

Abstract

L’obiettivo della tesi è quello di studiare algoritmi basati su Deep Learning per l’identificazione e l’autenticazione di utenti usando segnali inerziali, in particolare accelerometro e giroscopio. A tale scopo sono stati analizzati i principali approcci e dataset nello stato dell’arte. Dall’analisi fatta, è emerso che esistono dataset progettati per studiare specificamente il problema di identificazione e autenticazione e dataset che invece sono progettati per studiare algoritmi di riconoscimento attività che possono essere adattati per il nostro scopo. Riguarda la prima categoria di dataset, abbiamo scelto di utilizzare OUISIR che ha un numero consistente di utenti (oltre 700) che sono stati registrati durante l’attività di camminata, i segnali disponibili sono accelerometro e giroscopio. Riguardo la seconda categoria di dataset abbiamo scelto di utilizzare Unimib-SHar, Sbhar e Realdisp che sono dataset per il riconoscimento di Activity of Daily Living (ADL) e hanno la peculiarità di contenere nelle annotazioni anche le identità delle persone. In questo caso il numero di indentità disponibili è piuttosto basso e va in range da 17 a 30 utenti. Dopo aver implementato un metodo noto nello stato dell’arte che si preoccupa di riconoscere le identità e le attività simultaneamente, ci siamo concentrati sulla progettazione di una nostra soluzione basata su Residual Network (ResNet). Gli esperimenti effettuati sul task di identificazione e autenticazione dimostrano che la soluzione proposta, sebbene computazionalmente meno onerosa, ottiene prestazioni comparabili se non migliori degli approcci nello stato dell’arte.

Contents

1	Introduction	3
2	Biometrics system	5
2.1	Physiological biometrics	7
2.2	Behavioral biometrics	10
2.3	Inertial signal	14
2.4	Authentication system	17
2.4.1	Structure	17
2.4.2	Multi-biometrics system	18
2.4.3	Types of authentication	20
2.5	Evaluation metrics	21
3	State of the art	23
3.1	Public datasets	23
3.1.1	Gait datasets	23
3.1.2	ADL datasets	27
3.2	Preprocessing techniques	29
3.3	Segmentation methods	30
3.3.1	Gait/step cycle segmentation	30
3.3.2	Chunks of signal	32
3.4	Recognition methods	33
3.4.1	Gait	33
3.4.2	ADL	37
4	Modello	40
4.1	Preprocessing segnale	40
4.2	Data augmentation	42
4.3	ResNet like model	47

5	Experimental setups	51
5.1	Identification	51
5.2	Authentication	61
6	Risultati	68
6.1	ADL	68
6.2	Walking	70
7	Conclusioni e sviluppi futuri	71

Chapter 1

Introduction

Il diffondersi delle tecnologie moderne, che ci permettono di accedere in modo immediato e più semplice a servizi in remoto, introducono anche nuovi problemi legati alla sicurezza. Spesso le applicazioni che utilizziamo ormai in modo naturale, basti pensare al crescente uso di smartphone e tablet, hanno l'autorizzazione di leggere, modificare e condividere dati sensibili (e.g. numero/pin carte di credito, lista contatti, media). E' necessario quindi, seguendo il trend tecnologico, studiare e sviluppare nuove tecniche di autenticazione che blocchino l'accesso non autorizzato a servizi remoti (e.g. home banking), a dispositivi mobili (e.g. smartphone, tablet, smartwatch), ad aree fisiche.

Le tecniche usate per verificare l'autenticità di un utente possono essere raggruppate in 3 strategie principali: *knowledge-based*, *ownership-based* e *biometrics-based* [1]. Queste 3 categorie corrispondono a 3 domande elementari usate per verificare l'identità: *what you know*, *what you have* and *what you are* [2].

1. *Knowledge-based*. Queste è la tecnica di autenticazione maggiormente utilizzata. Sfrutta la conoscenza di una sequenza, text-based (PIN, password) o graphic-based (e.g. lock pattern), oppure di una o più risposte ad una o più domande. E' la tecnica maggiormente utilizzata perché semplice da implementare e da utilizzare per l'utente, però ha diversi svantaggi dal punto di vista della sicurezza, per esempio uno smartphone sbloccato può essere rubato e utilizzato liberamente, oppure spesso una persona che deve ricordare molte password tende ad utilizzare sempre la stessa.
2. *Ownership-based*. E' caratterizzata dal possesso di qualcosa di fisico,

come per esempio una chiavetta che genera dei token random. Spesso sono usate con delle password o PIN, in modo che la perdita dell'oggetto fisico non causi problemi di sicurezza.

3. *Biometrics-based*. Questa categoria comprende l'uso di una o più caratteristiche umane, fisiche o comportamentali, che rispettino le seguenti proprietà: 1) *universalità*, ogni persona deve possederla, 2) *unicità*, due persone non possono essere uguali secondo la caratteristica, 3) *permanenza*, deve essere invariante nel tempo, 4) *misurabile*, si deve poter misurare quantitativamente [3].

Per la tesi si è scelto di concentrarsi sullo studio e implementazione di algoritmi di autenticazione biometrics-based. Questo perché si è notato che gli approcci knowledge-based e ownership-based presentano, oltre a possibili limiti di sicurezza, un altro problema fondamentale: non permettono lo sviluppo di sistemi in grado di autenticare in modo continuo nel tempo (e.g durante una sessione) l'utente, ma offrono solo un punto sicuro di ingresso. Nello specifico si è scelto di studiare i segnali inerziali, in quanto a differenza di altri caratteri biometrici umani permettono una continua autenticazione che non richiede un'interazione esplicita dell'utente.

La tesi si sviluppa come segue: nel capitolo **2** verranno descritti con più precisione i sistemi biometrici, analizzando le principali caratteristiche umane sfruttate nei sistemi biometrici, descrivendo i segnali inerziali usati in seguito e definendo la struttura di un sistema di autenticazione basato su caratteristiche biometriche e le metriche utilizzate per quantificare la bontà del sistema. Nel capitolo **3** verranno descritte le principali tecniche utilizzate nello stato dell'arte e i principali dataset trovati dopo l'analisi della letteratura. Nel capitolo **4** sarà descritto in dettaglio il modello implementato, dal pre-processing dei dati all'autenticazione degli utenti. Nel capitolo **??** saranno descritte alcune tecniche di data augmentation utilizzate per migliorare le performance del modello. Infine nei capitoli **6** e **7** verranno presentati i risultati ottenuti, i possibili sviluppi futuri e le conclusioni.

Chapter 2

Biometrics system

Nel corso del capitolo saranno descritte le principali caratteristiche biometriche, con possibili vantaggi e svantaggi, che possono essere in gran parte riassunte nelle seguenti figure 2.1, 2.2 [4]. Successivamente verranno analizzati in modo più specifico i sistemi di acquisizione basati su wearable-sensor, utilizzati durante la tesi, e infine sarà descritto in generale un sistema di autenticazione basato su caratteristiche biometriche: struttura, tipologie e metriche utilizzate.

Come detto nel capitolo 1 i sistemi di autenticazione basati su caratteristiche biometriche possono essere divisi in due categorie: fisiologici e comportamentali.

Le differenze fondamentali tra le due categorie sono le seguenti:

- generalmente le caratteristiche fisiologiche sono relativamente invarianti nel tempo, a meno di fattori esterni, mentre gli aspetti comportamentali di un utente si modificano nel tempo oppure possono essere influenzati dall'aspetto psicologico e dalla condizione fisica [2]. Questa differenza si riflette nell'implementazione dei sistemi di autenticazione basati su biometria, infatti se sfruttano aspetti comportamentali dovrebbe sviluppare un meccanismo di aggiornamento dei dati raccolti per poter rappresentare anche le nuove possibili caratteristiche;
- come raffigurato nell'immagine 2.1, le caratteristiche fisiologiche non permettono di implementare un sistema di autenticazione continuo nel tempo, ma solo un punto di ingresso sicuro, in quanto richiedono che l'utente esegua un'azione specifica; analizzando invece il comportamento di un utente è possibile un monitoraggio che non prevede una

specifica azione da parte dell'utente e quindi un'autenticazione continua.

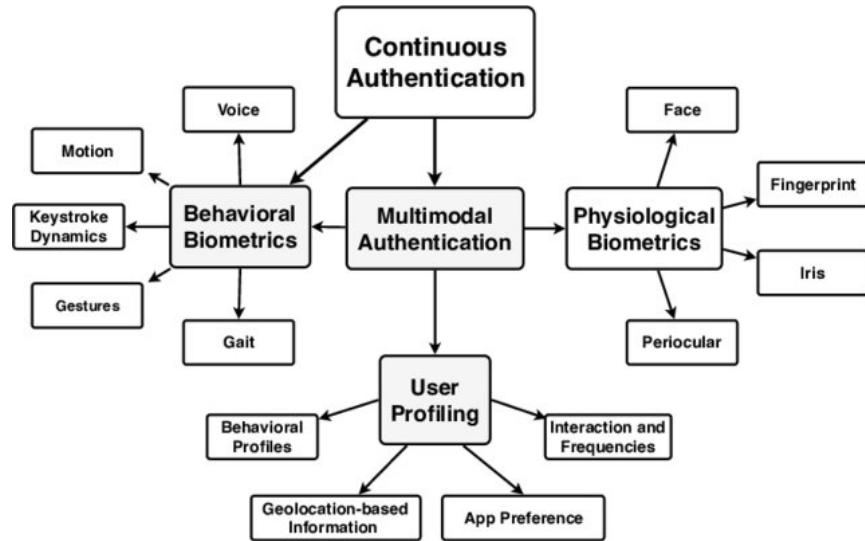


Figure 2.1: Tassonomia delle principali caratteristiche biometriche

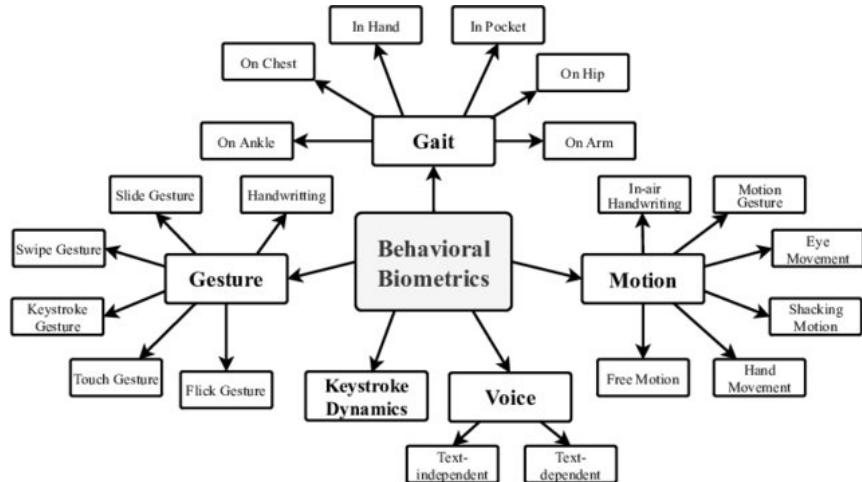


Figure 2.2: Tassonomia delle principali caratteristiche biometriche comportamentali

2.1 Physiological biometrics

Di seguito verranno analizzate le principali caratteristiche fisiologiche sfruttate nei sistemi biometrici.

Fingerprint

Un sistema di autenticazione basato sulle impronte digitali verifica l'autenticità di un utente analizzando i suoi polpastrelli. E' uno dei metodi di autenticazione più utilizzati al giorno d'oggi (e.g. smartphone, tablet) in quanto presenta un alto livello di sicurezza, le tecnologie di acquisizione sono sempre meno costose e inoltre per ogni persona è possibile registrare in modo univoco i dieci polpastrelli delle mani. L'idea base è quella di analizzare le features formate dalle increspature presenti nei polpastrelli, figura 2.3. I dispositivi di acquisizione più usati sono camere e scanner [5] (matrice di sensori) come quelli montati sugli smartphone moderni. Recentemente sono stati proposti nuovi sistemi basati su scanner ottici e a ultrasuoni [6]. Un attuale ramo di ricerca si basa su tecniche touch-less, che non implicano il contatto diretto con un sensore, come presentato in [7]

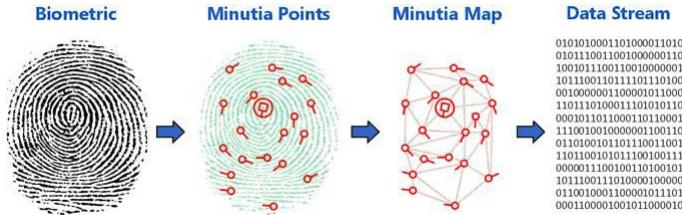


Figure 2.3: Analisi impronte digitali e rappresentazione binaria

Palmpoint

La caratteristica biometrica è molto simile alle impronte digitali ma vengono analizzate le increspature e features di tutto il palmo della mano [8], figura 2.4. Il problema principale è, a differenza delle impronte digitali, che piccole variazioni nella posizione della mano (e.g rotazioni) possono implicare modifiche nelle features estratte. I principali dispositivi di acquisizione sono le camere.

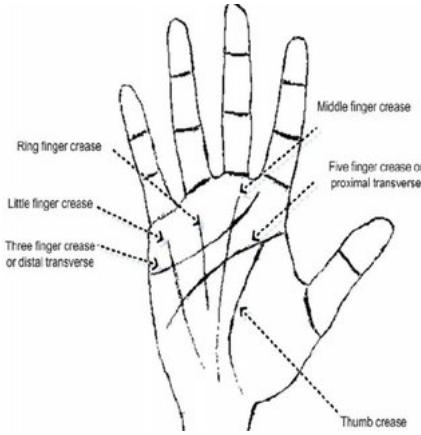


Figure 2.4: Features ad alto livello presenti sul palmo della mano

Hand geometry

Vengono analizzate alcune caratteristiche ad alto livello della mano (e.g lunghezza delle dita, larghezza palmo) come mostrato in figura 2.5, le quali sono invarianti ma spesso non univoche data un alto numero di utenti [9].

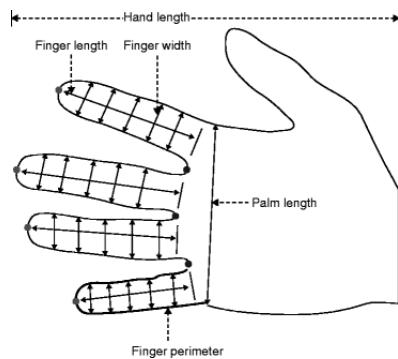


Figure 2.5: Esempi di features ad alto livello nella geometria della mano

Face

Il viso degli utenti è usato come caratteristica soprattutto in sistemi di video sorveglianza, e si basa sull'analisi delle possibili features estratte dalla forma e dalle differenti texture presenti sul viso. Questa caratteristica biometrica,

a differenza di quelle citate in precedenza, può essere sfruttata per implementare un sistema biometrico passivo (autenticazione continua), in quanto non prevede un'interazione obbligata dell'utente. Può essere usata per eseguire identificazioni all'interno di una grande quantità di persone (masse di persone in spazi ridotti), però a causa di vari problemi, per esempio cambi di illuminazione, posizione o espressioni del viso, è complicato implementare un sistema performante [10]. Oltre a possibili problemi di implementazioni il principale quesito rispetto all'uso di questa caratteristica è quella legata alla privacy degli utenti.

Iris

E' uno dei tratti biometrici più complicati da analizzare a causa della sua complessa struttura, ma nello stesso tempo uno dei migliori per le seguenti caratteristiche: è univoco (anche tra gemelli), presenta più di 266 gradi di libertà, è protetto da eventuali agenti esterni, non degenera con l'età e la sua struttura non viene influenzata da lenti o occhiali. Tra le caratteristiche biometriche attualmente utilizzate è quella che mostra i migliori risultati in termini di performance. Il principale aspetto negativo è dato dalla necessità di una distanza minima tra l'iride e il dispositivo di acquisizione, quindi non è possibile implementare sistemi di autenticazione continua.

Electrocardiograph and electroencephalograph

Indicate con le sigle ECG e EEG, indicano l'attività del cuore e le onde cerebrali, entrambi sono rappresentati da segnali unidimensionali. Recenti studi hanno dimostrato che presentano delle unicità rispetto a diversi individui [11]. Entrambe le caratteristiche possono essere sfruttate per sistemi di autenticazione continui.

2.2 Behavioral biometrics

Di seguito verranno analizzate le principali caratteristiche comportamentali sfruttate nei sistemi biometrici.

