



# UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

SISTEMI BIOMETRICI\*\*\*

MOVING\*\*\*

(HAR su App)

*Docenti: Prof. Donato Impedovo*

*Gruppo: Belforte Matteo, matricola 760396*

*Mantini Francesca, matricola 757122*

# Indice

1.	Introduzione.....	3
2.	Stato dell'arte.....	4
	Sensori e attributi collezionabili .....	4
	Architettura dei sistemi HAR .....	4
	Tecnica delle sliding window .....	5
	Algoritmi di classificazione.....	5
3.	Dataset .....	6
4.	Metodi .....	6
	Pre-Processing .....	6
	Feature Extraction .....	7
	Tecniche per la gestione di dati sbilanciati.....	8
	Tecniche per la Feature Selection.....	9
5.	Sperimentazione .....	9
	5.1. Esperimento 1 .....	10
	____ Selezione dei modelli di classificazione .....	10
	____ Leave One Subject Out Cross Validation.....	10
	____ Metriche di valutazione .....	10
	____ Implementazione .....	11
	5.2. Esperimento 2.....	12
	____ Selezione e caratteristiche della piattaforma di hosting .....	13
	____ Implementazione della web app .....	13
	____ Funzionamento e interfaccia di MovIng.....	15
	____ Metodologia e protocollo dei test con gli utenti .....	18
6.	Risultati .....	18
	6.1. Risultati 1 - Test dei modelli di classificazione.....	18
	____ Scelta del Modello.....	18
	6.2. Risultati 2 - Test della web app .....	20
	____ Risultati dei test comparativi sui modelli .....	20
	____ Valutazione dell'Efficacia e Usabilità della Web App .....	21
7.	Conclusioni.....	23
	Sviluppi futuri.....	23
	Riferimenti.....	24

# 1. Introduzione

Ad oggi l'informatica e le tecnologie moderne offrono opportunità senza precedenti per migliorare la sicurezza e il benessere degli individui, andando ben oltre la concezione tradizionale e semplicistica di tale disciplina. Molteplici innovazioni stanno infatti rivoluzionando il modo di comprendere e interagire con il mondo circostante. L'informatica ha subito una rapida evoluzione, capillarizzandosi in qualsiasi aspetto della nostra vita, dal mero supporto in ambito professionale fino al quotidiano, plasmando le interazioni sociali e le abitudini individuali.

Un campo di studio particolarmente interessante è quello dell'Interaction Dynamics (ID), ovvero lo studio dei modelli e dei processi che emergono dalle interazioni tra individui, oggetti e ambienti. Questo campo interdisciplinare combina elementi di informatica, psicologia, ergonomia e ingegneria per creare sistemi più intuitivi e reattivi alle esigenze umane e per migliorare le relazioni e le reti sociali.

Diramazione particolarmente interessante dell'ID è la Human Activity Recognition (HAR), che ricopre un ruolo fondamentale nell'identificazione e nell'analisi delle attività svolte da un individuo, utilizzando sensori e algoritmi di Machine Learning e sfruttando dati raccolti da dispositivi come smartphone e smartwatch. Di fatti, questa tecnologia cerca di determinare approcci possibili per discriminare le azioni umane basandosi sui dati raccolti da vari tipi di sensori, sfida non indifferente in quanto tali dati possono essere influenzati da fattori ambientali o dal posizionamento del dispositivo. Inoltre, la generalizzazione delle attività risulta molto ardua, dato che persone diverse eseguono la stessa azione in modi diversi.

Le applicazioni della HAR spaziano dal monitoraggio della salute dell'individuo, attraverso le attività fisiche o la rilevazione di anomalie di comportamento, all'implementazione di smart home, migliorando l'interazione con ambienti intelligenti, fino alla sorveglianza e sicurezza, riconoscendo comportamenti sospetti o pericolosi. Obiettivo comune, dunque, è quello di migliorare la qualità della vita degli individui attraverso una comprensione dettagliata delle attività quotidiane degli stessi.

Il progetto “MovIng” di seguito presentato si inserisce proprio questo contesto. L'obiettivo principale è sviluppare una Web App in grado di riconoscere in real-time i movimenti di una persona (quali camminata, corsa, caduta, seduta e salto), ponendo le basi per lo sviluppo di potenziali applicazioni avanzate per la sicurezza personale, che si occupino, ad esempio, dell'assistenza agli anziani o della prevenzione di incidenti in vari contesti.

A tal fine, si procederà a testare e valutare le prestazioni di diversi modelli di classificazione, utilizzando approcci sia di Deep Learning che di Shallow Learning nell'ambito dell'apprendimento supervisionato.

Nell'ambito del Deep Learning, si esamineranno le *Convolutional Neural Network* (CNN) e le *Bi-Directional Long Short-Term Memory Neural Network* (BiLSTM). Per quanto riguarda lo Shallow Learning, invece, si valuteranno le *Support Vector Machine* (SVM) e le *Random Forest* (RF).

L'obiettivo di questa analisi comparativa è identificare il modello che offre le migliori prestazioni di classificazione. Una volta individuato il modello più efficiente, si procederà alla sua implementazione all'interno dell'applicazione designata, al fine di ottenere un sistema robusto e accurato per il riconoscimento delle attività umane basato sui dati restituiti da uno dei sensori disponibili, ivi l'accelerometro triassiale.

## 2. Stato dell'arte

L'Human Activity Recognition è un campo di ricerca che si pone come obiettivo quello di riconoscere le attività umane a partire da dati di varia natura, quali possono essere sensori o immagini. Le possibili applicazioni di algoritmi di HAR sono le più disparate, ma tra queste spicca sicuramente la sicurezza. Comprendere quindi che tipo di azione è stata svolta da una persona può permettere, qualora l'azione sia pericolosa, come ad esempio una caduta o una scivolata, di attivare i soccorsi in tempi rapidi. Si presentano di seguito le principali ricerche allo stato dell'arte che sono state effettuate.

### Sensori e attributi collezionabili

Nell'ambito del riconoscimento delle attività umane (Human Activity Recognition, HAR), lo stato dell'arte presenta una varietà di metodologie, distinguibili principalmente per la natura e la provenienza dei dati utilizzati. Lo studio (1) identifica due approcci principali:

- Approcci Vision Based: basati sull'analisi di immagini e sequenze video, tipicamente acquisite da sistemi di videosorveglianza, per l'identificazione delle attività umane.
- Approcci Sensor Based: fondati sull'elaborazione di dati provenienti da sensori, sia indossabili (come accelerometri e giroscopi integrati in dispositivi mobili) che ambientali (come rilevatori di movimento distribuiti nell'area di interesse).

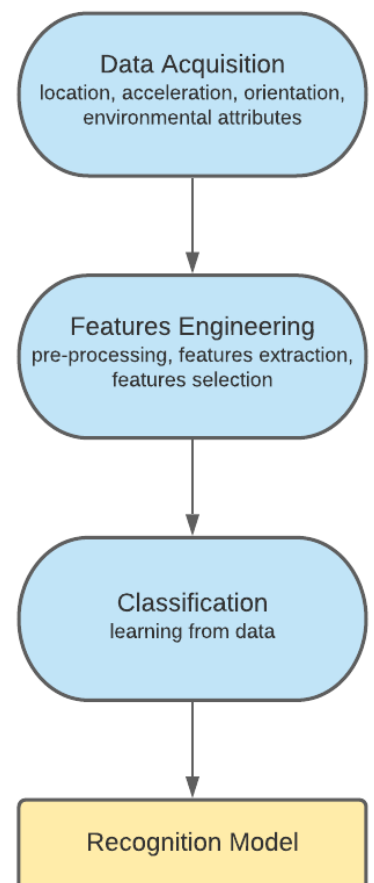
L'accelerometro misura l'accelerazione lungo tre assi: l'asse y, che corrisponde al movimento in avanti, l'asse x, che rappresenta il movimento orizzontale, e l'asse z, che cattura il movimento verticale. Questa tridimensionalità permette di ottenere una rappresentazione completa e dettagliata dei movimenti dell'utente, consentendo all'applicazione di distinguere tra diverse attività come camminare, correre, saltare, sedersi o cadere.

### Architettura dei sistemi HAR

I sistemi di riconoscimento delle attività umane (HAR) seguono un'architettura standard, comunemente adottata nella maggior parte degli studi avanzati nel settore. Questa struttura rappresenta il framework metodologico standard per la ricerca e l'implementazione di sistemi HAR, fornendo una base solida per l'analisi e l'interpretazione delle attività umane attraverso dati sensoriali o visivi.

Questa architettura comprende i seguenti passaggi:

- Raccolta dei dati: la raccolta avviene a intervalli predefiniti, utilizzando fonti selezionate quali immagini, video o sensori.
- Pre-elaborazione: i dati acquisiti vengono elaborati e normalizzati per migliorarne l'utilizzo e le prestazioni. Nel contesto HAR, diverse tecniche di pre-elaborazione vengono impiegate per il trattamento dei dati prodotti dai sensori, tra cui la tecnica della Sliding Window.
- Estrazione delle caratteristiche: alcuni studi utilizzano direttamente i dati grezzi dei sensori per gli esperimenti (vedasi (2) e (3)). Altri, invece, estraggono caratteristiche dai dati grezzi. Tra le caratteristiche spesso estratte vi sono la Signal Vector Magnitude, che rappresenta la lunghezza del vettore di accelerazione:  $a =$



$\sqrt{a_x^2 + a_y^2 + a_z^2}$ , come riportato in (4) e (5) e, parametri statistici quali medie, massimi e minimi.

