

POLITECNICO DI MILANO  
Facoltà di Ingegneria dell'Informazione  
Corso di Laurea Specialistica in Ingegneria Informatica  
Dipartimento di Elettronica e Informazione



Energy Coach: un sistema per il monitoraggio e  
la previsione dei consumi energetici domestici  
basato su reti di sensori

Relatore: Prof. Antonio CAPONE  
Correlatore: Ing. Antimo BARBATO

Tesi di Laurea di:  
Marta Michela RODOLFI Matr. 735051  
Davide TAGLIAFERRI Matr. 734704

Anno Accademico 2009-2010

*Ai miei compagni di viaggio*

*Marta*

*Alla mia famiglia*

*Davide*

# Ringraziamenti

*Ripercorrendo la mia strada, non posso fare a meno di pensare a quanti mi hanno fatto da ala, come quando un ciclista, impegnando ogni fibra del suo essere per vincere la vetta che lo separa dal traguardo, trova forza e sostegno dal tifo che le persone gli offrono lungo il percorso.*

*Il mio primo pensiero, ovviamente, va ai miei genitori, senza i quali non sarei mai potuta arrivare a questo punto, per quell'aiuto tacito o esplicito che tante volte è venuto dal loro cuore: mi riferisco a tutte le occasioni in cui mi hanno incoraggiata e sopportata dimenticandosi del mio caratterino non facile.*

*Alla mia sorellina, che mi ha sempre dato un sacco di consigli sia per l'università che per la vita, e alla piccola Sofia che piacevolmente mi distraeva dallo studio facendomi inventare ogni week-end giochi nuovi. A Pietro per essere stato come un fratello maggiore e avermi regalato un sacco di risate.*

*Un ringraziamento particolare va al Prof. Antonio Capone, per avermi dato l'opportunità di svolgere questa tesi e al mio correlatore Ing. Antimo Barbato per la sua costante presenza e disponibilità, per tutti i chiarimenti e le spiegazioni che ci ha fornito. Ringrazio anche tutti i compagni di lavoro del laboratorio ANTLab di Lambrate che in questi mesi ci hanno sostenuto.*

*Un grande grazie a tutti voi, miei favolosi compagni di viaggio con i quali ho condiviso questi cinque anni, fatti sì di studio e fatica per gli esami e le scadenze dei progetti, ma anche di vacanze trascorse insieme, pomeriggi passati a chiacchiere e passeggiare per le vie di Milano. In particolare ringrazio Matteo, grande organizzatore di cene ed eventi, dalle serate al cinema a quelle passate a giocare*

*tutti insieme a Risiko e a Poker.*

*Giulio, Oscar, Luca, Fabio, Mario, Max: i miei mitici conterranei che mi hanno fatto sentire sempre a casa (dai pizzoccheri preparati insieme alla ricerca del Braulio per le vie di Roma).*

*Pippo per essere stato un piacevole compagno di lunghe “chattate” pomeridiane e serali.*

*Sara ed Alessandro, per le serate trascorse a guardare “I cesaroni” e a mimare le cose più strane che ci venivano in mente.*

*Davide Rizzo, il nostro nerd, per tutti i consigli informatici.*

*Leonardo per le sue perle di geografia e la sua invidia dei miei dolci, Laura, Alessandra e Daniele per avermi sopportato nei progetti fatti insieme.*

*Infine grazie a Davide Colli, Anna, Sara e Cate per esservi uniti nel nostro grande gruppo ed aver condiviso le serate in compagnia.*

*Un pensiero anche a voi mie care amiche grosine: grazie Francy per avermi ascoltata, capita e sostenuta in questi anni trascorsi insieme a Milano, Anastasia per i spassosi viaggi in treno, i supporti morali e l’interessamento che mi hai sempre dimostrato, Noe per esserci sempre e comunque, nonostante la lontananza e i miei impegni.*

*Ultimo, come da copione, tutt’altro che ultimo per importanza, Davide, mio coautore di questa tesi per avere condiviso con me questo lavoro.*

*Gli angeli esistono, ma a volte non hanno le ali, allora li chiamiamo amici....*

*Grazie a tutti*

*Marta*

*Giunto al termine di un lungo percorso, in prossimità del traguardo, vorrei ringraziare tutte le persone che, in un modo o nell'altro, mi hanno sostenuto nel raggiungimento di questo importante obiettivo.*

*Ringrazio la mia famiglia, mamma e papà, mia sorella Sara, le mie nonne e zia Mariella per l'affetto e il supporto che non mi hanno mai fatto mancare in questi anni.*

*Ringrazio il mio relatore, Prof. Antonio Capone, per la possibilità che mi ha dato di svolgere questo lavoro di tesi e il mio correlatore, Ing. Antimo Barbato, per la professionalità e il tempo che ci ha dedicato in questi mesi. Ringrazio inoltre tutti i ragazzi del laboratorio ANTLab del Politecnico.*

*Un grande grazie a tutti i compagni di università che ho conosciuto in questi anni, non solo per l'aiuto che ho ricevuto da loro nello studio, ma anche per i momenti di svago e di divertimento passati insieme.*

*Un ringraziamento particolare a Marta per aver condiviso con me la realizzazione di questo lavoro di tesi.*

*Infine ringrazio i miei amici di sempre per tutti i momenti belli passati insieme.*

*Grazie a tutti*

*Davide*

# Indice

Elenco delle figure . . . . .	VI
Elenco delle tabelle . . . . .	XI
<b>Sommario</b>	<b>1</b>
<b>Abstract</b>	<b>2</b>
<b>Introduzione</b>	<b>3</b>
<b>1 Automazione domestica e Risparmio energetico</b>	<b>8</b>
1.1 Home automation . . . . .	8
1.2 Monitoraggio dell'energia . . . . .	12
1.2.1 Google Powermeter . . . . .	12
1.2.2 Control4 . . . . .	13
1.2.3 My Hohm Center . . . . .	14
1.2.4 MavHome . . . . .	16
1.3 Reti di sensori wireless . . . . .	17
1.3.1 Il Progetto AIM . . . . .	20
1.4 Algoritmi di predizione . . . . .	22
1.4.1 Modello di regressione . . . . .	25
1.4.2 Reti neurali . . . . .	26
1.4.3 Alberi di decisione . . . . .	26
1.4.4 Support Vector Regression . . . . .	27

<b>2</b>	<b>Architettura del sistema</b>	<b>28</b>
2.1	Introduzione . . . . .	28
2.2	Linguaggi di programmazione . . . . .	30
2.3	Rete di sensori . . . . .	31
2.4	Rmi Client . . . . .	32
2.5	Rmi Server . . . . .	33
2.6	Energy Coach . . . . .	34
<b>3</b>	<b>Rete di sensori</b>	<b>35</b>
3.1	Introduzione . . . . .	35
3.2	ACme . . . . .	36
3.2.1	Hardware . . . . .	36
3.2.2	Calibrazione . . . . .	38
3.2.3	La rete di sensori ACme . . . . .	39
3.3	Micaz e Telosb . . . . .	39
3.4	TinyOS . . . . .	41
3.5	Realizzazione della rete . . . . .	41
3.5.1	Formato del messaggio . . . . .	42
<b>4</b>	<b>Algoritmo di predizione</b>	<b>43</b>
4.1	Introduzione . . . . .	43
4.2	Funzionamento dell'algoritmo . . . . .	43
4.2.1	Creazione dei cluster . . . . .	45
4.2.2	Rappresentazione dell'andamento dell'utilizzo . . . . .	48
4.2.3	Predizione dell'utilizzo . . . . .	48
4.2.4	Predizione degli orari di accensione . . . . .	50
4.2.5	Predizione della durata di funzionamento . . . . .	51
4.3	Variante dell'algoritmo . . . . .	52

<b>5</b>	<b>Server</b>	<b>53</b>
5.1	Introduzione . . . . .	53
5.2	Rmi Server . . . . .	54
5.2.1	Raccolta dati dalla rete di sensori . . . . .	55
5.2.2	Gestione del database . . . . .	56
5.2.3	Predizione dell'utilizzo degli elettrodomestici . . . . .	56
5.2.4	Funzionalità di risparmio energetico . . . . .	57
5.2.5	Scheduling degli elettrodomestici . . . . .	57
5.3	Database <i>Energy Manager</i> . . . . .	58
<b>6</b>	<b>Energy Coach</b>	<b>60</b>
6.1	Introduzione . . . . .	60
6.2	Android . . . . .	60
6.3	Realizzazione dell'applicazione . . . . .	62
6.4	Energy Coach . . . . .	62
6.4.1	Configurazione delle stanze e degli elettrodomestici presenti in esse . . . . .	63
6.4.2	Homepage . . . . .	64
6.4.3	Controllo degli elettrodomestici . . . . .	65
6.4.4	Visualizzazione consumi e costi . . . . .	68
6.4.5	Pianificazione delle attività dei dispositivi . . . . .	70
6.4.6	Risparmio energetico . . . . .	71
<b>7</b>	<b>Test e risultati numerici</b>	<b>74</b>
7.1	Introduzione . . . . .	74
7.2	Risultati numerici delle simulazioni . . . . .	74
7.2.1	Parametri dell'algoritmo di predizione . . . . .	74
7.2.2	Scenari . . . . .	85
7.2.3	Periodo di training . . . . .	86



7.2.4	Prestazioni dell'algoritmo di predizione . . . . .	103
7.3	Confronto con la variante dell'algoritmo . . . . .	110
7.4	Considerazioni sui test fatti . . . . .	111
7.5	Risultati numerici dei test sperimentali . . . . .	112
7.5.1	Valutazione sperimentale dell'algoritmo di predizione . . .	117
<b>8</b>	<b>Conclusioni</b>	<b>120</b>
8.1	Considerazioni sul lavoro svolto . . . . .	120
8.2	Sviluppi futuri . . . . .	122

# Elenco delle figure

1	Previsione del consumo di elettricità . . . . .	4
1.1	Modello di un sistema di energy management . . . . .	11
1.2	Google Powermeter . . . . .	13
1.3	Control4 . . . . .	14
1.4	My Hohm Center . . . . .	15
1.5	Achitettura di MavHome . . . . .	17
1.6	Achitettura di AIM . . . . .	20
1.7	Architettura delle applicazioni context-aware . . . . .	22
1.8	Tecniche per predire i consumi elettrici . . . . .	25
2.1	Architettura del sistema . . . . .	29
2.2	Rete di sensori . . . . .	31
2.3	Struttura Rmi Client . . . . .	32
2.4	Struttura Rmi Server . . . . .	33
2.5	Struttura Energy Coach . . . . .	34
3.1	Architettura dell'ACme . . . . .	36
3.2	Micaz . . . . .	40
3.3	Telosb . . . . .	40
3.4	Struttura del messaggio inviato dall'ACme . . . . .	42
4.1	Tabella consumo . . . . .	44
4.2	Struttura delle classi Java presenti sul server . . . . .	45

4.3	Tabella cluster . . . . .	46
4.4	Profilo di utilizzo del forno AA213 . . . . .	46
4.5	Cluster on creati dall'algoritmo . . . . .	47
4.6	Cluster off creati dall'algoritmo . . . . .	47
4.7	Media cluster on . . . . .	47
4.8	Media cluster off . . . . .	47
4.9	Tabella predizione . . . . .	51
5.1	Architettura del sistema . . . . .	53
5.2	Funzionamento XML-RPC . . . . .	55
5.3	Database . . . . .	59
6.1	Architettura di Android . . . . .	61
6.2	Homepage Energy Coach . . . . .	64
6.3	Menù Energy Coach . . . . .	65
6.4	Scheda elettrodomestico . . . . .	66
6.5	Scheda costi e consumi . . . . .	67
6.6	Diagramma consumi per categoria . . . . .	68
6.7	Consumi per piano e per utente . . . . .	69
6.8	Consumi per settimana . . . . .	70
6.9	Scheduling . . . . .	71
6.10	Predizione . . . . .	72
7.1	Scenario per la valutazione dell'algoritmo . . . . .	75
7.2	Primo test sui parametri prima iterazione . . . . .	78
7.3	Primo test sui parametri seconda iterazione . . . . .	78
7.4	Primo test sui parametri terza iterazione . . . . .	79
7.5	Primo test sui parametri quarta iterazione . . . . .	79
7.6	Primo test sui parametri quinta iterazione . . . . .	80
7.7	Primo test sui parametri risultati finali . . . . .	80

7.8	Secondo test sui parametri risultati finali . . . . .	81
7.9	Terzo test sui parametri risultati finali . . . . .	82
7.10	Quarto test sui parametri risultati finali . . . . .	82
7.11	Quinto test sui parametri risultati finali . . . . .	83
7.12	Test sul parametro sigma . . . . .	84
7.13	Test lavastoviglie regolare considerando un periodo di training di 11 giorni . . . . .	87
7.14	Test lavastoviglie ON OFF considerando un periodo di training di 11 giorni . . . . .	87
7.15	Test lavastoviglie con eccezioni casuali considerando un periodo di training di 11 giorni . . . . .	88
7.16	Test lavastoviglie con cambio abitudini a metà anno considerando un periodo di training di 11 giorni . . . . .	88
7.17	test lavastoviglie con periodo di vacanza considerando un periodo di training di 11 giorni . . . . .	89
7.18	Test lavastoviglie regolare considerando un periodo di training di 21 giorni . . . . .	90
7.19	Test lavastoviglie ON OFF considerando un periodo di training di 21 giorni . . . . .	90
7.20	Test lavastoviglie con eccezioni casuali considerando un periodo di training di 21 giorni . . . . .	91
7.21	Test lavastoviglie con cambio abitudini a metà anno considerando un periodo di training di 21 giorni . . . . .	91
7.22	Test lavastoviglie con periodo di vacanza considerando un periodo di training di 21 giorni . . . . .	92
7.23	Test lavastoviglie regolare considerando un periodo di training di 28 giorni . . . . .	93

7.24	Test lavastoviglie ON OFF considerando un periodo di training di 28 giorni . . . . .	93
7.25	Test lavastoviglie con eccezioni casuali considerando un periodo di training di 28 giorni . . . . .	94
7.26	Test lavastoviglie con cambio abitudini a metà anno considerando un periodo di training di 28 giorni . . . . .	94
7.27	Test lavastoviglie con periodo di vacanza considerando un periodo di training di 28 giorni . . . . .	95
7.28	Test lavastoviglie regolare considerando un periodo di training di 45 giorni . . . . .	96
7.29	Test lavastoviglie ON OFF considerando un periodo di training di 45 giorni . . . . .	96
7.30	Test lavastoviglie con eccezioni casuali considerando un periodo di training di 45 giorni . . . . .	97
7.31	Test lavastoviglie con cambio abitudini a metà anno considerando un periodo di training di 45 giorni . . . . .	97
7.32	Test lavastoviglie con periodo di vacanza considerando un periodo di training di 45 giorni . . . . .	98
7.33	Test lavastoviglie regolare considerando un periodo di training di 60 giorni . . . . .	99
7.34	Test lavastoviglie ON OFF considerando un periodo di training di 60 giorni . . . . .	99
7.35	Test lavastoviglie con eccezioni casuali considerando un periodo di training di 60 giorni . . . . .	100
7.36	Test lavastoviglie con cambio abitudini a metà anno considerando un periodo di training di 60 giorni . . . . .	100
7.37	Test lavastoviglie con periodo di vacanza considerando un periodo di training di 60 giorni . . . . .	101

7.38	Test lavastoviglie regolare . . . . .	103
7.39	Test lavastoviglie ON OFF . . . . .	104
7.40	Test lavastoviglie con eccezioni casuali . . . . .	105
7.41	Test lavastoviglie con cambio abitudini a metà anno . . . . .	106
7.42	Test lavastoviglie con periodo di vacanza . . . . .	107
7.43	Mappa laboratorio . . . . .	112
7.44	Consumo giornaliero del forno . . . . .	113
7.45	Consumo giornaliero del frigorifero . . . . .	113
7.46	Consumo giornaliero della televisione . . . . .	114
7.47	Consumo giornaliero del monitor . . . . .	114
7.48	Consumo complessivo giornaliero . . . . .	116
7.49	Risultati sperimentali del monitor . . . . .	117
7.50	Risultati sperimentali del forno . . . . .	118
7.51	Risultati sperimentali della televisione . . . . .	118

# Elenco delle tabelle

1.1	Caratteristiche delle WSN . . . . .	19
7.1	Valori testati per parametro di clustering e di filtro . . . . .	76
7.2	Risultati dei test sul periodo di training della lavastoviglie . . . . .	102
7.3	Risultati dei test sulla lavastoviglie . . . . .	108
7.4	Risultati dei test sul forno . . . . .	109
7.5	Risultati dei test sul condizionatore . . . . .	110
7.6	Risultati dei test sperimentali . . . . .	119

# Sommario

Ogni luogo in cui l'uomo svolge la sua attività giornaliera, necessita di energia. L'ambito domestico, in particolare, è tra i principali responsabili del consumo energetico mondiale: viene consumata energia ogni volta che si accende la televisione, una lampadina o il riscaldamento. Per rendere disponibile questa energia vengono bruciate ingenti quantità di combustibili fossili, provocando l'emissione in atmosfera di gas ad effetto serra. Parte di questi consumi sono però facilmente evitabili semplicemente correggendo alcuni comportamenti sbagliati.

Questo lavoro si pone come obiettivo principale la sensibilizzazione degli utenti riguardo i loro consumi energetici fornendo un'applicazione per smartphone, *Energy Coach*. Tramite essa l'utente è in grado di monitorare e controllare in qualsiasi momento i consumi e i costi di ogni elettrodomestico presente nell'abitazione e, quando ritenuto opportuno, ricevere messaggi contenenti consigli su come utilizzare al meglio gli elettrodomestici, evitando per esempio di lasciare in stand-by i dispositivi non utilizzati. Tramite altri messaggi, è possibile inoltre, suggerire l'impiego dell'energia prodotta dai pannelli solari, quando disponibile, per svolgere le attività più dispendiose dal punto di vista energetico riducendo in questo modo anche le emissioni di  $CO_2$ .

Il sistema che proponiamo si basa sull'utilizzo di una rete di sensori wireless in grado di monitorare in tempo reale i comportamenti degli utenti raccogliendo i dati relativi ai consumi dei singoli elettrodomestici. Tramite un algoritmo in grado di analizzare i valori raccolti, è poi possibile profilare le abitudini degli utenti e le loro preferenze sulla base di come interagiscono con il sistema.



# Abstract

Every place in which man carries out its daily activities needs energy. The home environment, in particular, is the main agent of global energy consumption: energy is consumed every time someone turns on the television, a light bulb or heating. To provide this energy large quantities of fossil fuel are burned, causing the emission of greenhouse gases. However a large part of this energy consumption can be easily avoided by correcting some wrong behavior.

This work aims at raising awareness among people about their energy consumption by providing an application for smartphone called *Energy Coach*. Through it the user is able to monitor and control at any time energy consumption and costs of each home appliance and receive appropriated messages containing advices on how best to use the appliances, avoiding for example to leave in stand-by not used devices. Through other notifications, the system is able to suggest to use the available energy produced by solar panels, to carry out activities that are more expensive in terms of energy, thereby even reducing  $CO_2$  emissions.

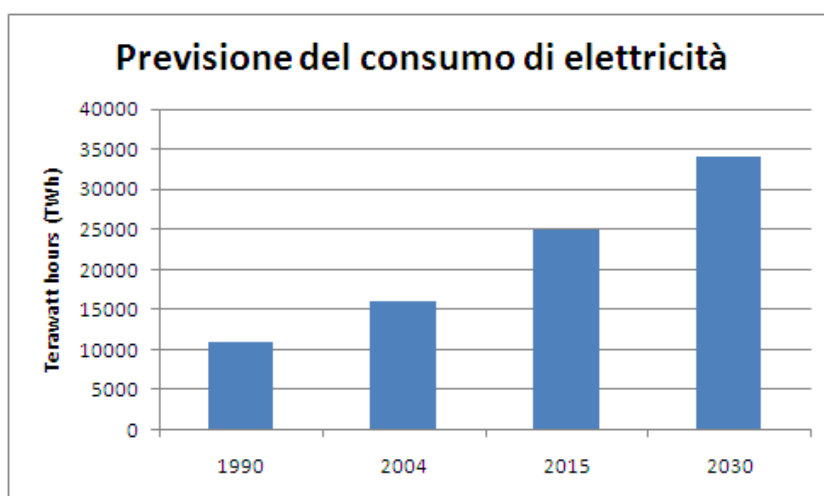
The system that we propose is based on a wireless sensor network to monitor real-time user behavior by collecting data on the consumption of individual appliances. Finally, we have designed and implemented an algorithm to analyze the data collected, in order to profile user behavior and their preferences on the basis of how they interact with the system.

# Introduzione

Sebbene il comfort e la sicurezza siano probabilmente i criteri di scelta più importanti in una casa, l'efficienza energetica e la capacità di ridurre l'impatto ambientale degli immobili stanno diventando sempre più importanti. La crescente domanda di energia, la riduzione dei combustibili fossili e la necessità di ridurre le emissioni di biossido di carbonio, hanno portato in primo piano l'importanza del risparmio e dell'efficienza energetica. Questo argomento, sia in Europa che nel resto del mondo, sta assumendo sempre maggior rilievo nelle politiche dei governi, nei programmi industriali e nella ricerca scientifica. Nel piano d'azione per l'efficienza energetica, infatti, la Commissione Europea ha posto come obiettivi da raggiungere entro il 2020 la riduzione di almeno del 20% dei gas ad effetto serra, la riduzione del 20% del consumo energetico e l'aumento del 20% dell'utilizzo di fonti di energia rinnovabili. Come mostrato nella Figura 1, quindi, senza interventi adeguati, nel 2030 il consumo mondiale di energia risulterà raddoppiato rispetto al 1990.

Il *settore industriale* assume già un ruolo chiave nella riduzione dei consumi energetici. Sia in relazione ai processi produttivi, sia nel sistema edificio-impianto, infatti, il risparmio energetico è stimolato da diversi fattori:

- un urgente bisogno di migliorare l'efficienza nella produzione di beni;
- nuove regolamentazioni.



**Figura 1:** *Previsione del consumo di elettricità*

Il settore residenziale, invece, ha un consumo energetico in notevole crescita, attualmente pari al 27.5% del totale. In particolare, da un'analisi media del consumo domestico, la spesa energetica risulta ripartita nel modo seguente: riscaldamento 26.4%, condizionamento 13%, riscaldamento acqua calda 12.5% e illuminazione 11.6%. In questo settore esistono però oggettive difficoltà a incentivare il risparmio a causa di:

- scarsa consapevolezza del problema: i consumatori sono consapevoli dei propri consumi energetici meramente attraverso la bolletta;
- mancanza di strumenti per la misura dei consumi delle attività quotidiane;
- mancanza di strumenti di controllo.

In tale ambito, tuttavia, si è registrato ultimamente un aumento dell'interesse verso queste problematiche, dettato dalla necessità di ridurre la spesa energetica e dalla realizzazione di sistemi di *Home Automation* che permettono il controllo e la visualizzazione dei consumi in tempo reale.

## Obiettivo della tesi

Questo lavoro di tesi ha come obiettivo la progettazione e lo sviluppo di un'architettura per la gestione del consumo energetico dei dispositivi domestici attraverso un sistema intelligente basato su una rete domestica. A tal fine considereremo tre aree principali di lavoro:

- realizzazione di una rete di sensori per la raccolta dei dati e il controllo degli apparati di consumo. Le reti di sensori e di controllo possono consentire nuovi meccanismi di interazione con i dispositivi che consumano energia, fornendo da un lato strumenti per il monitoraggio continuo dei consumi volti all'aumento della consapevolezza negli utilizzatori, e dall'altro dei sistemi intelligenti per il controllo remoto di tali apparati che riducano sensibilmente gli sprechi senza diminuire la qualità percepita dagli utenti;
- progettazione di un algoritmo che elabori i dati raccolti in modo da apprendere le abitudini dell'utente e fare predizione sull'utilizzo futuro dei dispositivi; L'efficienza e le prestazioni dell'algoritmo proposto saranno valutate tramite una fase di simulazione e sperimentazione.
- implementazione di un'applicazione in Android, *Energy Coach*, che permette il controllo dei dispositivi domestici e la visualizzazione dei risultati dell'algoritmo. Gli utenti potranno tenersi aggiornati in tempo reale sui propri consumi energetici e sui relativi costi. Il dispositivo, dotato di touch-screen, sarà in grado di connettersi in modalità wireless ai sensori applicati agli elettrodomestici presenti nell'abitazione per raccogliere le informazioni necessarie, e per programmare le modalità di utilizzo degli elettrodomestici.

L'apporto innovativo di questo lavoro di tesi consiste nel fatto che l'applicazione non richiederà interazione da parte dell'utente per la rilevazione dei consumi, che verranno direttamente monitorati e quantificati attraverso la rete di sensori, e sarà altresì in grado di identificare chi ha attivato l'elettrodomestico attribuendogli automaticamente il relativo consumo e costo. Tale applicazione, inoltre, suggerirà all'utente sia gli orari migliori per eseguire le attività più impegnative sotto il profilo del consumo energetico, come ad esempio l'asciugatrice o la lavastoviglie, sia la disponibilità d'utilizzo dei pannelli solari. In questo modo è incentivato l'impiego di fonti di energia rinnovabili in ambito domestico, al fine di ottenere significative riduzioni di consumo energetico e allo stesso tempo ridurre le emissioni di  $CO_2$ .

Gli utenti, infine, avranno la possibilità di confrontare i loro consumi con quelli degli altri; questo tipo di paragoni potrebbe generare maggiori risparmi andando a stimolare, in alcuni casi, migliori abitudini "energetiche".

L'applicazione diventerà così una sorta di consulente energetico, fornendo dati sull'andamento dei consumi e suggerimenti per un utilizzo ottimale dell'energia.

## Piano della tesi

La tesi è suddivisa nel seguente modo:

- *Capitolo 1*: descrive i sistemi di automazione domestica attualmente esistenti, gli algoritmi di predizione analizzati e le principali applicazioni basate su reti di sensori sviluppate in questo contesto;
- *Capitolo 2*: descrive l'architettura dell'applicazione e l'interazione tra i vari componenti;
- *Capitolo 3*: presenta la rete di sensori realizzata e la procedura di raccolta dei dati;

- *Capitolo 4*: illustra in dettaglio l'algoritmo sviluppato per l'apprendimento e la predizione dell'utilizzo dei dispositivi domestici;
- *Capitolo 5*: descrive la struttura e le funzioni del server;
- *Capitolo 6*: mostra l'applicazione *Energy Coach*, implementata in Android, per visualizzare costi e consumi, controllare i dispositivi e fornire suggerimenti all'utente volti al risparmio energetico;
- *Capitolo 7*: descrive i test svolti per valutare l'efficienza e la performance dell'algoritmo di predizione, sia sulla base di simulazioni che di una fase sperimentale;
- *Capitolo 8*: presenta alcune considerazioni finali sul lavoro svolto e su possibili sviluppi futuri.

# Capitolo 1

## Automazione domestica e Risparmio energetico

### 1.1 Home automation

Recentemente molti laboratori si sono occupati attivamente delle problematiche riguardanti il mondo dell'Informatica e delle Telecomunicazioni applicate alla Domotica, sia per quanto riguarda l'aspetto di ricerca che quello applicativo [1]. Oggigiorno, infatti, la domotica si è affermata come neologismo rappresentativo di un approccio moderno di governare la casa sfruttando i più recenti strumenti tecnologici. L'adozione di un sistema integrato, che permette di far convergere in un unico sistema di controllo tutti gli aspetti di funzionamento di un'abitazione, ha lo scopo principale di fornire ai proprietari un livello di comfort superiore alla media. In secondo luogo, una gestione centralizzata abilita nuove possibilità di controllo dei consumi, permettendo di ridurre gli sprechi e di ottimizzare le risorse disponibili.

La domotica può essere suddivisa in diverse specializzazioni: la *Home Automation*, rivolta all'automazione della casa intesa come singola unità abitativa, e la *Building Automation*, rivolta all'automazione delle funzioni di un edificio di dimensioni maggiori ad uso lavorativo (industriale, amministrativo, commerciale, ecc..). A differenza del mercato del *Home Automation* che oggi si trova in

espansione, il mercato della *Building Automation* è un mercato già consolidato da diversi anni ed ha già sviluppato un'offerta articolata e ben recepita dagli utenti, tramite progettisti e installatori specializzati in questo settore.

L'*Home Automation*, dopo aver monitorato le abitudini dell'utente, si occupa di automatizzare e ottimizzare, seguendo un'ottica di risparmio energetico, molteplici dispositivi all'interno dell'abitazione domestica, a partire da quelli più semplici da gestire come l'accensione degli irrigatori ad una certa ora della giornata, fino a controllare aspetti più avanzati come il rilevamento della presenza di persone in una certa stanza [2].

Gli impianti di *Home Automation* nascono dallo sviluppo delle soluzioni su larga scala per complessi di uffici e spazi industriali. Tuttavia, oggi la domotica su piccola scala si evolve parallelamente con soluzioni sempre più specifiche e mirate alla vita domestica. Tutti i principali produttori di apparecchi per gli impianti elettrici (ad esempio Bticino [3], Gewiss [4] e Vimar [5]) dispongono di linee di prodotti ad hoc per la realizzazione di soluzioni complete oppure per integrare un impianto già esistente anche di tipo tradizionale. Oltre a questi produttori sono tantissime le aziende che, grazie alla presenza di standard aperti, propongono sul mercato dispositivi per assolvere a funzioni specifiche da inserire in un impianto esistente. Grazie alla diffusione di dispositivi di sempre più facile configurazione e legati in modo stretto anche al mondo informatico è possibile anche per l'utente finale implementare da solo alcune funzioni domotiche di base [6].

Nell'ottica di aumentare significativamente il livello di consapevolezza dei consumi energetici delle famiglie un sistema automatico di lettura dei consumi domestici e di controllo dei dispositivi è in grado di fornire informazioni dettagliate riguardo ai consumi elettrici, [7]. Recenti studi [8] riportano la possibilità di risparmiare fino al 15% a fronte semplicemente di un consumo più consapevo-



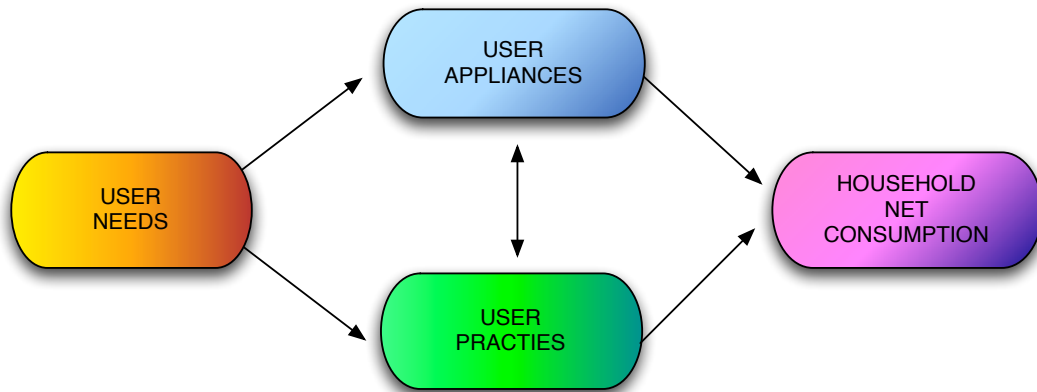
le e controllato. La progettazione di metodi di energy feedback risulta tuttavia non banale in quanto può avvenire solo attraverso sistemi e prodotti in grado di soddisfare le necessità dell'utente [9].

Nello scenario tradizionale, i consumatori hanno la consapevolezza dei propri consumi meramente attraverso il conto presente sulla bolletta che ricevono a casa. Abitualmente, le persone sono legate a credenze scorrette (ad esempio accendere e spegnere le lampade fluorescenti ogni volta che si esce dalla stanza seppur per un breve lasso temporale) oppure non sono a conoscenza delle metodologie di corretto utilizzo di un dispositivo elettrico. I consumatori, inoltre, non dispongono di informazioni riguardanti sia l'impatto di lungo termine che i loro comportamenti hanno sui processi di produzione dell'energia (ad esempio l'energia generata da un dispositivo può essere utilizzata da un altro), sia i consumi "invisibili" prodotti dalle apparecchiature in stand-by. Pertanto, la percezione degli utenti nei confronti dei consumi energetici risulta legata solo ed esclusivamente al tempo di utilizzo dei dispositivi.

In generale è possibile classificare gli utenti nelle seguenti tre categorie: saggi, detective, giudici. Essi dimostrano atteggiamenti differenti nei confronti dei dispositivi elettrici, essendo interessati ad apprendere, rilevare, o valutare i consumi energetici. Queste diversità indicano che l'utente può richiedere funzionalità differenziate da parte del sistema di energy feedback. Per fornire a tutte le tipologie di utenti informazioni adeguate è possibile utilizzare l'architettura, rappresentata nella Figura 1.1, composta da quattro componenti principali:

- *user needs*: mira ad individuare quali sono i requisiti e le aspettative dell'utente;
- *user appliances*: memorizza gli elettrodomestici presenti nell'abitazione;
- *user practices*: analizza comportamenti e abitudini dell'utente registrando l'interazione con i dispositivi domestici utilizzati;

- *household net consumption*: in base all'output dei livelli precedenti viene calcolato il consumo energetico degli elettrodomestici utilizzati dall'utente.



**Figura 1.1:** Modello di un sistema di energy management

E' importante sottolineare come il consumo energetico totale non sia determinato esclusivamente dalla scelta del dispositivo domestico, ma anche dal modo in cui esso viene utilizzato. In questo senso, una soluzione importante per aumentare la consapevolezza degli utenti, in modo da modificare le abitudini dei consumatori, consiste nel fornire feedback efficaci riguardanti i consumi energetici. Questi sistemi, infatti, non possono essere costituiti esclusivamente dalle stesse informazioni sui consumi energetici che sono abitualmente offerte agli utenti e che spesso vengono ignorate, quali i kWh, che non sono immediatamente traducibili in una soluzione per migliorare i consumi. Il feedback, invece, deve essere progettato su misura per i consumi effettivi delle famiglie [10] e deve essere il più vicino possibile alle azioni dell'utente.

Le osservazioni presenti in letteratura [11] evidenziano che l'utente sente il bisogno di capire quali azioni colmerebbero il gap tra lo stato attuale delle cose e gli obiettivi di risparmio energetico prefissati. Un feedback sul consumo energetico totale è sicuramente necessario, in quanto, potrebbe scoraggiare il cosiddetto effetto "rimbalzo", dove il risparmio ottenuto con un dispositivo è vanificato da

un uso scorretto di altri dispositivi e quindi l'adozione di pratiche di risparmio energetico non portano in realtà ad una diminuzione del consumo complessivo di energia. Oltre a tale informazione sui consumi globali è inoltre fondamentale fornire informazioni riguardanti i consumi di ciascun elettrodomestico sia nel breve che nel lungo termine.

Tutti questi aspetti possono essere realizzati attraverso un'infrastruttura informatica: sensori e algoritmi in grado di fornire feedback su dispositivi specifici, nonché sul nucleo familiare.

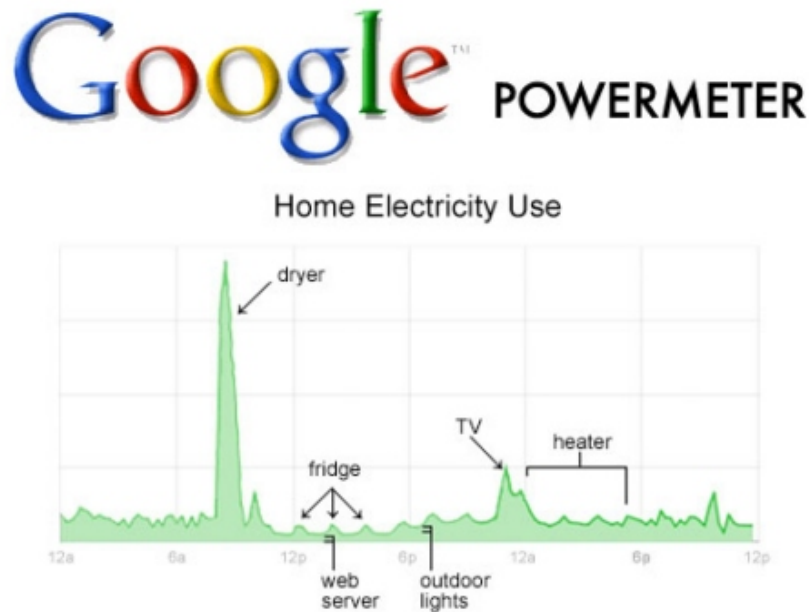
## 1.2 Monitoraggio dell'energia

Nel corso degli ultimi anni, nel campo della domotica e del risparmio energetico sono stati sviluppati numerosi strumenti per aumentare la consapevolezza degli utenti riguardo ai consumi domestici in un'ottica di risparmio energetico.

