



UNIVERSITÀ POLITECNICA DELLE MARCHE

Facoltà di Ingegneria

Corso di laurea in Ingegneria Elettronica

Studio e sviluppo di algoritmi per la gestione energetica dei carichi energivori in edifici residenziali

Study and development of algorithms for the energy management of energy-intensive loads in residential buildings

Relatore: Prof. Andrea Monteriù

Candidato: Beleggia Alessio

Correlatore: Dott. Francesco Ferracuti

ANNO ACCADEMICO 2019 – 2020

*Ad Andrea, mio amico
...che riposi in pace.*

INDICE

1. Introduzione	5
1.1. Panoramica	5
1.2. Obiettivo	7
2. Reinforcement Learning	8
2.1. Processi decisionali di Markov (MDP)	8
2.2. Q-Learning	9
2.2.1. Politica ϵ -greedy: Esplorazione e Sfruttamento	11
2.2.2. Equazione di Bellman e aggiornamento del Q-value	12
3. Acquisizione e Rielaborazione dati	13
3.1. Acquisizione dati	13
3.1.1. Casa A	16
3.1.2. Casa B	17
3.1.3. Casa C	18
3.2. Datasets	18
3.2.1. Meteo Dataset	19
3.2.2. Dataset Meter 1	21
3.2.3. Dataset Meter 2	22
3.3. Rielaborazione Dati e Formattazione Datasets	24
3.3.1. Timetable Meteo	25
3.3.2. Timetable mis	28
3.3.3. Timetable en	32
4. Formulazione Problema	35
4.1. Tipologie di carico	35
4.1.1. Carichi non shiftabili e carichi controllabili	35
4.1.2. Carichi shiftabili	36
4.1.3. Funzione Obiettivo	37

5. Implementazione Algoritmo Q-learning	38
5.1. Panoramica	38
5.1.1. Inizializzazione Q-table	40
5.1.2. PUN	40
5.1.3. Working Time	41
5.1.4. Politica ϵ -greedy: Esplorazione e Sfruttamento	43
5.1.5. Memorizzazione Q^{i-1}	46
5.1.6. Calcolo Ricompensa	47
5.1.7. Calcolo nuovo Q-value	48
5.1.8. Verifica convergenza al Q-value ottimale	50
6. Risultati	52
7. Conclusioni	55
8. Bibliografia	56

1. INTRODUZIONE

1.1 Panoramica

L'emergenza climatica in atto ci mette di fronte alle nostre responsabilità nei confronti del pianeta. Gli edifici lavorativi e residenziali, dal punto di vista del consumo energetico, rappresentano una parte significativa dell'energia consumata dai paesi avanzati. Per tale motivo, il miglioramento dell'efficienza energetica negli edifici è oggi un obiettivo di primaria importanza dal punto di vista sia ambientale che economico.

L'ultimo decennio, salvo l'anno della pandemia in cui i sono registrati valori fuori dall'ordinario, ha inoltre fatto osservare in generale un costante e continuo aumento del prezzo dell'energia elettrica. Analizzando proprio il prezzo dell'energia nel nostro Paese si osserva come abbia un valore tra i più elevati in tutta Europa.

La volontà di trovare stili di vita e modalità di consumo con un impatto ambientale sempre minore va di pari passo con la ricerca di nuove strategie per la produzione di energia sostenibile e con lo sviluppo di metodologie di ottimizzazione dei consumi energetici. Tali metodologie, devono essere flessibili per soddisfare i requisiti di oggi e le esigenze dei prossimi decenni.

La ricerca di soluzioni ai problemi sopra predefiniti ha portato alla nascita di sistemi intelligenti come l'EMS (Energy Management System) e l'HEMS (Home Energy Management System), ovvero sistemi che consentono la gestione dell'energia tramite la raccolta, l'archiviazione e l'analisi di una grande quantità di dati inerenti ad abitazioni ed edifici residenziali. Tali dati, attraverso una adeguata e attenta elaborazione, vengono tradotti in sofisticate strategie di risparmio energetico e ottimizzazione del consumo.

Al momento, la maggior parte degli HEMS si basa ancora su regole deterministiche o modelli astratti come MILP (Mixed-Integer Linear Programming) che possono essere criticati in tre principali aspetti:

- le regole deterministiche per la gestione dei sistemi energetici non garantiscono l'ottimizzazione in quanto eventuali alterazioni nelle variabili potrebbero portare a perdite finanziarie;
- i modelli astratti sono generalmente solo stime di situazioni reali e, poiché la loro efficacia dipende dall'abilità dell'operatore, potrebbero essere irrealistici;
- le soluzioni di ottimizzazione basate sull'approccio MILP sono centralizzate e prive di scalabilità.

I continui cambiamenti nelle variabili rilevanti per il mercato elettrico come il prezzo dell'elettricità e il consumo dell'energia, richiedono che gli EMS intraprendano azioni ottimali in modo adattivo in tempo reale.

La rapida evoluzione dell'Intelligenza Artificiale (AI) nel corso degli ultimi anni ha portato tale disciplina informatica ad essere impiegata per la realizzazione di processi decisionali ottimali.

1.2 Obiettivo

Una possibile soluzione ai problemi già evidenziati è data dall'impiego dei metodi di Machine Learning e più in generale di Intelligenza Artificiale. A tal riguardo, in questo elaborato verrà impiegato il Reinforcement Learning (RL). In dettaglio, il Reinforcement Learning, verrà implementato tramite un algoritmo di Q-learning in un HEMS con lo scopo di ottimizzare il comportamento di un elettrodomestico shiftabile (una lavasciuga) trovando un adeguato compromesso tra esigenze contrastanti per l'utente quali il costo del consumo elettrico ed il costo di insoddisfazione dovuto al tempo di attesa.

Il Reinforcement Learning è costituito da un agente e da un ambiente che interagiscono in una successione di fasi temporali discrete nelle quali l'agente seleziona un'azione da inviare all'ambiente. Conseguentemente a ciò, l'agente riceve dall'ambiente una ricompensa, di valore elevato o meno, e l'ambiente muta in un nuovo stato.

A seconda delle ricompense ottenute a seguito delle azioni compiute, l'agente, dopo ripetuti episodi, riuscirà a capire quale azione in un dato stato sarà meglio eseguire riuscendo così ad intraprendere un processo decisionale ottimale.

Come verrà spiegato più dettagliatamente nel successivo capitolo il principale vantaggio dell'uso del Reinforcement Learning risiede nella capacità da parte dell'agente di agire senza conoscere preliminarmente l'ambiente nel quale andrà ad eseguire le azioni.

2. REINFORCEMENT LEARNING

2.1 Processi Decisionali di Markov (MDP)

Il Reinforcement Learning è un'area del Machine Learning il quale scopo è quello di capire in che modo un dato soggetto può interagire con un dato ambiente al fine di massimizzare un certo risultato.

Il soggetto che prende decisioni (denominate *azioni*) e impara da queste è noto come *agente*. Ciò con cui interagisce l'agente e tutto ciò che è al di fuori di esso è definito come *ambiente*.

Agente e ambiente interagiscono sequenzialmente nel tempo: l'agente osserva innanzitutto la rappresentazione della situazione in cui versa l'ambiente (definita *stato*) e conseguentemente seleziona una determinata azione da intraprendere. Ad ogni fase temporale l'ambiente, in risposta all'azione intrapresa dall'agente, transita in un nuovo stato e conseguentemente fornisce all'agente una *ricompensa*, ovvero un particolare valore numerico che l'agente vuole massimizzare e che misura quanto l'azione dell'agente sia stata "adeguata" per raggiungere l'obiettivo prefissato.

I Processi Decisionali di Markov (MDP) sono una formalizzazione del processo decisionale sequenziale sopra descritto e sono alla base della struttura dei problemi che vengono risolti tramite il Reinforcement Learning. I componenti coinvolti in un MDP sono i seguenti: agente, ambiente, stato, azione e ricompensa. Più dettagliatamente [1], in un MDP, si dispone di un insieme di stati \mathbf{S} , un insieme di azioni \mathbf{A} e un insieme di ricompense \mathbf{R} . Ad ogni istante di tempo $h=0,1,2,3\dots N$ l'agente riceve una rappresentazione dello stato dell'ambiente $S_h \in \mathbf{S}$ e, sulla base di questo stato, l'agente seleziona un'azione $A_h \in \mathbf{A}$: stato e azione, forniranno la cosiddetta coppia stato-azione (S_h, A_h) . All'istante successivo, come conseguenza della propria azione, l'agente riceve una ricompensa numerica $R \in \mathbf{R}$ e l'ambiente transita nello stato S_{h+1} .

L'intero processo sopra descritto viene illustrato sinteticamente nel seguente diagramma:

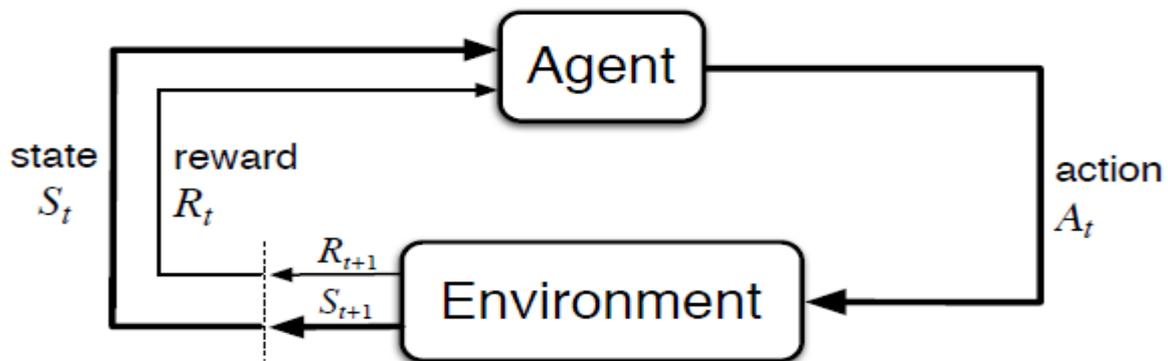


Figura 1: Schema Processo Decisionale di Markov (MDP)

2.2 Q-Learning

L'algoritmo di Reinforcement Learning che verrà adottato in questo elaborato prende il nome di Q-learning. L'obiettivo del Q-learning è quello di ottenere una *policy* ottimale ed, in particolare in questo elaborato, di ottenere tramite l'interazione dell'agente (*Q-Agent*) con l'ambiente (l'elettrodomestico), una sequenza di azioni operative che garantisca la massima ricompensa ottenibile per ciascuna coppia stato-azione in modo da ottimizzarne l'utilizzo.

Il Q-Agent raggiunge l'obiettivo preposto servendosi di una tabella denominata *Q-table*, la quale è una rappresentazione della memoria del Q-Agent che contiene i vari valori Q (*Q-value*) calcolati per ciascuna coppia stato-azione. Le colonne della tabella rappresentano le azioni, mentre le righe rappresentano gli stati.

I valori contenuti nella Q-table consentiranno al Q-Agent di selezionare in ogni stato l'azione migliore da eseguire.

Poiché il Q-Agent non ha una conoscenza preliminare dell'ambiente, inizialmente tutti i valori contenuti nella Q-Table saranno inizializzati a zero (*Figura 2*).

	AZIONI	
STATI	0	0
	0	0
	0	0
	0	0
	0	0
	0	0
	0	0
	0	0
	0	0
	0	0

Figura 2: Esempio di Q-table inizializzata a zero

Man a mano che, ad ogni episodio, la Q-table viene aggiornata, il Q-Agent potrà selezionare l'azione successiva da compiere sulla base del valore Q più elevato per lo stato corrente.

L'algoritmo del Q-Learning aggiorna iterativamente i valori di ciascuna coppia stato-azione della Q-table utilizzando una particolare equazione denominata *Equazione di Bellman* (che verrà illustrata e analizzata nel *paragrafo 2.2.2*). Quanto appena descritto avviene fino a quando i vari valori Q della tabella non convergono ai valori Q ottimali.

2.2.1 Politica ϵ -greedy: esplorazione e sfruttamento

In base a quale criterio il Q-Agent, soprattutto quando la Q-table contiene tutti (o quasi tutti) valori nulli, sceglierà un'azione piuttosto che un'altra? Al fine di rispondere a questa domanda è necessario introdurre il concetto di *politica* ϵ -greedy che fornisce all'agente due opzioni: quella dell'*esplorazione* dell'ambiente o quella dello *sfruttamento* dell'ambiente.

Se il Q-Agent opterà per l'*esplorazione* selezionerà un'azione dello stato corrente in maniera puramente randomica (*Figura 3*). Se il Q-agent opterà invece per lo *sfruttamento* utilizzerà le conoscenze relative all'ambiente già acquisite e selezionerà l'azione dello stato corrente avente il Q-value più elevato (*Figura 4*).

Di conseguenza, è semplice capire che esplorare solamente o sfruttare solamente non porterà a nessun risultato e, quindi, occorre ricercare un buon compromesso tra esplorazione e sfruttamento. Tale compromesso viene trovato per mezzo del *tasso di esplorazione* (exploration rate) ϵ (*paragrafo 5.1.4*).

Logica vuole che fintanto che non si hanno conoscenze sufficientemente approfondite dell'ambiente sia più conveniente esplorare e dunque far scegliere all'agente l'azione da intraprendere in maniera randomica.