Keystroke

Si basa sull'analisi della dinamica con cui l'utente preme una sequenza di tasti e le principali features, secondo [12], sono: la frequenza con cui vengono premuti i tasti, la frequenza con cui vengono rilasciati, la frequenza con cui viene usato il tasto per cancellare (figura 2.6). Solitamente tutte queste features vengono registrate durante la scrittura di uno specifico testo o una sequenza di numeri. Con l'introduzione dei dispositivi touch lo spazio di possibili key-input aumenta in quanto vengono considerati anche la pressione esercitata, le aree dello schermo maggiormente utilizzate e altre possibili gestures (e.g. swipe), che non implicano l'inserimento di un testo. Secondo alcuni recenti studi [13, 14] i sistemi di autenticazione basati su keystroke dynamics permettono di raggiungere ottime performance, inoltre permettono di implementare sistemi molto efficienti dal punto di vista energetico e indipendente dall'hardware disponibile (e.g. tastiera fisica o su screen). Esistono comunque ragioni per cui implementare sistemi di questo tipo è complicato per esempio è necessario tenere conto di come cambia il comportamento dell'utente in situazioni non controllate (e.g. durante lo svolgersi di diverse attività) oppure è naturale che non può essere utile nel momento in cui un utente non interagisce con il dispositivo.

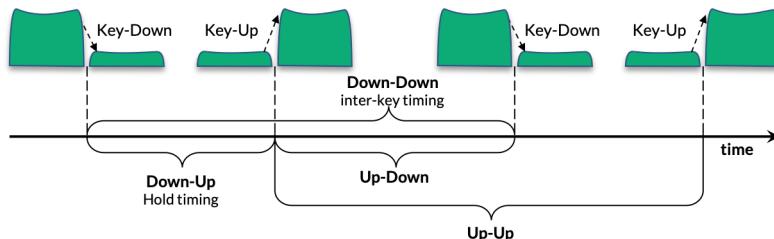


Figure 2.6: Esempi di features analizzate nell'approccio keystroke

Touch-gesture

L'uso delle gesture su schermi touch può essere visto come un'estensione rispetto all'approccio keystroke. In questo caso infatti non è previsto l'inserimento

di un input da parte dell'utente ma vengono analizzate gestures (e.g. swipes, flicks, slides, handwriting), figura 2.7, che in generale sono eseguiti su qualsiasi schermo touch (e.g. smartwatches, monitor, sistemi di navigazione). Alcuni studi hanno evidenziato come un approccio basato sulle gesture sia dipendente dall'applicazione utilizzata [15, 16], una possibile soluzione proposta da [17] è quella di utilizzare un approccio context-aware, che tiene conto dell'applicazione utilizzata. Un aspetto difficoltoso di questo approccio è la scelta della frequenza di campionamento della gestione, se troppo alta può essere più difficile catturare e replicare la gestione. In generale anche questo tipo di biometria è molto influenzata da cambiamenti comportamentali e attività svolte dall'utente.

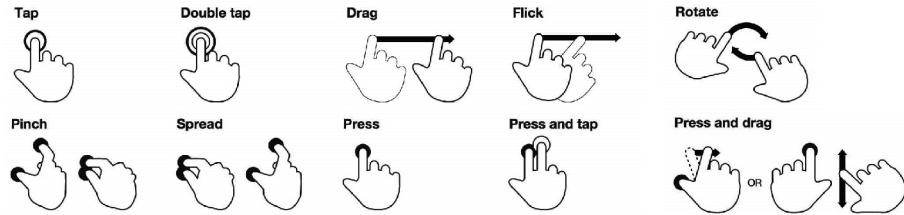


Figure 2.7: Esempi di gestures usate in sistemi di autenticazione

Voice

Le features legate alla voce comprendono aspetti fisiologici (e.g. caratteristiche labbra) e tratti comportamentali (e.g. tono legato alle emozioni o all'età). È uno degli aspetti biometrici più investigati in letteratura e si può dividere in due tipi di approcci: text-dependent e text-independent. Nel primo caso si sfrutta una frase specifica che l'utente deve pronunciare, è un sistema molto accurato ma non permette un'autenticazione continua; nel secondo approccio invece le features sono estratte in modo indipendente dalle parole pronunciate, quindi il sistema è molto flessibile e permette un'autenticazione passiva. Le principali sfide nell'implementazione di un'autenticazione basata sulla voce sono i possibili cambiamenti legati alla condizione fisica (e.g. persona malata, affaticata) ed emotiva e la necessità di un robusto sistema di acquisizione e pre-processing dei dati a causa del rumore introdotto dall'ambiente.

Gait

La dinamica della camminata di un utente può essere descritta come una serie di pattern periodici che a loro volta possono essere scomposti in 2 fasi principali e 8 configurazioni elementari, come viene riportato nello studio [18] e mostrato in figura 2.8.

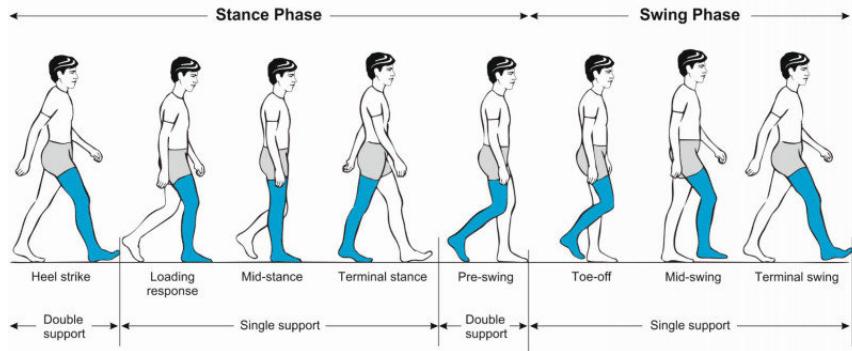


Figure 2.8: Varie fasi periodiche in cui viene divisa la camminata [19]

L'intero periodo viene definito gait cycle, mentre le prime quattro configurazione e le successive quattro compongono un left e right step.

Secondo l'analisi effettuata in [20] la durata media di un singolo gait cycle è nel range 0.98-1.07 secondi; questo dato è molto importante perché permette di decidere, data la frequenza di campionamento in fase di acquisizione del segnale, come dividere il segnale per catturare la periodicità della camminata.

Sulla base del dispositivo di acquisizione la camminata produce pattern visuali, distribuzione di pressione sul terreno e pattern in uno spazio 3D (e.g accelerometro) che sono caratteristici dell'utente e possono essere analizzati per discriminare i vari soggetti.

In accordo con [21] in letteratura si trovano 3 principali tipi di approcci per il riconoscimento basato su questa caratteristica biometrica:

- *Machine Vision-based*: modellano la camminata analizzando le caratteristiche visuali catturate da immagini e/o video. Soffre di alcuni problemi comuni nel campo di image preprocessing come per esempio variazioni nell'illuminazione, occlusioni, diverse pose e posizione delle camere, inoltre le performance possono essere anche influenzate dal tipo di abiti indossati e oggetti portati [22].

- *Floor Sensors-based*: sfruttano sensori di pressione che catturano proprietà quantitative del passo (e.g. forza nella pressione, durata). Questo approccio è stato "abbandonato" recentemente per le basse performance ottenute e per l'alto costo dei dispositivi d'acquisizione.
- *Wearable Sensors-based*: usano sensori "indossati" direttamente dall'utente (e.g accelerometro, giroscopio) e catturano la dinamica della camminata; non è necessario pre-impostare un ambiente di controllo e quindi possono essere sfruttati per un riconoscimento ubiquo. I pattern catturati possono subire variazioni più o meno significative per vari fattori come per esempio la velocità di camminata, il tipo di scarpe indossate, la pendenza della superficie e una diversa posizione del dispositivo.

Analizzando i tre gruppi si può concludere che il più promettente è l'approccio wearable sensors-based, in quanto a differenza del caso machine vision-based può operare a qualsiasi distanza.

Nel complesso l'uso del passo come caratteristica biometrica presenta delle buone caratteristiche per implementare un sistema di autenticazione per i seguenti motivi:

- non esistono limitazioni di distanze a cui può operare (eccetto machine vision);
- non è intrusivo, in quanto non richiede nessuna specifica azione da parte dell'utente;
- non è invasivo, in quanto in genere non richiede un esplicito contatto fisico con il dispositivo;
- è relativamente difficile copiare il passo di un altro utente

Motion

Questo gruppo di caratteristiche biometriche può essere visto come un'estensione della sola camminata in quanto vengono acquisiti i segnali degli utenti durante le Action Daily Living (ADL). Le ADL possono comprendere qualsiasi tipo di attività comunemente svolta durante la giornata (e.g. saltare, sdraiarsi, sedersi).

Nello sviluppo della tesi sono stati considerati dati biometrici sia della sola camminata sia delle ADL.

2.3 Inertial signal

Nella tesi sono stati usati segnali acquisiti con sensori inerziali quindi di seguito saranno descritti con maggior dettaglio.

Accelerometer

E' uno dei sensori maggiormente utilizzati nel campo del riconoscimento basato su sensori inerziali. Misura l'accelerazione relativa alla caduta libera di oggetti o persone, conosciuta anche come forza-g. La struttura interna di un accelerometro può essere vista come una massa attaccata a tre molle. Quando il sistema è sottoposto ad un'accelerazione la massa si sposta dalla posizione fissa iniziale, modificando la lunghezza delle tre molle. Il valore dell'accelerazione è calcolato misurando la differenza di lunghezza delle molle rispetto alla posizione iniziale, e come output restituisce una tripletta di valori (a_x , a_y , a_z) per ogni istante di tempo t . Nella figura 2.9 sono mostrati il sistema di assi collegato allo smartphone e alcuni valori ideali dell'accelerometro al variare della posizione. Come molti altri sensori è soggetto a leggeri problemi di calibrazione, infatti dati due accelerometri dello stesso modello possono differire nelle misurazioni effettuate.

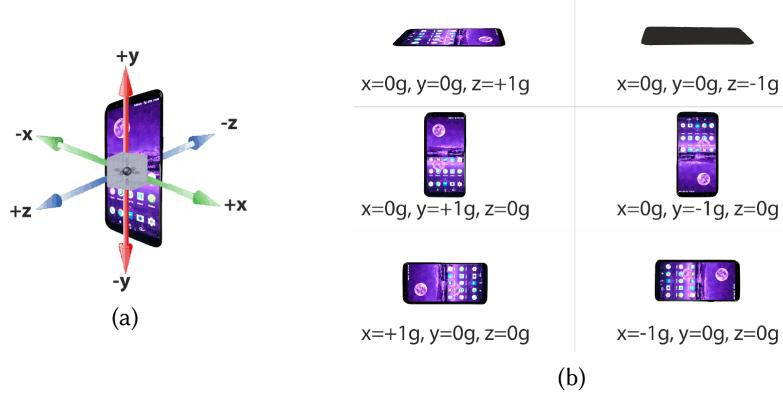


Figure 2.9: Sistema di riferimento intrinseco allo smartphone e valori ideali accelerometro [21]

Gyroscope

A differenza dell'accelerometro questo sensore si occupa di misurare l'angolo di rotazione intorno ad un sistema di 3 assi, quindi per ogni istante di tempo t si legge una tripletta (g_x, g_y, g_z) per ogni istante di tempo t , come mostrato in figura 2.10.

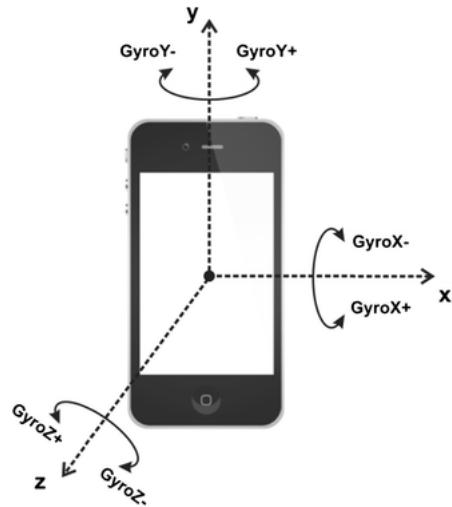
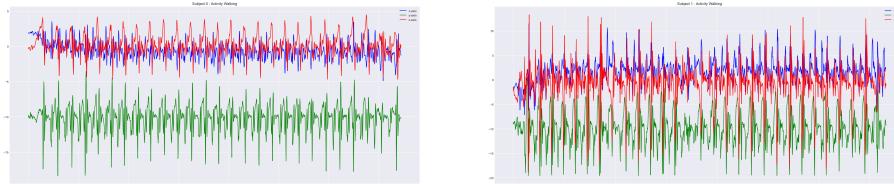


Figure 2.10: I tre assi di un giroscopio

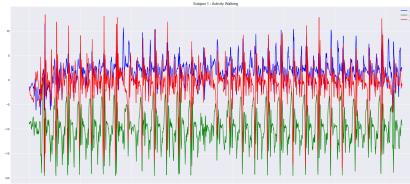
Magnetometer

Questo sensore si occupa di misurare il campo magnetico, ed è meno utilizzato nell'ambito del riconoscimento biometrico.

Walking

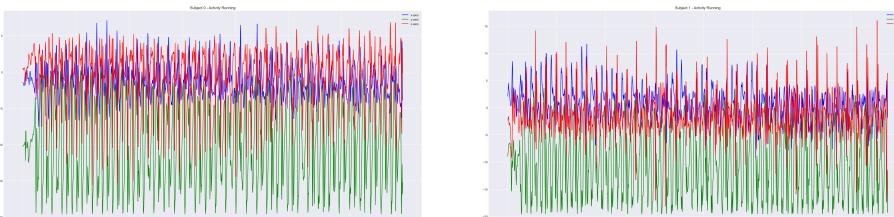


(a) Soggetto 0

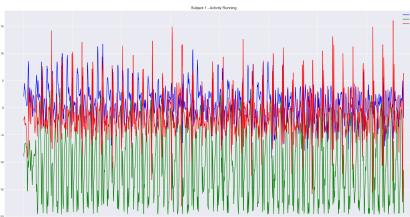


(b) Soggetto 1

Running

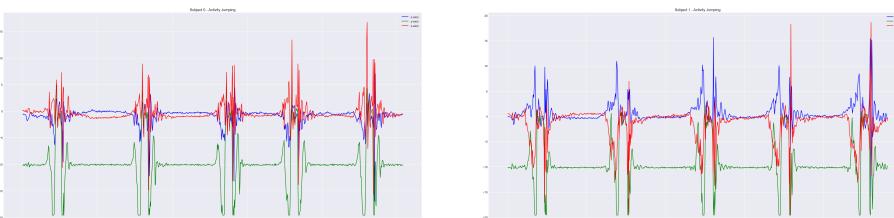


(c) Soggetto 0

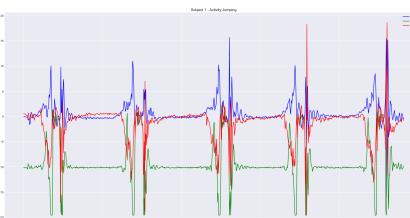


(d) Soggetto 1

Jumping

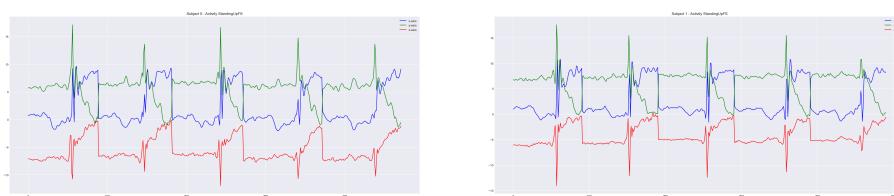


(e) Soggetto 0

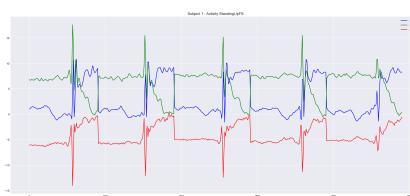


(f) Soggetto 1

Standing up from sitting



(g) Soggetto 0



(h) Soggetto 1

Figure 2.11: Esempio di segnali (accelerometro) di soggetti differenti durante le medesime azioni

2.4 Authentication system

2.4.1 Structure

I principali componenti di un sistema di autenticazione biometrico può essere riassunto nella figura 2.12.

Il sistema include due principali operazioni: *user enrollment* e *user verification*. Entrambe le operazioni richiedono l'acquisizione dei dati e l'estrazione delle features di interesse.