### 1.2.1 Google Powermeter

L'innovazione di maggior diffusione riguarda il servizio creato da Google chiamato *Google Powermeter* [12].

Come illustrato nella Figura 1.2 si tratta di un'applicazione per la pagina di iGoogle che permette di monitorare i consumi di energia domestici, consentendo quindi all'utente di individuare le fasi della giornata con maggior consumo e quali sono gli elettrodomestici responsabili di un'eccessivo dispendio di energia; il tutto è finalizzato ad aumentare la consapevolezza degli utenti riguardo ai consumi energetici. Il servizio *Google Powermeter* si compone di tre sottosistemi: il primo, completamente hardware, richiede l'inserimento di uno strumento di misura tra la sorgente di energia elettrica e un utilizzatore; il secondo si occupa della raccolta dei dati provenienti dagli strumenti di misura e della comunicazione via Internet dei dati rilevati, mentre il terzo sottosistema è costituito da un servizio di rete in grado sia di popolare un database, con i dati relativi alle misure provenienti via Internet



**Figura 1.2:** *Google Powermeter*

da tutte le abitazioni del mondo, sia della successiva elaborazione e visualizzazione sotto forma di applicazione web. Google PowerMeter fornisce un supporto diretto solamente a quest'ultimo sottosistema, interagendo con i produttori di dispositivi per gli altri due sottosistemi.

### 1.2.2 Control4

Control4 [13] è uno dei primi sistemi di home automation disponibili anche per iPad per controllare in modo intelligente qualsiasi dispositivo in casa e visualizzare il relativo consumo energetico. Molti degli utenti usano dispositivi touch-screen per accedere al sistema Control4, per cui il nuovo iPad rappresenta un'eccellente piattaforma capace di incontrare le diverse esigenze legate proprio a questi compiti (Figura 1.3). Questa nuova applicazione, sviluppata in collaborazione con Control UI, trasforma il nuovo iPad in un'interfaccia wireless per il controllo del sistema My Home, consentendo la creazione di lighting-scene personalizzate, il controllo delle apparecchiature audio/video, dei sistemi di illuminazione, temperatura, web-



Figura 1.3: Control4

cam, ecc. Inoltre, prossimamente, i possessori di dispositivi Control4 avranno la possibilità di controllare il sistema in congiunzione alla sottoscrizione Control4 4Sight Remote Monitoring, per la gestione in remoto attraverso Internet.

### 1.2.3 My Hohm Center

Hohm è un'applicazione web sviluppata da Microsoft che fornisce gratuitamente agli iscritti (per il momento in USA) una rappresentazione sintetica (dashboard) ed analitica dei consumi e dell'efficienza energetica della loro abitazione sulla base di informazioni (CAP, numero di persone ed area dell'appartamento, tipologie di elettrodomestici, sistema di riscaldamento, consumi, etc.) che possono essere fornite dagli utenti rispondendo a dei questionari e/o acquisendoli direttamente dai distributori di energia che stanno aderendo al servizio (Figura 1.4).

Tale servizio è dedicato alla gestione ed ottimizzazione dei consumi energetici domestici, per aiutare i consumatori ad individuare, misurare ed adottare le migliori strategie per limitare il consumo di energia elettrica e gas e facilitare la collaborazione tra fornitori di dispositivi di misurazione puntuale dei consumi degli



Figura 1.4: My Hohm Center

elettrodomestici, imprese di distribuzione energetica (gestione proattiva dei picchi di carico) e consumatori per conseguire un obiettivo comune di riduzione dei costi e dell'impatto ambientale (effetto serra ed inquinamento). Con questo sistema è possibile sapere esattamente quanta energia si sta consumando nell'abitazione, con un elevato livello di precisione e molti dettagli aggiuntivi.

Il sistema è stato concepito come un'applicazione cloud, basata sulla piattaforma Azure, così da consentire agli utenti l'accesso ai dati sui loro consumi da qualsiasi browser collegato alla rete, garantendo privacy e sicurezza con HealthVault. Il portale integra al suo interno anche il nuovo motore di ricerca Bing, utile per approfondire le tematiche legate al risparmio energetico. I consigli forniti agli utenti da Hohm sono basati sui modelli matematici e sui sistemi licenziati dal Lawrence Berkeley National Laboratory, che da anni si occupa della raccolta e dell'analisi dei dati per l'ottimizzazione dei consumi energetici.

Nello specifico tale sistema si compone di 2 diversi elementi: il PowerCost Monitor e il WiFi Gateway. Tali dispositivi possono essere collegati alla rete elettrica in modo da trasferire i dati sul consumo energetico al PC, consentendo agli utenti di leggere le informazioni qualunque sia il dispositivo abilitato a connettersi ad Internet (computer oppure smartphone). Tutto questo si traduce nella possibilità, per l'utente, di conoscere in profondità i dati del consumo elettrico per tutta la giornata, potendolo anche confrontare con i vari periodi dell'anno.

#### 1.2.4 MavHome

Il progetto MavHome (Managing An Intelligent Versatile HOME) [14] sviluppato dall'università del Texas ad Arlington, ha come idea di base quella di percepire lo stato della casa attraverso sensori, creando un ambiente che si comporta come un agente intelligente. Lo scopo dell'agente è massimizzare il comfort e la produttività dei suoi abitanti e allo stesso tempo minimizzare il costo operativo. Per ottenere questi obiettivi, la casa deve essere in grado di prevedere le azioni dei suoi abitanti.

Il sistema MavHome si può suddividere in una moltitudine di agenti di basso livello, ognuno dei quali gestisce delle azioni domestiche. Si costituisce quindi una gerarchia che sarà riorganizzata dinamicamente da MavHome, rendendo questo tipo di soluzione adatta nel caso in cui l'ambiente da controllare sia di dimensioni elevate (Figura 1.5).

Le tecnologie a disposizione di ciascun agente si articolano su quattro livelli:

- Livello Fisico: contiene l'hardware della casa, tra cui i vari dispositivi, i trasduttori e la rete di raccolta dati;
- Livello Informazione: raccoglie, immagazzina e genera le informazioni utili per le decisioni da prendere;
- Livello Decisione: seleziona azioni da eseguire, basate su informazioni fornite dagli altri livelli tramite il livello Informazione;

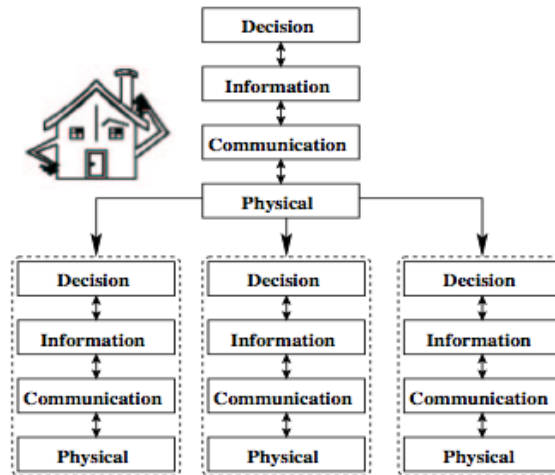


Figura 1.5: Architettura di MavHome

- Livello Comunicazione: facilita le comunicazioni di informazioni, richieste e domande tra gli agenti.

L'architettura si basa su due algoritmi che rivestono un ruolo principale in un ambiente automatizzato e adattabile come MavHome. Il primo algoritmo prevede le azioni dell'abitante della casa, mentre il secondo impara la politica di controllo della casa. L'uso dei due algoritmi rende MavHome in grado di controllare la casa basandosi sulla previsione di eventi, ma allo stesso tempo incapace di riscontrare le azioni che si verificano quando ci sono più abitanti contemporaneamente.

### 1.3 Reti di sensori wireless

Una rete di sensori wireless è un insieme di nodi dotati di CPU a basse prestazioni, basso quantitativo di memoria, radiotrasmettitore e sensori che effettuano delle misurazioni dell'ambiente (sensing) e le trasmettono ad un punto di raccolta, il quale poi le inoltra ad un sistema di elaborazione remoto. La caratteristica di questi nodi è che sono piccoli, consumano pochissima energia, sono collegati tra di loro e costano poco. La rete che formano è autoconfigurante ed autonoma, non richiede nessun intervento dell'uomo, può arrivare a coprire un'area molto estesa



e può rimanere attiva per molto tempo. Una soluzione tramite WSN consente di risparmiare parecchio sui costi, monitorare aree molto più estese, con condizioni ambientali estreme e in punti molto ristretti e difficili da raggiungere. Inoltre, dato il basso costo di ogni nodo, sono utilizzabili in situazioni di emergenza, dove è possibile che vengano distrutti o persi in qualche modo, come ad esempio in aree a pericolo di incendio o di frane. Le possibilità di utilizzo di una Wireless Sensor Network non si limitano solo a questo: possono essere adoperate per fini militari, di sorveglianza, di *Home Automation*, di monitoraggio di parametri biologici e tanti altri.

Per quanto riguarda il campo dell'*Home Automation* uno degli aspetti più critici riguarda l'interazione con l'utente. Studi recenti presenti in letteratura hanno dimostrato che si possono ottenere risparmi significativi solamente adattando dinamicamente il comportamento del sistema alle attività degli utenti. L'interazione con l'utente è quindi fondamentale per acquisire le informazioni necessarie.

Una rete di sensori, in questo campo applicativo, permette quindi un'interazione in modo naturale con l'utente, essendo in grado di ridurre la complessità di configurazione e gestione per mezzo di:

- monitoraggio in tempo reale delle attività degli utenti in casa (identificazione, presenza, uso dei dispositivi, ecc.) e dei parametri fisici (luce, temperatura, ecc.);
- profilazione delle abitudini degli utenti e delle loro preferenze sulla base di come loro interagiscono con il sistema.

Alcuni dei benefici introdotti grazie alle WSN nello sviluppo di sistemi di *Home Automation* possono essere riassunti in:

- definizione di strategie di gestione dell'energia individuali;
- definizione di metodi per il passaggio automatico tra differenti livelli di consumo dei dispositivi, basato sul monitoraggio e profilazione degli utenti;

- valutazione del livello di soddisfazione degli utenti per differenti strategie e livelli di consumo;
- indicazione dei processi quotidiani che consumano più energia e suggerimento di possibili modifiche alle abitudini.

Ogni nodo è in grado di acquisire informazioni e di trasmetterle ad altri nodi nel proprio range di trasmissione (circa 10-20m). Esso è a basso consumo energetico e può essere alimentato sia con batterie che tramite una rete cablata. Infine possono essere utilizzate due diverse topologie di rete (a stella o mesh). La Tabella 1.1 presenta alcune caratteristiche tipiche di una WSN per l'automazione domestica.

Caratteristiche	Valori
Frequenza	2.4 GHz o 900 MHz
Modulazione	PSK o FSK
Sensibilità di ricezione di un nodo	-90 dbm
Range del nodo trasmittente	10-20 m
Indirizzamento	Asseganto dal nodo centrale
Topologia di rete	Mesh o Stella
Rounting	Tabelle di routing
Alimentazione dei nodi	Batteria o cablata

**Tabella 1.1:** *Caratteristiche delle WSN*

### 1.3.1 Il Progetto AIM

Il ruolo chiave svolto dalle reti di sensori wireless, per il controllo automatico degli elettrodomestici in funzione e delle abitudini degli utenti, è messo in evidenza nel progetto europeo AIM [15] [16]. Esso si propone di sviluppare una tecnologia per la profilazione e ottimizzazione del consumo energetico domestico; a tale scopo è stata progettata un'architettura che permette il monitoraggio e la visualizzazione dei consumi energetici in tempo reale [17]. Al fine di creare un sistema in cui l'utente non abbia bisogno di perdere tempo per impostare i parametri di sistema, questo progetto mira inoltre ad automatizzare l'inserimento di tali parametri con un sistema in grado di prevedere le preferenze dell'utente, sulla base di un comportamento osservato in precedenza.

L'architettura si compone di cinque componenti base: *home gateway*, *Energy Management Device* (EMD), la rete domestica, la rete di sensori e i dispositivi domestici che si vogliono monitorare e controllare.

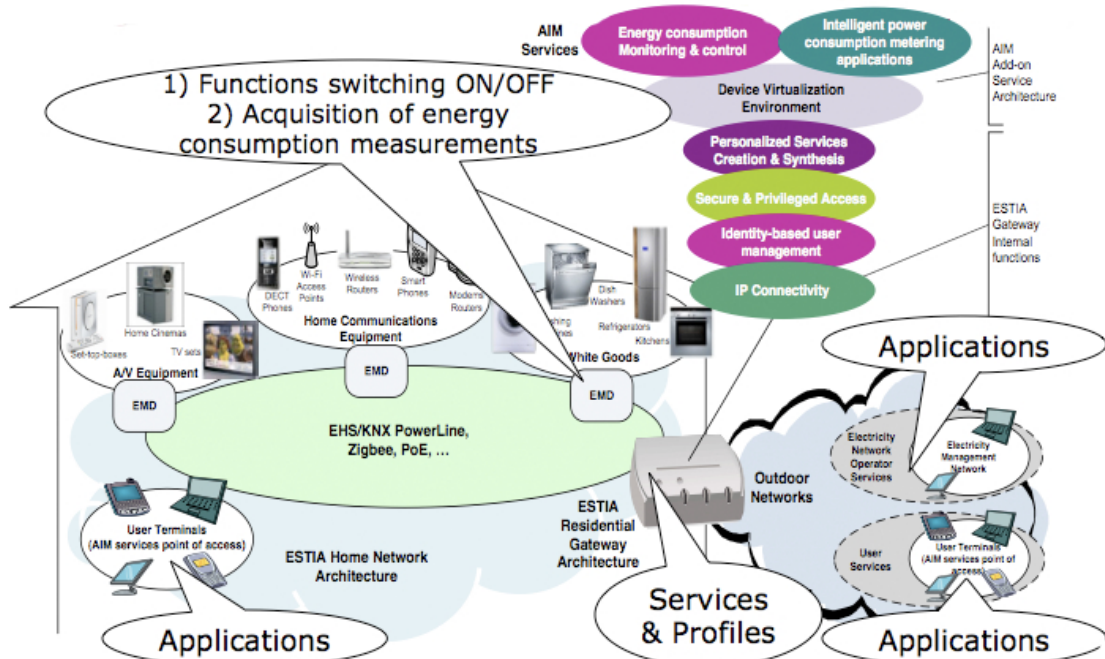


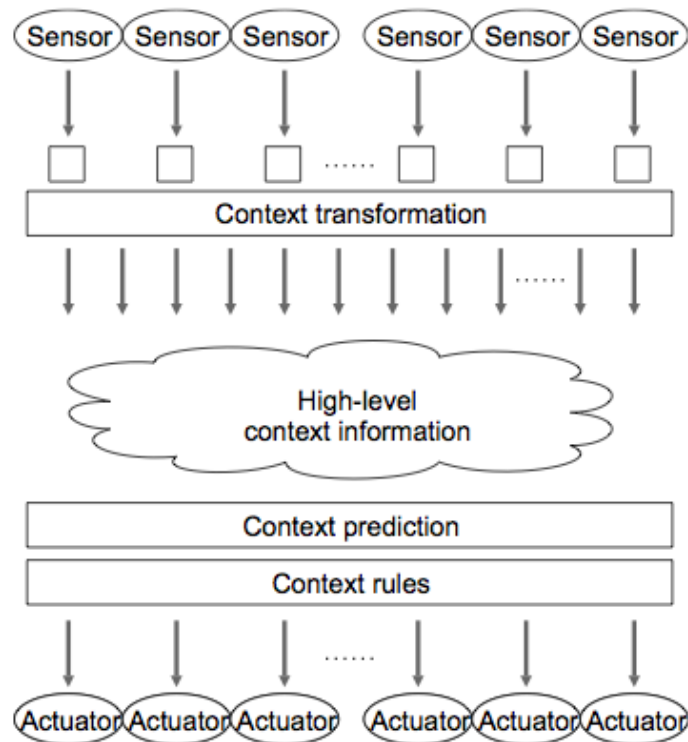
Figura 1.6: Architettura di AIM

Il gateway appare come un elemento costitutivo della logica del sistema AIM, in quanto seleziona e trasporta le informazioni all'interfaccia del corretto dispositivo, applica la logica centralizzata di controllo e assicura una comunicazione criptata nella rete. L'EMD è controllato dal gateway, utilizzando l'interfaccia bus; costituisce invece il fulcro locale del sistema di controllo energetico, sia per le apparecchiature attive che per quelle in stand-by, essendo in grado di monitorare e controllare i device stessi e di fornire un'interfaccia di comunicazione tra la rete domestica e il gateway AIM.

Il ruolo principale della rete di sensori, all'interno del progetto AIM, riguarda l'acquisizione di informazioni sul comportamento dell'utente e la sua interazione con le apparecchiature domestiche. Essa fornisce, inoltre, misure di alcuni parametri fisici, come la temperatura e la luce nella stanza, che possono essere utilizzati dal sistema di gestione dell'energia per eseguire delle modifiche in real time. Sulla base dei dati raccolti il sistema AIM è in grado di pianificare al meglio i tasks per tutti gli apparecchi domestici, per esempio il riscaldamento della stanza alla temperatura desiderata prima che l'utente entri nella stanza. La rete di sensori, inoltre, può essere personalizzata sulla base delle esigenze degli utenti e delle caratteristiche specifiche del contesto, aggiungendo nuovi gruppi di sensori anche in una fase successiva, quando il sistema è già in funzione (Figura 1.6).

## 1.4 Algoritmi di predizione

Il *Context Prediction* è un aspetto importante del Context Computing (ideazione, progettazione e implementazione di un *computer system* che utilizza la *context awareness*) che mira a inferire contesti futuri sulla base dei contesti osservati. L'architettura generale per le applicazioni basate sul contesto è evidenziata dalla Figura 1.7. A partire dai dati rilevati dai sensori vengono derivate, attra-



**Figura 1.7:** Architettura delle applicazioni context-aware

verso opportune trasformazioni, delle informazioni di alto livello, riguardanti la situazione attuale, che sono usate dalle applicazioni per ricavare una predizione del contesto attuale. Le applicazioni, infine, possono usare le informazioni di alto livello ricavate e la predizione formulata per esprimere delle regole che saranno successivamente utilizzate dagli attuatori per agire sull'ambiente. In questo senso, gli algoritmi di apprendimento statistico e il machine learning possono essere

utilizzati per monitorare e apprendere i contesti osservati, al fine di applicare la conoscenza ricavata per predire contesti futuri.

Il *machine learning* rappresenta una forma di adattamento all'ambiente attraverso l'esperienza, con l'obiettivo di migliorare un sistema senza il bisogno del continuo intervento umano. La tecnica generale parte dall'apprendimento di dati di input, sotto forma di esempi, per realizzare sistemi per la sintesi di nuova conoscenza attraverso classificazioni e generalizzazioni. Il suo fine è quello di sfruttare l'enorme potenza di calcolo disponibile nei computer per esplorare tutte le maniere in cui possa avvenire un apprendimento automatico e di realizzarlo attraverso l'utilizzo di metodologie statistiche e computazionali.

La forza del *machine learning* è dovuta alla qualità dei suoi algoritmi, che nel corso degli anni vengono migliorati e aggiornati; questi si dividono in diverse tipologie principali, a seconda del tipo di apprendimento automatico per cui sono realizzati:

- *Apprendimento Supervisionato*: è caratterizzato dalla presenza di un insegnante che conosce l'etichetta o il costo di ogni pattern del training set. Questa conoscenza dell'ambiente fornisce la capacità di restituire l'output desiderato ogni volta che viene ricevuto un input. In questo modo è possibile aggiornare i pesi della struttura che apprende, che può essere ad esempio una rete neurale, in base all'output restituito e all'output conosciuto. Questo processo iterativo di modifica dei pesi permette alla struttura di poter emulare l'insegnante, generalizzando su pattern mai incontrati;
  
- *Apprendimento non Supervisionato*: la struttura di apprendimento, non riceve alcuna informazione sulla correttezza dell'output fornito oppure sulla funzione target che deve essere approssimata: essa costruisce concetti attraverso la sperimentazione nell'ambiente. La stessa struttura deve individuare delle regolarità dai pattern di training e codificarle in risposte. Una volta

individuare queste regolarità, viene acquisita la capacità di generalizzare su pattern mai visti;

- *Apprendimento Semi-Supervisionato*: l'algoritmo fonde le due tecniche sopracitate per generare un'appropriata funzione o un classificatore;
- *Apprendimento con Rinforzo*: come nell'apprendimento supervisionato, la struttura adatta i propri parametri in base al feedback ricevuto dall'ambiente, che anziché essere di tipo istruttivo è di tipo valutativo.

Un sistema di *Home Automation* deve essere in grado di apprendere ed applicare la conoscenza acquisita sulle azioni del nucleo familiare, per soddisfare comfort ed efficienza energetica [18].

Come mostrato nella Figura 1.8, esistono diversi approcci per prevedere l'utilizzo dei singoli dispositivi domestici e il relativo consumo energetico [19]:

- modello di regressione;
- rete neurale;
- alberi di decisione;
- support vector regression.

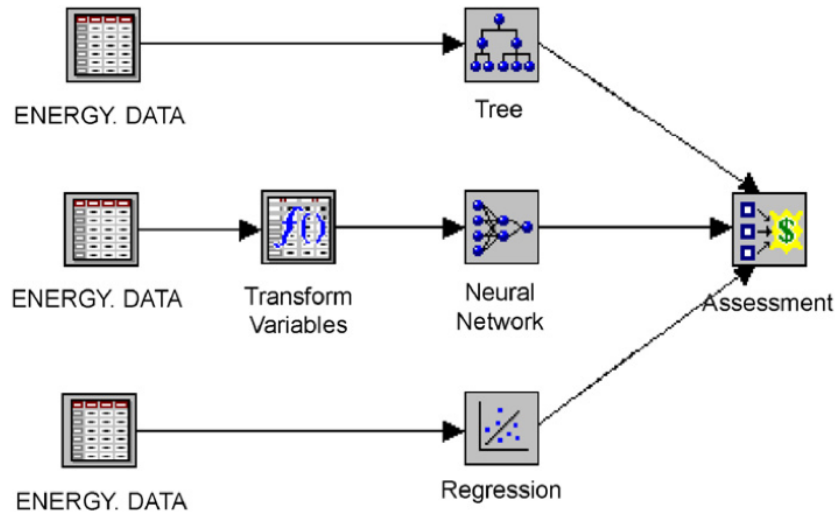


Figura 1.8: Tecniche per predire i consumi elettrici

### 1.4.1 Modello di regressione

Nella realtà quasi mai un evento dipende solamente dall'andamento di un certo fattore. Tutti gli eventi, anche i più comuni, sono infatti influenzati da numerosissimi elementi. Risulta pertanto molto utile formulare un modello che tenga conto di tutte queste influenze, o meglio delle influenze maggiori. Tale effetto può essere ottenuto con lo studio della regressione lineare multipla (o multivariata). In generale si indica con  $y$  la variabile dipendente, e con  $x$  seguito da un numero in pedice le variabili indipendenti chiamate predittori che si suppone abbiano un effetto. La formula generale del modello è la seguente:

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_px_p + \epsilon$$

dove  $\beta_i$  ( $i = 0, 1, 2, \dots, p$ ), rappresentano i parametri di regressione, e  $\epsilon$  rappresenta l'errore casuale del modello. Determinati i coefficienti di regressione, l'equazione può essere utilizzata per prevedere il valore dell'uscita, sulla base di una o più variabili di input indipendenti.



La maggior limitazione concettuale di questa tecnica è data dal fatto che può solo constatare le relazioni esistenti tra input e output, senza rilevare gli elementi casuali del meccanismo (cambiamento di abitudini dell'utente).

### 1.4.2 Reti neurali

Le reti di neuroni artificiali sono uno dei metodi più robusti per l'apprendimento di funzioni obiettivo i cui valori possono essere reali, discreti o vettoriali, basati principalmente sulla simulazione di neuroni artificiali opportunamente collegati. I singoli neuroni vengono interconnessi alla schiera di neuroni successivi, in modo da formare la rete neuronale; normalmente una rete di tale tipo è formata da tre strati: ingresso, che si preoccupa di trattare gli input in modo da adeguarli alle richieste dei neuroni, uno strato nascosto, che si preoccupa dell'elaborazione vera e propria e lo strato d'uscita che raccoglie i risultati e li adatta alle richieste del blocco successivo della rete.

Le *neural networks* manifestano buone prestazioni nelle applicazioni in cui vi è una conoscenza a priori della relazione tra gli ingressi e le uscite. Esse, tuttavia, vanno considerate come delle *black box* ossia, a differenza di un sistema algoritmico, dove si può esaminare passo-passo il percorso che dall'input genera l'output, una rete neurale è in grado di generare un risultato accettabile con un'alta probabilità, ma non è possibile spiegare come e perché tale risultato sia stato generato. Inoltre, oltre a non fornire dei *p-value* per testare l'accuratezza dei parametri stimati, questo approccio necessita di una fase di addestramento del sistema che fissi i pesi dei singoli neuroni, fase che può richiedere molto tempo, se il numero dei record e delle variabili analizzate è molto grande.

### 1.4.3 Alberi di decisione

Si tratta di un metodo basato su un albero di decisione, rappresentativo di una segmentazione di dati creata applicando una serie di semplici regole *if-then-else*. Tali regole possono essere utilizzate per fare predizione attraverso un processo

di splitting ricorsivo. Gli alberi di decisione, generalmente, non hanno le stesse prestazioni delle reti neurali per quanto riguarda dati non lineari, come nel nostro caso, e sono suscettibili al rumore. Questa tecnica, inoltre è adatta per prevedere risultati classificabili, viceversa è poco appropriata per analizzare serie storiche di dati.

#### 1.4.4 Support Vector Regression

Data la presenza, nei sistemi di *Home Automation*, di dati non lineari è stata esaminata l'opportunità di utilizzare un modello non parametrico per interpolare i dati di interesse. Le *Support Vector Regression* (SVR) sono una nuova tecnica, sviluppata da Vapnik [20] nell'ambito della *Statistical Learning Theory*, ed utilizzata per costruire *Support Vector Machines* per fare regressione. Questo recente algoritmo può essere visto come una tecnica di addestramento in grado di ottimizzare la capacità di generalizzazione della struttura risultante, tenendo sotto controllo il fenomeno dell'overfitting.

A partire dai dati di training  $(x_1, y_1), \dots, (x_2, y_2)$  l'obiettivo è identificare una funzione  $f(x)$  che per tutti i campioni di addestramento  $x$  abbia una deviazione massima pari a  $\epsilon$  rispetto agli output reali ottenuti. Rispetto ad altri approcci, in questa tecnica la complessità del modello è limitata dall'algoritmo di apprendimento, attraverso il quale si cerca di evitare l'utilizzo di funzioni complesse mediante opportune trasformazioni dei dati attraverso funzioni di Kernel. Gli svantaggi sono dovuti principalmente dalla dipendenza dei *support vectors* dal Kernel, e dalla scelta del Kernel stesso che è dettata spesso empiricamente.

# Capitolo 2

## Architettura del sistema

Nel presente capitolo viene illustrata l'architettura su cui si basa il sistema per il risparmio energetico che abbiamo realizzato, saranno descritti i componenti che la costituiscono e la loro interconnessione.

### 2.1 Introduzione

In questo lavoro di tesi abbiamo progettato e realizzato un'architettura (Figura 2.1) per il monitoraggio e la previsione dei consumi energetici domestici costituita da quattro componenti comunicanti tra di loro:

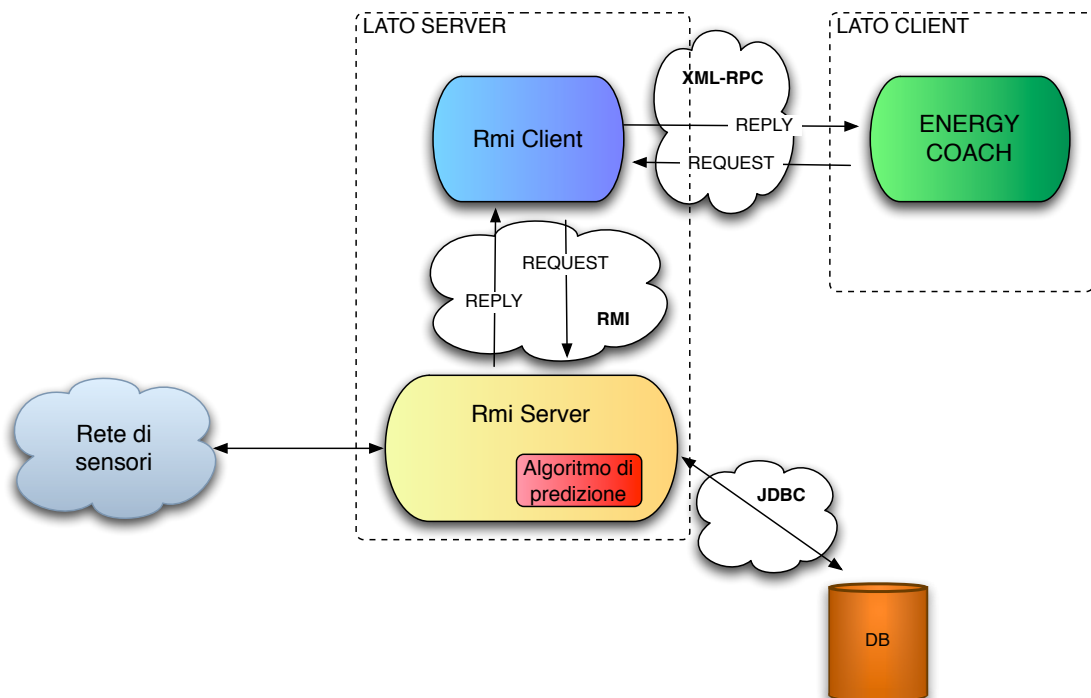
- Rete di sensori;
- Rmi Client;
- Rmi Server;
- Energy Coach.

Ciascuno di tali componenti, all'interno dell'architettura, ha un suo ruolo specifico che descriveremo nei capitoli successivi.

Le funzionalità base fornite dal sistema sono:

- lettura dei consumi energetici degli elettrodomestici rilevati dai sensori;

- raccolta dei dati ed elaborazione degli stessi tramite l'algoritmo di predizione. Tale algoritmo predice per ciascun elettrodomestico l'utilizzo, l'ora di accensione e la durata di funzionamento;
- visualizzazione dei risultati dell'algoritmo per mezzo dell'Energy Coach;
- visualizzazione di costi e consumi dei singoli elettrodomestici;
- controllo e programmazione delle accensioni e degli spegnimenti di ogni dispositivo;
- visualizzazione dei consigli su come ottimizzare l'utilizzo degli apparecchi domestici finalizzato al risparmio energetico.



**Figura 2.1:** Architettura del sistema

Nella progettazione del sistema descritto abbiamo, inoltre, fatto in modo che venissero rispettati i seguenti vincoli:

- *robustezza*: il sistema deve avere la capacità di sopportare situazioni critiche quali interferenze sul canale di comunicazione. Per evitare la perdita dei dati l'uso del canale deve essere regolato al fine di evitare interferenze;
- *basso consumo energetico*: nonostante gli ACme non siano alimentati con batterie è importante che il loro consumo energetico sia molto basso in modo tale da non alterare i dati inviati;
- *capacità di rilevazione della potenza*: il sistema deve essere in grado di effettuare le misurazioni dei consumi in maniera precisa ed affidabile;
- *efficacia di predizione*: il sistema deve essere in grado di predire i comportamenti degli elettrodomestici minimizzando l'errore commesso.

## 2.2 Linguaggi di programmazione

L'applicazione che abbiamo realizzato è basata sul paradigma client-server. Il lato client, in particolare, lo abbiamo sviluppato in Android così come descritto nel Capitolo 6, mentre il lato server lo abbiamo implementato tramite un'applicazione Java. Per facilitare la comunicazione tra client e server abbiamo deciso di dividere il server in due applicazioni distinte, Rmi Client e Rmi Server, in grado di comunicare tra loro tramite gli RMI di Java.

Per la memorizzazione dei dati ci si è avvalsi di un database MySQL. (Per maggiori dettagli si faccia riferimento al Capitolo 5) mentre per la programmazione dei sensori abbiamo scelto il linguaggio nesC utilizzato dal sistema operativo TinyOS.

## 2.3 Rete di sensori

La rete di sensori che abbiamo definito è costituita da due componenti principali: la *base station* e i sensori ACme. Essi sono organizzati in una rete a stella con al centro di essa la *base station*. Gli ACme, in particolare, misurano i consumi degli elettrodomestici e inviano in broadcast i dati raccolti. La *base station*, invece, ha il compito di ricevere e gestire i pacchetti inviati dai sensori (Figura 2.2).

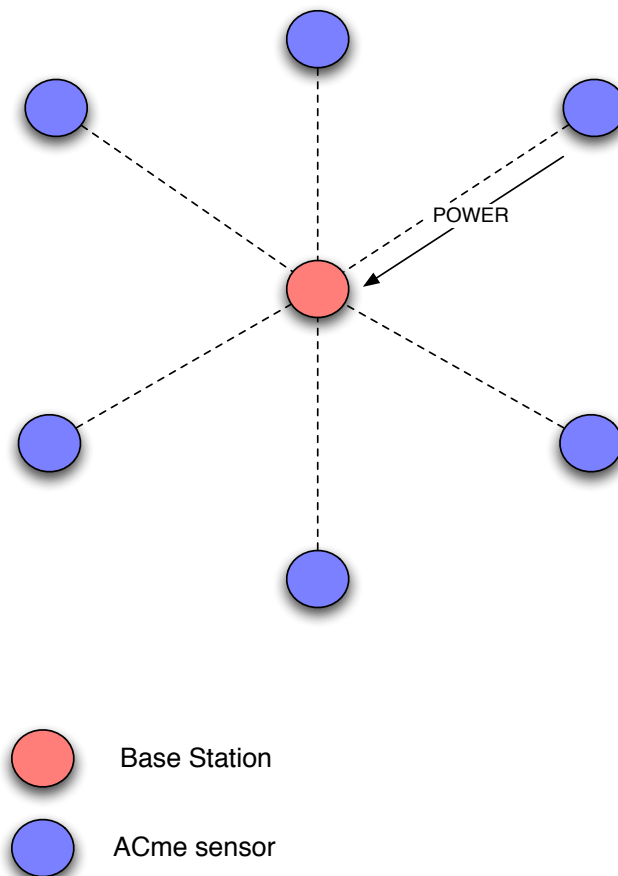


Figura 2.2: Rete di sensori

## 2.4 Rmi Client

Nella realizzazione del sistema proposto abbiamo deciso di implementare il modulo Rmi Client al fine di fornire una serie di primitive per la comunicazione e l'interazione tra Rmi Server e Energy Coach, scaricando l'utente dalla responsabilità di occuparsi manualmente di tutto ciò che si trova sul lato server dell'applicazione.

L'implementazione del Rmi Client avviene mediante due package Java principali (Figura 2.3):

- *client*: include la classe `apache_interface` che istanza un web server e si mette in attesa delle chiamate provenienti dall'Energy Coach. Ricevuta una chiamata invoca il metodo corrispondente presente nella classe `WSNGateway`;
- *interfaces*: include la classe `WSNGateway` che si connette al Rmi Server tramite tecnologia RMI e invoca i metodi presenti su di esso.

Terminata l'esecuzione della chiamata, Rmi Client invia all'Energy Coach i risultati provenienti dal Rmi Server.

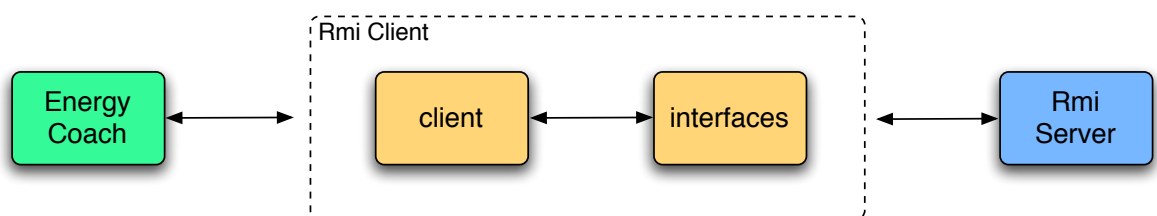


Figura 2.3: Struttura Rmi Client

## 2.5 Rmi Server

Nel sistema proposto abbiamo definito un secondo modulo Rmi, chiamato Rmi Server, con il compito di gestire i dati provenienti dalla rete di sensori, di implementare l'algoritmo di predizione e di interfacciarsi con il database.

