



UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea Triennale in Ingegneria Meccanica

**MISURA DI ACTIVITIES OF DAILY LIVING IN AMBIENTE DOMESTICO ATTRAVERSO
L'UTILIZZO DI UN TOOL DI SIMULAZIONE E CONFRONTO CON DATI REALI**

**MEASUREMENT OF ACTIVITIES OF DAILY LIVING IN HOME ENVIRONMENT
THROUGH A SIMULATION TOOL AND CORRELATION WITH REAL USERS' SIGNALS**

Relatore: Chiar.mo

Prof. Ing. Gian Marco Revel

Tesi di Laurea di:

Simone Salucci

Correlatori:

Ph.D. Ing. Sara Casaccia

Ing. Matteo Sorcinelli

A.A. 2019/2020

Sommario

Abstract	3
CAPITOLO 1:INTRODUZIONE	4
CAPITOLO 2:TOOL DI SIMULAZIONE	10
2.1 LA SIMULAZIONE DEI DATI	10
2.2 TOOL DI SIMULAZIONE UNIVPM HOME-CARE	11
2.2.1 Area statistiche	16
2.2.2 Machine learning (ML)	18
2.2.3 Misura della performance.....	21
CAPITOLO 3: ANALISI RESIDENZA 232	23
CAPITOLO 4: ALGORITMI E CONFRONTO CON DATI REALI	26
4.1 INTRODUZIONE AGLI ALGORITMI	27
4.2 TOILETING E EATING	28
4.3 ENTERING LEAVING	30
4.4 OUTSIDE.....	31
4.5 INSIDE	32
4.6 SLEEPING ASLEEP/SLEEPING AWAKE	34
4.7 CONFRONTO CON DATI REALI	36
CAPITOLO 5: OTTIMIZZAZIONE RETE DI SENSORI	40
5.1 1°CASO	43
5.2 2°CASO	44
5.3 3°CASO	45
5.4 4°CASO	47
5.5 STIMA DEI COSTI	49
CAPITOLO 6.....	51
6.1 CONCLUSIONI	51
6.2 FUTURE WORKS	52
BIBLIOGRAFIA	53

Abstract

L'aumento dell'età media della popolazione fa crescere il numero di soggetti a rischio di patologie e si stanno studiando nuove soluzioni per controllare lo stato di salute di queste persone, soprattutto anziane, consentendo loro di continuare a vivere in casa.

Peggioramenti dello stato di salute possono essere rilevati monitorando le Activities of Daily Living (ADLs), ossia le normali attività svolte tutti i giorni come lavarsi o mangiare, e andando a cercare delle variazioni nel comportamento dell'individuo in un certo intervallo di tempo attraverso le rilevazioni di alcuni sensori.

In questa tesi viene utilizzato un tool di simulazione sviluppato all'interno del progetto internazionale eWare che permette di creare una rete di sensori PIR virtuali e di simulare un certo periodo di rilevazioni.

È presentata un'analisi dei dati ottenuti in output dal tool effettuata tramite specifici algoritmi sviluppati in Matlab con cui è possibile rilevare attività complesse, cioè che coinvolgono più di un'attivazione dei sensori.

Sono stati simulati i dati relativi a un'abitazione reale implementata con una rete di sensori PIR e, attraverso l'uso degli algoritmi, è stato fatto un confronto tra dati reali e simulati.

Infine, considerando che non è possibile studiare in un contesto reale quale sia la migliore configurazione di sensori per un certo ambiente e una certa persona, viene effettuata un'ottimizzazione della rete di sensori nei limiti imposti dal programma di simulazione. Attraverso degli algoritmi di classificazione viene determinato il numero e la disposizione dei sensori che massimizzano i loro output considerando anche gli eventuali costi.

CAPITOLO 1

INTRODUZIONE

In questo capitolo vengono descritte in modo generale le motivazioni e i problemi che hanno portato a sviluppare il progetto eWare e il contesto generale in cui si colloca questa tesi.

Sono descritte inoltre diverse tipologie di sensori che possono essere installati in un ambiente smart, i loro vantaggi e svantaggi soffermandosi in particolare sui sensori PIR.

Il numero di persone affette da malattie legate all'età è in continuo aumento a causa dell'invecchiamento della popolazione [1].

Queste malattie, come demenza senile e Alzheimer, portano ad un progressivo declino delle capacità fisiche e cognitive e svolgere le attività più semplici della vita quotidiana risulta essere sempre più complicato.

Nell'ambito del progetto internazionale eWare [2], finanziato dalla Commissione Europea e dai Ministeri della Nazioni partecipanti, è stato proposto un sistema di tecnologia eHealth che favorisce il benessere degli anziani con demenza senile consentendo loro di trascorrere la loro vita in casa.

Inoltre un numero di malati sempre crescente potrebbe rendere impossibile un'adeguata assistenza ospedaliera per tutti.

Per soddisfare le esigenze dei pazienti e alleggerire il carico sanitario è stato progettato un sistema realizzato tramite l'integrazione di una rete di sensori che monitora le abitudini e lo stile di vita del paziente, di un robot che gli fornisce supporto sociale e di promemoria che si attivano in base al contesto.

Viene costituita quindi una Smart Home e tramite un'analisi delle informazioni rilevate si possono ricostruire le abitudini del paziente e, se sono cambiate rispetto alla sua tipica routine, possiamo probabilmente notare anche un peggioramento nello stato di salute [3].

Le informazioni ottenute dai sensori vengono trasmesse tramite un'app a colui che si occupa del paziente, che viene detto "Caregiver".

Considerando la complessità del comportamento umano, estrarre dai dati dei sensori informazioni rilevanti è molto complesso e la ricerca in merito è ancora aperta: non si può conoscere quindi a priori quale sia il miglior tipo di sensore, il loro numero e la migliore disposizione all'interno di un ambiente.

Nel progetto eWare i sensori utilizzati sono sensori di movimento ad infrarosso (Passive InfraRed, PIR) posizionati in varie zone della casa e sono presenti anche dei contatti magnetici applicati alle porte.

I sensori sono il punto di partenza per la misura, e l'abilità del sistema di identificare le attività dipende fortemente dalle loro caratteristiche.

I sensori che possono essere utilizzati appartengono a tre categorie principali [4]:

1) Sensori indossabili

Sono di piccole dimensioni e possono essere facilmente indossati in diverse posizioni, comprendono i sensori inerziali quali accelerometri, giroscopi e i biosensori che rilevano i parametri vitali. Molte di queste tecnologie sono oggi contenute in uno smartphone, il che li rende utili strumenti di misura considerando che sono frequentemente utilizzati e rendono la misura poco invasiva [5].

2) Sensori ambientali

Appartengono a questa tipologia i sensori di movimento, i contatti magnetici e sensori di pressione. Vanno installati su porte o pareti e la misura avviene tramite l'interazione di essi con l'utente.

3) Dispositivi multimediali

Microfoni, fotocamere, videocamere sono in grado di fornire informazioni molto dettagliate ma il loro utilizzo è limitato a causa di importanti problemi legati alla privacy.

Concentriamoci sui sensori di movimento ad infrarosso, quelli considerati in questa tesi e, come vedremo, gli unici disponibili sul tool di simulazione utilizzato.

Un sensore passivo ad infrarosso (PIR) è un sensore elettronico che misura i raggi infrarossi irradiati dagli oggetti nel suo campo di vista.

Tutti gli oggetti con temperatura superiore allo zero assoluto, compreso il corpo umano, emettono energia sotto forma di onde elettromagnetiche e questi dispositivi sono progettati per rilevarne alcune invisibili ad occhio nudo.

Il termine passivo indica che il sensore non emette energia ma rileva semplicemente quella sprigionata dagli oggetti che lo circondano.



Figura 1: *un sensore PIR*

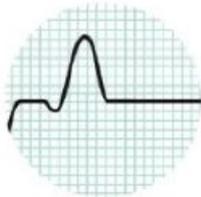


Figura 2: *segnale elettrico in uscita dal sensore*

Un sensore è mostrato in figura 1.

Il PIR è costituito da: sensori piroelettrici, che trasducono il calore in tensione, e una lente convessa che focalizza il segnale infrarosso su di essi facendoli quindi scaldare.

Per rilevare il movimento il sensore non deve misurare il livello di radiazione ma una sua variazione: per questo un PIR contiene due sensori piezoelettrici, uno responsabile di produrre un output positivo e l'altro un output negativo. Questi due sono collegati in modo da fornire un output nullo se non ci sono movimenti attorno al dispositivo.

Quando un corpo caldo attraversa il suo campo visivo viene intercettato prima dalla prima metà e in seguito dalla seconda.

Risulta quindi un picco nel segnale in output, come in figura 2.

Vediamo ora quali sono i vantaggi dei sensori PIR rispetto ad altre tipologie: garantiscono un totale rispetto della privacy e non sono invasivi, inoltre non è necessaria nessuna collaborazione da parte dell'utente; il loro funzionamento è garantito anche di notte o con luce scarsa, condizioni in cui altre tecnologie non possono lavorare ed infine sono economici e semplici da usare.

Non riescono da soli a fornire informazioni dettagliate sulle attività svolte dall'utente poiché sono sensori di tipo binario e perciò indicano semplicemente se c'è del movimento all'interno del loro campo d'azione.

Si possono ricavare informazioni sulle attività svolte organizzando i sensori in una rete e costruendo su essi una logica di controllo.

Per esempio se si vuole monitorare l'attività 'dormire' avendo a disposizione una rete di sensori, si può controllare l'attivazione di un sensore in camera da letto: se non si registra nessuna altra attivazione per un certo tempo dopo che il sensore si è spento (durante il sonno non c'è movimento) si può concludere che è iniziata questa attività.

Può essere anche fatto un controllo sull'orario di questa attivazione per verificare se il paziente dorme nei momenti corretti.

Questo tipo di problema può essere affrontato attraverso degli algoritmi di classificazione ossia tramite il machine learning.

Vediamo alcune situazioni reali: nell'articolo [6] viene presentato uno studio empirico sul riconoscimento del movimento umano attraverso una rete di sensori PIR. I sensori sono stati installati in un corridoio, uno posizionato nel soffitto al centro e due sulle pareti laterali, uno di fronte all'altro.

Sono stati raccolti dati relativi a otto diversi soggetti mentre camminano in differenti condizioni: due direzioni (avanti e indietro), tre diverse distanze (vicino al primo sensore a parete, al centro del corridoio, vicino al secondo sensore a parete) e a tre differenti velocità.

Attraverso l'analisi dei dati ottenuti è stato possibile ottenere una accuratezza maggiore al 92% nella classificazione della direzione e della velocità del movimento, nella distanza dai sensori e nell'identificazione del soggetto;

Nell'articolo [7] viene studiato un innovativo metodo di localizzazione per tracciare la posizione di una persona in un ambiente chiuso tramite sensori PIR e un algoritmo di machine learning. In questo studio è stata costruita una mappa che riflette le posizioni visitate e occupate dall'utente più frequentemente tramite dei dati noti a priori, a questo punto i sensori PIR sono stati posizionati nei punti più adeguati.

I sensori possono rilevare nel miglior modo le informazioni relative alla posizione dell'utente avendo ottimizzato la loro disposizione.

Nell'articolo [8] è stato introdotto un metodo per rilevare un cambiamento nel comportamento di un paziente all'interno di una Smart Home.

Sono stati analizzati tre casi reali ed è stato possibile notare qual è l'impatto di un evento sanitario sul comportamento di un individuo.

Si è notato che gli effetti di un evento importante, come la diagnosi di una malattia, possono ripercuotersi su gran parte delle attività della vita quotidiana come dormire, mangiare e uscire di casa mentre eventi meno gravi hanno un impatto più localizzato. Utilizzando un algoritmo è possibile avvisare il Caregiver quando si riscontrano variazioni nelle abitudini del paziente.

CAPITOLO 2

TOOL DI SIMULAZIONE

In questo capitolo sono spiegate le principali problematiche che portano alla necessità di simulare i dati: viene presentato il tool di simulazione che ci permette di digitalizzare una mappa e simulare virtualmente una rete di sensori PIR installata in ambiente domestico; vengono fornite le definizioni principali di machine learning, le misure della performance e le modalità con cui queste sono implementate all'interno del tool. Vediamo inoltre quali sono i vantaggi e i limiti di esso

2.1 LA SIMULAZIONE DEI DATI

Misurare quantità fisiche legate allo stile di vita di un anziano risulta essere uno dei metodi più efficaci per fornire assistenza sanitaria ad esso, migliorando notevolmente la qualità di vita e riducendo i costi sanitari.

Allo stesso tempo, per fornire soluzioni di alto livello, sono necessari algoritmi che siano in grado di identificare le attività svolte ed essi necessitano di notevoli quantità di dati per essere sviluppati e per funzionare al meglio [9].

Purtroppo sono ancora pochi e limitati i dataset necessari, a causa degli alti costi dei sensori, della loro disponibilità e dei lunghi tempi di acquisizione richiesti per la loro creazione. L'utilizzo di un ambiente simulato può risolvere alcuni di questi problemi e facilitare la generazione di tali dataset.