Riguardo la fase di enrollment si possono identificare due approcci principali:

- *Template-based*. Vengono acquisiti diversi campioni della biometria, da cui vengono estratte le features, e si crea una sorta di rappresentazione dell'utente chiamata *gallery*. La galleria di features sarà usata successivamente per eseguire dei confronti, basandosi su metriche di distanza o similarità.
- *Model-based*. Le features estratte sono usate per il train di un modello di machine-learning (un modello per ogni utente), come per esempio una One Class SVM. Il modello sarà usato in fase di inferenza per capire se le features appartengono all'utente legittimo oppure no.

Nella fase di verification, in base alla strategia di enrollment, si possono usare metriche di similarità per confrontare i dati in input, chiamati *probe*, con la gallery e, in base ad una soglia, legittimare l'utente. Nel caso di enrollment model-based si avrà uno score probabilistico e l'uso di una soglia per verificare l'utente.

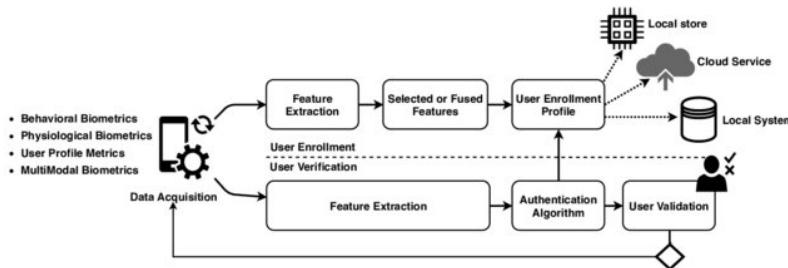


Figure 2.12: Componenti principali in un sistema di autenticazione biometrico [4]

2.4.2 Multi-biometrics system

Come rappresentato in figura 2.12 l'input del sistema di autenticazione può essere un insieme di caratteristiche biometriche. Sulla base di che cosa viene fuso, *sources of fusion*, si possono identificare 5 tipi di sistemi multi-biometrici [23]:

1. *Multi-sensor.* Rappresenta il caso in cui la stessa caratteristica biometrica viene acquisita con vari sensori, anche diversi tra di loro. Per esempio una camera e dei sensori inerziali per il movimento, oppure anche i soli diversi sensori inerziali disponibili.
2. *Multi-algorithm.* Il sistema usa più algoritmi per processare l'input, catturato da un singolo sensore, estraendo diversi tipi di features.
3. *Multi-instance.* Il sistema cattura diverse istanze dello stesso tratto biometrico. Per esempio in un sistema basato su impronte digitali può sfruttare i dati catturati da tutte e 10 le dita.
4. *Multi-sample.* Vengono catturati diversi campioni della stessa caratteristica biometrica, ma con alcune variazioni. Per esempio una foto scattata da diversi punti di vista nello stesso momento, oppure l'acquisizione del passo da sensori inerziali posizionati in punti differenti.
5. *Multi-modal.* Il sistema sfrutta diversi tratti biometrici, per esempio il passo e la frequenza del battito.

Alcuni esempi di sources of fusion sono mostrati in figura 2.13.

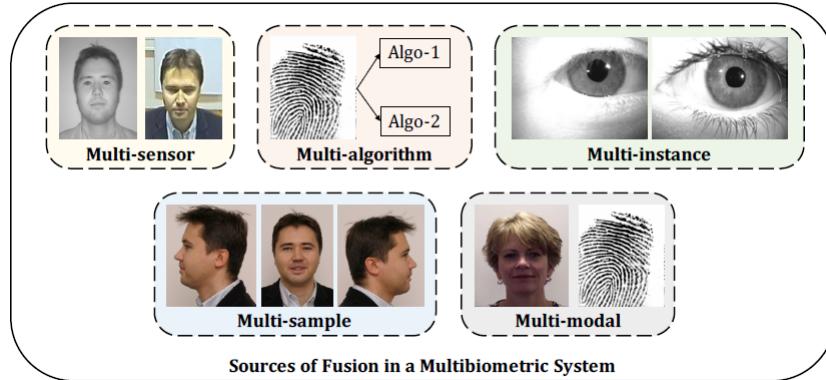


Figure 2.13: Esempi di sources of fusion [23]

Dati i sistemi biometrici appena definiti si possono ulteriormente specializzare in base al livello a cui i dati vengono fusi, ottenendo altri 5 sottogruppi:

1. *Sensor-level*. I dati sono uniti subito dopo l'acquisizione, per esempio combinando più immagini oppure concatenando segnali inerziali di più sensori.
2. *Feature-level*. In questo caso vengono combinate le features estratte dai dati.
3. *Score-level*. La combinazione avviene al livello in cui si hanno in output degli score di similarità, per esempio da diversi algoritmi applicati alla stessa feature biometrica.
4. *Rank-level*. La fusione viene effettuata dopo aver confrontato la probe con la gallery.
5. *Decision-level*. Avviene a livello di decisione, per esempio con algoritmo di voto di maggioranza tra n classificatori utilizzati.

2.4.3 Types of authentication

Sulla base di come viene gestito il confronto tra probe e gallery durante la fase di autenticazione possono essere definiti 3 tipi di operazioni [21]:

- *Verification.* Viene effettuato un match 1:1 tra la probe e i templates presenti nella gallery. Attraverso una soglia si decide se accettare o rifiutare la richiesta di autenticazione. Un esempio di verification è l'autenticazione su uno smartphone in cui implicitamente solo registrati i templates di chi possiede ed ha accesso al telefono.
- *Identification Closed Set.* Viene effettuato un match 1:N tra la probe e i template di tutte le gallery registrate (più utenti), ipotizzando che la probe appartiene sicuramente ad un utente registrato. L'identità associata alla probe è la prima in un rank basata su metriche di similarità o distanza.
- *Identification Open Set.* Come nel caso precedente viene effettuato un match 1:N tra probe e gallery, ma non è implicito il fatto che la probe appartenga ad un utente registrato.

2.5 Evaluation metrics

Per quanto riguarda le metriche di valutazione si possono dividere in due gruppi: quelle usate per valutare la qualità delle features estratte, eseguendo un task di classificazione, e quelle usate per valutare la bontà del sistema di autenticazione, eseguendo il task di verifica.

Classificazione

Vengono utilizzate quattro principali metriche:

- *Accuracy*. Indica il la percentuale di esempi associati alla classe giusta rispetto al numero totale di esempi classificati.

$$ACC = \frac{\text{number of correct classified examples}}{\text{total number of classified examples}} \quad (2.1)$$

- *Precision*. E' il rapporto tra il numero di esempi classificati correttamente positivamente rispetto a tutti gli esempi classificati positivamente (caso binario). Nel caso di più classi la definizione è estesa considerando classe per classe e unificando le misure con *macro* e/o *micro* precision.

$$PRECISION = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \quad (2.2)$$

- *Recall*. E' il rapporto tra il numero di esempi classificati correttamente positivamente rispetto a tutti gli esempi effettivamente positivi (caso binario). Nel caso di più classi la definizione è estesa considerando classe per classe e unificando le misure con *macro* e/o *micro* recall.

$$PRECISION = \frac{\text{True positive}}{\text{Truepositive} + \text{Falsenegative}} \quad (2.3)$$

- *F1-score*. E' una media pesata di recall e precision, calcolata come la media armonica tra precision e recall. E' molto più utile dell'accuracy nel caso di classi sbilanciate.

$$F1 = 2 \cdot \frac{p \cdot r}{p + r} \quad (2.4)$$

Verifica

Sono utilizzate tre metriche principalmente:

- *False acceptance rate (FAR)*. Rappresenta la frequenza con cui un utente viene autenticato come legittimo nel momento in cui non lo è. Indica la capacità del sistema di generalizzare su una larga popolazione di individui.

$$FAR = \frac{\text{number false acceptance}}{\text{number total attempts}} \quad (2.5)$$

- *False rejection rate (FRR)*. Rappresenta la frequenza in cui un utente legittimo non viene autenticato. Indica il livello di sicurezza del sistema.

$$FRR = \frac{\text{number false rejections}}{\text{number total attempts}} \quad (2.6)$$

- *Equal error rate (EER)*. È una metrica che sfrutta sia FAR e che FRR in modo da indicare con un singolo valore la bontà del sistema sia dal punto di vista della generalizzazione che della sicurezza, figura 2.14.

$$EER = FAR \approx FRR \quad (2.7)$$

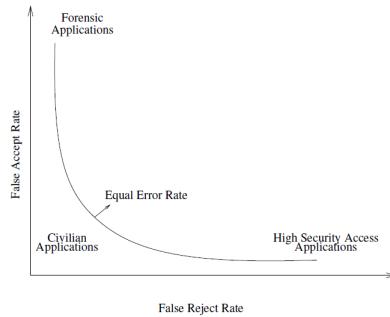


Figure 2.14: Equal error rate

Chapter 3

State of the art

Nella parte iniziale della tesi ci si è concentrati su uno studio dei metodi maggiormente utilizzati in letteratura, analizzando sia le tecniche di preprocessing sia gli algoritmi di riconoscimento e/o autenticazione maggiormente utilizzati.

3.1 Public datasets

Per poter validare la nostra soluzione è stato necessario ricercare dei dataset pubblici grazie ai quali poter confrontare i nostri risultati con quelli di altri lavori. Di seguito verranno descritti in modo dettagliato i dataset utilizzati, sia per quanto riguarda l'attività di camminata sia per le ADL.

3.1.1 Gait datasets

Dopo aver eseguito una ricerca approfondita, prendendo come punto di riferimento i datasets presentati in [21], siamo stati in grado di trovare un numero maggior di datasets ipoteticamente rilevanti nel campo dell'autenticazione basata su sensori inerziali. I datasets sono riassunti nella tabella 3.1.

Per ognuno di essi sono descritte le caratteristiche principali con le quali è possibile stabilire i pro/contro e capire per quale specifico task di ricerca possono essere utilizzati, per esempio ci sono alcuni dataset che presentano più sessioni di acquisizione dati, eseguite anche a giorni/mesi di distanza, altri che usano sensori posizionati in varie parti del corpo e alcuni che non vincolano le condizioni in cui l'utente esegue la camminata.

Basandosi sulla tabella riassuntiva 3.1 sarà ora descritto Ou-isir, il dataset usato negli esperimenti effettuati, e le motivazioni di tale scelta.

Paper	Dataset	Devices	Devices positions	Inertial sensor	# Subjects	Walk condition	Walk length	# Sessions	# Walk for Session
[24]	McGill	1 phone	pocket	acc, magn	20	normal up down	15 minutes	2 same day	1
[25]	OU-ISIR	3 IMU 1 phone	hip back waist	acc, gyro	745	normal up down	15 m (2 slopes of 3m, 1 level of 9m)	1	1
[26]	ZJU-GaitAcc	5 wii remote	left upper arm, right wrist, right hip, left thigh, right ankle	acc	175 (153 with 2 session)	normal	20 m	2 from 1 week to 6 months	6
[27]	IDNet	1 phone	right pocket	acc gyro magn	50	normal	5 minutes	from 1 to 14 6 months	-
[28]	BWR-MultiDevice	1 phone	hip	acc	25	normal	10 steps	2 15 days	2 for phone
[29]	WhuGait	1 phone	unconstrained	acc, gyro	118	unconstrained, walk and non-walk	1-2 day	-	-
[30]	GaitNet	2 phone	left waist right thigh	acc, gyro	93	normal	640 m	2	1
[31]	RecodGaitV2	1 phone	pocket or waist	acc, gyro	115	unconstrained	5 minutes	at least 2 different days	1
[32]	MMUISD	1 phone	left, right hand left, right pocket back-pack handbag	acc, gyro	299	slow normal fast	8 minutes	1	1

Table 3.1: Riassunto dei dataset pubblici riguardo l'attività di camminata.

Ou-isir^[25]

Il dataset è stato raccolto usando 4 sensori, figura 3.1, con una frequenza di campionamento di 100 Hz:

- 3 sensori IMUZ, ognuno di essi con un accelerometro ed un giroscopio, posizionati intorno alla vita del soggetto (sinistra, centro, destra);
- 1 smartphone con solo i dati dell'accelerometro, posizionato nella parte centrale della vita.

Il percorso seguito dai soggetti prevedeva 9 metri di camminata su livello piano e 3 metri di camminata con pendenza (up e down), come mostrato in figura 3.1

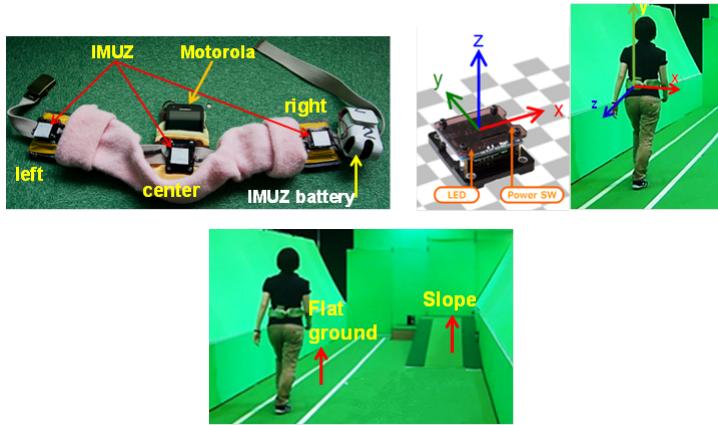


Figure 3.1: Setup dei sensori e percorso

Il dataset contiene 3 sotto-dataset ottenuti sulla base del sensore utilizzato e la parte del percorso effettuato:

- *Dataset A.* Contiene i dati di 745 soggetti acquisiti dall'IMUZ centrale. Comprendono solo la parte di percorso piano. Per ogni soggetto sono disponibili due sequenze di 4.5 metri.
- *Dataset B.* Contiene i dati di 495 soggetti raccolti con i tre IMUZ, introducendo una variazione nella posizione dei sensori, raccolti sia sul livello piano che in pendenza.
- *Dataset C.* Contiene i dati di 408 soggetti raccolti dallo smartphone sia sul livello piano che in pendenza.

Nella figura 3.2 è possibile notare come, per tutti e 3 i sotto-dataset, sia presente una buona distribuzione sia in termini di età che di rapporto maschio-femmine.

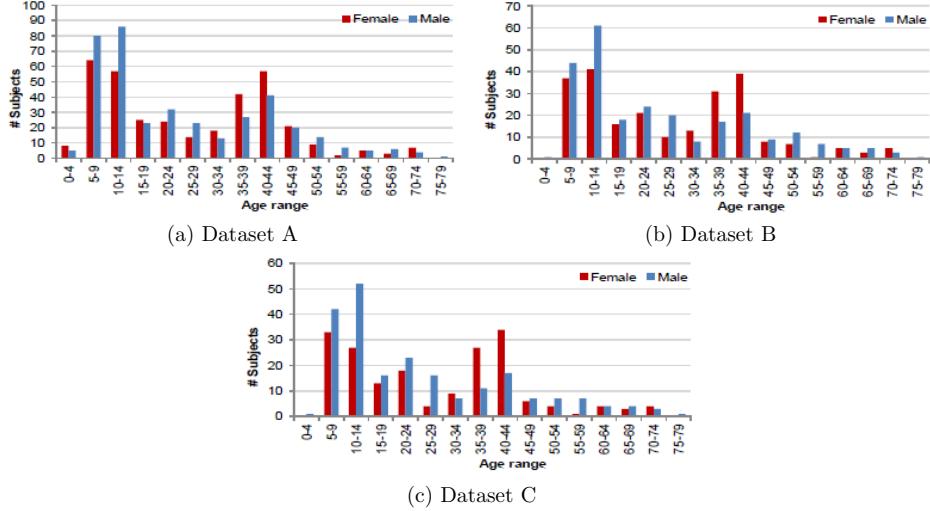


Figure 3.2: Distribuzione di età e sesso nei 3 sotto dataset disponibili

Di seguito sono elencate le motivazioni per cui è stato scelto tale dataset, nello specifico il sotto-dataset A:

- presenta il maggior numero di utenti, grazie al quale è possibile validare un sistema di autenticazione anche in termini di scalabilità rispetto al numero di utenti;
- presenta una buona distribuzione di genere ed età;
- in confronto agli altri dataset analizzati presenta una ridotta quantità di dati, al massimo 9 metri per ogni soggetto, il che rappresenta una difficoltà ulteriore nell'implementazione di un modello di Deep Learning;
- tra i dataset trovati in letteratura è uno dei maggiormente utilizzati e quindi è possibile confrontare le performance con un gran numero di lavori.

3.1.2 ADL datasets

Per quanto riguarda i datasets dedicati alle ADL non è stata effettuata un ricerca di quelli presenti in letteratura, ma sono stati scelti 3 dataset utilizzati in un lavoro recente [33], con cui poter confrontare i risultati. Di seguito sono descritte le principali caratteristiche di ognuno di essi.

