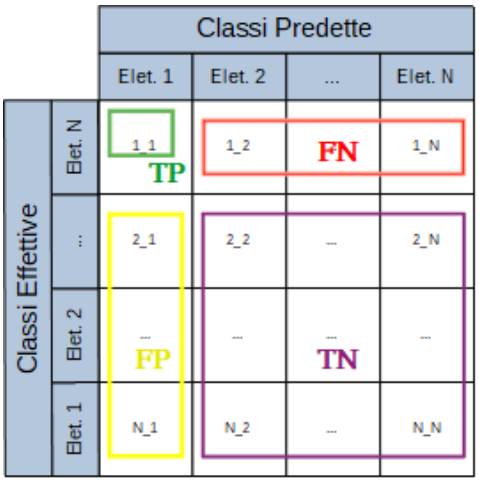


Figura 5.8: Matrice di confusione e i quattro casi possibili

Capitolo 6

Elaborazione dei dati e Risultati

Il presente capitolo mira ad illustrare i risultati ottenuti ed il processo che ha portato al loro conseguimento mediante l'utilizzo dell'algoritmo sviluppato.

6.1 Elaborazione dei dati

Il codice implementato può essere considerato come una serie di due black box, figura 6.1, dove:

- la prima è la macro Matlab *Elaborazione_ MP_ to_ txt.m*, che da singole curve di corrente prelevate dal dataset PLAID restituisce una matrice di valori **MP**, modulo e fase delle armoniche di corrente;
- la seconda è la macro *Esperimento* che riceve in input la matrice **MP** che utilizza per addestra la foresta decisionale. In seguito, tale macro esegue la classificazione mediante la foresta decisionale, ottenendo così le etichette degli elettrodomestici facenti parte del dataset PLAID.

All'interno di tale sezione sarà osservato il dettaglio di tali black boxes, definendo tutti i passaggi necessari ai fini dell'ottenimento dei risultati mostrati nella parte finale di questo capitolo.

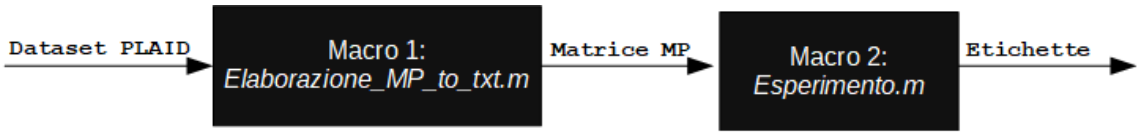


Figura 6.1: Schematizzazione in black box

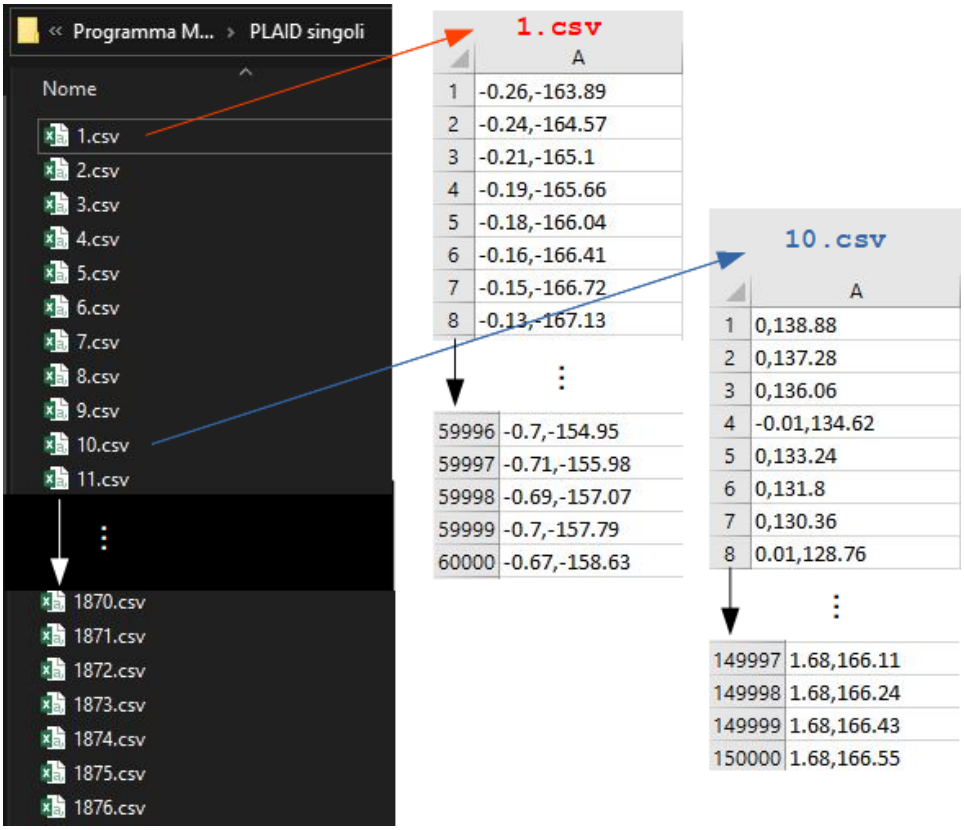
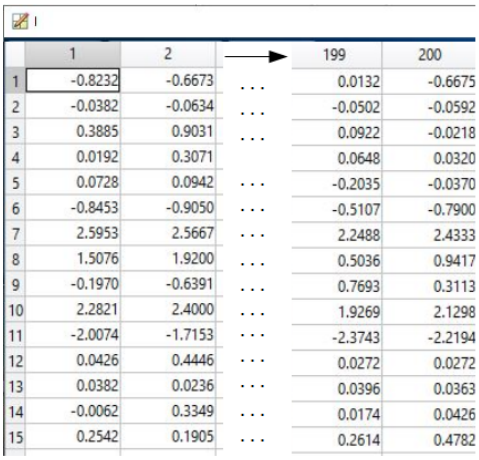


Figura 6.2: Come si presentano le curve del PLAID

Step 1: Le curve di corrente appartenenti al PLAID sono fornite singolarmente e catalogate all'interno di una cartella, figura 6.2, ognuna delle quali è contenuta in un file .csv rinominato semplicemente con un numero da 1 a 1876 uguale al numero totale delle curve presenti. Il primo passo consiste nell'importazione individuale di ciascun file.

Step 2: Prendendo in esempio la Lampada Fluorescente mostrata in figura 5.3 a pagina 36, il secondo passo consiste in un sottocampionamento ed estrazione di un segmento steady-state di 200 campioni, ottenuto tramite una soglia rms, utile nel momento in cui si dovrà calcolare la FFT. Tutti i segmenti steady-state delle curve acquisite sono memorizzati nella matrice **I** rappresentata in figura 6.3.

Step 3: La matrice **I** viene elaborata dalla FFT, ricavando così modulo e fase delle armoniche di corrente alla fondamentale e alle armoniche di ordine 3, 5 e 7, rispettivamente con frequenze 60 Hz, 180 Hz, 300 Hz e 420 Hz. Un esempio di output della FFT è riscontrabile in figura 5.4b a pagina 59.