- Selezione delle caratteristiche: fase che mira a identificare il sottoinsieme di caratteristiche più rilevanti per la classificazione. Tecniche comuni includono la Linear Discriminant Analysis (6) e la Principal Component Analysis (7), quest'ultima utilizzata per ridurre la dimensionalità del dataset.
- Classificazione: diversi modelli di classificazione sono utilizzati negli studi HAR. I *modelli di deep learning* più usati sono le Convolutional Neural Network, come evidenziato in (3), (8), (9), e le Long Short-Term Memory Neural Network, come in (3) e (10). Tra i *classificatori tradizionali*, si distinguono le Support Vector Machine, il k-Nearest Neighbor (11) e le Random Forest (12).

## Tecnica delle sliding window

Le Sliding Window rappresentano una tecnica fondamentale utilizzata nella fase di pre-elaborazione dei dati per il riconoscimento delle attività umane (HAR). Questa tecnica consiste nel suddividere una sequenza di dati continua in finestre temporali sovrapposte, ciascuna contenente una porzione dei dati originali.

Il processo prevede la definizione di una dimensione fissa per la finestra temporale e di un passo (stride o step), che determina di quanto la finestra si sposta in avanti ad ogni iterazione, da cui il termine *sliding*. Tipicamente, la dimensione dello step è inferiore a quella della window, motivo per cui vi è una sovrapposizione parziale dei dati tra finestre consecutive. La percentuale di sovrapposizione viene definita *overlap*.

Ogni finestra estratta rappresenta un segmento del flusso continuo di dati, permettendo di analizzare porzioni specifiche di dati temporali con maggiore dettaglio.

Le Sliding Window sono particolarmente utili per segmentare i dati dei sensori in intervalli che possono essere analizzati separatamente per rilevare attività specifiche, fattore fondamentale in HAR, dove le attività umane possono essere complesse e variare nel tempo, per ridurre l'impatto del rumore nei dati grezzi, poiché permette di lavorare con segmenti di dati più gestibili e omogenei e per estrarre caratteristiche specifiche da ciascun segmento, migliorando l'accuratezza dei modelli di riconoscimento. Infine, in applicazioni real-time, le Sliding Window permettono di aggiornare continuamente le analisi man mano che nuovi dati vengono raccolti, mantenendo il sistema sempre attivo e pronto a riconoscere nuove attività.

## Algoritmi di classificazione

Numerosi studi nel campo dell'Human Activity Recognition hanno analizzato l'effetto dell'estrazione di feature dai dati grezzi degli accelerometri triassiali sulle performance dei classificatori.

Lo studio (13) si propone di confrontare diversi approcci, utilizzando sia dati grezzi che feature estratte. In questo studio è stata impiegata la tecnica *Leave One Subject Out Cross Validation* e tra i dataset analizzati figura il dataset Unimib SHAR. Sono stati condotti vari test utilizzando solo i dati grezzi e integrandoli con feature come la *Signal Vector Magnitude* o con tecniche come la *Kernel Principal Component Analysis* e la *Sparse Principal Component Analysis*. Il miglior risultato ottenuto con il dataset Unimib SHAR è un'accuratezza del 75,65%. Inoltre, si è dimostrato che l'uso della *Signal Vector Magnitude* peggiora significativamente le prestazioni di classificazione.

Un altro studio (14) si focalizza sull'uso esclusivo dei dati grezzi dell'accelerometro triassiale, impiegando classificatori di tipo shallow learning come *Decision Tree*, *Random Forest* e *k-Nearest Neighbor*. Questo approccio ha portato a risultati molto alti, raggiungendo un'accuratezza del 98%.

Un diverso studio (2) esplora invece l'uso di classificatori deep learning, addestrando una *Convolutional Neural Network* (CNN) e una *Deep Convolutional Long Short-Term Memory Neural Network* (DCLSTM) sui dati grezzi senza una fase preliminare di feature extraction. Gli esperimenti, condotti su diverse attività, hanno mostrato che le CNN raggiungono ottime performance (accuracy e f1-score circa 96,5%), mentre i risultati con DCLSTM sono inferiori al 90%.

### 3. Dataset

Il dataset che si è utilizzato nel presente caso di studio è *ISLab*, sviluppato presso l'Università di Bari con l'obiettivo di supportare la ricerca nel campo dell'HAR. Questo dataset si concentra sulla raccolta di dati di movimento attraverso l'utilizzo dell'accelerometro triassiale e raccoglie una vasta gamma di movimenti. Si presentano schematicamente le specifiche di tale dataset:

- Partecipanti: 19 individui di cui 13 uomini e 6 donne.
- Strumentazione: utilizzo dell'accelerometro triassiale, sensore integrato nei dispositivi mobili, con frequenza di campionamento di 200 Hz.
- Protocollo di acquisizione: ogni partecipante ha svolto 2-3 prove per attività della durata di 15 secondi l'una, tenendo il dispositivo mobile nella tasca destra dei pantaloni. L'uso consistente dello stesso posizionamento favorisce la standardizzazione, poiché fornisce un punto di riferimento uniforme per l'analisi dei dati.
- Attività registrate: camminare, correre, saltare, sedersi e cadere, di cui l'ultima specializzata in avanti, all'indietro e laterale (destra e sinistra). L'inclusione di dati su vari tipi di cadute rende questo dataset particolarmente utile per lo sviluppo di sistemi di rilevamento delle cadute.
- Informazioni Statistiche: il numero totale di prove è compreso tra 304 e 456 (considerando 2-3 prove per 8 attività per 19 partecipanti), con durata totale stimata delle registrazioni tra 76 e 114 minuti. Il numero di campioni per prova è circa 3000 (considerando 15 secondi a 200 Hz)

Per il caso di studio presentato, dunque, si è scelto di utilizzare il dataset ISLab, poiché offre una varietà più ampia di attività svolte dagli utenti, fattore che consente di testare la Web app implementata in modo più completo e accurato, coprendo diverse tipologie di movimenti e situazioni.

### 4. Metodi

#### Pre-Processing

Il processo di analisi è stato avviato con una fase di preelaborazione del dataset importato, al fine di garantire l'integrità e la pertinenza dei dati. Questa fase ha comportato prima di tutto una pulizia dei dati superflui, al fine di rimuovere i valori nulli, cui è seguita una selezione mirata delle classi. Si è deciso, ad esempio, di focalizzarsi su una classificazione che non distinguesse tanto la direzione della caduta, quanto la caduta in sé, motivo per cui sono state raggruppate le istanze originariamente etichettate come "RightFall", "LeftFall", "ForwardFall" e "BackwardFall" nella classe "Fall".

Inoltre, il dataset è stato suddiviso in sliding windows, associate alle prove condotte dai rispettivi utenti, con tre diverse configurazioni per gli esperimenti (finestra di 400, di 600 e di 800 campioni) e con step fisso a 100 campioni.

## Feature Extraction

Si è scelto di non utilizzare esclusivamente i dati grezzi, ma di effettuare una fase di feature extraction per poter estrarre, a partire dagli stessi, altri feature che possano essere discriminanti ai fini della classificazione.

Il dataset di partenza è costituito da tre attributi che rappresentano i valori dell'accelerometro triassiale, uno per ogni asse  $a_x$ ,  $a_y$ ,  $a_z$  e da un attributo che rappresenta il timestamp  $t$ .

In prima istanza si sono estratte delle feature derivate per ogni record che hanno come scopo quello di incrementare la dimensione della matrice di feature che sarà fornita in input a modelli di deep learning, tra queste:

- Componenti delle *velocità*: da studi precedenti, è emerso che diversi modelli tendono a classificare erroneamente azioni simili che differiscono principalmente per la velocità di esecuzione. Ad esempio, le azioni di sedersi e di cadere all'indietro possono sembrare simili, ma la seconda avviene a una velocità maggiore rispetto alla prima. Un ragionamento analogo si applica alle azioni di camminare e correre. Per questo motivo, si è ritenuto opportuno estrarre la velocità dai dati dell'accelerometro. Utilizzando le leggi della fisica, un moto vario che avviene in un intervallo di tempo molto breve, come quello tra un record e l'altro, può essere approssimato a un moto uniforme. Pertanto, per la direzione  $x$  e analogamente per le direzioni  $y$  e  $z$ , la velocità relativa al record  $i$  è stata estratta nel modo seguente:

$$v_x(i) = \begin{cases} 0, & \text{se } i = 0 \\ v_x(i-1) + a_x(i) \cdot (t(i) - t(i-1)), & \text{se } i > 0 \end{cases}$$

- *Quadrati* delle componenti dell'accelerometro: dallo studio [15] è emerso che l'utilizzo dei quadrati delle componenti dell'accelerometro migliora le performance di classificazione. Per l'asse  $x$ , la componente quadratica è calcolata come:

$$a_x^2(i) = (a_x(i))(a_x(i))$$

Analogamente, i quadrati delle componenti dell'accelerometro sono calcolati per le altre direzioni ( $y$  e  $z$ ). Inoltre, per completezza, è stato calcolato, ma non utilizzato, il Signal Vector Magnitude (SVM), derivato dai quadrati delle componenti dell'accelerometro:

$$SVM(i) = \sqrt{(a_x^2(i) + a_y^2(i) + a_z^2(i))}$$

- *Quadrati* delle componenti delle velocità: per l'asse  $x$ ,  $v_x^2(i) = (v_x(i))^2$ . Analogamente per  $y$  e  $z$ .