Viene ora presentato il programma di simulazione che è stato utilizzato in questa tesi.

2.2 TOOL DI SIMULAZIONE UNIVPM HOME-CARE[4]

Il simulatore UNIVPM Home-Care, sviluppato in una tesi di dottorato svolta nel Dipartimento di Ingegneria Industriale e Scienze Matematiche, è una applicazione web che permette all'utente di simulare, analizzare e processare i dati che derivano da una rete di PIR virtuali e di generare dei dataset relativi alle ADLs.

Il tool è composto da:

- Un'area di simulazione che consente di simulare un ambiente fisico, dei sensori e la traiettoria relativa al comportamento di un utente virtuale
- Un'area di analisi in cui sono visualizzate varie statistiche e la performance degli algoritmi di machine learning, necessari per estrarre informazioni qualitative dai dataset generati
- Un'area di controllo in cui vengono fornite informazioni in tempo reale che derivano dalla rete di sensori virtuale.

All'apertura dell'applicazione, una schermata iniziale permette di definire tutte le caratteristiche dell'ambiente virtuale come il numero di stanze e le caratteristiche dei sensori PIR ed è richiesto di caricare la piantina della casa che si vuole analizzare. L'utente può definire l'area per ogni stanza che desidera, delimitando i suoi confini e associandole un nome.

Il tool di simulazione digitalizza l'immagine della piantina associando ad ogni punto di essa delle x e y; in questo modo, dopo aver definito le aree di forma rettangolare corrispondenti alle stanze, può essere individuato il loro perimetro, come in figura 3.

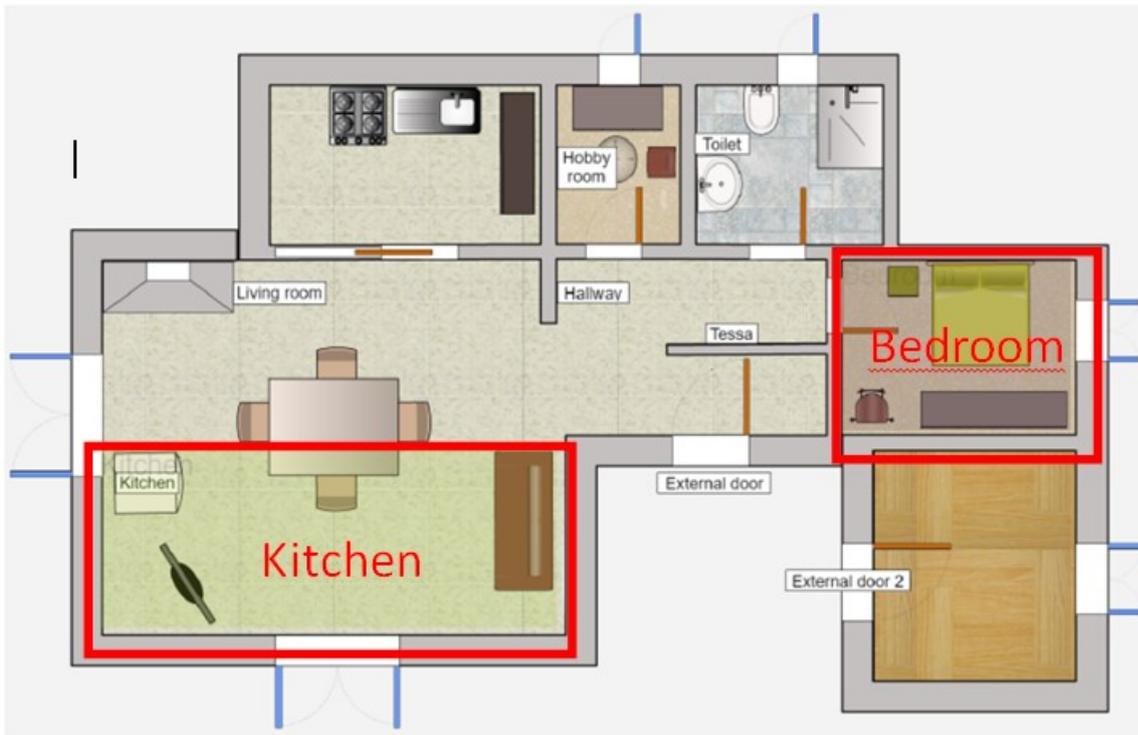


Figura 3: esempio di definizione delle stanze

Quando la definizione delle stanze è completa, l'utente può procedere con il posizionamento dei sensori nell'ambiente.

In questo tool sono stati implementati solamente sensori PIR, cioè una tipologia di sensore che varia il proprio output quando l'utente virtuale si muove nel suo campo visivo.

I sensori possono essere posizionati in ogni punto della mappa.

Per aggiungere un sensore PIR basta associarlo ad una stanza e posizionarlo nella sua area.

Come per le stanze, per facilitare il posizionamento, viene visualizzata l'area che esso va a ricoprire. Il raggio di azione del PIR risulta però limitato dai confini della stanza a cui viene associato, come mostrato in figura 4.

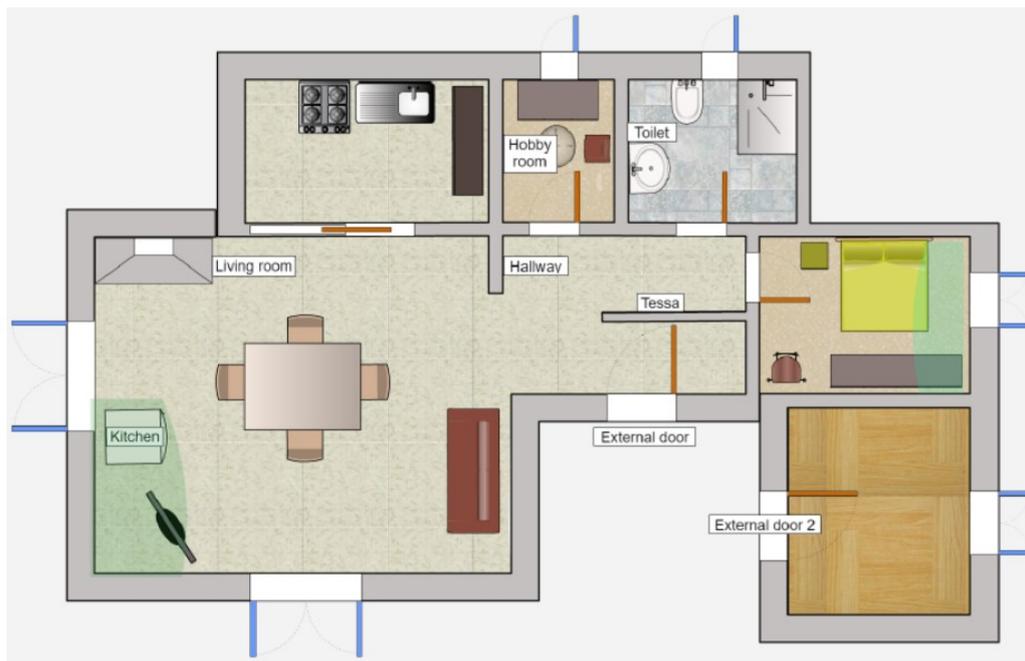


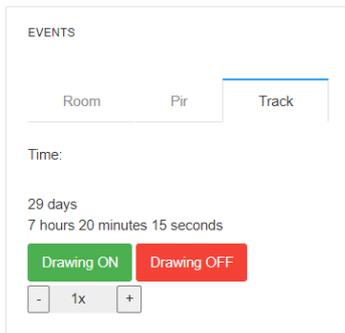
Figura 4: *posizione di un sensore PIR*

L'area relativa al suo raggio d'azione viene limitata dai bordi della stanza scelta

I sensori virtuali sono stati programmati per fornire una rilevazione quando si passa su di essi la traccia del puntatore e restano accesi (forniscono come output 'true') per tutto il tempo in cui il puntatore resta nel loro campo visivo, anche senza muoverlo. A questo punto può iniziare la simulazione che viene avviata premendo il tasto 'Drawing ON'.

Mentre esegue le sue attività, l'utente virtuale viene rilevato dai sensori relativi alle varie stanze; se la sua traiettoria va ad intersecare l'area di un sensore, viene registrata l'attivazione su un file txt.

Per velocizzare la simulazione e ottenere grandi quantità di dati in tempi contenuti è possibile utilizzare un acceleratore di tempo, mostrato in figura 5.



StartTime	Event Type	Meta data
2019-06-09 01:31:55	SENSOR	Door_Bedroom true FFFFFFFF006356f6
2019-06-09 01:31:55	SENSOR	pir_Bedroom false FFFFFFFF006356f6
2019-06-08 20:53:41	SENSOR	pir_Bedroom true FFFFFFFF006356f6
2019-06-08 20:53:41	SENSOR	Door_Bedroom false FFFFFFFF006356f6
2019-06-08 20:52:36	SENSOR	Door_Bedroom true FFFFFFFF006356f6
2019-06-08 17:55:01	SENSOR	fridge_Kitchen false FFFFFFFF006356f6
2019-06-08 17:53:57	SENSOR	fridge_Kitchen true FFFFFFFF006356f6
2019-06-08 16:21:03	SENSOR	fridge_Kitchen false FFFFFFFF006356f6
2019-06-08 16:19:59	SENSOR	fridge_Kitchen true FFFFFFFF006356f6
2019-06-08 11:07:10	SENSOR	fridge_Kitchen false FFFFFFFF006356f6
2019-06-08 11:06:07	SENSOR	fridge_Kitchen true FFFFFFFF006356f6
2019-06-08 08:38:01	SENSOR	fridge_Kitchen false FFFFFFFF006356f6
2019-06-08 08:36:57	SENSOR	fridge_Kitchen true FFFFFFFF006356f6
2019-06-08 08:21:06	SENSOR	Door_Bedroom false FFFFFFFF006356f6
2019-06-08 08:20:34	SENSOR	Door_Bedroom true FFFFFFFF006356f6
2019-06-08 08:20:01	SENSOR	pir_Bedroom false FFFFFFFF006356f6

Figura 5:

alto: area Track, si nota in verde il pulsante 'Drawing ON' e subito sotto l'acceleratore del tempo

basso: file scaricato in formato txt; le nuove rilevazioni sono aggiunte in cima all'elenco, in questo modo nel file il tempo cresce dal basso all'alto.

Il file contiene i seguenti dati:

- data e ora della rilevazione
- nome del sensore e relativa stanza
- tipologia di rilevazione (true/false)

Come vedremo, questo file è il punto di partenza per individuare le attività svolte.

Per ottenere grandi dataset in tempi ancora più brevi esiste un'area, mostrata in figura 6, in cui può essere effettuata una simulazione automatica dei dati: nella sezione 'Modifying TXT' basta caricare un file con almeno un giorno di attivazioni ed è possibile copiare questa giornata in modalità 'idem' o 'con variazioni', ossia con piccole variazioni sugli orari delle attivazioni, su un qualunque altro giorno del calendario.

In questo modo si possono ottenere diversi mesi di simulazione in pochissimi minuti.

The screenshot displays a user interface for generating data. It is divided into three main sections:

- Log:** A calendar for January 2019. A button labeled "Scarica File" is located below the calendar.
- Dettaglio giornaliero:** A list of sensor activation events for a specific day. Each event is represented as a timestamp followed by a boolean value and a sensor name, such as "07:17:23 |FALSE| pir_Bedroom".
- Copia:** A section for copying the data. It includes a date input field set to "03/01/2019", a "Data Base" label, a "Data Fine" label, radio buttons for "Idem" and "Con variazione", and an "Aggiungi" button.

Figura 6: generazione automatica dei dati

2.2.1 Area statistiche

In questa area sono visualizzati i risultati di diversi algoritmi statistici, utili per condurre analisi quantitative e qualitative sui dati simulati.

Le analisi avvengono in tempo reale: ogni 30 secondi la pagina si ricarica aggiornando tutti i risultati in base alle nuove attivazioni dei sensori che vengono inserite.

In figura 7 è mostrato il primo grafico, esso riporta le attivazioni di tutti i sensori nei giorni (sull'asse x) e nelle ore all'interno dei vari giorni (sull'asse y). Esso permette di visualizzare istantaneamente l'andamento delle attivazioni ed eventuali variazioni che avvengono con il trascorrere dei giorni.