	1	2		199	200
1	-0.8232	-0.6673	...	0.0132	-0.6675
2	-0.0382	-0.0634	...	-0.0502	-0.0592
3	0.3885	0.9031	...	0.0922	-0.0218
4	0.0192	0.3071		0.0648	0.0320
5	0.0728	0.0942	...	-0.2035	-0.0370
6	-0.8453	-0.9050	...	-0.5107	-0.7900
7	2.5953	2.5667	...	2.2488	2.4333
8	1.5076	1.9200	...	0.5036	0.9417
9	-0.1970	-0.6391	...	0.7693	0.3113
10	2.2821	2.4000	...	1.9269	2.1298
11	-2.0074	-1.7153	...	-2.3743	-2.2194
12	0.0426	0.4446	...	0.0272	0.0272
13	0.0382	0.0236	...	0.0396	0.0363
14	-0.0062	0.3349	...	0.0174	0.0426
15	0.2542	0.1905	...	0.2614	0.4782

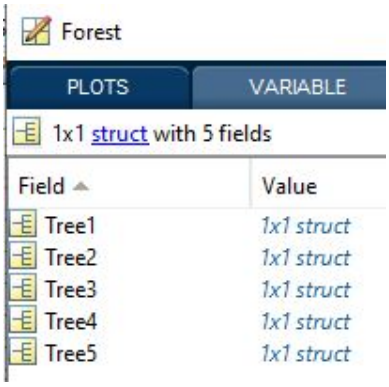
Figura 6.3: Matrice I

Ottenuti i valori di ampiezza e fase per tutte le correnti importate, vengono organizzati all'interno della matrice **MP** strutturata come segue:

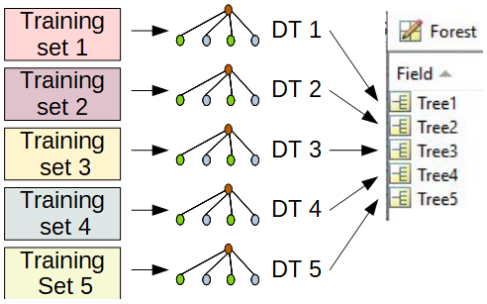
1. nella prima colonna viene memorizzata la classe dell'elettrodomestico;
2. dalla seconda alla quinta colonna sono memorizzati i valori di ampiezza rispettivamente delle armoniche di ordine 1, 3, 5 e 7;
3. dalla sesta alla nona colonna sono memorizzati i valori di fase rispettivamente delle armoniche di ordine 1, 3, 5 e 7;

Step 4: Questo è il passaggio di *boot* della seconda macro *Esperimento*. Tramite il numero delle righe della matrice **MP** il codice stabilisce con precisione quante curve sono state importate, ovvero il numero di record $r_{Dataset}$ e per confronto sulla prima colonna stabilisce il numero totale delle classi c ed il numero dei singoli elettrodomestici facente parte di tali classi. Questi dati ricavati vengono sfruttati per costruire le k matrici di trainig e la matrice di test rispettivamente un numero di record pari a r_{Traini} con $i = 1, 2, ..., k$ e r_{Test} , eseguendo un campionamento random partendo dalla **MP**. Per l'esperimento svolto: $r_{Dataset} = 1746$, $c = 11$, $k = 5$, $r_{Train1} = 230$, $r_{Train2} = 230$, $r_{Train3} = 230$, $r_{Train4} = 230$, $r_{Train5} = 230$, $r_{Test} = 52$. Da notare, dunque, che in ogni subset di training è presente lo stesso numero di elettrodomestici con le classi ripartite equamente.

Step 5: I subset di training vengono utilizzati per addestrare $k = 5$ alberi decisionali, uno per ogni subset. Gli alberi ottenuti sono in seguito raggruppati all'interno della foresta *Forest*, figura 6.4b.



(a) Come si presenta la foresta decisionale in matlab



(b) Processo logico

Figura 6.4: Step 5

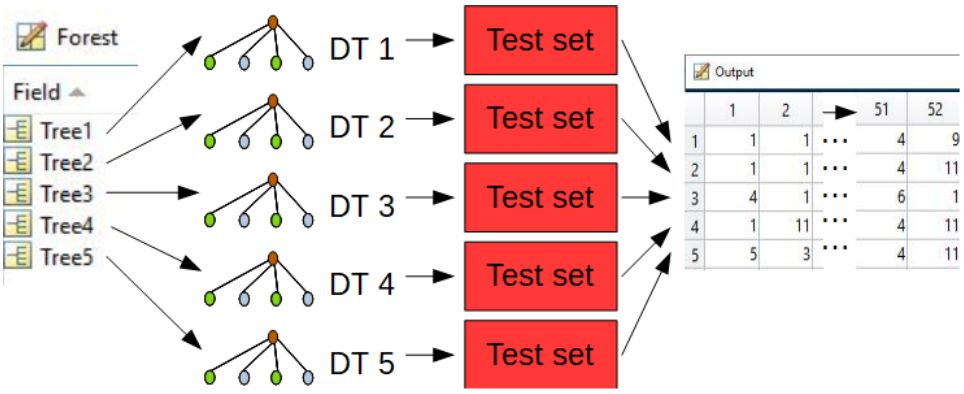


Figura 6.5: Step 6

Step 6: La foresta viene richiamata: ogni albero viene utilizzato per classificare gli elettrodomestici del medesimo subset di test. L'output della classificazione è la matrice 5x52 **Output**, in cui sono memorizzate le predizioni.

Step 7: La matrice **Output** viene presa in carico dal calcolatore del *majority vote* che seleziona, per ogni elettrodomestico, la predizione con più numerosità. Si ottiene il vettore $PREDIZIONI$, in figura 6.6, con il risultato finale della classificazione.

Step 8: Tramite le funzioni *TheRISULTATI* e *MConfusione* vengono calcolati i risultati descritti nella prossima sezione.

Figura 6.6: Vettore predizioni

6.2 Risultati

Al fine di valutare le capacità predittive del metodo sviluppato, si è svolta una comparazione con altre tipologie di alberi e foreste decisionali. Seguono i tre confronti effettuati:

- Singolo albero decisionale, ai fini di valutare quanto una foresta sia preferibile rispetto ad un singolo albero;
- Foresta decisionale con induttore CART tramite le funzioni *fitctree*[1] e *predict*[2] già implementate in Matlab;
- Foresta decisionale con induttore C4.5 con Guadagno dell'Informazione tramite la funzione *C4_5*[3] scritta dall'Ing. Wissem Boussetta (National School for Computer Science, Tunisia).

6.2.1 Risultati Foresta decisionale C4.5 con IGR

In questa sezione si presentano i risultati del metodo proposto. Gli Errori nella classificazione della foresta decisionale C4.5 con IGR sono 14 su 52, con una percentuale di correttezza del 73.08%. Nella tabella 6.1 sono riportati i risultati per ogni singolo elettrodomestico, calcolati tramite la funzione *TheRISULTATI*. La figura 6.8 riporta la matrice di confusione mentre nella tabella 6.2 sono indicati i valori delle *performance metric* per ogni classe, calcolati all'interno della funzione *MConfusione*.

A conclusione di tale analisi è possibile affermare che la foresta risulta avere notevoli abilità nella classificazione degli elettrodomestici aventi forma d'onda di corrente molto distorta, ovvero una grande presenza di armoniche successive (e.g., forno a microonde e aspirapolvere). E' interessante rilevare, inoltre, come la medesima foresta riscontri delle difficoltà per quanto concerne la classificazione degli elettrodomestici con forma d'onda di corrente molto pulita (e.g., stufa e frigorifero).