Unimib-Shar^[34]

Il dataset contiene i segnali di 30 utenti, di età compresa tra 18 e 60 anni, e raggruppa sia classi di ADL che di cadute (falls). Per il nostro task andremo ad utilizzare solamente le 9 classi di ADL raccolte: Walking, Running, Jumping, Going downstairs, Going upstairs, Sitting down, Standing up, Lying down from standing, Standing up from laying.

I dati sono stati raccolti sfruttando l'accelerometro di uno smartphone, seguendo un protocollo ben definito per due sessioni: nella prima il sensore è posto nella tasca destra mentre nella seconda nella tasca sinistra.

Come riferito nel paper originali i segnali sono stati ricampionati a 50 Hz

Sbhar^[35]

Il dataset contiene i segnali di 30 utenti, di età compresa tra 19 e 48 anni. Le classi di ADL registrate sono divise in 3 gruppi:

- *Statiche*: comprendono Standing, Sitting, Lying.
- *Dinamiche*: comprendono Walking, Walking downstairs, Walking upstairs.
- *Transizione*: rappresentano le transizioni posturali che avvengono tra le azioni statiche e comprendono Stand-to-sit, Sit-to-stand, Sit-to-lie, Lie-to-sit, Stand-to-lie, Lie-to-stand.

I dati sono raccolti sfruttando l'accelerometro e il giroscopio di uno smartphone. I soggetti eseguono un protocollo per due sessioni: nella prima il sensore è fissato nel lato sinistro della vita mentre nella seconda è posizionato come preferito dall'utente.

I dati sono raccolti con una frequenza di campionamento di 50 Hz.

Realdisp^[36]

Il dataset contiene i segnali di 17 utenti raccolti mediante un protocollo che simula 33 esercizi fisici eseguiti durante un tipico warm-up. I dati sono

raccolti da un insieme di 9 IMU posizionati in vari parti del corpo: polsi, polpacci, cosce, parte superiore delle braccia e centro della schiena. Per ogni sensore sono disponibili i segnali di accelerometro, giroscopio e magnetometro ad una frequenza di campionamento di 50 Hz.

Ogni soggetto ripete più volte l'intero warm-up (2/3), andando a modificare la posizione dei sensori e fornendo 3 diversi sotto-dataset come segue:

- *Self-placement dataset*: i supervisori indicano al soggetto dove posizionare il sensore, ma non vengono date particolari istruzioni sull'orientamento.
- *Ideal-placement dataset*: i sensori sono posizionati dai supervisori, nello stesso modo (orientazione) per ogni soggetto.
- *Mutual-placement dataset*: i sensori sono intenzionalmente posizionati in modo diverso rispetto alla configurazione ideale (solo per 3 utenti).

Dataset	Subjects	Activities	Devices	Devices position	Inertial sensor
Unimib-Shar	30	9	1 phone	left right pocket	acc
Sbhar	30	12	1 phone	waist	acc gyro
Realdisp	17	33	9 IMU	pocket, calf, thigh, lower arm, upper arm, back	acc gyro magn

Table 3.2: Riassunto dataset ADL

3.2 Preprocessing techniques

Le tecniche di preprocessing utilizzate non sono specifiche rispetto al solo passo o alle altre attività ma comprendono tecniche di preprocessing dei segnali utilizzate in genere per migliorare la qualità del segnale, oppure normalizzarne l’ampiezza.

Denoising

Per migliorare la qualità del segnale sono utilizzate principalmente due tecniche: Weighted Moving Average (WMA) (usata in [37, 38, 39, 40, 41]) e Wavelet Denoising (WD) (usata in [42, 43, 44]), entrambe utilizzate per ottenere un segnale meno rumoroso. WMA attraverso una finestra di n valori scorre sul segnale originale e per ogni punto calcola una media pesata dei valori all’interno della finestra. WD decomponete il segnale in coefficienti (alti e bassi), in base alla risposta ad una funzione d’onda, successivamente è necessario scegliere quali coefficienti tenere e ricostruire il segnale; solitamente vengono annullati i coefficienti alti che rappresentano il rumore del segnale.

Time interpolation

L’interpolazione viene usata sia per avere un segnale con valori ad una distanza fissa nel tempo sia per avere segnali con un numero prefissato di valori, per esempio in molti lavori in cui vengono estratti i cicli del passo, tecnica che sarà descritta successivamente, viene usata un’interpolazione lineare per campionare i segnali (usata in [44, 29, 45, 46]). Nel lavoro [47] viene usata una interpolazione cubica invece che lineare.

In [48] invece di applicare un’interpolazione al segnale viene aggiunto un pad di zeri.

Amplitude normalization

Viene usata per ridurre il segnale in un range comune di valori. La più utilizzata è la standardizzazione:

$$v_n(i) = \frac{v_0(i) - \mu(v_0)}{\sigma(v_0)} \quad (3.1)$$

dove v_n e v_0 sono il segnale originale e quello normalizzato, μ e σ rappresentano la media e deviazione standard del segnale originale.

Other

Altre tecniche di preprocessing riguardano il modo in cui vengono utilizzati e combinati i diversi assi del segnale. Alcuni lavori calcolano la magnitudo del segnale:

$$m = \sqrt{x^2 + y^2 + z^2} \quad (3.2)$$

il vantaggio del vettore magnitudo è che risulta essere invariante rispetto all'orientazione del dispositivo ma nello stesso tempo, essendo una sorta di aggregazione, può causare la perdita di alcune caratteristiche proprie solo di alcuni assi.

In altri lavori vengono usate delle formule ad-hoc per aggregare i vari assi dei sensori, come per esempio in [49, 50]:

$$v_i = \arcsin \left(\frac{z_i}{\sqrt{x_i^2 + y_i^2 + z_i^2}} \right) \quad (3.3)$$

oppure viene usato un singolo asse, solitamente quello dominante, come in [37, 43]

3.3 Segmentation methods

Dopo aver elencato le principali tecniche di preprocessing usate con i segnali inerziali è necessario definire come un segnale può essere diviso in varie parti elementari (fragments). Ogni fragment sarà associato ad una label che indica l'identità e l'attività svolta (nel caso delle ADL). Per quanto riguarda i dataset sulle ADL la tecnica principalmente usata si basa su una sliding window, mentre i dataset con la sola attività di camminata comprendono lavori sia basati su sliding window che su gait cycle segmentation. Entrambi gli approcci sono descritti di seguito.

3.3.1 Gait/step cycle segmentation

Dato che la camminata è un'azione periodica è possibile estrarre dei pattern elementari chiamati cicli di passo. Ogni ciclo di passo corrisponde a due step come raffigurato in figura 2.8. Questo approccio ha l'obiettivo di estrarre i vari cicli di passo e usarli come esempio fondamentale che rappresenta l'utente. Esistono vari metodi per estrarre i cicli, di seguito verranno brevemente descritte le principali trovate in letteratura. I vari metodi si basano

sempre sull'uso dell'accelerometro, il quale meglio rappresenta la ciclicità del passo, e i risultati sono proiettati sugli altri sensori se presenti.

In [51, 52, 53, 54] viene usato il vettore magnitudo per individuare gli indici che indicano l'inizio e fine di un ciclo, i risultati sono poi proiettati sugli assi originali. Viene estratta la parte centrale del segnale perché considerato più stabile e confrontato con l'intero segnale (sotto parti di dimensioni simili), ottenendo un vettore delle distanze $d_{0,\dots,n}$ per ogni punto del segnale confrontato. Per stimare la lunghezza del ciclo viene effettuata la differenza del vettore delle distanze tra i minimi individuati. La media delle differenze rappresenta la stima della lunghezza del ciclo. Successivamente vengono estratti i vari cicli: partendo dalla parte centrale del segnale viene estratto un segmento di 2 volte la lunghezza stimata e preso il minimo; dal minimo ci si sposta in avanti (indietro) nel segnale aggiunge (sottraendo) l'80% del ciclo e cercando altri minimo che indicano la fine (inizio) di un nuovo ciclo. Non viene aggiunta l'esatta lunghezza stimata del ciclo perché può variare durante la camminata.

In [47] i tre assi dell'accelerometro sono ridotti di dimensioni usando PCA. Sul segnale risultante vengono identificati i minimi e il primo rappresenta l'inizio del primo ciclo. Per stimare la lunghezza del ciclo viene applicata la Fast Fourier Transformation (FFT) e calcolata la frequenza massima f_{max} ; la lunghezza stimata è $len = f_s/f_{max}$ dove f_s è la frequenza di campionamento. I cicli vengono estratti spostandosi della lunghezza stimata dal primo minimo e cercando gli altri minimi. Nel lavoro i cicli estratti sono poi sommati e mediati per ottenere un unico ciclo per utente.

In [55, 44] viene sfruttato il solo asse z per estrarre i cicli. Inizialmente vengono identificati i minimi del segnale e filtrati con una soglia basata sulla loro media e deviazione standard. Per stimare la lunghezza del ciclo viene eseguita una funzione di autocorrelazione sul segnale, e la distanza tra il primo e secondo picco del vettore risultante rappresenta la lunghezza cercata. Sulla base dei minimi e della lunghezza vengono estratti i cicli.

Nel lavoro [56] vengono estratti i singoli step e non l'intero ciclo del passo. Per applicare il metodo descritto è necessario conoscere in anticipo il numero k di step effettuati dall'utente, per questo motivo è inapplicabile per molti dataset. Prima di tutto vengono calcolati la media del segnale, la frequenza dei valori del segnale e i massimi (ordinati); sulla base di questi sono calcolati due valori, $stepThreshold$ e $stepEquilibrium$, grazie ai quali vengono estratti i vari step. Il primo rappresenta il $k - esimo$ massimo e il secondo il valore del segnale con la più alta frequenza e maggiore della media. I valori compresi tra due di questi valori rappresentano uno step.

3.3.2 Chunks of signal

Nel caso di Action of Daily Living non è possibile segmentare il segnale sulla base della periodicità perché la maggior parte delle azioni non hanno una periodicità ben definita, per questo il metodo basato su sliding window è il più utilizzato. In questo caso il segnale originale è diviso in fragmenti di lunghezza ben definita, in alcuni casi è possibile che ci sia una certa quantità di overlap tra finestre successive.

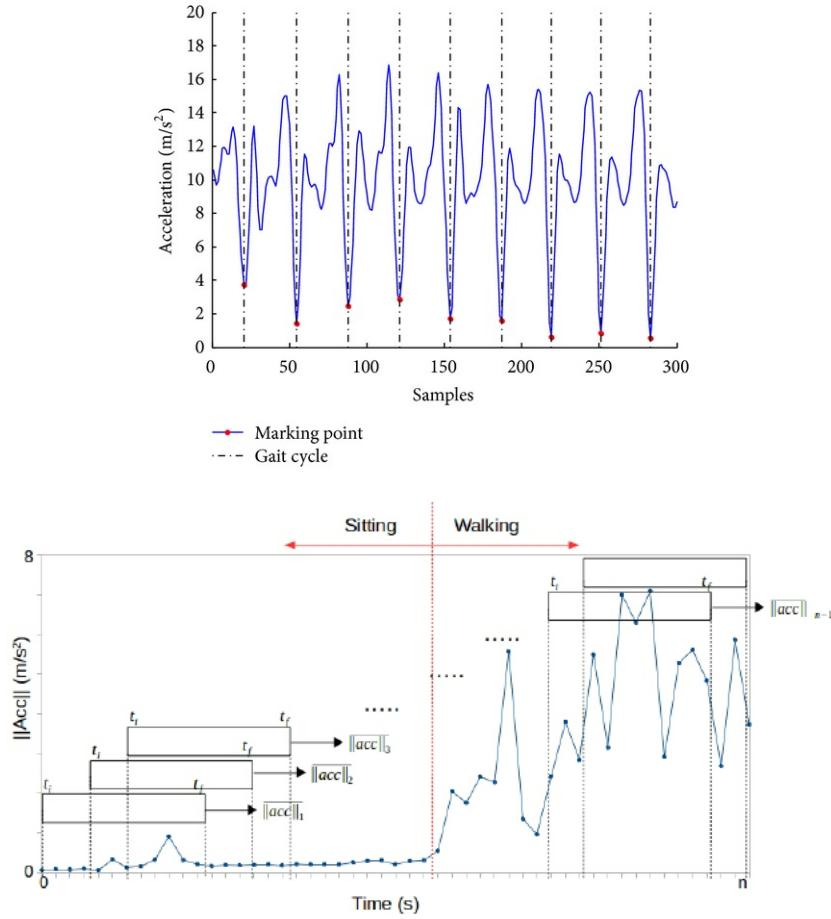


Table 3.3: Esempi di approcci basati su cicli del passo e sliding window

3.4 Recognition methods

In questa sezione saranno descritti i principali lavori trovati in letteratura, sia per il solo task legato alla camminata sia per le ADL, partendo dagli approcci più classici basati su template matching fino agli approcci moderni basati su Deep Learning.

3.4.1 Gait

In [40] viene usato un dataset composto da 25 utenti, di cui 5 effettuano la registrazione e saranno considerati utenti legittimi, e ci si basa sul solo accelerometro. Dai tre assi viene calcolato il vettore magnitudo, usato nella fase di riconoscimento, perché indipendente dall'orientazione ma comunque dipendente dalla posizione in cui viene posto lo smartphone. Nel paper vengono registrate tre differenti velocità di camminata: normale, veloce e lenta, per studiare la capacità dell'accelerometro di identificare l'identità di un utente anche con diversi tipi di camminate. Per ogni utente sono raccolti circa 30 minuti di dati per ogni tipo di camminata e viene creato un template per ogni coppia utente-camminata. Prima di estrarre i cicli il segnale viene interpolato per avere una frequenza di 150 Hz e viene applicata una WMA per rimuovere il rumore. Per la fase di matching vengono confrontate diverse metriche di distanze: Manhattan, Euclidean, Dynamic Time Warping (DTW) e infine Cross DTW metric (CDM).

Nel lavoro [57] viene affrontato il problema dell'orientazione sfruttando una tecnica di programmazione dinamica ciclica per andare a stimare la matrice di rotazione tra la probe e la gallery. Viene usato un dataset con 47 soggetti che percorrono un path di circa 90 metri in diverse sessioni e con il sensore posto in varie orientazioni.

In [47] viene affrontato il problema del riconoscimento di utenti con una certa età, infatti a causa dell'instabilità della camminata nelle persone anziane il task di riconoscimento risulta essere più difficile. Nel lavoro viene utilizzato un insieme di 64 utenti compresi tra i 50 e 79 anni, compresi nel più ampio dataset OUISIR. Per ridurre le fluttuazioni dei soggetti i cicli estratti vengono sommati e mediati. Vengono sfruttati due algoritmi di matching contemporaneamente e in modo indipendente: Pearson correlation coefficient (PCC) e Manhattan distance (MD). Se il risultato dei due algoritmi è diverso allora ne viene usato un terzo basato su normalization cross-correlation (NCC), più oneroso computazionalmente ma anche più accurato. I risultati dimostrano che l'approccio proposto permette di ottenere delle performance migliori.

In [58] viene proposta l'estrazione di una serie di features statistiche, compresi il Mel- and Bark-frequency cepstral coefficients (MFCC, BFCC), per allenare una SVM. L'approccio è testato su un dataset di 48 utenti, che effettuano due sessioni di camminate di circa 90 metri, con a disposizione i dati dell'accelerometro. I segnali vengono prima pre-processati per avere una frequenza di campionamento fissa e successivamente sono normalizzati sottraendo la media. Nel lavoro viene testato sia un approccio basato su cycle segmentation che su sliding window. Una volta ottenute le finestre o cicli, per ognuno viene estratto un vettore di features statistiche (e.g. meadia, massimo, minimo, binned distribution, ...). Vengono testati anche l'uso dei singoli assi dell'accelerometro e magnitudo, i risultati migliori si hanno usando tutti gli assi e il vettore magnitudo compreso. L'approccio ottiene dei buoni risultati usando finestre di circa 10 secondi.

Nel lavoro [59] viene proposto un approccio che non presuppone a priori la posizione del dispositivo o l'orientazione, ed è in grado di apprendere nuovi pattern dell'utente durante il tempo. Nella fase di preprocessing i dati dell'accelerometro sono divisi in fragments (circa 5 secondi), e proiettati in un sistema di riferimento globale per evitare problemi legati all'orientazione. Dopodiché viene estratto un primo insieme di features, usato per eliminare i fragments che non rappresentano l'attività di camminata; come classificatore viene utilizzato un albero di decisione. Sui rimanenti fragments vengono calcolate altre features che combinate con le prime formano il vettore di features finale. Come modello per rappresentare il passo viene utilizzato un Gaussian Mixture Model - Univariate Background Model (GMM-UBM). Durante la fase di verifica il vettore di feature viene confrontato con il modello universale creato in precedenza e con il modello dell'utente, se è più vicino a quest ultimo allora viene accettato, altrimenti l'utente può sbloccare il telefono con il PIN. Se l'utente non è stato accettato ma è legittimo il modello usa le ultime feature confrontate per modificare il modello dell'utente.