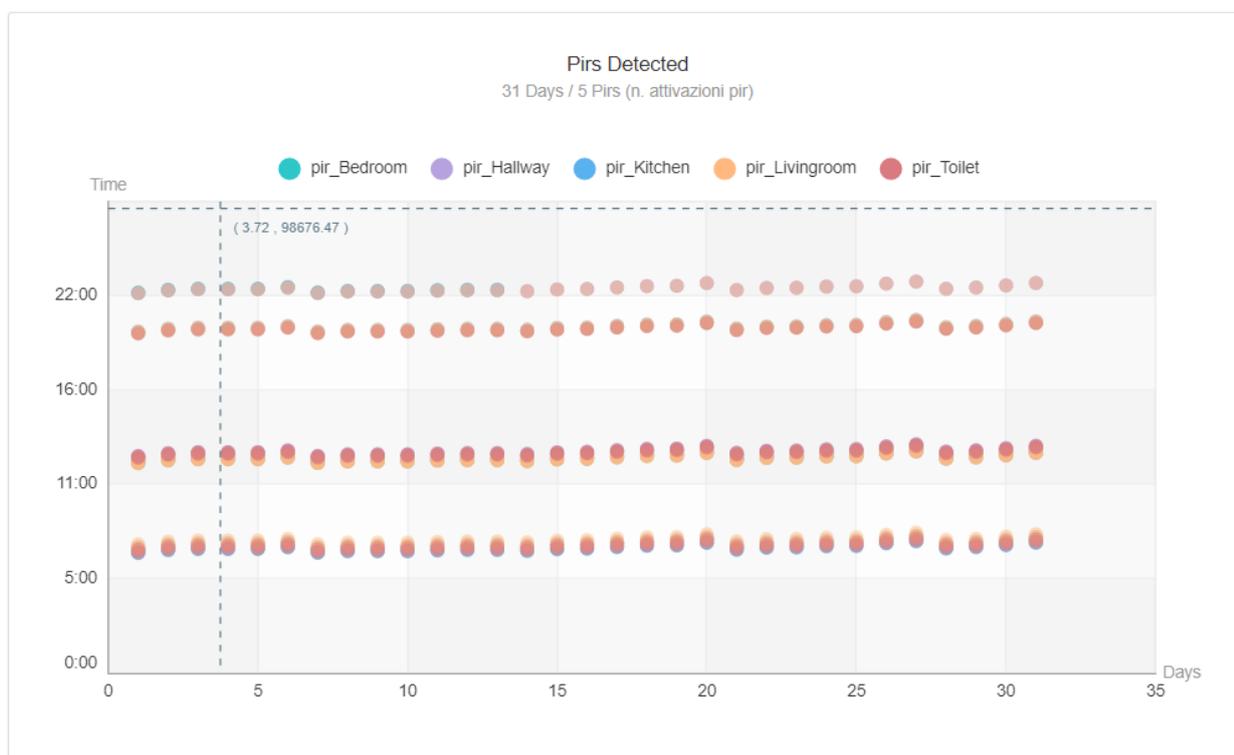


Figura 7: grafico delle attivazioni PIR

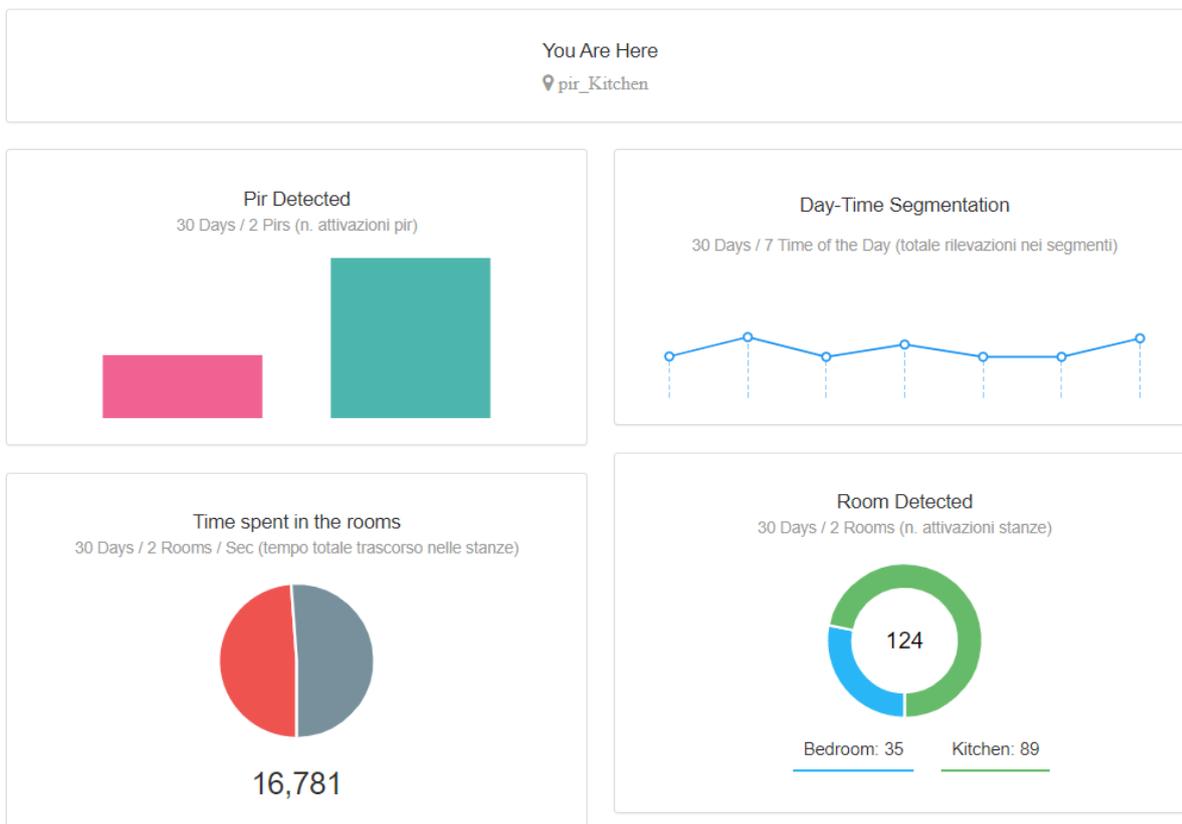


Figura 8: ulteriori grafici presenti nell'area statistiche

Si possono leggere ulteriori informazioni nei grafici di figura 8:

- L'istogramma mostra il numero di attivazioni per ogni sensore posizionato
- La linea spezzata mostra il totale delle attivazioni suddivise per fasce orarie
- Il diagramma a torta mostra il tempo totale trascorso all'interno delle varie stanze
- Il diagramma circolare mostra il numero totale di attivazioni divise per ogni stanza.

Muovendosi con il puntatore su questi grafici, vengono visualizzati in modo quantitativo e dettagliato tutti questi risultati.

2.2.2 Machine learning (ML)

Tutti i dati necessitano di essere studiati e interpretati ed è impossibile pensare di estrarre da essi tutte le informazioni necessarie, soprattutto quando i dataset sono molto grandi. Sono stati sviluppati per questo motivo diversi algoritmi di ML per lo studio delle ADLs.

Per ottenere un'analisi qualitativa completa dei dati inseriti, sono stati implementati nel tool di simulazione tre algoritmi utilizzati per estrarre le ADLs dai dati simulati in tempo reale. Il ML permette ad una macchina di interpretare in modo intelligente e automatico dei dati che non ha mai analizzato e di fare previsioni su di essi.

L'obiettivo principale è che la macchina sia in grado di generalizzare dalla sua esperienza cioè che sia in grado di sviluppare un ragionamento induttivo.

Gli algoritmi presenti nel tool di simulazione sono dei classificatori: nel ML la classificazione è un metodo di apprendimento supervisionato attraverso il quale l'algoritmo 'si allena' utilizzando dei dati che riceve in input, in una fase chiamata 'training', e quindi utilizza queste informazioni per imparare a classificare in modo autonomo nuove osservazioni.

Questa tipologia di algoritmi è detta supervisionata poiché vengono forniti ad essi i risultati desiderati per ogni istanza da classificare in una fase di training.

La performance della classificazione, cioè la sua accuratezza, viene misurata attraverso un test in cui sono forniti al classificatore nuovi dati mai visti.

Gli algoritmi presenti nel tool di simulazione sono:

- Decision Tree (DT)
- Naive Bayes (NB)
- K-Nearest Neighbors (KNN)

Vediamo brevemente il loro funzionamento:

2.2.2.1 Decision Tree[10]

È un classificatore con struttura ad albero in cui ogni nodo può essere o foglia o nodo interno: se foglia, indica il valore della classe assegnata all'istanza invece se è nodo interno, specifica il test effettuato su un attributo. Per ciascun valore assunto da un attributo in un test, l'algoritmo crea un ramo e il relativo sottoalbero.

Il criterio di suddivisione con cui viene creato un nuovo nodo si basa sul massimo guadagno di informazione: sceglie l'attributo che riesce a dividere "meglio" le istanze appartenenti a classi diverse.

Quando l'algoritmo termina, è possibile percorrere l'albero dalla radice e, seguendo il percorso risultante dai singoli test presenti su ogni nodo interno, si ottiene la classificazione dell'istanza.

2.2.2.2 Naive Bayes [11]

È un classificatore basato sul teorema di Bayes: questo descrive la probabilità di un evento in base alla conoscenza preliminare di alcune condizioni che potrebbero essere ad esso correlate.

È basato su una forte ipotesi (Naive): la presenza o l'assenza di una caratteristica non influenzano la presenza o l'assenza di altre

$$P(A_i|E) = \frac{P(E|A_i)P(A_i)}{P(E)}$$

Funzionamento:

- 1) **Calcolo della probabilità della classe:** Le probabilità di classe sono semplicemente le frequenze delle istanze che appartengono a ciascuna classe divisa per il numero totale di istanze.
- 2) **Calcolo della probabilità condizionata:** si applica il teorema di Bayes, per determinare le probabilità condizionate delle caratteristiche del problema.
- 3) **Prendere una decisione:** si calcola la probabilità per prevedere la classe di appartenenza della nuova istanza rispettando la verifica dell'indipendenza delle caratteristiche. La decisione finale è identificata nella classe che ottiene il valore di probabilità più elevato.

2.2.2.3 KNN [12]

KNN è un algoritmo di apprendimento supervisionato, il cui scopo è quello di predire una nuova istanza conoscendo certi data points che sono separati in diverse classi.

Il suo funzionamento si basa sulla somiglianza delle caratteristiche: più un'istanza è vicina a un data point, più il KNN li considererà simili.

Solitamente la somiglianza viene calcolata tramite la distanza euclidea (o un qualche altro tipo di distanza a seconda del problema in esame). Minore sarà la distanza e maggiore sarà la somiglianza tra data point e l'istanza da prevedere.

Oltre alla distanza, l'algoritmo prevede di fissare un parametro k , scelto arbitrariamente, che identifica il numero di data points più vicini. L'algoritmo valuta le k minime distanze così ottenute. La classe che ottiene il maggior numero di queste distanze è scelta come previsione.

Il KNN è uno strumento non parametrico, ossia non fa alcuna ipotesi sulla distribuzione dei dati che analizza.

In altre parole, la struttura del modello è determinata dai dati e questo è piuttosto utile perché nel “mondo reale” la maggior parte dei dati non obbedisce ai tipici assunti teorici fatti.

2.2.3 Misura della performance

L'efficienza del classificatore viene testata fornendogli un set di dati detto test set.

Valutiamo l'accuratezza[13]: è la percentuale di istanze correttamente classificate

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Figura 9: Definizione di accuratezza

Il risultato di una classificazione binaria può assumere solo due valori: positivo o negativo. Gli elementi contenuti nell'equazione di figura 9 sono:

- TP= true positives
- TN=true negatives
- FP= false positives
- FN=false negatives

TP e TN indicano le istanze correttamente classificate rispettivamente come positive e negative, FP e FN indicano le istanze classificate in modo errato rispettivamente come positive e negative cioè le istanze false che sono state classificate come vere e viceversa.

Cinque attività sono considerate all'interno del tool di simulazione:

- Waking up
- Sleep
- Breakfast
- Lunch
- Dinner

L'obiettivo del ML è quello di riconoscere l'esecuzione di queste cinque attività con una certa accuratezza.

Il tool di simulazione misura l'accuratezza attraverso la Hold Out Validation[4]: l'insieme dei dati viene suddiviso in due set, il primo 50% dei dati è utilizzato per il training set mentre il restante 50% per il test.

Vedremo questi aspetti nello specifico al capitolo 5

I risultati del test consentono di calcolare poi l'accuratezza del classificatore per le varie attività che viene mostrata nella apposita area, come in figura 10.