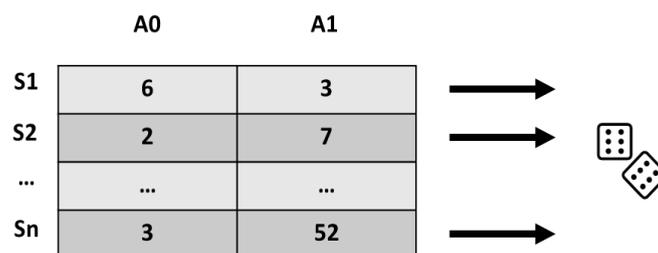


Figura 3: Scelta dell'azione tramite esplorazione

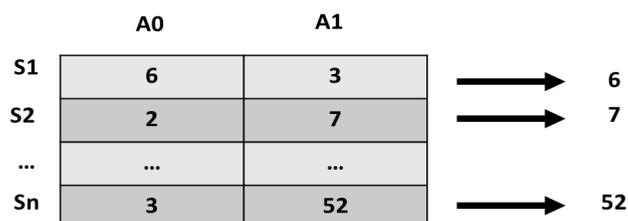


Figura 4: Scelta dell'azione tramite sfruttamento

2.2.2 Equazione di Bellman e aggiornamento del Q-value

Ad ogni azione il Q-value associato a quella determinata coppia stato-azione cambia, ovvero si aggiorna. Nell'aggiornamento del Q-value una parte fondamentale viene svolta dall'equazione di Bellman

$$Q_v^*(s_h, a_h) = r(s_h, a_h) + \gamma \cdot \max Q(s_{h+1}, a_{h+1}) \quad (1)$$

dove, per ottenere il cosiddetto *valore appreso* $Q_v^*(s_h, a_h)$ si effettua una somma che vede come primo addendo la *ricompensa* dovuta all'azione appena intrapresa e come secondo addendo il prodotto tra il cosiddetto *fattore di sconto* γ (paragrafo 5.1.7) e il Q-value di maggior valore dello stato successivo della Q-table (valore pari a 7 nella seguente tabella):

	34	53
S_t	3	5
S_{t+1}	7	2
	4	34

Figura 5: Scelta del valore massimo dello stato successivo da inserire nell'equazione di Bellman

Una volta ottenuto il valore appreso sarà possibile usufruirne per calcolare il Q-value aggiornato mediante l'utilizzo della seguente formula basata sull'equazione di Bellman:

$$Q(s_h, a_h) \leftarrow Q(s_h, a_h) + \vartheta \cdot [r(s_h, a_h) + \gamma \cdot \max Q(s_{h+1}, a_{h+1}) - Q(s_h, a_h)] \quad (2)$$

da cui utilizzando l'Eq. 1:

$$Q(s_h, a_h) \leftarrow Q(s_h, a_h) + \vartheta \cdot [Q_v^*(s_h, a_h) - Q(s_h, a_h)]$$

Tale processo verrà ripetuto iterativamente per un predeterminato numero di episodi o fino a quando il nuovo Q-value non convergerà al valore ottimale.

3. ACQUISIZIONE E RIELABORAZIONE DATI

3.1 Acquisizione Dati

Sviluppare un algoritmo Q-learning che si pone come obiettivo quello dell'ottimizzazione del consumo elettrico di un ambiente residenziale presuppone avere a disposizione dati elettrici affidabili e possibilmente reali su cui poter effettuare delle simulazioni.

A tal fine è stato preso in considerazione il contenuto dei dataset del progetto *Smart* Home Dataset* [2] il quale obiettivo era proprio quello di ottimizzare il consumo di diversi ambienti residenziali. La filosofia sposata dal suddetto progetto era basata sulla "profondità" dei dati raccolti, ovvero sulla raccolta di una vasta gamma di dati in pochi ambienti residenziali, anziché sulla "larghezza", ovvero sulla raccolta di una modesta quantità di dati in molto ambienti residenziali.

Di conseguenza, una prima ed importante parte del progetto *Smart* Home Dataset* comprendeva la realizzazione di una infrastruttura capace di raccogliere e memorizzare, nell'arco di tre anni, una vasta quantità di dati di tre case reali (che verranno denominate come Casa A, Casa B e Casa C) ubicate nel Massachusetts. La natura a lungo termine di questa raccolta dati fornisce un'ampia finestra temporale che permette di comprendere come il consumo di elettricità varia da stagione a stagione e di anno in anno.

La raccolta dati è stata realizzata mediante l'impiego di diversi tipi di sensori e rilevatori, i quali verranno descritti brevemente qui di seguito. L'impiego di misuratori di potenza ha permesso la rilevazione del consumo elettrico di ogni carico della casa. Nello specifico, la misurazione avviene per mezzo di un certo numero di trasduttori di corrente di cui è dotato ogni misuratore. Ogni trasduttore di corrente è in grado di registrare la potenza reale, la potenza apparente, la frequenza e la tensione per ogni secondo.

Generalmente i consumi rilevati vengono poi inviati al server principale del sistema via web. L'elevata precisione di cui vantano i misuratori di potenza permette loro di effettuare misure con un minimo errore relativo.

L'impiego di particolari interruttori a parete dimmerabili ha invece permesso il monitoraggio degli eventi di accensione/spegnimento/dim ogni qualvolta un interruttore venisse fisicamente premuto. Anche in questo caso le informazioni venivano inviate al server principale del sistema. Il livello dim viene riportato come percentuale tra 0 e 100, dove 100% significa piena illuminazione.

Viene anche monitorata a intervalli di 5 secondi la generazione di elettricità dovuta alla messa in uso di pannelli fotovoltaici e micro-turbine eoliche.

Inoltre, l'impiego di particolari termostati ha permesso il monitoraggio di riscaldamento e raffrescamento. Tali termostati inviavano informazioni al server principale ogni qualvolta che la temperatura di setpoint veniva modificata manualmente, quando la temperatura variava o quando il sistema veniva acceso o arrestato.

Specifici sensori di movimento sono invece stati impiegati al fine di assolvere al compito di monitorare l'occupazione delle stanze. Tali sensori inviavano informazioni al server principale nel momento in cui un'attività veniva rilevata in una stanza precedentemente inattiva o quando un'attività non veniva rilevata per due minuti consecutivi in una stanza precedentemente attiva.

Un altro tipo di sensore è stato installato al fine di monitorare quando le porte venivano aperte o chiuse registrando l'ora di apertura e chiusura.

Infine, l'installazione di una stazione meteo professionale è stata impiegata con lo scopo di monitorare le informazioni inerenti a dati meteorologici esterni. I dati rilevati venivano registrati con medie per ogni minuto.



Figura 6: Pannelli solari installati nella Casa C

Riassumendo, l'infrastruttura realizzata consente il monitoraggio delle seguenti grandezze:

- potenza media reale e apparente della casa e di ogni circuito per ogni secondo;
- consumo di potenza reale ogni pochi secondi di quasi tutti i cosiddetti plug-load;
- eventi accensione/spegnimento/dim degli interruttori a parete;
- generazione media di elettricità dai pannelli solari e dalle micro-turbine eoliche ogni 5 secondi;
- eventi collegati al consumo di energia quali rilevazioni di movimento e apertura/chiusura porte e rilevazioni di temperatura tramite termostati;
- dati ambientali ogni minuto tramite sensori meteorologici.

A seguire una breve descrizione dei tre edifici residenziali studiati in [2] alla fine della quale verrà segnalata la casa della quale si è deciso di analizzare i dataset.

3.1.1. Casa A

È un'abitazione di 158m² avente due piani occupata da 3 abitanti. La casa ha un totale di otto stanze incluso il seminterrato. Il piano terra comprende un soggiorno, una camera da letto, una cucina e un bagno mentre il primo piano comprende due camere da letto ed un altro bagno.

La casa non dispone di aria condizionata centralizzata (A/C) e gli occupanti, d'estate, utilizzano tre unità di A/C: una in soggiorno e una in ogni camera da letto del piano superiore. Le informazioni inerenti al riscaldamento/raffrescamento vengono inviate dai tre termostati al server principale del sistema.

I principali elettrodomestici di cui la casa dispone sono: un'asciugatrice, una lavatrice, una lavastoviglie, un frigorifero, un congelatore e un'unità di Ventilazione Meccanica Controllata VMC (in inglese Heat Recovery Ventilation HRV).

Nella casa A sono stati installati anche interruttori (descritti nella *Sezione 3.1*) che monitorano l'illuminazione (ovvero eventi on/off/dim), sensori che monitorano il consumo elettrico di tutti i circuiti per ogni secondo e sensori di movimento in tutte le otto stanze che la compongono.

Infine, la casa A dispone di una stazione meteo che registra i dati meteorologici.

3.1.2. Casa B

La casa B è occupata da quattro abitanti ed è simile alla casa A in quanto a dimensioni (due piani e approssimativamente 158m²). L'abitazione ha un totale di otto stanze: il piano terra comprende un salotto, una cucina e una sala pranzo mentre il primo piano comprende due camere da letto.

Diversamente dalla casa A, questa abitazione dispone di aria condizionata centralizzata e anche di un sistema di riscaldamento alimentato a gas. Il sistema di riscaldamento/raffrescamento utilizza un singolo termostato centralizzato.

Similarmente alla casa A sono stati installati dei sensori che consentono il monitoraggio del consumo di elettricità della casa e di quasi tutti i suoi circuiti per ogni secondo.

Infine, anche la casa B dispone di una stazione meteo che registra dati meteorologici.

3.1.3 Casa C

L'abitazione C è la più vasta delle tre (325m²) ed anch'essa è disposta su due piani. Similarmente alle due case sopra descritte anche la casa C presenta sensori che consentono il monitoraggio di consumo di elettricità della casa e dei suoi dispositivi per ogni secondo.

Una novità presente esclusivamente nella casa C e non nelle altre due abitazioni è l'installazione di tre pannelli solari dei quali viene monitorizzata l'elettricità generata ogni cinque secondi (mediati da campioni di ogni secondo).

La generazione utilizza i micro-inverter su ogni pannello per registrare la corrente individualmente, convertire la potenza in AC, sincronizzarla con la rete e immetterla nella rete di alimentazione domestica.

3.2 Datasets

Una volta acquisiti i dataset di ciascuna delle tre case si è passati al loro studio e alla loro analisi tramite Matlab al fine di comprendere quale casa avesse i dataset più adeguati per gli scopi e le finalità di questo studio.

Nonostante la Casa A fosse la più strumentalizzata, si è preferito analizzare i dataset della Casa C poiché erano gli unici che disponevano di dati inerenti alla generazione di elettricità tramite pannelli solari e perché tali dataset erano organizzati con una maggiore granularità: i dataset della Casa A e della Casa B non erano formattati al minuto per tutto il periodo di rilevamento differentemente dai dataset della Casa C. Dei tre anni disponibili, si è scelto di utilizzare il dataset più recente, ovvero quello del 2016.

Vi sono tre dataset riguardanti la Casa C: un dataset della stazione meteo inerente dunque ai dati meteorologici e due dataset di misuratori di potenza inerenti dunque ai valori di potenza dei vari carichi della casa (elettrodomestici, plug-load e generazione pannello solare).

3.2.1 Meteo Dataset

Il dataset (denominato “HomeC2016”) contiene i dati meteorologici con frequenza di un’ora che vanno dalle 6:00:00 del 1 Gennaio 2016 alle 5:00:00 del 31 Dicembre 2016; l’unità di tempo nel suddetto dataset è data dal UNIX Timestamp.

Il dataset, in Matlab, è rappresentato da una timetable di dimensione 8760x14 contenente i seguenti dati:

- Temperatura esterna;
- Icon: dato che conteneva scritta la condizione meteorologica e il momento della giornata (ad esempio: clear day, partly-cloudy night, rain, snow, ecc.);
- Umidità;
- Visibilità;
- Summary: dato che conteneva scritta sommariamente la condizione meteorologica senza specificare il momento della giornata (ad esempio: clear, partly-cloudy, rain, snow ecc.);
- Temperatura percepita;
- Pressione;
- Velocità del vento;
- Nuvolosità;
- Data e ora;
- Rilevamento vento;
- Intensità di precipitazione;
- Temperatura di rugiada;
- Probabilità di precipitazione.

Verranno allegate qui di seguito due figure (*Figura 7* e *Figura 8*) che mostrano solo le prime 20 ore della timetable (per ovvi limiti di spazio non è possibile mostrare tutti i giorni utilizzati per il rilevamento, ma è possibile quantomeno mostrare tutti i tipi di dato) al fine di dare un’idea al lettore di come il dataset è organizzato.

	1	2	3	4	5	6	7
	temperature	icon	humidity	visibility	summary	apparentTemperature	pressure
1	36.1400	clear-night	0.6200	10	Clear	29.2600	1.0169e+03
2	35.8700	partly-clou...	0.6100	10	Mostly Clou...	29.4000	1.0163e+03
3	35.4000	clear-night	0.6100	10	Clear	28.8700	1.0160e+03
4	35.2500	clear-night	0.6400	10	Clear	30.5100	1.0161e+03
5	35.4900	clear-night	0.6400	10	Clear	29.6500	1.0159e+03
6	35.2600	clear-night	0.6500	10	Clear	29.7900	1.0156e+03
7	34.9900	clear-night	0.6600	10	Clear	29.4100	1.0155e+03
8	35.0600	partly-clou...	0.6600	10	Mostly Clou...	30.0500	1.0158e+03
9	34.8300	clear-day	0.6800	10	Clear	29.6100	1.0156e+03
10	35.7000	clear-day	0.6500	10	Clear	30.9700	1.0157e+03
11	36.9200	clear-day	0.6200	10	Clear	32.7400	1.0157e+03
12	36.8700	clear-day	0.6200	10	Clear	31.1400	1.0152e+03
13	37.3100	clear-day	0.5900	10	Clear	30.5000	1.0142e+03
14	37.1400	cloudy	0.5900	10	Overcast	30.3600	1.0133e+03
15	37.1900	clear-day	0.5800	10	Clear	31.3800	1.0127e+03
16	36.5700	clear-day	0.6000	9.0700	Clear	30.9500	1.0124e+03
17	35.9100	clear-day	0.6100	9.9000	Clear	29.0900	1.0124e+03
18	34.8900	partly-clou...	0.6500	9.0500	Partly Cloudy	28.4200	1.0126e+03
19	33.8700	partly-clou...	0.6800	8.5200	Partly Cloudy	27.3800	1.0130e+03
20	32.8700	partly-clou...	0.7000	8.0600	Partly Cloudy	25.2500	1.0136e+03

Figura 7: Dataset Meteo (dal primo al settimo array)

	8	9	10	11	12	13	14
	windSpeed	cloudCover	time	windBearing	preciplIntensity	dewPoint	precipProbability
1	9.1800	NaN	1.4516e+09	282	0	24.4000	0
2	8.2900	0.7500	1.4516e+09	285	0	23.9000	0
3	8.2000	NaN	1.4516e+09	281	0	23.3900	0
4	5.4700	NaN	1.4516e+09	265	0	24.1800	0
5	7.0800	NaN	1.4516e+09	265	0	24.3500	0
6	6.4500	0	1.4516e+09	268	0	24.7900	0
7	6.5300	0	1.4516e+09	260	0	24.9200	0
8	5.7700	0.7500	1.4516e+09	259	0	24.8400	0
9	5.9900	NaN	1.4517e+09	258	0	25.2700	0
10	5.5500	NaN	1.4517e+09	255	0	25.2600	0
11	5.1400	NaN	1.4517e+09	238	0	25.0800	0
12	7.3700	NaN	1.4517e+09	239	0	25.2600	0
13	9.5900	NaN	1.4517e+09	272	0	24.3000	0
14	9.4500	1	1.4517e+09	273	0	24.0600	0
15	7.6100	NaN	1.4517e+09	268	0	23.8100	0
16	7.0900	NaN	1.4517e+09	260	0	23.8700	0
17	8.9500	NaN	1.4517e+09	256	0	23.7800	0
18	7.9100	0.3100	1.4517e+09	255	0.0011	24.5200	0.0200
19	7.5800	0.3100	1.4517e+09	265	0.0011	24.3900	0.0200
20	9.1400	0.4400	1.4517e+09	273	0	24.1000	0

Figura 8: Dataset Meteo (dall'ottavo al tredicesimo array)

3.2.2 Dataset Meter 1

Il dataset denominato “HomeC-meter1_2016” contiene i dati del primo misuratore di potenza della Casa C con frequenza di 1 minuto che iniziano dalle 00:00:00 del 1 Gennaio 2016 per arrivare fino alle 22:29:00 del 15 Dicembre 2016.

