



UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea Triennale in **Ingegneria Biomedica**

Monitoraggio delle attività della vita quotidiana tramite sensori inerziali

Monitoring of activities of daily living using inertial sensors

Relatore:

Ing. Alessandro Mengarelli

Correlatore:

Ing. Federica Verdini

Ing. Andrea Tigrini

Tesi di Laurea di:

Martina Druda

A.A. 2023 / 2024

INDICE

1. INTRODUZIONE	3
2. MATERIALI E METODI	5
2.1 Sensori utilizzati	5
2.2 Protocollo Sperimentale	6
2.3 Elaborazione dei dati	13
2.3.1 Segmentazione dei segnali	13
2.3.2 Estrazione delle features	14
2.4 Classificazione	18
3. RISULTATI	23
3.1 Classificazione Totale	23
3.2 Classificazione ternaria	29
3.3 Classificazione binaria	32
4. DISCUSSIONE	40
5. CONCLUSIONE	42
BIBLIOGRAFIA	43

1. INTRODUZIONE

La Human Activity Recognition (HAR) è un campo di studio che riguarda il riconoscimento di attività di routine quotidiana svolte dalle persone, sfruttando delle registrazioni di serie temporali tramite diverse tipologie di sensori (Gupta, 2021).

Ogni giorno le persone si dedicano ad un'ampia gamma di attività e movimenti. Esistono due tipi di attività fisica che si possono fare: le attività motorie semplici e le attività complesse. Le attività semplici comprendono i movimenti del corpo umano, come camminare, riposare, guidare, salire le scale. La seconda categoria, le attività complesse, comprende funzioni umane come leggere, usare il computer, guardare la televisione, fare sport, preparare il cibo o pulire la casa (Athota, Sumathi, 2022).

Per l'acquisizione dei segnali relativi a queste attività si possono sfruttare diversi tipi di sensori, ma in molti studi ci si concentra sull'utilizzo di sensori indossabili semplici, a basso costo e facilmente reperibili, come ad esempio gli Inertial Measurement Unit (IMU), che comprendono al loro interno accelerometri e giroscopi.

Negli approcci basati sui sensori indossabili, i rilevatori di movimento sono collegati a varie parti del corpo, ad esempio possono essere integrati in fasce, anelli, smartphone, o smartwatch. In genere più i dispositivi si trovano vicini al centro di massa e più accurata è la rappresentazione dei gesti (Athota, Sumathi, 2022).

Dunque, negli ultimi anni, grazie alla rapida diffusione di smartwatch e smartphone dotati di IMU a prezzi accessibili, è stato possibile osservare un rapido sviluppo di applicazioni per il rilevamento di movimenti in molti campi diversi quali l'analisi degli sport, la sanità, i fitness tracker, l'assistenza agli anziani e la sicurezza. Gli smartphone, infatti, possono essere facilmente portati in tasca e gli smartwatch possono essere indossati sulla mano dominante e possono essere molto utili per rilevare movimenti umani complessi, come lavarsi i denti, mangiare, scrivere, ecc. (Gupta, 2021).

Uno degli ambiti più interessanti della HAR, riguarda il monitoraggio dell'aderenza alle terapie di farmaci nelle persone malate o anziane che si trovano a casa da sole. Nell'ultimo decennio, infatti, si è capito quanto una scarsa aderenza alle terapie prescritte, può limitare notevolmente i benefici delle cure mediche ed aumentare il numero di ricoveri in ospedale e persino i decessi. Nello studio di Kalantarian et al. (2016) emerge come le tecniche adottate finora, basate su promemoria, confezioni semplificate dei farmaci e incentivi economici, non siano in realtà del tutto efficaci nell'aumento dell'aderenza. Per questo motivo è importante porre enfasi sullo sviluppo di tecnologie avanzate, basate sui dispositivi indossabili e su Machine Learning e Deep Learning, per il rilevamento di questa attività (Moccia et al. 2022).

Lo studio proposto in questo elaborato ha dunque l'obiettivo di inserirsi in questo contesto di riconoscimento delle attività della vita quotidiana, con un particolare focus su movimenti rilevanti per il monitoraggio della salute da remoto, come il prendere una pillola o bere con un bicchiere.

Per raggiungere l'obiettivo preposto sono stati raccolti dati da un gruppo di soggetti che hanno indossato due sensori inerziali (IMU) sul polso della mano dominante e hanno poi eseguito una serie di movimenti che simulano delle attività quotidiane. Successivamente i dati ottenuti sono stati analizzati e utilizzati per addestrare dei modelli di Machine Learning, i quali hanno il compito di identificare e classificare i gesti specifici. Alla fine, è stato fatto un confronto tra i vari modelli utilizzati sulla base dell'accuratezza ottenuta, per comprendere quali fossero i più efficaci nel rispondere alle richieste.

2. MATERIALI E METODI

2.1 Sensori utilizzati

I sensori indossabili utilizzati in questo studio sono due MIMU (Magnetic Inertial Measurement Unit) della tipologia Xsens MTw Awinda. Una MIMU è un dispositivo che include al suo interno tre diversi sensori, un giroscopio triassiale, un accelerometro triassiale ed un magnetometro triassiale, le cui informazioni possono essere fuse tra loro mediante algoritmi di sensor fusion, per poter stimare l'orientazione del sensore (Fig.1). (Paulich, Schepers)

- **Giroscopio:** il giroscopio 3D è un sensore inerziale che rileva la velocità angolare attorno ad ognuno dei tre assi. Quando integrato nel tempo, fornisce una stima dell'entità di rotazione attorno ad un determinato asse.
- **Accelerometro:** l'accelerometro 3D è un sensore inerziale che misura l'accelerazione lineare. Quando il sensore non è in movimento, l'accelerazione misurata è pari all'accelerazione gravitazionale.
- **Magnetometro:** un magnetometro 3D è in grado di misurare la forza e la direzione del campo magnetico terrestre. Eventuale presenza di disturbi elettromagnetici nei dintorni possono introdurre rumore nel segnale registrato dal magnetometro. Il vettore del campo magnetico terrestre viene spesso utilizzato come riferimento per la direzione. (Paulich, Schepers)

Il sistema di riferimento di questi sensori è definito come segue: l'asse X è rivolto in verticale, con direzione positiva verso la scritta "Xsens"; l'asse Y è in orizzontale con verso positivo rivolto a sinistra; l'asse Z è rivolto in direzione uscente dal sensore.

In alcuni soggetti, inoltre, è stato utilizzato un ulteriore sensore, un MYO Armband, con lo scopo di rilevare i segnali EMG (Fig.2). Il MYO è composto da 8 sensori elettromiografici, una Inertial Measurement Unit (IMU), che comprende un accelerometro triassiale e un giroscopio triassiale, e un modulo Bluetooth che consente di inviare i dati ad un computer.

Per entrambi i sensori utilizzati, la frequenza di campionamento adottata è di 100 Hz.



Fig.1: XSens MTw Awinda



Fig.2: MYO Armband

2.2 Protocollo Sperimentale

Per questo progetto sono stati inclusi 7 soggetti sani, uomini e donne, di età compresa tra i 23 e i 30 anni, tutti destrimani. L'acquisizione dei segnali è avvenuta nel Laboratorio di Analisi del Movimento, Ancona. Due sensori inerziali sono stati posizionati sul polso della mano dominante, uno nella parte superiore e uno nella parte inferiore (Fig.3). Per alcuni soggetti è stato aggiunto un ulteriore sensore di supporto sull'avambraccio, in grado di rilevare i segnali EMG.



Fig.3: Posizionamento sensori

A ciascun soggetto è stato chiesto di eseguire 14 azioni che riguardano attività della vita quotidiana, tenendo conto che ogni azione comprende: 1-Posizione di partenza con mano appoggiata sul tavolo; 2-Esecuzione del movimento; 3-Ritorno alla posizione di partenza.

I movimenti eseguiti sono brevemente descritti di seguito:

1. Bere da un bicchiere: consiste nel prendere un bicchiere con la mano, portarlo alla bocca simulando la bevuta e rimetterlo sul tavolo.
2. Mangiare con una forchetta: consiste nel prendere la forchetta, simulare la presa di cibo da un piatto, portarla alla bocca e infine rimettere la forchetta sul tavolo.
3. Bere da una tazzina di caffè: consiste nel prendere una tazzina in mano, portarla alla bocca e rimetterla sul tavolo.
4. Indossare gli occhiali: si prendono dal tavolo gli occhiali, vengono indossati per qualche istante e poi si appoggiano di nuovo sul tavolo.
5. Scrivere: si prende una penna dal tavolo, si simula la scrittura del proprio nome e si rimette la penna sul tavolo.
6. Aprire/chiudere tappo di una bottiglia: consiste nel ruotare il tappo di una bottiglia aprendola e poi ruotare in senso opposto per chiuderla.

7. Indossare le cuffie: consiste nel prendere le cuffie dal tavolo, portarle alle orecchie per qualche secondo e poi riportarle sul tavolo.
8. Versare acqua: consiste nel sollevare una bottiglietta, simulare il versamento di acqua in un bicchiere e riporre la bottiglietta sul tavolo.
9. Prendere una pillola: consiste nel premere un blister, prendere una pillola e portarla alla bocca, poi rimettere la mano sul tavolo.
10. Abbassare una maniglia: si posiziona la mano sulla maniglia di una porta, la si abbassa e poi si lascia la presa.
11. Lavare le mani: si sollevano le mani, si strofinano tra loro simulandone il lavaggio e poi si riappoggiano sul tavolo.
12. Rispondere al telefono: consiste nel sollevare il telefono e portarlo all'orecchio simulando una chiamata per pochi secondi, poi si ripone sul tavolo.
13. Fumare: si prende una sigaretta dal tavolo, si porta alla bocca per pochi secondi e si ripone sul tavolo.
14. Sbadigliare: si solleva una mano e la si porta alla bocca simulando uno sbadiglio, poi si torna alla posizione di riposo.