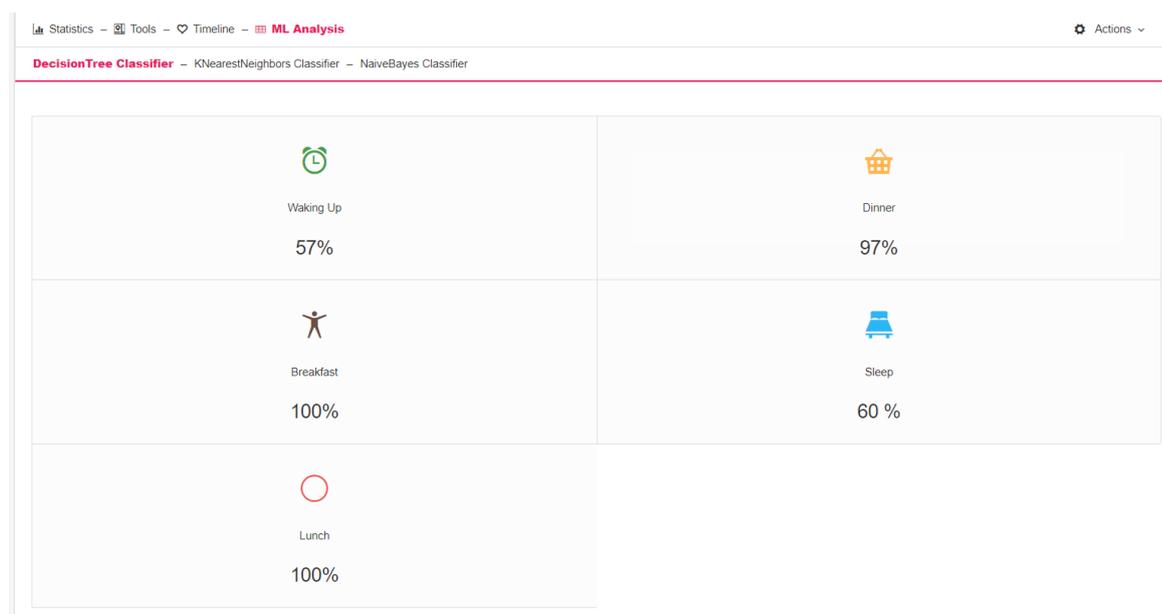


Figura 10: Area machine learning del tool di simulazione

CAPITOLO 3

ANALISI RESIDENZA 232

In questo capitolo sono descritte le caratteristiche di una residenza reale, denominata 232, in cui è stato implementato il sistema di sensori di eWare.

Andremo poi a riprodurre tutti i dati ottenuti sul tool di simulazione

La rete di sensori sviluppata nel contesto di eWare è stata realizzata e applicata in numerose abitazioni reali al fine di ottenere importanti informazioni sullo stato di salute dell'abitante.

Tutti i dati registrati sono fondamentali per valutare le nuove potenzialità e i limiti di uno Smart Environment, che dovrebbe permettere una migliore qualità di vita ai malati e una minore pressione sul sistema sanitario.

Questo studio si concentra sulla residenza numero 232 che ora è descritta nel dettaglio: la casa è abitata da una sola persona che non possiede animali domestici.

Sono stati installati i seguenti sensori:

- Contatto magnetico sulla porta principale
- Contatto magnetico sulla porta secondaria
- Contatto magnetico sullo sportello del frigorifero
- PIR in corridoio
- PIR in bagno
- PIR in sala
- PIR nella stanza hobby

L'installazione della rete è stata fatta il 18 ottobre 2019 e le rilevazioni sono proseguite da quella data fino ai primi giorni di aprile 2020.

L' unica problematica riscontrata si è verificata dal 19 febbraio 2020 alle ore 15:09, al 21 febbraio 2020 alle ore 13:33. In questo intervallo il sistema è stato scollegato dalla rete internet, problema poi risolto.

Le attività considerate nel progetto eWare sono queste:

1. Toileting
2. Eating
3. Breakfast
4. Eating_general
5. Entering_leaving
6. Door_opened
7. Inside
8. Outside
9. Sleeping_asleep
10. Sleeping_awake
11. Alarm_button
12. Visit_concurrent_events
13. Location_change

Non tutte le attività sono state registrate all'interno dell'abitazione 232, vediamo quindi quelle che saranno analizzate successivamente:

- 1) Toileting: si attiva il sensore PIR in bagno
- 2) Eating: si attiva il sensore posizionato sul frigorifero
- 5) Entering Leaving: la porta principale viene aperta e poi chiusa, senza uscire di casa
- 7) Inside: il residente entra in casa utilizzando una porta esterna
- 8) Outside: il residente esce di casa utilizzando una porta esterna
- 9) Sleeping Asleep: il residente va a letto o è inattivo durante le ore della notte/sera
- 10) Sleeping Awake: il residente si alza dal letto o diventa attivo durante le ore della notte/mattino

Tutti i dati registrati sono stati riprodotti sul tool di simulazione.

Bisogna sottolineare alcuni aspetti:

- La porta secondaria non è stata utilizzata quasi mai, il relativo sensore è stato trascurato.
- I contatti magnetici relativi alla porta principale e al frigorifero sono stati simulati con dei sensori PIR poiché essi sono l'unica tipologia di sensori disponibili nel programma. Consideriamo quindi che le porte vengono aperte quando il sensore fornisce come output 'true', mentre vengono chiuse quando l'output è 'false'.
- Non sono stati riprodotti i giorni dal 18 al 22 ottobre, contenuti nell'elenco dei dati reali ottenuti, che rappresentano il giorno dell'installazione e successive prove effettuate in casa prima dell'inizio delle rilevazioni.
- Non sono stati riprodotti i giorni 1 e 2 aprile 2020 poiché, confrontando mese per mese i dati reali e quelli simulati, si sarebbe ottenuto un confronto poco significativo rispetto ai mesi completi di rilevazioni.

CAPITOLO 4

ALGORITMI E CONFRONTO CON DATI REALI

Dopo aver riprodotto l'intero set di dati sul tool di simulazione, vediamo come riconoscere le attività svolte partendo dal file txt che può essere scaricato.

Confrontiamo poi i dati relativi all'abitazione reale con quelli ottenuti dagli algoritmi

L'intero periodo di rilevazioni della residenza 232 è stato riprodotto sul tool di simulazione. Attraverso il tool è possibile scaricare il file in formato txt che contiene semplicemente l'elenco di tutte le attivazioni dei sensori posizionati con la relativa data e ora.

Questi dati senza un'opportuna analisi non sono molto significativi.

La maggior parte delle attività, infatti, non si limita all'accensione di un sensore in una stanza ma possono comprendere rilevazioni multiple di più sensori a distanza di un certo tempo l'una dall'altra.

Per riconoscere nel file le attività svolte e contare quante volte sono state effettuate è necessario individuare, per ognuna di essa, una sequenza caratteristica che si ripete ogni volta durante l'esecuzione.

Viste le grandissime dimensioni del file, costituito da migliaia di righe, non si può pensare di effettuare tutta questa analisi a mano.

È necessario programmare un algoritmo per riconoscere queste sequenze e andare a contare il numero di volte in cui ogni attività viene svolta.

In questo modo si possono ottenere in poco tempo il numero esatto di attività svolte ed effettuare un confronto con i dati reali.

4.1 INTRODUZIONE AGLI ALGORITMI

In informatica un algoritmo non è altro che un semplice procedimento che permette la risoluzione di uno specifico problema mediante una sequenza finita di istruzioni[14].

Queste istruzioni devono essere interpretate ed eseguite fino alla loro conclusione seguendo un preciso ordine.

Le principali strutture di un algoritmo sono 3:

- 1) **Sequenza:** le operazioni indicate vengono eseguite in successione, una dopo l'altra
- 2) **Selezione o controllo:** permette di eseguire azioni diverse nel caso in cui si verifica o meno una certa condizione (istruzioni 'if'/'else')
- 3) **Iterazione:** consente di ripetere delle azioni come stampare a video un testo per un certo numero di volte, cercare qualcosa all'interno di un testo o di un vettore (ciclo 'for'/'while').

Per mostrare graficamente, in modo chiaro e immediato un algoritmo si possono utilizzare i diagrammi di flusso [15], questi mostrano la sequenza delle operazioni identificando la tipologia di ognuna di esse tramite una diversa forma geometrica.

In figura 11 [16] sono riportate le principali istruzioni che possono essere presenti in un algoritmo.

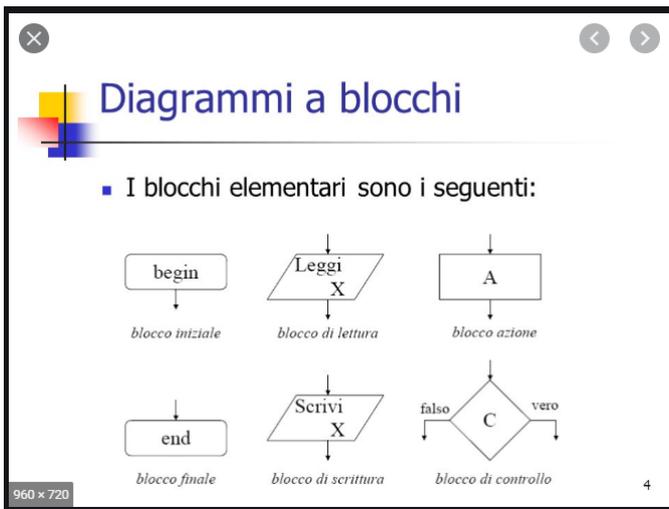


Figura 11: Blocchi elementari di un diagramma di flusso, sono utilizzate le seguenti forme:

Rettangolo raccordato=inizio e fine dell'algoritmo;

Quadrilatero=lettura o scrittura dei dati;

Rettangolo semplice=operazioni matematiche

Rombo=controllo, funzioni 'if' o iterazioni

Vediamo ora gli algoritmi per contare le attività svolte dall'utente virtuale

4.2 TOILETING E EATING

Sono le attività più semplici da riconoscere, si limitano ad una singola accensione di un sensore.

In particolare l'attività Toileting viene riconosciuta ogni volta che è presente un'accensione del PIR posizionato in bagno; lo stesso vale per l'attività Eating ma tramite il sensore posizionato sul frigorifero.

	A	B	C
pirdataexport16			
	StartTime	EventType	Metadata
	Datetime	▼ Categorical	▼ Categorical
4	2019-01-30 07:20:15	SENSOR	pir_Bedroom false FFFFFFFF...
5	2019-01-30 07:18:39	SENSOR	pir_Bedroom true FFFFFFFF...
6	2019-01-29 23:00:32	SENSOR	pir_Bedroom false FFFFFFFF...
7	2019-01-29 22:54:11	SENSOR	pir_Bedroom true FFFFFFFF...
8	2019-01-29 20:57:01	SENSOR	pir_Kitchen false FFFFFFFF0...
9	2019-01-29 20:55:57	SENSOR	pir_Kitchen true FFFFFFFF00...
10	2019-01-29 07:21:05	SENSOR	pir_Bedroom false FFFFFFFF...
11	2019-01-29 07:19:29	SENSOR	pir_Bedroom true FFFFFFFF...

Figura 12: finestra import su Matlab: il txt viene importato come vettori colonne. Metadata è il vettore che contiene l'elenco delle attivazioni, StartTime è il vettore che contiene i rispettivi orari.

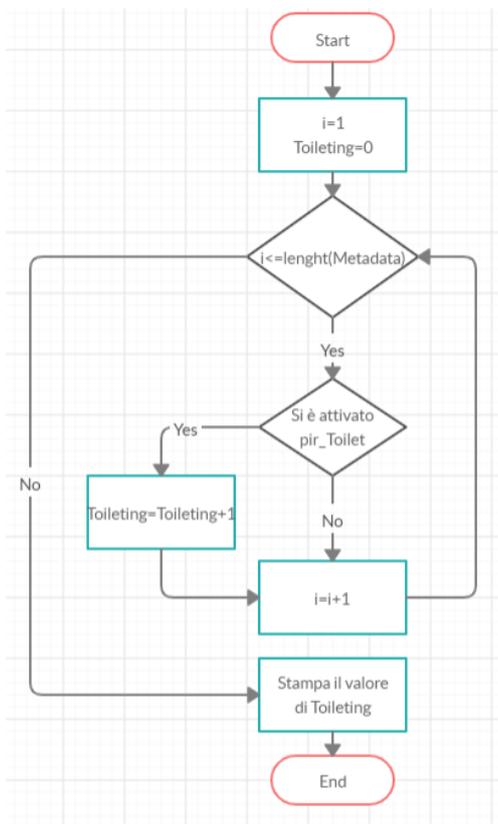


Figura 13: ALGORITMO TOILETING

Utilizzo due contatori: 'i' e 'Toileting'

Il contatore 'i' viene incrementato di un'unità per ogni iterazione di un ciclo 'for' [17] fino a raggiungere la dimensione del vettore Metadata.

Per ogni posizione di questo vettore viene verificato se il dato contenuto è 'pir_Toilet==true'

Se questo si verifica, incremento di una unità il contatore 'Toileting'.

Quando sono state controllate tutte le posizioni, l'algoritmo si interrompe stampando a video il numero di volte in cui l'attività è stata eseguita.

ALGORITMO EATING

Il funzionamento è totalmente analogo, ma viene ricercato il dato 'fridge_Kitchen==true'.

4.3 ENTERING LEAVING

L'attività corrisponde all'apertura e chiusura della porta principale senza uscire di casa. L'abitante deve quindi trovarsi in casa dopo le rilevazioni della porta.