In [60] viene proposta una nuova rappresentazione del passo, invariante rispetto all'orientazione, chiamata Gait Dynamics Image (GDI). È una matrice 2D che sfrutta delle invarianti definite su accelerometro e giroscopio, basate sulle relazioni tra i valori del segnale in timestamps consecutivi. Come modello per la verifica viene sfruttato i-vector, noto nel campo del riconoscimento vocale. L'approccio viene testato su due dataset: McGill su cui ottiene 87% (same day) e 66% (cross day) di accuracy; OUISIR su cui ottiene 5.6% di EER usando sia accelerometro che giroscopio e considerando l'intera camminata di 9 metri come fragment.

Nel paper [61] viene proposta una nuova tecnica per estrarre i cicli del passo e due metodi di verifica: uno basato su Majority Voting (MV) e uno

su Cycle Rotation Metric (CRM). I dati sono normalizzati sottraendo la media. La stima della lunghezza dei cicli e la loro estrazione sono basate sui salience vector (min e max), e su parametri riguardanti altezza e distanza minima. I cicli che meno rappresentano l’utente sono eliminati operando un confronto con DTW. Nell’approccio MV solo il ciclo più rappresentativo viene tenuto come template dell’utente, viene scelto sfruttando DTW e tenendo quello con distanza minima. Per autenticare un utente viene usata come metrica DTW e se almeno il 50% dei cicli di probe sono accettati allora l’utente è verificato. Nell’approccio basato su CRM vengono confrontati due insieme di cicli (gallery e probe) prendendo a coppie ogni ciclo dei due insiemi. Basandosi su la distanza di Manhattan e DTW viene trovato il match migliore e la distanza che permette di verificare o rifiutare l’utente. L’approccio è testato su un dataset di 48 soggetti che seguono un percorso predefinito, compreso di scale e apertura di porte, e i risultati in termini di EER sono: 28% per MV e 21.7% per CRM.

Uno dei primi lavori basati su Deep Learning è quello presentato in [27]. Il lavoro è testato su un dataset di 50 utenti che contiene i segnali di accelerometro e giroscopio. Nel paper dopo aver estratto i cicli del passo viene proposto la proiezione dei dati in un sistema indipendente dall’orientazione del dispositivo. L’input della rete comprende una matrice (1 x 200 x 8) che rappresenta un vettore di 200 campioni per ogni asse dei due sensori e i due vettori magnitudo. Durante la fase di verifica la rete, precedentemente allenata, viene usata come feature extractor e le features sono classificate allenando una One-Class SVM (OCSM) per ogni utente.

In [46] viene implementato un metodo basato su CNN e testato su un dataset di 175 utenti (ZJU-GaitAcc). L’approccio si basa sull’analisi dei cicli, che in questo caso sono già etichettati nel dataset stesso. Ai cicli viene applicato un filtro passa basso di Butterworth per diminuire il rumore e sono normalizzati con interpolazione lineare. L’input della rete è una matrice 2D (15 x 118 x 1) nella quale vengono concatenati i 3 assi di ognuno dei 5 sensori disponibili, viene quindi implementato un sistema multi-biometrico multi-sample sensor-level perché lo stesso tratto biometrico viene catturato da più sensori dello stesso tipo e i dati raw sono concatenati. Per aumentare il numero di esempi disponibili viene applicata una tecnica di data augmentation sommando al segnale originale del rumore ottenuto da una distribuzione uniforme nel range $[-0.2, 0.2]$ e raddoppiando il numero di esempi per ogni utente. Negli esperimenti effettuati viene considerata solo una sessione delle due presenti e i primi 5 record sono usati come train mentre l’ultimo come test. I risultati con e senza data augmentation sono intorno al 94% e 95% in termini di accuracy.

Nel lavoro [62] viene usata una rappresentazione del ciclo di passo già vista in precedenza, la GDI, ma viene analizzata applicando una CNN. Dopo aver estratto i cicli essi sono codificati in immagini AE-GDI (Angle Embedded - Gait Dynamic Image). Ogni immagine rappresenta l'invarianza del segnale rispetto all'orientazione in quanto codifica l'angolo, nello spazio xyz del sensore, di valori a timestamp successivi. L'angolo rimane invariato anche dopo traslazione e rotazioni del sensore in quanto la posizione relativa dei valori è la medesima. L'approccio viene testato su due dataset: McGill su cui ottiene in termini di accuracy 96.6% e 67.9% (same day e cross day) e su OUISIR su cui ottiene il 61%. Le basse performance su OUISIR possono essere causate dai pochi cicli disponibili per ogni utente.

In [63] viene sperimentato la fusione delle caratteristiche temporali del segnale (dati raw) e spettrali (in termini di frequenze del segnale). Inoltre vengono usati più dispositivi di acquisizione e confrontati approcci di fusione a sensor-level e score-level. Dopo aver estratto i cicli vengono codificati in uno spazio 2D tempo-frequenze applicando una TFD (time-frequency distribution) e analizzati con una CNN.

Nel lavoro [64] viene investigato come le performance di identificazione possano migliorare se durante il train della rete vengono aggiunti altri task, nello specifico viene creata una rete multi-task: identità-età-gender. La rete viene testata sul dataset OUISIR, usando sia accelerometro che giroscopio. Una CNN viene allenata sul task di classificazione e successivamente usata come feature extractor. Nella fase di autenticazione viene usata la distanza Euclidea per confrontare probe e gallery. Dai risultati del paper sembra che aggiungendo altri task alla rete le performance sull'identificazione migliorino di circa uno 0.6% (da 94.2% a 94.8%) mentre per quanto riguarda l'autenticazione non è così infatti nel caso singolo task si ha 1.1% di EER e nel caso multi-task 1.34%. L'approccio ottiene dei buoni risultati ma si basa sull'intera camminata in fase di autenticazione.

Il lavoro presentato in [48] è l'unico che si basa sull'uso di una RNN. L'approccio viene testato sul dataset OUISIR e si basa su una preliminare estrazione dei cicli, sia su accelerometro che giroscopio. I risultati dimostrano che anche l'uso di una RNN può essere adatto per il task di autenticazione, infatti i risultati variano tra 11.48% e 7.55% di EER in base alla quantità di utenti usati per allenare la RNN. Nella fase di autenticazione la gallery e la probe sono confrontate basandosi sulla distanza Euclidea.

Il lavoro più recente riguardo il task di autenticazione basato sul solo passo è presentato in [44]. Nello specifico vengono presentate due tecniche di data augmentation specifiche per i cicli di passo, nelle quali l'obiettivo è modificare la magnitudo e lo "stretch" temporale del segnale. Da un'analisi

dei cicli si nota che le maggiori differenze intra-soggetto sono infatti legate alla magnitudo e allo stretch temporale, per esempio a causa di un piccolo cambio di velocità del passo. Il paper dimostra come le tecniche implementate migliorino di molto le performance sia sul task di identificazione che autenticazione, ottenendo 89.99% di accuracy e 4.49% di EER.

3.4.2 ADL

Nel paper [65] viene proposto un approccio basato su Random Projections per ridurre le dimensioni delle features, sulle quali viene calcolata una funzione di distribuzione di probabilità (PDF) usata come template dell’utente. Per confrontare diverse PDF si sfrutta la distanza di Jaccard. Durante la fase di autenticazione viene eseguito un modulo di action recognition, quindi è dipendente dall’attività svolta. Per testare l’approccio viene usato il dataset USC Human Activity Dataset composto da 14 soggetti, 12 attività e 5 trials per ognuna. Sono disponibili sia i segnali di accelerometro che giroscopio, da cui sono estratte 99 features statistiche.

Un altro approccio che sfrutta delle features statistiche e tecniche classiche di machine learning è quello proposto in [66]. Il dataset su cui è testato è formato da 10 utenti, 6 attività e 5 differenti smartphone in altrettante diverse posizioni. Il segnale è pre-processato con un filtro di media 1×3 e viene calcolato il vettore magnitudo. Per dividere il segnale in fragments viene applicato una sliding window di 5 secondi con overlap del 50%. Dalle varie finestre sono estratte 16 features statistiche. Per la fase di classificazione veongono testati diversi modelli: Decision Tree, K-nn e SVM. Gli esperimenti di autenticazione sono effettuati in base all’attività svolta.

Nel lavoro [67] viene investigato il solo task di identificazione, testando l’approccio su 3 dataset di ADL: HAR, PAMAP2 e MobiAct, che comprendono sia azioni dinamiche che statiche. Il segnale è pre-processato con un filtro di media per ridurre il rumore e segmentato in finestre di 10 secondi. Dalle varie finestre sono estratte delle features sia nel dominio temporale che delle frequenze. Sono testati vari modelli: SVM, Decision Tree e Random Forest. I risultati mostrano buone performance sulle attività dinamiche mentre peggiorano nel caso di attività statiche.

In [68] viene studiato il task di autenticazione su un dataset che include 18 attività eseguite da 51 soggetti. Ogni soggetto esegue per 3 minuti ognuna delle 18 attività, quindi il dataset contiene una buona quantità di dati. Sono disponibili sia i segnali di accelerometro che giroscopio. Il dataset è disponibile online presso UCI repository (WISDM). Nel paper vengono estratte delle finestre di 10 secondi sulle quali sono calcolate delle features statistiche. Per

la classificazione vengono confrontati 3 modelli: k-nn, alberi di decisione e random forest. Per la fase di autenticazione viene sfruttata una OCSVM basata sull'attività, quindi per ogni coppia utente-attività verrà allenata una OCSVM. Nel dataset sono disponibili i segnali sia di uno smartphone che di uno smartwatch e nel paper vengono confrontati le possibili fusione dei dati dei due dispositivi e per ognuno di accelerometro e giroscopio. Dai risultati si nota come la miglior combinazione sia data da accelerometro di smartphone e smartwatch, ottenendo 11.3% di EER medio sulle diverse attività e un accuracy media di 94.6%.

Il lavoro presentato in [69] è uno dei primi che indaga il task di autenticazione rispetto alle ADL usando tecniche di Deep Learning. L'approccio viene testato su un dataset non pubblico formato da 1.500 utenti circa i cui dati sono collezionati in modo assolutamente passivo senza nessuna restrizione sull'attività svolta (durante la giornata degli utenti). Non vengono registrati i dati quando il telefono è in situazione di quiete, quindi nel caso di attività statiche. I dati sono divisi in finestre di 10 secondi. Le features sono estratte usando una particolare architettura di RNN: Dense Convolutional Clockwork RNN. Le features estratte sono usate per creare un GMM-UBM per rappresentare sia il modello dell'utente che il modello universale dei diversi pattern trovati. L'approccio ottiene un accuracy di 69% nella fase di identificazione e di 18% di EER nell'autenticazione.

Nell'architettura proposta in [70] sono utilizzate in sequenza una CNN e una RNN. Vengono sfruttati sia l'accelerometro che il giroscopio ($k = 2$). I segnali di entrambi i sensori sono concatenati in una matrice $3 \times n$ e trasformati nel dominio delle frequenze applicando la trasformata di Fourier. Il segnale originale viene diviso in finestre di 5 secondi senza overlap, ad ogni finestra viene applicata la trasformata di Fourier e un certo numero T di queste finestre vengono concatenate sull'asse temporale per ottenere un tensore di input delle dimensioni $d^k \times 2f \times T$. I tensori di ogni sensore vengono prima processati in modo indipendente e parallelo da k rami di CNN, decomponendoli lungo l'asse temporale T . Dopodiché sono concatenati e il risultato è procesato da un ramo convoluzionale. L'output della CNN viene dati in input ad una GRU a due layer. I diversi output della GRU ad ogni timestamp sono unificati attraverso una media per ottenere un singolo vettore di output. Decomponendo l'input iniziale sull'asse temporale la rete è in grado sia di apprendere le features delle singole finestre sia le correlazioni tra le varie finestre. L'approccio ottiene un accuracy del 99% su un dataset composto da 9 utenti, 6 attività e 6 dispositivi.

Nel lavoro presentato in [71] viene sfruttata un autoencoder a più layer come estrattore di features. I dati dell'accelerometro sono fusi in modo da

avere un unico vettore unidimensionale come input: $a = a_{x1}, a_{y1}, a_{z1}, \dots, a_{xn}, a_{yn}, a_{zn}$. L'approccio è testato su un dataset pubblico (H-MOG) con 100 utenti durante diverse attività (reading, writing, browsing a map) e posture (sitting, walking), e un dataset non pubblico composto da 20 volontari che usano il dispositivo giornalmente. Il segnale viene diviso con una finestra di 5 secondi. I risultati, 4.5% di EER, mostrano una buona capacità dell'autoencoder di apprendere i pattern dei diversi utenti.

In [72] viene sfruttata una architettura CNN-Siamese per estrarre le features e una OCSVM per la fase di autenticazione. L'approccio è valutato sul dataset H-MOG. I segnali di accelerometro, giroscopio e magnetometro sono divisi in fragments (0.5, 1 e 2 secondi). Sui 90 utenti disponibili 60 vengono utilizzati per allenare il classificatore e 30 per testare l'autenticazione. I risultati mostrano una buona capacità della rete di apprendere i diversi pattern, ottenendo un 96% circa di accuracy e un FAR di circa 3%.

L'architettura proposta in [73] è simile a quella di [70]. Anche in questo caso ci si basa su un'architettura formata da CNN-RNN in sequenza. Nello specifico i dati dell'accelerometro e giroscopio sono processati in modo indipendente e parallelo da due rami di convoluzione. Ogni ramo è a sua volta composta da due CNN indipendenti: una che processa i dati raw dei sensori e l'altra che invece riceve in input il segnale trasformato nel dominio delle frequenze con la trasformata di Fourier. I due rami principali di convoluzione vengono poi uniti in un unico ramo di CNN. L'output viene processato da una Clockwork RNN. L'approccio è testato su due dataset: HHAR composto da 9 utenti e 6 attività, ShakeLogin composto da 17 soggetti. I segnali vengono divisi in finestre di 10 secondi. I risultati mostrano un 96% circa di accuracy per il dataset HHAR e 91% per il dataset ShakeLogin.

In [74] viene usato un approccio simile a [64], un'architettura multi-task: identità-fall. L'approccio è testato su 4 dataset: DFNAPAS, SisFall, UniMiB-Shar e ASLH. Viene usata una sliding window di 50 campioni. I risultati ottenuti sono intorno all'80% per tutti i dataset.

Un lavoro che usa solamente una RNN è quello presentato in [75]. Nello specifico vengono confrontati diversi modelli di LSTM: semplice, multilayer e bidirezionale. Il dataset usato non è pubblico e contiene i dati di 84 soggetti ai quali viene chiesto di eseguire l'applicazione di raccolta dei dati per 5 giorni di normale utilizzo dello smartphone. I dati includono screen touches e inertial sensors, in totale 5 tipi di dati (accelerometro, giroscopio, touch, magnetometro e elevazione). Dai vari esperimenti effettuati si legge che i risultati migliori sono ottenuti usando accelerometro, giroscopio e magnetometro e come architettura una LSTM bidirezionale. I risultati indicano uno 0.41% di EER.

Chapter 4

Modello

Nel corso del capitolo sarà descritto come i dati vengono pre-processati, le tecniche di data augmentation che sono state provate e il modello implementato.

4.1 Preprocessing segnale

Per quanto riguarda la fase di preprocessing del segnale non sono state applicate tecniche di denoising o interpolazione, mentre su entrambi i tipi di dataset studiati (ADL e Walking) si è scelto di usare la stessa tecnica di segmentazione basata su sliding window. La segmentazione prevede la scelta di due parametri: la lunghezza delle finestre (numero di campioni n) e la percentuale di overlap tra finestre successive. Dagli esperimenti effettuati si è notato la presenza di bias nelle performance causato dell'overlap tra le finestre nel train set e quelle nel test set. Nello specifico all'aumentare della percentuale di overlap corrispondeva un aumento dell'accuracy. Per risolvere il problema durante la segmentazione viene associato un *ID* progressivo alle finestre estratte, successivamente durante la fase di splitting dei dati in train e test le finestre di train che presentano un certo overlap con quelle di test vengono eliminate.