Ogni movimento è stato ripetuto 10 volte, e ciascuna ripetizione è caratterizzata da un tempo di azione di 8s e un tempo di riposo di 3s. Per alcune azioni più rilevanti come bere da un bicchiere e prendere una pillola, che vogliamo poi distinguere rispetto a tutte le altre, è stato necessario aumentare le ripetizioni a 60, in modo da evitare di avere uno sbilanciamento nella quantità di dati considerati nel training dei classificatori

Inoltre, è importante precisare che ai soggetti non è stata imposta alcuna modalità per eseguire i movimenti, ma sono stati eseguiti in modo naturale, per rendere l'esperimento più realistico possibile. Nelle figure sottostanti (Fig.4-5) sono riportati due esempi di attività.



Fig.4: Esecuzione dell'attività "Bere con bicchiere"



Fig.5: Esecuzione dell'attività "Prendere una pillola"

Di seguito sono riportati i grafici di alcuni segnali acquisiti durante l'esecuzione delle attività (Fig. 6-19).

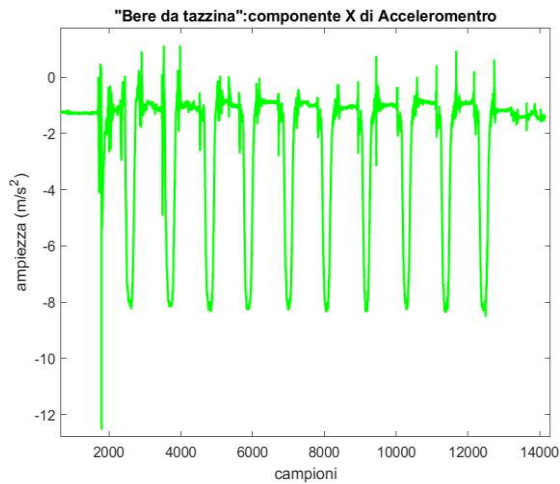


Fig. 6: "Bere da tazza". Componente X dell'Accelerometro

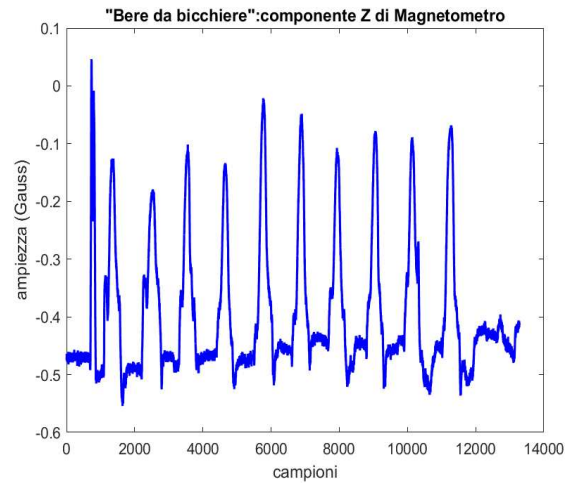


Fig. 7: "Bere da bicchiere". Componente Z del Magnetometro

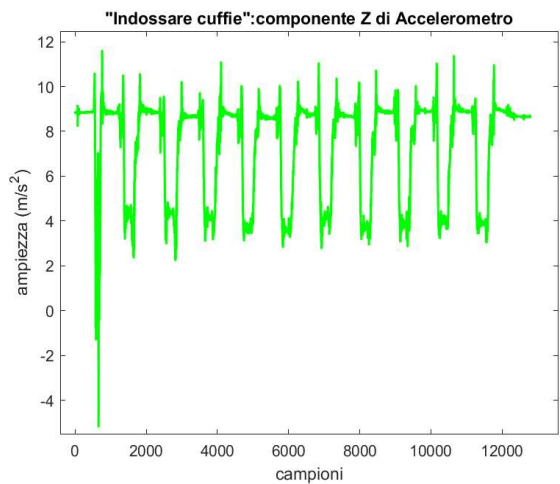


Fig. 8: "Indossare cuffie". Componente X dell'Accelerometro

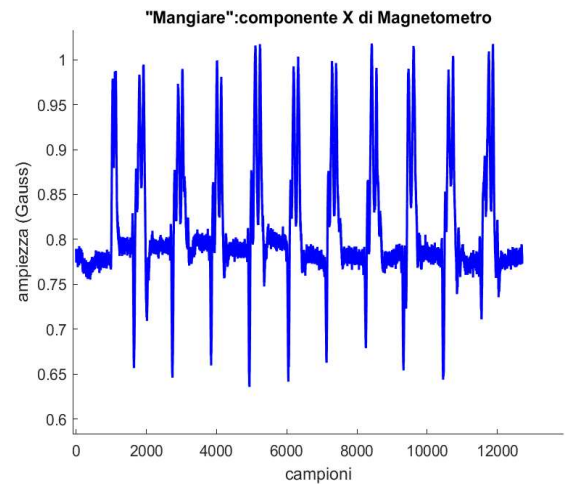


Fig. 9: "Mangiare". Componente X del Magnetometro

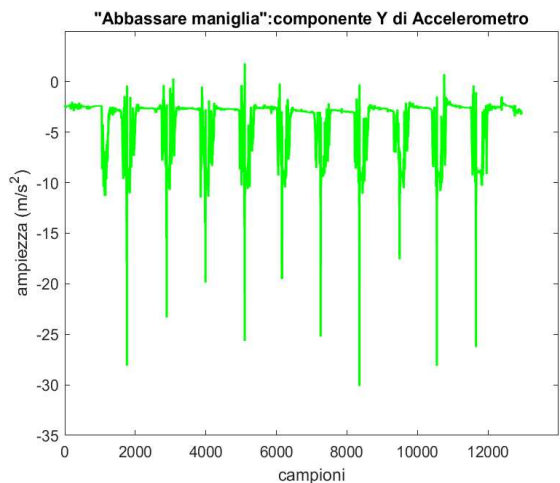


Fig. 10: "Abbassare maniglia". Componente Y dell'Accelerometro

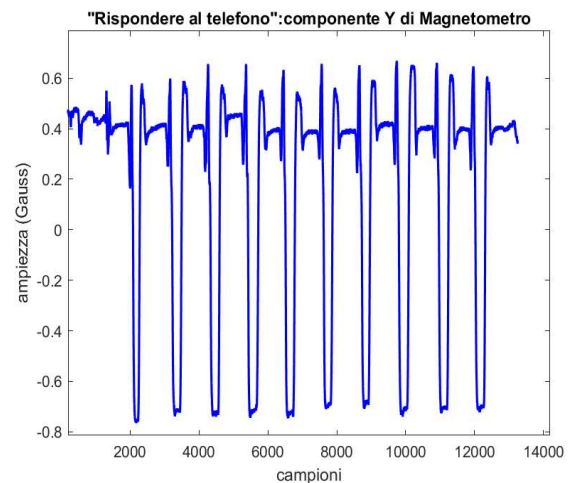


Fig. 11: "Rispondere al telefono". Componente Y del Magnetometro

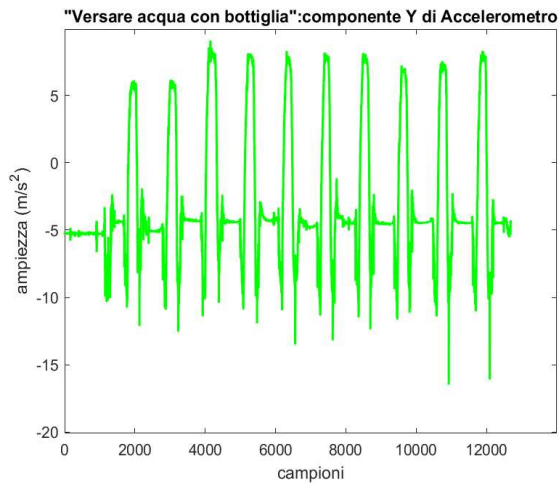


Fig.12: "Versare acqua". Componente Y dell'Accelerometro

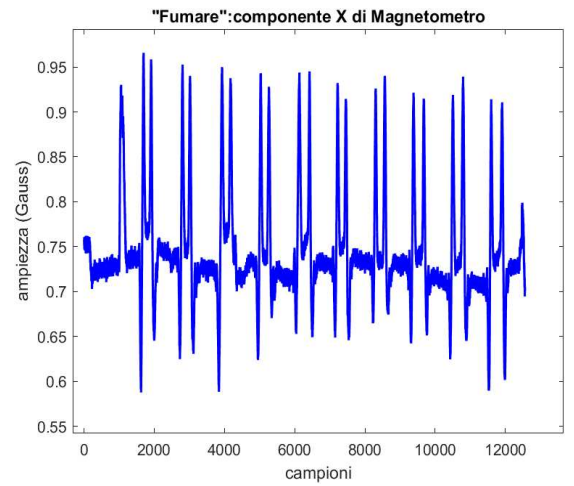


Fig.13: "Fumare". Componente X del Magnetometro

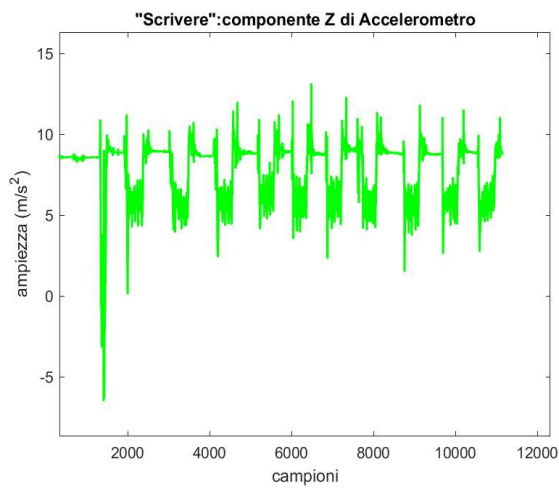


Fig.14: "Scrivere". Componente Z dell'Accelerometro

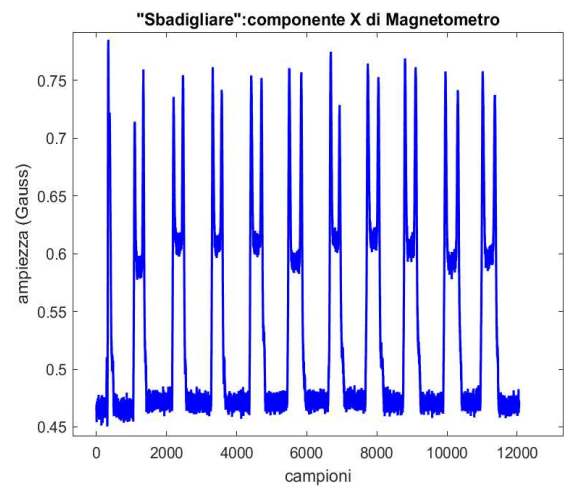


Fig.15: "Sbadigliare". Componente X del Magnetometro

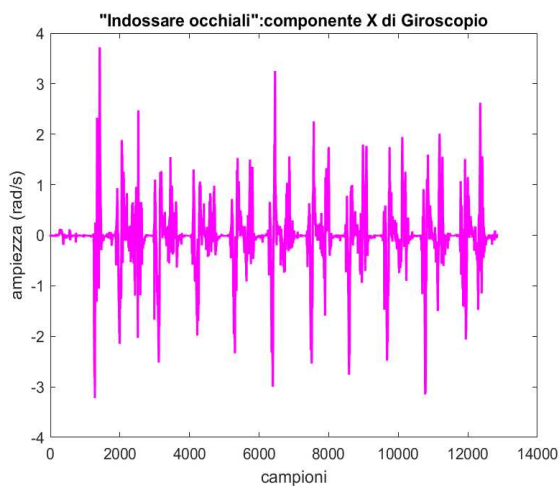


Fig.16: "Indossare occhiali". Componente X del Giroscopio

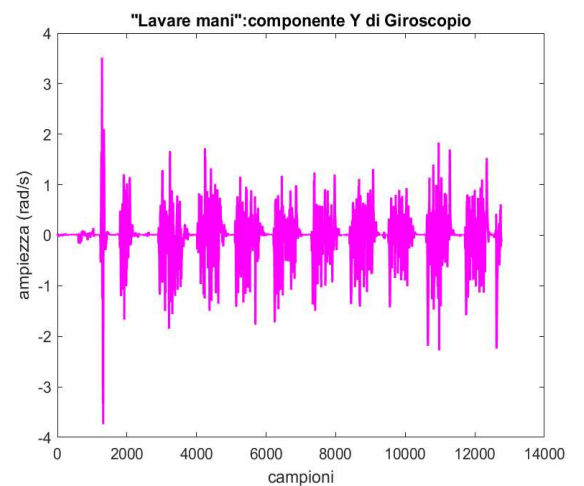


Fig.17: "Lavare mani". Componente Y del Giroscopio

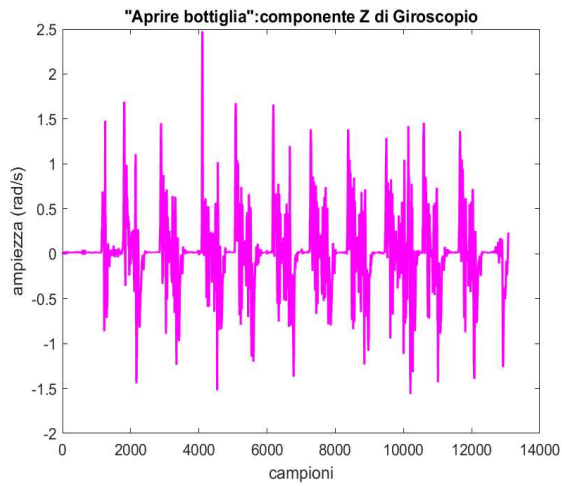


Fig.18: "Aprire bottiglia". Componente Z del Giroscopio

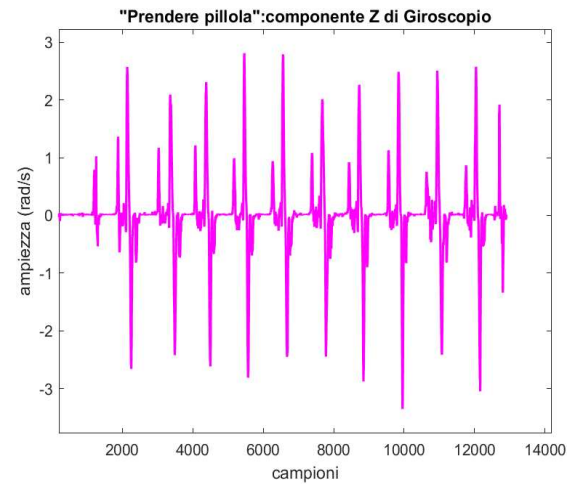


Fig.19: "Prendere pillola". Componente Z del Giroscopio

Nelle figure sopra riportate si trovano in blu i grafici relativi al segnale magnetometrico, in verde i grafici relativi al segnale accelerometrico e in rosa quelli relativi al segnale di velocità angolare (giroscopio).

2.3 Elaborazione dei dati

2.3.1 Segmentazione dei segnali

La fase successiva dello studio riguarda l'elaborazione dei dati acquisiti dai soggetti; per eseguire questa operazione è stato utilizzato il software Matlab.