[1] Per ulteriori approfondimenti si rinvia all'help Matlab al link: https://it.mathworks.com/help/stats/fitctree.html

[2] Per ulteriori approfondimenti si rinvia all'help Matlab al link: https://it.mathworks.com/help/stats/linearmodel.predict.html

[3] Reperibile al sito: https://github.com/Boussetta/C4.5

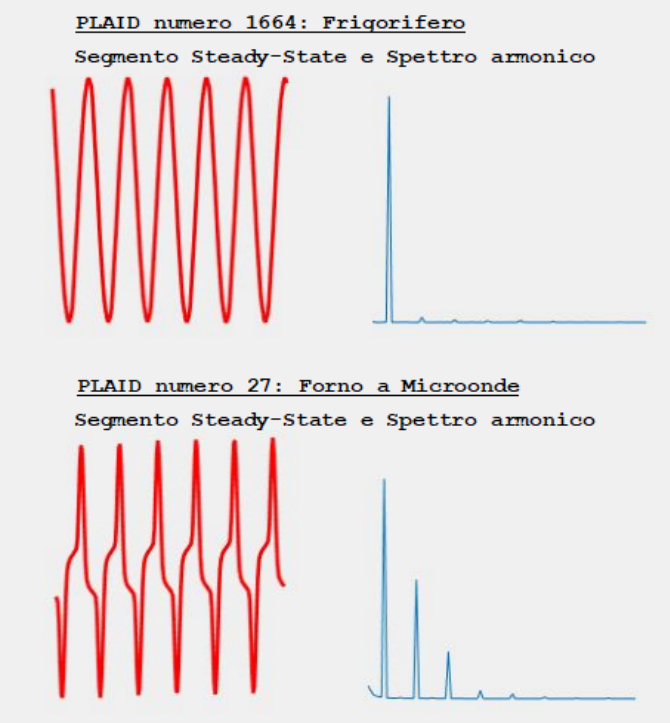


Figura 6.7: Confronto tra forma d'onda poco distorta e distorta

6.2.2 Risultati Albero decisionale C4.5 con IGR

In questa sezione si presentano i risultati del metodo proposto con un singolo albero decisionale. Gli errori nella classificazione sono 12 su 52, con una percentuale di correttezza del 76.92%. Nella tabella 6.3 sono riportati i risultati per ogni singolo elettrodomestico. La figura 6.9 riporta la matrice di confusione mentre nella tabella 6.4 sono indicati i valori delle *performance metric* per ogni classe.

Comparando i risultati raggiunti dal singolo albero, "addestrato" su un ampio dataset somma dei cinque dataset di training delle foreste, con quelli raggiunti dalla foresta in tutte le classi di elettrodomestici, è possibile rilevare come siano migliori le prestazioni del primo. Il singolo albero, infatti, presenta maggiori capacità di classificare apparecchi aventi forma d'onda sia molto distorta (e.g., forno a microonde e aspirapolvere) che poco distorta, sebbene presenti difficoltà nel predire la classe ad elettrodomestici con forma d'onda molto regolare (e.g., frigorifero e stufa).

6.2.3 Risultati Foresta decisionale CART

In questa sezione si presentano i risultati generati dalla foresta decisionale con induttore CART. Gli errori nella classificazione della foresta sono 11 su

52, con una percentuale di correttezza del 78.85%. Nella tabella 6.5 sono riportati i risultati per ogni singolo elettrodomestico. La figura 6.10 riporta la matrice di confusione mentre nella tabella 6.6 sono indicati i valori delle *performance metrics* per ogni classe.

6.2.4 Risultati Foresta decisionale C4.5 con IG

In questa sezione si presentano i risultati generati dalla foresta decisionale con induttore C4.5 con IG. Gli errori nella classificazione della foresta sono 19 su 52, con una percentuale di correttezza del 63.46%. Nella tabella 6.7 sono riportati i risultati per ogni singolo elettrodomestico. La figura 6.11 riporta la matrice di confusione mentre nella tabella 6.8 sono indicati i valori delle *performance metrics* per ogni classe.

6.2.5 Confronto

Nell'immagine 6.12 sono riportati in forma grafica i risultati di accuratezza calcolati ai fini del confronto. È possibile evincere che:

1. La foresta decisionale utilizzante l'algoritmo di induzione CART ha una maggiore accuratezza nella predizione rispetto a tutte le altre implementate. Questo risultato è da ricondurre al fatto che l'algoritmo in questione utilizza solamente split binari su attributi continui, o discreti trattati come continui, e il coefficiente di Gini [14] come misura dell'impurità.

2. Come già discusso in linea teorica nel capitolo 4.3.1, nell'induttore C4.5 è vincente la scelta del Rapporto del guadagno dell'informazione come misura di impurità rispetto al semplice Guadagno del rapporto.

3. Come nel caso in esame, ovvero dataset con un numero molto limitato di curve, è preferibile implementare metodi a singolo albero rispetto che a foreste.

4. La Cross Validation applicata agli alberi decisionali ai fini della risoluzione di problemi NILM è sicuramente una tecnica molto efficace per eliminare il problema dell'overfitting nei training-set.

Classi predette	Classi effettive	Match	Indice PLAID
"Air Conditioner"	"Air Conditioner"	"OK!"	"9"
"Air Conditioner"	"Air Conditioner"	"OK!"	"354"
"Air Conditioner"	"Air Conditioner"	"OK!"	"404"
"Fan"	"Air Conditioner"	"ERRORE"	"1089"
"Air Conditioner"	"Air Conditioner"	"OK!"	"1065"
"Air Conditioner"	"Air Conditioner"	"OK!"	"1517"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"699"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"1"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"78"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"179"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"140"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"510"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"947"
"Fan"	"Fan"	"OK!"	"654"
"Incandescent Light Bulb"	"Fan"	"ERRORE"	"56"
"Air Conditioner"	"Fan"	"ERRORE"	"211"
"Incandescent Light Bulb"	"Fan"	"ERRORE"	"298"
"Fan"	"Fan"	"OK!"	"259"
"Fan"	"Fan"	"OK!"	"416"
"Fan"	"Fan"	"OK!"	"1355"
"Incandescent Light Bulb"	"Fridge"	"ERRORE"	"1664"
"Fridge"	"Fridge"	"OK!"	"1131"
"Air Conditioner"	"Fridge"	"ERRORE"	"6"
"Air Conditioner"	"Hairdryer"	"ERRORE"	"738"
"Hairdryer"	"Hairdryer"	"OK!"	"585"
"Hairdryer"	"Hairdryer"	"OK!"	"7"
"Air Conditioner"	"Hairdryer"	"ERRORE"	"192"
"Hairdryer"	"Hairdryer"	"OK!"	"143"
"Hairdryer"	"Hairdryer"	"OK!"	"1234"
"Hairdryer"	"Hairdryer"	"OK!"	"264"
"Hairdryer"	"Hairdryer"	"OK!"	"630"
"Hairdryer"	"Heater"	"ERRORE"	"1152"
"Hairdryer"	"Heater"	"ERRORE"	"1299"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"50"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"320"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"446"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"566"
"Fan"	"Incandescent Light Bulb"	"ERRORE"	"1027"
"Compact Fluorescent Lamp"	"Laptop"	"ERRORE"	"338"
"Laptop"	"Laptop"	"OK!"	"23"
"Laptop"	"Laptop"	"OK!"	"482"
"Compact Fluorescent Lamp"	"Laptop"	"ERRORE"	"890"
"Microwave"	"Microwave"	"OK!"	"27"
"Microwave"	"Microwave"	"OK!"	"250"
"Microwave"	"Microwave"	"OK!"	"315"
"Microwave"	"Microwave"	"OK!"	"647"
"Microwave"	"Microwave"	"OK!"	"560"
"Microwave"	"Microwave"	"OK!"	"1317"
"Vacuum"	"Vacuum"	"OK!"	"1006"
"Vacuum"	"Vacuum"	"OK!"	"1406"
"Fridge"	"Washing Machine"	"ERRORE"	"1055"
"Washing Machine"	"Washing Machine"	"OK!"	"1410"