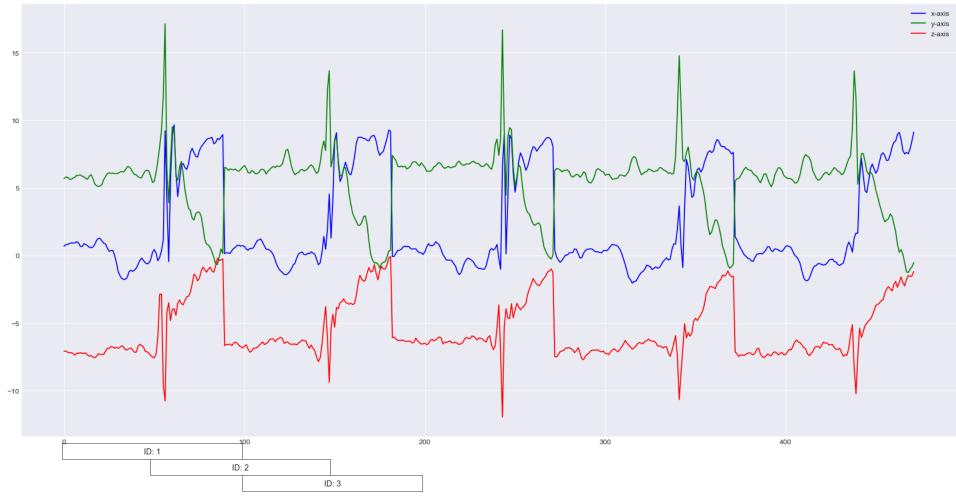


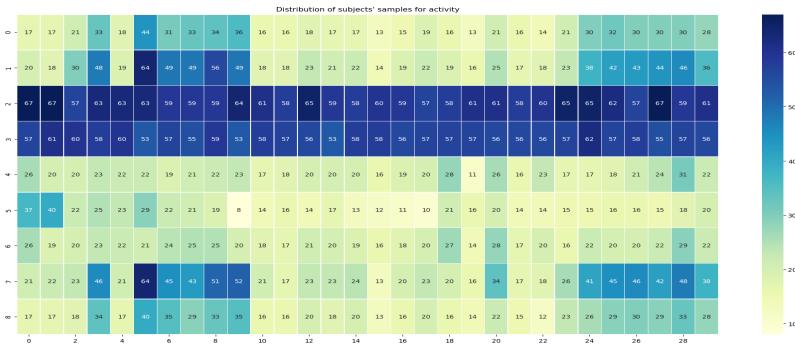
Figure 4.1: Esempio di segmentazione basato su sliding window

Per esempio basandosi sulla figura 4.1, dato un overlap del 50%, se la finestra con *ID* 2 è presente nel test set allora le finestre con *ID* 1,3 saranno eliminate dal train set.

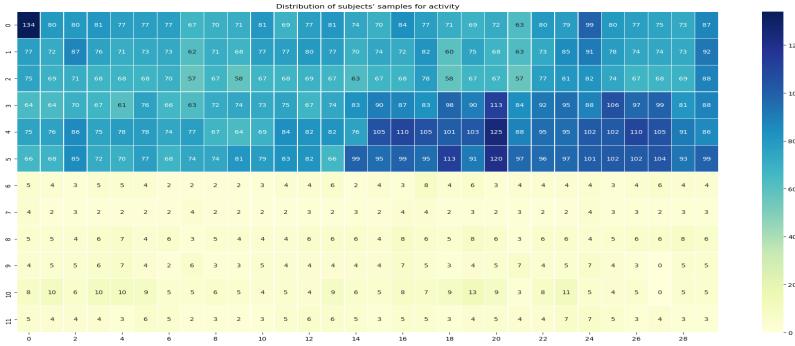
Durante la fase sperimentale saranno confrontate le performance con dimensioni di finestre e overlap differenti.

4.2 Data augmentation

Durante l'analisi dei dataset si è notato una certa sproporzione dei dati di uno stesso soggetto rispetto alle varie azioni effettuate. Come si può vedere in figura 4.2, dove sono rappresentate le distribuzione delle finestre estratte per i dataset Unimib-Shar e Sbhar, ci sono azioni con una numerosità di campioni minore rispetto ad altre. Nel caso del dataset Sbhar la sproporzione è maggiore per i segnali riguardanti le azioni di transizione.



(a) Distribuzione finestre Unimib



(b) Distribuzione finestre Sbhar

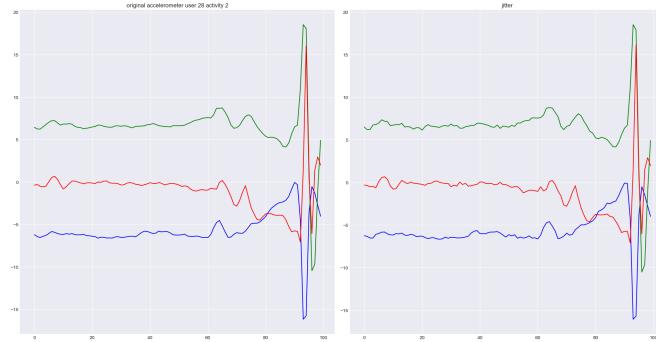
Figure 4.2: Distribuzione Unimib-Shar, Sbhar in base a soggetto e azione effettuata.

Per provare a risolvere questo problema, basandosi su lavoro effettuato in [76], sono state implementate delle tecniche di data augmentation per equilibrare i dataset. Partendo dalle seguenti 7 trasformazioni elementari: jitter, scaling, magnitude e time warping, rotation, permutation e random

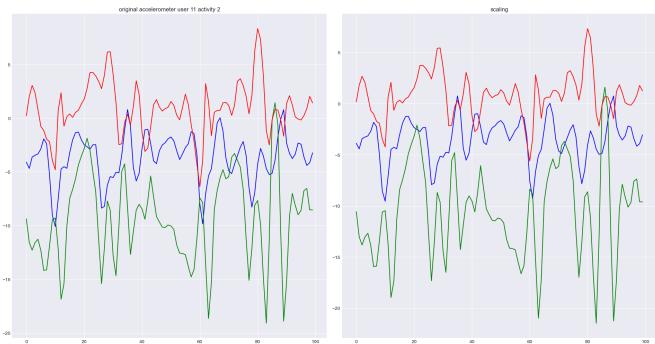
sampling, sono stati effettuati dei test per capire quale fosse la loro miglior combinazione. Di seguito sono descritte le trasformazioni elementari utilizzate:

- *jitter*: rumore addittivo con $\sigma = 0.1$;
- *scaling*: ogni asse del sensore viene scalato da un fattore moltiplicativo ottenuto da una normale con $\sigma = 0.1$;
- *magnitude warping*: convoluzione tra la finestra e una curva randomica generata da una normale con $\sigma = 0.2$ e $\mu = 1$;
- *time warping*: l'intervallo tra i campioni della finestra vengono distorti interpolando i valori con una curva randomica generata da una normale con $\sigma = 0.2$ e $\mu = 1$;
- *rotation*: rotazione random compresa tra $-\frac{\pi}{2}$ e $\frac{\pi}{2}$;
- *permutation*: la finestra viene divisa in 4 segmenti, con lunghezza minima di 20 campioni, e riordinati in modo randomico;
- *random sampling*: vengono eliminati il 10% dei valori nella finestra e il 90% rimanente viene interpolato per ottenere le dimensioni originali;

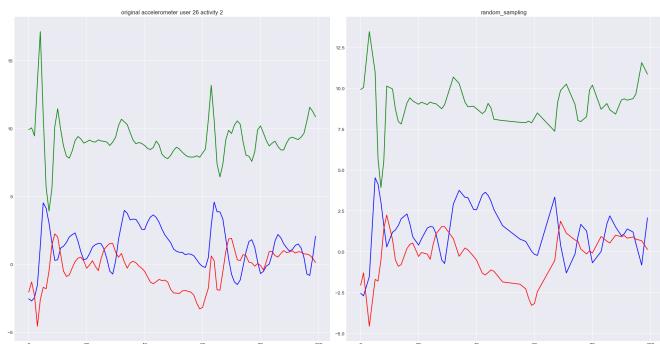
In figura 4.3 sono rappresentati alcuni esempi delle trasformazioni elementari elencate, mentre in figura 4.4 sono rappresentate alcune composizioni di queste trasformazioni. Nel capitolo 5 saranno descritti gli esperimenti effettuati in riferimento alle tecniche di data augmentation implementate con le rispettive performance ottenute.



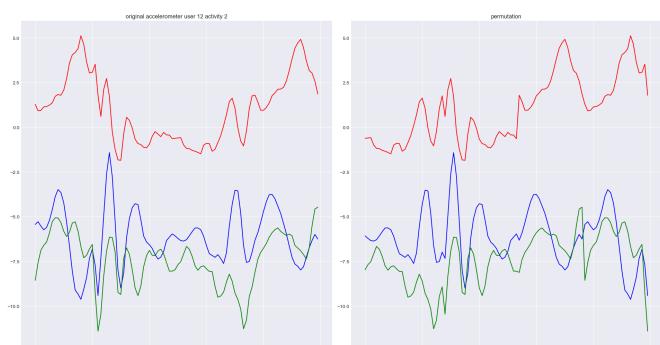
(a) Jitter



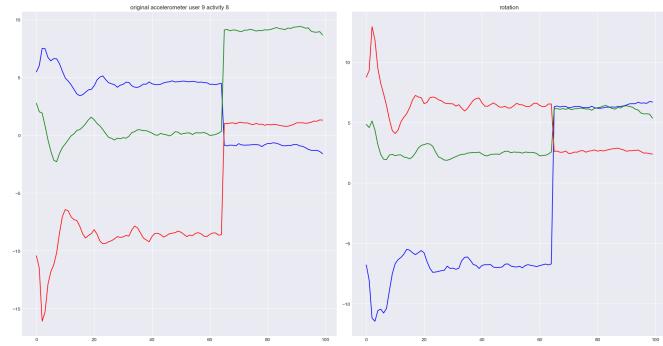
(b) Scaling



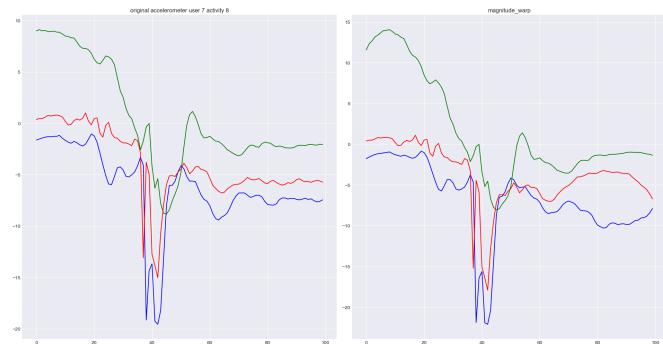
(c) Random sampling



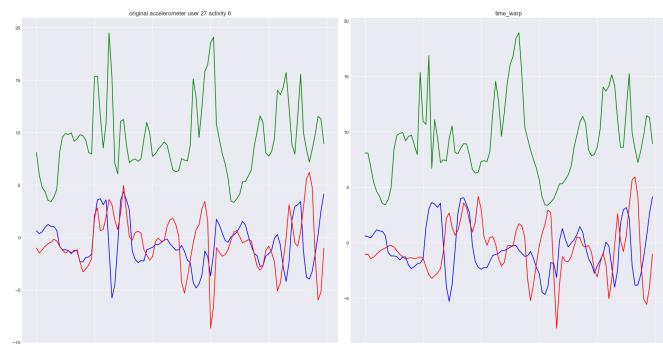
(d) Permutation



(e) Rotation

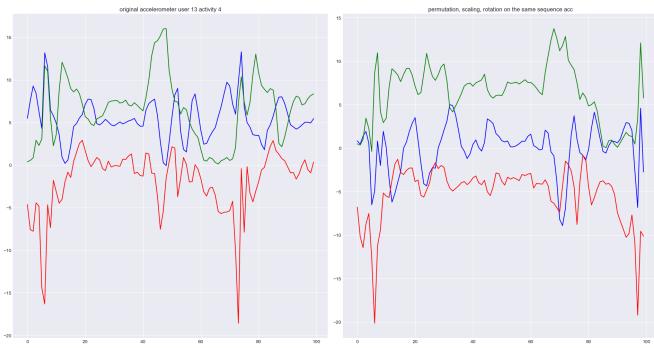


(f) Magnitude warp

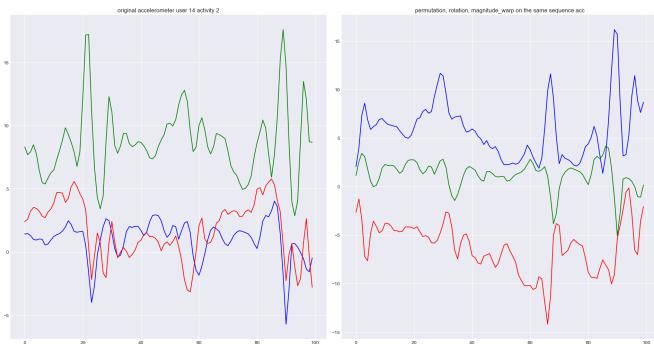


(g) Time warp

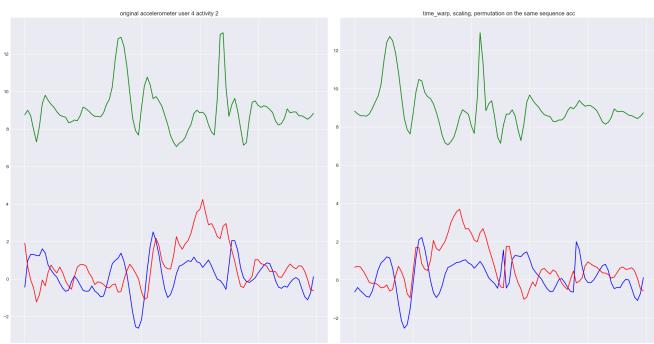
Figure 4.3: Trasformazioni elementari



(a) Permutation - Scaling - Rotation



(b) Permutation - Rotation - Magnitude warp



(c) Time warp - Scaling - Permutation

Figure 4.4: Composizione delle trasformazioni elementari

4.3 ResNet like model

Come modello di classificazione e features extraction si è scelto di ispirarsi all’architettura *ResNet* [77], nello specifico al modello ResNet18.

L’architettura ResNet permette di risolvere o alleviare il problema di Vanishing gradients per il caso di reti profonde. Il concetto di base è racchiuso nel Blocco Residuale rappresentato in figura ?? ed è la Skip Connection, il flusso identity nella figura. L’input viene processato in sequenza da: Convoluzione-Relu-Convoluzione. La Relu successiva viene applicata alla somma tra l’output ottenuto e l’input iniziale x del blocco: $H(x) = F(x) + x$. La funzione identità non ha alcun parametro ma il compito di cambiare la risoluzione spaziale dell’input, con covoluzioni di filtri 1x1, nel caso le dimensioni di x siano modificate all’interno del blocco.

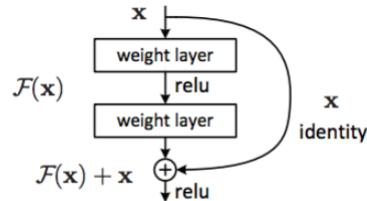


Figure 2. Residual learning: a building block.

Figure 4.5: Residual block.

Il modello implementato, rappresentato in figura 4.6, ha come input una matrice $n \times s$, dove n è la grandezza della sliding window e s è il numero di assi utilizzati. Nel caso in cui ci siano altri sensori oltre ad accelerometro il modello implementato rappresenta un sistema multi-biometrico: multi-sensor con fusione dei dati a livello di sensore (dati raw). Nell’esempio in figura viene rappresentato il caso in cui vengono utilizzati tre sensori (accelerometro, giroscopio, magnetometro) e la magnitudo per ognuno di essi. La matrice di input ha dimensioni 128×12 .

Il primo layer esegue una convoluzione 1-dimensionale applicando 32 filtri di dimensioni 5×1 lungo l’asse temporale dell’input. Usando dei filtri 1-dimensionalili il layer ha il compito di filtrare ogni dimensione dell’input separatamente, senza considerare le correlazioni tra i diversi assi e/o sensori. L’output di dimensioni $128 \times 12 \times 32$ viene compresso con un’operazione di Max Pooling lungo l’asse del tempo, con un kernel 3×1 e uno stride 2×1 .