Per prima cosa occorre segmentare i segnali grezzi, cioè si vanno ad indentificare le finestre di campioni che indicano gli intervalli in cui è avvenuto un movimento. Per ogni attività, la segmentazione è avvenuta a partire dai segnali provenienti dall'asse Z del giroscopio, poiché, visivamente, è risultato essere il grafico migliore per individuare i momenti di azione e di riposo. Possiamo notare infatti, in Fig.20, che negli istanti di riposo la componente Z del giroscopio risulta 0 rad/s e con un valore diverso da 0 negli istanti di azione. È stato sufficiente utilizzare un solo segnale, in quanto i due sensori IMU sono già tra loro sincronizzati. .

Gli intervalli ottenuti sono stati poi salvati in una tabella in cui si distingue precisamente l'inizio e la fine di ciascun movimento (Tab.1).

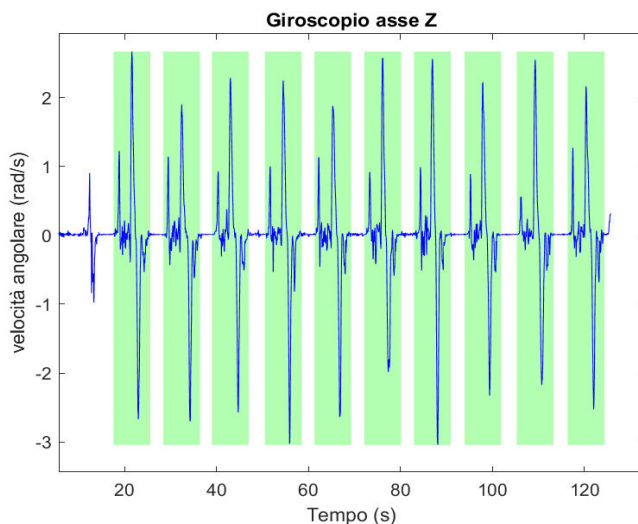


Fig.20: segmentazione del segnale relativo all'attività "prendere pillola"

	1 Inizio	2 Fine
1	1757	2557
2	2862	3662
3	3948	4748
4	5056	5856
5	6146	6946
6	7242	8042
7	8432	9232
8	9467	10267
9	10593	11393
10	11691	12491

Tabella 1: istanti di inizio e fine dei movimenti

2.3.2 Estrazione delle features

Una volta noti gli intervalli in cui sono avvenuti i movimenti relativi a ciascuna attività della vita quotidiana, si passa alla fase successiva di estrazione delle features. Le features rappresentano alcune caratteristiche dei segnali e la loro estrazione è di grande importanza poiché influisce direttamente sull'efficienza della classificazione. Inoltre, consente di manipolare le informazioni di partenza per ridurre la complessità del dataset (Mustafa, 2023). In sostanza, le features vengono utilizzate come input nei modelli di machine learning, permettendo quindi l'identificazione e la distinzione delle varie attività.

Le feature di interesse per questo studio sono sette, tutte definite nel dominio del tempo, e sono state estratte dai segnali grazie a funzioni specifiche implementate in Matlab:

- 1) Root Mean Square (RMS): estratta con la funzione “getrmsfeat”
- 2) Mean Absolute Value (MAV): estratta con la funzione “getmavfeat”
- 3) Integrated Absolute Value (IAV): estratta con la funzione “getiavfeat”
- 4) Zero Crossing (ZC): estratta con la funzione “getzcfeat”
- 5) Slope Sign Change (SSC): estratta con la funzione “getsscfeat”
- 6) Waveform Length (WL): estratta con la funzione “getwlfeat”
- 7) Autoregressive Coefficients(AR): estratta con la funzione “getarfeat”

I segnali utilizzati come input per le funzioni sono relativi agli assi X, Y, Z di Accelerometro, Giroscopio, Magnetometro, FreeAcc (accelerazione libera, cioè l'accelerazione a cui è stata tolta la componente relativa alla gravità) e il segnale relativo alla Norma dell'accelerazione.