La sequenza da individuare sui dati in uscita dal tool è la seguente:

- Una qualunque attivazione in casa
- La porta si apre
- La porta si chiude
- Una qualunque attivazione in casa

In figura 14 e in figura 15 sono presenti la sequenza da ricercare nei dati e il diagramma a blocchi dell'algoritmo.

2019-01-01 08:33:02	SENSOR	pir_Hallway	false	FFFFFFFF006356f6
2019-01-01 08:32:54	SENSOR	pir_Hallway	true	FFFFFFFF006356f6
2019-01-01 08:32:50	SENSOR	Door_Entrance	false	FFFFFFFF006356f6
2019-01-01 08:32:42	SENSOR	Door_Entrance	true	FFFFFFFF006356f6
2019-01-01 08:32:38	SENSOR	pir_Hallway	false	FFFFFFFF006356f6
2019-01-01 08:32:34	SENSOR	pir_Hallway	true	FFFFFFFF006356f6
2019-01-01 08:32:34	SENSOR	pir_Livingroom	false	FFFFFFFF006356f6

Figura 14: sequenza caratteristica dell'attività *Entering_Leaving*. Il tempo cresce dal basso all'alto!

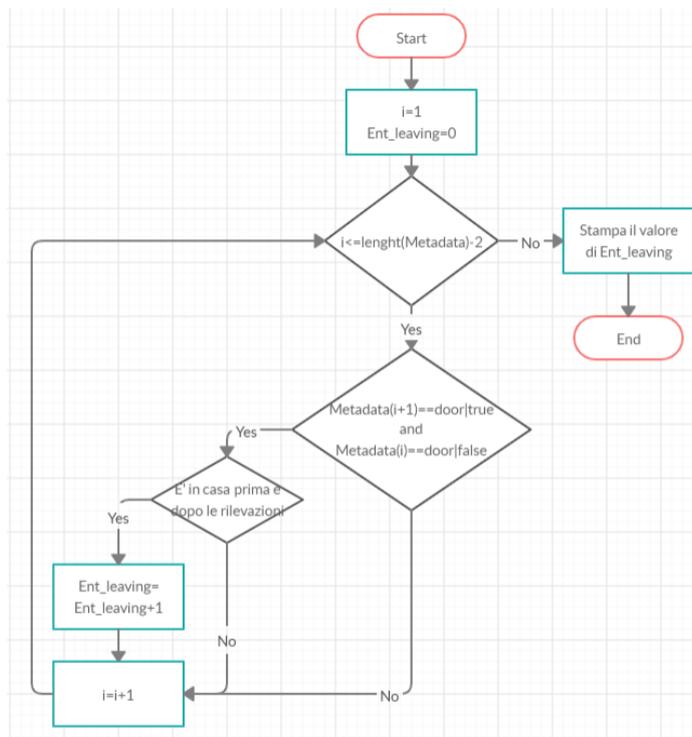


Figura 15: ALGORITMO ENTERING LEAVING

Il vettore 'Metadata' viene controllato, come nel caso precedente, in ogni sua posizione; in questo caso cerco una specifica successione di attivazioni dei sensori, come già spiegato.

Se viene identificata tale sequenza nel vettore, significa che l'attività desiderata è stata svolta: incremento il contatore 'Ent_leaving'.

4.4 OUTSIDE

L'attività Outside, cioè uscire di casa, viene identificata dalla seguente sequenza:

- La porta viene chiusa
- La porta viene aperta
- Le due rilevazioni sono separate da un certo intervallo di tempo

Una rete di sensori reali potrebbe rilevare l'attività quando, a distanza di un certo intervallo di tempo dopo che la porta principale viene chiusa, non ci sono nuove attivazioni dei sensori.

Il tool di simulazione, però, fornisce solo una sequenza dei dati già registrati quindi non possiamo rilevare l'uscita con il metodo appena descritto poiché non è possibile lavorare con il tempo in questo modo. Gli unici riferimenti di tempo sono presenti in relazione alle accensioni dei sensori.

Non è inoltre sufficiente ricercare la semplice rilevazione della porta che si apre e si chiude poiché si farebbe confusione con l'attività Entering Leaving.

Per poter scrivere un algoritmo che ci visualizza le uscite bisogna considerare che la persona sia già rientrata a casa in modo da avere una rilevazione registrata con il suo orario che va a definire in modo univoco l'attività.

Il controllo del tempo tra le due rilevazioni serve più che altro per evitare di conteggiare eventuali errori di simulazione e quindi è stato posto pari ad 1 minuto.

La sequenza tipica si può osservare in figura 16.

In figura 17 è riportato il diagramma a blocchi dell'algoritmo.

2019-03-06 18:25:15	SENSOR	pir_Livingroom	true	FFFFFFFF006356f6
2019-03-06 18:25:13	SENSOR	Door_Entrance	false	FFFFFFFF006356f6
2019-03-06 18:25:09	SENSOR	Door_Entrance	true	FFFFFFFF006356f6
2019-03-06 16:21:19	SENSOR	Door_Entrance	false	FFFFFFFF006356f6
2019-03-06 16:20:31	SENSOR	Door_Entrance	true	FFFFFFFF006356f6
2019-03-06 16:20:15	SENSOR	pir_Hallway	false	FFFFFFFF006356f6

Figura 16: sequenza caratteristica dell'attività Outside

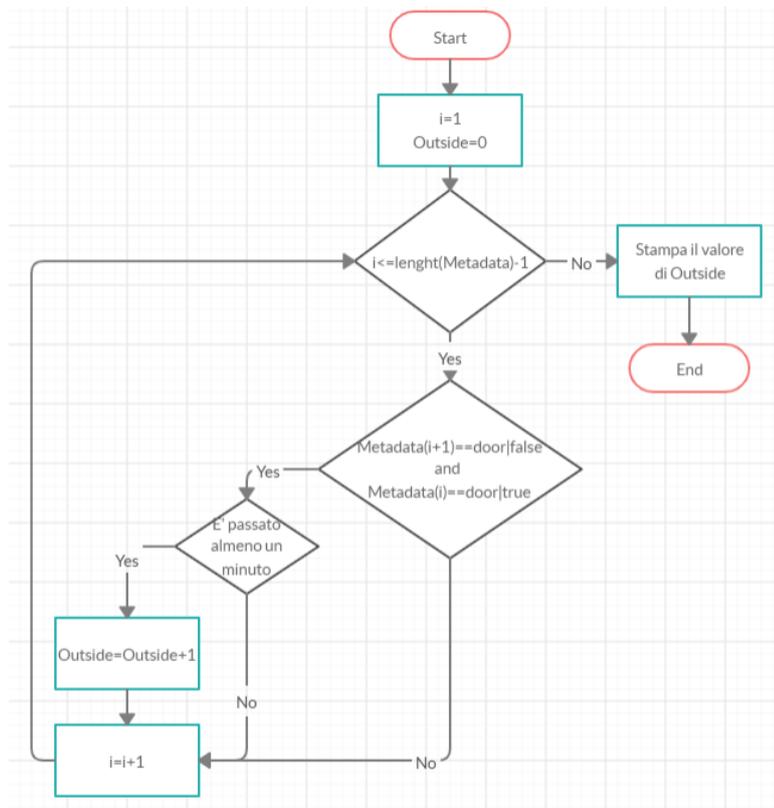


Figura 17: ALGORITMO OUTSIDE

Vado a ricercare nel vettore la sequenza di attivazioni del sensore porta. Il funzionamento è analogo agli algoritmi precedenti

4.5 INSIDE

Siccome ad ogni uscita da casa corrisponde sempre un successivo ingresso, un solo algoritmo era sufficiente per individuare entrambe le attività Inside e Outside.

È stato creato un algoritmo a parte per evidenziare meglio la differenza tra questa attività e l'attività Entering Leaving.

Inoltre, considerando che il numero di rilevazioni dell'attività Inside e Outside devono essere uguali, è possibile anche a verificare la correttezza degli algoritmi che devono ovviamente fornire lo stesso risultato.

Viene rilevata l'attività Inside se, in successione:

- La porta viene aperta
- La porta viene chiusa
- La porta viene aperta

Le prime due attivazioni si verificano quando l'utente virtuale esce di casa: apre la porta (Door_Entrance=='true') poi esce e la chiude (Door_Entrance== 'false').

A questo punto, appena il sensore sulla porta esterna si attiva, si può concludere che l'utente è rientrato in casa visto che, provenendo dall'esterno, il primo sensore ad attivarsi sarà proprio questo (Door_Entrance=='true').

Nell'attività Entering Leaving invece, prima di aprire la porta, l'utente virtuale non doveva essere uscito ma doveva trovarsi in casa (cioè doveva essere registrata una rilevazione qualsiasi diversa dal sensore della porta).

Il diagramma a blocchi è riportato in figura 18.

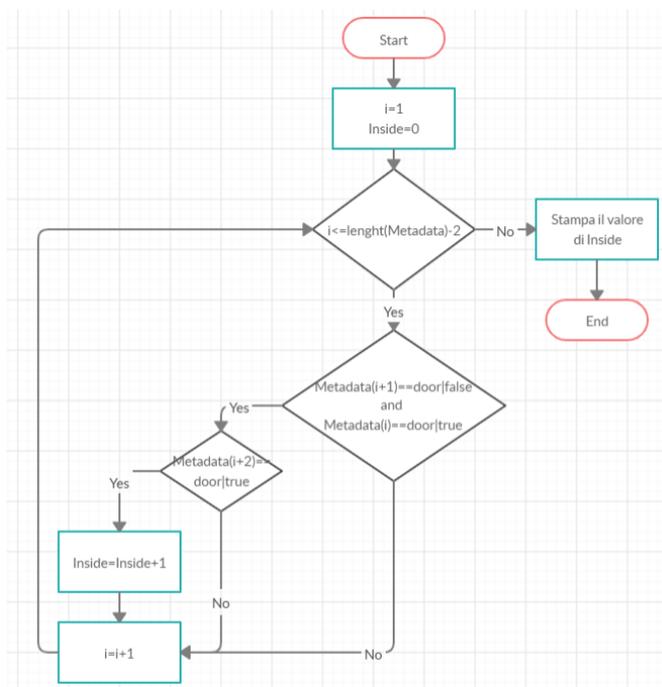


Figura 18: ALGORITMO INSIDE

4.6 SLEEPING ASLEEP/SLEEPING AWAKE

Riportiamo prima di tutto in figura 19 un'immagine della residenza 232 con i relativi sensori, così come è stata riprodotta sul tool di simulazione:



Figura 19: tool di simulazione, residenza 232

Non essendo presente un sensore PIR in camera da letto, è necessario utilizzare gli altri per verificare le attività di andare a dormire e svegliarsi.

In questo caso le due attività vengono conteggiate contemporaneamente.

Osservando la figura 19, si nota che, prima di entrare in camera da letto, è necessario passare attraverso il sensore del corridoio oppure il sensore del bagno, uno di questi due quindi è necessariamente l'ultimo ad attivarsi prima che l'utente virtuale vada a dormire.

Allo stesso modo, all'uscita dalla camera, sarà uno di questi due sensori a visualizzare per primo il passaggio.

Inoltre, per evitare di conteggiare le attività legate al sonno per ogni ingresso in camera effettuato anche durante il giorno, è necessario che la permanenza all'interno sia di una certa durata.

Quindi le attività sono conteggiate quando, in sequenza:

- Si attiva il sensore del corridoio o del bagno
 - Si attiva nuovamente uno dei due sensori
 - Tra le due rilevazioni è trascorsa almeno mezz'ora
- come si può vedere in figura 20.

```

2019-01-02 06:39:12  SENSOR  pir_Hallway | false | FFFFFFFF006356f6
2019-01-02 06:38:24  SENSOR  pir_Hallway | true  | FFFFFFFF006356f6
2019-01-01 20:40:55  SENSOR  pir_Hallway | false | FFFFFFFF006356f6
2019-01-01 20:39:51  SENSOR  pir_Hallway | true  | FFFFFFFF006356f6
2019-01-01 20:39:51  SENSOR  pir_Livingroom | false | FFFFFFFF006356f6
    
```

Figura 20: esempio di sequenza tipica delle attività Sleeping. Si notano in questo caso lo spegnimento e l'attivazione del sensore nel corridoio (pir_Hallway) separate da tutta la durata del sonno.