Tabella 6.1: Foresta decisionale C4.5 con IGR: Match tra classi predette e classi effettive

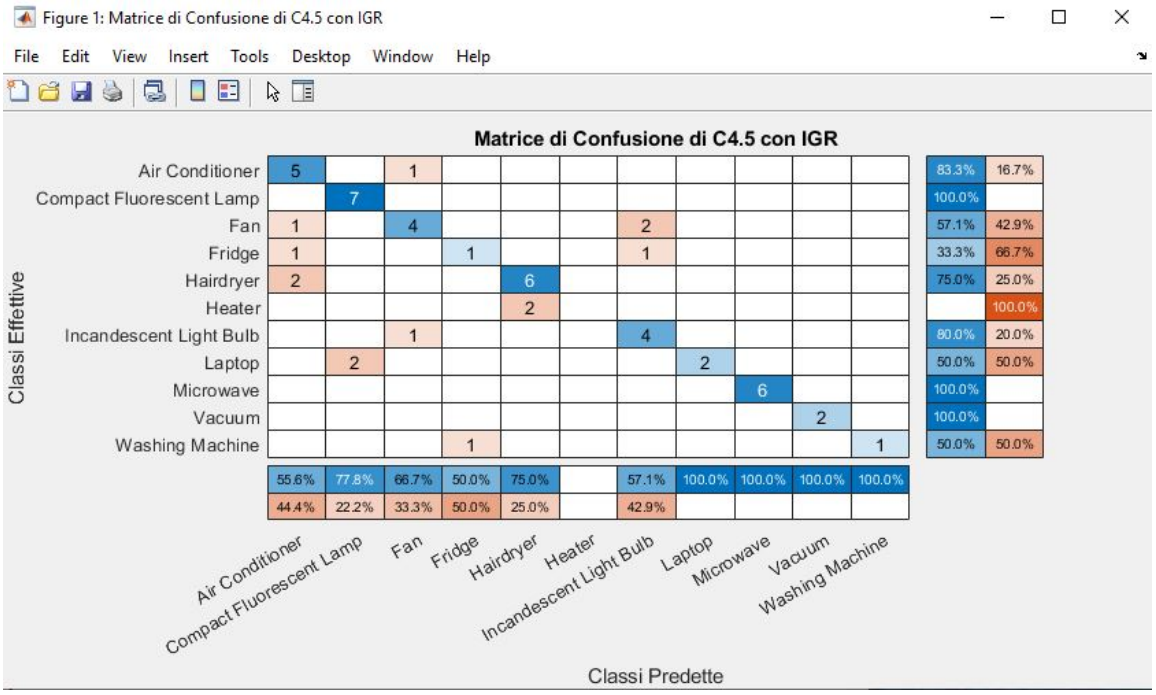


Figura 6.8: Foresta decisionale C4.5 con IGR: Matrice di confusione

Classe	Accuratezza	Precisione	Richiamo	F1 Score
"Air Conditioner"	0.90385	0.55556	0.83333	0.66667
"Compact Fluorescent Lamp"	0.96154	0.77778	1	0.875
"Fan"	0.90385	0.66667	0.57143	0.61538
"Fridge"	0.94231	0.5	0.33333	0.4
"Hairdryer"	0.92308	0.75	0.75	0.75
"Heater"	0.96154	NaN	0	NaN
"Incandescent Light Bulb"	0.92308	0.57143	0.8	0.66667
"Laptop"	0.96154	1	0.5	0.66667
"Microwave"	1	1	1	1
"Vacuum"	1	1	1	1
"Washing Machine"	0.98077	1	0.5	0.66667

Tabella 6.2: Foresta decisionale C4.5 con IGR: *performance metrics*

Classi predette	Classi effettive	Match	Indice PLAID
"Air Conditioner"	"Air Conditioner"	"OK!"	"9"
"Air Conditioner"	"Air Conditioner"	"OK!"	"354"
"Air Conditioner"	"Air Conditioner"	"OK!"	"404"
"Air Conditioner"	"Air Conditioner"	"OK!"	"1089"
"Air Conditioner"	"Air Conditioner"	"OK!"	"1065"
"Air Conditioner"	"Air Conditioner"	"OK!"	"1517"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"699"
"Laptop"	"Compact Fluorescent Lamp"	"ERRORE"	"1"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"78"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"179"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"140"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"510"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"947"
"Fan"	"Fan"	"OK!"	"654"
"Incandescent Light Bulb"	"Fan"	"ERRORE"	"56"
"Fan"	"Fan"	"OK!"	"211"
"Incandescent Light Bulb"	"Fan"	"ERRORE"	"298"
"Fan"	"Fan"	"OK!"	"259"
"Fan"	"Fan"	"OK!"	"416"
"Air Conditioner"	"Fan"	"ERRORE"	"1355"
"Incandescent Light Bulb"	"Fridge"	"ERRORE"	"1664"
"Fridge"	"Fridge"	"OK!"	"1131"
"Air Conditioner"	"Fridge"	"ERRORE"	"6"
"Hairdryer"	"Hairdryer"	"OK!"	"738"
"Hairdryer"	"Hairdryer"	"OK!"	"585"
"Hairdryer"	"Hairdryer"	"OK!"	"7"
"Heater"	"Hairdryer"	"ERRORE"	"192"
"Hairdryer"	"Hairdryer"	"OK!"	"143"
"Hairdryer"	"Hairdryer"	"OK!"	"1234"
"Hairdryer"	"Hairdryer"	"OK!"	"264"
"Hairdryer"	"Hairdryer"	"OK!"	"630"
"Hairdryer"	"Heater"	"ERRORE"	"1152"
"Hairdryer"	"Heater"	"ERRORE"	"1299"
"Fan"	"Incandescent Light Bulb"	"ERRORE"	"50"
"Fridge"	"Incandescent Light Bulb"	"ERRORE"	"320"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"446"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"566"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"1027"
"Laptop"	"Laptop"	"OK!"	"338"
"Laptop"	"Laptop"	"OK!"	"23"
"Compact Fluorescent Lamp"	"Laptop"	"ERRORE"	"482"
"Laptop"	"Laptop"	"OK!"	"890"
"Microwave"	"Microwave"	"OK!"	"27"
"Microwave"	"Microwave"	"OK!"	"250"
"Microwave"	"Microwave"	"OK!"	"315"
"Microwave"	"Microwave"	"OK!"	"647"
"Microwave"	"Microwave"	"OK!"	"560"
"Microwave"	"Microwave"	"OK!"	"1317"
"Vacuum"	"Vacuum"	"OK!"	"1006"
"Vacuum"	"Vacuum"	"OK!"	"1406"
"Washing Machine"	"Washing Machine"	"OK!"	"1055"
"Washing Machine"	"Washing Machine"	"OK!"	"1410"

Tabella 6.3: Albero decisionale C4.5 con IGR: Match tra classi predette e classi effettive