Ogni feature è stata calcolata considerando una finestra di 20 campioni sui segnali e utilizzando uno spostamento della finestra di 10 campioni, con una conseguente sovrapposizione di 10 campioni. In questo modo ogni dieci campioni del segnale originale si è ottenuto un campione di feature.

L'output delle funzioni utilizzate è un vettore che contiene i valori di ogni feature.

Tutti i vettori relativi alle features calcolate su ciascun asse dei sensori elencati precedentemente, vengono concatenati in un'unica grande matrice che contiene dunque le caratteristiche di ogni attività eseguita.

1 – Root Mean Square (RMS)

Rappresenta la radice del valore quadratico medio del segnale. È un parametro che dà un'indicazione dell'energia complessiva del segnale stesso.

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$$

2 – Mean Absolute Value (MAV)

Rappresenta la media del valore assoluto del segnale.

$$MAV = \frac{1}{N} \sum_{i=1}^N |x_i|$$

3 – Integrated Absolute Value (IAV)

Indica la somma dei valori assoluti del segnale.

$$IAV = \sum_{i=1}^N |x_i|$$

4 – Zero Crossing (ZC)

Conta il numero di volte in cui il segnale cambia segno. È un indicatore delle oscillazioni del segnale attorno allo zero.

$$ZC = \sum_{i=1}^{N-1} \text{sgn}[(x_i \cdot x_{i+1}) < 0]$$

5 – Slope Sign Change (SSC)

Conta i cambi di segno nella derivata del segnale. È un parametro che indica il numero di cambi di pendenza.

$$SSC = \sum_{i=2}^{N-1} \text{sgn}[(x_i - x_{i-1}) \cdot (x_{i+1} - x_i) < 0]$$

6 – Waveform Length (WL)

Indica la lunghezza del segnale calcolata come somma delle differenze assolute tra campioni successivi. Fornisce un'idea della complessità del segnale.

$$WL = \sum_{i=1}^{N-1} |x_{i+1} - x_i|$$

7 – Autoregressive Coefficients (AR)

Il modello AR rappresenta la combinazione lineare delle finestre precedenti più un termine di errore. L'ordine di un modello AR indica il numero di coefficienti considerati per rappresentare il segnale (in questo caso si è scelto l'ordine 4).

Modello AR:

$$x_i = \sum_{p=1}^P a_p x_{i-p} + w_i$$

Una volta ottenute tutte le features necessarie, si va a costruire un vettore di label, a partire dagli intervalli di movimento, in cui si assegna il valore 0 alle fasi di riposo e i valori da 1 a 14 alle fasi di azione di ciascuna attività eseguita. Queste etichette

consentono di assegnare una classe di appartenenza a ciascuna attività, permettendo poi ai modelli di Machine Learning di distinguerle.

In (Tab.2) sono riportate le attività con la corrispondente label assegnato.

ATTIVITA'	LABEL
Riposo	0
Bere con tazza	1
Bere con bicchiere	2
Indossare cuffie	3
Mangiare	4
Indossare occhiali	5
Abbassare maniglia	6
Lavare mani	7
Aprire bottiglia	8
Rispondere al telefono	9
Prendere pillola	10
Versare acqua	11
Fumare	12
Scrivere	13
Sbadigliare	14

Tabella 2: Attività con label corrispondente.

2.4 Classificazione

L'ultima fase di questo progetto riguarda la classificazione dei dati ottenuti, tramite degli algoritmi di Machine Learning.

Per Machine Learning (ML) si intende un metodo di apprendimento automatico che permette ai sistemi di acquisire conoscenze dai dati forniti, affrontando situazioni non previste dalla programmazione iniziale. Questo approccio si applica a dataset complessi, di grandi dimensioni, rivelando concetti nascosti, relazioni tra attributi e schemi utili per estrarre informazioni mirate. Gli algoritmi di Machine Learning non solo elaborano i dati, ma ottimizzano i processi di analisi e decisione attraverso l'apprendimento da esperienze passate e la rilevazione di pattern significativi. Grazie a queste capacità, il ML si configura come un campo scientifico e sperimentale cruciale in numerosi ambiti, come ad esempio le scienze mediche (Rathore, Mannepalli, 2021).

Quando si utilizza il ML per riconoscere le attività di interesse, occorre innanzitutto addestrare i modelli. Nella fase di training, viene fornita una grande quantità di dati, in modo che gli algoritmi possano apprendere le caratteristiche dei dati stessi. Nella fase di testing, invece, vengono forniti dei dati con le stesse caratteristiche di quelli di addestramento, ma sconosciuti ai modelli, per verificare se sono in grado di effettuare previsioni basandosi su ciò che hanno appreso precedentemente. In questo caso è stato fornito circa il 70% dei dati a disposizione per il training e, successivamente, il restante 30% per il testing.

Poiché i valori delle feature potrebbero essere molto diversi tra loro, prima di procedere con la classificazione, le matrici di feature vengono normalizzate tramite una funzione chiamata "normalize". Grazie alla funzione viene calcolato lo "zscore" di ogni elemento dei dati, che è definito come $z = (x - \mu) / \sigma$, in cui x è il valore del dato originale, μ è la media dei dati della colonna e σ è la deviazione standard della colonna considerata. Per l'addestramento e il testing dei modelli di ML è stato utilizzato il Classification Learner Toolbox di Matlab.

Per questo studio ci si è concentrati su tre scenari principali di classificazione:

- 1) **Classificazione totale** (15 classi): si vogliono riconoscere tutte le attività della vita quotidiana in modo distinto.
- 2) **Classificazione ternaria** (3 classi): si vogliono identificare in modo specifico le attività di bere con un bicchiere e prendere una pillola, distinguendole da tutte le altre.
- 3) **Classificazione binaria** (2 classi): si vuole distinguere solo il gesto di prendere la pillola, rispetto a tutto il resto.

Per ogni scenario di classificazione, sono state effettuate numerose prove, addestrando vari modelli di ML e combinando i dati provenienti da diversi sensori. Inoltre, per questo studio, sono stati considerati i dati di un solo soggetto tra tutti quelli che hanno eseguito le acquisizioni.

In ogni modello è stato scelto lo schema di validazione cross-validation con 10 fold. La validazione stima le prestazioni del modello su nuovi dati rispetto a quelli di addestramento e aiuta a scegliere l'algoritmo di classificazione migliore per il caso specifico (Matlab Documentation).

I modelli utilizzati sono brevemente descritti di seguito.

Linear Discriminant Analysis (LDA): è un algoritmo di classificazione lineare che proietta i dati in uno spazio a minore dimensionalità, massimizzando la separazione tra le classi. Il modello LDA presuppone che ogni input sia tratto da una distribuzione gaussiana multivariata, con un vettore medio multivariato specifico per ogni classe e una matrice di covarianza comune a tutte le classi. Ogni predittore della distribuzione gaussiana multivariata dovrebbe avere una distribuzione normale unidimensionale e una certa correlazione con gli altri predittori (Wasef, Rafla, 2020).

Medium KNN (k-Nearest Neighbor): è un algoritmo di classificazione basato sulla vicinanza dei punti. Esso funziona secondo il principio della raccolta e della

memorizzazione di tutti i campioni disponibili e poi classifica un nuovo campione in base alla maggioranza dei voti dei suoi k vicini (Sharma, Chauhan, 2024). In particolare, il termine “medium” indica un valore medio del parametro k (in questo caso 10).

Bagging Tree: è un insieme di alberi decisionali utilizzato in problemi di classificazione multi-classe. Si basa sul partizionamento ricorsivo dello spazio degli input. Lo spazio specificato dai campioni di addestramento viene suddiviso in regioni, dette “terminal leaves”, poi una sequenza di test viene introdotta e applicata ai nodi decisionali, in modo da definire in quale nodo un oggetto debba essere classificato, in base all’input fornito (Elshami et al., 2018).

Bilayered Neural Network: le reti neurali sono un modello computazionale complesso, che ricorda la struttura di un cervello umano. Questo modello è composto da molti neuroni, disposti in diversi strati e interconnessi tra loro, che ricevono ed inviano informazioni (Myo et al., 2019). In particolare, con “bilayered” si intende una rete neurale con due strati nascosti.

Linear Support Vector Machine (SVM): è un algoritmo di machine learning che cerca di trovare il miglior iperpiano che separa due classi diverse nello spazio degli input (Rochim et al., 2021). In questo caso con “linear” si intende un SVM che sfrutta un iperpiano lineare per separare due classi.

Medium Neural Network: si tratta di una rete neurale che ha un numero intermedio di strati e di neuroni per ogni strato.

Fine Gaussian SVM: è un modello di SVM che sfrutta un kernel gaussiano, o RBF (Radial Basis Function). Nel kernel RBF è presente un parametro gamma che influisce sulla mappatura dei dati, cioè maggiore è gamma e maggiore è la curvatura dell’iperpiano (Rochim et al., 2021).

Subspace KNN: si tratta di un'implementazione avanzata di KNN che sfrutta il metodo di "subspace ensemble".

SVM Kernel: nei modelli SVM sono presenti diverse funzioni kernel. Il compito del kernel è quello di prendere i dati in input e convertirli nella forma desiderata (Rochim et al., 2021).