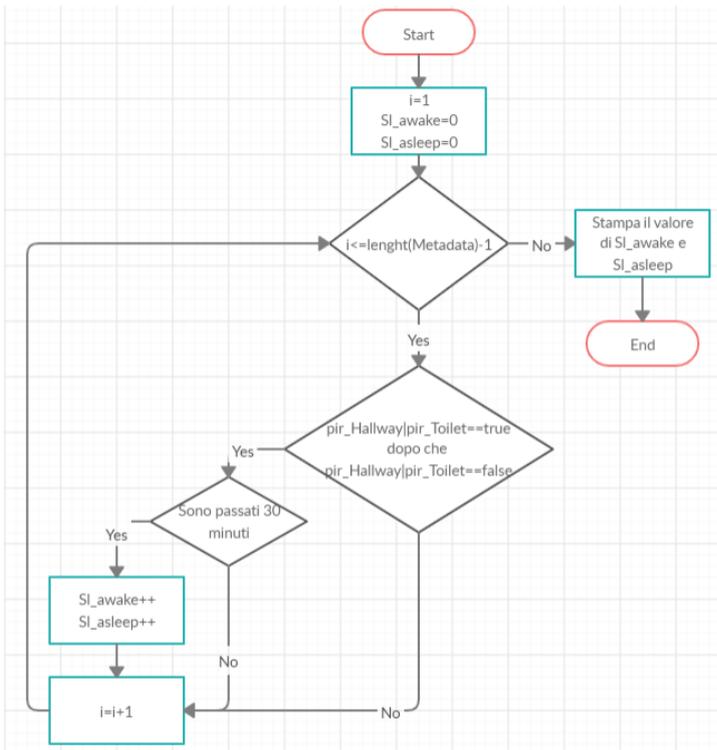


Figura 21: ALGORITMO SLEEPING ASLEEP/AWAKE

4.7 CONFRONTO CON DATI REALI

Sono riportati ora i grafici relativi alle attività svolte nella residenza reale e un immediato confronto con quelli che derivano dagli algoritmi.

I valori ricavati in output dagli algoritmi sono stati utilizzati per calcolare, rispetto al totale delle attività svolte, la percentuale di esecuzione di ognuna.

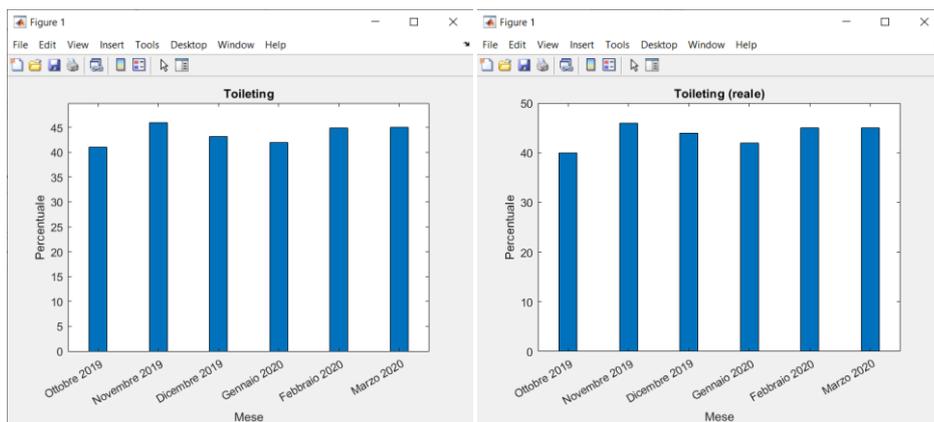


Figura 22: confronto attività Toileting

L'attività Toileting è quella prevalente durante tutto il periodo di registrazione e quella per cui si registrano il maggior numero di attivazioni; la percentuale è sempre maggiore o uguale al 40%

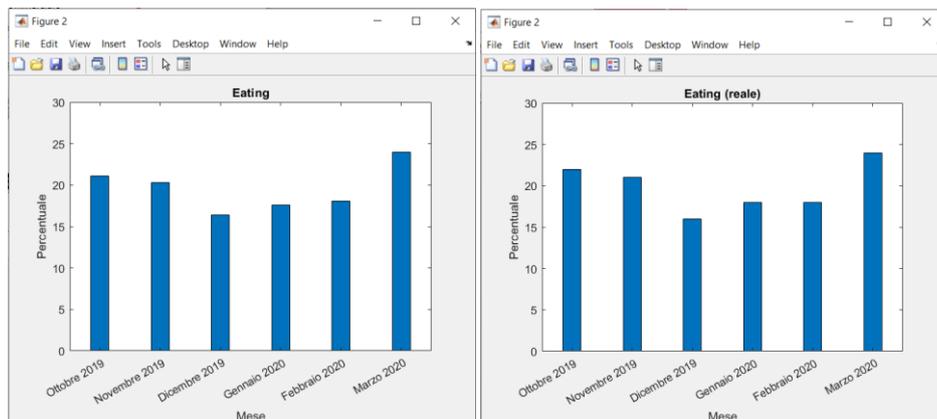


Figura 23: confronto attività Eating

Anche l'attività Eating, legata ad un bisogno fisiologico, ha un numero di rilevazioni, e dei conseguenti valori percentuali, considerevoli.

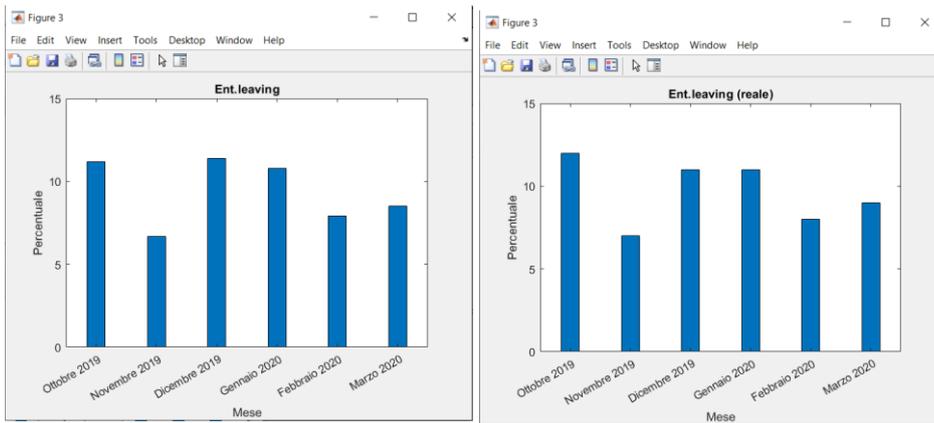


Figura 24: confronto attività Entering Leaving

L'attività Entering Leaving assume valori percentuali sempre diversi durante il corso dei mesi, oscillando tra il 7% e il 12%.

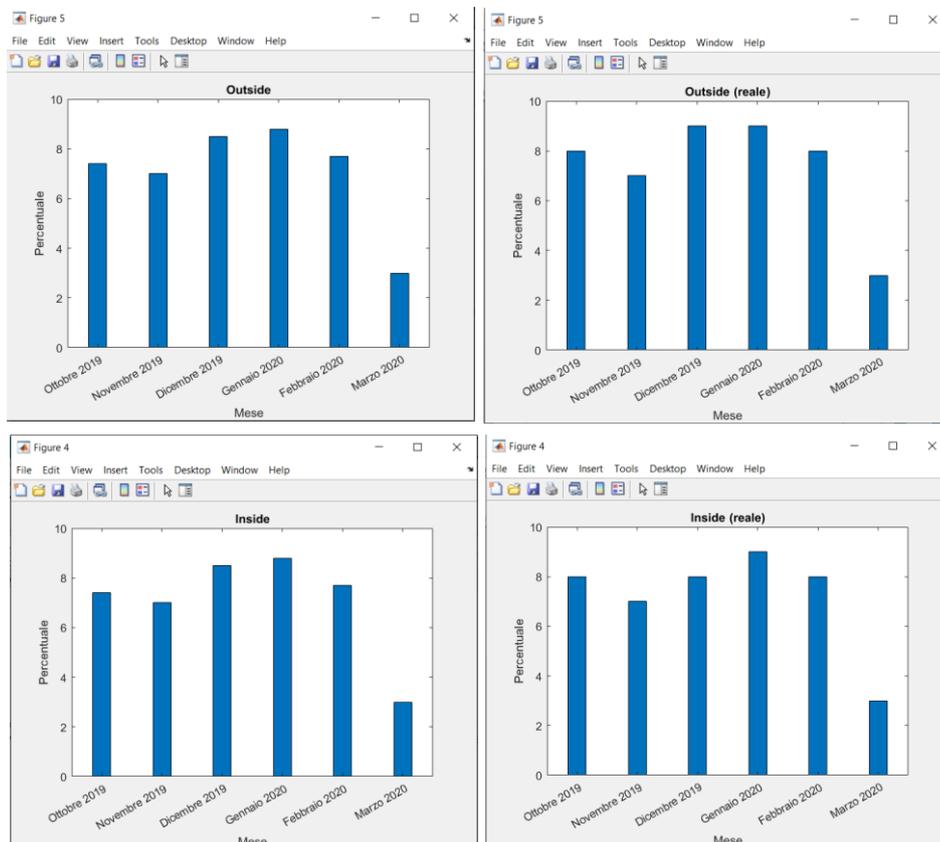


Figura 25: confronto attività Outside e Inside

Le attività Inside e Outside mantengono i loro valori percentuali senza importanti variazioni nell'arco temporale tra ottobre e febbraio, mentre diminuiscono in modo considerevole a marzo a causa dell'emergenza covid-19.

Si nota quindi che il residente ha rispettato le regole dettate dalla particolare situazione.

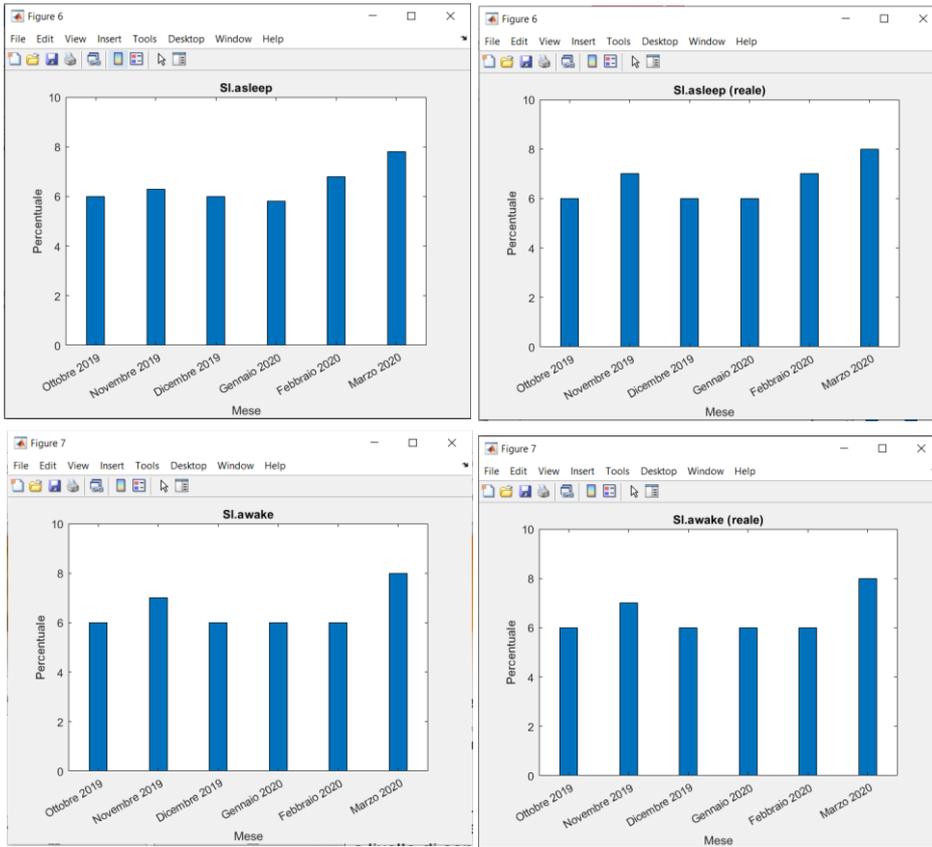


Figura 26: confronto attività Sleeping Asleep e Sleeping Awake

Le attività presentano gli stessi valori percentuali durante il corso di tutti i mesi di registrazione. Fa eccezione il mese di febbraio a causa delle problematiche riscontrate a livello di connessione tra il 19 e il 21. Infine le attività di sonno e risveglio del paziente mantengono delle percentuali piuttosto costanti e aumentano solo nel mese di marzo, sempre a causa delle restrizioni dettate dall'emergenza coronavirus.

Tutti i dati riportati possono essere raccolti nella tabella di figura 27:

DATI SIMULATI						
	Ottobre 2019	Novembre 2019	Dicembre 2019	Gennaio 2020	Febbraio 2020	Marzo 2020
1-toileting	41%	46%	43,2%	42%	44.9%	45%
2-eating	21.1%	20.3%	16,4%	17.6%	18.1%	24%
5-entering leaving	11.2%	6.7%	11,4%	10.8%	7.9%	8,5%
7-inside	7.4%	7%	8.5%	8.8%	7.7%	3%
8-outside	7.4%	7%	8.5%	8.8%	7.7%	3%
9-sleeping asleep	6%	6.3%	6%	5.8%	7%	7.8%
10-sleeping awake	6%	6.3%	6%	5.8%	6.8%	7.8%

DATI REALI						
	Ottobre 2019	Novembre 2019	Dicembre 2019	Gennaio 2019	Febbraio 2019	Marzo 2020
1 - toileting	40 %	46 %	44 %	42 %	45 %	45 %
2 - eating	22 %	21 %	16 %	18 %	18 %	24 %
5 - entering leaving	12 %	7 %	11 %	11 %	8 %	9 %
7 - inside	8 %	7 %	8 %	9 %	8 %	3 %
8 - outside	8 %	7 %	9 %	9 %	8 %	3 %
9 - sleeping asleep	6 %	7 %	6 %	6 %	7 %	8 %
10 - sleeping awake	6 %	7 %	6 %	6 %	6 %	8 %

Figura 27: residenza 232, confronto con dati reali. Riepilogo

CAPITOLO 5

OTTIMIZZAZIONE RETE DI SENSORI

In questo capitolo gli algoritmi di machine learning implementati nel tool di simulazione sono utilizzati per andare a cercare il numero e la disposizione ottimale di sensori.