Successivamente, sono state estratte delle feature statistiche aggregate a partire sia dalle feature di base (dati grezzi) che dalle feature derivate. Queste costituiranno un vettore di feature utilizzato come input per modelli di shallow learning:

- Massimo di ogni feature (utilizzato nel processo di scaling dei dati)
- Minimo di ogni feature (utilizzato nel processo di scaling dei dati)
- Media di ogni feature
- Potenza
- Deviazione standard di ogni feature

- Covarianza per ogni coppia di assi dell'accelerometro

## Tecniche per la gestione di dati sbilanciati

Nel contesto dell'apprendimento supervisionato, la qualità e la distribuzione dei dati di addestramento rivestono un ruolo cruciale per l'efficacia dei modelli di classificazione. Un problema ricorrente in questo ambito è lo sbilanciamento delle classi nei dataset, condizione per cui alcune categorie, dette minoritarie, sono rappresentate in misura significativamente inferiore rispetto ad altre, dette maggioritarie. Si parla in questo caso di sbilanciamento dei dati, fenomeno che, come analizzato in [17], può avere natura intrinseca, derivante dal dominio in esame, o estrinseca, causata da fattori esterni al problema stesso.

L'impatto di un dataset sbilanciato sull'apprendimento automatico può portare alla generazione di modelli di classificazione che favoriscono le classi maggioritarie, compromettendo l'accuratezza complessiva del sistema. Per affrontare questa problematica sono state sviluppate diverse strategie di bilanciamento del training set, tra cui i metodi di sampling, di cui si individuano due categorie principali: *Undersampling* e *Oversampling*

La prima è una tecnica che riduce il numero di esempi della classe maggioritaria per equipararli a quelli delle classi minoritarie. L'approccio più semplice, noto come *Random Undersampling*, prevede l'eliminazione casuale di esempi dalle classi sovrarappresentate. Sebbene questa tecnica possa risultare efficace in determinate situazioni, comporta inevitabilmente una perdita di informazioni potenzialmente rilevanti. Per far fronte a questa problematica, sono state sviluppate tecniche di *Informed Undersampling*, che adottano criteri più sofisticati per la selezione degli esempi da rimuovere, cercando di preservare il massimo contenuto informativo possibile.

La seconda, invece, aumenta la rappresentazione delle classi minoritarie, generalmente attraverso la duplicazione di esempi esistenti. Il *Random Oversampling* rappresenta l'approccio più elementare, selezionando casualmente esempi da replicare.

Una tecnica particolarmente innovativa nell'ambito dell'oversampling è rappresentata da *SMOTE* (Synthetic Minority Oversampling Technique). Questo approccio si distingue per la sua capacità di generare nuovi esempi sintetici per le classi minoritarie, basandosi sulle relazioni di similarità tra gli esempi esistenti. Il processo SMOTE consiste in:

- Individuazione dei vicini, in cui per ogni esempio  $\mathbf{x}_i$  della classe minoritaria, vengono identificati i  $k$  esempi più simili a  $\mathbf{x}_i$ , utilizzando l'algoritmo *k-Nearest Neighbors* e la distanza euclidea come misura di similarità. La distanza euclidea è la misura della lunghezza del segmento avente per estremi due punti A e B:  $d(A, B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$
- Generazione di esempi sintetici: dato  $\mathbf{y}_i$  scelto casualmente tra i  $k$  esempi più simili a  $\mathbf{x}_i$  e scelto casualmente un numero  $\delta$  nell'intervallo  $[0,1]$ , il nuovo esempio sintetico è costruito come  $\mathbf{x}_{new} = \mathbf{x}_i + \delta(\mathbf{y}_i - \mathbf{x}_i)$  e si trova quindi lungo il segmento che congiunge  $\mathbf{x}_i$  e  $\mathbf{y}_i$ .
- Iterazione del processo: questi passi vengono ripetuti fino a ottenere un dataset più bilanciato, con un numero maggiore di esempi sintetici per la classe minoritaria.

L'approccio SMOTE offre il vantaggio di incrementare la diversità del dataset, riducendo al contempo il rischio di overfitting, in quanto un esempio viene duplicato solo se  $\delta=0$  oppure  $\delta=1$ . Questo contribuisce a migliorare la capacità del modello di generalizzare e identificare pattern significativi, anche in presenza di classi inizialmente sottorappresentate.



## Tecniche per la Feature Selection

Fondamentale è anche il ruolo delle tecniche di feature selection, che mirano a identificare e selezionare gli attributi più rilevanti per il processo di classificazione. Questo processo porta a una riduzione della dimensione del dataset utilizzato per l'addestramento e il test dei modelli, ottimizzando così l'efficienza computazionale e migliorando le prestazioni del classificatore.

Esistono due principali categorie di tecniche di feature selection: filter e wrapper. Le tecniche filter vengono applicate preliminarmente all'addestramento del modello e sono indipendenti dal classificatore scelto. Queste si basano su metriche specifiche, come l'Information Gain, per valutare la significatività degli attributi e selezionare quelli più rilevanti. I metodi wrapper, invece, operano in sinergia con il classificatore scelto, valutando l'importanza degli attributi attraverso l'addestramento e la validazione iterativa del modello. Sebbene impieghino più tempo di calcolo, presentano una capacità di selezione più mirata e potenzialmente più efficace degli attributi.

Nel contesto specifico del riconoscimento di azioni basato su dati provenienti dall'accelerometro, le tecniche di feature selection di tipo filter sono ampiamente utilizzate. Un esempio è l'applicazione in [18] della Linear Discriminant Analysis. Oltre alle tecniche di feature selection tradizionali, esistono altri approcci per la riduzione della dimensione dei dati. Tra questi, la Principal Component Analysis (PCA) si distingue in [19] come una metodologia potente e versatile. Sebbene non sia strettamente classificabile come una tecnica di feature selection, la PCA offre un metodo efficace per comprimere l'informazione contenuta nel dataset, preservando al contempo la maggior parte della varianza dei dati originali.

Negli esperimenti effettuati non sono state utilizzate implementazioni esplicite di tecniche di feature selection, si è invece preferita una forma molto standard di feature selection, dove le feature sono state scelte manualmente basandosi sulla conoscenza del dominio (`column_names = ['x_axis', 'y_axis', 'z_axis']`). Inoltre, si è utilizzato un approccio wrapper implicito, in quanto il modello Random Forest utilizzato ha una capacità intrinseca di valutare l'importanza delle feature, le reti neurali (CNN e BiLSTM) hanno la capacità di apprendere rappresentazioni rilevanti dei dati, e, infine, i pesi del modello SVM con kernel lineare assegnano l'importanza relativa delle feature.

## 5. Sperimentazione

Si presentano di seguito i due esperimenti effettuati con i loro rispettivi obiettivi.

Esperimento 1: testare e valutare le prestazioni sul dataset ISLab di diversi modelli di classificazione, utilizzando approcci sia di Deep Learning che di Shallow Learning nell'ambito dell'apprendimento supervisionato. L'obiettivo di questa analisi comparativa è identificare il modello che offre le migliori prestazioni di classificazione, per poter poi utilizzarlo nell'esperimento successivo.

Esperimento 2: creare una web app che, sfruttando un modello pre-addestrato, sia in grado di riconoscere in tempo reale l'azione svolta dall'utente. Obiettivo dell'esperimento, dunque, è quello di partire dai dati dell'accelerometro degli smartphone per riuscire a classificare il tipo di azione compiuta in real-time.

## 5.1. Esperimento 1

### Selezione dei modelli di classificazione

Primo passo, come detto precedentemente, è stato quello di testare dei classificatori allo Stato dell'Arte. Nell'ambito del Deep Learning, si è optato per due architetture di reti neurali, la Convolutional Neural Network e la Bi-directional Long Short Term Memory Neural Network

Le CNN si distinguono per i loro livelli di convoluzione, che utilizzano filtri per estrarre caratteristiche aggiuntive dai dati, rendendole particolarmente efficaci in domini come l'elaborazione di immagini. Le BiLSTM, d'altra parte, sono progettate per processare sequenze di dati, grazie ai loro neuroni interconnessi e ai blocchi di memoria che consentono di considerare il contesto temporale. La natura bidirezionale di queste reti permette un'analisi dei dati in entrambe le direzioni temporali, offrendo una comprensione più completa delle sequenze.

Per quanto riguarda lo Shallow Learning, la scelta è ricaduta su due approcci consolidati: le Support Vector Machine (SVM) e le Random Forest. Le SVM sono ampiamente utilizzate per risolvere problemi di separabilità lineare, sfruttando funzioni kernel per trovare un iperpiano ottimale che separi le istanze di classi diverse nello spazio delle caratteristiche. Le Random Forest, invece, rappresentano una tecnica di ensemble learning che combina multiple Decision Tree per effettuare predizioni, addestrando ciascun albero su una versione leggermente modificata del training set per aumentare la diversità e la robustezza del modello complessivo.