Per quanto riguarda la classificazione totale delle attività (quindi 15 classi distinte da riconoscere), sono state effettuate un totale di 6 prove sui modelli di ML elencati precedentemente. La prima prova è stata eseguita fornendo come input di addestramento una matrice composta dalle features degli assi x, y, z di Accelerometro, Giroscopio e Magnetometro, provenienti da uno solo dei due sensori IMU indossati dal soggetto, per un totale di 88 colonne di features più 1 per le labels. Per la seconda prova è stata utilizzata una matrice di 30 colonne contenente solo le features calcolate sugli assi x, y, z dell'accelerazione libera (da qui in poi indicata come FreeAcc). La terza prova è stata eseguita su una matrice contenente le features dell'accelerazione libera unite alla precedente matrice con Accelerometro, Giroscopio e Magnetometro, per un totale di 118 colonne. Nella quarta prova sono stati uniti i dati provenienti da entrambi i sensori IMU indossati sul polso, fornendo quindi come input una matrice con le stesse colonne della precedente, ma un numero di righe doppio. Per la quinta prova sono state considerate solo le feature calcolate sulla norma dell'accelerazione, dunque l'input era una matrice di 10 colonne. Infine, l'ultima prova è stata eseguita fornendo come input la matrice composta dalle feature degli assi x, y, z di Accelerometro, Giroscopio, Magnetometro, FreeAcc e norma dell'accelerazione, per un totale di 127 colonne.

Per la classificazione ternaria (3 classi da riconoscere), le prove eseguite sono tre. Nella prima prova l'input per l'addestramento dei modelli è la matrice contenente le features calcolate su Accelerometro, Giroscopio, Magnetometro e FreeAcc. Per la seconda prova sono state unite le matrici dei due sensori IMU, inferiore e superiore. Nella terza

prova, infine, si è utilizzata la matrice totale contenente le features di Accelerometro, Giroscopio, Magnetometro, FreeAcc e norma dell'accelerazione.

Per la classificazione binaria (distinzione tra 2 classi di dati), sono state effettuate due prove: la prima utilizzando la matrice di features calcolate su Accelerometro, Giroscopio, Magnetometro e FreeAcc; mentre per la seconda sono state unite alla matrice precedente anche le features calcolate sulla norma dell'accelerazione.

Una volta effettuate tutte le prove su ciascun modello di Machine Learning, si è ottenuta una percentuale di accuratezza, che viene utilizzata come parametro di riferimento per stabilire quali fra i modelli considerati sono i più accurati e i più adatti agli scopi di questo studio. L'accuratezza viene calcolata tramite la seguente formula:

$$Accuracy(\%) = \frac{TP + TN}{TP + TN + FP + FN} \cdot 100$$

dove TP, TN, FP, FN indicano rispettivamente True Positive, True Negative, False Positive e False Negative.

La tabella sottostante (Tab.3) riassume le prove eseguite per ogni scenario di classificazione.

PROVE ESEGUITE	CLASSIFICAZIONE TOTALE	CLASSIFICAZIONE TERNARIA	CLASSIFICAZIONE BINARIA
Prima prova	[Acc,Gyr,Mag]	[Acc,Gyr,Mag,FreeAcc]	[Acc,Gyr,Mag,FreeAcc]
Seconda prova	[FreeAcc]	[Acc,Gyr,Mag,FreeAcc] 2 IMU	[Acc,Gyr,Mag,FreeAcc, NormaAcc]
Terza prova	[Acc,Gyr,Mag,FreeAcc]	[Acc,Gyr,Mag,FreeAcc, NormaAcc]	
Quarta prova	[Acc,Gyr,Mag,FreeAcc] 2 IMU		
Quinta prova	[NormaAcc]		
Sesta prova	[Acc,Gyr,Mag,FreeAcc, NormaAcc]		

Tabella 3: Prove effettuate per ogni classificazione.

Acc: Accelerometro; Gyr: Giroscopio; Mag: Magnetometro; FreeAcc: Accelerazione libera; NormaAcc: Norma accelerazione

3. RISULTATI

3.1 Classificazione Totale

1) Prima Prova

La matrice fornita come input per l'addestramento è composta da Accelerometro, Giroscopio, Magnetometro provenienti da un solo sensore IMU (per comodità, in seguito, saranno indicati come Acc, Gyr, Mag rispettivamente).

MODELLI	TRAINING ACCURATEZZA	TEST ACCURATEZZA
Linear Discriminat	66,9%	67,9%
Medium KNN	72,2%	72,0%
Bagged Trees	84,4%	77,7%
Bilayered Neural Network	78,5%	74,8%
Linear SVM	75,8%	75,8%
Medium Neural Network	79,9%	74,9%
Fine Gaussian SVM	76,3%	77,7%
Subspace KNN	75,6%	73,9%
Logistic Regression Kernel	68,8%	72,0%

Tabella 4: risultati prima prova classificazione totale. Matrice con [Acc, Gyr, Mag].

2) Seconda prova

In input viene fornita la matrice contenente solo le features della FreeAcc.

MODELLI	TRAINING ACCURATEZZA	TEST ACCURATEZZA
Linear Discriminat	48,8%	49,2%
Medium KNN	43,6%	43,2%
Bagged Trees	58,0%	55,3%
Bilayered Neural Network	53,9%	52,1%
Linear SVM	50,9%	50,1%
Medium Neural Network	56,2%	52,4%
Fine Gaussian SVM	54,3%	54,8%
Subspace KNN	52,4%	55,0%
Logistic Regression Kernel	50,2%	51,3%

Tabella 5:risultati seconda prova classificazione totale. Matrice con [FreeAcc]

3) Terza prova

La matrice di features è costituita da Acc, Gyr, Mag e FreeAcc.

MODELLI	TRAINING ACCURATEZZA	TEST ACCURATEZZA
Linear Discriminat	68,3%	69,4%
Medium KNN	67,6%	62,3%
Bagged Trees	84,3%	77,9%
Bilayered Neural Network	77,7%	73,6%
Linear SVM	76,6%	75,9%
Medium Neural Network	78,8%	73,8%
Fine Gaussian SVM	77,5%	77,4%
Subspace KNN	78,4%	76,1%
Logistic Regression Kernel	70,0%	71,9%

Tabella 6: risultati terza prova classificazione totale. Matrice con [Acc,Gyr,Mag,FreeAcc]

4) Quarta prova

Sono state considerate come input le feature di Acc, Gyr, Mag, FreeAcc, provenienti da entrambi i sensori IMU.

MODELLI	TRAINING ACCURATEZZA	TEST ACCURATEZZA
Linear Discriminat	63,5%	63,6%
Medium KNN	73,2%	70,7%
Bagged Trees	83,7%	75,2%
Bilayered Neural Network	74,2%	73,6%
Linear SVM	72,6%	71,3%
Medium Neural Network	79,2%	74,6%
Fine Gaussian SVM	77,5%	75,1%
Subspace KNN	78,9%	74,0%
SVM Kernel	76,0%	74,3%

Tabella 7: risultati quarta prova classificazione totale. Matrice con [Acc,Gyr,Mag,FreeAcc] di 2 IMU

5) Quinta prova

Per la quinta prova si è utilizzata solo la matrice con le features calcolate sulla Norma dell'accelerazione (per comodità verrà indicata in seguito come NormaAcc).

MODELLI	TRAINING ACCURATEZZA	TEST ACCURATEZZA
Linear Discriminat	45,3%	48,7%
Medium KNN	46,9%	50,4%
Bagged Trees	47,7%	50,7%
Bilayered Neural Network	49,2%	52,1%
Linear SVM	46,7%	46,9%
Medium Neural Network	49,2%	52,8%
Fine Gaussian SVM	47,1%	49,7%
Subspace KNN	46,3%	48,8%
SVM Kernel	43,7%	46,1%

Tabella 8: risultati quinta prova classificazione totale. Matrice con [NormaAcc]

6) Sesta prova

La matrice di input è costituita dalle features concatenate di Acc, Gyr, Mag, FreeAcc, NormaAcc.

MODELLI	TRAINING ACCURATEZZA	TEST ACCURATEZZA
Linear Discriminat	68,4%	69,7%
Medium KNN	73,1%	72,9%
Bagged Trees	84,2%	77,7%
Bilayered Neural Network	78,1%	74,2%
Linear SVM	76,6%	75,9%
Medium Neural Network	78,8%	74,9%
Fine Gaussian SVM	43,2%	45,4%
Subspace KNN	76,7%	75,2%
SVM Kernel	76,2%	77,3%

Tabella 9: risultati sesta prova classificazione totale. Matrice con [Acc, Gyr, Mag, FreeAcc, NormaAcc]

Si può notare, dalle tabelle sopra riportate, come l'utilizzo delle feature calcolate solo sull'accelerazione libera e sulla norma dell'accelerazione (prove 2 e 5), porti a valori di accuratezza molto bassi, alcuni al di sotto del 50%. Questo significa che il numero di feature fornite ai modelli di ML non è sufficiente per distinguere in modo corretto tutte le attività.

Per questo motivo negli scenari di classificazione successivi, si è preferito effettuare le prove direttamente con matrici più complete.

3.2 Classificazione ternaria

1) Prima prova

In input viene fornita la matrice di features ottenuta concatenando Acc, Gyr, Mag e FreeAcc.

MODELLI	TRAINING ACCURATEZZA	TEST ACCURATEZZA
Linear Discriminat	76,0%	75,1%
Medium KNN	82,6%	82,0%
Bagged Trees	88,1%	86,2%
Bilayered Neural Network	85,9%	85,6%
Linear SVM	76,7%	76,2%
Medium Neural Network	85,2%	84,7%
Fine Gaussian SVM	85,9%	85,7%
Subspace KNN	84,5%	83,1%
SVM Kernel	83,1%	82,8%

Tabella 10: risultati prima prova classificazione ternaria. Matrice con [Acc, Gyr, Mag, FreeAcc]

2) Seconda prova

In questa prova la matrice di features è composta da Acc, Gyr, Mag, FreeAcc, provenienti da entrambi i sensori IMU posizionati sul polso.