Il dataset, in Matlab, è rappresentato da una timetable di dimensione 503910x19. La *Figura 9* e la *Figura 10* allegate qui di seguito mostrano solo i primi 20 minuti della timetable (per ovvi limiti di spazio non è possibile mostrare tutti i giorni utilizzati per il rilevamento, ma è possibile quantomeno mostrare tutti i tipi di dato) al fine di dare un’idea al lettore di come il dataset è organizzato.

	1	2	3	4	5	6	7	8	9
	DateTime	usekW	genkW	HouseOverallkW	DishwasherkW	Furnace1kW	Furnace2kW	HomeOfficekW	FridgekW
1	2016-01-01 00:00:00	0.9328	0.0035	0.9328	3.3333e-05	0.0207	0.0619	0.4426	0.1242
2	2016-01-01 00:01:00	0.9343	0.0035	0.9343	0	0.0207	0.0638	0.4441	0.1240
3	2016-01-01 00:02:00	0.9318	0.0035	0.9318	1.6667e-05	0.0207	0.0623	0.4461	0.1235
4	2016-01-01 00:03:00	1.0221	0.0035	1.0221	1.6667e-05	0.1069	0.0685	0.4466	0.1231
5	2016-01-01 00:04:00	1.1394	0.0035	1.1394	1.3333e-04	0.2369	0.0640	0.4465	0.1229
6	2016-01-01 00:05:00	1.3919	0.0034	1.3919	2.8333e-04	0.5033	0.0637	0.4470	0.1223
7	2016-01-01 00:06:00	1.3662	0.0035	1.3662	2.8333e-04	0.4994	0.0637	0.4433	0.1221
8	2016-01-01 00:07:00	1.4319	0.0034	1.4319	2.5000e-04	0.4779	0.1786	0.4443	0.1218
9	2016-01-01 00:08:00	1.6273	0.0034	1.6273	1.8333e-04	0.4477	0.3657	0.4415	0.1216
10	2016-01-01 00:09:00	1.7354	0.0034	1.7354	1.6667e-05	0.1716	0.6825	0.4387	0.1216
11	2016-01-01 00:10:00	1.5851	0.0034	1.5851	5.0000e-05	0.0221	0.6787	0.4402	0.1215
12	2016-01-01 00:11:00	1.5103	0.0034	1.5103	3.3333e-05	0.0220	0.6207	0.4370	0.1213
13	2016-01-01 00:12:00	1.4599	0.0035	1.4599	5.0000e-05	0.0219	0.5775	0.4400	0.1210
14	2016-01-01 00:13:00	0.8406	0.0034	0.8406	0	0.0210	0.1448	0.4448	0.0350
15	2016-01-01 00:14:00	0.7032	0.0034	0.7032	1.6667e-05	0.0207	0.0620	0.4438	0.0048
16	2016-01-01 00:15:00	0.5719	0.0035	0.5719	0	0.0207	0.0637	0.3078	0.0049
17	2016-01-01 00:16:00	0.4857	0.0035	0.4857	1.6667e-05	0.0206	0.0634	0.2205	0.0050
18	2016-01-01 00:17:00	0.5232	0.0034	0.5232	0	0.0206	0.0621	0.2601	0.0050
19	2016-01-01 00:18:00	0.5362	0.0035	0.5362	0	0.0207	0.0629	0.2721	0.0050
20	2016-01-01 00:19:00	0.5342	0.0035	0.5342	1.6667e-05	0.0207	0.0627	0.2701	0.0050

Figura 9: Dataset Meter1 (dal primo al nono array)

	10	11	12	13	14	15	16	17	18	19
	WineCellarkW	GarageDoorkW	Kitchen12kW	Kitchen14kW	Kitchen38kW	BarnkW	WellkW	MicrowavekW	LivingRoomkW	SolarkW
1	0.0070	0.0131	4.1667e-04	1.5000e-04	0	0.0314	0.0010	0.0041	0.0015	0.0035
2	0.0070	0.0131	4.1667e-04	1.5000e-04	0	0.0315	0.0010	0.0041	0.0017	0.0035
3	0.0070	0.0131	4.3333e-04	1.6667e-04	1.6667e-05	0.0315	1.0000e-03	0.0041	0.0017	0.0035
4	0.0070	0.0130	4.3333e-04	2.1667e-04	0	0.0315	0.0010	0.0041	0.0016	0.0035
5	0.0069	0.0128	4.5000e-04	3.3333e-04	0	0.0315	0.0010	0.0041	0.0016	0.0035
6	0.0067	0.0124	4.8333e-04	5.6667e-04	0	0.0315	0.0010	0.0041	0.0016	0.0034
7	0.0067	0.0124	5.1667e-04	5.5000e-04	0	0.0316	0.0010	0.0041	0.0015	0.0035
8	0.0068	0.0126	4.8333e-04	4.5000e-04	0	0.0317	0.0010	0.0042	0.0016	0.0034
9	0.0070	0.0127	4.6667e-04	3.0000e-04	1.6667e-05	0.0318	0.0010	0.0042	0.0016	0.0034
10	0.0072	0.0134	3.6667e-04	5.0000e-05	0	0.0317	0.0010	0.0042	0.0016	0.0034
11	0.0074	0.0136	3.5000e-04	1.1667e-04	3.3333e-05	0.0317	1.0000e-03	0.0042	0.0016	0.0034
12	0.0073	0.0135	3.3333e-04	1.0000e-04	0	0.0318	1.0000e-03	0.0042	0.0016	0.0034
13	0.0072	0.0135	3.6667e-04	8.3333e-05	1.6667e-05	0.0318	1.0000e-03	0.0042	0.0016	0.0035
14	0.0070	0.0132	6.5000e-04	1.8333e-04	1.6667e-05	0.0318	0.0010	0.0042	0.0016	0.0034
15	0.0070	0.0131	7.3333e-04	2.3333e-04	0	0.0318	0.0010	0.0042	0.0016	0.0034
16	0.0071	0.0131	7.3333e-04	1.5000e-04	0	0.0317	1.0000e-03	0.0042	0.0016	0.0035
17	0.0070	0.0131	7.5000e-04	8.3333e-05	0	0.0318	1.0000e-03	0.0042	0.0016	0.0035
18	0.0070	0.0131	7.3333e-04	1.0000e-04	1.6667e-05	0.0319	0.0010	0.0043	0.0017	0.0034
19	0.0070	0.0132	7.3333e-04	1.1667e-04	0	0.0319	0.0010	0.0042	0.0017	0.0035
20	0.0071	0.0132	7.3333e-04	1.0000e-04	0	0.0319	1.0000e-03	0.0042	0.0018	0.0035

Figura 10: Dataset Meter 2 (dal decimo al diciannovesimo array)

3.2.3 Dataset Meter 2

Il dataset denominato “HomeC-meter2_2016” contiene i dati del secondo misuratore di potenza impiegato nella Casa C con frequenza di 1 minuto che iniziano dalle 00:00:00 del 1 Gennaio 2016 per arrivare fino alle 22:29:00 del 15 Dicembre 2016.

Il dataset, in Matlab, è rappresentato da una timetable di dimensione 503910x9. Così come per il dataset precedente verrà allegata qui di seguito la *Figura 11* che mostra solo i primi 20 minuti della timetable al fine di dare un’idea al lettore di come il dataset è organizzato:



	1	2	3	4	5	6	7	8	9
	DateTime	usekW	genkW	LivingRoomOutletskW	BarnkW	WellkW	MicrowavekW	WasherDrierkW	SolarkW
1	2016-01-01 00:00:00	0	0.0045	9.1667e-04	0.0285	1.6667e-05	0.0031	3.3333e-05	0.0045
2	2016-01-01 00:01:00	0	0.0045	9.3333e-04	0.0285	1.6667e-05	0.0031	1.6667e-05	0.0045
3	2016-01-01 00:02:00	0	0.0045	9.1667e-04	0.0285	0	0.0031	3.3333e-05	0.0045
4	2016-01-01 00:03:00	0	0.0045	9.3333e-04	0.0285	0	0.0031	1.6667e-05	0.0045
5	2016-01-01 00:04:00	0	0.0045	9.0000e-04	0.0285	0	0.0031	0	0.0045
6	2016-01-01 00:05:00	0	0.0045	9.3333e-04	0.0285	0	0.0031	5.0000e-05	0.0045
7	2016-01-01 00:06:00	0	0.0045	9.1667e-04	0.0286	1.6667e-05	0.0031	3.3333e-05	0.0045
8	2016-01-01 00:07:00	0	0.0044	9.3333e-04	0.0287	3.3333e-05	0.0032	0	0.0044
9	2016-01-01 00:08:00	0	0.0044	9.1667e-04	0.0288	3.3333e-05	0.0032	6.6667e-05	0.0044
10	2016-01-01 00:09:00	0	0.0044	9.3333e-04	0.0287	1.6667e-05	0.0032	1.1667e-04	0.0044
11	2016-01-01 00:10:00	0	0.0044	9.1667e-04	0.0287	3.3333e-05	0.0032	1.6667e-04	0.0044
12	2016-01-01 00:11:00	0	0.0044	9.1667e-04	0.0288	1.6667e-05	0.0032	1.3333e-04	0.0044
13	2016-01-01 00:12:00	0	0.0044	9.3333e-04	0.0288	3.3333e-05	0.0032	1.3333e-04	0.0044
14	2016-01-01 00:13:00	0	0.0045	9.3333e-04	0.0288	5.0000e-05	0.0032	1.0000e-04	0.0045
15	2016-01-01 00:14:00	0	0.0044	9.3333e-04	0.0288	0	0.0032	8.3333e-05	0.0044
16	2016-01-01 00:15:00	0	0.0045	9.3333e-04	0.0288	5.0000e-05	0.0032	1.1667e-04	0.0045
17	2016-01-01 00:16:00	0	0.0044	9.3333e-04	0.0289	1.6667e-05	0.0033	1.1667e-04	0.0044
18	2016-01-01 00:17:00	0	0.0045	9.3333e-04	0.0288	1.6667e-05	0.0032	1.1667e-04	0.0045
19	2016-01-01 00:18:00	0	0.0045	9.3333e-04	0.0289	3.3333e-05	0.0032	1.1667e-04	0.0045
20	2016-01-01 00:19:00	0	0.0045	9.5000e-04	0.0289	1.6667e-05	0.0032	1.1667e-04	0.0045

Figura 11: Dataset Meter2

3.3 Rielaborazione dati e formattazione datasets

Una volta acquisiti i tre dataset sopradescritti è stato necessario rielaborare i dati in essi contenuti al fine di adattarli alle necessità di questo studio.

Per quanto concerne alla frequenza dei dati si è deciso di mantenere la frequenza di un'ora per il dataset meteo e di riformattare invece i due dataset inerenti al consumo in potenza con una frequenza di 15 minuti.

Per quanto concerne invece alla data di inizio e di fine si è deciso di riformattare i dataset in modo da avere una data e un'ora di inizio e di fine comune a tutti e tre i dataset e tutti i giorni completi delle 24 ore.

Al fine di rispettare le condizioni sopra scritte e ricordando, come descritto già nei paragrafi precedenti, che:

- il dataset inerente ai dati meteorologici iniziava dalle 6:00:00 del 1 Gennaio 2016 e terminava alle 5:00:00 del 31 Dicembre 2016;
- i dataset inerenti al consumo di potenza iniziavano dalle 00:00:00 del 1 Gennaio 2016 e terminavano alle 22:29:00 del 15 Dicembre 2016;

si è deciso di adottare come data e ora di inizio e fine rispettivamente il 2 Gennaio 2016 alle 00:00:00 e il 14 Dicembre 2016 alle 23:00:00.

Infine, si è deciso anche di creare una nuova timetable contenente il consumo di energia (in kilowattora) di tutti i carichi calcolati in base ai dati contenuti nei due dataset inerenti al consumo di potenza.