### Leave One Subject Out Cross Validation

L'esperimento è stato condotto utilizzando la tecnica di Leave One Subject Out Cross Validation, un'estensione della più nota Leave One Out Cross Validation. Questa metodologia è particolarmente significativa nel contesto del riconoscimento di attività basato su dati provenienti da sensori di smartphone. Considerando che ogni individuo ha un comportamento unico, questa tecnica permette di addestrare il modello su un gruppo di soggetti e testarlo su un soggetto completamente sconosciuto al classificatore.

In pratica, supponendo un dataset composto da trial relativi a  $n$  persone, il training viene effettuato utilizzando i dati di  $n-1$  soggetti, mentre il test viene condotto sugli esempi del soggetto escluso. Questo approccio è stato preferito a tecniche come lo Stratified Sampling, che avrebbe potuto includere nel test set esempi di azioni compiute da persone già presenti nel training set, compromettendo la valutazione della capacità di generalizzazione del modello a soggetti completamente nuovi.

Questa metodologia di validazione incrociata offre una valutazione più realistica delle prestazioni del modello in scenari del mondo reale, dove il sistema deve essere in grado di riconoscere attività di individui non visti durante la fase di addestramento. Ciò è particolarmente cruciale nello sviluppo di sistemi di riconoscimento attività robusti e quanto più generalizzabili possibili.

### Metriche di valutazione

Per decretare quale modello sia effettivamente il più efficace, si è partiti dalle matrici di confusione generate per ciascun utente del dataset. Queste matrici forniscono una rappresentazione dettagliata delle predizioni del classificatore, evidenziando sia i casi correttamente classificati che quelli errati.

Grazie a queste matrici è possibile definire alcune metriche utili per la valutazione del modello di seguito discusse.

L'*accuratezza* del modello è definita come il rapporto tra il numero di esempi correttamente classificati ed il numero di esempi totale

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

ove TP, TN, FP e FN sono elementi chiave nelle matrici di confusione e rappresentano i diversi risultati possibili di una classificazione binaria. I True Positives e i True Negatives rappresentano il numero di esempi correttamente classificati, rispettivamente, come appartenenti e non alla classe positiva. I False Positives e i False Negatives rappresentano, invece, il numero di esempi erroneamente classificati, rispettivamente, come appartenenti e non alla classe positiva.

Tuttavia, un alto valore di accuratezza non è sufficiente per valutare positivamente un classificatore, ad esempio, quando il test set è molto sbilanciato. Per questo motivo si è presa in considerazione un'altra metrica, detta *F1-score* o *F1-measure*, che fornisce una misura complessiva della qualità del classificatore, calcolata come media armonica tra la precisione ed il richiamo:

$$F_1 = \frac{2PR}{P + R}$$

dove con  $P = \frac{TP}{TP+FP}$  si indica la *precisione*, che rappresenta la percentuale degli esempi che viene correttamente classificata, mentre con  $R = \frac{TP}{TP+FN}$  si indica il *richiamo*, ovvero la percentuale degli esempi classificati come positivi che sono realmente positivi. Entrambi hanno valori compresi tra 0 ed 1. Idealmente, un classificatore ottimale dovrebbe raggiungere valori di precisione e richiamo entrambi prossimi ad 1.

## Implementazione

Per addestrare e testare i modelli si è utilizzata la piattaforma Google Colaboratory, un servizio che ci ha permesso di utilizzare il blocco note Jupyter Notebook, particolarmente adatto per machine learning e data science. Colab è un ambiente di sviluppo integrato (IDE), che non richiede alcuna configurazione per essere utilizzato e fornisce accesso gratuito alle risorse di elaborazione, tra cui GPU e TPU, insieme a strumenti per la scrittura, l'esecuzione e la condivisione di codice Python. Nell'IDE si sono utilizzate le seguenti librerie:

- Scikit-learn (v. 1.2.2): contiene i principali strumenti per effettuare analisi dei dati e apprendimento automatico, a partire da strumenti per il pre-processing, a quelli per la feature extraction e feature selection, ai principali modelli di apprendimento supervisionato o non supervisionato, per finire con le metriche di valutazione del modello.
- Seaborn (v. 0.13.2): contiene gli strumenti per la rappresentazione visuale dei dati, usata per visualizzare grafici e matrici di confusione
- Tensorflow (v. 2.15.0): fornisce metodi per addestrare e testare modelli di machine learning.
- Keras (v. 2.15.0): basata su Tensorflow, fornisce la possibilità di costruire ed addestrare modelli di deep learning.
- Numpy (v. 1.25.2): libreria con le principali funzioni matematiche per effettuare operazioni matematiche nello spazio  $\mathbb{R}^n$ .

- Pandas (v. 2.3.0): libreria che fornisce la possibilità di sistemare i dati in dataframe e di effettuare agevolmente delle operazioni su di essi.
- Joblib (v.1.4.2): insieme di strumenti per fornire pipeline leggera in Python. È ottimizzato per essere veloce e robusto in particolare su dati di grandi dimensioni e presenta ottimizzazioni specifiche per gli array Numpy.

Il processo inizia con la configurazione dei parametri iniziali, tra cui la selezione del dataset e delle feature rilevanti. L'esecuzione genera una struttura di cartelle temporalmente identificata per archiviare i risultati degli esperimenti, quindi tabelle dei risultati, plot delle matrici di confusione e configurazioni degli esperimenti.

La fase di pre-processing comprende il caricamento dei dati in un dataframe, mentre la fase di feature extraction comporta l'estrazione sia di feature per ogni record, come la Signal Vector Magnitude, la velocità e i quadrati delle componenti [si veda [Feature Extraction](#)], sia di feature statistiche aggregate.

Una volta che i dati sono stati preparati, nella fase successiva di costruzione dei modelli di classificazione, ogni modello viene validato utilizzando la tecnica del Leave One Subject Out Cross Validation (LOSO-CV), che isola iterativamente i dati di un singolo utente come test set, mentre i dati degli altri utenti costituiscono il training set.

Per gli esperimenti di deep learning, dal training set viene estratto un validation set, costituito dal 20% degli esempi del training set iniziale. L'estrazione del validation set è stratificata, mantenendo la proporzione delle classi rispetto al training set iniziale.

Prima di addestrare un classificatore, è necessario effettuare lo scaling dei dati per uniformare i range degli attributi. Per questo esperimento sono state usate diverse tecniche di scaling:

- MinMaxScaler: utilizzato per i dati da impiegare con classificatori di tipo shallow learning.
- Robust Scaler: utilizzato per i classificatori di deep learning.

Lo scaler viene fittato sul training set e applicato al test set.

A questo punto, viene inizializzato il modello di classificazione e viene avviata la fase di training. L'addestramento dei modelli di deep learning incorpora un meccanismo di EarlyStopping basato sul validation loss per prevenire l'overfitting. La valutazione utilizza metriche come l'f1-score e il validation loss. Al termine dell'addestramento, il modello viene testato e vengono generate matrici di confusione per ogni utente test. I modelli di shallow learning seguono un processo LOSO-CV semplificato, senza l'uso di un validation set separato. Questi esperimenti possono includere tecniche di bilanciamento dei dati o selezione delle feature utilizzando strumenti di scikit-learn.

La performance complessiva di ciascun modello è determinata dalla media delle valutazioni ottenute nelle singole iterazioni LOSO-CV, fornendo così una misura pesata dell'efficacia del classificatore su diversi soggetti.

## 5.2. Esperimento 2

Individuato il modello più efficiente, si procede alla sua implementazione all'interno dell'applicazione designata, al fine di ottenere un sistema robusto e accurato per il riconoscimento delle attività umane basato sui dati restituiti dall'accelerometro triassiale.

## Selezione e caratteristiche della piattaforma di hosting

*MovIng* rappresenta un'innovativa web application sviluppata con l'obiettivo di soddisfare esigenze specifiche nel campo della sicurezza informatica. Il suo scopo principale è quello di discriminare in tempo reale i movimenti effettuati dagli utenti, fornendo così uno strumento potente per il monitoraggio e l'analisi delle attività umane. In questo caso di studio, l'attenzione si è concentrata esclusivamente sull'approccio Sensor Based e nello specifico sull'utilizzo dell'accelerometro triassiale integrato nei dispositivi mobili degli utenti per la raccolta dei dati. Questo sensore fornisce i valori istantanei delle tre componenti dell'accelerazione. La sua capacità di misurare sia le accelerazioni lineari che l'accelerazione gravitazionale lo rende uno strumento ideale per determinare l'inclinazione del dispositivo e, di conseguenza, riconoscere le attività umane.

Per lo sviluppo e l'hosting della web application, si è fatto ricorso a PythonAnywhere, una piattaforma che offre un ambiente di sviluppo integrato (IDE) online e un servizio di web hosting basato sul linguaggio di programmazione Python. Questa scelta è stata motivata dalla versatilità e dalla facilità d'uso di PythonAnywhere, che fornisce un'ampia gamma di strumenti utili per lo sviluppo web anche a fronte della necessità di creare un'applicazione multiplatforma.

PythonAnywhere offre accesso dal browser alle interfacce a riga di comando Python e Bash, permettendo una gestione efficiente del progetto direttamente online. Un altro vantaggio significativo è il supporto per le applicazioni web scritte utilizzando qualsiasi framework applicativo basato su WSGI (Web Server Gateway Interface), offrendo così una grande flessibilità nella scelta delle tecnologie da utilizzare.