MODELLI	TRAINING ACCURATEZZA	TEST ACCURATEZZA
Linear Discriminat	74,6%	74,1%
Medium KNN	82,2%	81,5%
Bagged Trees	87,7%	85,2%
Bilayered Neural Network	84,5%	83,1%
Linear SVM	74,6%	74,3%
Medium Neural Network	85,0%	84,5%
Fine Gaussian SVM	85,6%	85,2%
Subspace KNN	84,6%	83,7%
SVM Kernel	83,0%	82,6%

Tabella 11: risultati seconda prova classificazione ternaria. Matrice con [Acc, Gyr, Mag, FreeAcc] di 2 IMU

3) Terza prova

La matrice di input è costituita dalla concatenazione delle features di Acc, Gyr, Mag, FreeAcc e NormaAcc.

MODELLI	TRAINING ACCURATEZZA	TEST ACCURATEZZA
Linear Discriminat	76,2%	75,1%
Medium KNN	82,4%	81,9%
Bagged Trees	88,0%	86,1%
Bilayered Neural Network	85,9%	85,4%
Linear SVM	76,7%	76,3%
Medium Neural Network	84,8%	84,4%
Fine Gaussian SVM	85,8%	85,9%
Subspace KNN	84,5%	83,2%
SVM Kernel	83,4%	82,2%

Tabella 12: risultati terza prova classificazione ternaria. Matrice con [Acc,Gyr,Mag,FreeAcc,NormaAcc]

Si può notare dalle Tabelle 7 e 11 che in entrambi gli scenari di classificazione appena mostrati, le prove in cui sono implicati i dati provenienti dai sensori IMU inferiore e superiore, presentano dei valori di accuratezza molto simili, e in alcuni casi inferiori, a quelli che troviamo nelle prove effettuate con le stesse features ma provenienti da un solo sensore IMU (Tabelle 6 e 10). Tuttavia, il tempo impiegato per il training con la matrice dei due sensori, è risultato nettamente maggiore, pur ottenendo dei risultati simili. Dunque per questo motivo, nell'ultima classificazione si è preferito limitare l'addestramento dei modelli con le features provenienti da un solo sensore, per evitare lunghi tempi di attesa.

3.3 Classificazione binaria

1) Prima prova

È stata eseguita utilizzando la matrice ottenuta dalla concatenazione di Acc, Gyr, Mag e FreeAcc.

MODELLI	TRAINING ACCURATEZZA	TEST ACCURATEZZA
Linear Discriminat	86,5%	85,6%
Medium KNN	89,6%	89,7%
Bagged Trees	92,9%	91,9%
Bilayered Neural Network	91,2%	91,1%
Linear SVM	86,5%	85,7%
Medium Neural Network	89,4%	89,1%
Fine Gaussian SVM	91,8%	91,9%
Subspace KNN	90,9%	90,4%
SVM Kernel	90,4%	90,4%

Tabella 13: risultati prima prova classificazione binaria. Matrice con [Acc, Gyr, Mag, FreeAcc]

2) Seconda prova

L'ultima prova è stata eseguita addestrando i modelli con la matrice contenente le feature di Acc, Gyr, Mag, FreeAcc e NormaAcc.

MODELLI	TRAINING ACCURATEZZA	TEST ACCURATEZZA
Linear Discriminat	86,7%	85,8%
Medium KNN	89,6%	89,7%
Bagged Trees	92,9%	91,9%
Bilayered Neural Network	91,2%	91,5%
Linear SVM	86,7%	86,0%
Medium Neural Network	89,6%	89,1%
Fine Gaussian SVM	91,8%	92,0%
Subspace KNN	90,9%	90,5%
SVM Kernel	90,2%	90,6%

Tabella 14: risultati seconda prova classificazione binaria. Matrice con [Acc,Gyr,Mag,FreeAcc,NormaAcc]

Un modo per poter valutare visivamente l'accuratezza di un modello di Machine Learning è grazie alle Matrici di Confusione. Ogni riga della matrice rappresenta il valore effettivo della classe da prevedere e ogni colonna il valore previsto. Sulla diagonale principale si trovano i valori predetti correttamente (colore blu), mentre al di fuori della diagonale si trovano gli errori (colore rosa). Il metodo della matrice di confusione viene spesso utilizzato per valutare le prestazioni di classificazione. I True Positive (TP) e i True Negative (TN) indicano il numero di classi positive e negative identificate correttamente, mentre i False Positive (FP) e i False Negative (FN) indicano il numero di classi positive e negative identificate nel modo sbagliato (Kaur, Bath, 2024).

Di seguito sono riportate le matrici di confusione del test di alcuni modelli che sono risultati i migliori, per ciascuno degli scenari di classificazione (Fig.21-26).

Model 4

0	2203	49	11	7	14	9	61	3	27	5	15	32	8	13	2	
1	62	157	1	1		9				11						
2	39	11	130	4		7		5			4	1				
3	24		4	118	12	33			1	1	10	1				
4	53		5	14	98	13		3	14	3	14	11	2			
5	22		8	11	13	136		3	3	4	5	10	1			
6	33	5					135					2				
7	51		1		1	1		189	4	2		1	1			
8	14			1	9	2			187		6		1			
9	20					2	1	1		166			10	18		
10	53		2	5	3	15			1	4	148	5	4		3	
11	34	1	2	2	5	4			6	5	13	141	1		1	
12	28				9	1				4	4	1	156	9	3	
13	26						2			9				161		
14	29	2		2	2	2		1	1		1	1	4	1	88	
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Predicted Class

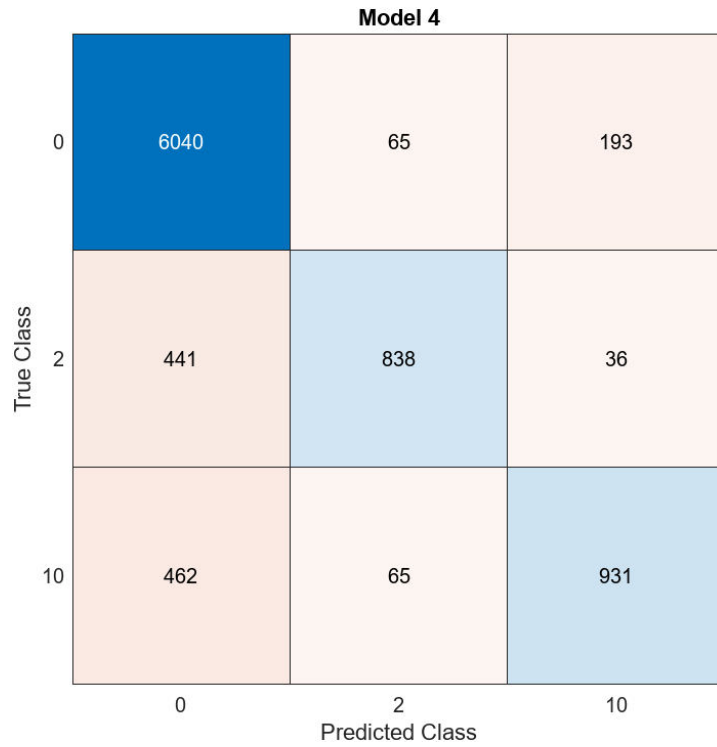
Fig.21: Matrice di confusione modello " Bagged Trees".
 Classificazione totale con [Acc, Gyr, Mag, FreeAcc, NormaAcc]

Model 10

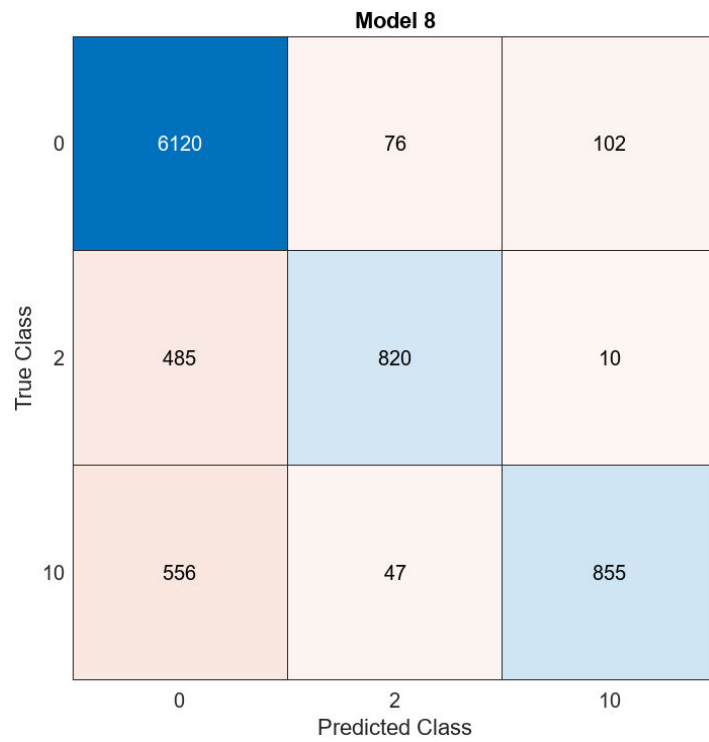
0	2357		3	7	13	5	26	23	10	2	4	1	4	4		
1	78	132	2			6				22			1			
2	49		109	2		18		6	1	7	7	2				
3	31		4	124	5	16			1	9	11	3				
4	67		7	2	85	11		8	10	6	16	10	4	4		
5	30		4	3	6	136			2	12	6	16	1			
6	42	1					116	11						5		
7	51			1		5		190			3		1			
8	29				9			5	165	4	8					
9	25					5		3		154	2	1	5	23		
10	72		5	1	4	17		1	1	5	132	2	3			
11	55			2	4	6		5	5	7	7	124				
12	38				2	1			1	2	13		148	10		
13	38							10		12			4	134		
14	29				1	1		1		5	5		7	1	84	
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Predicted Class