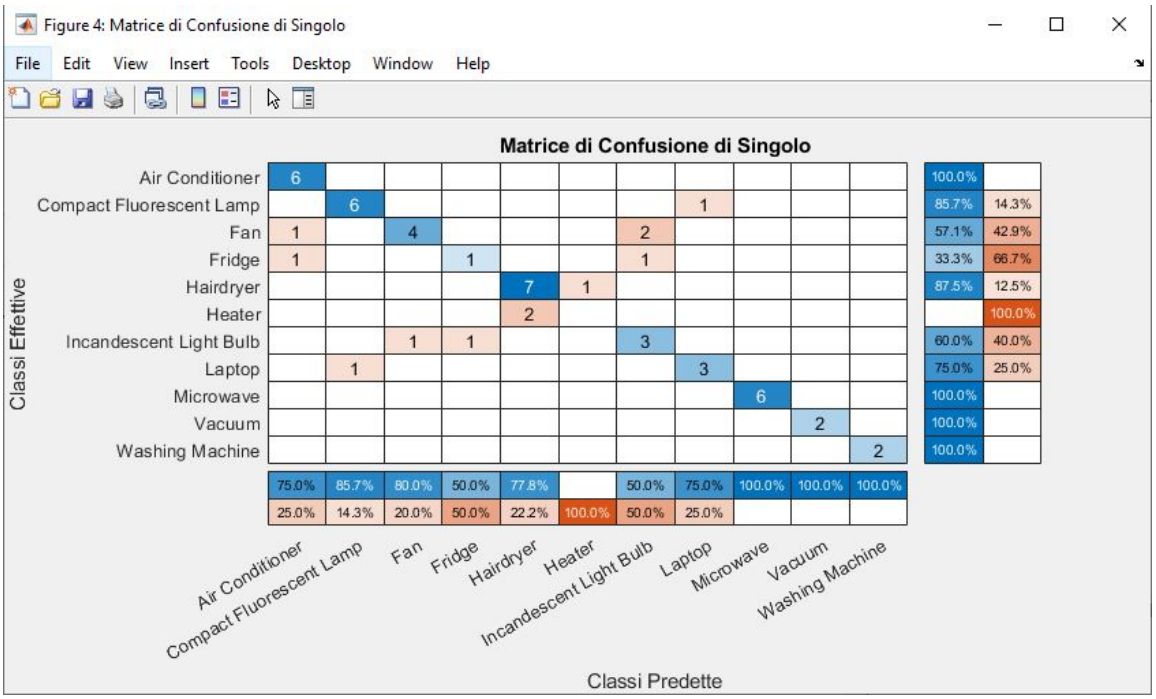


Figura 6.9: Albero decisionale C4.5 con IGR: Matrice di confusione

Classe	Accuratezza	Precisione	Richiamo	F1 Score
"Air Conditioner"	0.96154	0.75	1	0.85714
"Compact Fluorescent Lamp"	0.96154	0.85714	0.85714	0.85714
"Fan"	0.92308	0.8	0.57143	0.66667
"Fridge"	0.94231	0.5	0.33333	0.4
"Hairdryer"	0.94231	0.77778	0.875	0.82353
"Heater"	0.94231	0	0	NaN
"Incandescent Light Bulb"	0.90385	0.5	0.6	0.54545
"Laptop"	0.96154	0.75	0.75	0.75
"Microwave"	1	1	1	1
"Vacuum"	1	1	1	1
"Washing Machine"	1	1	1	1

Tabella 6.4: Albero decisionale C4.5 con IGR: *performance metrics*

Classi predette	Classi effettive	Match	Indice PLAID
"Air Conditioner"	"Air Conditioner"	"OK!"	"9"
"Air Conditioner"	"Air Conditioner"	"OK!"	"354"
"Hairdryer"	"Air Conditioner"	"ERRORE"	"404"
"Air Conditioner"	"Air Conditioner"	"OK!"	"1089"
"Air Conditioner"	"Air Conditioner"	"OK!"	"1065"
"Air Conditioner"	"Air Conditioner"	"OK!"	"1517"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"699"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"1"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"78"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"179"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"140"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"510"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"947"
"Fan"	"Fan"	"OK!"	"654"
"Fan"	"Fan"	"OK!"	"56"
"Fan"	"Fan"	"OK!"	"211"
"Fan"	"Fan"	"OK!"	"298"
"Air Conditioner"	"Fan"	"ERRORE"	"259"
"Fan"	"Fan"	"OK!"	"416"
"Fan"	"Fan"	"OK!"	"1355"
"Incandescent Light Bulb"	"Fridge"	"ERRORE"	"1664"
"Fridge"	"Fridge"	"OK!"	"1131"
"Air Conditioner"	"Fridge"	"ERRORE"	"6"
"Air Conditioner"	"Hairdryer"	"ERRORE"	"738"
"Hairdryer"	"Hairdryer"	"OK!"	"585"
"Hairdryer"	"Hairdryer"	"OK!"	"7"
"Hairdryer"	"Hairdryer"	"OK!"	"192"
"Hairdryer"	"Hairdryer"	"OK!"	"143"
"Hairdryer"	"Hairdryer"	"OK!"	"1234"
"Hairdryer"	"Hairdryer"	"OK!"	"264"
"Hairdryer"	"Hairdryer"	"OK!"	"630"
"Hairdryer"	"Heater"	"ERRORE"	"1152"
"Hairdryer"	"Heater"	"ERRORE"	"1299"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"50"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"320"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"446"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"566"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"1027"
"Compact Fluorescent Lamp"	"Laptop"	"ERRORE"	"338"
"Laptop"	"Laptop"	"OK!"	"23"
"Laptop"	"Laptop"	"OK!"	"482"
"Compact Fluorescent Lamp"	"Laptop"	"ERRORE"	"890"
"Microwave"	"Microwave"	"OK!"	"27"
"Microwave"	"Microwave"	"OK!"	"250"
"Microwave"	"Microwave"	"OK!"	"315"
"Microwave"	"Microwave"	"OK!"	"647"
"Microwave"	"Microwave"	"OK!"	"560"
"Microwave"	"Microwave"	"OK!"	"1317"
"Vacuum"	"Vacuum"	"OK!"	"1006"
"Vacuum"	"Vacuum"	"OK!"	"1406"
"Fridge"	"Washing Machine"	"ERRORE"	"1055"
"Microwave"	"Washing Machine"	"ERRORE"	"1410"

Tabella 6.5: Foresta decisionale CART: Match tra classi predette e classi effettive