L'implementazione di tale modulo è realizzata mediante quattro package Java (Figura 2.4):

- *engine*: è il package principale dell'applicazione: avvia il server, si connette al Rmi Client tramite tecnologia RMI e contiene le classi che implementano l'algoritmo di predizione;
- *interfaces*: pubblica i metodi del server che posso essere invocati dal Rmi Client;
- *mysql*: gestisce le query per l'interazione con il database;
- *power*: si occupa della gestione della *base station* per l'acquisizione e l'elaborazione dei dati provenienti dai sensori.

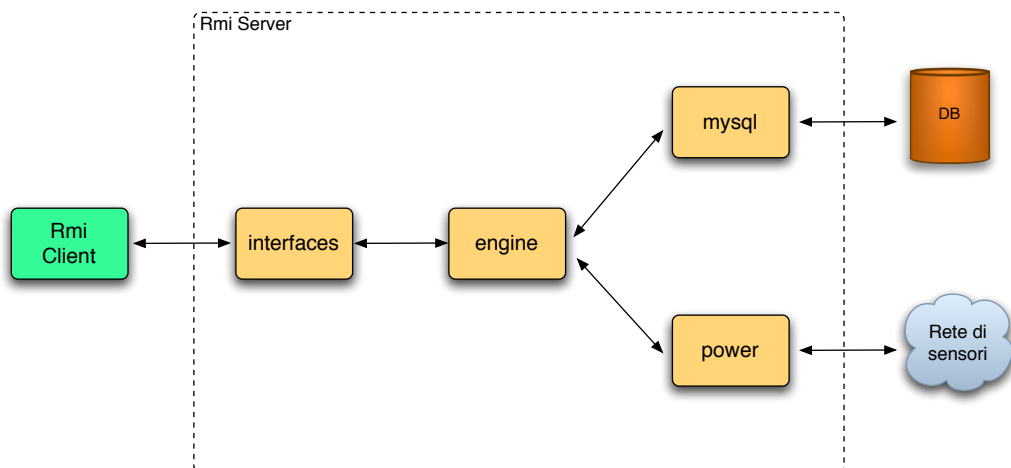


Figura 2.4: Struttura Rmi Server

Per una trattazione più approfondita della struttura e delle funzioni del Rmi Server si rimanda al Capitolo 5.



## 2.6 Energy Coach

È il lato client del sistema e permette all'utente la visualizzazione dei costi e dei consumi degli elettrodomestici e il loro controllo. Mostra, inoltre, graficamente i risultati dell'algoritmo di predizione.

L'implementazione dell'Energy Coach l'abbiamo realizzata definendo quattro package principali (Figura 2.5):

- *Energy Coach*: gestisce la connessione al server tramite Rmi Client e avvia l'applicazione;
- *configuration*: permette la configurazione delle stanze della casa e l'inserimento in esse degli elettrodomestici;
- *activity*: include le funzioni principali quali visualizzazione di consumi e costi, controllo degli elettrodomestici;
- *back*: gestisce tutte le attività in background quali i messaggi riguardanti i consigli per il risparmio energetico.

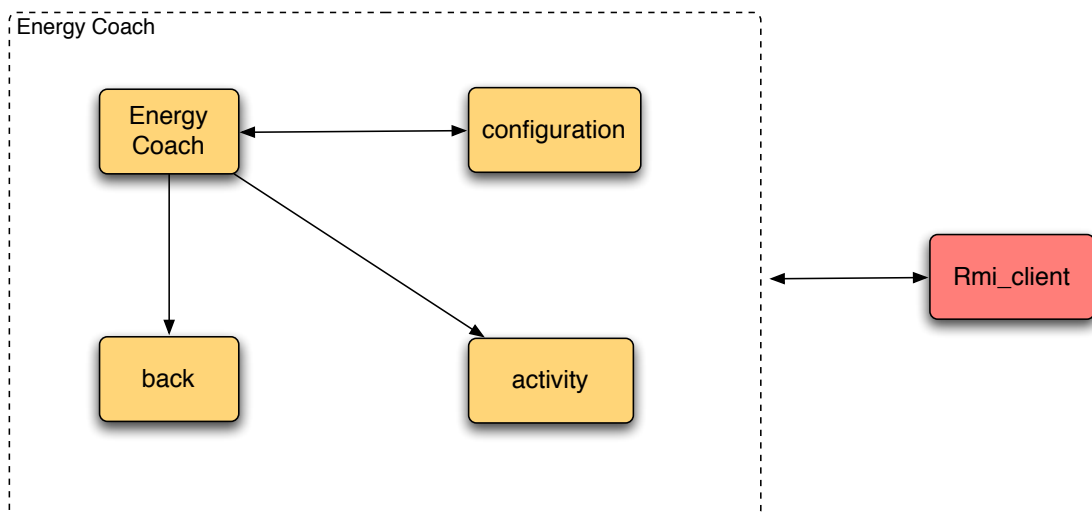


Figura 2.5: Struttura Energy Coach

Per maggiori dettagli si faccia riferimento al Capitolo 6.

# Capitolo 3

## Rete di sensori

### 3.1 Introduzione

L'utilizzo di strumenti per la misurazione dell'energia e della potenza è consolidato in ambito industriale per il controllo dei macchinari di produzione e negli ultimi anni si sta diffondendo anche in ambito domestico. Non-Intrusive Load Monitoring (NILM) [21] è una tecnica per disaggregare i consumi energetici di ogni elettrodomestico presente nell'abitazione a partire da misure aggregate di tensione e corrente. Una delle maggiori limitazioni di questo approccio è la necessità di una fase di training durante la quale gli utenti devono inserire manualmente le accensioni e gli spegnimenti di ogni singolo dispositivo. Studi recenti propongono un approccio per automatizzare la fase di training e raccogliere i consumi di ogni singolo dispositivo basati sulle reti wireless [22]. La scelta delle reti wireless risulta maggiormente conveniente rispetto alle reti cablate, in quanto più difficoltosa e dispendiosa l'aggiunta di nuovo hardware ad una rete preesistente.

Pertanto, in questo lavoro, per la raccolta dati riguardanti i consumi energetici dei singoli elettrodomestici, abbiamo deciso di utilizzare una rete di sensori wireless. Ad ogni elettrodomestico corrisponde un sensore che ne misura il consumo e invia i dati raccolti alla *Base Station*. In questo capitolo verranno presentati i sensori e le piattaforme hardware e software che abbiamo utilizzato per realizzare la rete.

## 3.2 ACme

Per la realizzazione della rete di sensori abbiamo deciso di utilizzare i sensori ACme (AC meter) realizzati presso l'università di Berkeley che permettono la misurazione della potenza e dei consumi degli elettrodomestici e la trasmissione wireless dei dati rilevati [23].

### 3.2.1 Hardware

Come mostrato dalla Figura 3.1 Il sensore ACme è costituito da cinque componenti principali [24] [25]:

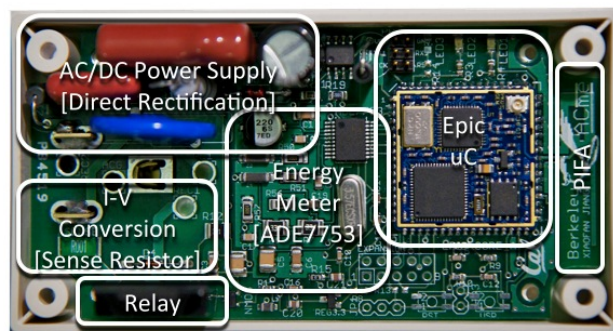


Figura 3.1: Architettura dell'ACme

- *Current-to-Voltage Conversion*: per la conversione corrente-tensione viene utilizzato un sensore *Hall Effect* [26]. Questo approccio permette di isolare le alte tensioni in input dalle basse tensioni in output, consentendo l'isolazione elettrica dei Kilovolts;
- *Energy metering*: permette di calcolare la potenza e l'energia a partire dai valori di corrente e tensione rilevati. Questo processo può essere effettuato sia via software che hardware; l'ACme utilizza la modalità hardware servendosi di un *Analog Device ADE7753* che fornisce la potenza reale, reattiva ed apparente. Internamente integra la potenza per calcolare l'energia, memorizza i valori calcolati in opportuni registri e comunica con il microcontrollore *Epic core* attraverso l'SPI (Serial Peripheral Interface) bus;
- *AC/DC Power Supply*: effettua la conversione della corrente da alternata a continua, fornendo al resto del circuito una fonte di alimentazione stabile DC. Al fine di ridurre i costi e le dimensioni del circuito è stato utilizzato un raddrizzatore a semionda che si collega direttamente alla linea CA, evitando l'impiego di un trasformatore;
- *Microcontroller with radio*: l'ACme utilizza un *Epic Core* come microcontrollore e radio, poiché fornisce tutte le funzionalità di un tipico *mote*, ma risulta essere di dimensioni ridotte e quindi facilmente integrabile nei circuiti. L'Epic core usa il microcontrollore TI MSP430F16 e il modulo radio CC2420, entrambi supportati da TinyOS. Infine, l'antenna PIFA (Planar Inverted F-Antenna) permette buone trasmissioni e minimizza i costi;
- *Solid state AC Relay*: consente il controllo remoto dell'elettrodomestico associato all'ACme ed è controllato direttamente dall'Epic.

### 3.2.2 Calibrazione

Una volta creata la rete, per poter utilizzare i sensori ACme in modo opportuno, abbiamo eseguito una fase di calibrazione [27]. Gli Acme, infatti, non sono in grado di effettuare misure accurate sotto 1W e presentano alcune non linearità. Per poter ovviare a questi errori è stata utilizzata una calibrazione lineare. Come valori di riferimento abbiamo, in particolare, utilizzato i consumi di un televisore inizialmente in stand-by (17W) e successivamente acceso (212W) i quali sono stati misurati sia attraverso gli ACme che mediante un multimetro digitale.

La formula utilizzata per la calibrazione è la seguente:

- *reale\_l* valore rilevato dal multimetro per il carico da 17 W;
- *acme\_l* valore rilevato dall'ACme per il carico da 17 W;
- *reale\_h* valore rilevato dal multimetro per il carico da 212 W;
- *acme\_h* valore rilevato dall'ACme per il carico da 212 W.

$$y = \frac{reale_h - reale_l}{acme_h - acme_l} * x + reale_l - \frac{reale_h - reale_l}{acme_h - acme_l} * acme_l \quad (3.1)$$

Sostituendo i valori si ottiene la seguente equazione:

$$y = 0.013 * x - 0.98 \quad (3.2)$$

I dati raccolti sono stati così interpolati trovando in questo modo la funzione di calibrazione dove  $x$  rappresenta il valore di potenza misurato dell'ACme e  $y$  il consumo reale (misurato in Watt) dell'elettrodomestico.

### 3.2.3 La rete di sensori ACme

Per la comunicazione in una rete con dispositivi a bassa potenza (come gli ACme) è possibile utilizzare *blip* (Berkeley Low-power IP stack), un'implementazione in TinyOS del protocollo IPv6 per i link a basso consumo. Esso permette di creare una rete multihop di *mote* differenti comunicanti attraverso protocolli condivisi. A differenza delle reti di sensori standard in cui i nodi vengono posizionati accuratamente per ottenere una buona connettività, in ambito domestico gli utenti installano gli ACme dietro gli elettrodomestici e quindi la comunicazione con un access point wireless risulta difficile. Per questo viene creata una rete multihop tra i vari sensori.

Ogni ACme assume le funzionalità di un router IP e seleziona le rotte di default per i suoi vicini sulla base di *router advertisements* inviati da tutti i nodi. I nodi più esterni, *edge routers*, raccolgono i pacchetti che ricevono dagli altri nodi e li inviano al server.

Per quanto riguarda il livello di trasporto il protocollo blip supporta sia TCP che UDP. Tuttavia quello maggiormente utilizzato è UDP in quanto la perdita dei pacchetti è minore dell'1% e quindi l'overhead associato agli *acknowledged byte stream* non risulta necessario.

## 3.3 Micaz e Telosb

Per la realizzazione della *base station* abbiamo utilizzato 2 tipi di sensori: *Micaz* e *Telosb*.

I Micaz (Figura 3.2) sono formati da un microcontrollore principale AtMega128L, una memoria Flash da 128 KByte per la programmazione, una memoria Flash seriale da 512 KByte e una memoria EEPROM da 4 KByte. Questi dispositivi, inoltre, contengono un chip Radio CC2420 e sono dotati di un'antenna esterna. I Micaz, infine, sono dotati di un connettore a 51 pin per l'interfacciamento con le schede di programmazione, mentre l'alimentazione è garantita da due batterie di

tipo AA situate in un apposito alloggiamento posto sotto la scheda.



**Figura 3.2:** *Micaz*

I Telosb (Figura 3.3) sono formati da un microcontrollore MSP430F1611, una memoria RAM da 10 KByte e una memoria Flash da 48KByte. Tali sensori sono dotati di un chip Radio CC2420, di un antenna integrata con un range di 50 metri e possono essere alimentati da due batterie di tipo AA o direttamente dalla porta USB connessa a un computer.



**Figura 3.3:** *Telosb*

## 3.4 TinyOS

Per la programmazione dei sensori ACme e della *base station* abbiamo utilizzato il sistema operativo *TinyOS* [28]. Esso risulta essere ottimale per la programmazione di architetture che dispongono di poca memoria, scarsa capacità di elaborazione e fonti di energia limitate. TinyOS presenta infatti le seguenti due caratteristiche che permettono un notevole risparmio di energia:

- non possiede un nucleo, ma permette l'accesso diretto all'hardware;
- non è presente la memoria virtuale: la memoria viene considerata come un unico spazio fisico allocato a tempo di compilazione.

In TinyOS sono presenti due tipologie di esecuzione: eventi hardware e software. Gli eventi hardware fanno riferimento a interrupt generati da timer, sensori o dispositivi di comunicazione. Per quanto riguarda gli eventi software, invece, i task sono procedure ad esecuzione differita nel tempo. Quando un task viene chiamato viene inserito in una coda FIFO gestita da uno scheduler.

## 3.5 Realizzazione della rete

La rete che abbiamo realizzato è composta dai nodi ACme collegati agli elettrodomestici e dalla *base station*.

I nodi ACme sono stati programmati utilizzando l'applicazione *ACMeterAppX2* presente in *TinyOS-2.x-contrib* sviluppata dai produttori del sensore. Tramite questa applicazione l'ACme calcola la potenza istantanea consumata e invia in broadcast ogni secondo il valore. Per il sistema proposto abbiamo deciso di utilizzare un timer di un minuto in quanto un campionamento troppo fitto avrebbe portato ad una quantità eccessiva di dati rispetto all'obiettivo da raggiungere.

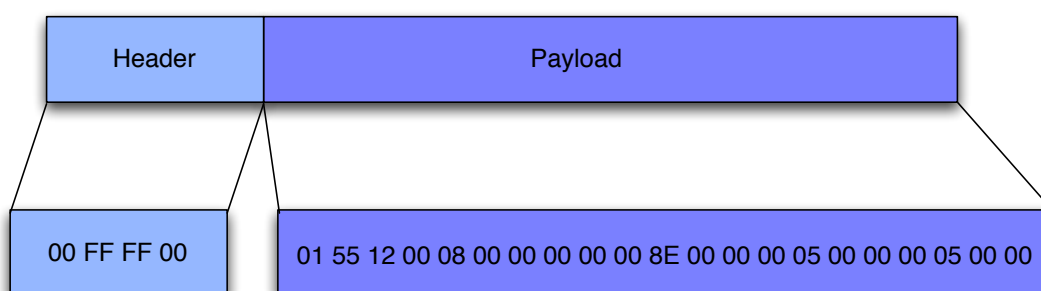
La *base station*, invece, si mette in ascolto dei dati provenienti dai sensori che trasmettono sullo stesso canale. L'applicazione Java *Rmi Server* tramite il



package *power* legge i pacchetti dalla *base station*, li interpreta e salva i dati nel database. Come *base station* abbiamo deciso di utilizzare un Telosb.

### 3.5.1 Formato del messaggio

Ogni messaggio inviato dall'ACme è formato da 25 byte e come mostrato nella Figura 3.4 ha la seguente struttura:



**Figura 3.4:** Struttura del messaggio inviato dall'ACme

- I primi 4 byte formano l'*header* del messaggio e hanno nel nostro caso sempre valore pari a 00 FF FF 00, che indicano la trasmissione in broadcast del messaggio;
- il quinto byte contiene l'ID del nodo che ha inviato il messaggio;
- i byte 11, 12, 13 rappresentano i valori della potenza totale;
- i byte 15, 16, 17 consentono di calcolare la potenza attiva;
- i byte 19, 20, 21 contengono il valore della potenza apparente;
- gli ultimi byte 23, 24, 25 riportano i valori della potenza reattiva.

I dati d'interesse all'applicazione presenti nel payload riguardano unicamente l'ID del nodo e la potenza totale. *Rmi Server* memorizza tali informazioni in un array e dopo aver effettuato la conversione da esadecimale a decimale procede con la relativa memorizzazione nel database.

# Capitolo 4

## Algoritmo di predizione

### 4.1 Introduzione

L'algoritmo di predizione ricopre un ruolo fondamentale in questo lavoro di tesi. Esso ha il compito di elaborare i dati raccolti attraverso la rete di sensori descritta nel Capitolo 3 e di predire, per ogni giorno, il comportamento di ogni elettrodomestico presente nell'abitazione. La predizione riguarda in particolare l'utilizzo, gli orari di accensione e le durate di funzionamento di ciascun elettrodomestico. Una volta terminata l'esecuzione, i risultati sono mostrati agli utenti attraverso una particolare funzione presente nell'applicazione Android *Energy Coach*. In questo capitolo verrà presentato nel dettaglio l'algoritmo di predizione che abbiamo definito e implementato per il sistema proposto. Tale algoritmo, anche se non presente in letteratura, può essere classificato nella categoria degli algoritmi non supervisionati.

### 4.2 Funzionamento dell'algoritmo

Come ogni algoritmo di predizione anche quello da noi definito, ha una fase di training che utilizzando un subset di dati impara come dovrà comportarsi a seguito di nuovi dati in ingresso. Al fine di realizzare tale fase si utilizzano i dati raccolti nel periodo di monitoring, contenuti nella tabella *consumo* (Figura 4.1) del database. Essa tiene traccia di tutte le attività svolte da ogni dispositivo

	ID	id_ha	data_on	data_off	ora_on	ora_off
▶	5524	AA118	2010-10-26	2010-10-26	18:00:00	21:00:00
	5523	AA118	2010-10-25	2010-10-25	18:00:00	22:00:00
	5521	AA115	2010-10-22	2010-10-22	06:00:00	22:00:00
	5520	AA127	2010-10-26	2010-10-26	21:00:00	21:30:00

**Figura 4.1:** *Tabella consumo*

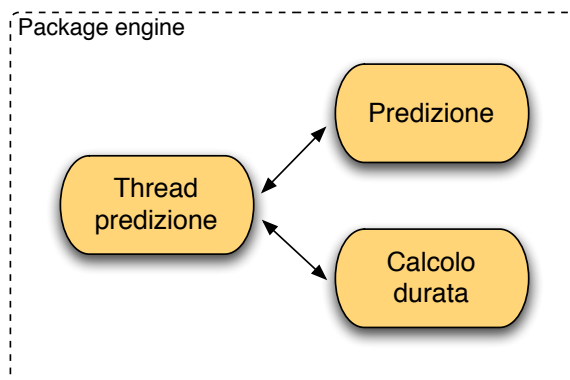
domestico, memorizzando giorno e ora di accensione e di spegnimento. Per ogni singolo elettrodomestico viene creato un ArrayList contenente l'ora\_on e l' ora\_off.

L'algoritmo che abbiamo definito prevede, in particolare, cinque fasi di funzionamento sequenziali:

1. raggruppamento in cluster degli orari di accensione e di spegnimento;
2. rappresentazione dell'andamento dell'utilizzo;
3. predizione dell'ultizzo dell'elettrodomestico;
4. predizione degli orari di accensione;
5. predizione della durata di funzionamento.

Le fasi descritte sono implementate tramite le seguenti classi Java presenti sul Rmi Server nel package *engine* Figura 4.2:

- *Thread predizione* si occupa della creazione dei cluster, della predizione del numero e degli orari di accensione;
- *Predizione* predice se l'elettrodomestico viene utilizzato o meno nel giorno considerato;
- *Calcolo durata* implementa il calcolo della durata di ogni singola accensione che riceve come parametro.



**Figura 4.2:** *Struttura classi Java presenti sul server*

### 4.2.1 Creazione dei cluster

La creazione dei cluster consiste in una fase di filtraggio dei dati in ingresso. In particolare si analizzano i dati di ogni giorno, raggruppando negli stessi cluster, chiamati cluster on, gli orari di accensione che distano tra loro al massimo 15 min. Lo stesso tipo di procedura è realizzata per gli orari di spegnimento, definendo questa volta i cluster off. Questa operazione è molto importante poiché si evita che ripetute accensioni e spegnimenti del singolo elettrodomestico, a distanza di pochi minuti, possano inficiare il risultato finale della predizione.

Una volta terminato questo procedimento si calcola la media di ogni cluster e si memorizzano i risultati trovati in un ArrayList. Ad ogni elettrodomestico corrisponde quindi un ArrayList contenente per ogni giorno i valori medi degli orari di accensione e spegnimento. Nel caso in cui l'elettrodomestico non sia stato utilizzato in quel dato giorno nell'ArrayList viene inserito uno zero.

#### **Esempio**

Abbiamo esaminato, durante una giornata, l'accensione e lo spegnimento del forno mediante i dati raccolti dal corrispondente ACme (avente ID=AA213) e rappresentati nella tabella 4.3 da cui si ricava il profilo illustrato in Figura 4.4. La Figura 4.5 e la Figura 4.6 rappresentano i dati raggruppati nei vari cluster. La

Figura 4.7 e la Figura 4.8 rappresentano i valori medi calcolati per ogni cluster.

	ID	id_ha	data_on	data_off	ora_on	ora_off
▶	5531	AA213	2010-11-03	2010-11-03	07:00:00	07:05:00
	5532	AA213	2010-11-03	2010-11-03	07:10:00	07:24:00
	5533	AA213	2010-11-03	2010-11-03	07:37:00	07:40:00
	5534	AA213	2010-11-03	2010-11-03	07:45:00	08:01:00
	5535	AA213	2010-11-03	2010-11-03	08:16:00	08:20:00
	5536	AA213	2010-11-03	2010-11-03	08:24:00	08:40:00
	5537	AA213	2010-11-03	2010-11-03	16:00:00	16:20:00

Figura 4.3: Tabella cluster

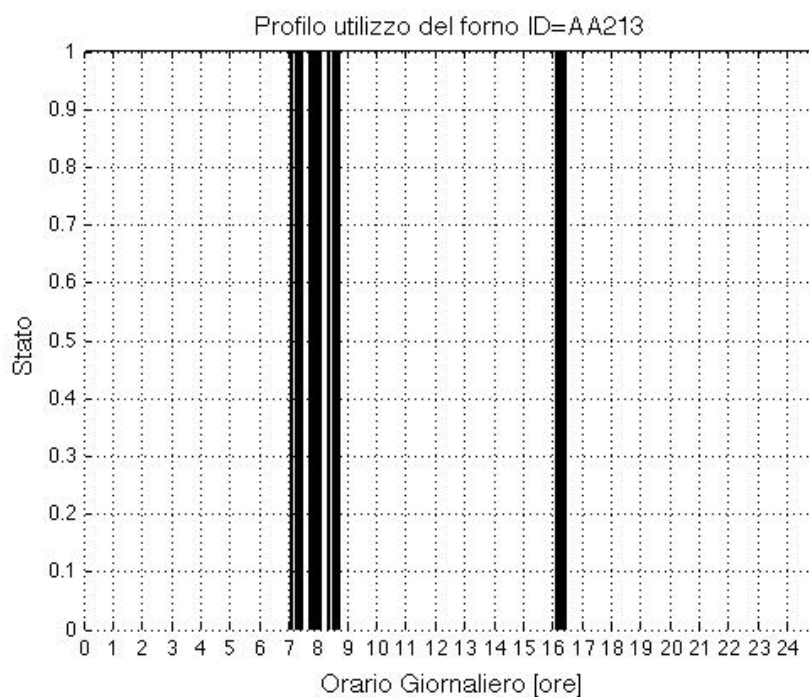


Figura 4.4: Profilo di utilizzo del forno AA213

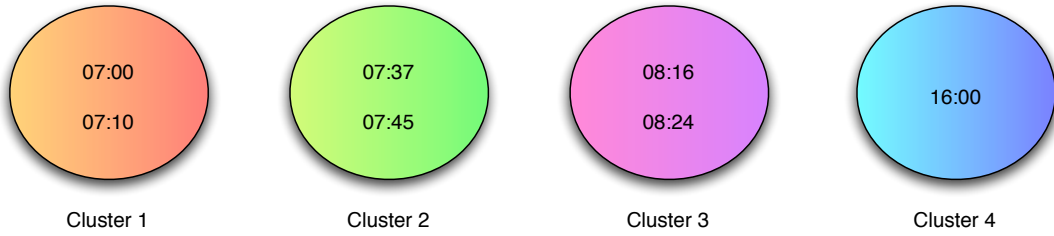


Figura 4.5: Cluster on creati dell' algoritmo

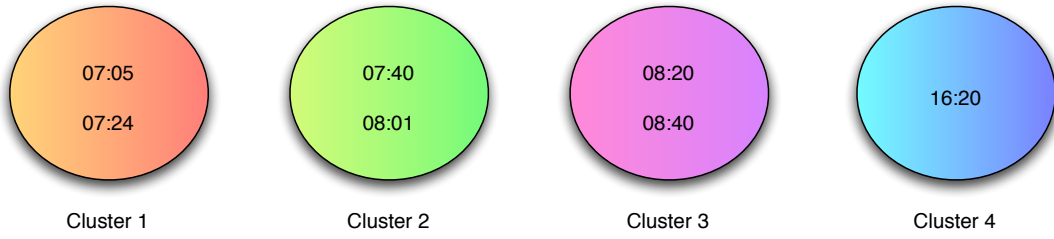


Figura 4.6: Cluster off creati dell' algoritmo

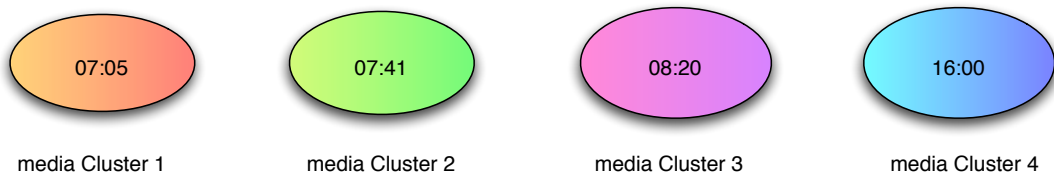


Figura 4.7: Media cluster on

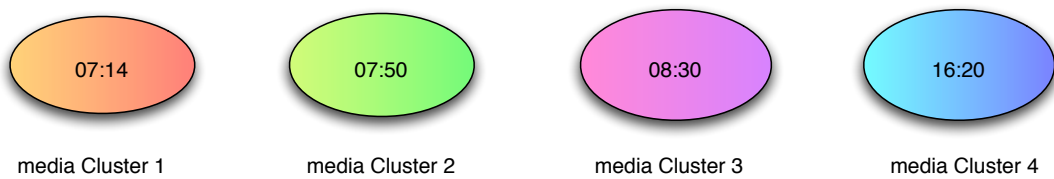


Figura 4.8: Media cluster off

### 4.2.2 Rappresentazione dell'andamento dell'utilizzo

A questo punto inizia la parte vera e propria di predizione: utilizzando i dati estrapolati dai cluster viene creata una stringa di  $N$  caratteri (ogni carattere corrisponde a un giorno del periodo di monitoring avente lunghezza  $N$ ) che raccoglie la storia dell'utilizzo del dispositivo domestico negli ultimi  $N$  giorni. Ogni posizione della stringa assume valore 1 se in quel giorno l'elettrodomestico è stato acceso, 0 in caso contrario. Ad esempio, considerando un ipotetico utilizzo del forno e supponendo  $N=28$  otteniamo la stringa seguente:

$$I = 1001100100110010011001001100 \quad (4.1)$$

Successivamente viene creato un oggetto *predizione* passando come parametri la stringa appena creata e le medie dei cluster precedentemente calcolate per ciascuna giornata. L'obiettivo di questo oggetto è quello di predire se l'elettrodomestico verrà usato o meno nel giorno di interesse.

### 4.2.3 Predizione dell'utilizzo

L'oggetto *predizione* analizza la stringa passata come parametro e cerca al suo interno il ripetersi di una certa ricorrenza di uno e zero. Inizialmente il pattern da ricercare ha lunghezza unitaria e corrisponde al valore dell'ultimo carattere (relativo al giorno precedente a quello da predire). Viene contato il numero di volte che questo carattere si ripete nella stringa (*count\_ultimo*), il numero di volte che è seguito da un 1 (*count\_uno*) e il numero di volte che è seguito da uno 0 (*count\_zero*). Una volta calcolati i valori dei tre contatori si calcolano le due probabilità condizionate: la probabilità che la predizione abbia valore 1 e la probabilità che la predizione abbia valore 0 dato l'ultimo carattere della stringa, ossia

$$P(0|ultimo) = \frac{count\_zero}{count\_ultimo} \quad (4.2)$$

$$P(1|ultimo) = \frac{count\_uno}{count\_ultimo} \quad (4.3)$$

Questo procedimento viene ripetuto 10 volte aumentando di un carattere ad ogni iterazione la lunghezza del pattern da riconoscere: con lunghezza pari a 2 si utilizzano gli ultimi 2 caratteri della stringa, con lunghezza pari a 3 gli ultimi 3 e così via fino ad arrivare a una lunghezza massima di 10.

Avendo a disposizione tutti i valori di probabilità viene scelto il valore più alto per fare predizione: se viene predetto uno 0 l'algoritmo termina, se invece viene predetto un 1 l'algoritmo continua con il calcolo degli orari di accensione.

Esiste un caso nel quale il procedimento descritto fallisce ossia nella circostanza nella quale nessun pattern di qualunque lunghezza viene riconosciuto nella stringa; in questo caso il calcolo delle probabilità è impossibile da effettuare. Ogni qualvolta si verifica questa situazione si è deciso di scegliere a maggioranza. Se nella stringa il numero di 0 è maggiore del numero di 1 sarà predetto uno 0, 1 se si verifica il contrario.

### **Esempio**

Facendo riferimento alla stringa I 4.1 si calcolano i valori di probabilità a partire dall'ultimo carattere della stringa.

$$P(0|0) = 0.53$$

$$P(1|0) = 0.46$$

$$P(0|00) = 0$$

$$P(1|00) = 1$$

$$P(0|100) = 0$$

$$P(1|100) = 1$$

.....

$$P(0|001001100) = 0$$

$$P(1|001001100) = 1$$

In questo caso viene predetta l'accensione dell'elettrodomestico e il pattern utilizzato ha lunghezza pari a 2.



#### 4.2.4 Predizione degli orari di accensione

Nel caso in cui venga predetta un'accensione per il giorno considerato si passa a stimarne il numero e gli orari di accensione. Il primo passo consiste nel raccogliere tutti i dati che riguardano i giorni “simili” a quello da predire. Si utilizza il pattern selezionato al punto precedente, ossia quello con la probabilità maggiore, e si cercano tutte le sue occorrenze presenti nella stringa  $I$ . Si analizzano le occorrenze trovate e si selezionano gli 1 che le seguono, in quanto questi 1 rappresentano i giorni “simili” a quello da predire.

Di questi giorni si prendono i valori degli orari di accensione e le durate relative. Inizialmente si predice il numero di accensioni facendo una media delle volte che è stato acceso nei giorni “simili”.

In seguito vengono inseriti in un array ( $ora\_on$ ) tutti gli orari di accensione selezionati: questi dati vengono utilizzati per calcolare le ore di accensione il cui numero è uguale al numero medio di accensioni predette.

A tal fine vengono generate tante gaussiane quanti sono gli elementi dell'array  $ora\_on$  ognuna delle quali ha media pari al valore  $i$ -esimo contenuto nel vettore e varianza 0.5, in modo da considerare la mezz'ora precedente e la mezz'ora successiva. Per la scelta di questo valore sono stati effettuati alcuni test, per maggiori dettagli si rimanda al Capitolo 7. Si genera una funzione  $f$  data dalla somma di tutte le gaussiane e di questa si trova il picco. Il valore corrispondente sull'asse delle ascisse è il primo valore di ora di accensione predetto.

Definita la prima ora di accensione si esegue una chiamata ricorsiva a una nuova funzione ( $somma\_ricorsiva$ ) che ha il compito di calcolare i restanti orari di accensione, questa chiamata viene invocata un numero di volte pari al numero medio di accensioni predette -1.  $Somma\_ricorsiva$  opera iniziando ad eliminare dal vettore  $ora\_on$  tutti i valori che hanno contribuito al calcolo della prima ora di accensione. Per capire quali orari effettivamente vadano eliminati dal vettore sono stati eseguiti e valutati alcuni test che saranno presentati nel Capitolo 7, in seguito ai

quali si è deciso di usare come valore di filtro 30 minuti. Vengono quindi eliminati dal vettore `ora_on` tutte gli orari che sono a una distanza di mezz'ora dall'orario predetto. L'algoritmo a questo punto procede come descritto precedentemente: si calcola la nuova funzione  $f$  e si trova il nuovo picco.

#### 4.2.5 Predizione della durata di funzionamento

L'ultimo passo dell'algoritmo riguarda la predizione della durata di funzionamento per ogni orario d'accensione calcolato al punto precedente. In questo caso si usano sempre i dati, che abbiamo memorizzato nel vettore `durata`, che come descritto precedentemente fanno riferimento esclusivamente alle durate dei giorni utilizzati per effettuare la predizione. Dei valori riguardanti la durata viene semplicemente calcolata la media e il risultato viene utilizzato come valore predetto per la durata.

Terminato il calcolo di tutti i valori necessari per la predizione i risultati vengono inseriti nella tabella `predizione` del database (Figura 4.9) che verrà letta dall'applicazione Energy Coach per poi mostrarli agli utenti.

	id	acme	ora_on	durata
▶	207	AA124	16:00:00	01:04:00
	208	AA124	07:59:00	01:04:00
	209	AA131	09:59:00	03:00:00

**Figura 4.9:** *Tabella predizione*

### 4.3 Variante dell'algoritmo

È utile precisare che abbiamo definito una variante dell'algoritmo descritto nel paragrafo precedente. Nella stringa di predizione per ogni elettrodomestico viene memorizzato non solo il fatto di essere stato acceso o meno, ma anche il numero di accensioni. Questa variante a livello implementativo ha comportato solo un nuovo modo di memorizzare la stringa: al posto del valore 1 è stato inserito il numero effettivo di accensioni per ogni giorno. Ad esempio considerando l'utilizzo del forno dell'esempio precedente la stringa diventa la seguente:

$$I = 200320030030020022003002300 \quad (4.4)$$

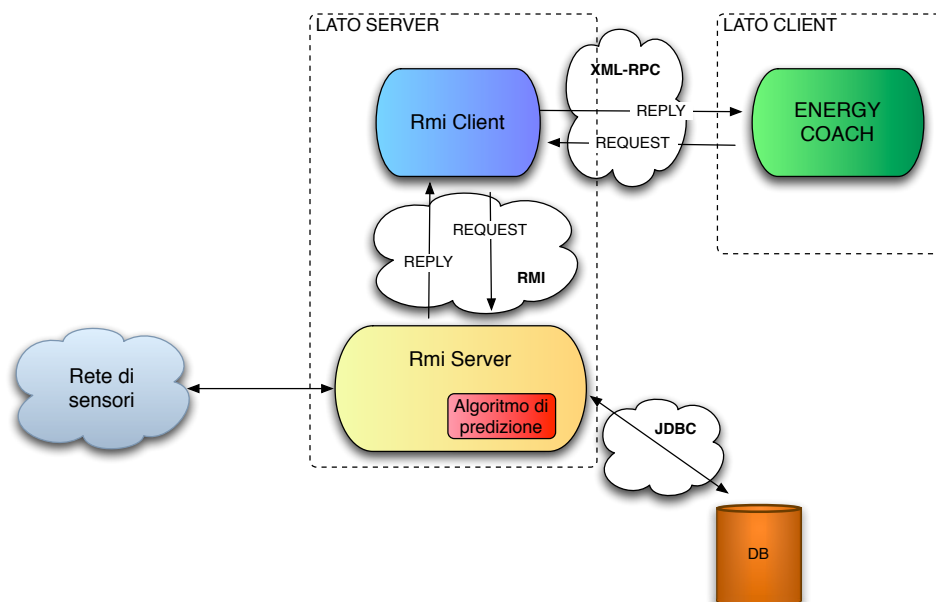
Il motivo per cui abbiamo definito tale variante è che intuitivamente sembra poter fornire migliori prestazioni, in termini di predizione degli orari di accensione dei dispositivi, rispetto alla versione dell'algoritmo descritto nel Paragrafo 3.2. Per un confronto numerico si rimanda al Capitolo 7.