## Implementazione della web app

Il processo di sviluppo è iniziato con la configurazione dell'ambiente su PythonAnywhere. Il primo passo è stato la creazione del sito utilizzando Flask, un micro-framework web per Python noto per la sua semplicità e flessibilità. Nonostante Python 3.9 sia più stabile come versione, si sono riscontrati problemi con la compatibilità di altri pacchetti, motivo per cui si è optato per Python 3.10.

Successivamente, si è creato e configurato l'ambiente virtuale attraverso la console Bash di PythonAnywhere. Questo passaggio è stato fondamentale per garantire la compatibilità tra i vari pacchetti utilizzati. Durante questa fase, sono stati installati Flask e le librerie necessarie per il funzionamento dell'applicazione che coincidono con quelle utilizzate in fase di addestramento del modello.

Una volta preparato l'ambiente, si è passati al caricamento dello scaler e del modello pre-addestrato in formato pickle, un formato binario che permette di serializzare e deserializzare oggetti Python in modo efficiente. Uno *scaler* è uno strumento di preelaborazione dei dati che viene utilizzato per normalizzare i dati di input, assicurando che tutte le feature abbiano lo stesso formato. Ciò non solo migliora la qualità dei dati passati al modello, ma aumenta anche le prestazioni del modello stesso. Un *modello* di machine learning è un algoritmo che è stato addestrato su un insieme di dati per fare previsioni o classificazioni. Dopo l'addestramento, il modello contiene le informazioni apprese dai dati di addestramento, come i pesi delle connessioni in una rete neurale o i coefficienti di un modello di regressione.

Il passo successivo è stato quello di adattare `app.py` al modello e allo scaler caricati e creare i file `script.js` e `index.html` per gestire il lato client della web app, cercando di impostare un'architettura che garantisse una netta separazione tra front-end e back-end, in modo da assicurare una maggiore flessibilità e scalabilità dell'applicazione.

Il fulcro del back-end è rappresentato da *app.py*, implementata utilizzando il framework Flask, che gestisce le richieste HTTP del front-end attraverso due route principali:

- `/`: serve la pagina `index.html`;
- `/predict`: riceve i dati dell'accelerometro, li elabora usando il modello di machine learning caricato e restituisce una previsione al client.

Il file *wsgi.py* non è parte della logica applicativa, ma è cruciale per il deployment su PythonAnywhere, poiché configura l'ambiente per l'esecuzione dell'applicazione e funge da interfaccia tra il server web e l'applicazione Flask.

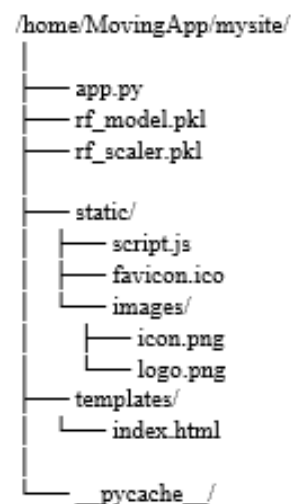
La gestione dell'interazione con l'utente e la logica dell'interfaccia sono delegate al front-end, implementate in *script.js* e *index.html*. Il primo si occupa di gestire gli eventi dal lato client (clic sui pulsanti, invio del form, ecc.), raccogliere i dati dell'accelerometro del dispositivo, inviare richieste al server e aggiornare l'interfaccia utente in base ai risultati ottenuti, mentre il secondo si occupa di gestire la parte puramente grafica dell'interfaccia utente.

Segue il funzionamento della web app MovIng. Quando un utente accede al sito, il server fornisce la pagina *index.html*. Il browser dell'utente carica quindi *script.js*, che gestisce le interazioni dell'utente con la pagina. La web app è stata resa cross-platform, di modo da rendere possibile un utilizzo e un testing più ampio su dispositivi differenti.

Quando l'utente avvia la registrazione dei movimenti, *script.js* inizia a raccogliere i dati dell'accelerometro del dispositivo, che vengono inviati al server attraverso una richiesta POST all'endpoint `/predict`. Il server, attraverso *app.py*, riceve questa richiesta, elabora i dati utilizzando il modello di machine learning precedentemente caricato, e restituisce una previsione sul tipo di movimento effettuato.

Lo script riceve questa risposta e, di conseguenza, aggiorna l'interfaccia utente, fornendo un feedback immediato all'utente sul tipo di attività rilevata. Questo ciclo continuo di raccolta dati, elaborazione e feedback permette agli utenti di ricevere informazioni in tempo reale sulle attività che stanno svolgendo.

La struttura finale delle directory rispecchia la separazione delle diverse componenti dell'applicazione e può essere schematizzata come in figura.



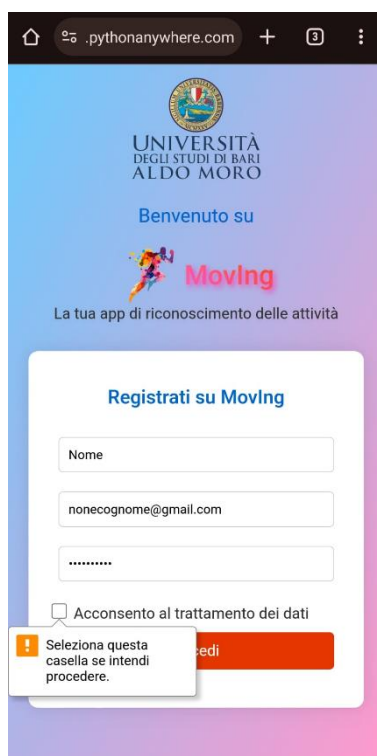
## Funzionamento e interfaccia di MovIng

Si presenta in figura la Home Page di MovIng. La schermata iniziale, in questo prototipo, è solo una registrazione fittizia, che richiede l'inserimento della mail in previsione del fatto che, per sviluppi futuri, sarà possibile implementare un profilo personale in grado di imparare a conoscere i comportamenti dell'utente e fornire report riassuntivi. Se l'utente non acconsente al trattamento dei dati, l'applicazione mostra un messaggio di richiesta esplicita per confermare la spunta dei dati.

Inoltre, il campo Password richiede:

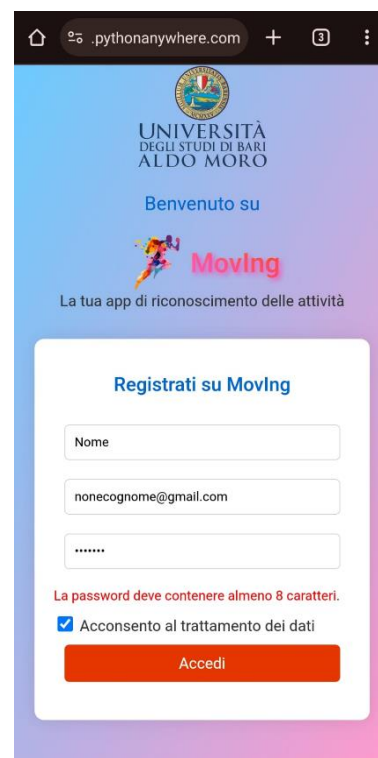
- una lunghezza non inferiore a 8 caratteri;
- l'inserimento di almeno un numero;
- l'inserimento di almeno un carattere speciale;
- l'inserimento di almeno una lettera maiuscola.

Questi meccanismi permettono all'utente di inserire una password che sia più sicura possibile.



The screenshot shows the registration page of the MovIng app. At the top, there is a header with the University of Bari Aldo Moro logo and the text "Benvenuto su Moving". Below this, a subtitle reads "La tua app di riconoscimento delle attività". The main form is titled "Registrati su Moving" and contains fields for "Nome", "Email" (pre-filled with "nonecognome@gmail.com"), and "Password" (pre-filled with "\*\*\*\*\*"). Below the password field is a checkbox labeled "Acconsento al trattamento dei dati". A red warning box is overlaid on the checkbox, stating: "Seleziona questa casella se intendi procedere." The "Accedi" button is partially visible at the bottom right.

Figura 1 – Consenso al trattamento dei dati



The screenshot shows the registration page of the MovIng app. At the top, there is a header with the University of Bari Aldo Moro logo and the text "Benvenuto su Moving". Below this, a subtitle reads "La tua app di riconoscimento delle attività". The main form is titled "Registrati su Moving" and contains fields for "Nome", "Email" (pre-filled with "nonecognome@gmail.com"), and "Password" (pre-filled with "\*\*\*\*\*"). Below the password field, a red message states: "La password deve contenere almeno 8 caratteri." Below this message is a checkbox labeled "Acconsento al trattamento dei dati" which is checked. The "Accedi" button is visible at the bottom right.

Figura 2 – Password non completa

L'applicazione è stata progettata per essere utilizzata sia da dispositivi Android, sia da dispositivi aventi come sistema operativo iOS. In questi ultimi è stata aggiunta la richiesta di consenso esplicito per accedere al Device Motion e al Device Orientation [Figura 1]. Nei dispositivi con sistema operativo Android, invece, viene utilizzato l'Accelerometer API e non viene richiesto questo consenso esplicito in quanto è già configurato come "accettato", sebbene possa essere revocato dalle impostazioni del sito [Figura 2].