L'ottimizzazione è fatta considerando delle tipiche traiettorie all'interno dell'abitazione e verificando i risultati nell'accuratezze di misura relative a diverse disposizioni dei sensori.

Una Smart Home è un'abitazione in cui viene collocata una rete intelligente di sensori utile per rilevare le attività, monitorare le abitudini e lo stato di salute di un abitante. Risulta fondamentale la progettazione dell'installazione dei sensori: la rete va ottimizzata per ottenere i migliori risultati al minor costo.

Vista la complessità del comportamento umano, non esistono in letteratura configurazioni ottimali già sviluppate ma è necessario andare a studiare nel dettaglio la disposizione per ogni singolo individuo considerando le sue abitudini e i suoi spostamenti tipici all'interno della casa.

La simulazione dei dati consente di risolvere alcune problematiche relative ai tempi e ai costi di acquisizione necessari per questi studi.

Come già visto nel capitolo 2, il tool di simulazione utilizzato contiene tre algoritmi di classificazione: Decision Tree, Naive Bayes e K-Nearest Neighbour.

La loro accuratezza viene misurata tramite la Hold Out Validation e pertanto l'insieme dei dati inseriti viene suddiviso in due set di dimensione pari al 50% del totale

- Training set
- Test set

Il training set è quell'insieme di dati che vengono utilizzati per addestrare il sistema supervisionato. Questo set quindi contiene tutti i risultati corretti della classificazione e tramite esso l'algoritmo apprende quali caratteristiche discriminano gli elementi appartenenti alle differenti classi.

Una volta effettuata la fase di apprendimento, la correttezza dell'algoritmo è verificata eseguendo lo stesso sul test set.

Per studiare la migliore configurazione è stato simulato un mese di dati per diverse disposizioni dei sensori.

All'interno del mese, le prime due settimane costituiscono il set di dati di training, mentre le seconde due sono quelle per il test.

Il training set è sviluppato seguendo le caratteristiche in figura 28:

Attività	Intervallo	Durata	Stanza
Wake up	6.30-7.30	1-5min	Bedroom
Breakfast	7.00-8.30	10-30min	Kitchen
Lunch	12.00-13.30	40-120min	Kitchen
Dinner	19.00-21-00	40-120min	Kitchen
Sleep	22.00-6.30	5-8h	Bedroom

Figura 28: caratteristiche del training test

Per ognuna delle attività sono specificati:

- un intervallo di tempo nel quale si deve svolgere
- una durata
- una stanza nella quale deve trovarsi il relativo sensore (o sensori).

L'attivazione di un sensore nel giusto intervallo di tempo e nella giusta stanza viene riconosciuta come una attività svolta tra quelle della prima colonna della tabella.

Andiamo ad ipotizzare, in figura 29, una mappa degli spostamenti all'interno dell'abitazione:

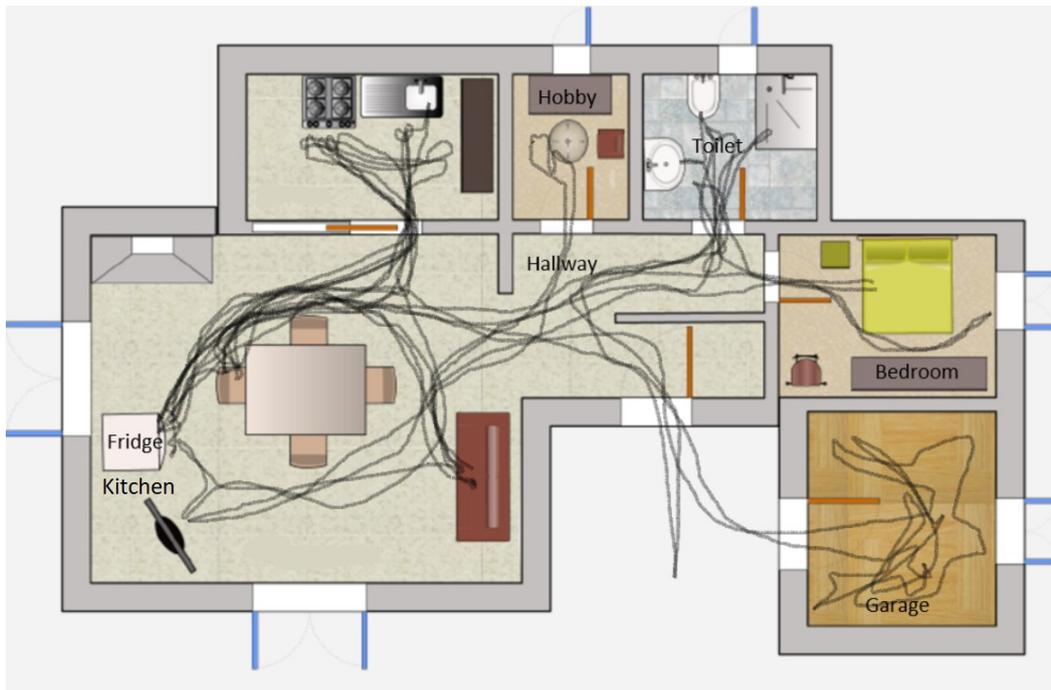


Figura 29: *mappa delle traiettorie*

La figura mostra le traiettorie tipiche dell'utente virtuale all'interno della casa, nelle zone in cui le linee sono più dense significa che il passaggio in quell'area è molto frequente invece dove le linee sono meno dense il passaggio è poco frequente.

5.1 1°CASO

Come configurazione iniziale consideriamo la disposizione di sensori della residenza reale 232. Come abbiamo visto in figura 28, è necessaria la presenza di un sensore nelle stanze denominate 'Kitchen' e 'Bedroom'. Pertanto aggiungiamo un sensore PIR in camera da letto visto che non era presente nel caso reale.

La disposizione è mostrata in figura 30:



Figura 30: disposizione dei sensori, 1°caso; il sensore in camera è posizionato in modo da coprire l'area attorno alla sua porta di ingresso.

È stato simulato con questa configurazione un mese di acquisizioni, le settimane di training sono costruite seguendo le indicazioni in tabella 28 e viene poi simulato il test set seguendo le ipotesi dei movimenti interni nell'abitazione.

Queste simulazioni sono state condotte tramite la funzione di generazione automatica dei dati presente nel programma, come spiegato al capitolo 2.

I risultati del test per i tre algoritmi sono riportati in figura 31:

Accuracy					
	<u>Waking up</u>	<u>Sleep</u>	<u>Breakfast</u>	<u>Lunch</u>	<u>Dinner</u>
DT	100	100	100	100	100
KNN	0	0	100	100	100
NB	100	0	3	100	100

Figura 31: tabella delle accuratèzze, 1° caso

5.2 2° CASO

Abbiamo visto che le attività sono riconosciute solo se svolte all'interno di due specifiche stanze denominate 'Kitchen' e 'Bedroom'.

Tutti i sensori che si trovano in altre stanze quindi non forniscono alcuna informazione utile agli algoritmi di ML per quanto riguarda le cinque attività considerate. Proviamo a togliere tutti i sensori lasciando solo i sensori nelle due stanze indicate come mostrato in figura 32.



Figura 32: disposizione dei sensori, 2° caso; sono presenti solo il sensore frigo per la stanza 'Kitchen' e il sensore nella stanza 'Bedroom'. Le loro posizioni sono uguali a quelle che avevano nel 1° caso.

Ripetiamo la stessa prova di prima simulando le stesse settimane di training e test e vediamo i risultati ottenuti in figura 33:

Accuracy					
	<u>Waking up</u>	<u>Sleep</u>	Breakfast	Lunch	<u>Dinner</u>
DT	100	100	100	100	100
KNN	0	0	100	100	100
NB	100	0	23	100	100

Figura 33: tabella delle accuratezze, 2° caso.

Come previsto, solo i sensori nelle due stanze indicate contribuiscono al calcolo dell'accuratezza e la situazione rispetto al 1° caso risulta quindi invariata.

Si può inoltre notare come l'unico algoritmo che fornisce risultati ragionevoli è il Decision Tree.

Dimostriamo ora che le posizioni di questi due sensori sono effettivamente le migliori per rilevare le attività, iniziamo con il sensore della stanza 'Kitchen'.

5.3 3°CASO

Ricordando gli spostamenti ipotizzati in figura 29, si nota che le traiettorie nella stanza 'Kitchen' risultano più addensate nella parte superiore e dal lato del frigorifero mentre sono presenti meno passaggi dal lato del divano.

Abbiamo visto inoltre che l'attivazione di un sensore nella stanza 'Kitchen' nei giusti intervalli di tempo viene registrata come una attività legata all'attività mangiare.

Proviamo a posizionare il sensore relativo a questa stanza come in figura 34, cioè in una zona che viene attraversata con più bassa frequenza:

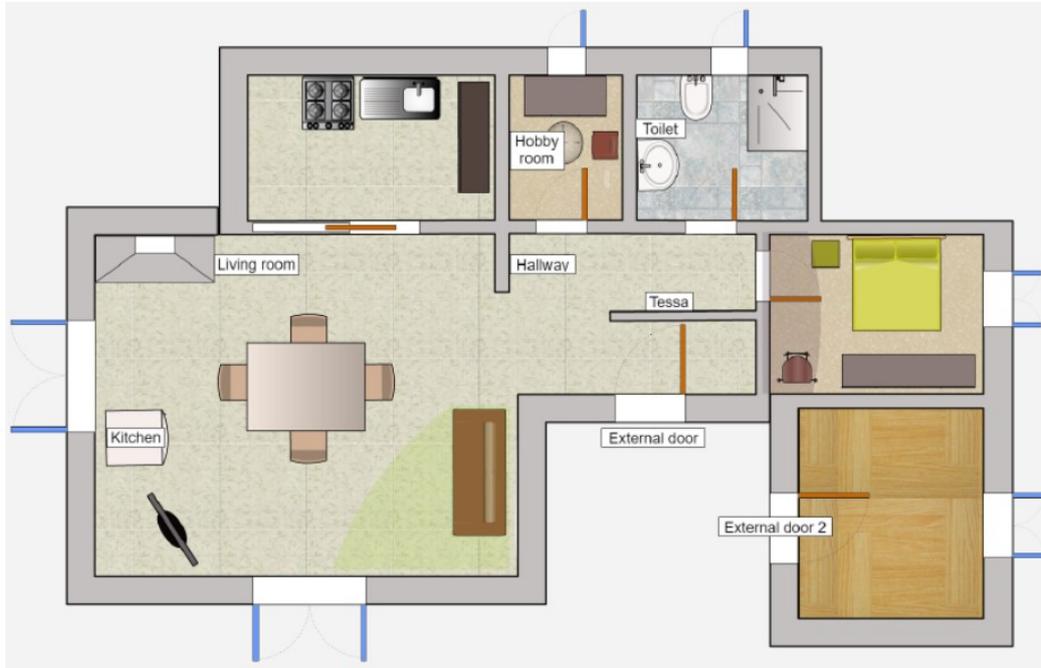


Figura 34: disposizione dei sensori, 3° caso; il sensore della stanza 'Kitchen' è stato spostato dal lato opposto rispetto al frigorifero.

I risultati sono mostrati in figura 35:

Accuracy					
	<u>Waking up</u>	<u>Sleep</u>	<u>Breakfast</u>	<u>Lunch</u>	<u>Dinner</u>
DT	97	100	67	67	60
KNN	0	0	100	67	97
NB	100	0	3	67	97

Figura 35: tabella delle accuratezze, 3° caso

Considerando solo il DT, che risulta essere il migliore algoritmo, notiamo che in questo caso le percentuali relative alle accuratezze delle attività 'Breakfast', 'Lunch' e 'Dinner' si abbassano. Questo è dovuto alla posizione errata del sensore che è collocato in una zona che l'utente non attraversa sempre nei tipici orari dei pasti.

L'assenza delle attivazioni nella stanza 'Kitchen' negli specifici intervalli di tempo porta ad avere valori più bassi di accuratezza.

Si può dire che, prima di mangiare, l'utente apre sistematicamente il frigorifero quindi, per avere risultati migliori, il sensore della cucina va posizionato sopra di esso. In questo modo il sensore viene sempre attivato all'interno dei periodi di tempo corretti.