L’output viene successivamente filtrato da due layer residuali, ognuno composto da due blocchi come rappresentato in figura 4.7. Ogni blocco

contiene a sua volta due layer di convoluzione. I blocchi del primo layer residuale applicano 32 filtri 3x3, andando a catturare le correlazioni tra i vari assi dei sensori; le dimensioni del dato rimangono invariate. Nel secondo layer residuale vengono applicati 64 filtri 3x3 e si dimezzano le dimensioni utilizzando uno stride 2x2.

L'output della CNN viene trasformato in un vettore 1×64 applicando un'operazione di Global Average Pooling.

Infine il vettore viene usato per eseguire il task di classificazione attraverso la funzione di softmax, oppure, come vettore di features durante il task di autenticazione.

Ogni layer Conv2D è composto dalla seguente tripletta di operazioni: Convoluzione-Batch normalization-Relu. Nei layer residuali solo il primo filtro di convoluzione del primo blocco ha lo stride indicato, gli altri filtri hanno stride 1x1. Le dimensioni dei filtri nei vari blocchi del medesimo residual layer hanno le stesse dimensioni.

Le dimensioni di ogni filtro e le features map in output sono descritti nella tabella 4.1.

<i>Layer name</i>	<i>Kernel size</i>	<i>Filters</i>	<i>Stride</i>	<i>Features map</i>
Conv 2D	5x1	32	1x1	124x12x32
Max pool	3x1	-	2x1	61x12x32
Residual 1	3x3	32	1x1	61x12x32
Residual 2	3x3	64	2x2	31x6x64
Global pool	-	-	-	1x64

Table 4.1: Dettagli delle dimensioni dei filtri e delle features map in output.

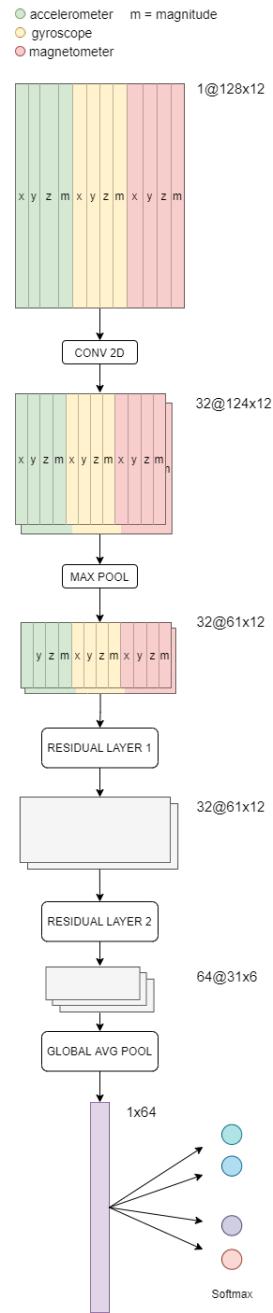


Figure 4.6: Architettura End-to-End del modello implementato e dettaglio dei residual layers.

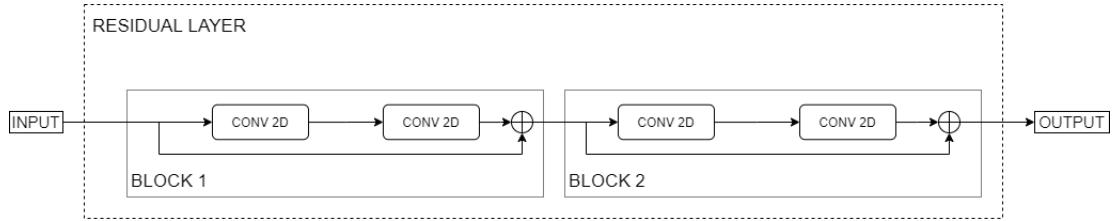


Figure 4.7: Dettaglio dei blocchi di convoluzione nei layer residuali.

Per la fase di training del modello è stato utilizzato l'ottimizzatore Adam con learning rate iniziale di 0.001 e 100 epoche di train. In base alle performance calcolate sul set di validation, se non ci sono miglioramenti per 5 epoche il learning rate viene diminuito secondo un fattore 0.5. Il modello interrompe il training nel caso in cui si raggiunge il learning rate minimo di 0.000001 o attraverso una politica di early stop.

Chapter 5

Experimental setups

In questo capitolo saranno mostrati i vari esperimenti effettuati, le relative performance e le considerazioni effettuate.

5.1 Identification

ADL

Per poter confrontare i risultati con il paper di riferimento [33] è stato necessario impostare lo stesso set up negli esperimenti. Nel paper le performance sono espresse in termini di accuracy e f1-score medie su una 10-cross fold validation, quindi per tutti gli esperimenti riguardanti l'identificazione i risultati saranno espressi nella medesima maniera. Durante la fase di 10-cross fold validation i dati sono divisi in 10 fold, in cui la distribuzione di soggetti e attività è la medesima, e a turno un fold è scelto come test. Sulla base del fold di test vengono eliminate dal train set tutte le sequenze che presentano overlap con le sequenze di test. Successivamente il train set viene diviso in 5 fold: 4 formeranno il train set e 1 il validation set. Per questo task vengono considerati tutti i soggetti all'interno dei dataset.

Per quanto riguarda il dataset REALDISP sono presenti 9 IMUZ e 3 situazioni di acquisizione: self, ideal e mutual. I dati di ogni IMU sono considerati come dataset separati e delle 3 situazioni di acquisizione vengono considerate solo la self e l'ideal, la mutual è esclusa perchè presente solo per 3 soggetti su 17. Nel caso ideal sono stati eliminati i soggetti 6 e 13 in quanto non presentano dati disponibili. Le performance relative a REALDISP saranno presentate sia in termini di dataset singoli sia come media dei risultati sui singoli dataset.

Il primo test effettuato ha l'obiettivo di confrontare il nostro modello con quello proposto nel paper di riferimento, per questo verranno usate la medesima sliding window di 100 campioni (2 secondi) e percentuale di overlap del 50%. In tabella 5.1 sono mostrati i risultati ottenuti. Le performance per il dataset Realdisp sono le medie dei risultati ottenuti sulle singole posizioni dei sensori, mostrati nella tabella 5.2.

Model	UNIMIB		SBHAR		REALDISP IDEAL		REALDISP SELF		REALDISP IDEAL SELF	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
Metier	0.857	0.865	0.876	0.875	-	-	-	-	0.960	0.960
Our	0,872	0,879	0,882	0,882	0,990	0,990	0,993	0,993	0,991	0,991

Table 5.1: Confronto modello Metier e nostro apporoccio.

Positions	IDEAL		SELF		IDEAL SELF	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
RT	0,995	0,996	0,994	0,994	0,997	0,997
LT	0,996	0,996	0,995	0,995	0,995	0,995
RLA	0,980	0,980	0,987	0,987	0,983	0,983
LLA	0,982	0,982	0,989	0,989	0,983	0,983
RUA	0,980	0,981	0,990	0,990	0,987	0,987
LLUA	0,989	0,989	0,989	0,989	0,989	0,986
RC	0,998	0,998	0,998	0,998	0,997	0,997
LC	0,998	0,997	0,998	0,998	0,996	0,996
BACK	0,995	0,995	0,995	0,995	0,994	0,994

Table 5.2: Performance sulle varie posizioni di Realdisp, set up secondo Metier.

Come si può vedere i risultati su Realdisp sono buoni per tutte le posizioni dei sensori, le uniche che hanno un leggero degrado solo RLA, LLA, RUA, LUA che indicano la parte superiore e inferiore del braccio, il motivo può essere l'introduzione di oscillazioni a causa del movimento del braccio.

Dopo aver confrontato i due modelli si sono testate differenti percentuali di overlap tra finestre successive ed eseguendo gli esperimenti si è notata la presenza del bias tra le sequenze nel train e nel test. Con un overlap del 90% le performance sia su Unimib che su Sbahr raggiungevano il 99% circa. Per questo motivo a seconda della percentuale di overlap scelta si andranno a eliminare dal train set le finestre ad una certa distanza, che aumenta con l'aumentare dell'overlap. Per capire quale fosse la percentuale di overlap migliore sono stati usati solo i dataset Unimib e Sbhar, i risultati

sono mostrati in tabella 5.3.

Overlap	UNIMIB		SBHAR	
	Accuracy	F1-score	Accuracy	F1-score
50%	0,872	0,879	0,883	0,883
60%	0,870	0,872	0,883	0,883
70%	0,859	0,866	0,881	0,881
80%	0,855	0,864	0,877	0,876
90%	0,767	0,773	0,801	0,799

Table 5.3: Performance in base alle diverse percentuali di overlap.

Dai risultati emerge come la percentuale migliore sia il 50%, per questo sarà usata in tutti gli esperimenti successivi.

Come esperimento successivo si è scelto di indagare come l'aggiunta di sensori potesse migliorare le performance, per questo nei dataset dove possibile sono stati effettuati dei test usando diverse aggregazione dei sensori disponibili. In tabella 5.4 sono rappresentati i risultati. Per il dataset Realdisp si è scelto solo una posizione di test (RT), in quanto le performance sono simili tra le diverse posizioni.

Sensors	UNIMIB		SBHAR		REALDISP IDEAL RT	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
Accelerometer	0,872	0,879	0,858	0,857	0,961	0,962
Accelerometer Gyroscope	-	-	0,882	0,882	0,980	0,980
Accelerometer Gyroscope Magnetometer	-	-	-	-	0,995	0,996

Table 5.4: Performance sin base all'uso di diversi sensori.

Come si può vedere dai risultati ottenuti, sui dataset utilizzati, l'uso di più sensori permette di ottenere delle performance migliori in quanto catturano aspetti differenti della stessa azione.

Prendendo spunto dai paper raccolti nello stato dell'arte si è deciso di aggiungere ai dati anche il vettore magnitudo calcolato per ogni sensore disponibile e studiarne le performance. Nella tabella 5.5 sono mostrati i risultati con una sliding window di 100 campioni, 50% di overlap e l'aggiunta del vettore magnitudo. L'input con il vettore magnitudo è una matrice del tipo ($acc_x, acc_y, acc_z, acc_{magn}, gyro_x, gyro_y, gyro_z, gyro_{magn}$).

Come si può vedere l'aggiunta del vettore magnitudo permette un au-

Model	UNIMIB		SBHAR		REALDISP IDEAL		REALDISP SELF		REALDISP IDEAL SELF	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
Metier	0,857	0,865	0,876	0,875	-	-	-	-	0,960	0,960
Our	0,872	0,879	0,882	0,882	0,990	0,990	0,993	0,993	0,991	0,991
Our.magnitude	0,873	0,879	0,896	0,895	0,992	0,992	0,995	0,995	0,993	0,993

Table 5.5: Risultati aggiungendo il vettore magnitude.

mento delle performance per tutti i datasets, in particolare nel dataset Sbhar con un incremento del 1,5% circa. Di seguito sono mostrati i risultati sui singoli dataset di Realdisp.

Positions	IDEAL		SELF		IDEAL SELF	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
RT	0,996	0,996	0,997	0,997	0,996	0,996
LT	0,996	0,996	0,998	0,997	0,997	0,996
RLA	0,985	0,986	0,992	0,992	0,987	0,987
LLA	0,981	0,981	0,991	0,991	0,988	0,988
RUA	0,990	0,990	0,991	0,991	0,990	0,990
LUA	0,989	0,989	0,996	0,995	0,991	0,991
RC	0,998	0,998	0,999	0,999	0,998	0,998
LC	0,996	0,995	0,998	0,998	0,997	0,997
BACK	0,996	0,995	0,994	0,994	0,996	0,996

Table 5.6: Performance sulle varie posizioni di Realdisp aggiungendo il vettore magnitude.

Un ulteriore modifica rispetto al paper di partenza consiste nella lunghezza della finestra, si è scelto di testare una finestra di 128 campioni (2.56 secondi) come nei paper originali dei dataset che si occupano di Activity Recognition. In tabella 5.7 sono mostrati i risultati.

Model	UNIMIB		SBHAR		REALDISP IDEAL		REALDISP SELF		REALDISP IDEAL SELF	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
Metier	0,857	0,865	0,876	0,875	-	-	-	-	0,960	0,960
Our	0,872	0,879	0,882	0,882	0,990	0,990	0,993	0,993	0,991	0,991
Our.magnitude	0,873	0,879	0,896	0,895	0,992	0,992	0,995	0,995	0,990	0,993
Our.magnitude.128	0,891	0,896	0,899	0,898	0,986	0,986	0,990	0,990	0,994	0,994

Table 5.7: Risultati aggiungendo il vettore magnitude.

A questo punto mantenendo un set up così definito, 128 campioni per finestra, 50% di overlap e vettore magnitudo, si è deciso di testare alcune tecniche di data augmentation per provare a migliorare le performance. Aumentando gli esempi disponibili si è cercato di equilibrarne la distribuzione all'interno dei dataset. Date le performance attuali gli esperimenti di data augmentation sono stati effettuati solo sui dataset Unimib e Sbhar.

Per capire quali trasformazioni elementari applicare ai dati sono stati eseguiti diversi esperimenti con differenti insiemi di trasformazioni, inoltre si sono testate anche composizioni di più trasformazioni elementari sullo stesso segnale. I risultati dei vari esperimenti sono mostrati nella tabella 5.8, le varie sigle sono da interpretare nel seguente modo: J (jitter), S (scaling), RND (random sampling), MW (magnitude warp), TW (time warp), P (permutation), R (rotation), NC (le trasformazioni non vengono composte in sequenza sul segnale), C (le trasformazioni sono composte in sequenza sul segnale), OC (vengono tenute solo i segnali finali dopo la composizione di trasformazioni). Gli esperimenti per capire quale fosse la miglior combinazione di trasformazioni da usare sono stati effettuati solo sul dataset Unimib.

Trasformazioni	Composizione	Accuracy	F1-score
J, S, RND MW, TW, P, R	NC	0,887	0,893
	C	0,873	0,878
	OC	0,867	0,875
J, S, RND MW, TW, P, R	NC	0,902	0,901
	C	0,904	0,910
	OC	0,906	0,911
MW, TW, P	NC	0,906	0,912
	C	0,897	0,902
	OC	0,902	0,909
J, S, RND	NC	0,900	0,908
	C	0,894	0,900
	OC	0,875	0,880

Table 5.8: Esperimenti su data augmentation.

Come riportato in tabella i risultati migliori si hanno usando le trasformazioni Jitter, Scaling, Random sampling, Permutation, Time warp e Magnitude warp, nel caso OC quindi tenendo solo la finestra finale dopo la serie di trasformazioni. Negli esperimenti effettuati vengono applicate 3 funzioni random tra quelle disponibili.

Di seguito è riportata una tabella finale che riassume tutti gli esperimenti effettuati per quanto riguarda il task di identificazione sui dataset ADL.

Model	UNIMIB		SBHAR		REALDISP IDEAL		REALDISP SELF		REALDISP IDEAL SELF	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
Metier	0.857	0.865	0.876	0.875	-	-	-	-	0.960	0.960
Our	0.872	0.879	0.882	0.882	0.990	0.990	0.993	0.993	0.991	0.991
Our.magnitude	0.873	0.879	0.896	0.895	0.992	0.992	0.995	0.995	0.990	0.993
Our.magnitude.128	0.891	0.896	0.900	0.898	0.986	0.986	0.990	0.990	0.994	0.994
Our.magnitude.100 data aug	0.891	0.898	0.858	0.857	-	-	-	-	-	-
Our.magnitude.128 data aug	0.906	0.911	0.859	0.859	-	-	-	-	-	-

Table 5.9: Riassunto dei test effettuati sul task di identificazione.

Come si vede dalla tabella 5.9 le tecniche di data augmentation implementate permettono di migliorare le performance sul dataset Unimib, raggiungendo un accuracy del 90%, mentre per il dataset Sbhar peggiorano le performance. Il motivo di tale comportamento potrebbe essere dato dalla distribuzione troppo squilibrata di esempi per attività presenti nel dataset.

Nelle figure 5.1 e 5.2 sono mostrate le performance di identificazione basate sulle attività svolte per i dataset Unimib-Shar e Sbhar. Per Realdisp, date le performance, non sono state calcolate le accuracy per attività.