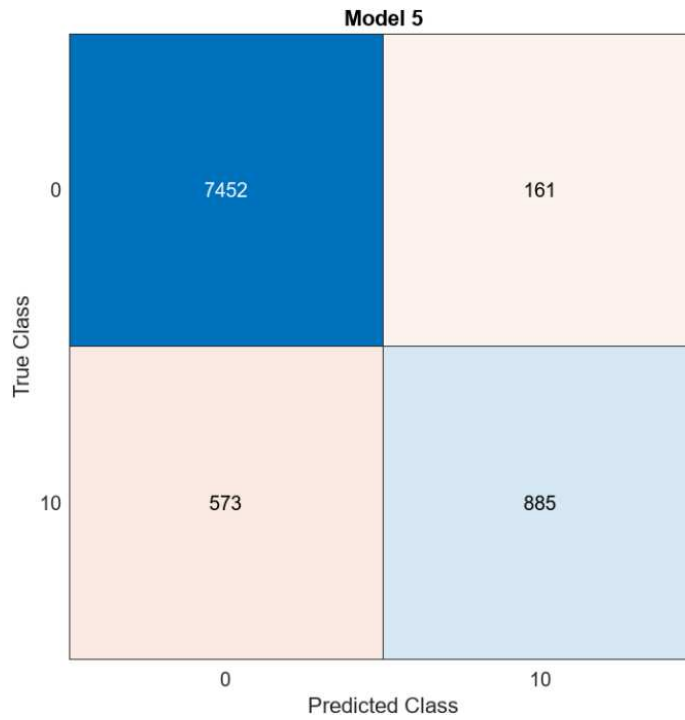
Fig.22: Matrice di confusione modello " SVM kernel".
 Classificazione totale con [Acc, Gyr, Mag, FreeAcc, NormaAcc]



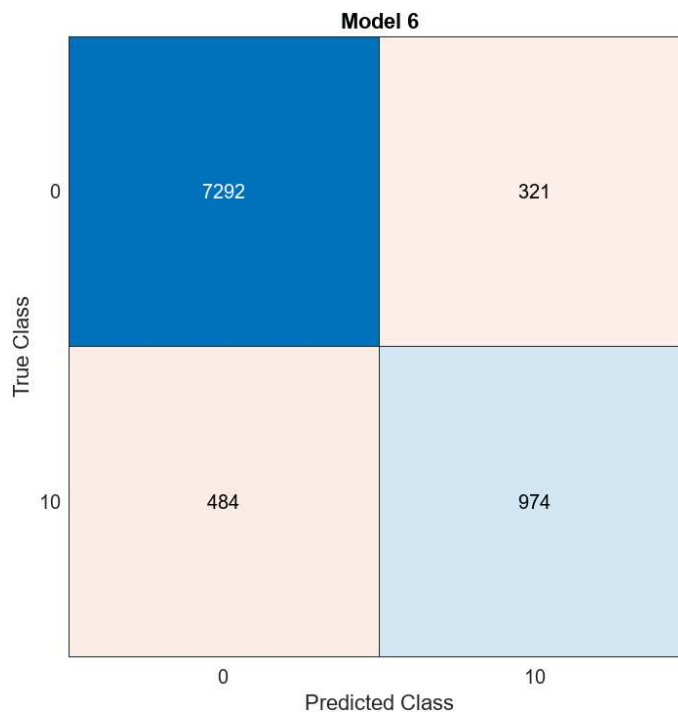
*Fig.23: Matrice di confusione modello "Bagged Trees".
Classificazione ternaria con [Acc, Gyr, Mag, FreeAcc,
NormaAcc]*



*Fig.24: Matrice di confusione modello "Fine Gaussian SVM".
Classificazione ternaria con [Acc, Gyr, Mag, FreeAcc,
NormaAcc]*



*Fig.25: Matrice di confusione modello "Bagged Trees".
Classificazione binaria con [Acc, Gyr, Mag, FreeAcc]*



*Fig.26: Matrice di confusione modello "Bilayered Neural Network".
Classificazione binaria con [Acc, Gyr, Mag, FreeAcc]*

I grafici sottostanti mostrano un confronto tra i migliori quattro modelli per ogni scenario di classificazione (Fig.27-29).