# Capitolo 5

## Server

### 5.1 Introduzione

In questo capitolo verrà descritto il lato server del sistema. Come detto precedentemente per facilitare la comunicazione tra client e server abbiamo deciso di dividere il server in due applicazioni distinte: Rmi Client e Rmi Server, comunicanti tra loro tramite gli RMI di Java Figura 5.1.



**Figura 5.1:** *Architettura del sistema*

Per la memorizzazione dei dati, inoltre, ci siamo avvalsi di un database My-

SQL, che sarà descritto in seguito.

## 5.2 Rmi Server

Il motivo principale dell'utilizzo di RMI nello sviluppo dell'applicazione lato server è dovuto alla possibilità futura di interfacciare la nostra applicazione con il gateway già esistente nel laboratorio ANTLab del Politecnico di Milano, il quale utilizza RMI per la comunicazione. Il metodo RMI [29] (*Remote Method Invocation*) consente ad applicazioni distribuite di comunicare tra loro mediante l'invocazione di metodi tra oggetti remoti.

Android non supporta tale tipo di comunicazione, per questo motivo abbiamo diviso la parte lato server nelle due applicazioni Rmi Client e Rmi Server.

Rmi Client fornisce interoperabilità tra l'Energy Coach e Rmi Server essendo in grado di comunicare con Android tramite *xml-rpc* [30] (*Remote Procedure Call*), un protocollo che permette di eseguire chiamate a procedure remote attraverso la rete Internet. Come mostrato dalla Figura 5.2 utilizza lo standard xml per codificare le richieste che vengono trasportate tramite il protocollo HTTP.

A tale scopo abbiamo realizzato il package *interfaces* contenente le due interfacce:

- *Androidmanager.java* è implementata da Rmi Client stesso e mette a disposizione tutti i metodi che possono essere invocati dall'Energy Coach;
- *WSNGateway.java* è implementata da Rmi Server e contiene la replica dei metodi presenti nell'interfaccia precedente necessari al Rmi Client per inoltrare le richieste ricevute al Rmi Server utilizzando RMI.

Rmi Server è la parte dominante del lato server la cui classe principale è *ComputeEngine.java*. Essa effettua la connessione con Rmi Client tramite RMI, e interagendo con altre classi Java svolge le seguenti funzionalità:

- *Raccolta dati dalla rete di sensori*: il package *power.comm* gestisce la raccolta dei dati dalla rete di sensori;

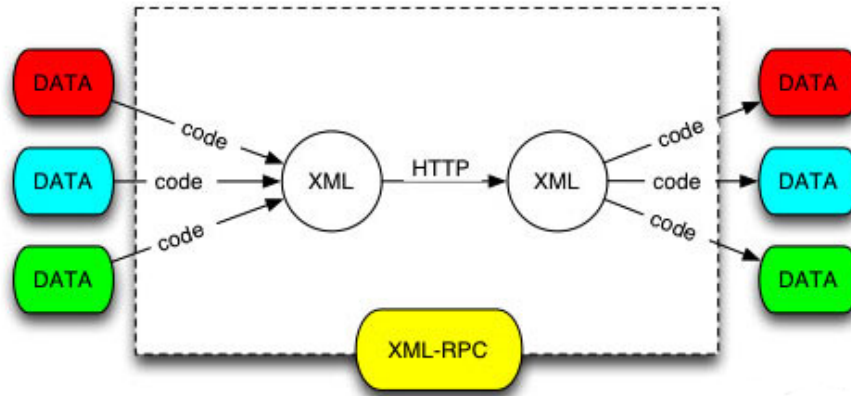


Figura 5.2: Funzionamento XML-RPC

- *gestione del database*: la classe *HomeAppliance.java* contiene le query utilizzate per interagire con il database MySQL;
- *predizione dell'utilizzo degli elettrodomestici*: il *thread\_predizione.java* raccoglie i dati relativi all'utilizzo degli elettrodomestici nel periodo di monitoring e in base ad essi apprende un insieme di abitudini per predire il comportamento del giorno corrente;
- *risparmio energetico*: il *thread\_DB.java* si occupa della segnalazione tramite messaggi all'Energy Coach riguardanti la possibilità d'utilizzo dei pannelli solari e di gestire le luci nelle stanze in cui non viene rilevata la presenza;
- *scheduling degli elettrodomestici*: il *thread\_DB.java* gestisce l'accensione e spegnimento degli elettrodomestici schedulati dall'utente.

### 5.2.1 Raccolta dati dalla rete di sensori

Attraverso la creazione dell'oggetto *SerialWSNInterface*, Rmi Server apre un'interfaccia verso il *mote* Telosb che funge da *Base Station*, crea e lancia il thread *InteractionThread* che si mette in attesa dei pacchetti provenienti dalla rete di sensori. Successivamente, ogni pacchetto letto viene inserito in un array di byte

(*packet*), il quale viene passato come parametro all'oggetto *Message* per essere interpretato. Da tale array vengono poi estratte le informazioni riguardanti l'identificativo del sensore e il valore della potenza rilevata, memorizzando tali dati nella tabella *consumo* del database. Per maggiori dettagli sulla struttura del pacchetto si rimanda al Capitolo 3.

### 5.2.2 Gestione del database

Come detto precedentemente, al fine di gestire e memorizzare i dati abbiamo utilizzato un database MySQL [31], chiamato *Energy Manager*. La comunicazione tra il database e Rmi Server avviene attraverso l'interfaccia *JDBC* (*Java Database Connectivity*) che permette ad applicazioni Java l'accesso a basi di dati indipendentemente dal tipo di DBMS utilizzato.

Avendo scelto un database MySQL si è reso necessario includere il *Connect/J*[32], un driver di tipo 4 che, a differenza delle altre versioni, è basato esclusivamente su un'implementazione Java che non richiede le librerie client di MySQL.

Impostato il driver JDBC, Rmi Server si connette a MySQL Server invocando il metodo *getConnection()* ed accede al database *Energy Manager* impostando il nome utente e la password. Terminata la fase di connessione Rmi Server inizia ad invocare le query necessarie.

### 5.2.3 Predizione dell'utilizzo degli elettrodomestici

L'algoritmo di predizione presente nel Rmi Server implementa le funzioni necessarie per l'apprendimento e la predizione dell'uso degli elettrodomestici. In base ai dati ricavati dalla tabella *consumo* del database predice l'utilizzo, gli orari di accensione e le rispettive durate di funzionamento per ciascun elettrodomestico predetto per il giorno corrente. Per maggiori dettagli si rimanda al Capitolo 4.

### 5.2.4 Funzionalità di risparmio energetico

Le politiche di risparmio energetico sono gestite tramite il thread Java (*thread\_DB.java*), presente sul Rmi Server. Esso controlla lo stato dei pannelli solari presente nel database e, se è possibile utilizzare l'energia derivante da essi, avvisa l'utente tramite un messaggio mostrato dall'*Energy Coach*. Per visualizzare questi avvisi sull'*Energy Coach* è presente un servizio in background che rimane in attesa dei messaggi provenienti dal server così come descritto nel Capitolo 6.

Tale thread, infine, ha anche il compito di verificare la presenza di persone nelle varie stanze dell'abitazione e il relativo stato delle luci. Nel caso in cui vi siano luci accese in stanze in cui non viene rilevata la presenza, l'utente viene avvisato tramite l'*Energy Coach* con un opportuno messaggio di segnalazione.

### 5.2.5 Scheduling degli elettrodomestici

Un'altra funzionalità del *thread\_DB.java* consiste nel verificare le tabelle *scheduling* e *scheduling\_hvac* per prelevare i dati relativi agli elettrodomestici programmati. Allo scattare dell'ora di accensione programmata, per ciascun elettrodomestico, si preoccupa di inserire i dati relativi l'identificativo dell'elettrodomestico, la data e ora di accensione e l'identificativo dell'utente nella tabella *consumo*.

Il *thread\_DB* si occupa, infine, di attivare il *Thread\_off* che rileva lo spegnimento dell'elettrodomestico, inserisce nella tabella *consumo* data e ora di spegnimento e aggiorna lo stato a 0 nelle tabelle *HomeAppliance*, *Light* o *Hvac* a seconda della tipologia di elettrodomestico.



### 5.3 Database *Energy Manager*

Il database è stato organizzato nelle seguenti tabelle (Figura 5.3) :

- *HomeAppliance, Light, Hvac*: contengono i dati relativi agli elettrodomestici;
- *consumo*: raccoglie i dati relativi ai consumi di ciascun elettrodomestico;
- *predizione*: memorizza per i dispositivi previsti dall' algoritmo di predizione, gli orari d'accensione e le relative durate di funzionamento;
- *scheduling*: tiene traccia dei dati relativi alla programmazione degli elettrodomestici e dell'impianto di illuminazione;
- *scheduling\_hvac*: contiene i dati relativi alla programmazione dell'HVAC (Heating, Ventilation, Air Condition);
- *user*: contiene username e password di ogni utente registrato;
- *stanza*: memorizza il nome, il tipo, il piano, la temperatura e lo stato di presenza di qualcuno in ogni stanza;
- *category*: elenca le categorie nelle quali sono stati divisi gli elettrodomestici;
- *security*: registra le informazioni relative alla sicurezza dell'abitazione quali sistemi antifurto e antifumo. Inoltre tiene traccia dello stato dei pannelli solari. Trattandosi di informazioni globali sull'abitazione non è stata rilevata la corrispondenza con altre tabelle.

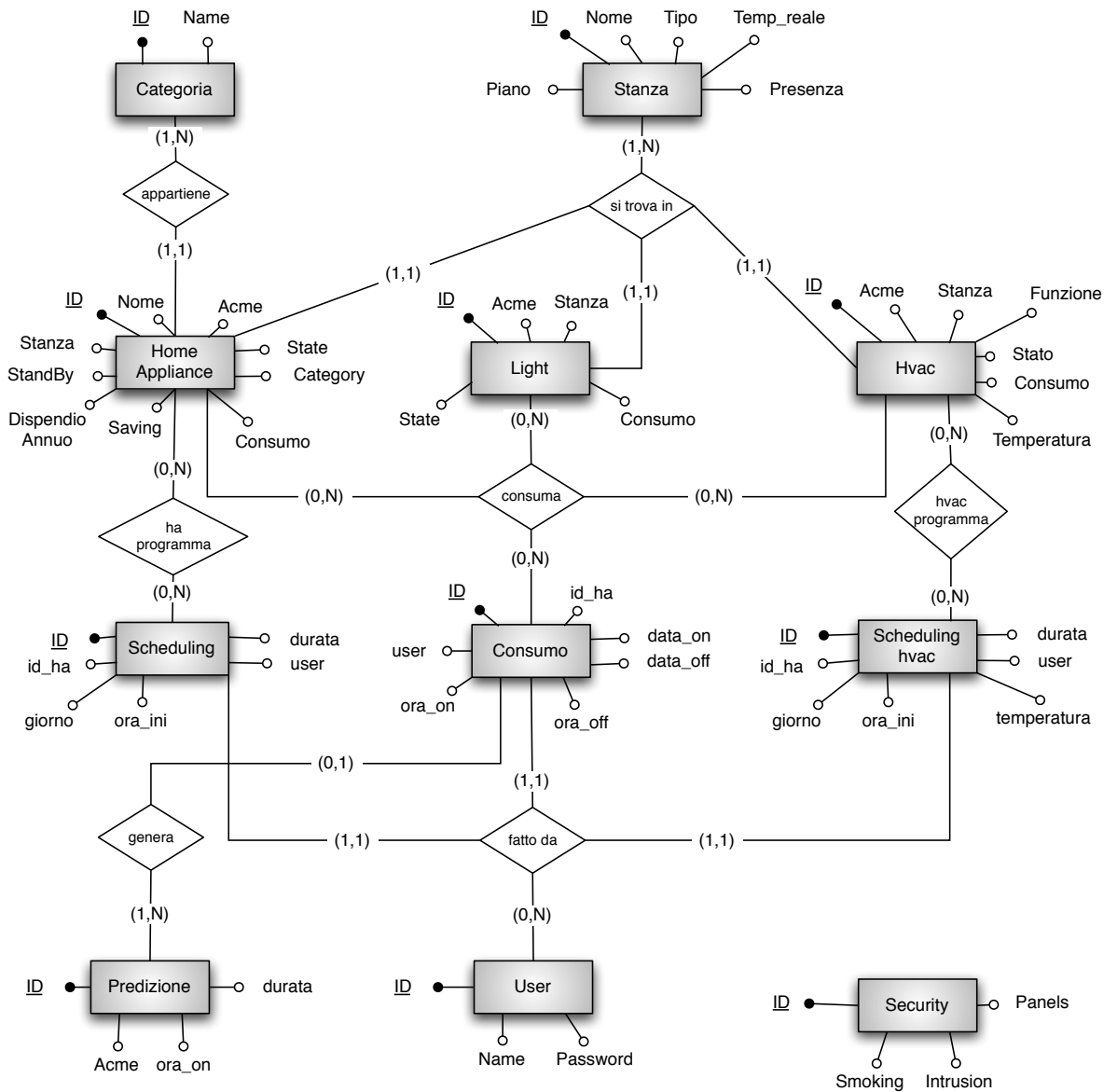


Figura 5.3: Database

# Capitolo 6

## Energy Coach

### 6.1 Introduzione

Il lato client del nostro sistema consiste in un'applicazione per dispositivi mobili sviluppata in Android. Tramite essa l'utente può interagire con tutti gli elettrodomestici della casa, controllarne lo stato, i rispettivi consumi e costi e infine visualizzare gli avvisi riguardanti il risparmio energetico provenienti dal server. In questo capitolo sarà illustrata l'implementazione di tale applicazione, nonché le sue funzionalità.

### 6.2 Android

Android [33] è una piattaforma open source per dispositivi mobili, basata sul sistema operativo Linux e sviluppata dall'Open Handset Alliance. La scelta di questa piattaforma è legata principalmente sia alla disponibilità di codice sorgente, essendo Android rilasciato con licenza open-source, sia alla semplicità con cui l'applicazione da noi sviluppata può integrarsi perfettamente nel sistema operativo e adattarsi all'uso in mobilità. Android dispone infatti di un servizio chiamato Android Market attraverso il quale gli utenti possono scaricare nuove applicazioni sul proprio smartphone con estrema facilità, senza dover seguire complicate procedure di installazione. L'user experience in Android è interamente basata sull'interazione dell'utente attraverso il touch screen. Gli utenti possono quindi interagire con



**Figura 6.1:** Architettura di Android

L'Energy Coach tramite il tocco delle dita e, per scrivere, possono utilizzare una pratica tastiera touch sullo schermo o, nei modelli che ne dispongono, una tradizionale tastiera qwerty.

L' SDK di Android consente di effettuare lo sviluppo di applicazioni di telefonia mobile sui principali sistemi operativi (Linux, Mac OS e Windows) usando Java come linguaggio di programmazione.

Come mostrato dalla Figura 6.1 il sistema operativo Android è costituito dalle seguenti componenti:

- *Applicazioni:* Android mette a disposizione dell'utente un set di applicazioni base, interamente scritte in linguaggio Java;
- *Framework per applicazioni:* progettata per semplificare il riutilizzo dei componenti, consente infatti agli sviluppatori l'accesso alle API utilizzate dalle

applicazioni di base;

- *Librerie*: comprende un set di librerie C/C++. *Android Runtime*, invece, consiste in un set di librerie centrali che forniscono le funzionalità base su cui si appoggiano le altre librerie;
- *Kernel Linux*: Android si appoggia sulla versione 2.6 di Linux per i servizi del sistema centrale.

### 6.3 Realizzazione dell'applicazione

Per la realizzazione dell'applicazione abbiamo deciso di utilizzare l'ambiente di programmazione Eclipse [34] nel quale è stato incluso il plugin per lo sviluppo di applicazioni Android, la cui versione utilizzata è la 2.1. Inizialmente, per verificare il funzionamento di quanto realizzato, è stato utilizzato l'emulatore Android AVD. L'applicazione finale, oltre che sull'emulatore, è stata inoltre testata anche con il Galaxy tab, il nuovo tablet Samsung basato sul sistema operativo Android [35]

### 6.4 Energy Coach

L'utente tramite l'Energy Coach ha la possibilità di interagire con tutti gli elettrodomestici presenti nell'abitazione e in particolare può usufruire delle seguenti funzionalità:

- *Configurazione delle stanze e degli elettrodomestici presenti in esse*: l'utente può configurare l'applicazione inserendo il numero e la tipologia di stanze (bagno, cucina, camera...) presenti nella propria abitazione e per ogni stanza dichiarare quali elettrodomestici sono presenti nella stessa;
- *Controllo degli elettrodomestici*: tramite funzioni apposite è possibile comandare lo stato di elettrodomestici, impianto di illuminazione e HVAC;
- *Visualizzazione consumi e costi*: l'utente può visualizzare, tramite tabelle e grafici, i consumi e i costi degli apparati presenti nella casa disaggregandoli

per elettrodomestico, per stanza, per piano, e per utente. È possibile, inoltre, effettuare un'analisi comparata del consumo dei vari utenti;

- *Risparmio energetico*: tramite opportune finestre popup vengono visualizzati i messaggi provenienti dal server riguardo le abitudini dell'utente e i consigli per risparmiare energia.

### **6.4.1 Configurazione delle stanze e degli elettrodomestici presenti in esse**

Appena avviata l'applicazione all'utente viene chiesto di inserire nome e password per effettuare il login nel sistema. Una volta eseguita questa operazione viene data la possibilità di configurare l'abitazione.

Nella prima fase di configurazione vengono messe a disposizione vari tipi "standard" di stanze e cliccando su di esse si apre una finestra di dialogo che permette l'assegnazione del nome da attribuire alla stanza stessa e del piano in cui si trova. Terminato il setting delle stanze è possibile poi proseguire cliccando sul bottone *Continue* presente in fondo all'elenco.

La seconda fase consiste, invece, nel configurare i dispositivi presenti in ogni stanza. Per agevolare l'utente viene visualizzata una lista di categorie e cliccando su ognuna di esse vengono mostrati i dispositivi facenti parte della stessa. Selezionando il dispositivo desiderato appare una finestra nella quale è possibile inserire il numero dell'ACme associato all'elettrodomestico, la potenza nominale e, tramite un menù a tendina, la stanza dove si trova. Per confermare le scelte fatte si clicca sul bottone *Save* presente in fondo alla schermata.

Terminata la configurazione è possibile avviare l'applicazione vera e propria, selezionando il bottone *Save your configurations and start the Energy Coach application*.

## 6.4.2 Homepage

L'homepage dell'applicazione permette di accedere a tutte le sezioni fondamentali dell'applicazione (Figura 6.2). Essa presenta sette icone che permettono di accedere alle diverse sezioni nelle quali abbiamo suddiviso l'applicazione.



Figura 6.2: Homepage Energy Coach

- *HVAC (Heating, Ventilation, Air Condition)*: permette il controllo del riscaldamento, dell'aria condizionata e della ventilazione;
- *Security*: si occupa della gestione degli apparati di sicurezza, come allarmi antifurto e anti incendio;
- *Switch*: visualizza tutti gli elettrodomestici accesi, con l'indicazione della stanza dove si trovano e permette di spegnerli con un semplice clic;
- *Light*: gestisce l'impianto di illuminazione;

- *Program*: consente la programmazione degli orari di accensione e spegnimento dei vari elettrodomestici per ogni giorno della settimana;
- *User*: visualizza l'utente corrente e permette la modifica della password;
- *Habits Predicted*: mostra i risultati dell'algoritmo di predizione.

Per il controllo e la visualizzazione delle altre funzionalità quali consumi, costi e schede dei singoli elettrodomestici, abbiamo deciso di sfruttare il menù offerto da Android e quindi di non inserire nell'homepage il bottone specifico. Si è ritenuto che questi aspetti fossero fondamentali per l'utente e per questo si è deciso di non renderli accessibili solo dalla homepage, ma da tutte le pagine dell'applicazione. In particolare, premendo il bottone menù presente sull'emulatore o sullo smartphone, vengono mostrate una serie di icone per la visualizzazione dei consumi e il controllo degli elettrodomestici (Figura 6.3).



**Figura 6.3:** Menù Energy Coach

### 6.4.3 Controllo degli elettrodomestici

Al fine di gestire gli elettrodomestici abbiamo inserito nel menù l'icona setting, tramite la quale si accede alle schede relative ai singoli elettrodomestici. Attraverso un menù a tendina si ha la possibilità di decidere se visualizzarli divisi per stanza, per piano o per categoria. Una volta selezionato il tipo di visualizzazione viene caricata una gallery dove sono presenti tutti gli elettrodomestici elencati in quella stanza, piano o categoria. Nella parte bassa della scheda sono in particolare raffigurati tutti gli elettrodomestici presenti e, semplicemente cliccando sull'immagine, si visualizzano le informazioni relative. La scheda (Figura 6.4) è composta da un titolo dove viene indicato il nome dell'elettrodomestico in questione e un



riquadro nel quale viene indicata la stanza nella quale si trova il dispositivo selezionato, il numero del sensore ACme ad esso associato e l'utente. Tramite un toggle button si ha inoltre la possibilità di accendere o spegnere l'elettrodomestico in questione.



**Figura 6.4:** Scheda elettrodomestico

Alcune funzionalità aggiuntive sono infine accessibili tramite i quattro tasti della scheda:

- *Energy Consumption*: visualizza il consumo giornaliero, mensile e annuale dell'elettrodomestico in termini di kWh; inoltre offre la possibilità di visionare un grafico del consumo settimanale suddiviso per giorni;
- *Cost*: mostra il costo giornaliero, mensile e annuale dell'energia consumata dal dispositivo. Fornisce altresì dei consigli all'utente sul miglior modo di

utilizzo dell'elettrodomestico per risparmiare energia (Figura 6.5);

- *Add a new appliance*: permette di aggiungere un nuovo elettrodomestico nella stanza selezionata;
- *Delete appliance*: consente di eliminare l'elettrodomestico selezionato dalla stanza.

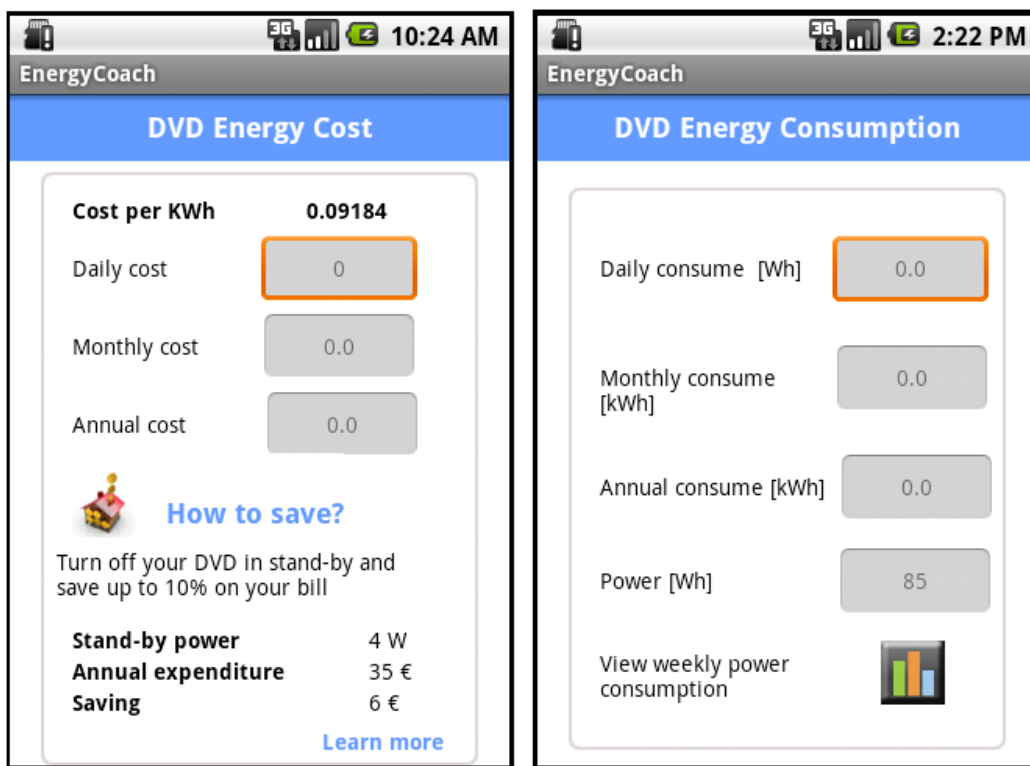


Figura 6.5: Scheda costi e consumi

Quello che è stato descritto fino ad ora riguarda il controllo degli elettrodomestici e delle luci. Per facilitare l'utente nella gestione degli apparati che fanno riferimento all'HVAC si è deciso di agire in modo leggermente diverso: l'utente, selezionando l'icona HVAC dalla homepage, può scegliere la stanza di proprio interesse in modo da visualizzare una finestra con l'indicazione della temperatura in quella stanza e scegliere quale funzionalità controllare: il riscaldamento, la ventilazione e l'aria condizionata. Una volta selezionato l'apparato sul quale si

vuole agire, si visualizza una scheda con indicato il nome della stanza, l'ACme di riferimento, la temperatura che si vuole impostare, lo stato e il bottone *save* per confermare le scelte fatte. Anche in questo caso è presente il bottone *add hvac* per eventualmente aggiungere nuovi dispositivi alla stanza e il bottone *delete hvac* per la relativa eliminazione.

#### 6.4.4 Visualizzazione consumi e costi

Nel paragrafo precedente è stato descritto come visualizzare i consumi e i costi dei singoli elettrodomestici. Un'importante funzionalità offerta dall'Energy Coach è la possibilità di controllare i consumi dell'intera abitazione. Selezionando dal menù il bottone *energy* si apre un nuovo menù a tendina che permette di visualizzare i consumi e i costi suddivisi per categoria, piano, utente e settimana. Per quanto riguarda i **consumi per categoria** viene mostrata una tabella dove sono elencati tutti i valori di consumo e di costo dell'ultimo mese.

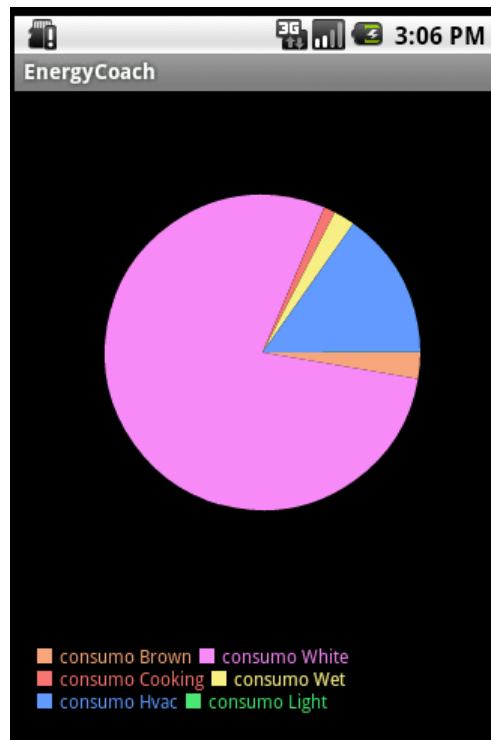


Figura 6.6: Diagramma consumi per categoria

Tramite il bottone *show graph* viene poi visualizzato un grafico a torta che indica le percentuali di consumo per ogni categoria (Figura 6.6).

Selezionando il **consumo per piano**, invece, viene presentata un'immagine della casa (Figura 6.7) nella quale ogni piano assume un colore diverso, rispettivamente, verde, arancione o rosso, in base al numero di elettrodomestici accesi e quindi in base al consumo in tempo reale di tutti i dispositivi del piano. Cliccando sull'immagine di ogni piano viene poi mostrato un elenco degli apparati accesi in quell'istante.



Figura 6.7: Consumi per piano e per utente

La terza modalità di visualizzazione dei consumi è **consumo per utente**: essa permette di selezionare l'utente desiderato e il periodo rispetto al quale si vuole misurare il consumo. Cliccando sul bottone *Comparison*, inoltre, si apre una scheda (Figura 6.7) nella quale si evidenzia una comparazione tra gli utenti in base al consumo.

Infine, la quarta e ultima visualizzazione dei consumi e costi riguarda il **consumo settimanale** di tutti gli elettrodomestici divisi per giorno. Viene mostrata all'utente sia una tabella riassuntiva che un istogramma (Figura 6.8).

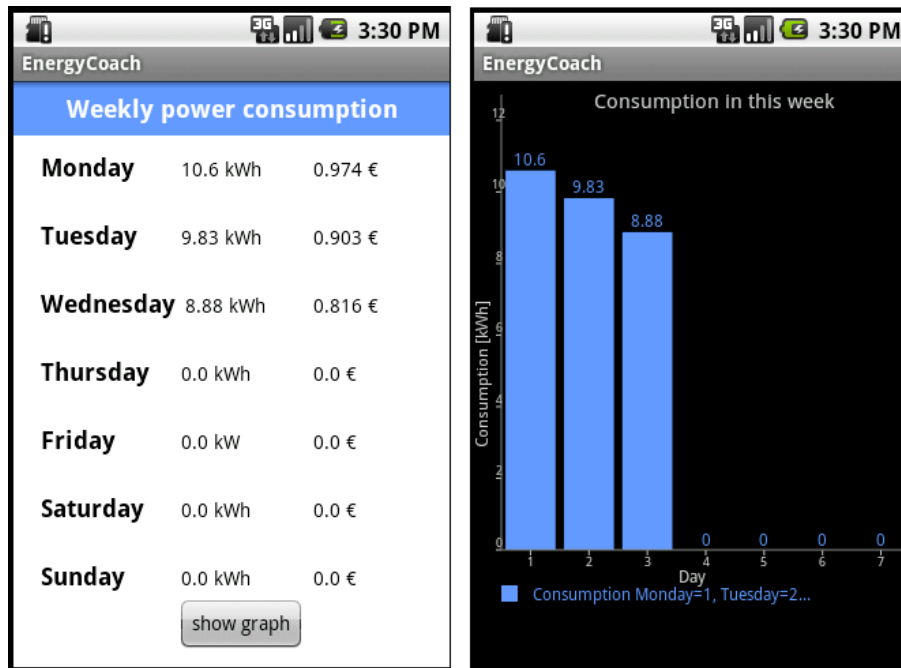


Figura 6.8: Consumi per settimana

### 6.4.5 Pianificazione delle attività dei dispositivi

L'Energy Coach offre all'utente la possibilità di programmare l'accensione e lo spegnimento di tutti i dispositivi domestici. Cliccando sull'icona Program, presente nella home, si apre una finestra nella quale è possibile scegliere il giorno desiderato. In seguito, scelta la stanza di interesse, appare la scheda di programmazione nella quale è possibile selezionare l'elettrodomestico da programmare, l'orario di accensione e la durata di funzionamento. Impostati tutti i parametri si conferma cliccando sul bottone *save*. In questa scheda è presente, inoltre, un secondo bottone *View all scheduling* che permette di visualizzare un elenco di dispositivi schedulati per il giorno selezionato. Cliccando sull'immagine è possibile

annullare il programma precedentemente impostato (Figura 6.9).

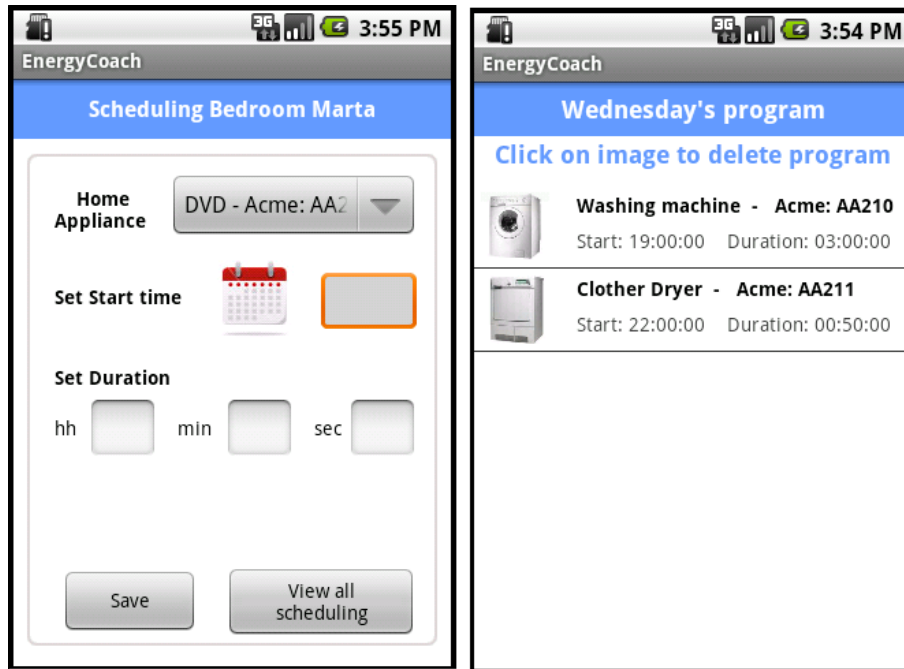


Figura 6.9: Scheduling

### 6.4.6 Risparmio energetico

Le funzionalità riguardanti il risparmio energetico, descritte in questo paragrafo, sono gestite in maniera diversa rispetto al resto dell'applicazione poiché la loro interazione con il server risulta essere differente rispetto ai casi precedenti. Nelle sezioni dell'applicazione analizzate fino a questo momento è sempre stato l'Energy Coach che inoltrava richieste al server e attendeva le risposte. In questo caso, invece, il comportamento del sistema è diverso: il server elabora i dati e solo quando lo ritiene opportuno invia i risultati all'Energy Coach senza che esso ne abbia fatto richiesta. Per tale motivo abbiamo implementato un servizio Android costantemente in attesa dei dati provenienti dal server. I servizi Android (al contrario delle attività fino ad ora utilizzate) hanno la caratteristica principale di poter essere lanciati da una qualsiasi activity, di essere eseguiti in background e

di mandare avvisi all'utente tramite textbox solo quando necessario.

L'Energy Coach offre diverse soluzioni per quanto riguarda il risparmio energetico:

1. In primo luogo ad ogni avvio dell'applicazione vengono mostrati i risultati dell'algoritmo di predizione eseguito dal server visualizzando le abitudini dell'utente. Egli viene avvisato della disponibilità di tali dati e se acconsente alla visualizzazione, essi vengono rappresentati in un elenco. Si è ritenuto opportuno dare la possibilità all'utente di visionare queste informazioni non solo all'avvio dell'applicazione, ma in qualsiasi momento lo desideri. Per questo motivo nell'homepage abbiamo inserito un bottone *Habits predicted* che se selezionato mostra nuovamente l'elenco delle abitudini (Figura 6.10).

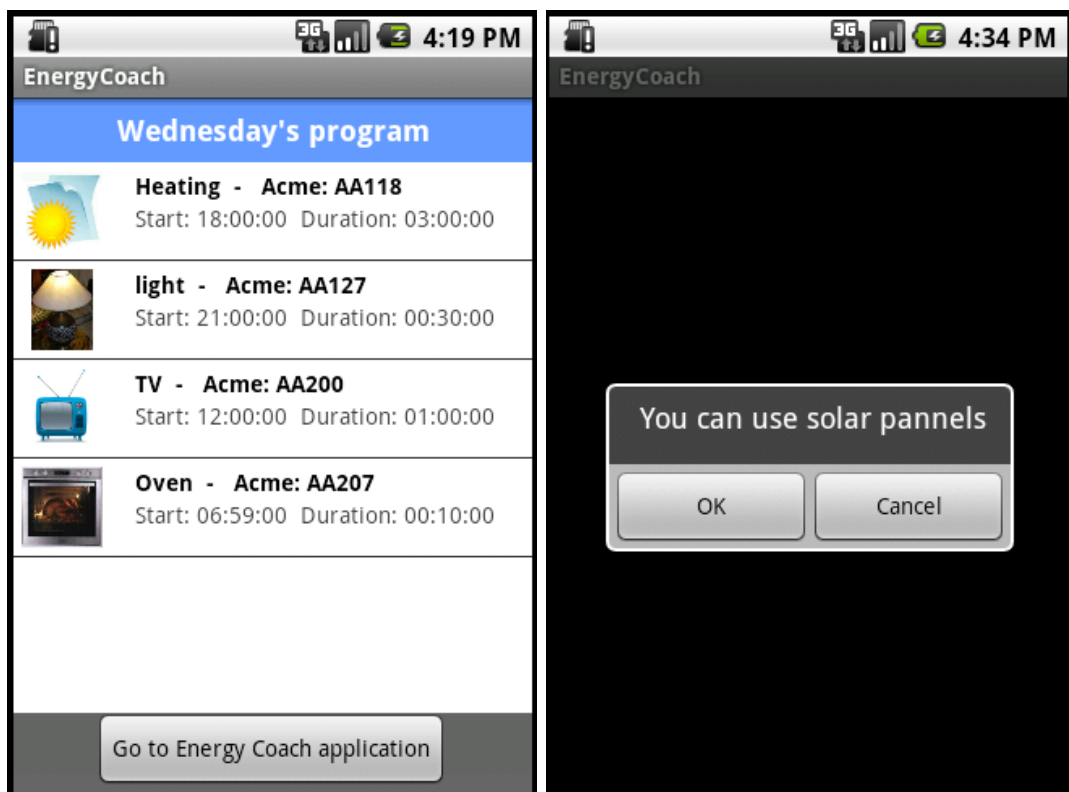


Figura 6.10: Predizione e messaggio utilizzo pannelli

2. Vengono mostrati altri messaggi e consigli, quando necessario, che riguardano principalmente:

- la possibilità di sfruttare, se disponibile, l'energia prodotta dei pannelli fotovoltaici e solari (Figura 6.10);
- il consiglio di spegnere le luci in una stanza quando viene rilevato non essere presente nessuna persona.