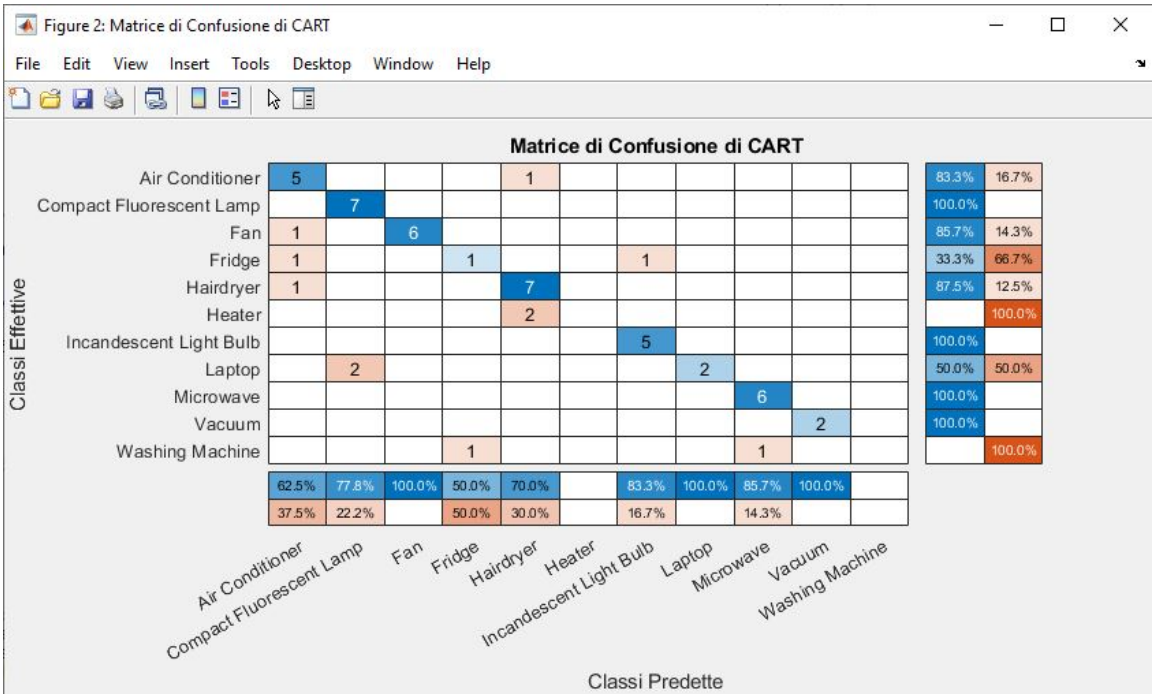


Figura 6.10: Foresta decisionale CART: Matrice di confusione

Classe	Accuratezza	Precisione	Richiamo	F1 Score
"Air Conditioner"	0.92308	0.625	0.83333	0.71429
"Compact Fluorescent Lamp"	0.96154	0.77778	1	0.875
"Fan"	0.98077	1	0.85714	0.92308
"Fridge"	0.94231	0.5	0.33333	0.4
"Hairdryer"	0.92308	0.7	0.875	0.77778
"Heater"	0.96154	NaN	0	NaN
"Incandescent Light Bulb"	0.98077	0.83333	1	0.90909
"Laptop"	0.96154	1	0.5	0.66667
"Microwave"	0.98077	0.85714	1	0.92308
"Vacuum"	1	1	1	1
"Washing Machine"	0.96154	NaN	0	NaN

Tabella 6.6: Foresta decisionale CART: *performance metrics*

Classi predette	Classi effettive	Match	Indice PLAID
"Air Conditioner"	"Air Conditioner"	"OK!"	"9"
"Air Conditioner"	"Air Conditioner"	"OK!"	"354"
"Microwave"	"Air Conditioner"	"ERRORE"	"404"
"Fan"	"Air Conditioner"	"ERRORE"	"1089"
"Air Conditioner"	"Air Conditioner"	"OK!"	"1065"
"Incandescent Light Bulb"	"Air Conditioner"	"ERRORE"	"1517"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"699"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"1"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"78"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"179"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"140"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"510"
"Compact Fluorescent Lamp"	"Compact Fluorescent Lamp"	"OK!"	"947"
"Laptop"	"Fan"	"ERRORE"	"654"
"Fan"	"Fan"	"OK!"	"56"
"Fan"	"Fan"	"OK!"	"211"
"Fan"	"Fan"	"OK!"	"298"
"Air Conditioner"	"Fan"	"ERRORE"	"259"
"Air Conditioner"	"Fan"	"ERRORE"	"416"
"Fan"	"Fan"	"OK!"	"1355"
"Compact Fluorescent Lamp"	"Fridge"	"ERRORE"	"1664"
"Fridge"	"Fridge"	"OK!"	"1131"
"Hairdryer"	"Fridge"	"ERRORE"	"6"
"Hairdryer"	"Hairdryer"	"OK!"	"738"
"Hairdryer"	"Hairdryer"	"OK!"	"585"
"Hairdryer"	"Hairdryer"	"OK!"	"7"
"Microwave"	"Hairdryer"	"ERRORE"	"192"
"Air Conditioner"	"Hairdryer"	"ERRORE"	"143"
"Hairdryer"	"Hairdryer"	"OK!"	"1234"
"Air Conditioner"	"Hairdryer"	"ERRORE"	"264"
"Hairdryer"	"Hairdryer"	"OK!"	"630"
"Microwave"	"Heater"	"ERRORE"	"1152"
"Heater"	"Heater"	"OK!"	"1299"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"50"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"320"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"446"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"566"
"Incandescent Light Bulb"	"Incandescent Light Bulb"	"OK!"	"1027"
"Compact Fluorescent Lamp"	"Laptop"	"ERRORE"	"338"
"Laptop"	"Laptop"	"OK!"	"23"
"Compact Fluorescent Lamp"	"Laptop"	"ERRORE"	"482"
"Incandescent Light Bulb"	"Laptop"	"ERRORE"	"890"
"Microwave"	"Microwave"	"OK!"	"27"
"Microwave"	"Microwave"	"OK!"	"250"
"Microwave"	"Microwave"	"OK!"	"315"
"Microwave"	"Microwave"	"OK!"	"647"
"Microwave"	"Microwave"	"OK!"	"560"
"Microwave"	"Microwave"	"OK!"	"1317"
"Microwave"	"Vacuum"	"ERRORE"	"1006"
"Microwave"	"Vacuum"	"ERRORE"	"1406"
"Microwave"	"Washing Machine"	"ERRORE"	"1055"
"Fan"	"Washing Machine"	"ERRORE"	"1410"

Tabella 6.7: Foresta decisionale C4.5 con IG: Match tra classi predette e classi effettive