3.3.1 Timetable *meteo*

Le operazioni che verranno ora descritte porteranno alla definizione di una timetable composta da 13 array di dimensione 8352x1 elencati nella *Figura 12*:

	meteo	8352x13 timetable	8352x13
	meteo_apparentTemperature_C...	8352x1 double	8352x1
	meteo_cloudCover	8352x1 double	8352x1
	meteo_dewPoint	8352x1 double	8352x1
	meteo_humidity	8352x1 double	8352x1
	meteo_icon	8352x1 categorical	8352x1
	meteo_precipIntensity	8352x1 double	8352x1
	meteo_precipProbability	8352x1 double	8352x1
	meteo_pressure	8352x1 double	8352x1
	meteo_summary	8352x1 categorical	8352x1
	meteo_temperature_Celsius	8352x1 double	8352x1
	meteo_time	8352x1 datetime	8352x1
	meteo_visibility	8352x1 double	8352x1
	meteo_windBearing	8352x1 double	8352x1
	meteo_windSpeed	8352x1 double	8352x1

Figura 12: Array contenuti nella Timetable Meteo

Innanzitutto, sono state eliminate, all'inizio e alla fine di ogni array, le righe in eccesso in modo che la nuova timetable risulti iniziare e terminare nelle date indicate nella sezione precedente.

Al fine di avere valori definiti e accettabili, i valori NaN (Not a Number) sono stati impostati come valori nulli mediante l'utilizzo della funzione *isnan*. Il comando è stato eseguito per tutti gli array della timetable ed è descritto dalla seguente sintassi:

```
array_generico(isnan(array_generico))=0
```

Nello specifico, verrà riportato qui di seguito come esempio di validità generale il comando eseguito solo per un array:

```
cloudCover(isnan(cloudCover))=0;
```

Gli array inerenti alla temperatura (*temperature* e *apparentTemperature*) sono stati convertiti da gradi Fahrenheit a gradi Celsius utilizzando il seguente comando:

```
meteo_apparentTemperature_Celsius=((apparentTemperature-32)*(5/9));
meteo_temperature_Celsius=((temperature-32)*(5/9));
```

Le semplici operazioni sopra descritte avranno come risultato la timetable illustrata nella *Figura 13* e *Figura 14*:

	meteo_time	1 meteo_apparentTemperature_Celsius	2 meteo_cloudCover	3 meteo_dewPoint	4 meteo_humidity	5 meteo_icon	6 meteo_precipIntensity
1	2016-01-02 00:00:00	-3.7500	0.4400	24.1000	0.7000	partly-cloudy-...	0
2	2016-01-02 01:00:00	-3.8222	0.1300	22.7900	0.6800	clear-night	0
3	2016-01-02 02:00:00	-4.1444	0.1900	20.2100	0.6300	clear-night	0
4	2016-01-02 03:00:00	-4.3167	0.2500	18.9600	0.6100	partly-cloudy-...	0
5	2016-01-02 04:00:00	-5.0778	0.1600	18.3300	0.6000	clear-night	0
6	2016-01-02 05:00:00	-5.3111	0.2100	17.8200	0.6000	clear-night	0
7	2016-01-02 06:00:00	-5.6389	0.1900	17.4400	0.6000	clear-night	0
8	2016-01-02 07:00:00	-6.1333	0.1500	17.6400	0.6200	clear-night	0
9	2016-01-02 08:00:00	-5.9000	0.1400	18.1800	0.6300	clear-night	0
10	2016-01-02 09:00:00	-5.3389	0.3100	18.6000	0.6300	partly-cloudy-...	0
11	2016-01-02 10:00:00	-5.7556	0.2700	18.5700	0.6400	partly-cloudy-...	0
12	2016-01-02 11:00:00	-5.4556	0.2800	18.1700	0.6400	partly-cloudy-...	0
13	2016-01-02 12:00:00	-5.7222	0.2800	17.6200	0.6300	partly-cloudy-...	0
14	2016-01-02 13:00:00	-6.1889	0.1700	17.2700	0.6200	clear-day	0
15	2016-01-02 14:00:00	-4.7833	0.0500	17.0700	0.5600	clear-day	0
16	2016-01-02 15:00:00	-4.3222	0.1000	17.0800	0.5300	clear-day	0
17	2016-01-02 16:00:00	-3.5444	0.2600	17.2700	0.5100	partly-cloudy-...	0
18	2016-01-02 17:00:00	-2.5500	0.1600	17.2500	0.4800	clear-day	0
19	2016-01-02 18:00:00	-2.5056	0.2900	17.4000	0.4700	partly-cloudy-...	0
20	2016-01-02 19:00:00	-1.9167	0.1500	17.6000	0.4700	clear-day	0

Figura 13: Timetable Meteo (dal primo al sesto array)

	7 meteo_precipProbability	8 meteo_pressure	9 meteo_summary	10 meteo_temperature_Celsius	11 meteo_visibility	12 meteo_windBearing	13 meteo_windSpeed
1	0	1.0136e+03	Partly Cloudy	0.4833	8.0600	273	9.1400
2	0	1.0138e+03	Clear	0.0944	9.8000	278	7.8900
3	0	1.0139e+03	Clear	-0.3167	10	275	7.4000
4	0	1.0139e+03	Partly Cloudy	-0.5667	10	266	7.0600
5	0	1.0140e+03	Clear	-0.8389	10	269	8.2300
6	0	1.0142e+03	Clear	-1.0889	10	275	8.0200
7	0	1.0142e+03	Clear	-1.3556	10	273	8.0400
8	0	1.0143e+03	Clear	-1.7556	10	270	8.0400
9	0	1.0144e+03	Clear	-1.5833	9.8100	267	8
10	0	1.0143e+03	Partly Cloudy	-1.3611	10	260	7.2200
11	0	1.0144e+03	Partly Cloudy	-1.6111	10	249	7.5100
12	0	1.0148e+03	Partly Cloudy	-1.7111	9.9100	254	6.4700
13	0	1.0153e+03	Partly Cloudy	-1.8500	10	254	6.7100
14	0	1.0157e+03	Clear	-1.8611	10	253	7.8600
15	0	1.0161e+03	Clear	-0.6722	10	254	7.9700
16	0	1.0164e+03	Clear	0.0722	10	258	9.3700
17	0	1.0159e+03	Partly Cloudy	0.8500	10	262	10.0100
18	0	1.0149e+03	Clear	1.7278	10	279	10.3500
19	0	1.0143e+03	Partly Cloudy	1.9056	10	270	11.0700
20	0	1.0137e+03	Clear	2.1444	10	280	9.9200

Figura 14: Timetable Meteo (dal settimo al tredicesimo array)

3.3.2 Timetable *mis*

I comandi che verranno esposti qui di seguito andranno a definire una serie di array che avranno come prefissi *mis1* o *mis2*.

Gli array aventi come prefisso *mis1* si riferiscono ai dati del dataset “HomeC-meter1_2016” mentre gli array aventi come prefisso *mis2* si riferiscono ai dati del dataset “HomeC-meter2_2016”. Il tutto porterà alla definizione di una timetable composta da 26 array di dimensione 33408x1 elencati nella *Figura 15*:

	mis	33408x26 timetable	33408x26
	mis1_BarnkW	33408x1 double	33408x1
	mis1_DateTime	33408x1 datetime	33408x1
	mis1_DishwasherkW	33408x1 double	33408x1
	mis1_FridgekW	33408x1 double	33408x1
	mis1_Furnace1kW	33408x1 double	33408x1
	mis1_Furnace2kW	33408x1 double	33408x1
	mis1_GarageDoorkW	33408x1 double	33408x1
	mis1_genkW	33408x1 double	33408x1
	mis1_HomeOfficekW	33408x1 double	33408x1
	mis1_HouseOverallkW	33408x1 double	33408x1
	mis1_Kitchen12kW	33408x1 double	33408x1
	mis1_Kitchen14kW	33408x1 double	33408x1
	mis1_Kitchen38kW	33408x1 double	33408x1
	mis1_LivingRoomkW	33408x1 double	33408x1
	mis1_MicrowavekW	33408x1 double	33408x1
	mis1_SolarkW	33408x1 double	33408x1
	mis1_usekW	33408x1 double	33408x1
	mis1_WellkW	33408x1 double	33408x1
	mis1_WineCellar kW	33408x1 double	33408x1
	mis2_BarnkW	33408x1 double	33408x1
	mis2_genkW	33408x1 double	33408x1
	mis2_LivingRoomOutlets kW	33408x1 double	33408x1
	mis2_Microwave kW	33408x1 double	33408x1
	mis2_SolarkW	33408x1 double	33408x1
	mis2_usekW	33408x1 double	33408x1
	mis2_WasherDrier kW	33408x1 double	33408x1
	mis2_WellkW	33408x1 double	33408x1

Figura 15: Array contenuti nella Timetable Mis

Così come per la timetable *meteo* sono state innanzitutto eliminate, all'inizio ed alla fine di ogni array, le righe in eccesso in modo che la nuova timetable risulti iniziare e terminare nelle date e negli orari indicate nel *paragrafo 3.3*.

Come già anticipato nel *paragrafo 3.3* si è stabilito a priori di modificare la frequenza di questa timetable a 15 minuti. Avendo già a disposizione i dati con frequenza di 1 minuto (*Figura 9, Figura 10 e Figura 11*) la scelta più naturale è stata quella di eseguire una media ogni 15 valori per tutto il dataset. Tale scelta è stata messa in atto per tutti gli array del dataset eseguendo il comando descritto dalla seguente sintassi:

```
n=15; b=array_campionato; a=array_da_campionare;  
b=arrayfun(@(i) mean(a(i:i+n-1)),1:n:length(a)-n+1)';
```

Verrà riportato qui di seguito come esempio di validità generale il comando eseguito solo per un array:

```
n=15;  
mis1_DishwasherKW=arrayfun(@(i) mean(DishwasherKW(i:i+n-1)),1:n:length(DishwasherKW)-n+1)';
```

dove la funzione:

- *arrayfun* applica una funzione (in questo caso la funzione *mean* ad ogni elemento dell'array;
- *mean* esegue la media aritmetica ad intervalli di 15 valori;
- *length* misura la dimensione dell'array.

Conseguentemente al comando sopra indicato i nuovi array campionati al quarto d'ora avranno una dimensione di 33408x1 diversamente dagli array originali aventi una dimensione di 501120x1.

Le operazioni sopra descritte porteranno alla definizione della nuova timetable (denominata *mis*) illustrata nelle Figure 16, 17 e 18:

	1	2	3	4	5	6	7	8
mis1_DateTime	mis1_usekW	mis2_usekW	mis1_HouseOverallkW	mis1_genkW	mis2_genkW	mis1_BarnkW	mis1_DishwasherkW	mis1_FridgekW
2016-01-02 00:00:00	1.4497	0	1.4497	0.0034	0.0044	0.0346	0.1238	0.1350
2016-01-02 00:15:00	1.2909	0	1.2909	0.0034	0.0044	0.0348	0.2714	0.1651
2016-01-02 00:30:00	0.8163	0	0.8163	0.0034	0.0044	0.0348	0.0883	0.0129
2016-01-02 00:45:00	1.8407	0	1.8407	0.0034	0.0044	0.0346	1.2126	0.0044
2016-01-02 01:00:00	1.1538	0	1.1538	0.0034	0.0044	0.0347	0.4220	0.0049
2016-01-02 01:15:00	0.7037	0	0.7037	0.0034	0.0044	0.0348	0.0028	0.0051
2016-01-02 01:30:00	0.5125	0	0.5125	0.0034	0.0044	0.0347	1.7778e-04	0.0050
2016-01-02 01:45:00	0.7087	0	0.7087	0.0035	0.0045	0.0345	2.3778e-04	0.0051
2016-01-02 02:00:00	0.7021	0	0.7021	0.0034	0.0044	0.0348	3.0667e-04	0.0743
2016-01-02 02:15:00	0.7840	0	0.7840	0.0034	0.0044	0.0451	2.9778e-04	0.1368
2016-01-02 02:30:00	0.8543	0	0.8543	0.0034	0.0044	0.0348	3.0556e-04	0.1671
2016-01-02 02:45:00	0.7958	0	0.7958	0.0034	0.0044	0.0346	2.7778e-04	0.1205
2016-01-02 03:00:00	1.1499	0	1.1499	0.0034	0.0044	0.0344	1.1444e-04	0.0049
2016-01-02 03:15:00	1.0232	0	1.0232	0.0034	0.0044	0.0344	1.0444e-04	0.0049
2016-01-02 03:30:00	1.0172	0	1.0172	0.0034	0.0044	0.0343	1.0778e-04	0.0049
2016-01-02 03:45:00	0.7876	0	0.7876	0.0034	0.0044	0.0343	2.4778e-04	0.0051
2016-01-02 04:00:00	0.7008	0	0.7008	0.0034	0.0044	0.0343	2.0444e-04	0.0050
2016-01-02 04:15:00	0.7618	0	0.7618	0.0034	0.0044	0.0344	2.6111e-04	0.1096
2016-01-02 04:30:00	0.8257	0	0.8257	0.0034	0.0044	0.0344	2.3778e-04	0.1313
2016-01-02 04:45:00	0.8259	0	0.8259	0.0034	0.0044	0.0343	2.4222e-04	0.1223

Figura 16: Timetable Mis (dal primo all'ottavo array)