Figura 3 – Autorizzazione su iOS

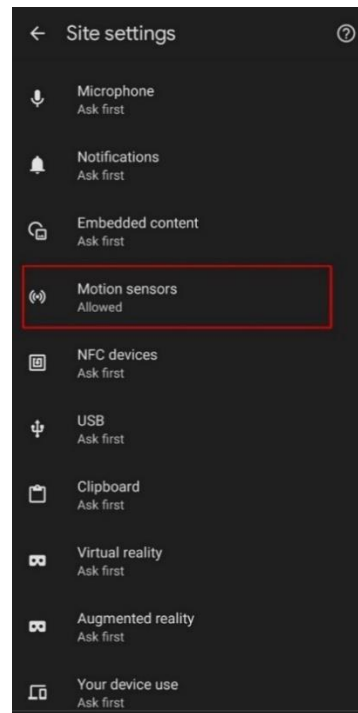


Figura 4 – Autorizzazione su Android

Una volta effettuato l'accesso, viene presentata all'utente la schermata di rilevamento. Solo quando l'utente seleziona il bottone "Inizia Rilevamento" [Figura 5] si avviano contemporaneamente del timer e del rilevamento. Durante il processo [Figura 6], che si è scelto di fissare a 45 secondi e di rendere non interrompibile se non in caso di logout dall'app, vengono raccolti e inviati i dati grezzi, e, una volta processati, vengono inserite le predizioni in un vettore che verrà utilizzato poi nella stampa dei risultati. Al termine del rilevamento, viene stampato a schermo il messaggio "Tempo scaduto" e l'utente ha la possibilità di scegliere se visualizzarne i risultati selezionando l'omonimo tasto o iniziarne uno nuovo [Figura 7].



Figura 5 – Inizio rilevamento

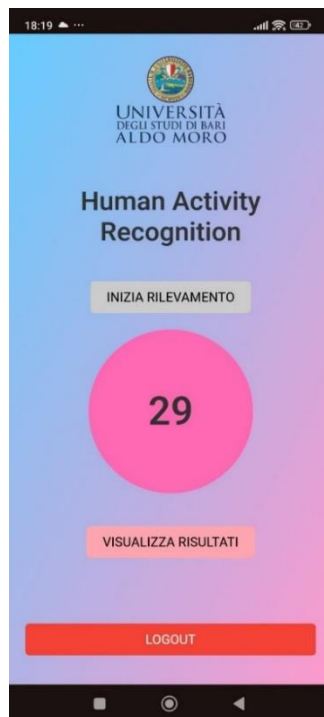


Figura 6 – Fase di raccolta dati



Figura 7 - Termine del rilevamento



Se l'utente sceglie di visualizzare i risultati, l'app presenta una nuova schermata, contenente una tabella con tutte le tipologie di azioni rilevabili e le relative percentuali. Queste rappresentano quanto, rispetto al numero totale dei rilevamenti avvenuti nell'arco temporale scelto, una determinata azione è stata riscontrata [Figura 8].



Azione	Percentuale
Fall	33.33%
Walking	0.00%
Jogging	0.00%
Hopping	66.67%
Sitting	0.00%

TORNA INDIETRO

Figura 8 –Tabella dei risultati

Una volta pigiato il bottone “Torna indietro” è possibile effettuare o nuovi test o il logout. In quest'ultimo caso viene chiesta all'utente la conferma [Figura 9], se la risposta è affermativa viene ripresentata la Home Page, con i dati precedentemente inseriti per effettuare il login [Figura 10].

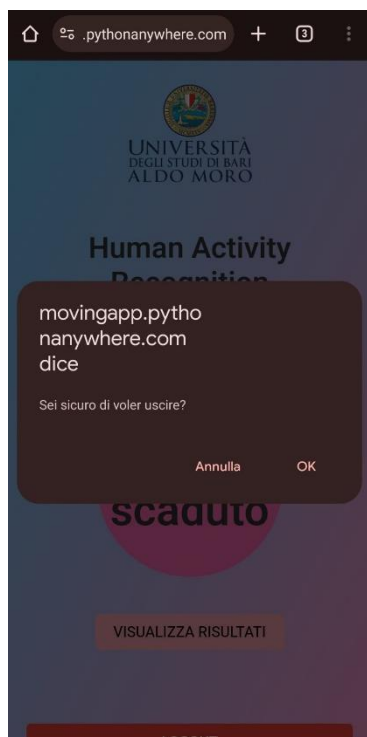


Figura 9 – Conferma di logout

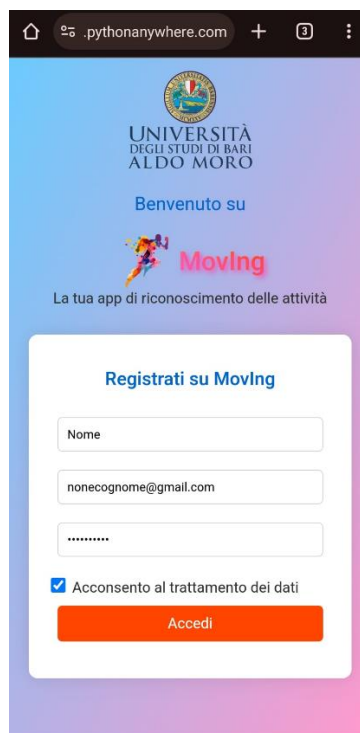


Figura 10 – Home Page dopo logout

## Metodologia e protocollo dei test con gli utenti

Per valutare l'efficacia e l'usabilità della web app sviluppata, è stato condotto un test su un campione di 5 utenti (di cui 2 con dispositivo iOS e 3 con dispositivo Android). Questo approccio ha permesso di seguire e osservare in modo diretto l'interazione degli utenti con l'interfaccia e di guidarli nel corretto svolgimento delle attività. Precondizioni al fine di un test eseguito correttamente sono che lo smartphone sia o posto nella tasca destra dei pantaloni, ben stabilizzato, o in alternativa che sia utilizzata una fascia per gamba.

Il protocollo di test è stato strutturato in fasi sequenziali, ciascuna progettata per esplorare aspetti specifici dell'applicazione:

1. Fase di accesso: gli utenti hanno avviato la web app e si sono interfacciati con la schermata di login.
2. Prima sessione di rilevamento: fase che ha incluso l'attivazione del timer, seguita dall'esecuzione di due attività fisiche distinte: camminata e salto sul posto. Agli utenti è stato poi chiesto di esaminare i risultati generati dal sistema e di tornare alla schermata principale.
3. Seconda sessione di rilevamento: gli utenti hanno eseguito un nuovo ciclo di attività, comprendente jogging e una simulazione di caduta. Anche in questo caso, è stato chiesto loro di esaminare i risultati e tornare alla schermata delle rilevazioni.
4. Terza sessione di rilevamento: ultima fase del test, in cui è stato chiesto agli utenti di simulare un'ulteriore sequenza di attività, che includeva camminata e l'assunzione della posizione seduta.

La struttura del test, con sessioni multiple e diverse combinazioni di attività, ha permesso di valutare non solo l'accuratezza del sistema di rilevazione, ma anche l'esperienza utente in termini di navigazione nell'app, comprensione dei risultati e facilità d'uso complessiva dell'interfaccia.

A seguito della sessione di test, è stato infatti richiesto agli utenti di compilare i questionari appositamente creati per esprimere il loro giudizio riguardo l'[efficacia](#) e dell'[usabilità](#) della Web App.

## 6. Risultati

### 6.1. Risultati 1 - Test dei modelli di classificazione

#### Scelta del Modello

Si riportano di seguito le tabelle delle valutazioni dei classificatori precedentemente enunciati con relativi punteggi di accuratezza e F1-score, rispettivamente con configurazioni di lunghezza di sliding window di 400, 600 e 800 (e valore di step pari a 100).

<i>Modello di classificazione</i>	<i>Accuracy Media</i>	<i>F1 media macro</i>	<i>F1 media weighted</i>
CNN	0.9488	0.9429	0.9470
Bi-LSTM	0.9419	0.9322	0.9369
SVM	0.8883	0.8658	0.8800

<u>RF</u>	<u>0.9507</u>	<u>0.9471</u>	<u>0.9492</u>
-----------	---------------	---------------	---------------

Tabella 1 - Sliding window di dimensione 400

<i>Modello di classificazione</i>	<i>Accuracy Media</i>	<i>F1 media macro</i>	<i>F1 media weighted</i>
CNN	0.9357	0.9241	0.9312
Bi-LSTM	0.9419	0.9312	0.9370
SVM	0.8941	0.8733	0.8871
<u>RF</u>	<u>0.9566</u>	<u>0.9533</u>	<u>0.9547</u>

Tabella 2 - Sliding window di dimensione 600

<i>Modello di classificazione</i>	<i>Accuracy Media</i>	<i>F1 media macro</i>	<i>F1 media weighted</i>
CNN	0.9430	0.9311	0.9376
Bi-LSTM	0.9598	0.9492	0.9542
SVM	0.9049	0.8848	0.8986
<u>RF</u>	<u>0.9641</u>	<u>0.9605</u>	<u>0.9611</u>

Tabella 3 - Sliding window di dimensione 800

L'analisi dei risultati ottenuti rispetto agli esperimenti precedenti condotti sul dataset preso in analisi rivela un miglioramento generale delle prestazioni dei modelli predittivi grazie all'uso delle feature aggiuntive estratte durante la fase di feature extraction. Le matrici di confusione indicano una riduzione degli errori nelle classificazioni di azioni simili, in particolare per le azioni di "caduta" e "seduta". L'analisi dettagliata delle matrici di confusione, ottenute tramite diverse prove con vari classificatori, evidenzia un comportamento interessante: alcuni utenti vedono una quasi perfetta accuratezza nella classificazione delle loro azioni, mentre per altri utenti il classificatore commette più errori. Questo fenomeno è principalmente attribuibile alla tecnica del Leave One Subject Out Cross Validation (LOSO-CV). Poiché il classificatore viene testato su utenti non presenti nel training set, le differenze individuali nei comportamenti, anche per la stessa azione, risultano più evidenti.