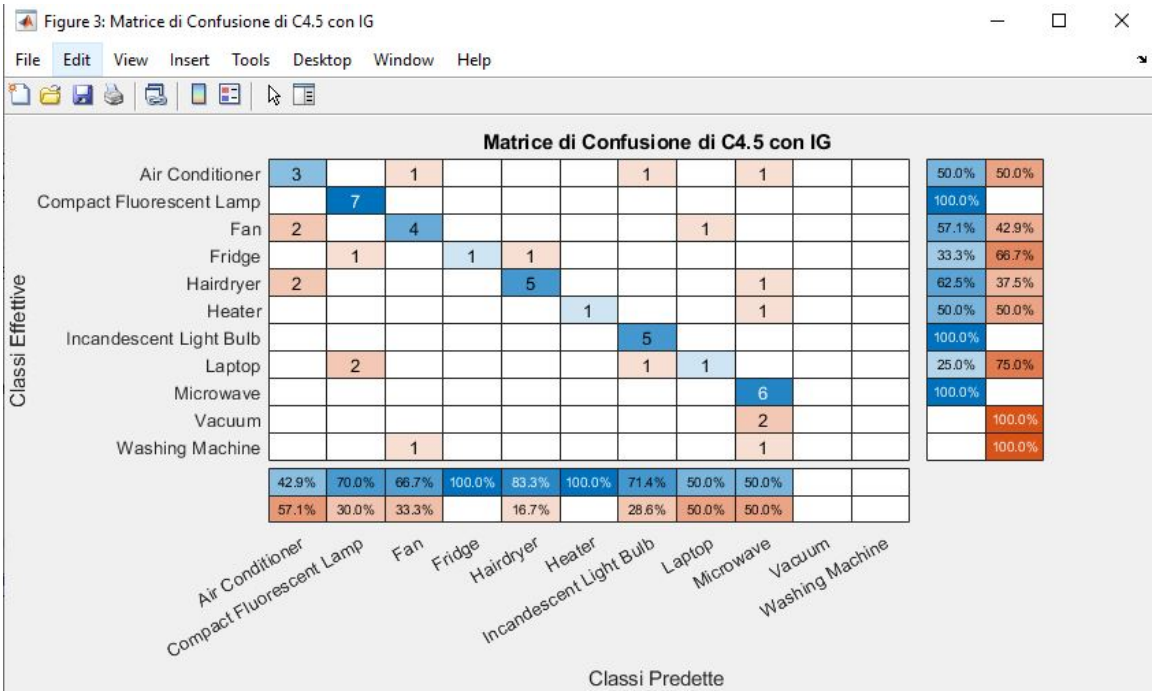


Figura 6.11: Foresta decisionale C4.5 con IG: Matrice di confusione

Classe	Accuratezza	Precisione	Richiamo	F1 Score
"Air Conditioner"	0.86538	0.42857	0.5	0.46154
"Compact Fluorescent Lamp"	0.94231	0.7	1	0.82353
"Fan"	0.90385	0.66667	0.57143	0.61538
"Fridge"	0.96154	1	0.33333	0.5
"Hairdryer"	0.92308	0.83333	0.625	0.71429
"Heater"	0.98077	1	0.5	0.66667
"Incandescent Light Bulb"	0.96154	0.71429	1	0.83333
"Laptop"	0.92308	0.5	0.25	0.33333
"Microwave"	0.88462	0.5	1	0.66667
"Vacuum"	0.96154	NaN	0	NaN
"Washing Machine"	0.96154	NaN	0	NaN

Tabella 6.8: Foresta decisionale C4.5 con IG: *performance metrics*

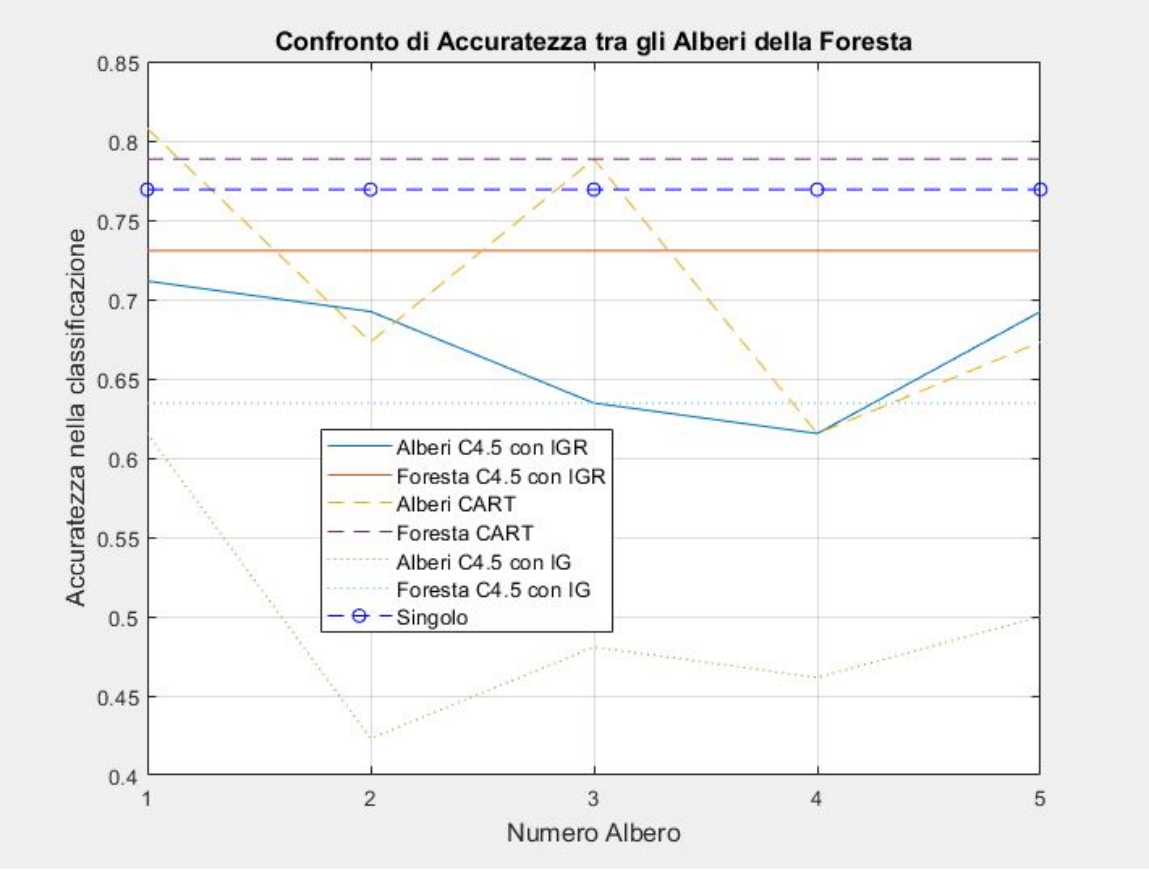


Figura 6.12: Confronto: accuratezza dei diversi metodi

Parte III

Conclusioni

Capitolo 7

Riepilogo

7.1 Riepilogo parte I: Stato dell'arte

La parte I di questa tesi si è concentrata sull'analisi del carico, introducendo i contenuti dell'analisi dei Cluster e approfondendone i principali concetti del monitoraggio del carico non intrusivo, soffermandosi anche su tutto ciò che riguarda i dataset finalizzati a tale scopi. Tale parte dell'elaborato mira a fornire al lettore un quadro su come la comunità scientifica sta cercando di affrontare queste tematiche assai attuali nei giorni nostri e nel prossimo futuro.

Il clustering è una procedura che individua divisioni interne accettabili, dal punto di vista logico, di un dataset spesso ritenuto troppo grande o eterogeneo per essere trattato come unico. L'analisi dei cluster emula un'attività istintiva della mente umana di confrontare oggetti diversi cercandone somiglianze e differenze per categorie via via più distinte.

La disaggregazione dell'energia ha applicazioni molto diffuse, poiché consente di fornire la capacità di identificare e tracciare con precisione il comportamento e lo stato di sistemi elettrici senza la richiesta di grandi modifiche all'impianto o strumenti personalizzati per ciascun elettrodomestico o carico. A poter trarre beneficio dall'analisi svolta è qualsiasi sistema che utilizza energia elettrica, data l'applicabilità di questa analisi al contesto economico, sociale ed ambientale.

7.2 Riepilogo parte II: Classificazione degli elettrodomestici tramite firma armonica e foresta decisionale

La parte II del presente documento mira a proporre una soluzione ad una delle più grande sfide del monitoraggio non intrusivo del carico: la classificazione di apparecchi che presentano un consumo energetico simile.

Al fine di conseguire tale classificazione è stato proposto un metodo implementato in Matlab e che si basa sulla firma armonica ricavata dal calcolo della trasformata veloce di Fourier con l'algoritmo di Cooley-Tukey. L'utilizzo di tali elementi e tecnicismi è opportunamente preceduto da una introduzione descrittiva degli stessi e del loro utilizzo. A partire dal dataset pubblico PLAID, è stato ricavato un nuovo set di dati MP contenente l'ampiezza e la fase della fondamentale a 60 Hz e delle armoniche di ordine 3, 5 e 7 di corrente, rispettivamente con frequenze 180 Hz, 300 Hz e 420 Hz.