	9	10	11	12	13	14	15	16	17
mis1_Furnace1kW	mis1_Furnace2kW	mis1_GarageDoorkW	mis1_HomeOfficekW	mis1_Kitchen12kW	mis1_Kitchen14kW	mis1_Kitchen38kW	mis1_LivingRoomkW	mis1_MicrowavekW	
0.1159	0.6746	0.0133	0.0506	3.1889e-04	2.8333e-04	1.7778e-05	0.0016	0.0042	
0.0871	0.4265	0.0131	0.0508	3.2111e-04	2.0444e-04	1.0000e-05	0.0016	0.0042	
0.0848	0.3432	0.0132	0.0509	6.8333e-04	1.0111e-04	1.2222e-05	0.0017	0.0043	
0.0598	0.3197	0.0120	0.0514	8.3333e-04	8.9889e-04	1.1111e-05	0.0016	0.0042	
0.1389	0.3222	0.0128	0.0511	7.5333e-04	3.8000e-04	1.1111e-05	0.0016	0.0043	
0.1934	0.2417	0.0131	0.0511	7.3111e-04	2.8889e-05	7.7779e-06	0.0016	0.0043	
0.2138	0.0692	0.0129	0.0512	7.5444e-04	1.4111e-04	1.2222e-05	0.0016	0.0043	
0.1574	0.2224	0.0131	0.0509	6.9889e-04	1.6444e-04	1.1111e-05	0.0017	0.0041	
0.0577	0.2830	0.0134	0.0509	5.3333e-04	1.6444e-04	1.0000e-05	0.0016	0.0043	
0.0700	0.2697	0.0133	0.0507	3.6778e-04	1.6667e-04	5.5557e-06	0.0016	0.0043	
0.0705	0.3044	0.0134	0.0508	3.1000e-04	2.2444e-04	1.0000e-05	0.0017	0.0043	
0.1159	0.2673	0.0132	0.0507	4.1222e-04	1.7556e-04	1.2222e-05	0.0016	0.0043	
0.4908	0.3326	0.0126	0.0644	7.3889e-04	2.1000e-04	1.0000e-05	0.0016	0.0042	
0.4848	0.2727	0.0127	0.0543	7.4222e-04	2.1556e-04	8.8891e-06	0.0016	0.0042	
0.4800	0.2748	0.0127	0.0511	7.5222e-04	2.0778e-04	8.8891e-06	0.0016	0.0042	
0.1886	0.2742	0.0131	0.0509	6.9333e-04	5.2222e-05	6.6667e-06	0.0016	0.0041	
0.2380	0.1972	0.0129	0.0510	7.2556e-04	1.4556e-04	6.6668e-06	0.0016	0.0042	
0.1422	0.2293	0.0131	0.0508	4.4889e-04	1.5444e-04	1.1111e-05	0.0016	0.0042	
0.1880	0.2217	0.0130	0.0521	4.0000e-04	1.9333e-04	1.2222e-05	0.0016	0.0042	
0.1815	0.2362	0.0130	0.0518	4.1778e-04	1.2111e-04	1.1111e-05	0.0016	0.0042	

Figura 17: Timetable Mis (dal nono al diciassettesimo array)

	18	19	20	21	22	23	24	25	26
	mis1_WellkW	mis1_WineCellarkW	mis1_SolarkW	mis2_BarnkW	mis2_LivingRoomOutletskW	mis2_MicrowavekW	mis2_WasherDrierkW	mis2_WellkW	mis2_SolarkW
1	0.0010	0.0077	0.0034	0.0316	9.0889e-04	0.0032	2.2556e-04	1.4445e-05	0.0044
2	0.0010	0.0075	0.0034	0.0318	9.3778e-04	0.0032	1.3889e-04	2.8889e-05	0.0044
3	0.0010	0.0072	0.0034	0.0318	9.4778e-04	0.0033	1.6111e-04	1.6667e-05	0.0044
4	0.0010	0.0066	0.0034	0.0316	9.4111e-04	0.0032	6.0000e-05	3.5556e-05	0.0044
5	0.0010	0.0070	0.0034	0.0318	9.4778e-04	0.0033	1.4000e-04	2.1111e-05	0.0044
6	0.0010	0.0071	0.0034	0.0318	9.5000e-04	0.0033	1.5222e-04	1.5556e-05	0.0044
7	0.0010	0.0070	0.0034	0.0317	9.4667e-04	0.0033	1.2000e-04	2.0000e-05	0.0044
8	0.0010	0.0071	0.0035	0.0315	9.4889e-04	0.0031	1.0667e-04	2.1111e-05	0.0045
9	0.0010	0.0072	0.0034	0.0318	9.5778e-04	0.0033	1.7111e-04	2.1111e-05	0.0044
10	0.0010	0.0072	0.0034	0.0412	9.7111e-04	0.0033	2.0778e-04	3.4444e-05	0.0044
11	0.0010	0.0072	0.0034	0.0318	9.5889e-04	0.0033	1.6556e-04	2.5555e-05	0.0044
12	0.0010	0.0071	0.0034	0.0316	9.5222e-04	0.0033	1.5000e-04	2.5556e-05	0.0044
13	0.0010	0.0069	0.0034	0.0314	9.4889e-04	0.0032	9.5556e-05	1.8889e-05	0.0044
14	0.0010	0.0069	0.0034	0.0314	9.4778e-04	0.0032	1.0000e-04	1.7778e-05	0.0044
15	0.0010	0.0070	0.0034	0.0314	9.4444e-04	0.0032	9.4444e-05	1.7778e-05	0.0044
16	0.0010	0.0071	0.0034	0.0313	9.4000e-04	0.0031	1.1556e-04	2.0000e-05	0.0044
17	0.0010	0.0070	0.0034	0.0313	9.4222e-04	0.0032	1.1444e-04	1.6667e-05	0.0044
18	0.0010	0.0071	0.0034	0.0314	9.3889e-04	0.0032	1.2889e-04	2.3333e-05	0.0044
19	0.0010	0.0071	0.0034	0.0314	9.3556e-04	0.0032	1.1556e-04	2.5556e-05	0.0044
20	0.0010	0.0070	0.0034	0.0313	9.4222e-04	0.0032	1.2333e-04	2.4444e-05	0.0044

Figura 18: Timetable Mis (dal 18esimo al 26esimo array)

3.3.3 Timetable *en*

In aggiunta alla rielaborazione e alla formattazione dei dati contenuti nel dataset è stata creata una nuova timetable contenente i dati inerenti al consumo di energia (kilowattora) dei carichi. Ciò è stato possibile utilizzando gli array inerenti al consumo di potenza descritti nella sezione precedente. Il tutto porterà alla definizione di una timetable composta da 26 array di dimensione 33408x1 elencati nella Figura 19:

en	33408x26 timetable	33408x26
en1_mis1_BarnkW	33408x1 double	33408x1
en1_mis1_DishwasherkW	33408x1 double	33408x1
en1_mis1_FridgekW	33408x1 double	33408x1
en1_mis1_Furnace1kW	33408x1 double	33408x1
en1_mis1_Furnace2kW	33408x1 double	33408x1
en1_mis1_GarageDoorkW	33408x1 double	33408x1
en1_mis1_genkW	33408x1 double	33408x1
en1_mis1_HomeOfficekW	33408x1 double	33408x1
en1_mis1_HouseOverallkW	33408x1 double	33408x1
en1_mis1_Kitchen12kW	33408x1 double	33408x1
en1_mis1_Kitchen14kW	33408x1 double	33408x1
en1_mis1_Kitchen38kW	33408x1 double	33408x1
en1_mis1_LivingRoomkW	33408x1 double	33408x1
en1_mis1_MicrowavekW	33408x1 double	33408x1
en1_mis1_SolarkW	33408x1 double	33408x1
en1_mis1_usekW	33408x1 double	33408x1
en1_mis1_WellkW	33408x1 double	33408x1
en1_mis1_WineCellarkW	33408x1 double	33408x1
en1_mis2_BarnkW	33408x1 double	33408x1
en1_mis2_genkW	33408x1 double	33408x1
en1_mis2_LivingRoomOutletskW	33408x1 double	33408x1
en1_mis2_MicrowavekW	33408x1 double	33408x1
en1_mis2_SolarkW	33408x1 double	33408x1
en1_mis2_usekW	33408x1 double	33408x1
en1_mis2_WasherDrierkW	33408x1 double	33408x1
en1_mis2_WellkW	33408x1 double	33408x1

Figura 19: Array contenuti nella Timetable *en*

Essendo l'energia un dato cumulativo è stata impiegata la funzione Matlab *cumtrapz*, la quale esegue l'integrazione utilizzando il metodo dei trapezi.

Tale funzione è stata utilizzata per tutti gli array del dataset *mis* eseguendo il comando descritto dalla seguente sintassi:

$$b=array_energia; a=array_potenza;$$

$$b=cumtrapz((0:900:33408*900-900)',a/3600);$$

Verrà riportato qui di seguito come esempio di validità generale il comando eseguito solo per un array:

```
en1_mis1_DishwasherKW=cumtrapz((0:900:33408*900-900)',mis1_DishwasherKW/3600);
```

E' importante notare che il numero 900 rappresenta il numero di secondi contenuti in un quarto d'ora.

Il comando sopra indicato porterà alla definizione della nuova timetable (denominata *en*) illustrata nella *Figura 20*, *Figura 21* e *Figura 22*.

	mis1_DateTime	en1_mis1_usekW	en1_mis2_usekW	en1_mis1_HouseOverallkW	en1_mis1_genkW	en1_mis2_genkW	en1_mis1_BarnkW	en1_mis1_DishwasherKW	en1_mis1_FridgekW
1	2016-01-02 00:00:00	0	0	0	0	0	0	0	0
2	2016-01-02 00:15:00	0.3426	0	0.3426	8.4806e-04	0.0011	0.0087	0.0494	0.0375
3	2016-01-02 00:30:00	0.6060	0	0.6060	0.0017	0.0022	0.0174	0.0944	0.0598
4	2016-01-02 00:45:00	0.9381	0	0.9381	0.0026	0.0033	0.0260	0.2570	0.0619
5	2016-01-02 01:00:00	1.3124	0	1.3124	0.0034	0.0044	0.0347	0.4613	0.0631
6	2016-01-02 01:15:00	1.5446	0	1.5446	0.0043	0.0055	0.0434	0.5144	0.0643
7	2016-01-02 01:30:00	1.6966	0	1.6966	0.0051	0.0066	0.0521	0.5148	0.0656
8	2016-01-02 01:45:00	1.8493	0	1.8493	0.0060	0.0077	0.0607	0.5148	0.0669
9	2016-01-02 02:00:00	2.0256	0	2.0256	0.0068	0.0088	0.0694	0.5149	0.0768
10	2016-01-02 02:15:00	2.2114	0	2.2114	0.0077	0.0100	0.0794	0.5150	0.1032
11	2016-01-02 02:30:00	2.4162	0	2.4162	0.0086	0.0111	0.0894	0.5150	0.1412
12	2016-01-02 02:45:00	2.6224	0	2.6224	0.0094	0.0122	0.0980	0.5151	0.1771
13	2016-01-02 03:00:00	2.8656	0	2.8656	0.0103	0.0133	0.1067	0.5152	0.1928
14	2016-01-02 03:15:00	3.1373	0	3.1373	0.0111	0.0144	0.1152	0.5152	0.1940
15	2016-01-02 03:30:00	3.3923	0	3.3923	0.0120	0.0155	0.1238	0.5152	0.1952
16	2016-01-02 03:45:00	3.6179	0	3.6179	0.0128	0.0166	0.1324	0.5152	0.1965
17	2016-01-02 04:00:00	3.8040	0	3.8040	0.0137	0.0177	0.1410	0.5153	0.1978
18	2016-01-02 04:15:00	3.9868	0	3.9868	0.0146	0.0188	0.1496	0.5154	0.2121
19	2016-01-02 04:30:00	4.1852	0	4.1852	0.0154	0.0199	0.1582	0.5154	0.2422
20	2016-01-02 04:45:00	4.3917	0	4.3917	0.0162	0.0210	0.1668	0.5155	0.2739

Figura 20: Timetable mis (dal primo all'ottavo array)

	9	10	11	12	13	14	15	16
	en1_mis1_Furnace1kW	en1_mis1_Furnace2kW	en1_mis1_GarageDoorkW	en1_mis1_HomeOfficekW	en1_mis1_Kitchen12kW	en1_mis1_Kitchen14kW	en1_mis1_Kitchen38kW	en1_mis1_LivingRoomkW
1	0	0	0	0	0	0	0	0
2	0.0254	0.1376	0.0033	0.0127	8.0000e-05	6.0972e-05	3.4722e-06	3.9708e-04
3	0.0469	0.2339	0.0066	0.0254	2.0556e-04	9.9167e-05	6.2501e-06	8.0361e-04
4	0.0649	0.3167	0.0097	0.0382	3.9514e-04	2.2417e-04	9.1668e-06	0.0012
5	0.0898	0.3970	0.0128	0.0510	5.9347e-04	3.8403e-04	1.1945e-05	0.0016
6	0.1313	0.4675	0.0161	0.0637	7.7903e-04	4.3514e-04	1.4306e-05	0.0020
7	0.1822	0.5063	0.0193	0.0765	9.6472e-04	4.5639e-04	1.6806e-05	0.0024
8	0.2286	0.5428	0.0226	0.0893	0.0011	4.9458e-04	1.9722e-05	0.0028
9	0.2555	0.6059	0.0259	0.1020	0.0013	5.3569e-04	2.2361e-05	0.0032
10	0.2714	0.6750	0.0293	0.1147	0.0014	5.7708e-04	2.4306e-05	0.0037
11	0.2890	0.7468	0.0326	0.1274	0.0015	6.2597e-04	2.6250e-05	0.0041
12	0.3123	0.8183	0.0359	0.1401	0.0016	6.7597e-04	2.9028e-05	0.0045
13	0.3881	0.8932	0.0392	0.1545	0.0017	7.2417e-04	3.1806e-05	0.0049
14	0.5101	0.9689	0.0423	0.1694	0.0019	7.7736e-04	3.4167e-05	0.0053
15	0.6307	1.0373	0.0455	0.1825	0.0021	8.3028e-04	3.6389e-05	0.0057
16	0.7143	1.1060	0.0487	0.1953	0.0023	8.6278e-04	3.8334e-05	0.0061
17	0.7676	1.1649	0.0520	0.2080	0.0025	8.8750e-04	4.0001e-05	0.0065
18	0.8151	1.2182	0.0552	0.2207	0.0026	9.2500e-04	4.2223e-05	0.0069
19	0.8564	1.2745	0.0585	0.2336	0.0027	9.6847e-04	4.5140e-05	0.0073
20	0.9026	1.3318	0.0617	0.2466	0.0028	0.0010	4.8056e-05	0.0077

Figura 21: Timetable mis (dal nono al 16esimo array)