Inoltre, se l'utente dispone di una tariffa bioraria, l'applicazione, può consigliare di utilizzare elettrodomestici quali lavapiatti e lavatrice nella fascia di minor costo.

3. Infine, viene effettuata una gestione dei carichi elettrici. In seguito all'accensione di un elettrodomestico viene eseguito un controllo su tutti i dispositivi accesi e, nel caso in cui venga rilevato il raggiungimento della soglia massima di consumo, viene bloccato l'utilizzo avvisando l'utente che la soglia massima di potenza disponibile è stata superata. Tale funzionalità consente di evitare sovraccarichi e cadute di tensione che potrebbero danneggiare gli elettrodomestici collegati alla linea elettrica. Lo stesso controllo viene effettuato anche prima dell'avvio di ogni azione schedata: se nel giorno e nell'ora prevista non si verificano le condizioni di sicurezza appena descritte l'accensione programmata viene annullata.



# Capitolo 7

## Test e risultati numerici

### 7.1 Introduzione

Nel seguente capitolo saranno presentati i test svolti per valutare l'efficienza dell'algoritmo di predizione e per trovare la configurazione migliore dei parametri di funzionamento. A tal fine abbiamo condotto una fase di simulazione e una di sperimentazione in uno scenario d'uso reale.

### 7.2 Risultati numerici delle simulazioni

Al fine di avere una prima stima sulle prestazioni dell'algoritmo e scegliere in maniera ottima i parametri dello stesso abbiamo condotto una fase di simulazione in cui si è ricreato il comportamento degli elettrodomestici durante l'arco temporale di un anno. Per svolgere questi test abbiamo utilizzato il software di calcolo *Matlab*, creando un'implementazione dell'algoritmo di predizione e un simulatore di dati.

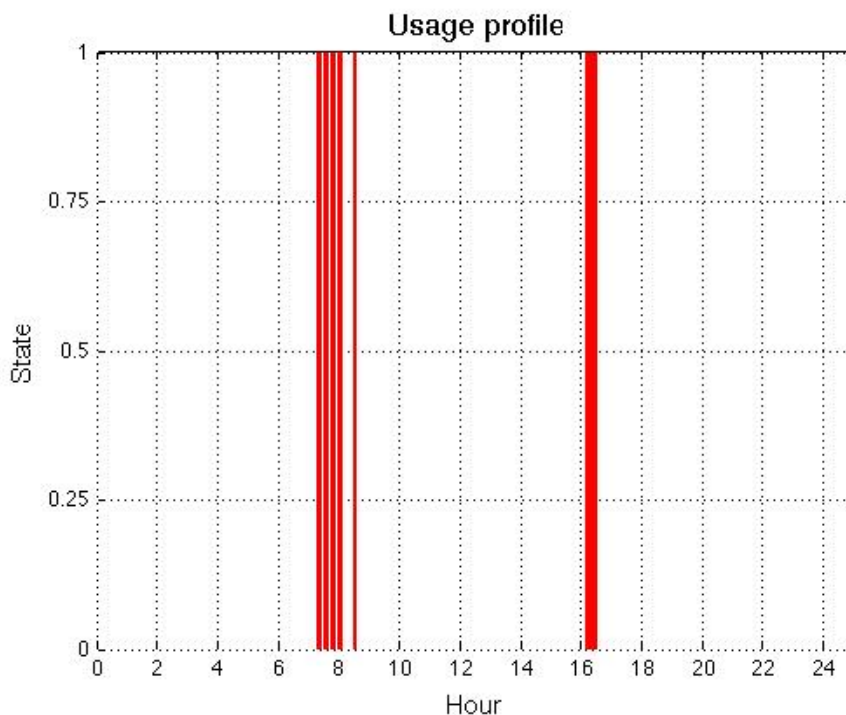
#### 7.2.1 Parametri dell'algoritmo di predizione

Come descritto nel Capitolo 4, l'algoritmo di predizione è caratterizzato da alcuni parametri che ne determinano le prestazioni:

- *parametro di cluster*;

- *filtro sulle gaussiane;*
- *varianza delle gaussiane;*
- *periodo di training;*

Per tale motivo abbiamo deciso di condurre alcune simulazioni al fine di individuare il valore ottimo dei primi tre parametri, mentre per il periodo di training si è utilizzato una finestra temporale di 28 giorni, rimandando al Paragrafo 7.2.3 per le motivazioni alla base di tale scelta. Durante tali test abbiamo generato dati d'utilizzo di un singolo elettrodomestico ricreando in particolare uno scenario che fosse significativo per la valutazione dell'algoritmo: abbiamo inserito accensioni e spegnimenti ravvicinati tra le 7:00 e le 8:00 e un'accensione isolata alle 16:00, così come rappresentato in figura Figura 7.1 .



**Figura 7.1:** Scenario per la valutazione dell'algoritmo

A partire da tali dati, un metodo di predizione efficace avrebbe dovuto individuare, oltre a quella delle 16:00, un numero di accensioni tra le 7:00 e le 8:00 che non fosse né troppo basso (perché non rappresentativo dei dati iniziali), né troppo grande (al fine del sistema proposto non è necessaria una rappresentazione eccessivamente dettagliata delle attività degli elettrodomestici).

Inizialmente abbiamo condotto dei test per valutare congiuntamente il valore ottimo da assegnare al *parametro di cluster* e di *filtro sulle gaussiane*. Tali parametri infatti, così come possibile intuire da quanto descritto nel Capitolo 4 sono tra loro correlati. Per i test in particolare abbiamo considerato le seguenti combinazioni di valori (Tabella 7.1):

Parametro di clustering	Parametro di filtro
0.25 (15 minuti)	0.25 (15minuti)
0.5 (30 minuti)	0.5 (30 minuti)
0.25 (15 minuti)	0.5 (30 minuti)
0.083 (5 minuti)	0.083 (5 minuti)
0.083 (5 minuti)	0.17 (10 minuti)
0.083 (5 minuti)	0.25 (15 minuti)
0.17 (10 minuti)	0.17 (10 minuti)
0.17 (10 minuti)	0.33 (20 minuti)
0.17 (10 minuti)	0.5 (30 minuti)
0.75 (45 minuti)	0.5 (30 minuti)
0.75 (45 minuti)	1 (1 ora)
0.75 (45 minuti)	1.5 (1 ora e 30 minuti)
1 (1 ora)	0.5 (30 minuti)
1 (1 ora)	1 (1 ora)
1 (1 ora)	2 (2 ore)

**Tabella 7.1:** Valori testati per parametro di clustering e di filtro

Per le prime tre combinazioni della tabella (Tabella 7.1) sono stati riportati tutti i grafici di ogni chiamata ricorsiva della funzione *somma\_gaussiana* (si veda da Figura 7.2 a Figura 7.7).

Considerando il terzo caso in cui la soglia del parametro di cluster è 0.25 (sono stati raggruppati nello stesso cluster gli orari di accensione distanti tra loro al massimo 15 minuti) e il filtro delle gaussiane 0.5 (sono state eliminate le gaussiane la cui media è compresa in un intervallo di 30 minuti rispetto all'ora di accensione trovata) analizziamo le varie iterazioni eseguite dall'algoritmo.

Alla prima iterazione è stato individuato il primo orario di accensione equivalente a 7:50. Come si nota nel passaggio dalla Figura 7.2 alla Figura 7.3 sono state eliminate le gaussiane corrispondenti all'orario di accensione trovato.

Alla seconda iterazione è stata trovata l'ora 7:26, alla terza 16:04, alla quarta ed ultima iterazione 7:38.

Come si evince dai grafici il terzo caso risulta essere il migliore poiché riesce ad individuare tutti gli orari di accensione.

Il primo caso trova troppi orari di accensione poiché non riesce a raggruppare gli orari vicini tra di loro. Come si nota dalla Figura 7.6 dopo 5 iterazioni l'algoritmo non ha ancora terminato il calcolo.

Il secondo caso invece non riesce a trovare esattamente tutti gli orari di accensione determinando infatti solo 2 orari di accensione: 7:55 e 16:04.

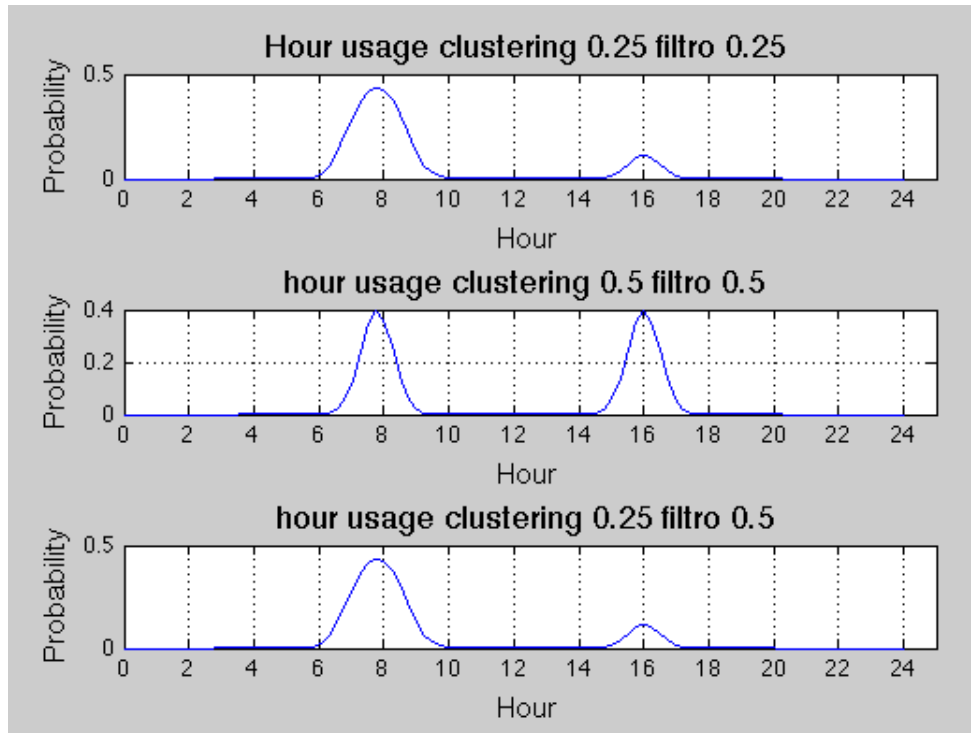


Figura 7.2: Primo test sui parametri prima iterazione

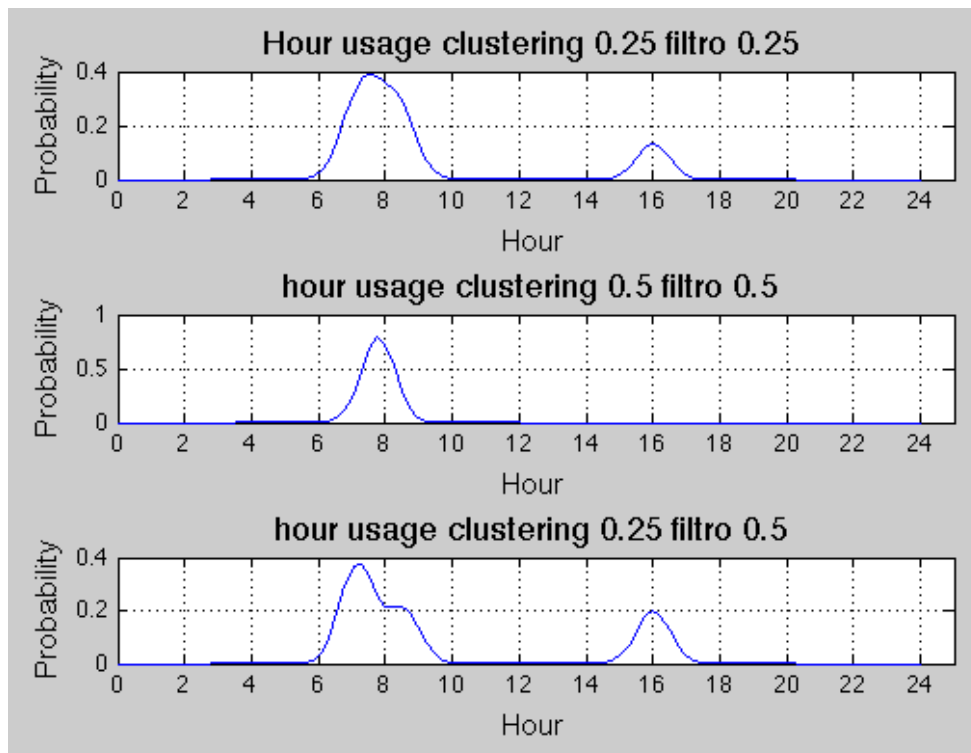


Figura 7.3: Primo test sui parametri seconda iterazione

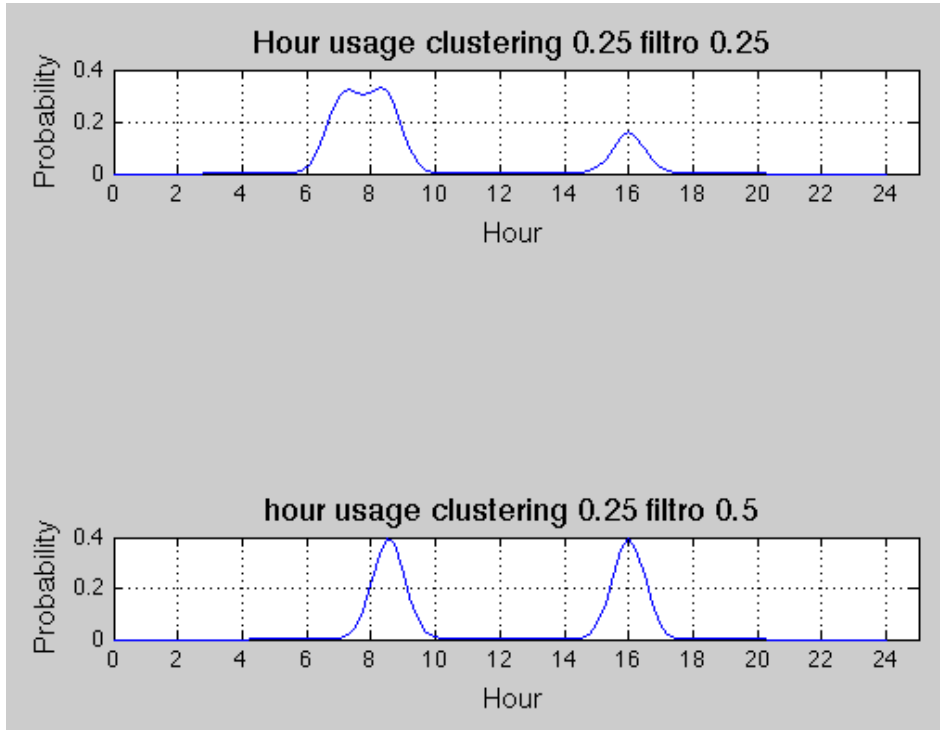


Figura 7.4: Primo test sui parametri terza iterazione

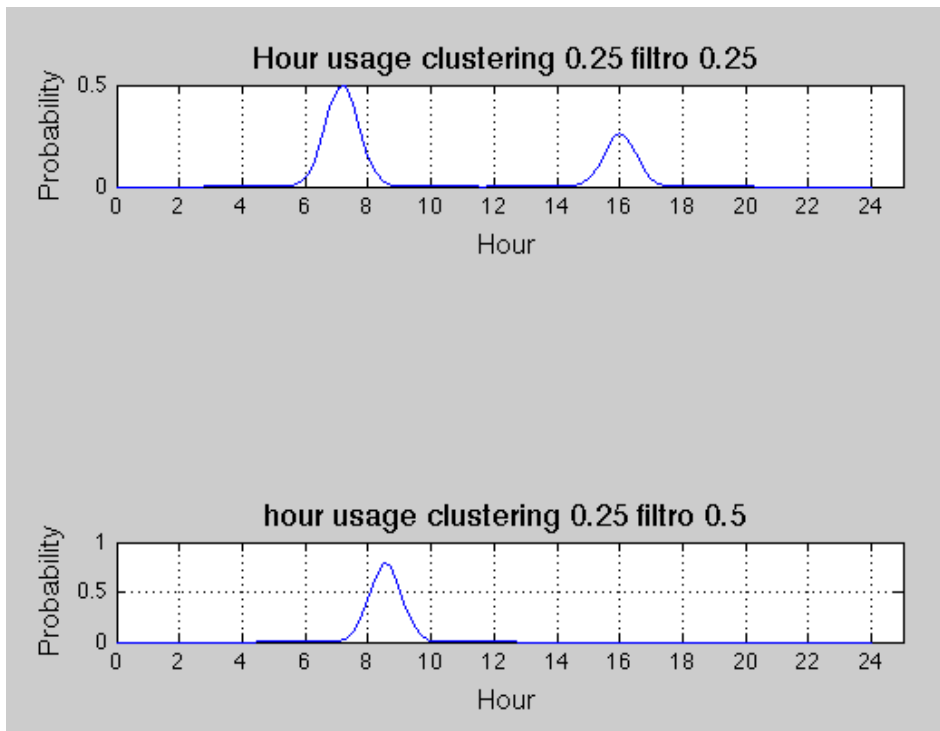


Figura 7.5: Primo test sui parametri quarta iterazione

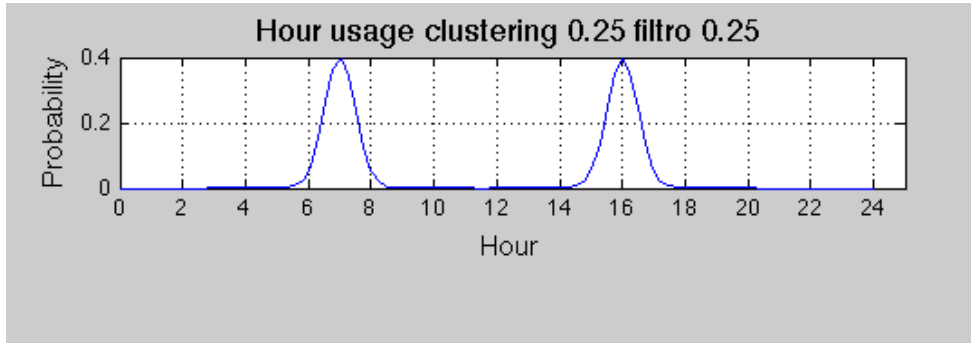


Figura 7.6: Primo test sui parametri quinta iterazione

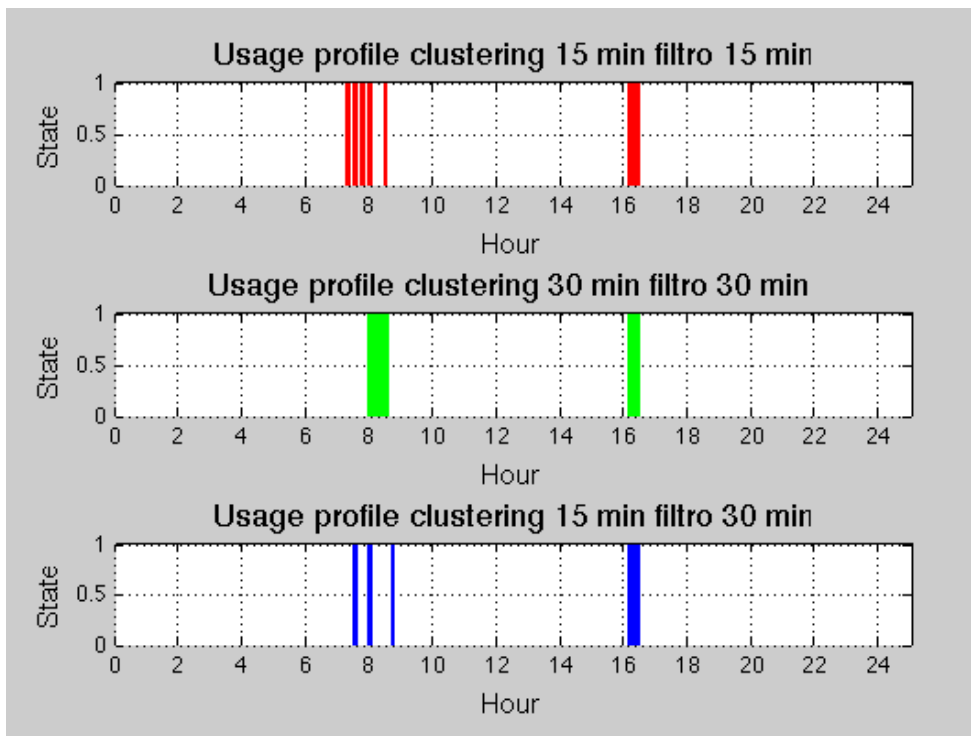


Figura 7.7: Primo test sui parametri risultati finali

La Figura 7.7 rappresenta una sintesi visuale con gli orari di accensione trovati nei tre casi analizzati.

Per quanto riguarda, invece, le altre combinazioni di valori rappresentate nella tabella Tabella 7.1 riportiamo di seguito solo il grafico con i risultati finali. Per esse sono stati effettuati ulteriori test variando il valore dei parametri: inizialmente scegliendo valori minori rispetto al caso precedente, successivamente utilizzando valori maggiori. Si veda da Figura 7.8 a Figura 7.11.

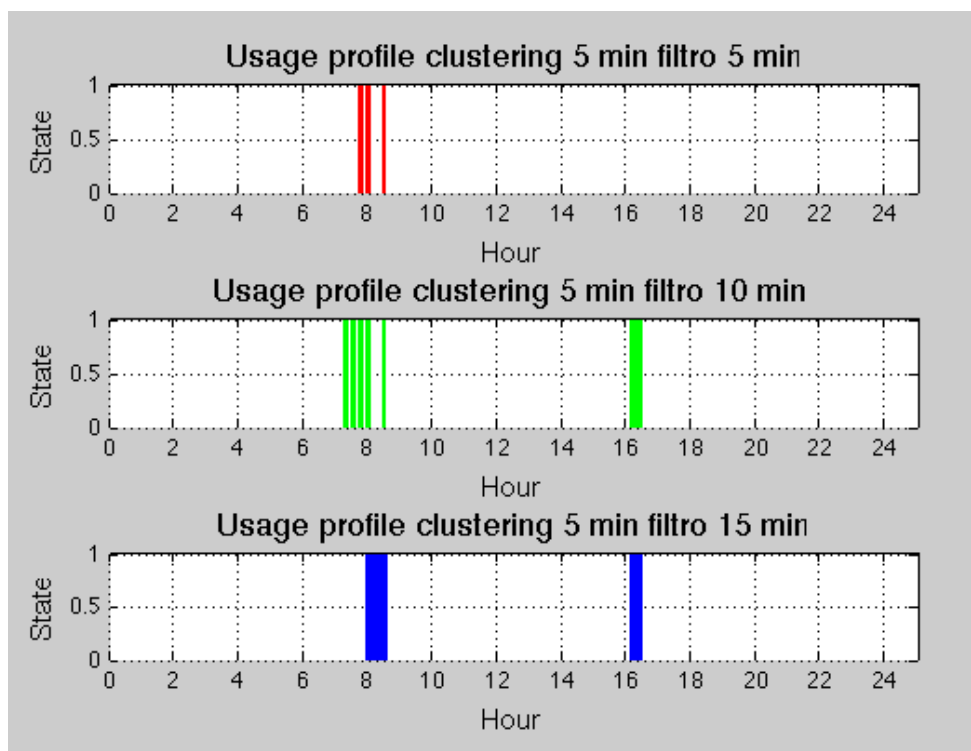


Figura 7.8: Secondo test sui parametri risultati finali



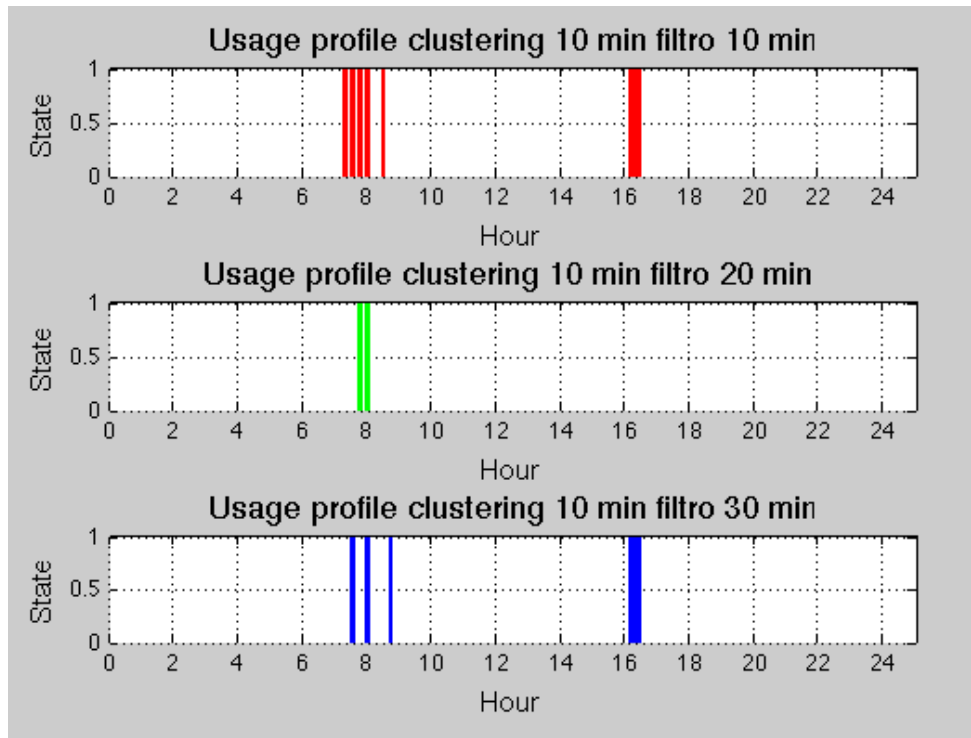


Figura 7.9: Terzo test sui parametri risultati finali

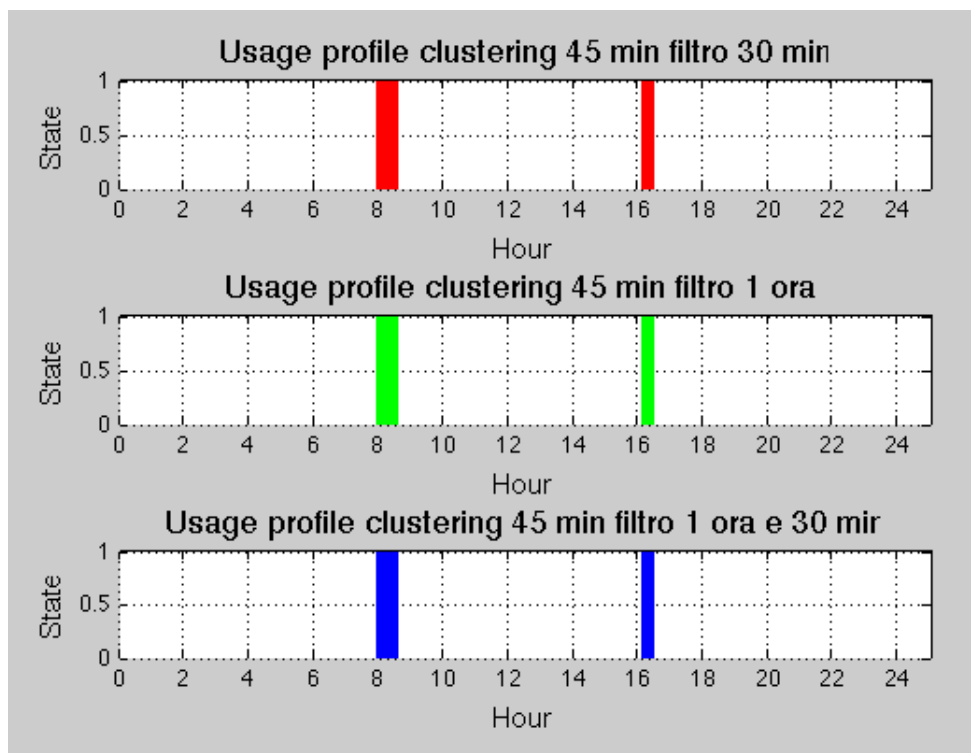
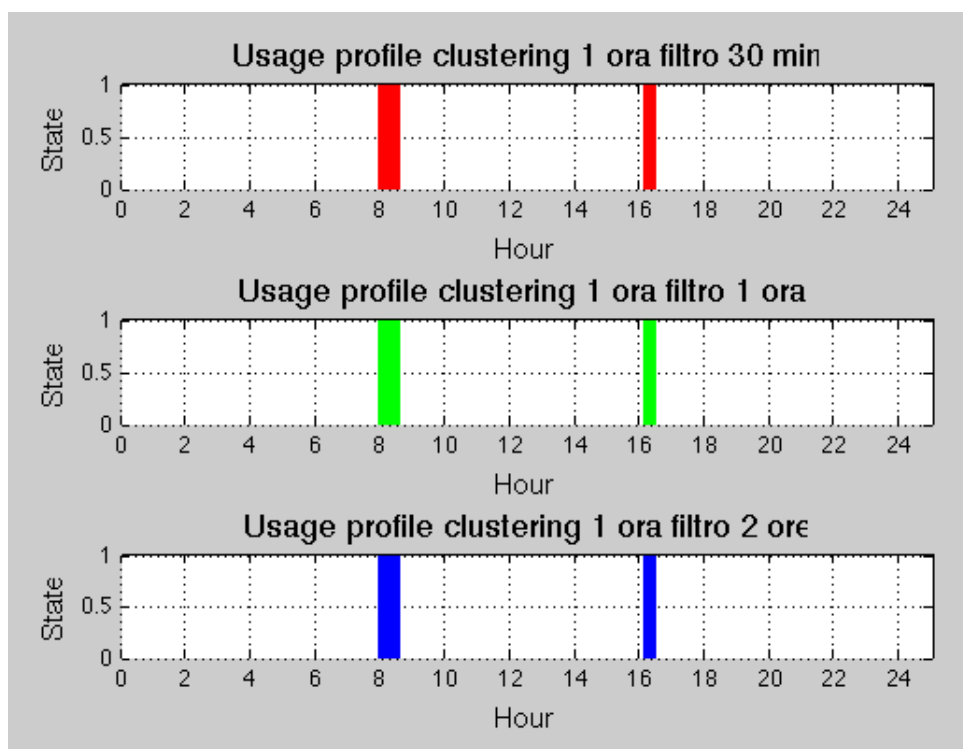


Figura 7.10: Quarto test sui parametri risultati finali

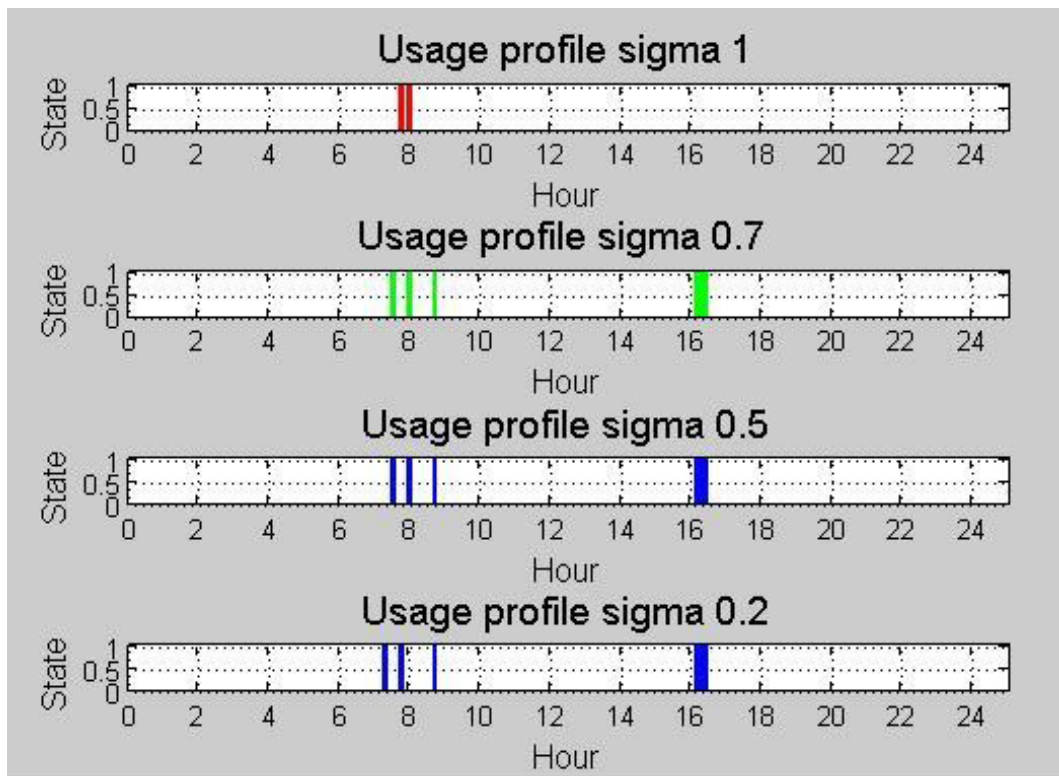


**Figura 7.11:** Quinto test sui parametri risultati finali

Analizzando in modo più approfondito i dati si può notare come un utilizzo di un parametro di cluster troppo basso (5 o 10 minuti) non riesca a raggruppare e a calcolare la media degli orari di accensione ravvicinati, mentre la scelta di un valore alto (30 minuti o 1 ora) approssima in modo poco preciso la predizione sull'ora di accensione. Lo stesso ragionamento può essere fatto per il valore del filtro: un valore troppo basso (ad esempio 5 minuti) causa il non rilevamento di alcuni orari di accensione, al contrario, un valore troppo alto porta al raggruppamento di orari di accensione molto distanti tra di loro.

In conclusione, come si evince dai grafici, per quanto riguarda il valore di questi due parametri la configurazione migliore trovata è la seguente: parametro di cluster= 0.25 (un quarto d'ora) e di filtro= 0.5 (mezz'ora).

Definiti tali parametri sono stati effettuati test sull'algoritmo per diversi valori di varianza delle gaussiane (1, 0.7, 0.5, 0.2). I risultati ottenuti sono rappresentati in Figura 7.12.



**Figura 7.12:** Test sul parametro sigma

Dai grafici è possibile notare che una varianza unitaria determina maggiori errori di predizione (in particolare manca l'accensione delle 16:00), mentre per gli altri valori di tale parametro non si evidenziano differenze significative. Nel seguito utilizzeremo la varianza 0.5 per la realizzazione dei test.

### 7.2.2 Scenari

I test svolti riguardano in particolare tre elettrodomestici: il condizionatore, il forno e la lavastoviglie. Per ogni elettrodomestico abbiamo ipotizzato alcuni scenari di utilizzo:

- *regolare*: viene ipotizzato un utilizzo regolare durante tutto l'anno. Per quanto riguarda il caso della lavastoviglie i dati relativi sono stati generati in questo modo: si è ipotizzato che venisse utilizzata 2 giorni alla settimana, in particolare il lunedì e il giovedì, con una sola accensione al giorno. Durante il week-end si è previsto un utilizzo più intensivo: sia il sabato che la domenica per 2 volte al giorno;
- *cambiamenti settimanali ON OFF*: vengono alternate una settimana di utilizzo e una settimana nella quale l'apparecchio in questione non viene mai acceso;
- *eccezioni casuali*: all'interno di un utilizzo regolare sono state inserite alcune eccezioni casuali (giorni nei quali l'elettrodomestico non viene utilizzato);
- *cambiamento a metà anno*: le abitudini di utilizzo vengono variate a metà dell'anno considerato. Per la lavastoviglie durante i primi 200 giorni le modalità di utilizzo sono uguali al caso regolare. A partire dal giorno 201 la lavastoviglie è stata utilizzata il mercoledì e il venerdì mentre il week-end è rimasto invariato;
- *periodo di vacanza*: all'interno di un comportamento regolare sono stati inseriti quindici giorni consecutivi durante i quali l'elettrodomestico non viene mai utilizzato.