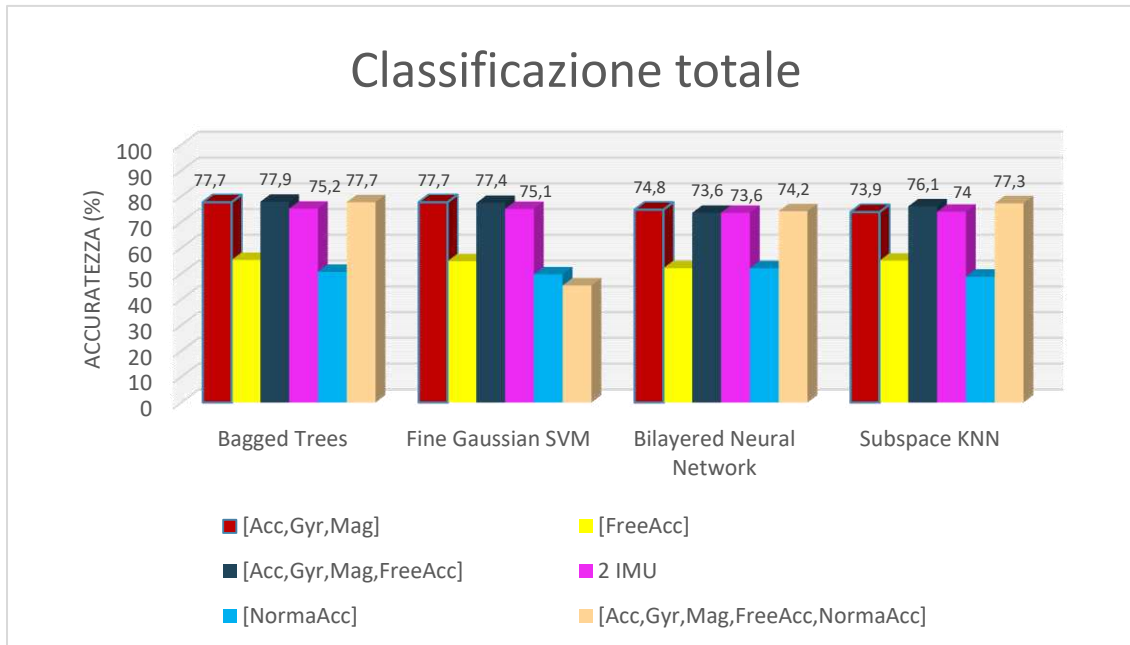


Fig.27: Confronto tra i migliori 4 modelli della classificazione totale

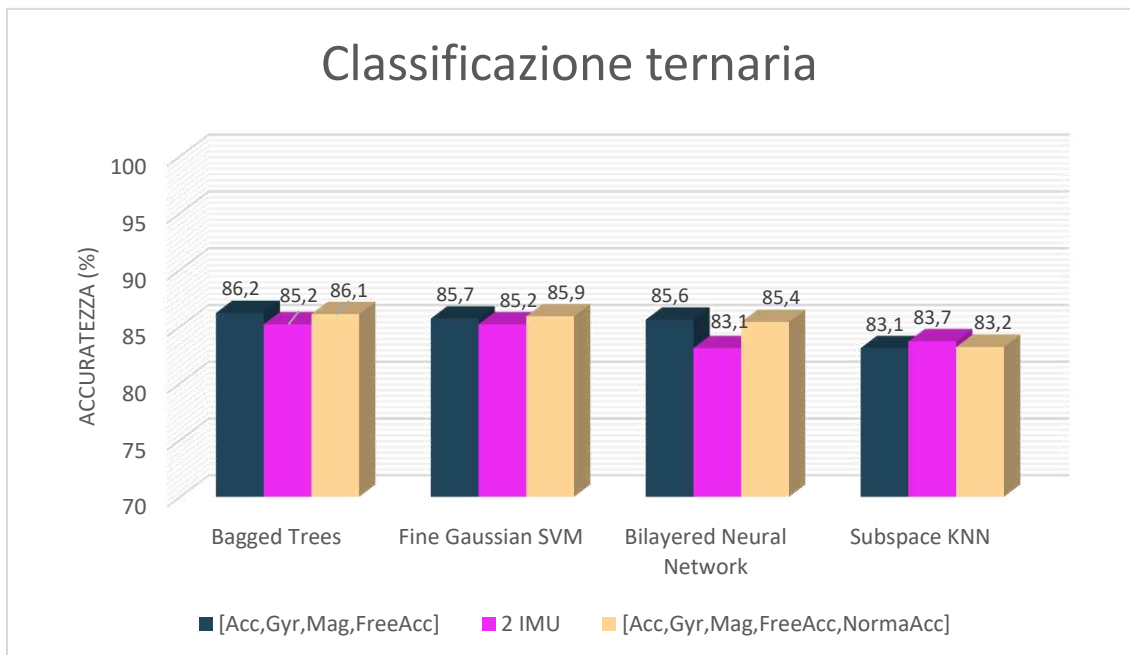


Fig.28: Confronto tra i migliori 4 modelli della classificazione ternaria

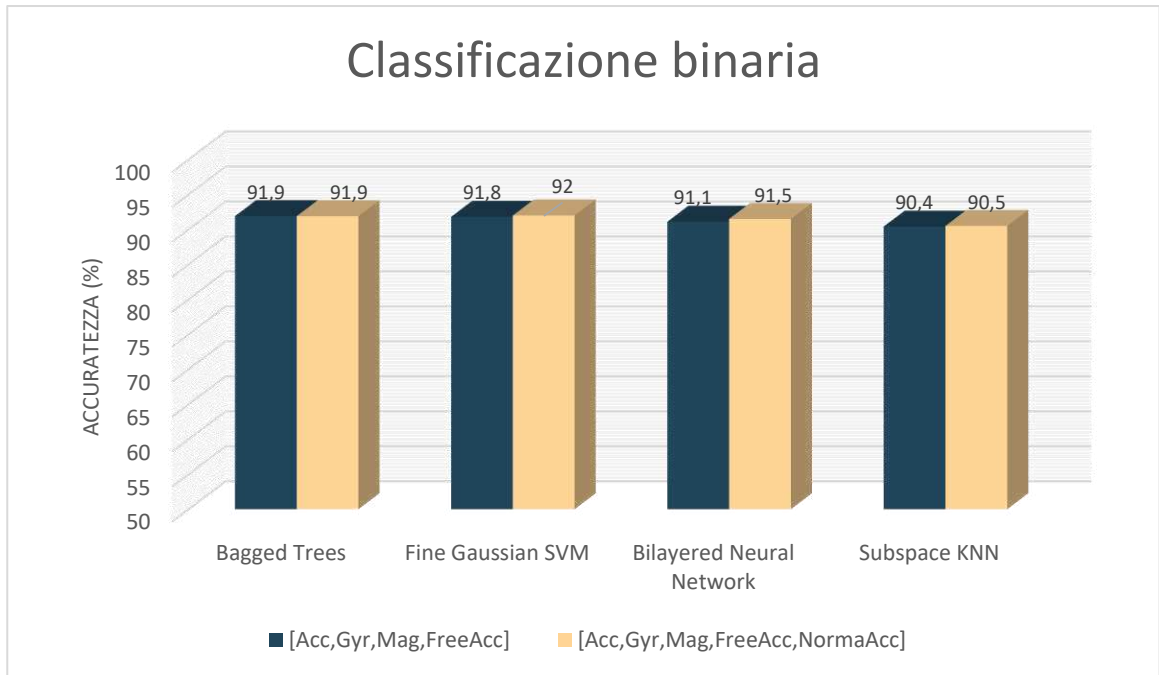


Fig.29: Confronto tra i migliori 4 modelli della classificazione binaria

4. DISCUSSIONE

I risultati ottenuti con questo progetto evidenziano come l'accuratezza della classificazione delle attività quotidiane può variare in modo significativo a seconda dei sensori considerati e dei modelli di Machine Learning impiegati per la classificazione.

In particolare, è possibile notare dai grafici in Fig. 27, 28, 29 come il modello con percentuale di accuratezza maggiore nel test rispetto a tutti gli altri, sia il Bagged Trees, che raggiunge il valore massimo di 77,9% nella classificazione multi-classe, il valore massimo di 86,2% nella classificazione ternaria e il valore massimo di 91,9% nella classificazione binaria. Con valori leggermente più bassi, ma molto simili tra loro troviamo i modelli Fine Gaussian SVM, Biletered Neural Network e Subspace KNN. Si può invece notare, ad esempio nelle Tabelle 4, 6, 11 e 13, come il modello che fornisce i risultati peggiori sia il Linear Discriminant, in tutti e tre gli scenari di classificazione.

I risultati mostrano anche come ci siano differenze importanti nell'utilizzo delle matrici con features estratte dai segnali di un solo sensore rispetto a quelle ottenute dalla concatenazione di più sensori, come nel caso delle Prove 2 e 5 della classificazione totale, effettuate rispettivamente con la sola accelerazione libera e con la sola norma dell'accelerazione. In questi casi si può vedere dalle Tabelle 5 e 8, come la percentuale di accuratezza sia intorno al 50%. Gli scarsi risultati di questo tipo di prove trovano riscontro anche nel lavoro di Scattolini et al.(2024), in cui si evidenzia come l'uso esclusivo del Giroscopio, incorporato in un sensore IMU, porti ad un'accuratezza inferiore al 70%.

Un ulteriore dato significativo riguarda le differenze tra la classificazione totale e le classificazioni ternaria e binaria. Osservando i grafici in Fig. 27, 28, 29 si nota come nel primo scenario, le percentuali di accuratezza dei migliori 4 modelli siano tutte al di sotto dell' 80%, mentre nella classificazione ternaria le percentuali si trovano tra 80% e 85% circa e in quella binaria i risultati arrivano addirittura a superare il 90% in tutti

e quattro i modelli mostrati. Questo significa che i modelli di Machine Learning impiegati sono più efficaci nel distinguere una sola o due attività contro tutto il resto. Tale risultato è coerente con lo studio effettuato da Moccia et al.(2022), in cui si evidenzia come la classificazione multi-classe, applicata ai modelli SVM, KNN e RF, porti a risultati in termini di accuratezza tra 79% e 81,5%; mentre la classificazione binaria, applicata agli stessi modelli, porti a risultati tra 93,2% e 96,3%.

I risultati ottenuti mostrano dunque una buona efficienza dei metodi proposti per il riconoscimento delle attività della vita quotidiana, ma è necessario specificare la presenza di alcune limitazioni nello studio effettuato.

Per prima cosa, le acquisizioni dei segnali sono state eseguite in laboratorio, dunque in condizioni controllate, che non rispecchiano del tutto la vita reale e, inoltre, i soggetti impiegati sono tutti giovani adulti senza particolari problemi motori e di salute. Come sottolineato anche da Scattolini et al.(2024), l'impiego di un numero maggiore di soggetti e con caratteristiche demografiche diverse potrebbe portare ad un incremento della variabilità, oltre al fatto che l'esecuzione delle acquisizioni in un ambiente non controllato renderebbe l'esperimento più realistico.

Un altro aspetto da sottolineare è l'utilizzo di modelli di Machine Learning per la classificazione. Secondo lo studio di Mustafa (2023), l'impiego di ML implica l'estrazione manuale delle features di interesse, che richiede tempo e una certa competenza dell'utente; inoltre, alcune caratteristiche potrebbero anche andare a ridurre l'accuratezza. In particolare, alcuni modelli con un gran numero di caratteristiche potrebbero soffrire di un'eccessiva volatilità. Per questo motivo nello studio di Mustafa (2023) si propone l'utilizzo di algoritmi di Deep Learning, che sono in grado di individuare autonomamente le features necessarie dai segnali grezzi, riducendo il rischio di over-fitting.

5. CONCLUSIONE

Il presente studio ha mostrato come sia possibile riconoscere delle attività della vita quotidiana, grazie all'utilizzo di sensori indossabili non invasivi e sfruttando degli algoritmi di Machine Learning per la classificazione dei movimenti.

I sensori indossati dai soggetti sono due IMU, posti sul polso, e i segnali acquisiti sono stati elaborati tramite il software Matlab che ha permesso l'estrazione delle features di interesse. Infine, sono state eseguite numerose prove, variando i sensori a disposizione, per addestrare nove modelli di ML da cui si è ottenuto un valore di accuratezza che ha permesso un confronto tra i modelli stessi.

Gli scenari di classificazione sono stati tre: totale, ternaria e binaria. Questo per osservare come rispondevano i modelli alla richiesta di riconoscere tutte le attività in modo distinto, e poi al riconoscimento di attività specifiche come bere da un bicchiere e prendere una pillola rispetto a tutte le altre.

Dallo studio emerge come il riconoscimento di una o due attività rispetto al resto delle classi, raggiunga un notevole livello di accuratezza, raggiungendo valori oltre il 90% in alcuni casi. Questo potrebbe essere un risultato interessante per delle applicazioni pratiche come il monitoraggio della salute a distanza di persone malate o anziane, per verificare ad esempio, se assumono correttamente i farmaci o rimangono idratati nell'arco della giornata.

In futuro, lo studio potrebbe essere migliorato integrando sensori più avanzati o testando i modelli su un campione maggiore di soggetti, in modo da aumentare l'affidabilità del sistema in scenari reali.

BIBLIOGRAFIA

- A. F. Rochim, K. Widyaningrum and D. Eridani, "Performance Comparison of Support Vector Machine Kernel Functions in Classifying COVID-19 Sentiment" (2021).
- A. R. Elshami, A. Youssef and M. W. Fakh, "Multi-step Ahead Time Series Prediction via Bagging Trees Based Neighborhood," (2018).
- D. K. Rathore, P. K. Mannepalli, "A Review of Machine Learning Techniques and Applications for Health Care", (2021).
- D.Sharma, U. Chauhan, "Human Activity Prediction Studies Using Wearable Sensors and Machine Learning" (2024).
- H. Kalantarian, N. Alshurafa, M. Sarrafzadeh, "Detection of Gestures Associated with Medication Adherence Using Smartwatch-Based Inertial Sensors" (2016).
- K. Kaur, J. Batth, "Human Activity Recognition using Machine Learning" (2024).
- M. R. Wasef, N. Rafla, "HLS Implementation of Linear Discriminant Analysis Classifier" (2020).
- M. Scattolini, A. Tigrini, F. Verdini, G. Iadarola, S. Spinsante, S. Fioretti, L. Burattini, A. Mengarelli, "Leveraging Inertial Information From a Single IMU for Human Daily Activity Recognition" (2024).
- Matlab Documentation
- Monique Paulich, Martin Schepers, Nina Rudigkeit, and Giovanni Bellusci. "Xsens MTw Awinda: Miniature Wireless Inertial-Magnetic Motion Tracker for Highly Accurate 3D Kinematic Applications".
- R. K. Athota, D. Sumathi, "Human Activity Recognition based on hybrid learning algorithm for wearable sensor data" (2022).

- S. Gupta, “Deep Learning based human activity recognition (HAR) using wearable sensor data” (2021).
- S. Moccia, S. Solbiati, M. Khornegam, F. Bossi, E. Caiani, “Automated classification of hand gestures using a wristband and machine learning for possible application in pill intake monitoring” (2022).
- W. W. Myo, W. Wettayaprasit and P. Aiyarak, "Designing Classifier For Human Activity Recognition Using Artificial Neural Network" (2019).
- Z. Mustafa, “A study of Machine Learning techniques based on Human Daily Living activities via Inertial Sensors” (2023).