	17	18	19	20	21	22	23	24	25	26
	en1_mis1_MicrowavekW	en1_mis1_WellkW	en1_mis1_WineCellarW	en1_mis1_SolarkW	en1_mis2_BarnkW	en1_mis2_LivingRoomOutletskW	en1_mis2_MicrowavekW	en1_mis2_WasherDrierkW	en1_mis2_WellkW	en1_mis2_SolarkW
1	0	0	0	0	0	0	0	0	0	0
2	0.0011	2.5139e-04	0.0019	8.4806e-04	0.0079	2.3083e-04	8.1028e-04	4.5556e-05	5.4167e-06	0.0011
3	0.0021	5.0319e-04	0.0037	0.0017	0.0159	4.6653e-04	0.0016	8.3056e-05	1.1111e-05	0.0022
4	0.0032	7.5806e-04	0.0055	0.0026	0.0238	7.0264e-04	0.0024	1.1069e-04	1.7639e-05	0.0033
5	0.0042	0.0010	0.0072	0.0034	0.0317	9.3875e-04	0.0032	1.3569e-04	2.4722e-05	0.0044
6	0.0053	0.0013	0.0089	0.0043	0.0397	0.0012	0.0041	1.7222e-04	2.9306e-05	0.0055
7	0.0064	0.0015	0.0107	0.0051	0.0476	0.0014	0.0049	2.0625e-04	3.3750e-05	0.0066
8	0.0074	0.0018	0.0124	0.0060	0.0555	0.0017	0.0057	2.3458e-04	3.8889e-05	0.0077
9	0.0085	0.0020	0.0142	0.0068	0.0634	0.0019	0.0065	2.6931e-04	4.4167e-05	0.0088
10	0.0095	0.0023	0.0160	0.0077	0.0725	0.0021	0.0073	3.1667e-04	5.1111e-05	0.0100
11	0.0106	0.0025	0.0178	0.0086	0.0817	0.0024	0.0081	3.6333e-04	5.8611e-05	0.0111
12	0.0117	0.0028	0.0196	0.0094	0.0896	0.0026	0.0089	4.0278e-04	6.5000e-05	0.0122
13	0.0127	0.0030	0.0214	0.0103	0.0975	0.0028	0.0097	4.3347e-04	7.0556e-05	0.0133
14	0.0138	0.0033	0.0231	0.0111	0.1053	0.0031	0.0105	4.5792e-04	7.5139e-05	0.0144
15	0.0148	0.0035	0.0248	0.0120	0.1131	0.0033	0.0114	4.8222e-04	7.9583e-05	0.0155
16	0.0159	0.0038	0.0266	0.0128	0.1210	0.0036	0.0121	5.0847e-04	8.4306e-05	0.0166
17	0.0169	0.0041	0.0283	0.0137	0.1288	0.0038	0.0129	5.3722e-04	8.8889e-05	0.0177
18	0.0180	0.0043	0.0301	0.0146	0.1366	0.0040	0.0137	5.6764e-04	9.3889e-05	0.0188
19	0.0191	0.0046	0.0319	0.0154	0.1445	0.0043	0.0146	5.9819e-04	1.0000e-04	0.0199
20	0.0201	0.0048	0.0336	0.0162	0.1523	0.0045	0.0154	6.2806e-04	1.0625e-04	0.0210

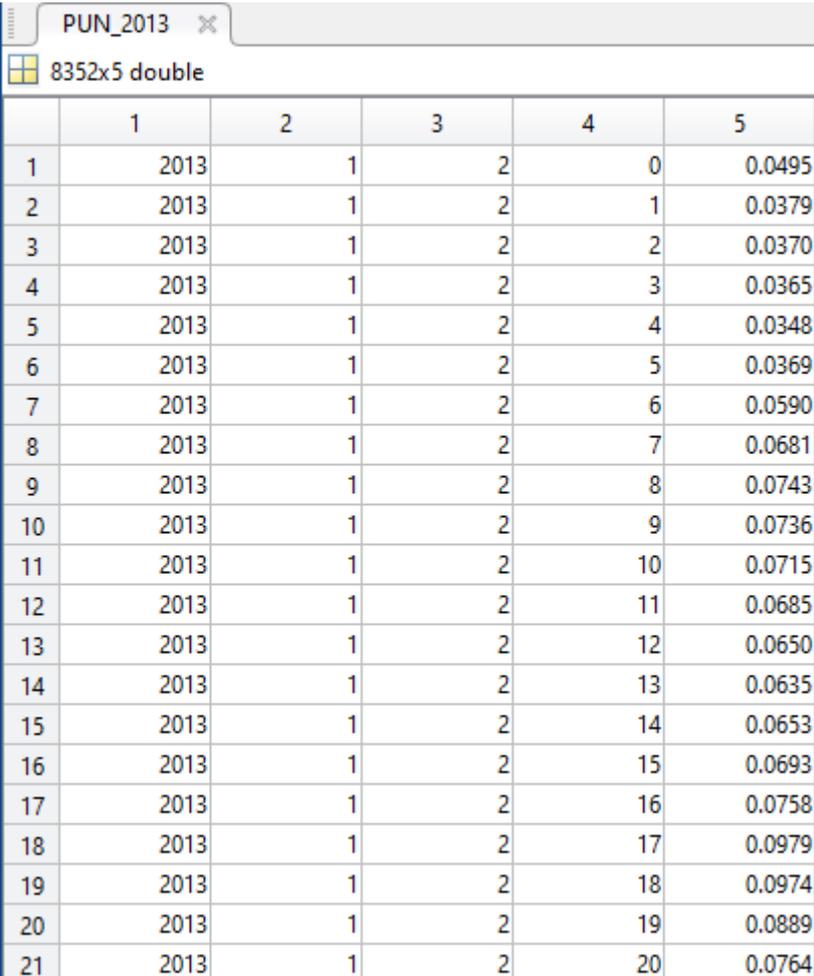
Figura 22: Timetable mis (dal 17esimo al 26esimo array)

3.3.4 PUN

A completare il Workspace di Matlab vi è il PUN (Prezzo Unico Nazionale), documento contenente il prezzo di riferimento dell'energia elettrica di tutto l'anno rilevato sulla borsa elettrica italiana.

Il PUN messo a disposizione di questo elaborato aveva frequenza di un'ora ed era una matrice 8352x5. Nelle prime quattro colonne vi erano rispettivamente anno, mese, giorno e ora mentre nella quinta ed ultima colonna vi era il prezzo dell'elettricità indicato in €/MWh.

Dopo una semplice conversione in €/kWh e dopo aver eliminato le righe in eccesso l'aspetto assunto dal PUN era quello riportato nella *Figura 23*:



	1	2	3	4	5
1	2013	1	2	0	0.0495
2	2013	1	2	1	0.0379
3	2013	1	2	2	0.0370
4	2013	1	2	3	0.0365
5	2013	1	2	4	0.0348
6	2013	1	2	5	0.0369
7	2013	1	2	6	0.0590
8	2013	1	2	7	0.0681
9	2013	1	2	8	0.0743
10	2013	1	2	9	0.0736
11	2013	1	2	10	0.0715
12	2013	1	2	11	0.0685
13	2013	1	2	12	0.0650
14	2013	1	2	13	0.0635
15	2013	1	2	14	0.0653
16	2013	1	2	15	0.0693
17	2013	1	2	16	0.0758
18	2013	1	2	17	0.0979
19	2013	1	2	18	0.0974
20	2013	1	2	19	0.0889
21	2013	1	2	20	0.0764

Figura 23: PUN

4. FORMULAZIONE PROBLEMA

4.1 Tipologie di carico

Questo elaborato, come già indicato nel *paragrafo 2.2*, propone un algoritmo di Reinforcement Learning il cui scopo è quello di ottimizzare l'utilizzo di un carico shiftabile tenendo in considerazione sia il costo dell'elettricità sia il costo di insoddisfazione del cliente.

Generalmente, gli elettrodomestici sono suddivisi in tre tipologie principali in base alle loro caratteristiche e priorità: *carichi controllabili*, *shiftabili* e *non shiftabili* [3]. Di seguito verranno esposte sinteticamente le caratteristiche dei carichi controllabili e non shiftabili mentre verranno analizzate approfonditamente le caratteristiche, le formulazioni matematiche e i vincoli di funzionamento dei carichi shiftabili.

4.1.1 Carichi non shiftabili e carichi controllabili

Sono compresi nella categoria dei carichi non shiftabili elettrodomestici come il frigorifero o persino i sistemi di allarme di sicurezza. Nel momento in cui un carico non shiftabile entra in funzione, il suo funzionamento deve essere garantito in modo continuo e non può essere pianificato. Il consumo di energia di questo tipo di carico sarà sempre uguale alla richiesta di energia mentre il costo di queste apparecchiature elettriche è dato solamente dal costo dell'energia.

Sono invece compresi nella categoria dei carichi controllabili le luci e i condizionatori d'aria. Differentemente dagli altri tipi di carichi, i carichi controllabili possono essere azionati con modalità di consumo di energia flessibile comprese all'interno di un intervallo che va da un valore minimo ad un valore massimo di energia consumata. L'obiettivo di questo tipo di carico è quello di diminuire la richiesta di energia durante le fasce orarie di programmazione di altri carichi considerando però che una diminuzione di energia richiesta ha come conseguenza una maggiore insoddisfazione per l'utente.

4.1.2 Carichi shiftabili

Vanno a comporre questa categoria elettrodomestici come lavatrici, asciugatrici e lavastoviglie. E' possibile programmare la domanda di energia dei carichi shiftabili nelle ore della giornata in cui il prezzo dell'energia ha valori non elevati.

Questa tipologia di carico dispone di due opzioni operative: ON e OFF.

Il consumo energetico effettivo dell'elettrodomestico n all'ora h è indicato con $E_{n,h}$ ed è dato dalla formula indicata qui di seguito:

$$E_{n,h} = I_{n,h} \cdot e_{n,h} \quad (3)$$

dove:

- $n \in \{1, 2, 3, \dots, N\}$ rappresenta il carico n ;
- N è il numero totale di carichi;
- $h \in \{1, 2, 3, \dots, N\}$ rappresenta l'ora h corrente;
- $e_{n,h}$ indica l'energia consumata dal carico n all'ora h in caso di messa in funzione.

Qualunque siano le ore nelle quali lavorerà il carico shiftabile, queste devono trovarsi all'interno del cosiddetto *Working Time* $[T_{n,ini} ; T_{n,end}]$. Il Working Time (periodo di lavoro) è la finestra temporale che il carico shiftabile ha a disposizione per iniziare e terminare il proprio compito.

La differenza tra l'ora in cui l'elettrodomestico entra in un funzione e l'ora in cui inizia la finestra temporale del Working Time darà come risultato il costo di insoddisfazione del cliente. Se, ad esempio, vi è un carico shiftabile avente un Working Time che va dalle 18:00 alle 23:00 che entra in funzione alle 21:00 si avrà un tempo di attesa di 3 ore che inciderà sul costo di insoddisfazione del comfort del cliente.

Conseguentemente a quanto scritto sopra i carichi shiftabili presentano due tipologie di costo: il costo dovuto al consumo di energia e il costo di insoddisfazione del cliente dato dal tempo di attesa. La funzione utilità sotto riportata è data dalla somma di queste due tipologie di costo:

$$U_{n,h} = P_h \cdot E_{n,h} + k_n \cdot (T_{n,w} - T_{n,ini}) \quad (4)$$

$$T_{n,ini} \leq T_{n,w} \leq T_{n,end} \quad (5)$$

dove:

- il primo addendo rappresenta il costo dell'energia elettrica;
- il secondo addendo rappresenta il costo del tempo di attesa;
- P_h è il prezzo dell'energia all'ora h ;
- K_n è un coefficiente che dipende dal sistema;
- $T_{n,ini}$ e $T_{n,end}$ rappresentano rispettivamente l'orario di inizio e fine del Working Time;
- $T_{n,w}$ indica l'ora di inizio dell'operazione.

4.1.3 Funzione Obiettivo

La funzione obiettivo dell'utente è sia minimizzare il costo dell'energia sia minimizzare il costo di insoddisfazione e può essere espressa come segue:

$$\min \sum_{n=1}^N \sum_{h=1}^H \{(1-\rho) \cdot (P_h \cdot E_{n,h}) + \rho \cdot [k_n \cdot (T_{n,w} - T_{n,ini})]\} \quad (6)$$

dove ρ è un parametro di bilancio che permette di ottenere un compromesso tra costo di elettricità e costo del tempo di attesa.

5. IMPLEMENTAZIONE ALGORITMO Q-LEARNING

5.1. Panoramica

Questo elaborato prende spunto dal lavoro svolto in [2] nel quale è stato proposto un algoritmo di gestione del consumo con un'ora di anticipo (one hour-ahead) mediante una rete neurale artificiale ANN (Artificial Neural Network) e un approccio di Reinforcement Learning multi-agente.

La rete neurale artificiale aveva il compito di prevedere, ricevuto il prezzo dell'energia dell'ora successiva con un'ora di anticipo dal service provider, tutti i futuri prezzi delle ore successive della giornata. Questo processo veniva ripetuto ad ogni ora fino alla fine della giornata.

Per semplicità in questo elaborato non è stata creata una rete neurale artificiale e, per simulare la presenza della ANN, si è deciso di utilizzare il PUN (illustrato nel *paragrafo 3.3.4*) avente i prezzi della giornata già noti.

Inoltre si è utilizzato un solo agente e, poiché un agente corrisponde ad un elettrodomestico, è stato ottimizzato il consumo di un singolo elettrodomestico: la lavasciuga.