È importante sottolineare che anche nel caso di scenario regolare gli orari di accensione di ogni giorno non sono identici poiché sono stati creati con un generatore di numeri casuali gaussiani: fissato il valore medio gli orari generati

sono compresi nell'intervallo di un quarto d'ora. Con la stessa procedura sono stati generati anche gli orari di spegnimento.

### 7.2.3 Periodo di training

Nei test svolti precedentemente è sempre stato utilizzato un periodo di training di 28 giorni. Tuttavia, al fine di ottimizzare le prestazioni dell'algoritmo di predizione abbiamo deciso di svolgere alcuni test per verificare se la scelta effettuata sia corretta o meno identificando eventualmente un valore migliore per questo parametro. Nel presentare i risultati ricavati si terrà conto del solo errore sulla predizione d'utilizzo, in quanto più importante nel nostro caso rispetto all'errore sull'ora e sulla durata. Nel seguito in particolare faremo sempre riferimento al solo caso della lavastoviglie, utilizzando gli scenari descritti precedentemente, con periodo di training 11, 21, 28, 45 e 60 giorni.

1. *Periodo di training di 11 giorni:* dai grafici riportati si nota che tale periodo risulta troppo breve e l'algoritmo non è in grado di apprendere le abitudini dell'utente. L'errore commesso risulta essere maggiore del 50% per tutti gli scenari, l'algoritmo calcola infatti una predizione errata per 200 giorni l'anno (si veda da Figura 7.13 a Figura 7.17).



Figura 7.13: Test lavastoviglie regolare considerando un periodo di training di 11 giorni

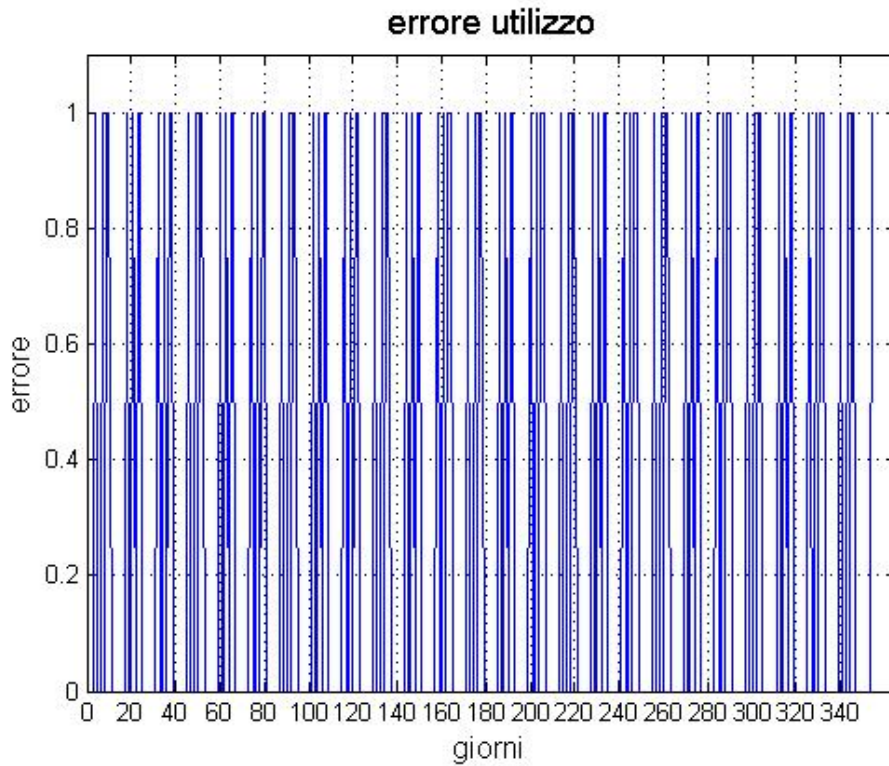


Figura 7.14: Test lavastoviglie ON OFF considerando un periodo di training di 11 giorni

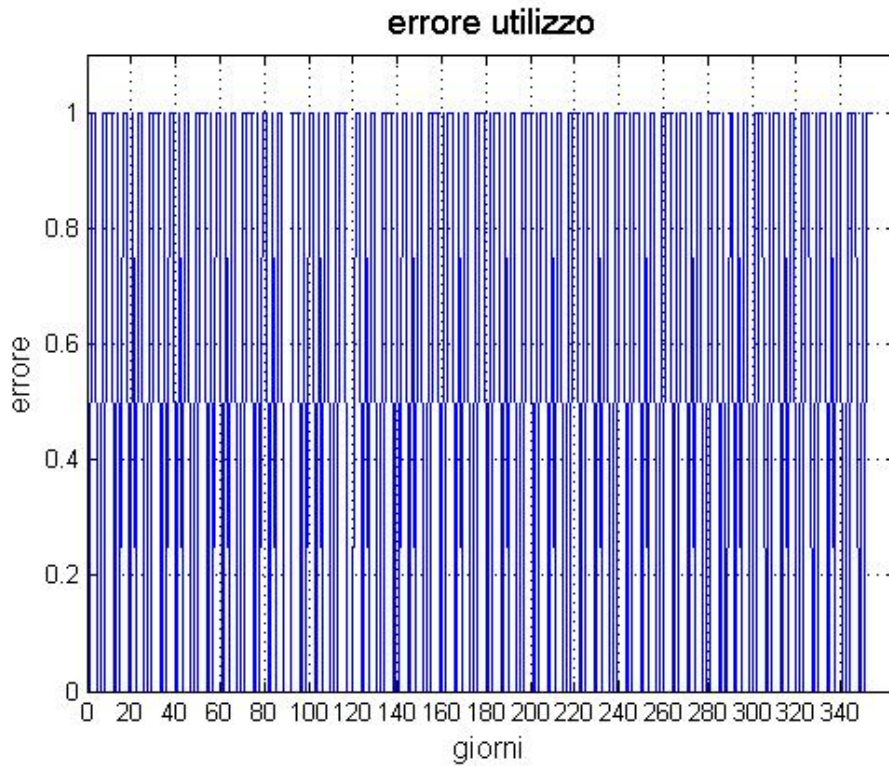
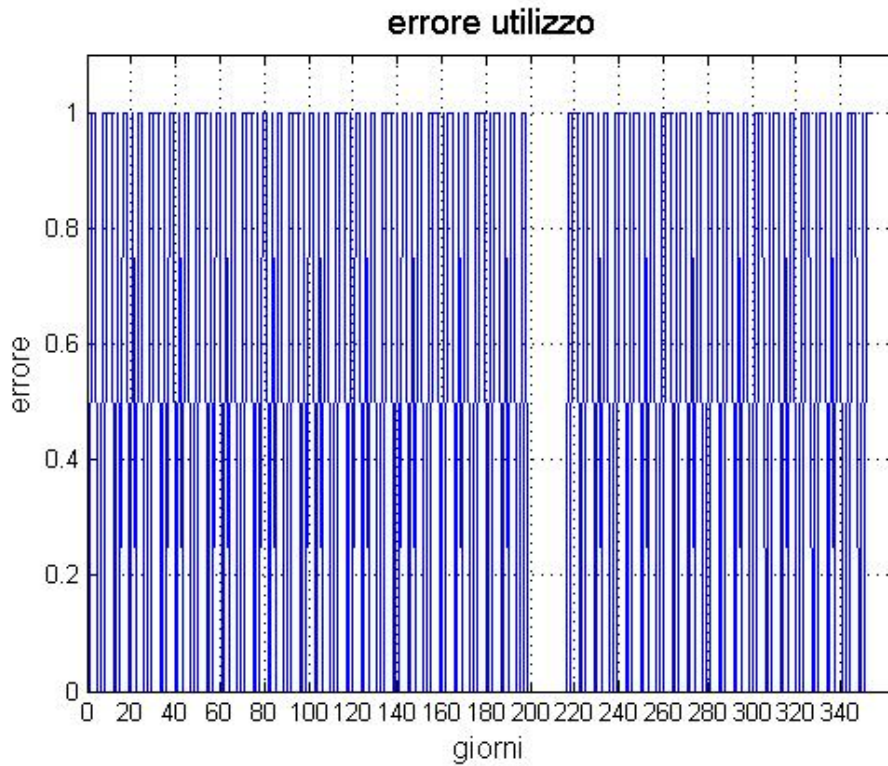


Figura 7.15: Test lavastoviglie con eccezioni casuali considerando un periodo di training di 11 giorni



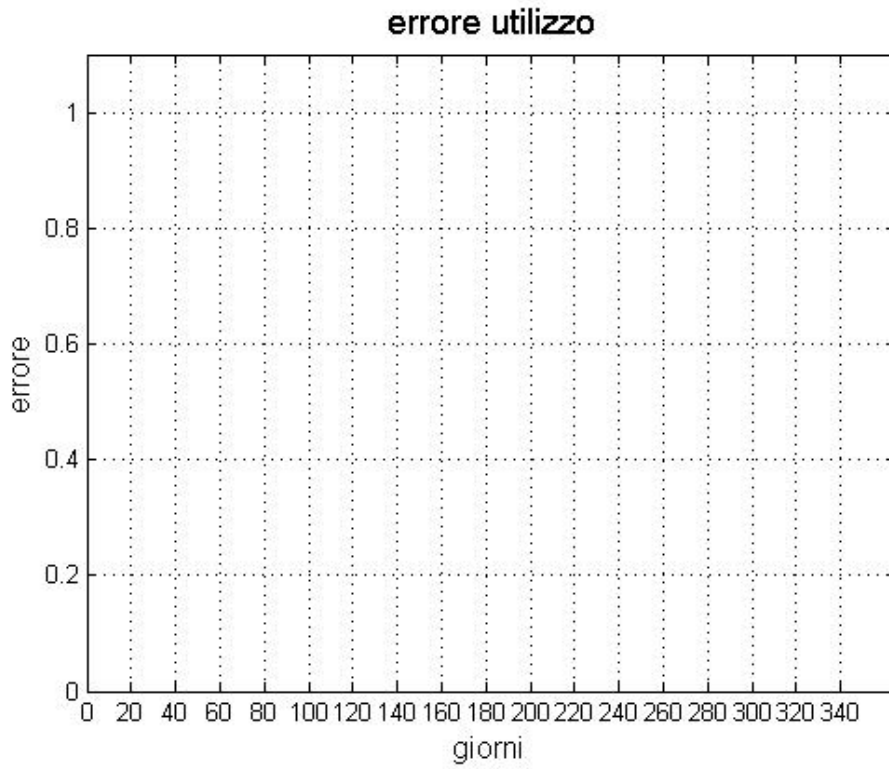
Figura 7.16: Test lavastoviglie con cambio abitudini a metà anno considerando un periodo di training di 11 giorni



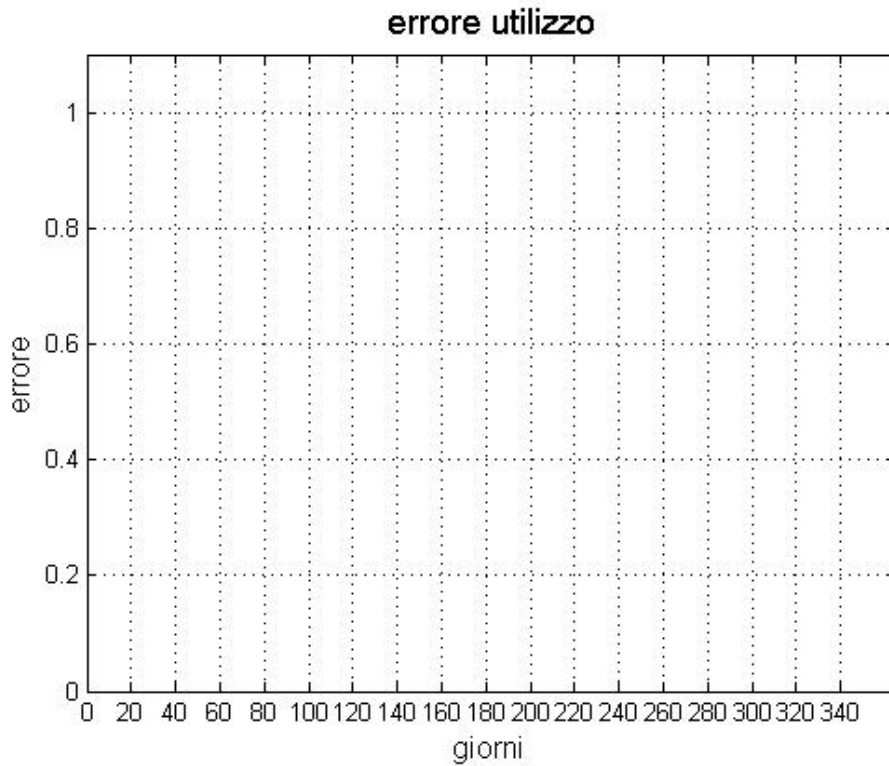
**Figura 7.17:** test lavastoviglie con periodo di vacanza considerando un periodo di training di 11 giorni

2. *Periodo di training di 21 giorni:* si evidenzia un netto miglioramento rispetto al caso precedente. Per quanto riguarda il caso regolare e di cambiamenti settimanali ON OFF l'errore passa a 0% mentre negli altri casi non supera mai il 5% (si veda da Figura 7.18 a Figura 7.22);

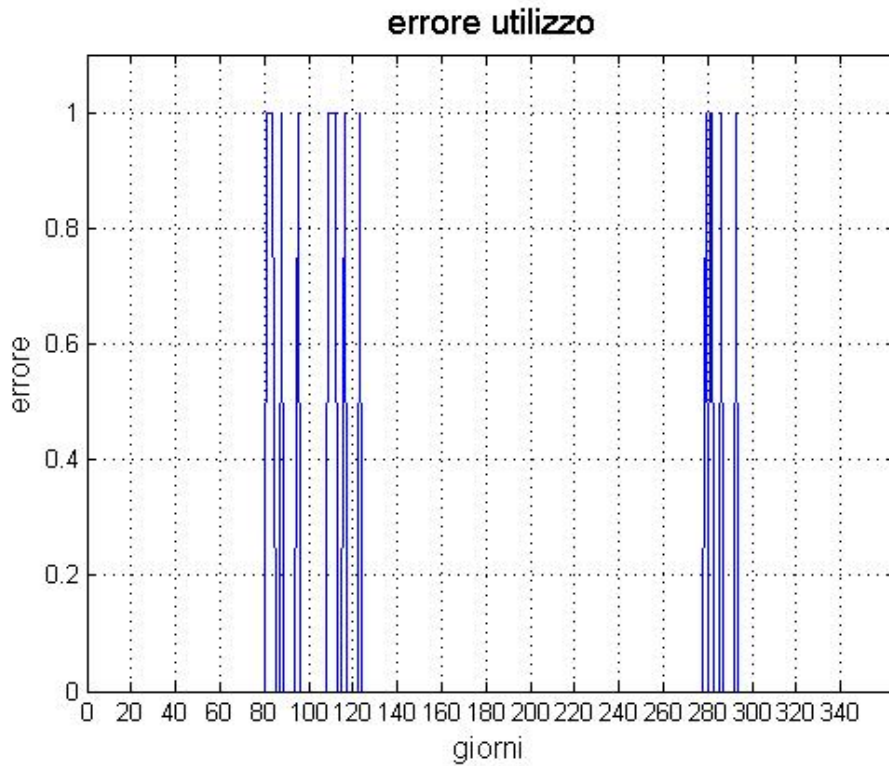




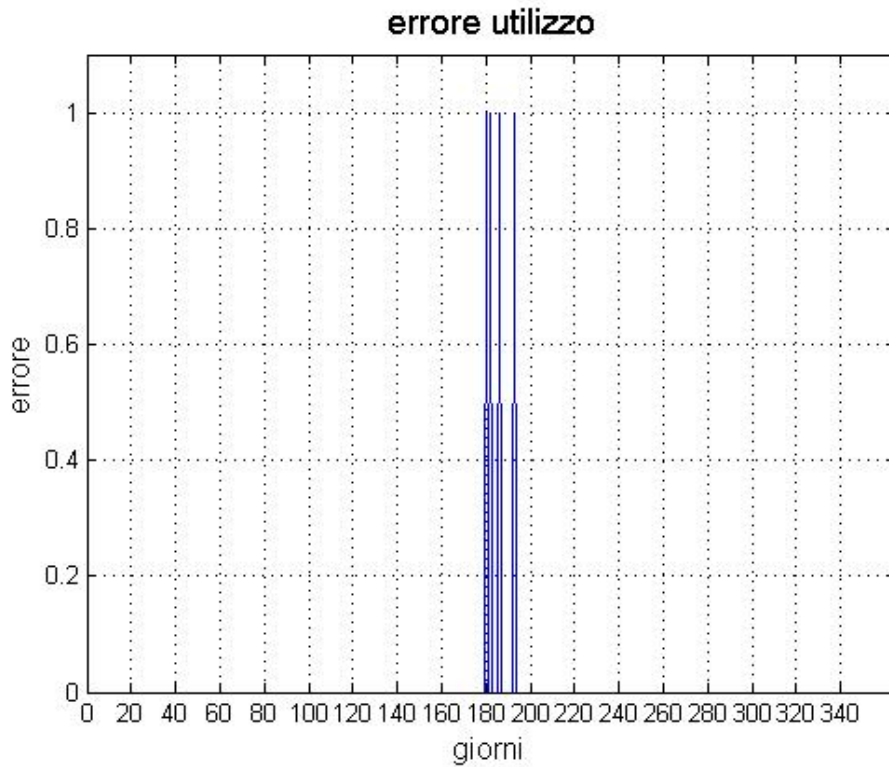
**Figura 7.18:** Test lavastoviglie regolare considerando un periodo di training di 21 giorni



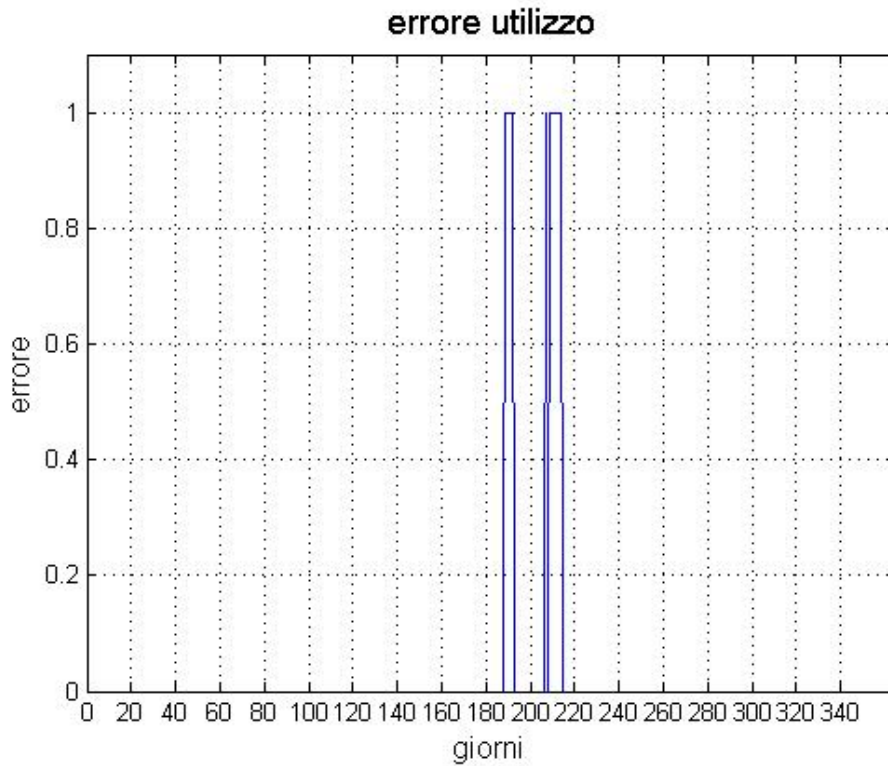
**Figura 7.19:** Test lavastoviglie ON OFF considerando un periodo di training di 21 giorni



**Figura 7.20:** Test lavastoviglie con eccezioni casuali considerando un periodo di training di 21 giorni



**Figura 7.21:** Test lavastoviglie con cambio abitudini a metà anno considerando un periodo di training di 21 giorni



**Figura 7.22:** Test lavastoviglie con periodo di vacanza considerando un periodo di training di 21 giorni

3. *Periodo di training di 28 giorni:* rispetto al caso precedente si nota un ulteriore miglioramento nei casi di eccezioni casuali, nello scenario con cambio di abitudini a metà anno e periodo di vacanza (si veda da Figura 7.23 a Figura 7.27);

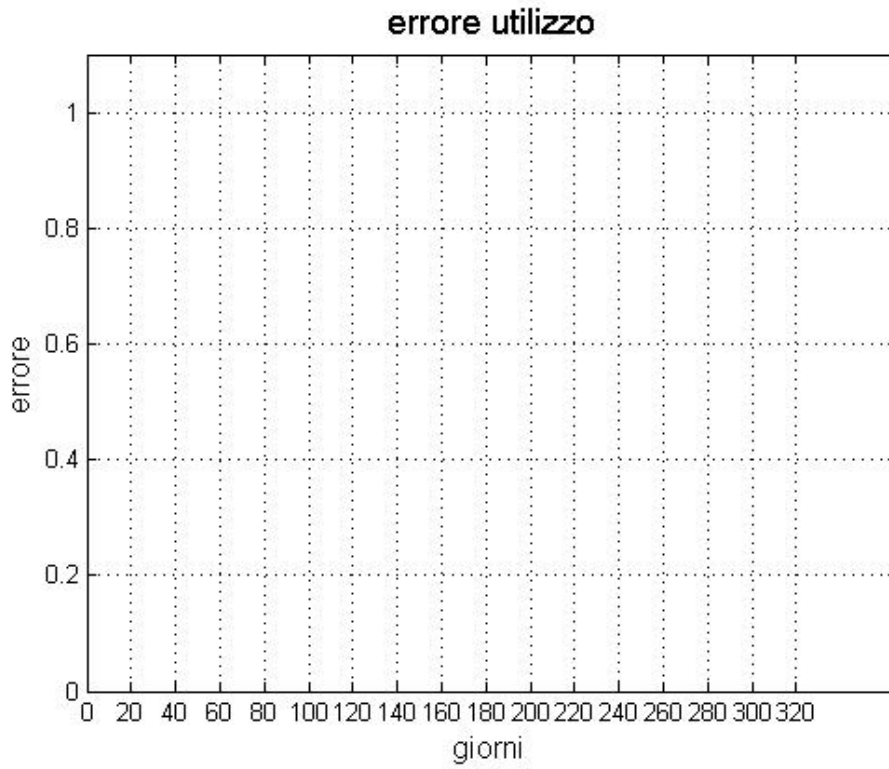


Figura 7.23: Test lavastoviglie regolare considerando un periodo di training di 28 giorni

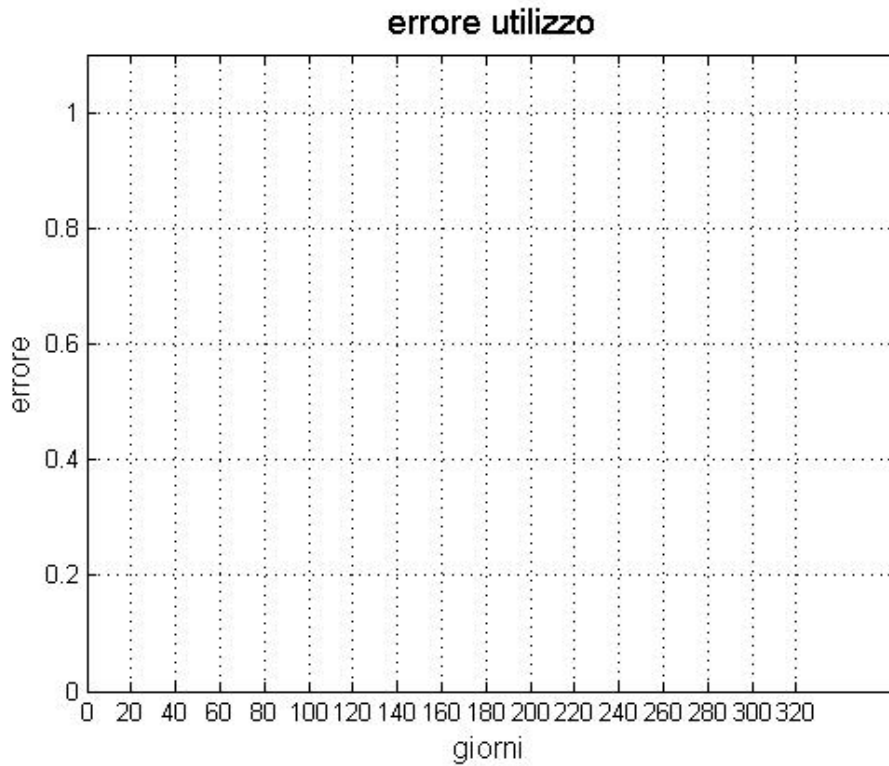
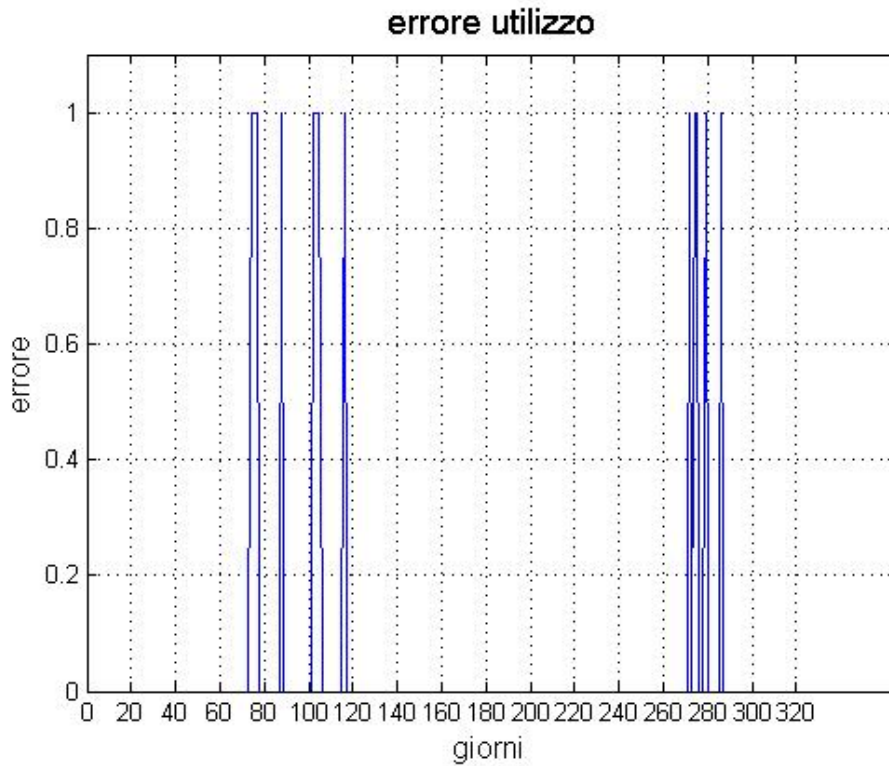
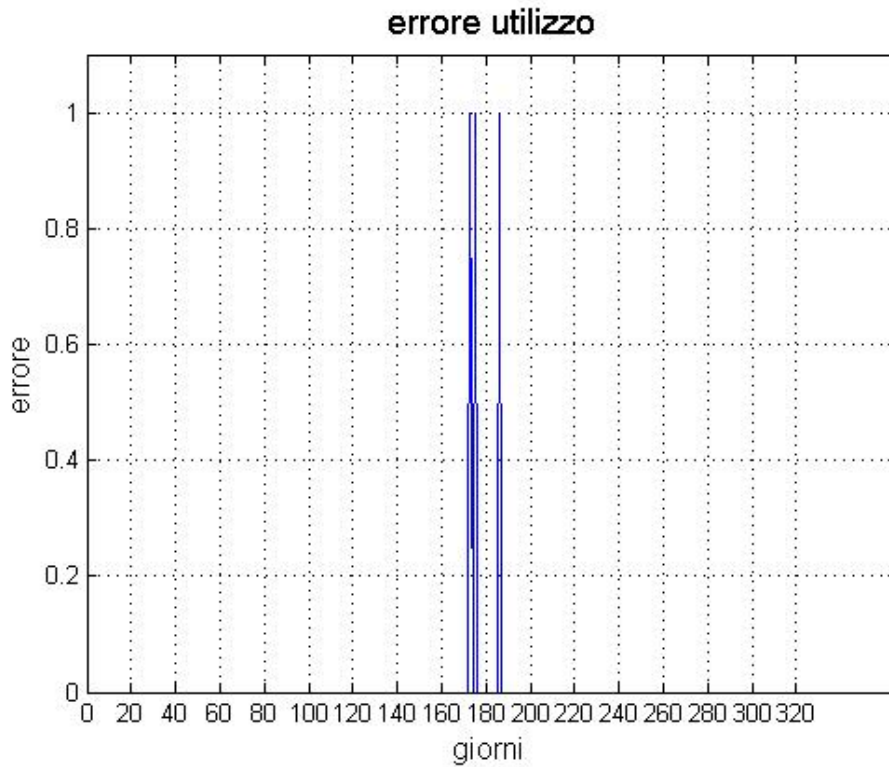


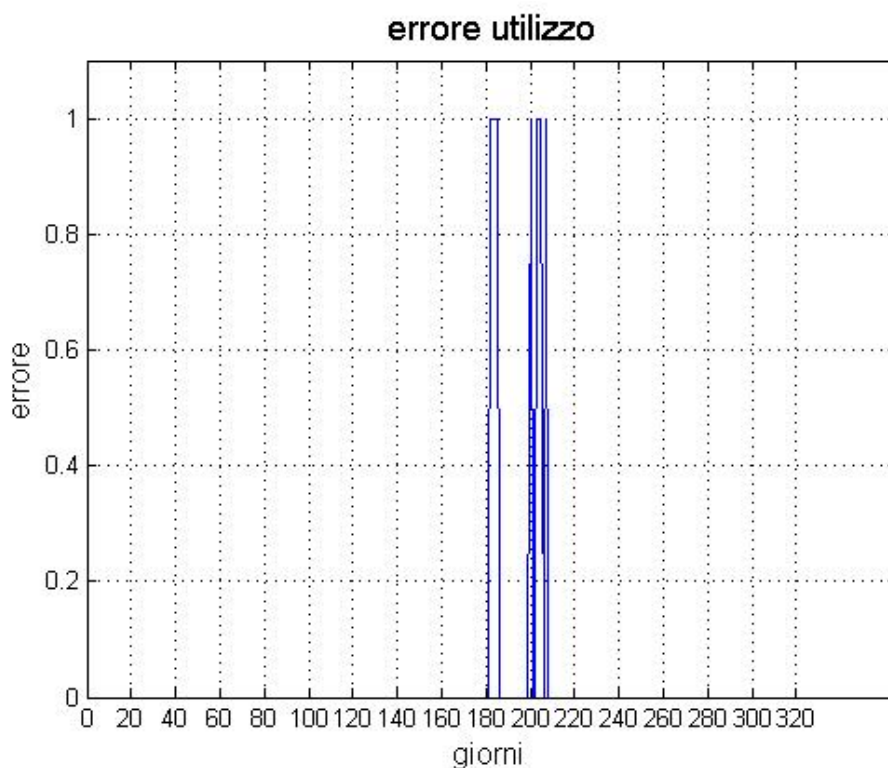
Figura 7.24: Test lavastoviglie ON OFF considerando un periodo di training di 28 giorni



**Figura 7.25:** Test lavastoviglie con eccezioni casuali considerando un periodo di training di 28 giorni



**Figura 7.26:** Test lavastoviglie con cambio abitudini a metà anno considerando un periodo di training di 28 giorni



**Figura 7.27:** Test lavastoviglie con periodo di vacanza considerando un periodo di training di 28 giorni

4. *Periodo di training di 45 giorni:* come previsto, è possibile osservare un nuovo miglioramento nella predizione di utilizzo. In questo caso lo scenario con errore maggiore risulta essere sempre quello che contiene eccezioni isolate con una percentuale di errore uguale a 2.50%. Questo miglioramento rispetto al caso con training di 28 giorni, nel quale l'errore risulta essere minore del 5%, può essere trascurato. Inoltre, oltre a non portare vantaggi significativi, per i primi 45 giorni di utilizzo l'algoritmo non sarà in grado di fare predizione con la conseguenza che l'utente non potrà visionare i risultati. Di seguito vengono riportati i grafici che mostrano l'andamento dell'errore nel caso di periodo di training uguale a 45 giorni (si veda da Figura 7.28 a Figura 7.32).