5.4 4°CASO

Con le stesse modalità viste nel 3° caso, spostiamo questa volta il sensore della camera in un'altra posizione, mostrata in figura 36:

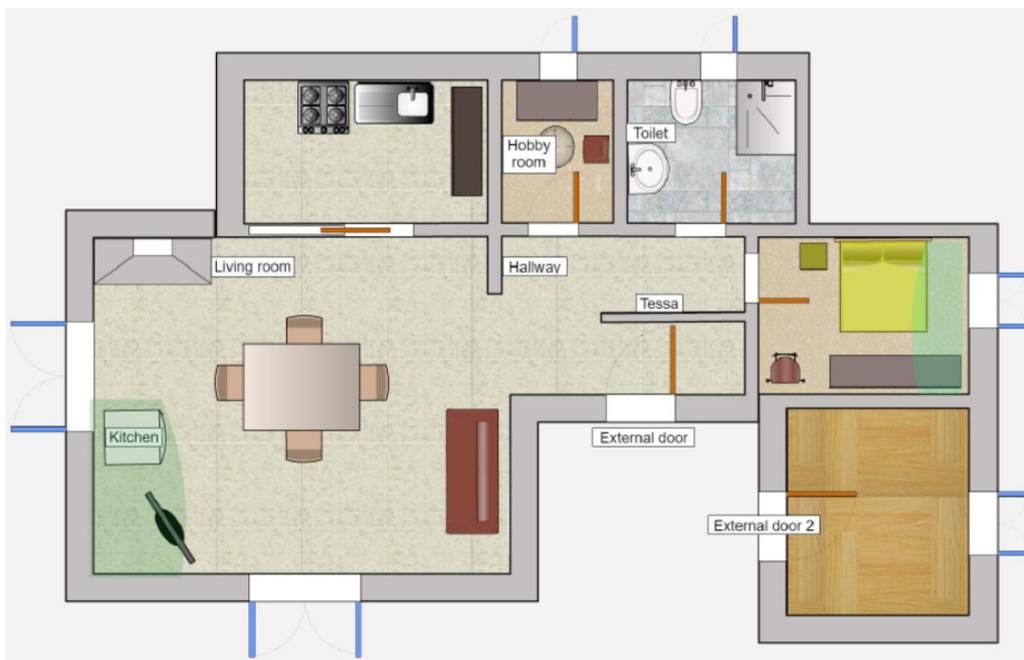


Figura 36: disposizione dei sensori, 4° caso

Il sensore è stato inserito nel lato più esterno della stanza in modo da coprire l'area di fronte alla finestra.

Il sensore in cucina è stato ricollocato nella migliore posizione.

Ripetendo le simulazioni otteniamo i risultati di figura 37:

Accuracy					
	Waking up	Sleep	Breakfast	Lunch	Dinner
DT	57	60	100	100	97
KNN	0	0	100	100	97
NB	83	0	23	100	97

Figura 37: tabella delle accuratezze, 4° caso

È possibile notare che i risultati delle attività legate alla stanza 'Bedroom' sono notevolmente peggiorati.

Il motivo è lo stesso visto nel 3° caso, in questa nuova posizione, il sensore copre con il suo raggio d'azione un'area che non sempre viene attraversata dall'utente e le mancate attivazioni nei tempi definiti dal training portano ad un peggioramento dell'accuratezza.

La posizione migliore per riconoscere le attività 'Waking-up' e 'Sleep' è quindi quella ipotizzata nel 2° caso dove il sensore copre l'ingresso della stanza da letto, un'area sicuramente attraversata ogni volta che l'utente va a dormire o esce dalla stanza al mattino.

5.5 STIMA DEI COSTI

Calcoliamo per ognuno dei quattro casi e per i tre algoritmi l'accuratezza media:

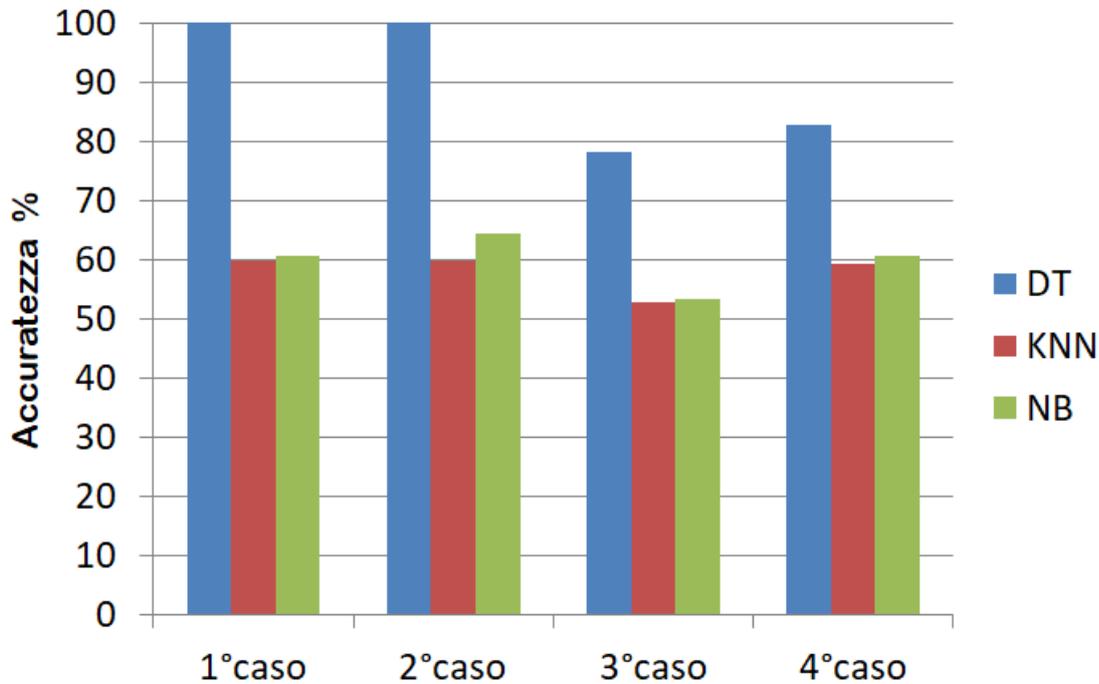


Figura 38: istogramma riassuntivo: per i quattro casi sono riportate le accuratèzze medie totali per ognuno dei tre algoritmi

Come già sottolineato, il Decision Tree risulta essere l'algoritmo che fornisce i migliori risultati in tutte le configurazioni studiate.

Si può osservare in figura 38 che i primi due casi portano entrambi allo stesso livello di accuratezza, la differenza sta nel numero di sensori da cui è costituita la rete:

- Nel 1° caso ci sono 7 sensori PIR
- Nel 2° caso ci sono soltanto 2 sensori PIR.

Per valutare l'installazione di una rete di sensori reali sulla base di queste analisi vanno sicuramente considerati i costi: confrontando i due casi migliori, abbiamo che essi forniscono la stessa accuratezza ma il secondo caso presenta ben 5 sensori in meno. La migliore configurazione quindi risulta essere quella del 2° caso.

Va sottolineato che queste analisi e il risultato ottenuto sono ovviamente limitati dalle potenzialità del tool di simulazione che è stato utilizzato, infatti è stato programmato per riconoscere solo alcune delle attività che erano invece presenti nel progetto reale eWare e inoltre un'ulteriore rigidità è presente nella definizione di due sole stanze nel quale le tali attività vanno svolte.

Ovviamente in una installazione reale risulteranno molto comodi anche ulteriori sensori come per esempio quello sulla porta.

Restando all'interno dei limiti del programma e per massimizzare i suoi output sono sufficienti due soli sensori posizionati in modo corretto.

Risulterebbe inoltre inutile posizionare un secondo sensore rispettivamente all'interno delle stanze 'Kitchen' e 'Bedroom' poiché quelli già presenti, se ben posizionati, forniscono un risultato ottimo.

La migliore configurazione realizzata è quindi quella del 2° caso e il migliore algoritmo risulta essere il DT.

CAPITOLO 6

6.1 CONCLUSIONI

In questa tesi è stato utilizzato il simulatore UNIVPM Home-Care che consente di condurre analisi in tempo reale sui dati acquisiti da sensori PIR virtuali permettendo di indentificare variazioni nel comportamento di un utente virtuale.

La simulazione facilita lo studio e la progettazione di una rete di sensori reali tramite la creazione dei dataset necessari al suo sviluppo, permettendo di accelerare i tempi di lavoro e risparmiando sui costi dei sensori.

Con algoritmi programmati per lavorare sui dati in output dal tool di simulazione è possibile ricostruire esattamente le attività svolte da un utente virtuale, confrontando i dati relativi ad una residenza reale con i risultati ottenuti sui dati simulati e si è riscontrata una perfetta corrispondenza, a meno di eventuali arrotondamenti.

Il simulatore contiene tre algoritmi di ML che consentono di classificare cinque attività fondamentali: Waking Up, Sleep, Breakfast, Lunch, Dinner; è stato possibile trovare una configurazione di sensori massimizza la loro accuratezza.

Il migliore algoritmo è il Decision Tree e l'accuratezza massima si può ottenere con due soli sensori posizionati rispettivamente nelle stanze 'Kitchen' e 'Bedroom'.

Posizioni differenti da quelle ottimali portano a scarsa accuratezza e inoltre un numero più elevato di sensori risulta solamente più costoso senza alcun miglioramento nei risultati.

Il simulatore consente quindi di svolgere analisi fondamentali per lo sviluppo delle reti di sensori reali che vanno a costituire gli ambienti Smart: questi permettono di

fornire assistenza ad un numero sempre crescente di malati, migliorando la loro situazione di vita e alleggerendo il carico sulle strutture sanitarie.

6.2 FUTURE WORKS

Il tool di simulazione utilizzato offre la possibilità di condurre un'analisi 'a posteriori' riguardante la migliore configurazione di sensori per massimizzare l'accuratezza e ridurre i costi. Uno sviluppo futuro potrebbe permettere di ottenere direttamente una configurazione ottimizzata partendo da una ipotetica mappa degli spostamenti. Inoltre il simulatore potrebbe essere programmato per funzionare con ulteriori tipologie di sensori e per riconoscere tramite il ML numerose altre attività.

BIBLIOGRAFIA

- [1] Majumder S., Aghayi E., Noferesti M., Memarzadeh-Tehran H., Mondal T., Pang Z., & Deen, M. (2017): *“Smart homes for elderly healthcare - Recent advances and research challenges”*.
- [2] Casaccia S., Revel G.M., Scalise L., Bevilacqua R., Rossi L., Paauwe R.A., Karkowsky I., Ercoli I., Serrano J.A., Suijkerbuijk S., Luikkien D., Nap H.H., (2019) *“Social robot and sensor network in support of activity of daily living for people with dementia”*
- [3] Gina Sprint, Diane Cook, Roschelle Fritz, and Maureen Schmitter-Edgecombe Washington State University, Pullman, WA 99164-2752, *“Using Smart Homes to Detect and Analyze Health Events”*
- [4] Pirozzi Michela (XVIII edizione), *“Development of a simulation tool for measurements and analysis of simulated and real data to identify ADLs and behavioral trends through statistics techniques and ML algorithms”*
- [5] Miguel Pires I., Marques G., Garcia N.M., Florez-Revuelta F., Teixeira M.C., Zdravevski E., Spinsante S., Coimbra M.,(2020) *“Pattern recognition techniques for the identification of activities of daily living using a mobile device accelerometer”*
- [6] Jaeseok Yun, Sang-Shin Lee: *“Human Movement Detection and Identification Using Pyroelectric Infrared Sensors”*. Sensors (2014)

[7] Yang D., Xu B., Rao K., Sheng W., (2018), *“Passive Infrared (PIR)- Based indoor position tracking for smart homes using accessibility maps and A-star algorithm”*, School of Information Science and Engineering, Northeastern University Shenyang 110819 China, School of Physics, Northeast Normal University, Changchun 130000, China; School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74074, USA

[8] Gina Sprint, Diane Cook, Roschelle Fritz, and Maureen Schmitter-Edgecombe *“Using Smart Homes to Detect and Analyze Health Events”* Washington State University, Pullman, WA 99164-2752

[9] Renoux J., Klugl F., (2018), *“Simulating daily activities in a smart home for data generation”*, School of Science and Technology, Orebro University, Sweden

[10] <https://vitolavecchia.altervista.org/caratteristiche-degli-alberi-di-decisione-decision-tree-in-informatica/>

[11] <https://lorenzogovoni.com/naive-bayes>

[12] <https://lorenzogovoni.com/knn/>

[13] <https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b#:~:text=We%20can%20use%20classification%20performance,primarily%20used%20by%20search%20engines>

[14] Informaticapertutti.com

[15] <https://creately.com/it/lp/generatore-diagramma-a-blocchi/>
per la creazione dei diagrammi di flusso

[16] <https://slideplayer.it/slide/5372578/>

[17] *“Introduzione alla programmazione in MATLAB, parte 3 (strutture iterative)”*,
dispense dell’Università degli Studi di Salerno