Il fine ultimo della presente analisi, pienamente raggiunto, era quello di elaborare un metodo di data mining efficiente per la classificazione degli elettrodomestici mediante il nuovo set MP, utilizzato come input per l'apprendimento di una Foresta decisionale, e di un singolo albero, pilotata dall'algoritmo C4.5 di Quinlan e Ross con Rapporto del Guadagno di Informazione e con l'Entropia di Shannon come misura dell'impurità.

Il modello Foresta ha funzionato bene su un vasto insieme di classi di elettrodomestici in cui la distorsione armonica della corrente era molto presente mentre ha avuto prestazioni inferiori con elettrodomestici aventi forma d'onda tendente alla sinusoide perfetta. Il modello Albero singolo ha avuto risultati molto incoraggianti, dimostrando di essere in grado di classificare correttamente anche quegli elettrodomestici la cui forma d'onda di corrente era meno distorta.

Capitolo 8

Lavori futuri

L'obiettivo principale dei futuri lavori sarà quello di sviluppare un'interfaccia unificata mediante la quale sia possibile elaborare più tipologie di dati proveniente da dataset diversi, scrivendo anche nuove funzioni ad hoc per ciascuno di tali dataset. L'utilizzo di tale nuova forma del codice fornirà nuovi mezzi efficaci ai fini della disaggregazione, consentendo alla conseguente interfaccia implementata di riportare in modo automatico lo stato dei carichi elettrici da più fonti diverse. E' innegabile come NILM trarrà sempre vantaggio ed occasione di sviluppo dalla creazione di nuovi metodi di identificazione e diagnostica.

Una volta effettuati tali lavori sarebbe interessante, ancorché auspicabile, che lo sviluppo mirasse a campionare quante più curve di corrente a 50 Hz nel nostro Paese, dando vita ad un nuovo dataset per le finalità NILM ma anche per studiarne l'effettiva qualità in punti diversi del territorio.

Innegabile è anche la grande importanza che avrebbe lo sviluppo di un modello combinato in cui siano addestrate contemporaneamente più foreste con più algoritmi di induzione. Tale modello, infatti, consentirebbe non solo l'aumento dell'accuratezza del metodo generale, ma anche di aumentare la specificità rispetto alle classi, implementando un codice capace di scegliere l'induttore corretto per ogni classe specifica di elettrodomestici. La creazione di un algoritmo in grado di modellarsi autonomamente in base ai valori in ingresso costituisce un obiettivo assai sfidante dal punto di vista tecnologico, ma con evidente impatto migliorativo della quotidianità data la sua applicabilità a più contesti e soluzioni.

La sfida a lungo termine sarà quella di riuscire a riconoscere vari tipi di carico all'interno di un appartamento, a partire da curve di carico aggregate.

Bibliografia

[1] I. Abubakar, S.N. Khalid, M.W. Mustafa, Hussain Shareef, and M. Mustapha. Application of load monitoring in appliances' energy management – a review. *Renewable and Sustainable Energy Reviews*, 67:235–245, 2017.

[2] MICHAEL R Anderberg. Cluster analysis for applications (no. oas-tr-73-9). *Office of the assistant for study support kirtland AFB n mex*, 1973.

[3] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[4] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[5] Sarah Darby et al. The effectiveness of feedback on energy consumption. *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, 486(2006):26, 2006.

[6] Leen De Baets, Joeri Ruyssinck, Chris Develder, Tom Dhaene, and Dirk Deschrijver. On the bayesian optimization and robustness of event detection methods in nilm. *Energy and Buildings*, 145:57–66, 2017.

[7] Commissione Europea. Green deal. https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_en.

[8] Commissione Europea. Quadro 2030 per il clima e l'energia. https://ec.europa.eu/clima/policies/strategies/2030_it.

[9] Brian S Everitt. Unresolved problems in cluster analysis. *Biometrics*, pages 169–181, 1979.

[10] Luigi Fabbris. *L'indagine campionaria: metodi, disegni e tecniche di campionamento*. La Nuova Italia Scientifica, 1989.

[11] Jingkun Gao, Suman Giri, Emre Can Kara, and Mario Bergés. Plaid: a public dataset of high-resoultion electrical appliance measurements for load identification research: demo abstract. In *proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, pages 198–199, 2014.

[12] Paul E Green, Ronald E Frank, and Patrick J Robinson. Cluster analysis in test market selection. *Management science*, 13(8):B–387, 1967.

[13] Michael Heideman, Don Johnson, and Charles Burrus. Gauss and the history of the fast fourier transform. *IEEE ASSP Magazine*, 1(4):14–21, 1984.

[14] Wikipedia Italia. Coefficiente di gini. https://it.wikipedia.org/wiki/Coefficiente_di_Gini.

[15] Nicholas Jardine and Robin Sibson. Mathematical taxonomy. Technical report, 1971.

[16] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[17] Matthias Kahl, Anwar Haq, Thomas Kriechbaumer, and Hans-Arno Jacobsen. Whited - a worldwide household and industry transient energy data set. 05 2016.

[18] Hyungsul Kim, Manish Marwah, Martin Arlitt, Geoff Lyon, and Jiawei Han. Unsupervised disaggregation of low frequency power measurements. In *Proceedings of the 2011 SIAM international conference on data mining*, pages 747–758. SIAM, 2011.

[19] Stephen William Makonin. *Real-time embedded low-frequency load disaggregation*. PhD thesis, Applied Sciences: School of Computing Science, 2014.

[20] A Matthews. Standardization of measures prior to cluster-analysis. In *Biometrics*, volume 35, pages 892–892. INTERNATIONAL BIOMETRIC SOC 808 17TH ST NW SUITE 200, WASHINGTON, DC 20006-3910, 1979.

[21] Roberto Medico, Leen De Baets, Jingkun Gao, Suman Giri, Emre Kara, Tom Dhaene, Chris Develder, Mario Bergés, and Dirk Deschrijver. Plaid 2018. https://doi.org/10.6084/m9.figshare.10084619.v2, 2020.

[22] Roberto Medico, Leen De Baets, Jingkun Gao, Suman Giri, Emre Kara, Tom Dhaene, Chris Develder, Mario Bergés, and Dirk Deschrijver. A voltage and current measurement dataset for plug load appliance identification in households. *Scientific data*, 7(1):1–10, 2020.

[23] Andrea missinato. Le reti neurali convoluzionali, ovvero come insegnare alle macchine a riconoscere per astrazione. https://www.spindox.it/it/blog/reti-neurali-convoluzionali-il-deep-learning-ispirato-alla-corteccia-visiva/, 2018.

[24] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of big data*, 2(1):1–21, 2015.

[25] Thomas Picon, Mohamed Nait Meziane, Philippe Ravier, Guy Lamarque, Clarisse Novello, Jean-Charles Le Bunetel, and Yves Raingeaud. Cooll: Controlled on/off loads library, a public dataset of high-sampled electrical signals for appliance identification. *arXiv preprint arXiv:1611.05803*, 2016.

[26] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[27] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[28] Antonio Ridi, Christophe Gisler, and Jean Hennebert. A survey on intrusive load monitoring for appliance recognition. In *2014 22nd international conference on pattern recognition*, pages 3702–3707. IEEE, 2014.

[29] Data Science Team. Introduzione al concetto di lstm. https://datascience.eu/it/apprendimento-automatico/comprensione-delle-reti-lstm/, 2020.