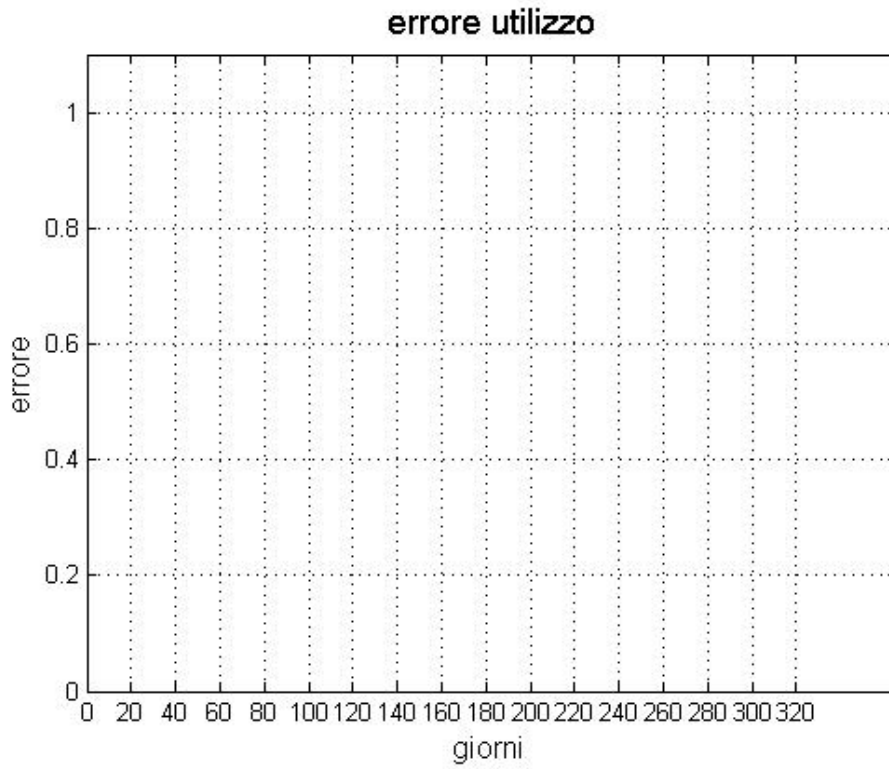


Figura 7.28: Test lavastoviglie regolare considerando un periodo di training di 45 giorni

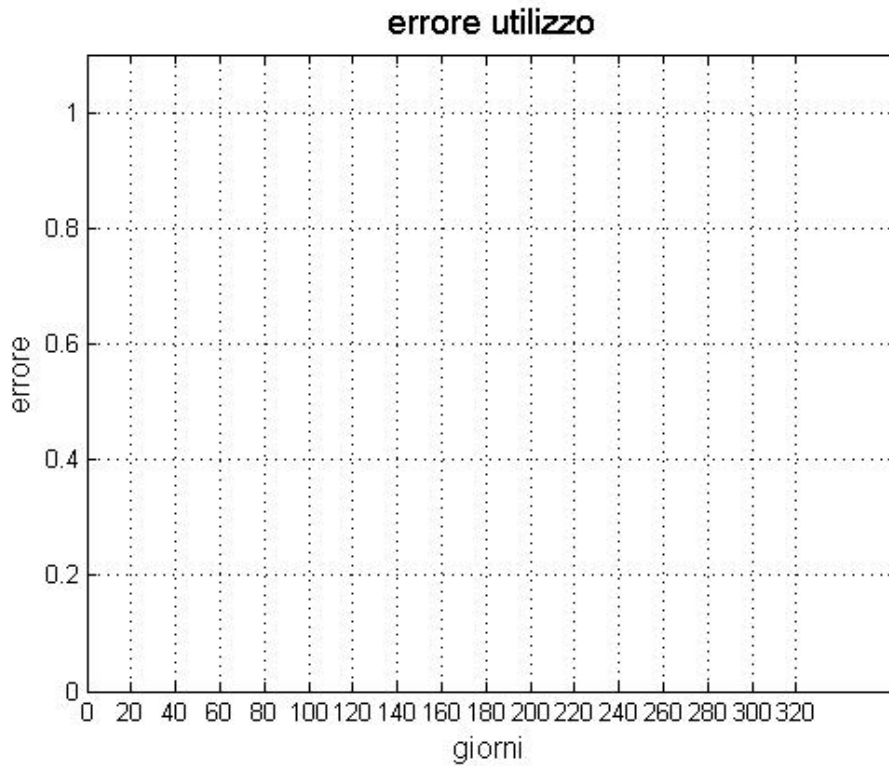
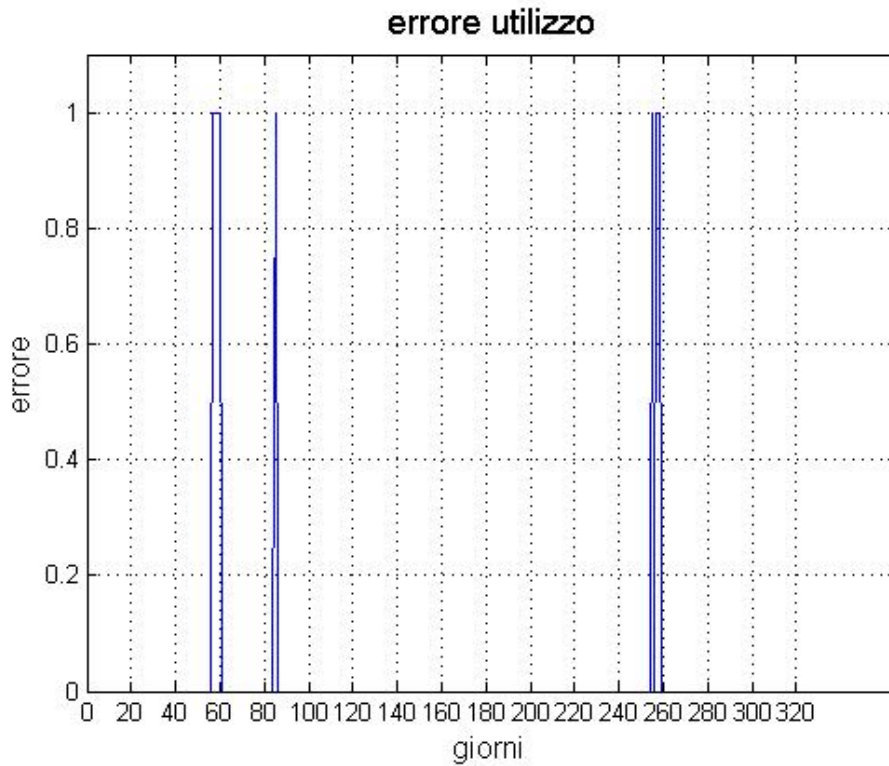
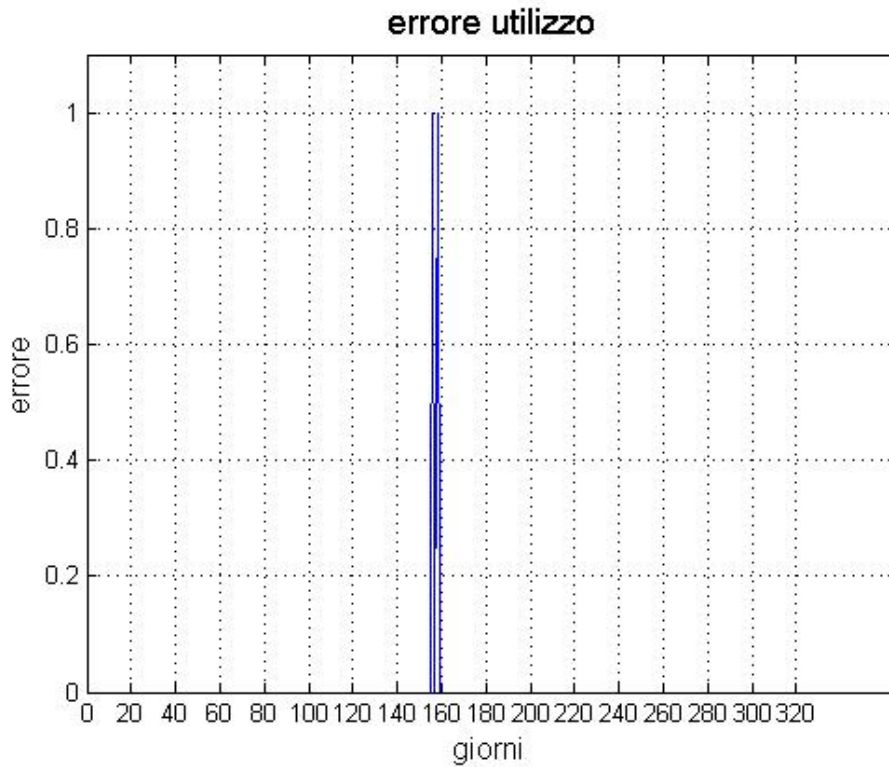


Figura 7.29: Test lavastoviglie ON OFF considerando un periodo di training di 45 giorni



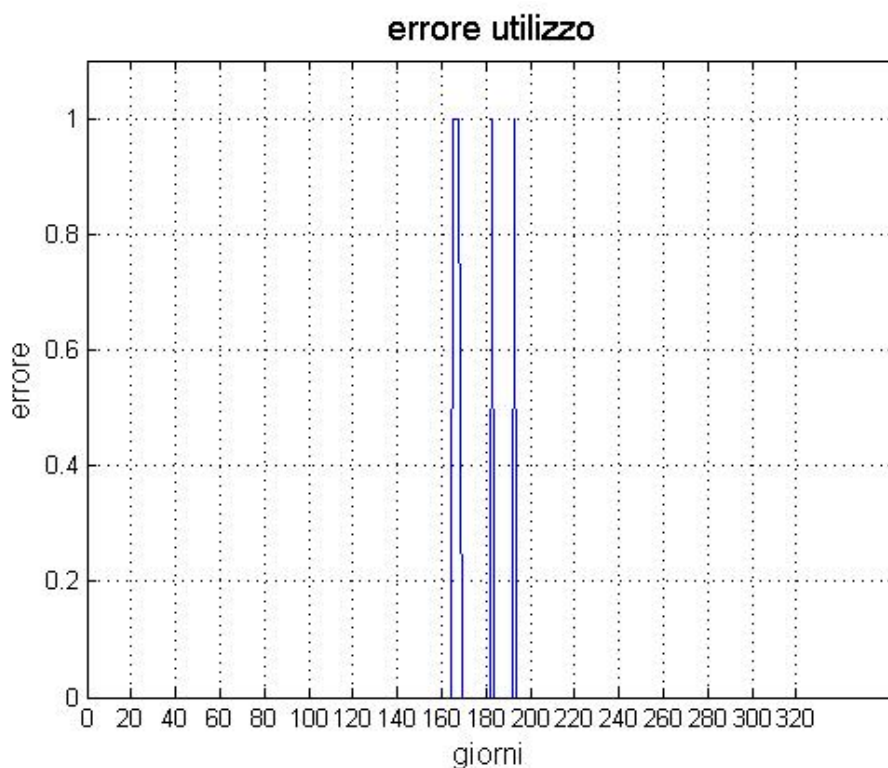


**Figura 7.30:** Test lavastoviglie con eccezioni casuali considerando un periodo di training di 45 giorni



**Figura 7.31:** Test lavastoviglie con cambio abitudini a metà anno considerando un periodo di training di 45 giorni





**Figura 7.32:** Test lavastoviglie con periodo di vacanza considerando un periodo di training di 45 giorni

5. *Periodo di training di 60 giorni:* dai grafici si evidenzia un leggero peggioramento delle prestazioni dell'algoritmo, infatti, sia nello scenario di eccezioni isolate che in quello con il cambiamento di abitudini a metà anno la percentuale di errore di utilizzo risulta essere maggiore ai casi precedenti. In questo caso l'algoritmo non è in grado di generalizzare in quanto l'apprendimento è stato effettuato troppo a lungo. Il modello si è adattato a caratteristiche che sono specifiche solo del training set, ma che non hanno riscontro nel resto dei casi. Siamo in presenza di un caso di *overfitting* nel quale le prestazioni sui dati di training aumentano, mentre quelle sui dati non visionati risultano peggiori. Bisogna inoltre sottolineare che la scelta di un periodo di training così lungo renderebbe inutilizzare l'algoritmo per i primi 60 giorni durante i quali avviene solo la raccolta dei dati. Di seguito è possibile osservare i grafici (da Figura 7.33 a Figura 7.37).

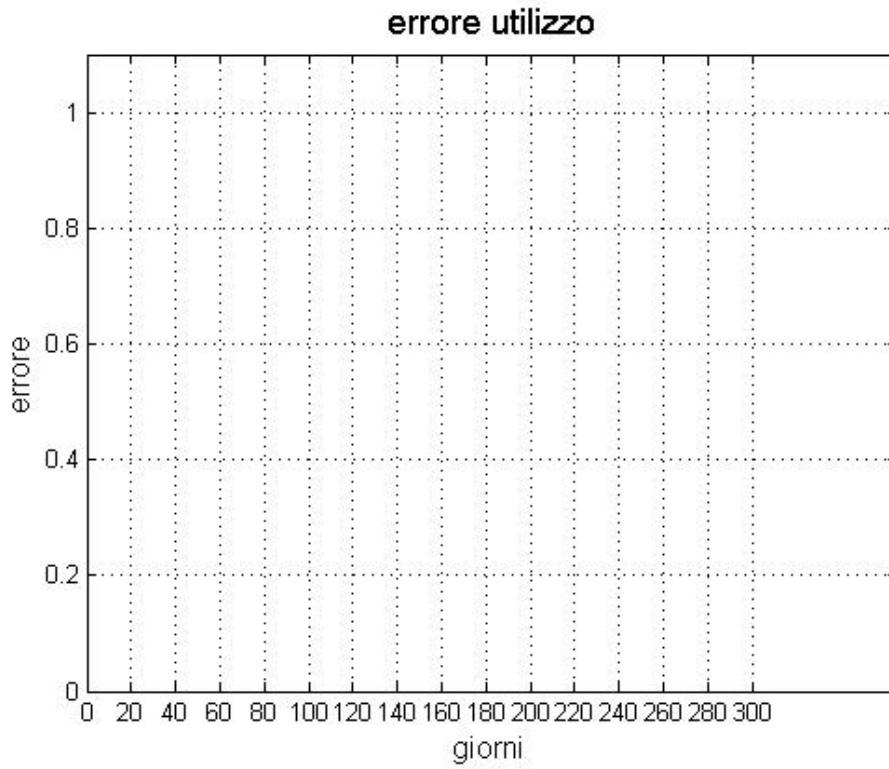


Figura 7.33: Test lavastoviglie regolare considerando un periodo di training di 60 giorni

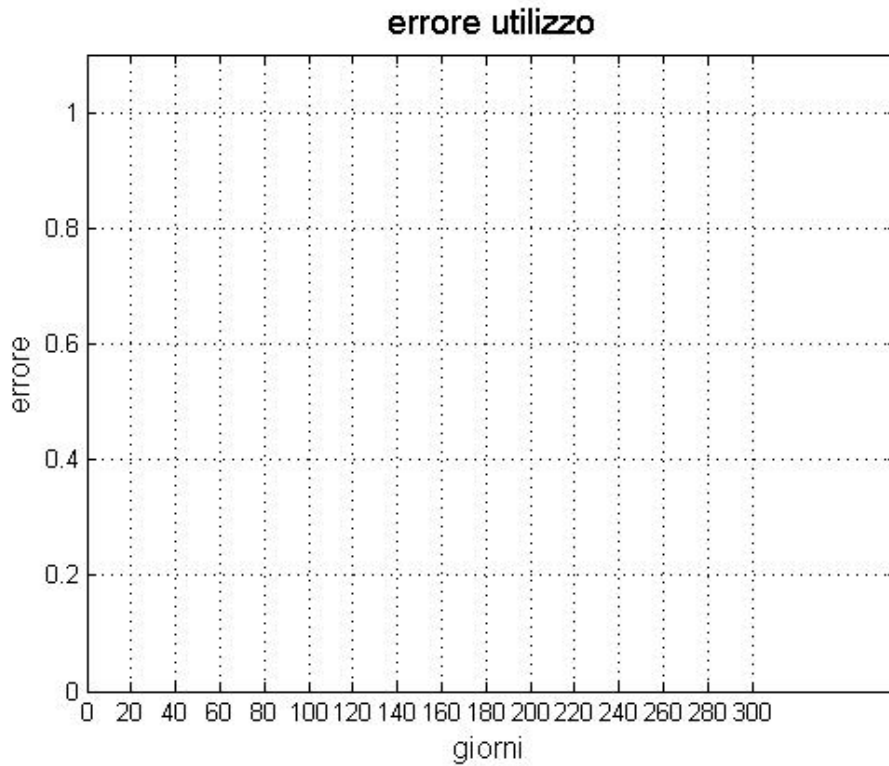
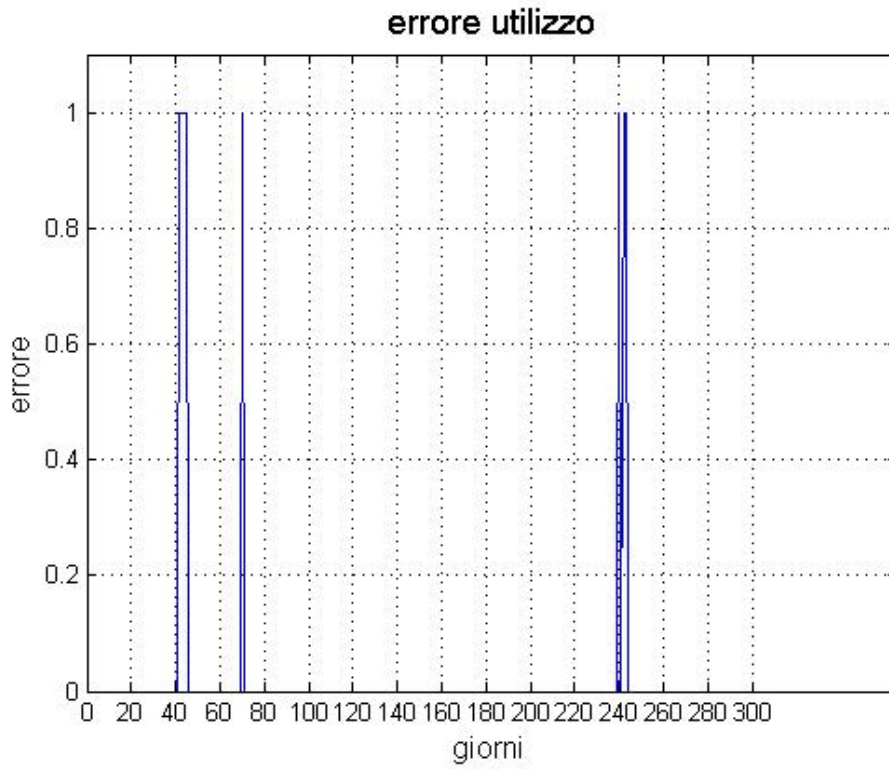
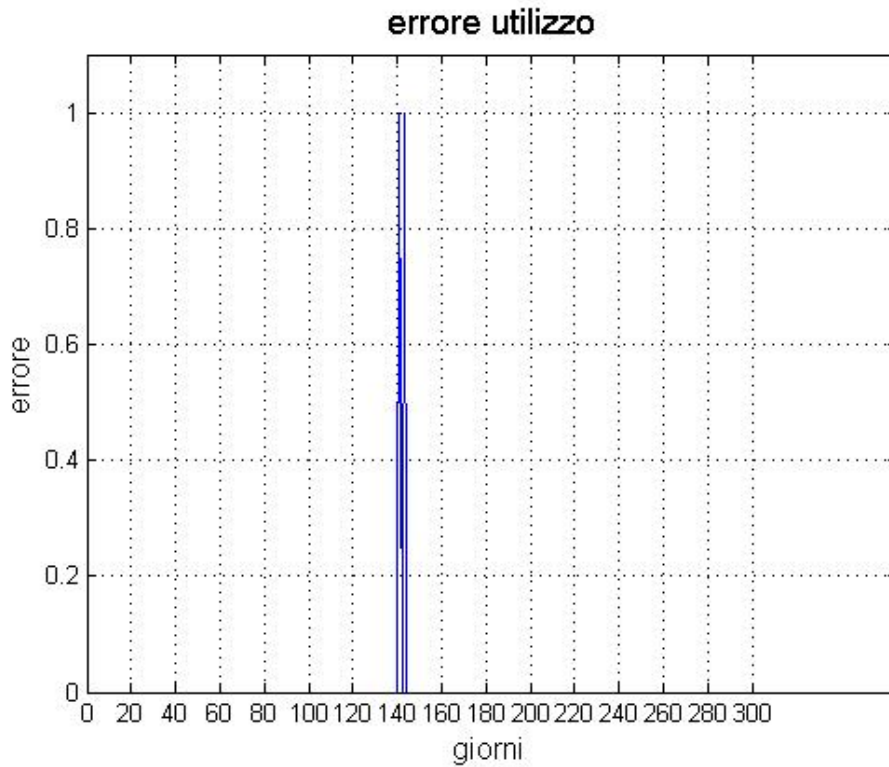


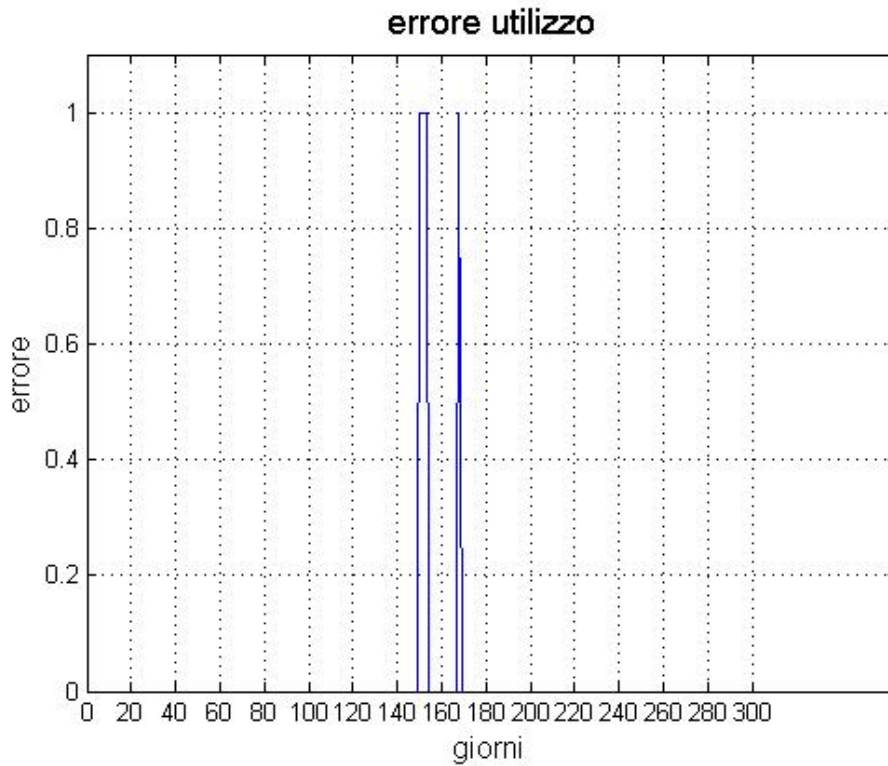
Figura 7.34: Test lavastoviglie ON OFF considerando un periodo di training di 60 giorni



**Figura 7.35:** Test lavastoviglie con eccezioni casuali considerando un periodo di training di 60 giorni



**Figura 7.36:** Test lavastoviglie con cambio abitudini a metà anno considerando un periodo di training di 60 giorni



**Figura 7.37:** Test lavastoviglie con periodo di vacanza considerando un periodo di training di 60 giorni

In conclusione si è deciso di utilizzare 28 giorni come periodo di training in quanto consente di ottenere un buon compromesso tra la durata di tale periodo e le performance offerte. In questo caso, come mostrato dalla Tabella 7.2, l'errore maggiore si ha nello scenario con eccezioni casuali, e nel periodo di vacanza dove l'errore risulta essere minore del 5%.

		Errore utilizzo	
Caso	Periodo di training	%	Giorni
Regolare	11gg	57.34	203
	21 gg	0.00	0
	28 gg	0.00	0
	45 gg	0.00	0
	60 gg	0.00	0
Eccezioni isolate (eccezioni=5)	11gg	56.50	200
	21 gg	4.94	17
	28 gg	4.45	15
	45 gg	2.50	8
	60 gg	2.62	8
Cambiamenti settimanali ON OFF	11gg	28.53	101
	21 gg	0.00	0
	28 gg	0.00	0
	45 gg	0.00	0
	60 gg	0.00	0
Cambio abitudini metà anno	11gg	57.34	203
	21 gg	1.16	4
	28 gg	0.89	3
	45 gg	0.63	2
	60 gg	0.66	2
Periodo di vacanza	11gg	54.52	193
	21 gg	3.20	11
	28 gg	2.67	9
	45 gg	1.87	6
	60 gg	1.64	5

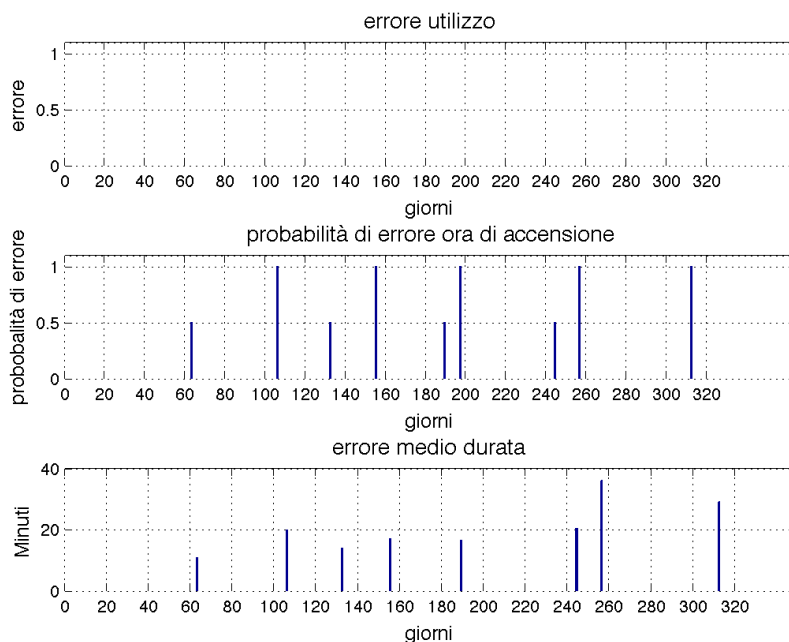
**Tabella 7.2:** Risultati dei test sul periodo di training della lavastoviglie

## 7.2.4 Prestazioni dell'algoritmo di predizione

Terminata la scelta dei parametri sono state misurate efficienza e performance dell'algoritmo. I test sono stati eseguiti considerando un periodo di training di 28 giorni e simulando l'utilizzo di tre elettrodomestici: lavastoviglie, forno e condizionatore, utilizzando gli scenari descritti precedentemente. Nei grafici seguenti vengono riportati l'errore di utilizzo, la probabilità di errore sull'ora di accensione e l'errore di durata. I giorni che appartengono al periodo di training non vengono rappresentati nei grafici pertanto l'origine dell'asse temporale corrisponde al 29° giorno dell'anno.

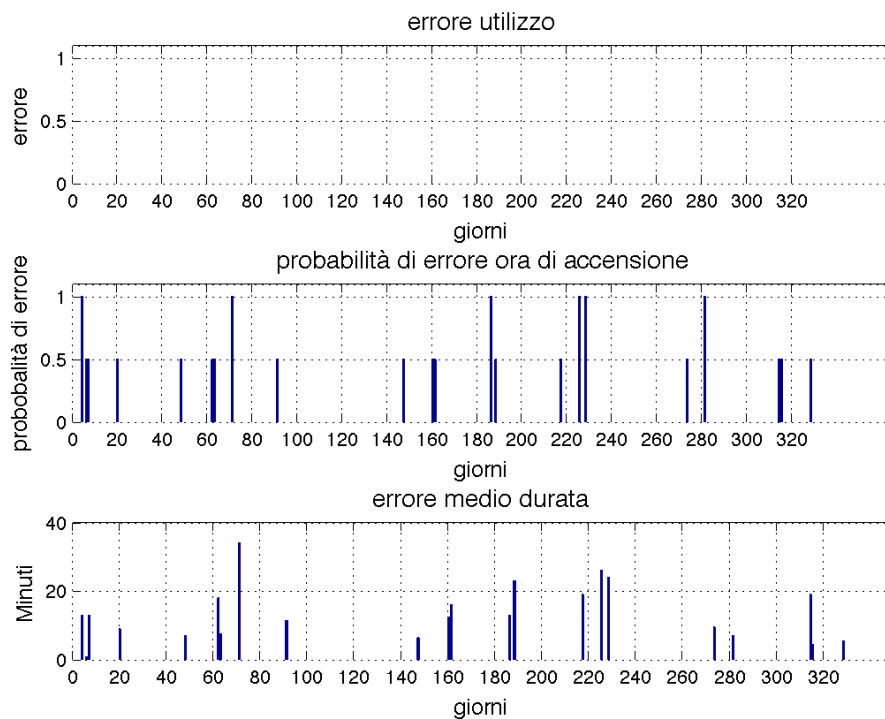
Il primo dispositivo considerato è la lavastoviglie per il quale si riportano più nel dettaglio i risultati ottenuti:

1. *caso regolare*: dai grafici (Figura 7.38) si evince che l'errore sulla predizione di utilizzo è pari allo 0% mentre nell'arco di un anno vi sono 15 giorni nei quali commette errori nella predizione dell'ora di accensione e nella durata.



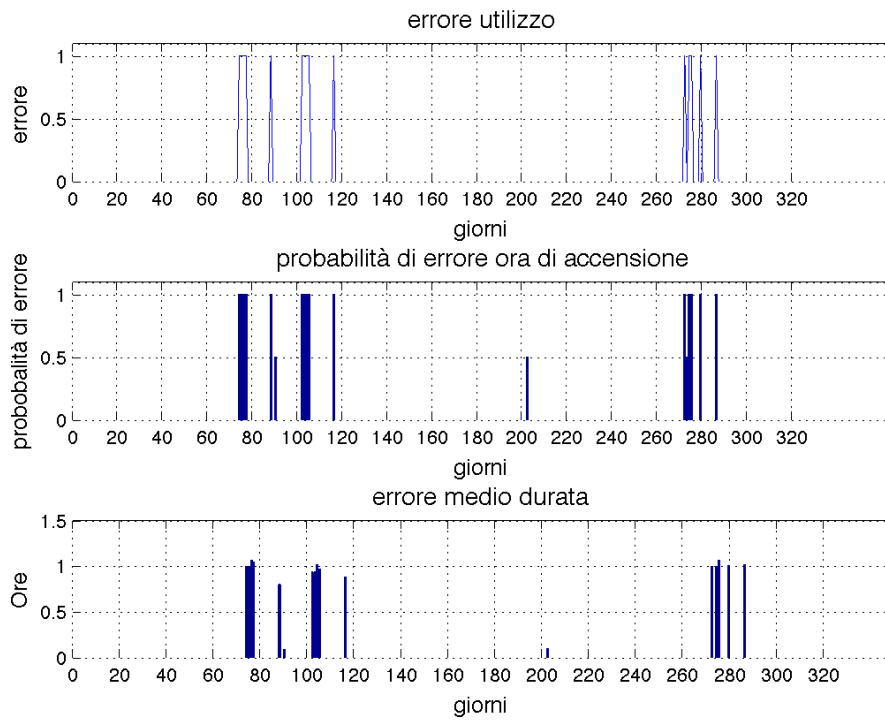
**Figura 7.38:** Test lavastoviglie regolare

2. *Cambiamenti settimanali ON OFF*: i dati sull'errore commesso sono molto simili al caso regolare così come rappresentato in Figura 7.39. L'errore sulla predizione di utilizzo risulta essere ancora dello 0%, mentre peggiora leggermente la predizione sull'ora e sulla durata.



**Figura 7.39:** Test lavastoviglie ON OFF

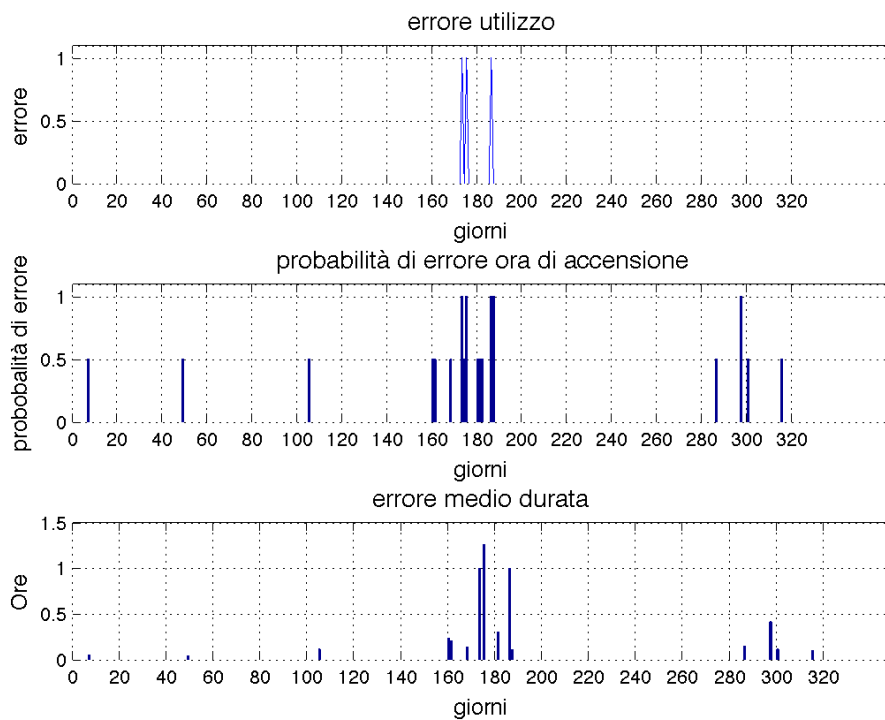
3. *Eccezioni casuali*: in questo caso durante tutto l'anno l'algoritmo sbaglia per 15 giorni la predizione sull'utilizzo e per 19 giorni la predizione sull'ora e sulla durata (Figura 7.40).



**Figura 7.40:** *Test lavastoviglie con eccezioni casuali*

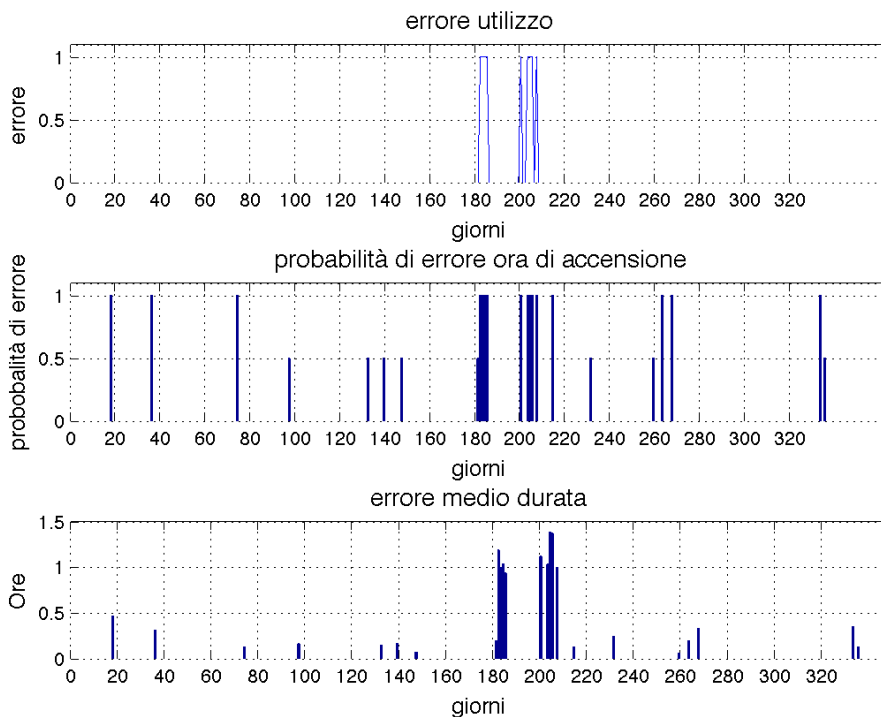


4. *Cambiamento abitudini a metà anno*: come prevedibile l'errore di utilizzo si concentra nei giorni seguenti al giorno del cambiamento. Si riscontrano 3 giorni nei quali viene sbagliata la predizione di utilizzo e 15 giorni nei quali sbaglia la predizione sull'ora di utilizzo e sulla durata (Figura 7.41).



**Figura 7.41:** Test lavastoviglie con cambio abitudini a metà anno

5. *Periodo di vacanza*: dal grafico (Figura 7.42) si nota che l'errore di utilizzo si concentra principalmente in due punti: all'inizio della vacanza e alla fine di essa. Questo perché l'algoritmo apprende che per 15 giorni l'elettrodomestico non è stato utilizzato e di conseguenza continua a sbagliare la predizione anche alla fine della vacanza. Gli errori di predizione di ora e di durata, inoltre, sono principalmente concentrati nello stesso periodo. In questo caso, però, risultano maggiori rispetto ai test precedenti sbagliando la predizione per 28 giorni.



**Figura 7.42:** Test lavastoviglie con periodo di vacanza

Per una trattazione più dettagliata si faccia riferimento alla Tabella 7.3.

Caso	Periodo di training	Errore utilizzo		Errore ora		Errore durata	
		%	Giorni	%	Giorni	%	Giorni
Regolare	28 gg	0.00	0	4.45	15	4.45	15
Eccezioni isolate (eccezioni=5)	28 gg	4.45	15	5.64	19	5.34	18
Cambiamenti settimanali ON OFF	28 gg	0.00	0	4.75	16	4.75	16
Cambio abitudini metà anno	28 gg	0.89	3	5.64	19	4.45	15
Periodo di vacanza	28 gg	2.67	9	8.31	28	8.31	28

**Tabella 7.3:** Risultati dei test sulla lavastoviglie

Il secondo elettrodomestico di cui presentiamo i risultati dei test sull'algoritmo è il forno, per il quale abbiamo ipotizzato l'utilizzo tutti i giorni alle 7:00, 12:00 e 19:00, spostando alle 9:00 l'ora di accensione nel caso della domenica. Analizzando i risultati numerici presenti nella Tabella 7.4 si riscontra, nel caso regolare, un errore d'utilizzo pari allo 0%. Gli errori sulla predizione dell'utilizzo, evidenziati in particolare nel caso di eccezioni isolate e nello scenario con il periodo di vacanza, sono dovuti all'introduzione di giorni casuali nei quali tale elettrodomestico non viene adoperato.

Nel caso di eccezioni isolate, il rapporto tra il numero di eccezioni introdotte e il numero di giorni nei quali l'algoritmo sbaglia la predizione è pari a 1. Nel caso di eccezioni continue, come nel periodo di vacanza, tale rapporto scende a 0.27. Tale risultato è dovuto al fatto che nel primo caso, essendo le eccezioni casuali, l'algoritmo non è in grado di prevederle e quindi sbaglia sempre, viceversa nel secondo caso l'algoritmo sbaglia solo all'inizio e al termine del periodo di vacanza.

Caso	Periodo di training	Errore utilizzo		Errore ora		Errore durata	
		%	Giorni	%	Giorni	%	Giorni
Regolare	28 gg	0.00	0	14.24	48	14.24	48
Eccezioni isolate (eccezioni=5)	28 gg	1.48	5	15.73	54	15.73	54
Cambiamenti settimanali ON OFF	28 gg	0.00	0	2.37	8	2.37	8
Cambio abitudini metà anno	28 gg	0.00	0	11.28	38	11.28	38
Periodo di vacanza	28 gg	1.19	4	14.54	19	14.54	19

**Tabella 7.4:** Risultati dei test sul forno

L'ultimo dispositivo su cui abbiamo effettuato i test è il condizionatore per il quale si è ipotizzato l'utilizzo durante il periodo estivo, nei mesi tra giugno e settembre tra le 10:00 e 17:00.