A fronte dei risultati sopra esposti, si è deciso di utilizzare il modello Random Forest, poiché ha mostrato le migliori prestazioni in termini di Accuracy e F1-Score rispetto agli altri modelli considerati, con una sliding window di dimensione 800. Si è inoltre rilevato come possibile alternativa dal punto di vista del deep learning il modello CNN.

## 6.2. Risultati 2 - Test della web app

Dalla fase di testing dei vari modelli di classificazione è emerso che i più promettenti fossero le Convolutional Neural Networks (CNN) e i Random Forest. Per determinare quale dei due approcci fosse più idoneo all'integrazione all'interno di MovIng, sono stati condotti ulteriori test comparativi. Dopo l'analisi dei risultati di tali test, si riportano anche i risultati dei test con gli utenti, ovvero la valutazione da parte degli utenti stessi dell'efficacia e dell'usabilità della Web App

### Risultati dei test comparativi sui modelli

Le CNN hanno dimostrato una notevole facilità e velocità nella fase di addestramento. Grazie alla loro capacità di estrarre automaticamente caratteristiche rilevanti dai dati grezzi, hanno raggiunto performance di classificazione elevate. Tuttavia, l'implementazione delle CNN all'interno della web app MovIng ha presentato diverse difficoltà. Una delle principali problematiche riscontrate è stata la dimensione dei pacchetti necessari per l'esecuzione e il funzionamento corretto delle CNN, che ha imposto pesanti requisiti di memoria e capacità computazionale del server, complicando la distribuzione e l'uso in ambiente web.

D'altra parte, i Random Forest si sono rivelati molto più adatti per l'integrazione in MovIng. Nonostante la maggiore complessità nelle fasi di apprendimento e computazione, i Random Forest hanno mostrato una straordinaria capacità di adattamento. La loro implementazione è risultata meno onerosa in termini di risorse computazionali, facilitando l'integrazione nel contesto della web app. Inoltre, la loro efficienza e le ottime performance predittive hanno garantito un funzionamento ottimale di MovIng, rendendola capace di gestire efficacemente le esigenze operative senza compromettere la qualità delle previsioni.

Nell'implementazione finale di MovIng, si è optato per un modello di Random Forest che utilizza esclusivamente i *dati grezzi* dell'accelerometro, escludendo le feature statistiche e derivate. Questa scelta è stata dettata dalla necessità di bilanciare l'efficienza computazionale con la tempestività delle predizioni. Sebbene l'utilizzo di feature aggiuntive avesse il potenziale di migliorare l'accuratezza del modello, si è osservato che ciò rallentava significativamente il processo di predizione, compromettendo l'obiettivo primario di un'applicazione che fosse in grado di riconoscere le azioni in tempo reale.

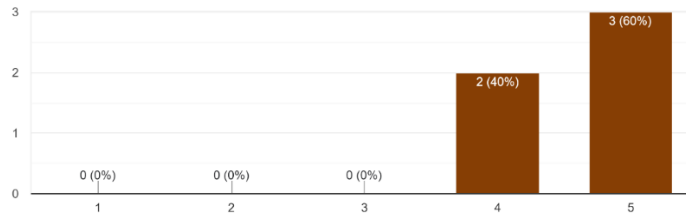
Si è fissata dunque come priorità attuale quella di garantire una risposta in tempo reale, essenziale per l'efficacia e l'usabilità dell'applicazione nel monitoraggio delle attività fisiche. Tuttavia, guardando al futuro, si prevede di esplorare strategie per incorporare feature aggiuntive senza compromettere le prestazioni in tempo reale. Questo potrebbe includere l'ottimizzazione degli algoritmi di estrazione delle feature, l'implementazione di tecniche di calcolo parallelo, o l'adozione di approcci di feature selection più sofisticati. L'obiettivo a lungo termine è quello di arricchire il modello con informazioni più dettagliate, mantenendo al contempo la reattività e l'efficienza che caratterizzano l'attuale versione di MovIng. Si è inoltre preferito utilizzato un modello di Random Forest che prendesse in input solo i dati grezzi dell'accelerometro, senza considerare le feature statistiche e derivate. Questo perché, provando a utilizzare il modello addestrato con feature aggiuntive, ha rallentato di molto il processo di predizione.

## Valutazione dell'Efficacia e Usabilità della Web App

L'efficacia e l'usabilità dell'applicazione sono state valutate attraverso due strumenti principali: il questionario System Usability Scale (SUS) e un questionario specifico per la valutazione dell'efficacia. Questi strumenti hanno fornito insights preziosi sulla percezione degli utenti e sulle prestazioni dell'applicazione.

Il questionario SUS, riconosciuto come standard nell'industria per la valutazione dell'usabilità, ha prodotto risultati particolarmente incoraggianti. L'applicazione ha ottenuto uno score complessivo molto positivo, indicando un alto livello di gradimento da parte degli utenti.

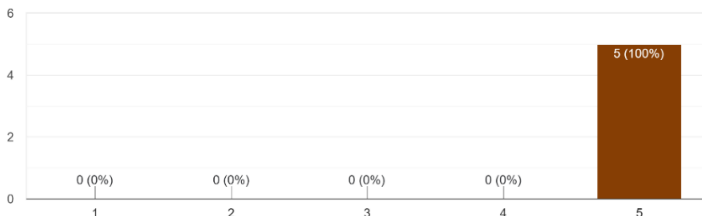
Mi sono sentito a mio agio nell'utilizzare il sito  
5 risposte



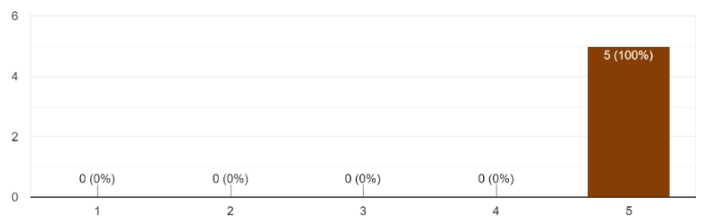
Questo punteggio elevato riflette in particolare l'apprezzamento degli utenti per due aspetti chiave dell'applicazione:

1. **Semplicità:** Gli utenti hanno valutato positivamente la facilità d'uso dell'interfaccia, suggerendo che l'applicazione è intuitiva e non richiede una curva di apprendimento ripida.

Ho trovato il sito molto semplice da usare  
5 risposte

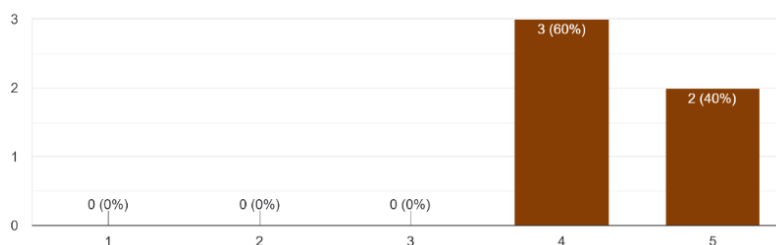


Penso che la maggior parte delle persone possano imparare ad utilizzare il sito facilmente  
5 risposte



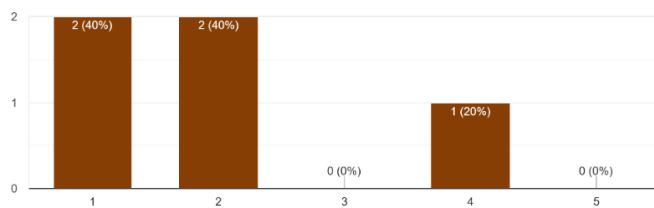
2. **Usabilità:** Il design e la struttura dell'applicazione sono stati percepiti come efficienti e user-friendly, permettendo agli utenti di navigare e utilizzare le funzionalità con facilità.

Ho trovato le varie funzionalità del sito bene integrate  
5 risposte

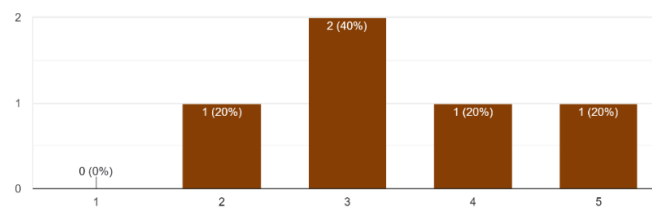


Questi risultati sono particolarmente significativi considerando l'importanza dell'esperienza utente nel determinare il successo e l'adozione di nuove tecnologie.