Osservando lo schema qui di seguito è possibile osservare la metodologia che verrà utilizzata per l'applicazione del Reinforcement Learning:

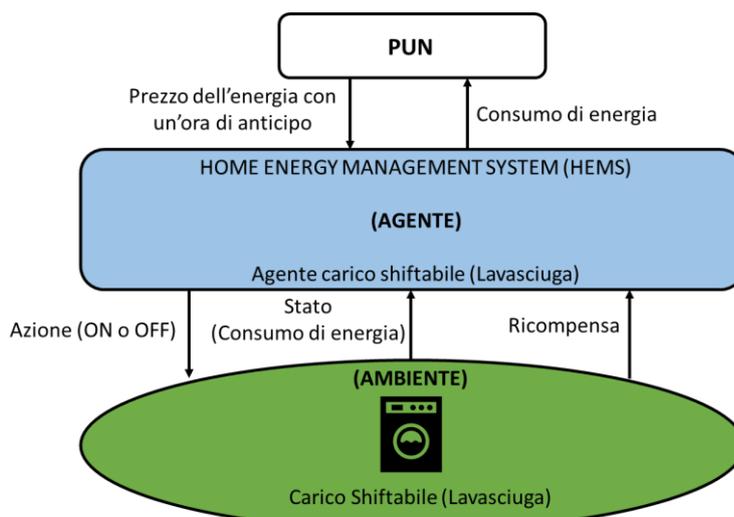


Figura 24: Home Energy Management System (HEMS)

Quanto sopra descritto è stato realizzato sulla base del seguente diagramma di flusso estrapolato dal documento [2]:

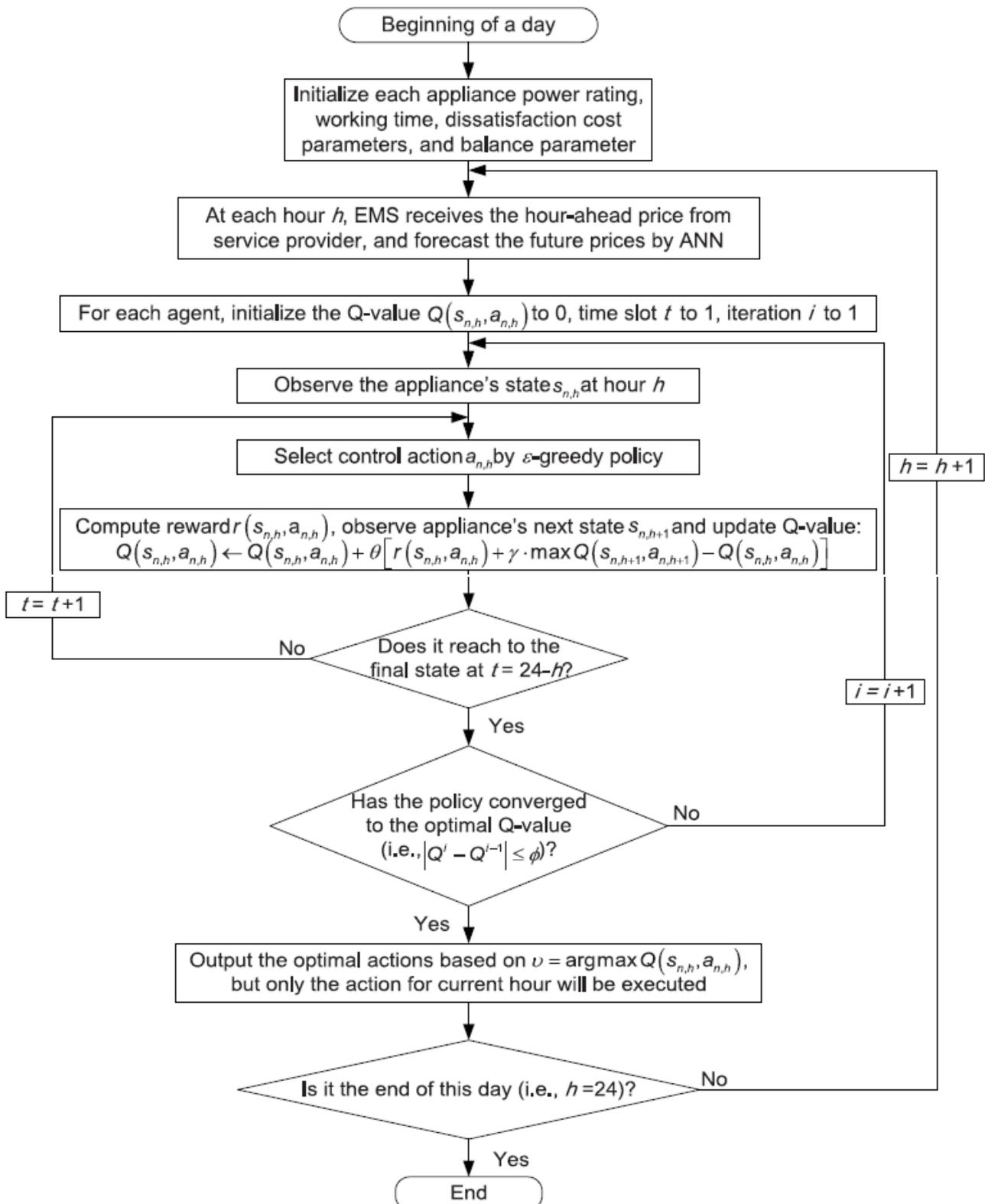


Figura 25: Diagramma di flusso per implementare l' algoritmo di Q-learning

5.1.1. Inizializzazione Q-table

Prima ancora di inizializzare la Q-table sono stati inizializzati i differenti parametri di sistema (k_n , $T_{n,ini}$, $T_{n,end}$, ecc.). Più avanti si avrà modo di analizzare tali parametri e i loro rispettivi valori man a mano che saranno necessari nelle stringhe di codice mostrate nei successivi paragrafi.

Successivamente è stata inizializzata la Q-table. Prendendo in considerazione quanto scritto nel *paragrafo 2.2*, ovvero considerando che le azioni sono disposte orizzontalmente andando a formare le colonne della Q-table, e che gli stati sono disposti verticalmente andando a formare le righe della Q-table, sono state definite le azioni come un array 1x2 con valori [0,1] (dove 0 indica l'azione OFF e 1 indica l'azione ON) e sono stati definiti gli stati come un array 24x1 avente valori da 1 a 24, ovvero le 24 ore che compongono la giornata. Il risultato sarà dunque una matrice di zeri 24x2 descritta dalle seguenti stringhe di codice:

```
%% CREAZIONE Q-TABLE
action=[0,1];
state=[1;2;3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19;20;21;22;23;24];
qtable=zeros(length(state),length(action));
```

5.1.2 PUN

Come scritto nel primo riquadro della *figura 25* ad ogni ora h si riceve il prezzo dell'energia dell'ora successiva e, tramite la ANN, verranno predetti i prezzi di energia futuri di tutte le ore della giornata. Come già spiegato invece nel *paragrafo 5.1*, nel presente studio non è stata utilizzata alcuna ANN ma bensì il PUN e si è fatto in modo di emulare il comportamento della ANN.

Dunque, si riceveranno direttamente tutti i prezzi dell'energia delle ore successive tramite la stringa di codice sotto riportata:

```
for h=1:24
    prices=PUN_2013(33408+h:33432)';
```

La *Figura 26* mostra il contenuto della variabile *prices* nei casi in cui $h=1$, $h=5$ e $h=19$ (osservare che $h=19$ si riferisce all'ora 18:00-19:00):

Casella 33408

h=1	0	0.0495	33408+h
	1	0.0379	
	2	0.0370	
	3	0.0365	
	4	0.0348	
	5	0.0369	
	6	0.0590	
	7	0.0681	
	8	0.0743	
	9	0.0736	
	10	0.0715	
	11	0.0685	
	12	0.0650	
	13	0.0635	
	14	0.0653	
	15	0.0693	
	16	0.0758	
	17	0.0979	
	18	0.0974	
	19	0.0889	
	20	0.0764	
	21	0.0715	
	22	0.0697	
	23	0.0657	

h=5	4	0.0348	33408+h
	5	0.0369	
	6	0.0590	
	7	0.0681	
	8	0.0743	
	9	0.0736	
	10	0.0715	
	11	0.0685	
	12	0.0650	
	13	0.0635	
	14	0.0653	
	15	0.0693	
	16	0.0758	
	17	0.0979	
	18	0.0974	
	19	0.0889	
	20	0.0764	
	21	0.0715	
	22	0.0697	
	23	0.0657	

h=19	18	0.0974	33408+h
	19	0.0889	
	20	0.0764	
	21	0.0715	
	22	0.0697	
	23	0.0657	

Figura 26: Contenuto della variabile prices per $h=1$, $h=5$ e $h=19$

5.1.3 Working Time

Qualunque siano le due ore nelle quali lavorerà l'elettrodomestico (in questo caso la lavasciuga) queste due ore si troveranno all'interno del cosiddetto Working Time la quale definizione è stata già approfondita nel *paragrafo 4.1.2*.

Il working time per questo elettrodomestico è stato impostato tra le 18:00 e le 23:00, ne consegue che la lavatrice avrà a disposizione 5 ore tra cui scegliere 2 ore nelle quali è più conveniente lavorare.

```
par_tn_ini=19; %18:00-19:00
par_tn_end=23; %22:00-23:00
```

Per tale motivo si è forzato l'elettrodomestico ad eseguire l'azione OFF (azione 0) nelle prime 18 ore della giornata. Ciò è stato reso possibile dalla *condizione if* illustrata qui di seguito:

```

while(converge==false)
    current_state=h;
    % Non lavoro nelle prime 18 ore
    if current_state < par_tn_ini
        qtable(current_state,1)=0;
        qtable(current_state,2)=-1;
        break
    end

```

Se lo stato (l'ora corrente) è minore della prima ora del Working Time (ovvero l'ora 19) verrà eseguita l'azione di OFF che ha come indirizzo la prima colonna della Q-table. Per far sì che ciò avvenga è stato assegnato alle prime 18 ore della prima colonna (rappresentante l'azione OFF) un valore maggiore rispetto a quello delle prime 18 ore della seconda colonna (rappresentante l'azione ON), rispettivamente 0 e -1. Infine, per mezzo della funzione *break*, si uscirà dal ciclo *while* e si transiterà nell'ora successiva.

Di conseguenza, le prime 18 righe della Q-table saranno così organizzate:

	A₀ (OFF)	A₁ (ON)
S1	0	-1
S2	0	-1
S3	0	-1
S4	0	-1
S5	0	-1
S6	0	-1
...	0	-1
S18	0	-1

Figura 27: Q-table nelle prime 18 ore della giornata

5.1.4 Politica ϵ -greedy: Esplorazione e Sfruttamento

Dall'ora 18:00 in avanti entrerà in gioco l'algoritmo Q-learning. Il Q-agent dovrà dunque intraprendere un'azione decidendo se optare per l'*esplorazione* o per lo *sfruttamento* dell'ambiente (terminologie già approfonditamente discussi nel *paragrafo 2.2.1*).

È stato deciso di far innanzitutto esplorare il Q-agent per poi, una volta acquisita una sufficiente conoscenza dell'ambiente, passare allo sfruttamento. Per far sì che ciò avvenga sono stati dichiarati all'inizio del codice due parametri: *par_epsilon* e *par_epsilon_decay*.

```
par_epsilon=1;
par_epsilon_decay=0.99;
```

Le righe di codice sotto riportate mostreranno sotto quale condizione il Q-agent opterà per sfruttare l'ambiente:

```
for i=1:24-h
%% SELEZIONE DI UN'AZIONE TRAMITE SFRUTTAMENTO O ESPLORAZIONE
random_value=rand;
%Sfruttamento:
if random_value >= par_epsilon
    par_epsilon=par_epsilon*par_epsilon_decay;
    [old_qvalue,position_action]= max(qtable(current_state,:));
    if position_action==1
        current_action=0;
    else if position_action==2
        current_action=1;
    end
end
end
```

Per mezzo della funzione *rand* verrà assegnato randomicamente un valore compreso tra 0 e 1 alla variabile *random_value*. Se il valore contenuto nella variabile *random_value* sarà maggiore o uguale al valore contenuto nella variabile *par_epsilon* si entrerà in questa prima parte della condizione *if* e il Q-agent sfrutterà l'ambiente.

Quanto sopra scritto significa che il Q-agent assegnerà alla variabile *old_qvalue* il valore maggiore presente nella riga *current_state* (ovvero l'ora che si sta attualmente valutando) mentre assegnerà alla variabile *position_action* il valore corrispondente all'indice della colonna nella quale si trova l'*old_qvalue*.

Dunque se la variabile *position_action* conterrà il valore 1 verrà selezionata l'azione 0 (dunque *current_action=0*) mentre se la variabile *position_action* conterrà il valore 2 verrà selezionata l'azione 1 (dunque *current_action=1*). La seguente figura illustra un semplice esempio della casistica sopra descritta:

position_action=1 current_action=0	position_action=2 current_action=1
45	64
7	5
42	9
...	...
3	23

Figura 28: Esempio di sfruttamento

Ad ogni azione scelta la variabile rappresentativa del *tasso di esplorazione* ϵ , *par_epsilon*, verrà fatta decadere venendo moltiplicata per la variabile *par_epsilon_decay*. Ciò permetterà di avere, ad ogni azione scelta, un tasso di esplorazione sempre più basso e dunque, col passare delle iterazioni, aumenterà le probabilità che il Q-agent opti per lo sfruttamento.

È evidente che, siccome inizialmente il tasso di esplorazione è pari a 1, nelle prime iterazioni non si entrerà praticamente mai nella prima parte della condizione *if* inerente allo sfruttamento, ma bensì nell'*else*, la parte inerente all'esplorazione:

```
else %Esplorazione:
    par_epsilon=par_epsilon*par_epsilon_decay;
    current_action=randi(2)-1;
    if current_action==0
        position_action=1;
        old_qvalue=qtable(current_state,position_action);
    else if current_action==1
        position_action=2;
        old_qvalue=qtable(current_state,position_action);
    end
end
end
```

Attraverso il comando *randi(2)-1* verrà sorteggiato randomicamente un valore pari a 0 o 1 e tale valore verrà assegnato alla variabile *current_action*. Se il valore assegnato alla variabile *current_action* sarà uguale a 0, verrà assegnato alla variabile *current_position* il valore 1 e la variabile *old_qvalue* conterrà il valore localizzato nella riga *current_state* (ovvero l'ora *h* corrente) alla colonna *position_action* (ovvero la prima colonna. Mentre, se il valore assegnato alla variabile *current_action* sarà uguale a 1, verrà assegnato alla variabile *current_position* il valore 2 e la variabile *old_qvalue* conterrà il valore localizzato nella riga *current_state* (ovvero l'ora *h* corrente) alla colonna *position_action* (ovvero la seconda colonna).