In tutti i quattro gli scenari analizzati, come evidenziato nella Tabella 7.5 sono stati riscontrati degli errori minimi nella predizione d'utilizzo, degli orari d'accensione e nella durata di funzionamento dovuti ai due cambiamenti d'abitudine a inizio e fine stagione.

Caso	Periodo di training	Errore utilizzo		Errore ora		Errore durata	
		%	Giorni	%	Giorni	%	Giorni
Regolare	28gg	1.19	4	1.19	4	1.19	4
Eccezioni isolate (eccezioni=6)	28 gg	4.15	14	4.15	14	4.15	14
Cambiamenti settimanali ON OFF	28 gg	1.48	5	1.48	5	1.48	5
Periodo di vacanza	45 gg	2.37	8	2.37	8	2.37	8

**Tabella 7.5:** Risultati dei test sul condizionatore

### 7.3 Confronto con la variante dell'algoritmo

Per testare la variante dell'algoritmo di predizione abbiamo deciso di utilizzare i dati riguardanti l'uso del condizionatore, ipotizzando in particolare che nel periodo estivo variasse il numero di accensioni giornaliere. Se tale variazione a prima vista poteva sembrare più precisa per il calcolo degli orari di accensione rispetto alla versione base, a seguito dei test si è invece potuto constatare che a fronte di un leggero miglioramento della predizione sull'orario e sulla durata si evidenzia un peggioramento delle prestazioni per quanto riguarda la predizione dell'utilizzo. Questo risultato è dovuto al fatto che in questo caso si ricerca la periodicità su stringhe che rappresentano il numero di accensioni al giorno e che quindi sono meno regolari rispetto a quelle usate nella versione base dell'algoritmo. Si è deciso dunque di scartare questa variante di algoritmo, in quanto l'obiettivo principale è quello di predire se l'elettrodomestico sarà usato o meno e solo in seguito alla correttezza di tale predizione, passare al calcolo degli orari di accensione e delle rispettive durate di funzionamento.

## 7.4 Considerazioni sui test fatti

Dall'analisi dei test effettuati si è ottenuta la configurazione finale dell'algoritmo di predizione creato per questa applicazione. Inoltre, come descritto nei paragrafi precedenti, ai vari parametri dell'algoritmo sono stati assegnati i seguenti valori:

- parametro di cluster= 0.25 (un quarto d'ora);
- varianza delle gaussiane=0.5 (mezz'ora);
- filtro sulle gaussiane=0.5;
- periodo di training= 28 giorni.

## 7.5 Risultati numerici dei test sperimentali

L'architettura del sistema è stata testata mediante una fase di sperimentazione, realizzata presso il laboratorio ANTLab del Politecnico di Milano la cui planimetria è mostrata in Figura 7.43 .

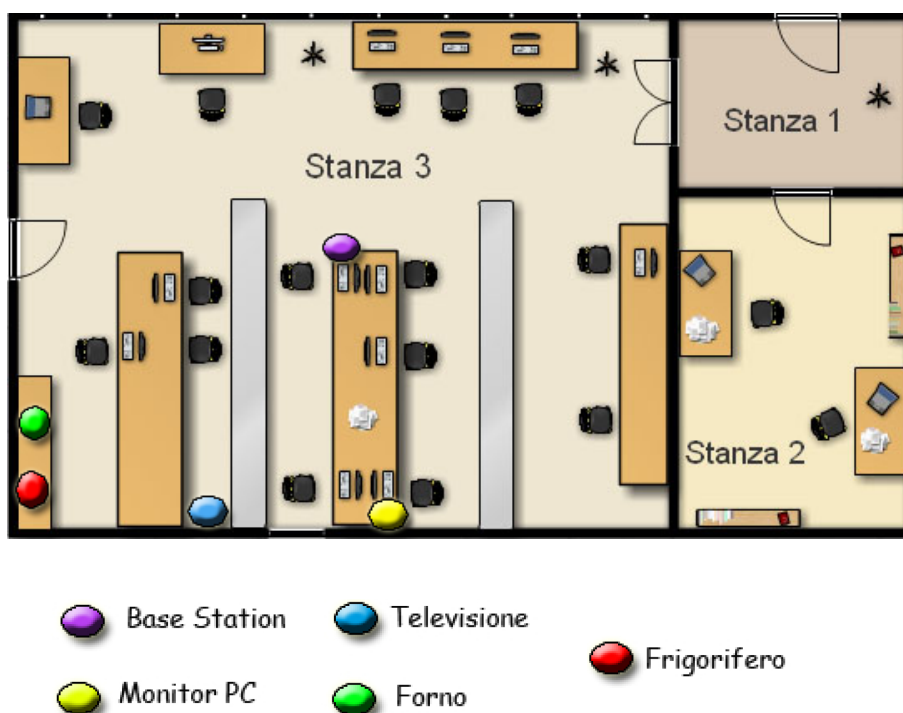
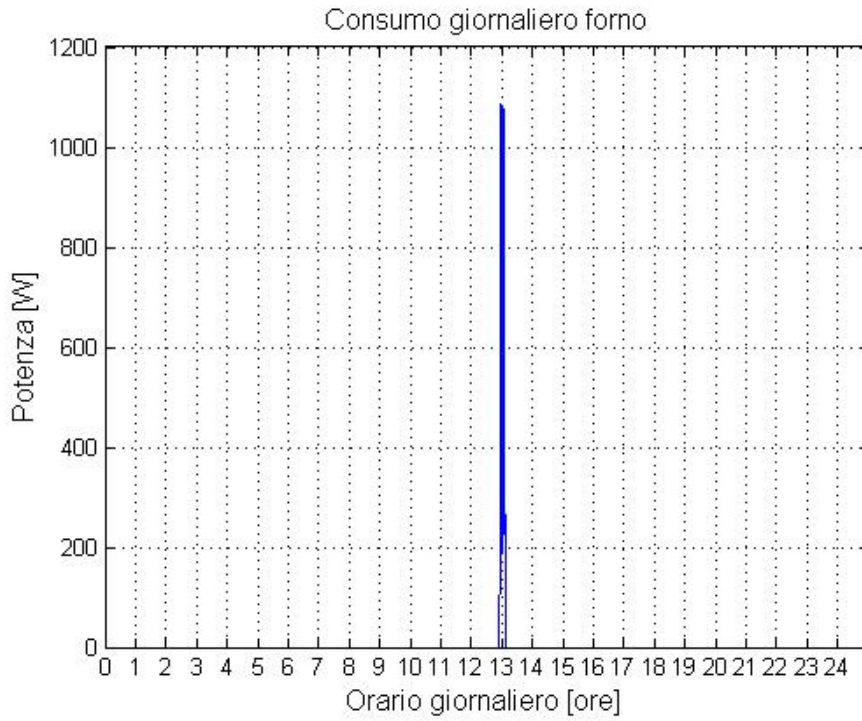
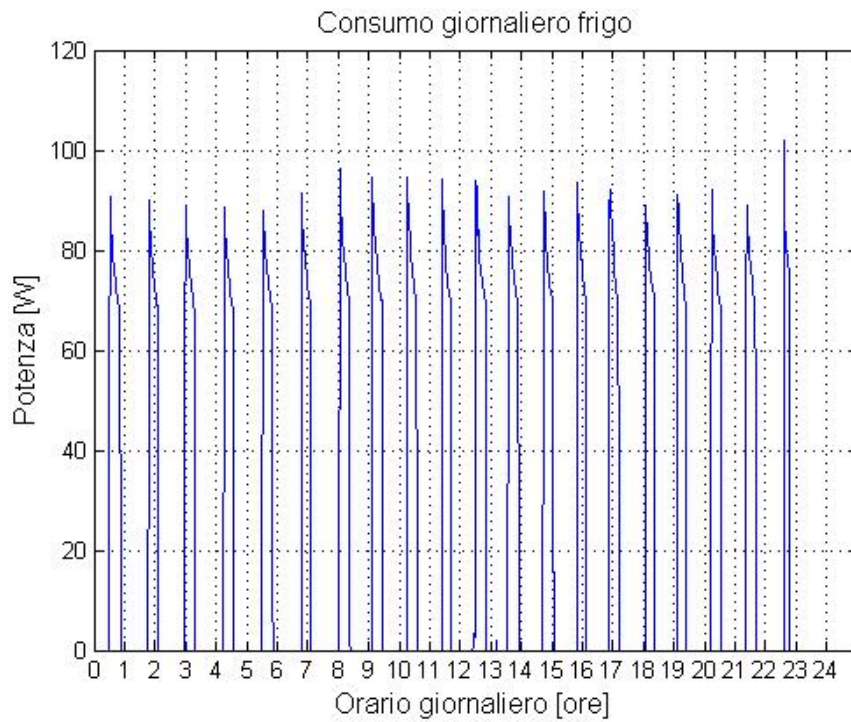


Figura 7.43: Mappa laboratorio

Abbiamo, in particolare realizzato la rete di sensori mediante quattro ACme ai quali sono stati rispettivamente collegati un frigorifero, un forno a microonde, una televisione e un monitor per pc. I quattro sensori inviano i dati alla *base station* anch'essa posizionata all'interno del laboratorio. In primo luogo abbiamo verificato il corretto funzionamento della rete di sensori, la trasmissione affidabile dei dati e la relativa memorizzazione nel database. In seguito abbiamo analizzato i dati raccolti e per ogni elettrodomestico abbiamo tracciato il profilo di consumo giornaliero. Nei grafici seguenti vengono presentati gli andamenti dell'utilizzo degli apparecchi considerati.



**Figura 7.44:** Consumo giornaliero del forno



**Figura 7.45:** Consumo giornaliero del frigorifero



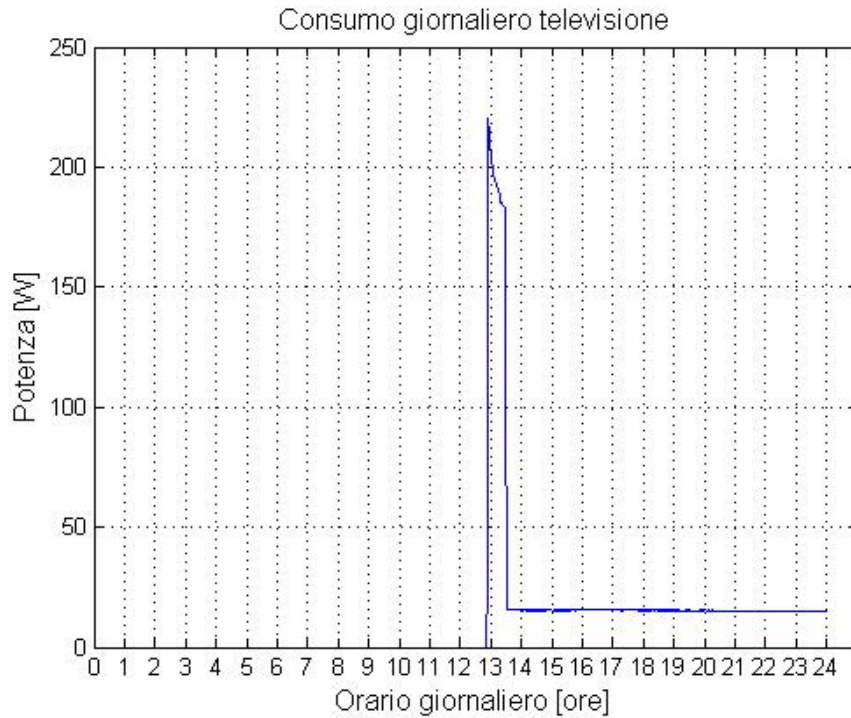


Figura 7.46: Consumo giornaliero della televisione

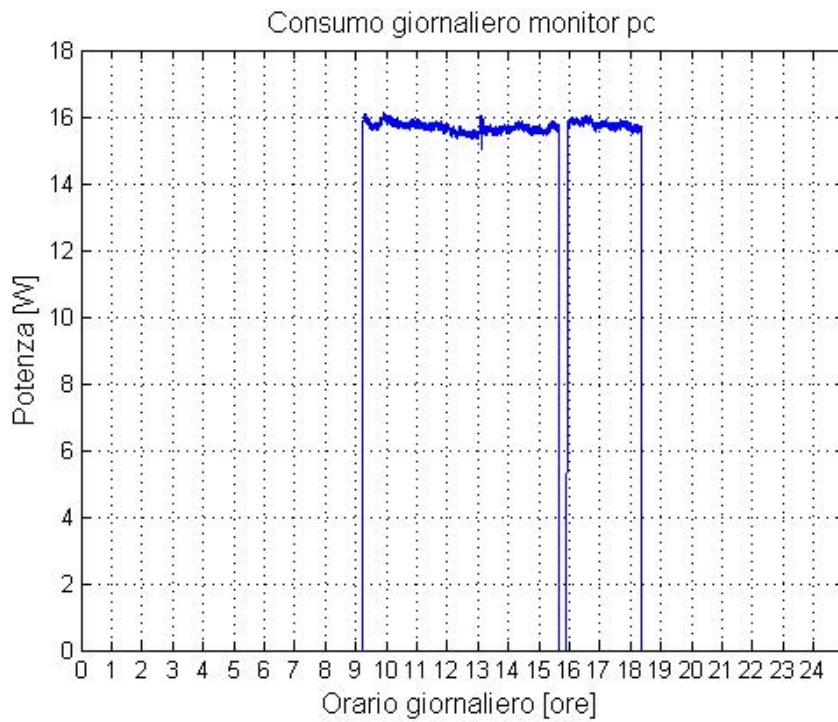


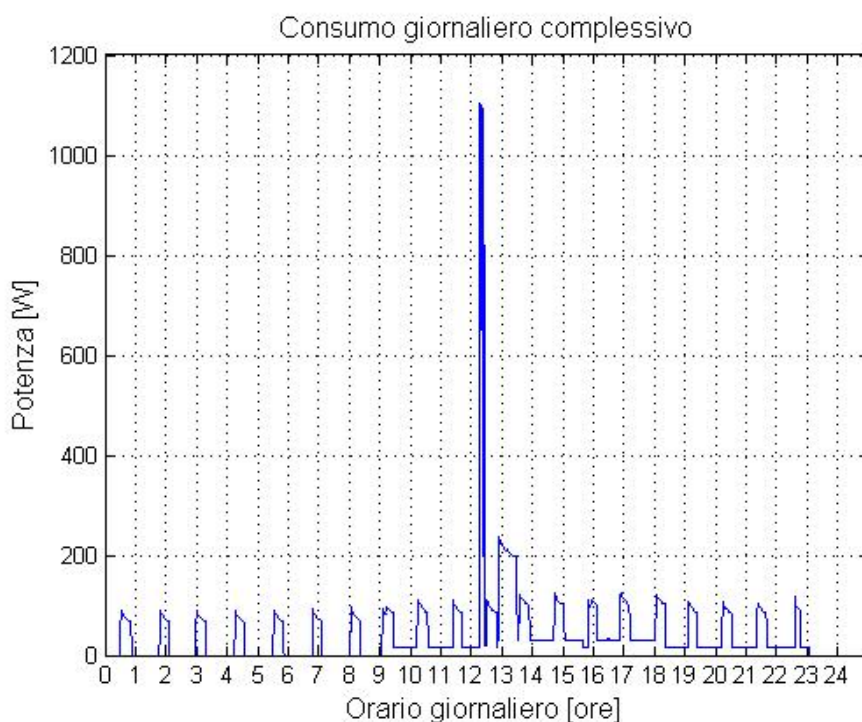
Figura 7.47: Consumo giornaliero del monitor

Come illustrato dalla Figura 7.44 il forno viene utilizzato unicamente a metà giornata, in particolare l'accensione avviene alle ore 13 e la relativa durata di funzionamento è di pochi minuti, con un picco di consumo pari a 1100 W.

La Figura 7.45 mostra il profilo di consumo di un frigorifero di classe A. Quest'ultimo rimane in funzione 24 ore al giorno, ma presenta un consumo energetico periodico solo in corrispondenza all'entrata in funzione del compressore pari a 90 W.

La televisione rimane spenta durante l'intera mattinata e viene utilizzata durante la pausa pranzo per una mezzora; in corrispondenza di tale utilizzo viene rilevato un consumo medio di potenza di circa 200 W. In seguito fino a mezzanotte lo stato della televisione risulta essere di stand-by. Dalla Figura 7.46 si può notare come lasciare la televisione in stand-by comporta un consumo medio di 17 W.

La Figura 7.47 illustra, infine, un profilo di consumo del monitor di un computer durante una giornata di lavoro. Viene acceso alle 9:00 del mattino quando l'utente inizia a lavorare e viene spento alle 18:30. Il monitor presenta un ridotto consumo, in media 16 W, in quanto viene utilizzato nella modalità di risparmio energetico. Come è possibile notare dal grafico se l'utente spegnesse il monitor ad ogni pausa introdurrebbe un risparmio che nel lungo periodo potrebbe rilevarsi significativo.



**Figura 7.48:** Consumo complessivo giornaliero

La Figura 7.48 mostra il consumo totale degli elettrodomestici considerati nell'arco di una giornata. Si può notare come il picco più elevato corrisponde all'utilizzo del forno a microonde durante la pausa pranzo. Si evidenzia, inoltre, un secondo picco dovuto all'utilizzo della televisione. La restante parte dei consumi risulta essere periodica dovuta al consumo del frigorifero che si rileva essere maggiore nella seconda parte della giornata causata soprattutto dalla televisione lasciata in stand-by.

Ad essi viene anche sommato il contributo dovuto all'utilizzo del monitor del computer.

### 7.5.1 Valutazione sperimentale dell'algoritmo di predizione

Durante la fase sperimentale abbiamo, inoltre, utilizzato i dati raccolti per valutare le prestazioni dell'algoritmo. Sono stati considerati tutti gli elettrodomestici descritti precedentemente ad eccezione del frigorifero poiché ovviamente risulta essere acceso 24 ore al giorno. A tal fine per la realizzazione dei test abbiamo deciso di utilizzare un periodo di training di soli 14 giorni esclusivamente per motivi di tempo (si ricorda che dai risultati illustrati in precedenza la lunghezza ideale del periodo di training è di 28 giorni). Per la valutazione dell'algoritmo sono stati considerati ulteriori 10 giorni per una fase complessiva di raccolta dati di 24 giorni. Di seguito vengono riportati i grafici (da Figura 7.49 a Figura 7.51) che mostrano l'andamento dell'errore di predizione per i tre elettrodomestici considerati.

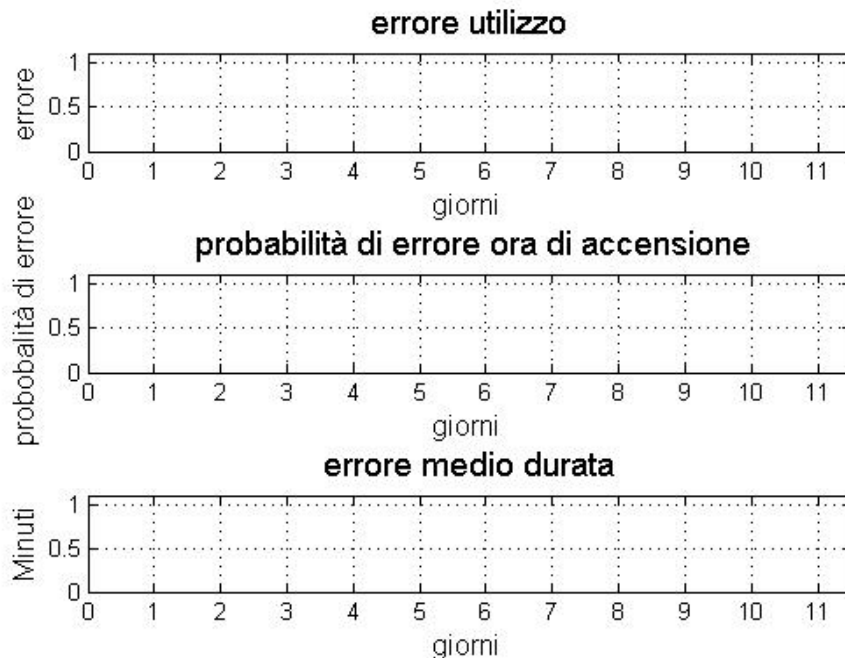


Figura 7.49: Risultati sperimentali del monitor

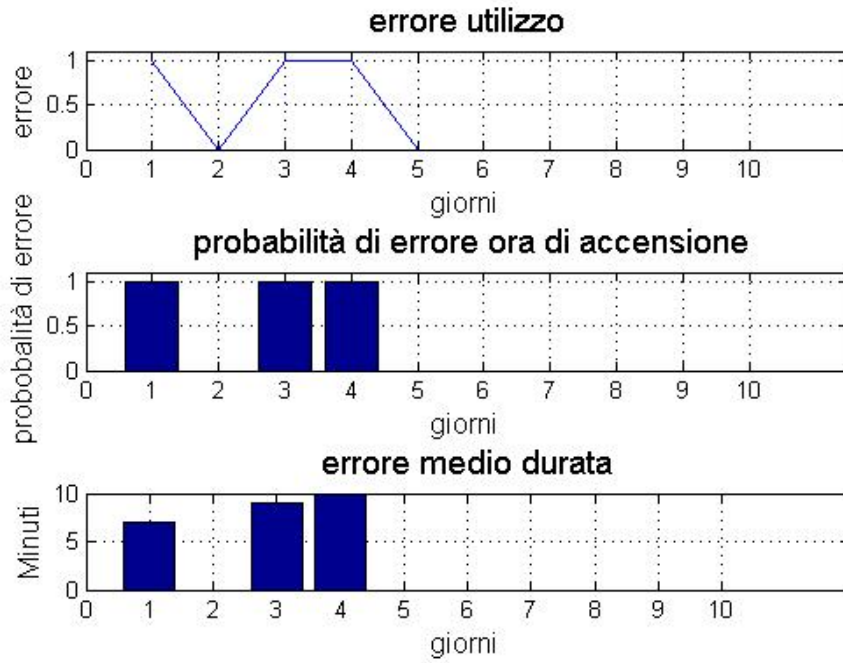


Figura 7.50: Risultati sperimentali del forno

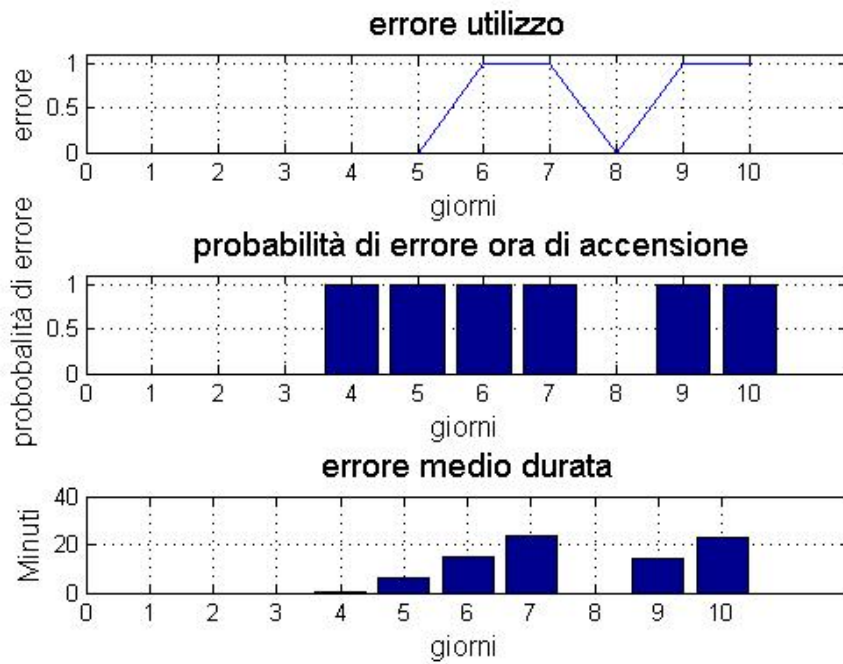


Figura 7.51: Risultati sperimentali della televisione

Per quanto riguarda il monitor si osserva un errore pari allo 0% (si ricorda che sono trascurati scostamenti di 15 minuti tra l'utilizzo predetto e quello sperimentato per l'errore relativo all'ora e alla durata) in tutti le predizioni essendo acceso e spento regolarmente ogni giorno all'incirca alla stessa ora. Gli altri due elettrodomestici presentano, invece, un errore di predizione alto, dovuto al fatto che sono presenti delle eccezioni durante la fase di validazione dell'algoritmo, che con un periodo di training così basso non riesce a cogliere. Come mostrato dalla Figura 7.50 l'algoritmo sbaglia la predizione sull'accensione del forno il primo venerdì, la prima domenica e il lunedì a causa dell'eccezione presentatasi il venerdì stesso. La Figura 7.51, infine evidenzia un errore sulla predizione dell'accensione della televisione il mercoledì, il giovedì, il sabato e la domenica poiché è presente un'eccezione il mercoledì. I risultati numerici di questi test sperimentali sono riassunti in Tabella 7.6.

	<b>Errore utilizzo</b>	<b>Errore ora</b>	<b>Errore durata</b>
Monitor	0%	0%	0%
Forno	30%	30%	30%
Televisione	40%	60%	60%

**Tabella 7.6:** *Risultati dei test sperimentali*

In conclusione, nel caso regolare l'algoritmo evidenzia ottime prestazioni, mentre in presenza di eccezioni si riscontra un errore medio sulla predizione di utilizzo del 35% che sicuramente si ridurrà utilizzando un periodo di training più lungo così come mostrato nella fase di simulazione.

# Capitolo 8

## Conclusioni

In questo capitolo vengono presentate le conclusioni sul lavoro di tesi svolto e considerazioni su possibili sviluppi futuri.

### 8.1 Considerazioni sul lavoro svolto

La crescente domanda di energia, la riduzione dei combustibili fossili e la necessità di ridurre le emissioni di biossido di carbonio, hanno portato in primo piano l'importanza del risparmio e dell'efficienza energetica. Buona parte di questi consumi è in particolare da addebitarsi al settore residenziale rendendo dunque importante la sensibilizzazione dei consumatori su tali temi e lo sviluppo di strumenti di supporto adeguati. Molte delle soluzioni già esistenti consentono la visualizzazione dei consumi e il controllo remoto dei dispositivi, ma non ne predicono l'utilizzo e soprattutto non supportano l'utente con consigli pratici per evitare sprechi energetici.

Per tale motivo, in questo lavoro di tesi abbiamo progettato e realizzato un sistema innovativo che oltre a controllare i dispositivi domestici, predice il comportamento dell'utente e fornisce consigli per un utilizzo efficiente degli elettrodomestici.

L'intero sistema si basa sul paradigma client-server. Il lato server si occupa della gestione della rete di sensori e dell'algoritmo di predizione, il lato client invece è costituito dalla applicazione Android *Energy Coach*.

Per la misurazione dei consumi abbiamo realizzato una rete wireless che risulta essere maggiormente conveniente rispetto alle reti cablate, in quanto è più semplice ed economica l'aggiunta di nuovo hardware a reti preesistenti. La rete realizzata, in particolare, è costituita da sensori ACme (AC Meter) che permettono la rilevazione dei consumi dell'elettrodomestico associato e la trasmissione wireless dei dati acquisiti.

L'algoritmo di predizione che abbiamo definito in questo lavoro di tesi ricopre un ruolo fondamentale all'interno del sistema. Esso ha il compito di elaborare i dati raccolti attraverso la rete di sensori, relativi a ciascun apparato domestico, e predire per ogni giorno, il comportamento di quest'ultimi indicandone in particolare l'utilizzo, gli orari di accensione e la durata media del periodo di funzionamento. L'algoritmo che abbiamo descritto nella tesi risulta innovativo rispetto alle altre metodologie presenti in letteratura per diversi motivi. Nel corso degli ultimi anni, infatti, sono state definite tecniche di predizione dei consumi totali delle abitazioni senza però entrare mai nel dettaglio dei singoli dispositivi. Lo stesso metodo di predizione, inoltre, è del tutto originale in quanto non prende spunto da nessun meccanismo di learning fino ad ora definito. Per valutare l'efficienza e le prestazioni di questo algoritmo sono stati condotti alcuni test, sia mediante una fase di simulazione che una sperimentale descritte nel Capitolo 7. Per quanto riguarda le simulazioni abbiamo preso in considerazione l'utilizzo di tre elettrodomestici ovvero, lavastoviglie, forno e condizionatore, definendo per ciascuno di essi scenari d'utilizzo sufficientemente realistici. Il parametro di predizione sul quale ci si è concentrati maggiormente è stato l'utilizzo dei dispositivi, per il quale, nei test condotti, si è rilevata un'efficienza superiore al 90%. In tali test inoltre si è potuto constatare come gli errori di predizione siano principalmente dovuti a comportamenti anomali dell'utente e pertanto risultino essere non predicibili. Per la valutazione dell'algoritmo abbiamo inoltre effettuato dei test sperimentali monitorando i consumi di quattro dispositivi presenti presso l'ANTLab: televiso-



re, monitor di un pc, forno a microonde e frigorifero. L'algoritmo di predizione ha evidenziato ottime prestazioni nel caso di utilizzo regolare degli elettrodomestici, mentre si sono riscontrati degli errori in presenza di eccezioni isolate che, a causa di un periodo di training particolarmente ridotto, l'algoritmo non è stato in grado di rilevare e gestire in maniera appropriata.

L'ultima area di ricerca sulla quale ci siamo concentrati ha riguardato la realizzazione di un'applicazione Android per smartphone chiamata *Energy Coach*. Essa permette la configurazione dell'abitazione, il controllo dei dispositivi, la visualizzazione dei consumi e dei costi tramite tabelle e grafici, mostra all'utente i risultati dell'algoritmo di predizione e fornisce consigli per risparmiare energia. Tramite essi l'utente può individuare gli sprechi dovuti a dispositivi in stand-by, può ricevere informazioni su come utilizzare gli elettrodomestici in modo corretto, sfruttando in maniera ottima le fasce orarie di minor costo e infine può ottenere segnalazioni su eventuali luci accese nelle stanze nelle quali non viene rilevata la presenza di persone.

L'*Energy Coach* inoltre è in grado di informare gli utenti relativamente alla disponibilità dell'energia prodotta dai pannelli solari termici e fotovoltaici in modo da ridurre i consumi energetici, i costi e le emissioni di  $CO_2$ . Il sistema, infine, interviene a segnalare, mediante messaggi, sovraccarichi di consumo di energia impedendo l'accensione di apparecchi che implicherebbe un prelievo di potenza superiore a quella consentita.

## 8.2 Sviluppi futuri

Il sistema sviluppato, può essere ulteriormente ampliato attraverso l'integrazione di sensori di presenza, temperatura e luminosità per rendere ancora più accurati i risultati dell'algoritmo di predizione. L'accensione del condizionatore, ad esempio, potrebbe essere vincolata non solo alla predizione di utilizzo, ma anche

a dati riguardanti la temperatura esterna e interna rilevata e alla presenza effettiva di utenti nell'abitazione. Altro miglioramento potrebbe inoltre essere ottenuto analizzando eventuali correlazioni dell'utilizzo di vari dispositivi. Si potrebbe, ad esempio, correlare l'accensione dell'asciugatrice con l'utilizzo della lavatrice.

Un ulteriore sviluppo futuro potrebbe consistere nell'implementazione di un algoritmo di aggiornamento real time della predizione sulla base dei dispositivi utilizzati e dei dati provenienti dagli altri sensori. La predizione verrebbe quindi adattata dinamicamente in seguito ad eventi che si potrebbero verificare. In questo modo, se l'algoritmo prevedesse normalmente un utilizzo degli elettrodomestici in base ad uno scenario lavorativo, in caso di cambiamento casuale di abitudine l'algoritmo si adatterebbe dinamicamente ad esso.

# Bibliografia

- [1] Home Automation. Website <http://home-automation.org/>.
- [2] Jiakang Lu, Dagnachew Birru, and Kamin Whitehouse. Using simple light to achieve smart daylight harvesting. *BuildSys 2010 -2nd ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings*, 2010 Zurich.
- [3] Bticino. Website [http://www.myhome-bticino.it/web/home\\_page.page](http://www.myhome-bticino.it/web/home_page.page).
- [4] Gewiss. Website [http://www.gewiss.com/irj/portal/login\\_IT](http://www.gewiss.com/irj/portal/login_IT).
- [5] Vimar. Website <http://www.vimar.eu/>.
- [6] D. Cook, M. Huber, K. Gopalratnam, and M. Youngblood. Learning to control a smart home environment. *In Innovative Appl. of Artificial Intelligence*, 2003.
- [7] Jeff Hsu, Prashanth Mohan, Xiaofan Jiang, Jorge Ortiz, Sushant Shankar, Stephen Dawson-Haggerty, and David Culler. Hbci: Human-building-computer interaction. *BuildSys 2010 -2nd ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings*, 2010 Zurich.
- [8] Sis statistica. Website <http://www.sis-statistica.it/magazine/spip.php?article145>.
- [9] Liikkanen. Extreme-user approach and the design of energy feedback systems. *To appear in Proc. of Energy Efficiency in Domestic Appliances and Lighting EEDAL*, 2009 Berlin, Germany.

- [10] Abrahamse, Steg, Vlek, and Rothengatter. A review of intervention studies aimed at household energy conservation. *Journal of Environmental Psychology*, volume 25, issue 3, pages 273- 291, 2005.
- [11] Kluger e Dinisi. The effects of feedback interventions on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theor. *Journal of Applied Psychology*, volume 119, issue 2, pages 254-284, 1996.
- [12] Google Powermeter. Website <http://www.google.com/powermeter>.
- [13] Control4. Website <http://www.control4.com/>.
- [14] D. J. Cook, M. Youngblood, III E. O. Heierman, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. Mavhome: An agent-based smart home. *In First IEEE International Conference on Pervasive Computing and Communications (PerCom 03) pages 521- 524*, 2007.
- [15] Antimo Barbato, Luca Borsani, and Antonio Capone. A wireless sensor network based system for reducing home energy consumption. *Sensor MESH and Ad Hoc Communications and Networks 7th Annual IEEE Communications Society Conference On*, 2010.
- [16] Antimo Barbato, Luca borsani, Antonio Capone, and Stefano Melzi. Home energy saving through a user profiling system based on wireless sensors. *BuildSys '09 Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, 2009, Berkeley.
- [17] A. Capone, M. Barros, H. Hrasnica, and S. Tomprosi. A new architecture for reduction of energy consumption of home appliances. *European conference of the Czech Presidency of the Council of the EU TOWARDS eENVIRONMENT*, 2009 Prague.

- [18] Guy R. Newsham and Benjamin J. Birt. Building-level occupancy data to improve arima-based electricity use forecasts. *BuildSys 2010 -2nd ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings*, 2010, Zurich.
- [19] Geoffrey K.F. Tso and Kelvin K.W. Yau. Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. *Energy, Volume 32, Issue 9, pages 1761-1768*, 2007.
- [20] Debasish Basak, Srimanta Pal, and Dipak Chandra Patranabisi. Support vector regression. *Neural Information Processing- Letters and Reviews, volume 10, issue 10, pages 203-224*, 2007.
- [21] G.W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE, Volume 80, Issue 12, pages: 1870 - 1891*, 2002.
- [22] Anthony Rowe, Mario Berges, and Raj Rajkumar. Contactless sensing of appliance state transitions through variations in electromagnetic fields. *BuildSys 2010 -2nd ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings*, 2010 Zurich.
- [23] Berkeley. Website <http://acme.cs.berkeley.edu>.
- [24] Xiaofan Jiang, Stephen Dawson-Haggerty, Jay Taneja, Prabal Dutta, and David Culler. Creating greener homes with ip-based wireless ac energy monitors. *SenSys '08 Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008, Raleigh.
- [25] Xiaofan Jiang, Stephen Dawson-Haggerty, Jay Taneja, Prabal Dutta, and David Culler. Design and implementation of a high-fidelity ac metering network. *Proceeding IPSN '09 Proceedings of the 2009 International Conference of Information Processing in Sensor Networks*, 2009 San Francisco.

- [26] Hall effect sensing and application. Website <http://content.honeywell.com/sensing/prodinfo/solidstate/technical/hallbook.pdf>.
- [27] moteware. Website <http://www.moteware.com/acmecal.php>.
- [28] Tiny OS. Website <http://www.tinyos.net/>.
- [29] RMI. Website <http://www.javacoffeebreak.com/articles/javarmi/javarmi.html>.
- [30] XMLRPC. Website <http://www.xmlrpc.com/>.
- [31] MySQL. Website <http://dev.mysql.com/>.
- [32] JDBC. Website <http://dev.mysql.com/usingmysql/java/>.
- [33] Android. Website <http://www.android.com/>.
- [34] Eclipse. Website <http://www.eclipse.org/>.
- [35] Samsung. Website <http://galaxytab.samsungmobile.com/>.