Ho trovato incoerenze tra le varie funzionalità del sito  
5 risposte



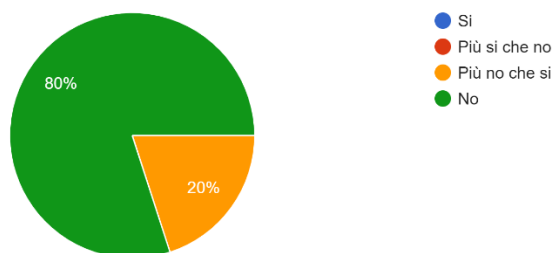
Mi piacerebbe utilizzare questo sito frequentemente  
5 risposte



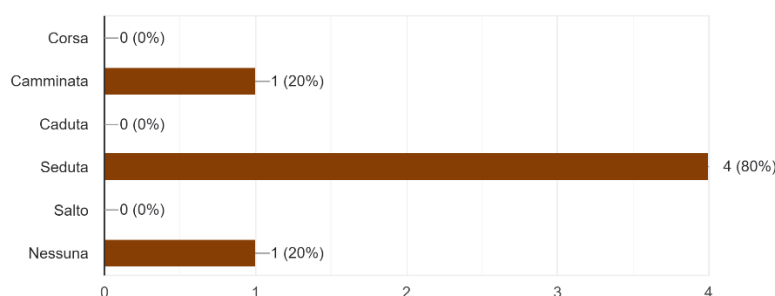
La combinazione di semplicità e usabilità suggerisce che l'applicazione ha raggiunto un equilibrio ottimale tra funzionalità e accessibilità, un fattore cruciale per applicazioni destinate a un uso quotidiano o frequente.

Il questionario di valutazione dell'efficacia ha rivelato risultati notevoli. Gli utenti hanno apprezzato significativamente l'estetica e l'usabilità dell'applicazione e hanno riportato feedback positivi sia riguardo la poca invadenza sia riguardo la quasi totale assenza di problemi nei rilevamenti effettuati dall'applicazione.

I dati personali richiesti nella pagina di login ti sono sembrati invadenti?  
5 responses



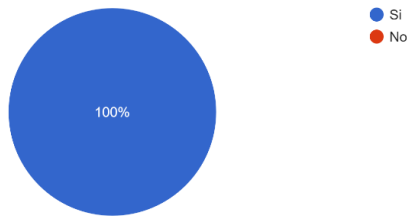
Hai rilevato problemi nel rilevamento di qualche specifica attività tra quelle possibili?  
5 responses



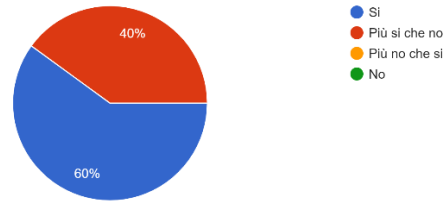
Sicuramente uno spazio importante di miglioramento è da prevedere nella distinzione tra fall e sitting e tra walking e jogging, con un eventuale calcolo delle feature derivate come i quadrati dei componenti e la velocità, note per la loro capacità di diminuire il margine di errore.

Un aspetto che ha suscitato particolare attenzione e interesse tra gli utenti è stato quello delle potenziali implementazioni e sviluppi futuri di MovIng. Le possibilità di espansione e miglioramento dell'applicazione sono state accolte con entusiasmo, suggerendo un forte coinvolgimento e un'aspettativa positiva per le future versioni del prodotto.

Ti piacerebbe se l'app fosse estesa al riconoscimento di altre attività?  
5 responses



Saresti interessat\* ad avere un'applicazione personalizzata in grado di riconoscere le tue attività e avvisarti in caso di comportamenti anomali?  
5 responses



Questi feedback confermano l'efficacia dell'attuale implementazione di MovIng e indicano anche direzioni promettenti per ulteriori ricerche e sviluppi.

Quali attività potrebbero essere incluse?  
3 responses

salire/scendere scale  
Andamento giornaliero e annuo  
Stare fermo sul posto, stendersi

## 7. Conclusioni

A seguito degli esperimenti effettuati, si può concludere che MovIng rappresenti efficacemente un'applicazione innovativa nel campo della sicurezza informatica e del monitoraggio delle attività umane in real-time.

Nell'implementazione finale di MovIng, si è optato per un modello di Random Forest, poiché ha mostrato le migliori prestazioni in termini di Accuracy e F1-Score rispetto agli altri modelli considerati, con una sliding window di dimensione 800. Questo utilizza esclusivamente i dati grezzi dell'accelerometro, escludendo le feature statistiche e derivate. Questa scelta è stata dettata dalla necessità di bilanciare l'efficienza computazionale con la tempestività delle predizioni. Sebbene l'utilizzo di feature aggiuntive avesse il potenziale di migliorare l'accuratezza del modello, si è osservato che ciò rallentava significativamente il processo di predizione, compromettendo l'obiettivo primario di un'applicazione che fosse in grado di riconoscere le azioni in tempo reale.

I risultati dei questionari hanno indicato un successo significativo nell'implementazione dell'applicazione, particolarmente in termini di design centrato sull'utente e facilità d'uso. Questi fattori sono fondamentali per assicurare non solo l'adozione iniziale dell'applicazione, ma anche il suo uso continuo nel tempo. In conclusione, MovIng dimostra come l'integrazione di diverse tecnologie possa portare alla creazione di soluzioni innovative e attuabili con efficacia in un contesto reale.

### Sviluppi futuri

La struttura modulare dell'applicazione, con una chiara separazione tra front-end e back-end, non solo migliora la manutenibilità del codice, ma permette anche future espansioni e miglioramenti.

Sviluppi futuri possibili sarebbero integrare altri sensori per una rilevazione più accurata dei movimenti, o implementare funzionalità di analisi dei dati a lungo termine per identificare pattern di movimento nel tempo. È sicuramente da includere, infatti, l'implementazione del calcolo di features derivate e statistiche, riuscendo a bilanciare le stesse con la velocità di predizione, poiché è chiaro che queste potrebbero portare a una riduzione del margine di errore tra fall e sitting, e tra walking e jogging. Questa applicazione apre la strada a numerose possibilità nel campo del monitoraggio delle attività umane, con potenziali applicazioni che spaziano dalla sicurezza personale al monitoraggio della salute e del fitness.

## Riferimenti

1. L. Minh Dang, K. Min, H. Wang, M. J. Piran, C. H. Lee e H. Moon. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition*. 2020.
2. Gupta, S. Deep learning based human activity recognition (HAR) using wearable sensor data. *International Journal of Information Management Data Insights*. 2021.
3. K. Muralidharan, R. Anirudh, G. Rithvik, P. Saket, A. A. Reghunaath e M. P. Dr. Gopinath. 1D Convolution approach to human activity recognition using sensor data and comparison with machine learning algorithms. *International Journal of Cognitive Computing in Engineering*. 2021.
4. Yoon, H. Cho e S. M. Applying Singular Value Decomposition on Accelerometer Data for 1D Convolutional Neural Network Based Fall Detection. *Electronics Letters*, vol. 55, n. 6. 2019.
5. J. Pilataxi Piltaxi, M. Trujillo Guerrero, V. Benavides Laguapillo e J. Rosales Acosta. Human Activity Recognition Using an Accelerometer Magnitude Value. *Communications in Computer and Information Science*, vol. 1194. 2019.
6. L. Ye, J. Shi, H. Ferdinando, T. Seppanen e E. Alasaarela. A Multi-sensor School Violence Detecting Method Based on Improved Relief-F and D-S Algorithms. *Mobile Networks and Applications*. 2020.
7. Zhanfeng, Z. Zihan e Z. Campus bullying detection based on motion recognition and speech emotion recognition. *Journal of Physics: Conference Series*. 2019.
8. Roggen, F. Ordóñez e D. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors*. 2016.
9. Ghate, V. Hybrid deep learning approaches for smartphone sensor-based human activity recognition. *Multimedia Tools and Applications*. 2021.
10. L. Atallah, B. Lo, R. King e G. Z. Yang. Sensor Positioning for Activity Recognition Using Wearable Accelerometers. *IEEE Transactions on Biomedical Circuits and Systems*. 2011.
11. A. Bayat, M. Pomplun e D. Tran. A Study on Human Activity Recognition Using Accelerometer Data from Smartphones. *Procedia Computer Science*. 2014.
12. H. Chunyu, C. Yiqiang, H. Lisha e P. Xiaohui. A novel random forests based class incremental learning method for activity recognition. *Pattern Recognition*. 2018. 28
13. Yoon, H. Cho e S. M. Applying Singular Value Decomposition on Accelerometer Data for 1D Convolutional Neural Network Based Fall Detection. *Electronics Letters*. 2019.
14. A. E. Minarno, W. A. Kusuma, H. Wibowo, D. R. Akbi e N. Jawas. Single Triaxial Accelerometer-Gyroscope Classification for Human Activity Recognition. *8th International Conference on Information and Communication Technology*. 2020.
15. Stephen J. Preece, John Yannis Goulermas, Laurence P. J. Kenney, and David Howard. A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities From Accelerometer Data. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*. 2009.
16. Kuang-Hsuan Chen, Jing-Jung Yang & Fu-Shan Jaw. Accelerometer-Based Fall Detection using Feature Extraction and Support. *Instrumentation Science & Technology*. 2015.



17. Haibo He, Edwardo A. Garcia. *Learning from Imbalanced Data*. s.l. : IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 2009.
18. L. Ye, J. Shi, H. Ferdinando, T. Seppanen, E. Alasaarela. A Multi-sensor School Violence Detecting Method Based on Improved Relief-F and D-S Algorithms. *Mobile Networks and Applications*. 2020.
19. H. Mengqi, W. Haixia, X. Zechen e Z. Guilin. An SVM fall recognition algorithm based on a gravity acceleration sensor. *Systems Science & Control Engineering*. 2018.