Ad esempio, nella tabella sotto riportata se $current_state=1$ e il valore sorteggiato porterà ad avere $current_action=0$, di conseguenza si avrà $position_action=1$ e alla variabile old_qvalue verrà assegnato il valore 45:

	position_action=1	position_action=2
	current_action=0	current action=1
current_state=1	45	64
	7	5
	42	9

	3	23

Figura 29: Esempio di esplorazione

5.1.5 Memorizzazione Q^{i-1}

Prima ancora di procedere con in calcolo della ricompensa e con il calcolo del nuovo q-value occorre memorizzare, se si è nel caso in cui la variabile $current_state$ è uguale all'ora h , il valore contenuto nella variabile di quello che diventerà a breve il "vecchio" q-value: l' old_qvalue . Quanto appena descritto avviene per mezzo delle seguenti righe di codice:

```

%% Memorizzo solo il vecchio valore all'ora h che poi confronterò col nuovo:
save_hour=h;
if save_hour==current_state
    save_old_qvalue=qtable(save_hour,position_action);
end

```

La memorizzazione di tale valore sarà necessaria per verificare la convergenza del q-value al valore ottimale. Si acquisirà una maggiore comprensione di questa parte di codice dopo la lettura del *paragrafo 5.1.8*.

5.1.6 Calcolo Ricompensa

La ricompensa è data dall'inverso della funzione utilità.

La funzione utilità è data dall'Eq. (4) che, per comodità, verrà scritta nuovamente:

$$U_{n,h} = P_h \cdot E_{n,h} + k_n \cdot (T_{n,w} - T_{n,ini})$$

È lampante notare come nel codice alcuni termini di questa equazione siano a noi ancora ignoti e vadano dunque calcolati.

Procedendo con ordine, si comincerà con l'assegnazione del prezzo dell'energia all'ora h , ovvero P_h , ad una variabile che verrà denominata *current_price*:

```
current_price=prices(i);
```

dove *prices* è la variabile dichiarata nel *paragrafo 5.1.2.* contenente tutti i prezzi della giornata mentre, osservando le righe di codice del *paragrafo 5.1.4,* si può notare che i sia la variabile inizializzata nel *ciclo for* che itero secondo tale regola: $i=1:24-h$.

Come già indicato nel *paragrafo 4.1.2.* $E_{n,h}$ rappresenta il consumo dell'elettrodomestico n all'ora h e sarà noto nel codice come la variabile denominata *current_price*. Il valore di tale variabile viene calcolato tramite la formula (3) che, anch'essa, verrà riscritta per comodità:

$$E_{n,h} = I_{n,h} \cdot e_{n,h}$$

dove $I_{n,h}$ è il valore assegnato alla variabile *current_action* mentre $e_{n,h}$ viene ricavato analizzando l'array "*en1_mis2_WasherDrierkWh*" della timetable *en*, ovvero la timetable contenente i valori del consumo di energia di ogni carico. In particolare, è stato verificato il consumo di energia in un'ora della lavasciuga quando quest'ultima è in funzione. Osservando il consumo di energia di diverse ore nelle quali la lavasciuga era in funzione ed operando una semplice media aritmetica si è adottato come $e_{n,h}$ il valore di 0.04 kwh. Risulta dunque:

```
current_energy=current_action*0.04;
```

Per quanto concerne il parametro kn al fine di avere i due addendi della formula (4) dello stesso ordine di grandezza si è impostato $kn=0.01$.

Il parametro $T_{n,w}$ rappresenta l'ora nel quale voglio verificare quanto sia conveniente azionare o meno il mio carico shiftabile e sarà la mia variabile *current_state*.

Infine, considerando la funzione obiettivo (6), si è impostato il parametro di bilancio $\rho=0.3$ in modo che l'algoritmo Q-learning dia leggermente più peso al costo dell'elettricità ($1-\rho=0.7$) a discapito del costo di insoddisfazione del cliente dovuto al tempo di attesa ($\rho=0.3$).

Ricapitolando, la funzione di utilità assume nel codice questa forma:

```
current_utility=((1-par_ro)*(current_price*current_energy))+1+(par_ro*(par_kn*(par_tn_w-par_tn_ini)));
```

Volendo riassumere quanto scritto in questo paragrafo:

```
%% CALCOLO REWARD
current_energy=current_action*0.04;
par_tn_w=current_state;
current_price=prices(i);
current_utility=((1-par_ro)*(current_price*current_energy))+1+(par_ro*(par_kn*(par_tn_w-par_tn_ini)));
current_reward=1/current_utility;
```

5.1.7 Calcolo nuovo Q-value

Ad ogni azione il q-value associato a quella determinata azione e a quel determinato stato varia, si aggiorna. Nell'aggiornamento del q-value una parte fondamentale viene svolta dall'equazione di Bellman (1) che, per comodità, verrà riscritta:

$$Q_v^*(s_h, a_h) = r(s_h, a_h) + \gamma \cdot \max Q(s_{h+1}, a_{h+1})$$

Il fattore di sconto γ è un parametro compreso tra 0 e 1. Il fattore di sconto indica quanto siano importanti per l'agente le ricompense future. Un $\gamma=0$ renderà l'agente "opportunista" facendo sì che consideri solamente la ricompensa corrente, mentre un $\gamma=1$ renderà l'agente particolarmente attento alle ricompense future. Se si desidera che l'agente trovi un bilanciamento tra le ricompense attuali e future occorre impostare γ ad un valore compreso tra 0 e 1.

Dato che il Q-agent deve valutare quanto sia conveniente o meno azionare l'elettrodomestico all'ora h rispetto alle successive ore è necessario che gran parte dell'attenzione dell'agente sia rivolta alle ricompense future e si è così impostato il valore della variabile relativa al fattore di sconto *par_gamma* a 0.95

```
par_gamma=0.95;
```

mentre la formula di Bellman utilizzata per calcolare il cosiddetto "valore appreso" $Q_v^*(s_h, a_h)$ diverrà:

```
next_state=current_state+1;
star_qvalue=current_reward+(par_gamma*max(qtable(next_state, :)));
```

Una volta calcolato il valore appreso sarà possibile applicare la formula che permette di ricavare il valore del q-value aggiornato (2):

$$Q(s_h, a_h) \leftarrow Q(s_h, a_h) + \vartheta \cdot [r(s_h, a_h) + \gamma \cdot \max Q(s_{h+1}, a_{h+1}) - Q(s_h, a_h)]$$

Il tasso di apprendimento ϑ determina in che misura il nuovo Q-value sovrascrive i vecchi Q-value. Un $\vartheta=0$ impedirebbe all'agente di apprendere, facendogli sfruttare solamente la conoscenza pregressa. Un $\vartheta=1$ farebbe sì che l'agente tenesse conto solamente della stima corrente, ignorando la conoscenza pregressa. È possibile ottenere un compromesso tra le due possibilità sopra citate impostando ϑ ad un valore compreso tra 0 e 1.

È di notevole importanza che l'agente tenga conto della conoscenza pregressa e, di conseguenza, il valore della variabile relativa al tasso di apprendimento *par_theta* è stato impostato a 0.1

```
par_theta=0.1;
```

La formula che permette di ricavare il nuovo Q-value assume nel codice la seguente forma:

```
qtable(current_state, position_action)=old_qvalue+(par_theta*(star_qvalue-old_qvalue));
new_qvalue=qtable(current_state, position_action);
```

Riassunto quanto scritto in questo paragrafo:

```
%% CALCOLO NUOVO Q-VALUE
next_state=current_state+1;
star_qvalue=current_reward+(par_gamma*max(qtable(next_state,:)));
qtable(current_state,position_action)=old_qvalue+(par_theta*(star_qvalue-old_qvalue));
new_qvalue=qtable(current_state,position_action);
```

5.1.8 Verifica convergenza al Q-value ottimale

Nel *paragrafo 5.1.5.* era stato assegnato il valore Q^{i-1} nella variabile *save_old_qvalue*. Al fine di verificare la convergenza al Q-value ottimale è necessario assegnare il valore Q^i (Q-value aggiornato) alla variabile *save_new_qvalue*. Il tutto, viene eseguito solamente se l'ora h è uguale allo stato corrente, come indicato nelle seguenti righe di codice:

```
%% Memorizzo solo il nuovo valore all'ora h che poi confronterò col vecchio:
if save_hour==current_state
    save_new_qvalue=qtable(save_hour,position_action);
end
```

È necessario ora transitare nello stato successivo al fine di aggiornare i valori di ogni stato della Q-table arrivando sino alla fine della giornata:

```
%% Al prossimo ciclo esegue le stesse operazioni ma per lo stato successivo
current_state=current_state+1;
```

Una volta aggiornato anche il Q-value dell'ultimo stato verrà verificata la convergenza del Q-value relativo all'ora h al valore ottimale utilizzando la formula presente nel diagramma di flusso:

$$|Q^i - Q^{i-1}| \leq \phi$$

Dunque se la differenza tra il Q-value aggiornato ed il Q-value precedente è minore di un parametro di sistema ϕ la convergenza è stata raggiunta. Osservando in che misura gli Q-value aumentano di valore ad ogni iterazioni è stato impostato il valore della variabile inerente al parametro ϕ *par_phi* a 0.00001, dunque:

```
par_phi=0.00001;
```

Ricapitolando quanto sopra scritto la condizione di convergenza assume nel codice la seguente forma:

```
%% CONVERGENZA AL Q-VALUE OTTIMALE:
converge_value=abs(save_new_qvalue-save_old_qvalue);
if converge_value <= par_phi
    converge=true;
    qtable
end
```

dove *qtable* è necessario per visualizzare a schermo la Q-table nel momento in cui il Q-value dell'ora *h* converge al valore ottimale.

Se la convergenza non risultasse verificata la variabile *converge* rimarrebbe impostata al valore *false* e non si uscirebbe dal *ciclo while*. Di conseguenza le operazioni sopra descritte verrebbero ripetute fino a quando la condizione di convergenza risulterà verificata.

Una volta che la condizione di convergenza risulterà verificata la variabile *converge* assumerà il valore *true* e si uscirà dal *ciclo while*. In questo caso la variabile *h* verrà incrementata di uno, si riceveranno i nuovi prezzi dall'ora *h* fino al termine della giornata e si ripeteranno tutte le azioni sopra descritte fino a quando il Q-agent riuscirà a verificare nuovamente la condizione di convergenza trovando il Q-value ottimale per la nuova ora *h*.

Il ciclo si ripete fino a quando la variabile *h* non sarà uguale a 24, valore che indicherà la fine della giornata.

6. RISULTATI

Il codice sviluppato è stato eseguito numerose volte al fine di valutarne il potenziale. Sono stati ottenuti risultati discreti ma che dispongono ancora di un notevole margine di miglioramento.

Al fine di poter esprimere un giudizio sulla base di valutazioni affidabili verranno illustrati qui di seguito i prezzi estratti dal PUN (riportati in €/kWh) dalle ore 18:00 alle ore 23:00 (è bene ricordare che, nel PUN, il valore 18 indica l'ora 18:00-19:00 e così tutte le altre):

18	0.0974
19	0.0889
20	0.0764
21	0.0715
22	0.0697

Figura 30: PUN dalle ore 18:00 alle ore 23:00

Osservando la *Figura 30* e ricordando che per questa simulazione è stato scelto $p=0.3$ (*paragrafo 5.1.6*), si può dedurre che l'ora che va dalle 20:00 alle 21:00 è quella più "equilibrata" per quanto concerne il costo di elettricità e il costo dovuto al tempo di attesa. Questo poiché presenta un prezzo dell'energia non troppo distante dal prezzo più conveniente delle cinque ore riportate nella tabella (ovvero il prezzo dell'ora 22:00–23:00, con 0.0697 €/kWh) ed ha un tempo di attesa non eccessivamente lungo (2 ore).

Dalle prove effettuate è infatti risultato che l'ora più spesso coinvolta (nel 64% dei casi) tra tutte nell'operazione di funzionamento (azione ON) del carico shiftabile è stata quella che va dalle 20:00 alle 21:00.

A seguire, con una percentuale del 56%, vi sono le ore 21:00-22:00 e 22:23-23:00, ovvero le ore aventi il prezzo dell'energia più conveniente ma il costo di attesa più elevato.

7. CONCLUSIONI

In questo elaborato è stato proposto un algoritmo di Reinforcement Learning con approccio Q-learning avente l'obiettivo di ottimizzare il funzionamento di un carico shiftabile (una lavasciuga) ricercando un adeguato compromesso tra costo di elettricità e costo di insoddisfazione del cliente dovuto al tempo di attesa.

Vi sono ancora significativi margini di miglioramento da poter raggiungere riguardo l'efficienza del codice ma i risultati attuali sono promettenti e incoraggianti.

La grande mole di dati riorganizzata nelle timetable del *capitolo 3.3* può essere utilizzata per ambire a progetti più ambiziosi nei quali porsi l'obiettivo di ottimizzare il consumo di un'intera casa considerando altri carichi shiftabili (come la lavastoviglie), carichi controllabili (come l'impianto di riscaldamento/raffreddamento), carichi non shiftabili (come il frigorifero) e inserendo anche la generazione del pannello solare; in questo caso l'approccio potrebbe essere realizzato mediante un algoritmo di Reinforcement Learning Multi-Agent.

BIBLIOGRAFIA

- [1]Richard S. Sutton, Andrew G. Barto, “Reinforcement Learning: An Introduction”, a Bradford Book, Cambridge, MA, USA, 2018
- [2]Sean Barker, Aditya Mishra, David Irwin, Emmanuel Cecchet, and Prashant Shenoy, “Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes”, University of Massachusetts Amherst, 2021
- [3]Renzhi Lu, Seung Ho Hong, Mengmeng Yu, “Demand Response for Home Energy Management Using Reinforcement Learning and Artificial Neural Network”, IEEE Transactions on Smart Grid, Vol. 10, No. 6, pp. 6629-6639, 2019