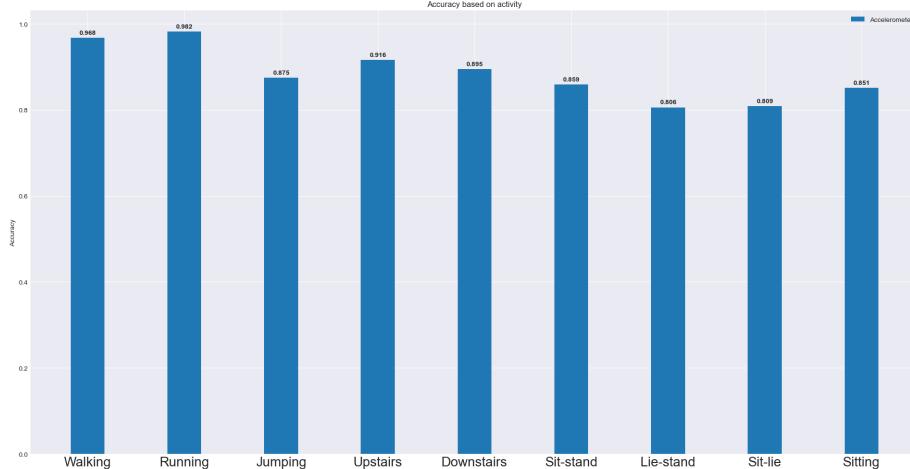


Figure 5.1: Accuracy sul task di identificazione in base all’attività svolta, dataset Unimib-Shar.

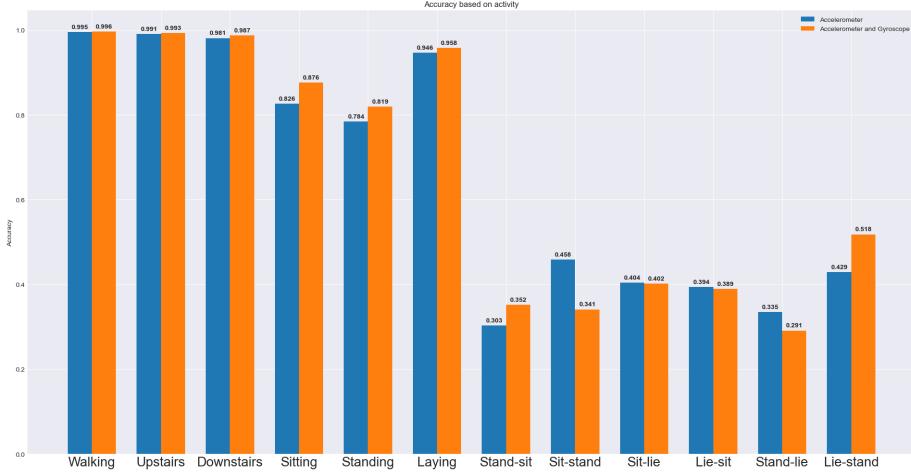


Figure 5.2: Accuracy sul task di identificazione in base all’attività svolta, dataset Sbhar, con differenza rispetto ai sensori utilizzati.

Walk

Per OUISIR è stato usato il dataset con più soggetti disponibili (745), il quale viene diviso in train, validation e test secondo una classica proporzione 70%, 20%, 10%. Anche per questo dataset sono state testati vari aspetti: dimensioni sliding window, percentuale di overlap, l’uso del solo accelerometro oppure di entrambi i sensori e infine alcune tecniche di data augmentation. Il vettore magnitude viene utilizzato in tutti gli esperimenti. In tabella 5.10 sono rappresentati i test effettuati su sliding window, overlap e utilizzo di uno o più sensori.

Sensors	Window Length	Overlap	
		50%	75%
Accelerometer	50	0,71	0,71
	75	0,80	0,79
	100	0,84	0,83
	128	0,85	0,80
Accelerometer Gyroscope	50	0,80	0,79
	75	0,84	0,84
	100	0,85	0,86
	128	0,86	0,84

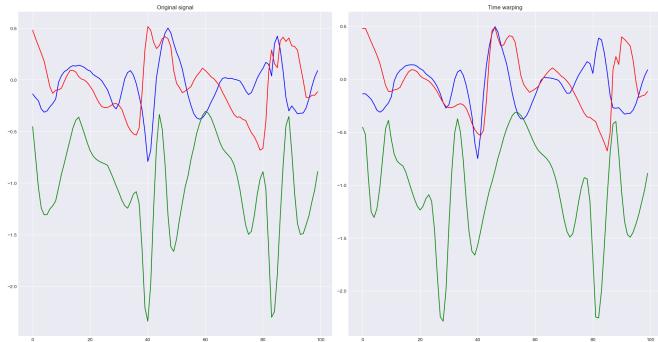
Table 5.10: Test effettuati su OUISIR.

Per quanto riguarda data augmentation si è preso ispirazione dal paper [44], nel quale vengono implementate tecniche di data augmentation specifiche per i cicli riguardo la magnitudo del segnale e l'intervallo temporale tra i campioni, quindi tra le trasformazioni elementari implementate precedentemente verranno considerate solamente la Time e Magnitude warping. Ad ogni finestra nel train set vengono applicate entrambe le trasformazioni e in più la loro composizione, ottenendo per ogni finestra originale 3 nuovi esempi. In tabella 5.11 sono presentati i risultati ottenuti.

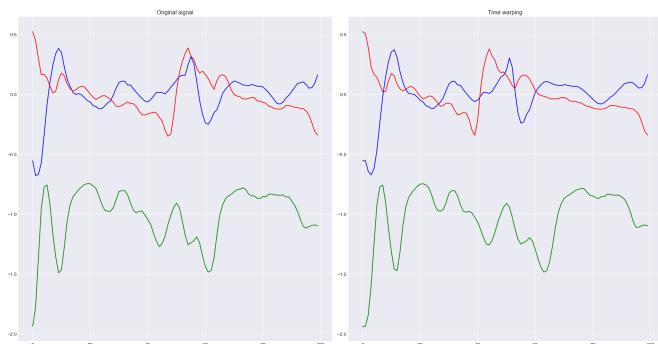
Sensors	Window Length	Overlap	
		50%	75%
Accelerometer	50	0,72	0,71
	75	0,80	0,79
	100	0,85	0,83
	128	0,83	0,84
Accelerometer Gyroscope	50	0,83	0,82
	75	0,87	0,87
	100	0,89	0,89
	128	0,88	0,86

Table 5.11: Data augmentation su OUISIR.

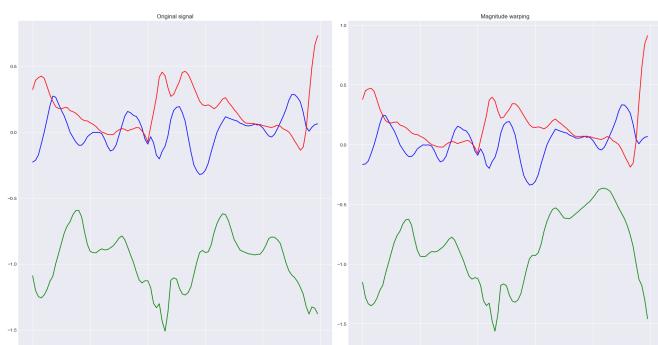
Come mostrato in tabella le tecniche di data augmentation non migliorano di molto le performance nel caso del solo accelerometro, al contrario nel caso di accelerometro e giroscopio l'accuracy aumenta di circa il 3-4% nel caso con 100 campioni e 50%-75% di overlap. La seguente figura 5.3 mostra alcuni esempi di segnali originali e trasformati con le tecniche descritte.



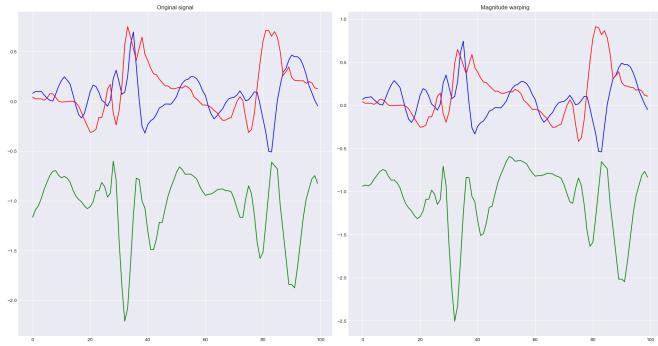
(a) Time warping



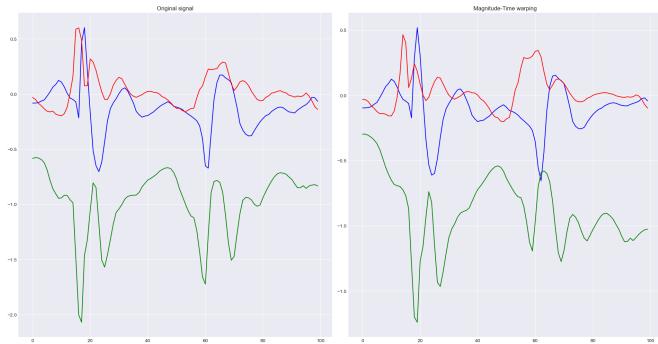
(b) Time warping



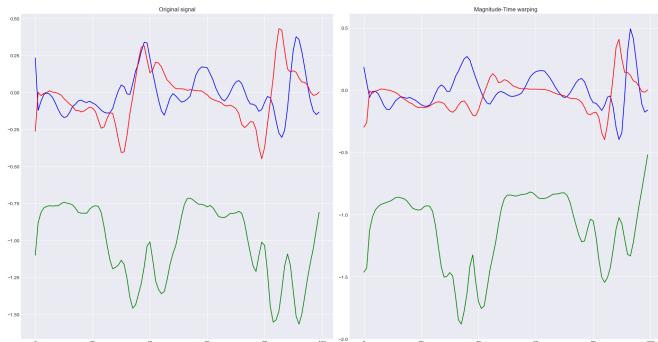
(c) Magnitude warping



(d) Magnitude warping



(e) Time-Magnitude warping



(f) Time-Magnitude warping

Figure 5.3: Esempi di trasformazioni applicate al dataset OUISIR.

5.2 Authentication

Per quanto riguarda il task di autenticazione l'approccio seguito è quello mostrato in figura 5.4. Il dataset viene diviso in due set in base agli utenti: il 70% di essi sarà usato per trainare il classificatore, mentre sul restante 30% sarà testata l'autenticazione. Ai due insieme di dati viene applicata la medesima fase di preprocessing, con la divisione in finestre secondo un certo overlap.

Nella fase di train del classificatore i dati sono standardizzati per avere media 0 deviazione standard 1 sulla base del train set. Il valori di media e deviazione standard usati vengono salvati e saranno successivamente utilizzati sui dati nella fase di features extraction.

Il modello, usato successivamente come features extractor, viene trainato con gli stessi iper-parametri usati nel task di identificazione, ed una divisione dei dati in 90% train e 10% validation. Terminato il training del modello vengono salvati i pesi e usati per inizializzare un nuovo modello, privo di layer di classificazione, che termina con il layer di global average pooling. I dati degli utenti di autenticazione, dopo essere stati preprocesati e standardizzati, vengono trasformati in vettori di 64 features usando il modello precedentemente allenato.

Alle features in output viene applicata una normalizzazione l2:

$$x_{norm} = \frac{x}{\sqrt{\sum_{i=1}^n x_i^2}} \quad (5.1)$$

Ottenute le features finali è necessario dividerle in due gruppi: gallery e probe. La gallery è composta da quelle features che sappiamo appartenere ad un determinato soggetto, possono essere interpretate come i dati raccolti durante la registrazione di un utente. La probe invece è formata da un insieme di features la cui identità è sconosciuta, possono appartenere allo stesso soggetto oppure no, e servono per stabilire quanto un sistema di autenticazione sia efficace e sicuro. Per i dataset scelti sono state implementate due strategie per creare la gallery e la probe:

- *extra-session*: nel caso ci siano più sessioni i dati della prima vengono usati per costruire la gallery e i dati delle altre come probe;
- *intra-session*: nel caso ci sia solo una sessione allora viene ordinata temporalmente e la prima metà viene usata come gallery mentre la seconda come probe.

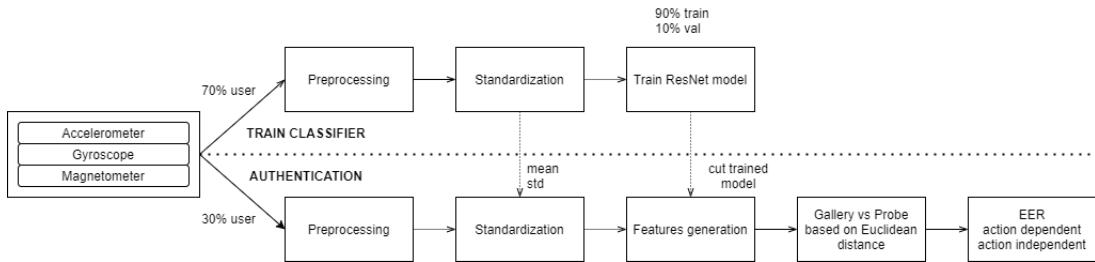


Figure 5.4: Pipeline seguita per task di autenticazione

Una volta definite gallery e probe è necessario confrontare le features. Nel nostro caso si è deciso di usare un soft-classifier basato su distanza Euclidea. Ogni feature della gallery viene confrontata con ogni feature della probe, ottenendo un vettore delle distanze. Il vettore delle distanze serve per calcolare l'Equal Error Rate, in base alla soglia scelta: data la distanza d e la soglia θ se $d > \theta$ allora le due features non sono riconosciute come appartenenti allo stesso soggetto, se invece $d \leq \theta$ le due features sono riconosciute come appartenenti allo stesso soggetto.

ADL

Per i dataset ADL viene calcolato l'Equal Error Rate sia per quanto riguarda le azioni specifiche sia in modo indipendente dall'azione svolta. Sui tre dataset è stata applicata la medesima strategia di pre-processing: 128 campioni per finestra e 50% di overlap.

Per quanto riguarda il dataset Unimib i risultati sono presentati nella tabella 5.12. Dei 30 utenti, 21 sono usati per il train del classificatore e 9 per testare l'autenticazione. La divisione si basa anche sul gender degli utenti. E' stato testato sia il caso intra-session che quello extra-session. Per il caso intra-session le performance migliori si hanno nel caso dell'attività di walking, mentre le peggiori sono nel caso di azioni di transizione. Nel caso extra-session le performance sono peggiori in tutte le attività in quanto le due sessioni sono acquisite con differenti posizioni dello smartphone (tasca destra e sinistra).

Per il dataset Sbhar si è deciso di escludere dal task di autenticazione le azioni di transizione in quanto i dati disponibili non sono sufficienti, alcuni utenti non presentano nemmeno una finestra disponibile, e si sono tenute le 6 azioni presentate nella tabella 5.13. Dei 30 utenti, 21 sono usati per il train del classificatore e 9 per testare l'autenticazione. Anche in questo caso si sono testati sia il caso intra-session che extra-session. Come per il

dataset Unimib le performance nel caso extra-session degradano a causa del diverso posizionamento dello smartphone durante le sessioni di acquisizione. In confronto al dataset Unimib si possono notare dei miglioramenti nel caso delle azioni: Walking upstairs e downstairs. Il miglioramento può essere attribuito alla presenza del giroscopio, assente un Unimib. Le performance peggiori si hanno nelle azioni statiche: Sitting, Standing e Laying, probabilmente perché il segnale è molto simile tra i vari utenti e non presenta particolari componenti (per lo più è piatto).

Activities	Session 0		Session 1		Extra session	
	Action dependent					
Walking	10,3	32,8	8,1	30,5	27,5	39,5
Running	11,1		15,5		26,4	
GoingUpS	11,9		10,1		28,9	
GoingDownS	11,7		13,5		33,8	
Jumping	12,1		14,0		32,9	
StandingUpFS	15,3		17,5		42,6	
StandingUpFL	23,7		23,5		47,1	
LyingDownFS	26,1		28,6		48,6	
SittingDown	19,2		20,1		42,0	
Mean all activities	15,7		16,8		36,6	

Table 5.12: EER su dataset Unimib

Activities	Session 0		Session 1		Extra session	
	Action dependent					
Walking	2,7	47,2	2,4	44,9	14,6	45,7
Walking upstairs	4,0		6,4		16,8	
Walking downstairs	8,0		7,7		23,0	
Sitting	48,1		38,1		45,9	
Standing	25,0		36,3		40,4	
Laying	24,0		28,5		26,6	
Mean all activities	18,6		19,9		27,9	

Table 5.13: EER su dataset Shar

Activities	LT	RT	LLA	RLA	LUA	RUA	LC	RC	BACK	Mean on dataset
Walking	6,1	7,5	18,4	11,1	12,4	11,4	1	2	3,3	8,1
Jogging	3,2	6,9	7	7	17,2	3,2	2,5	0,4	7,3	6,0
Running	0,7	4,9	9,5	7,1	13	2,9	1,8	0,2	5,6	5,0
Jump up	9,2	3,5	7,4	3,6	21,3	3,6	1,8	6	10,1	7,3
Jump front-back	1,7	7	9,7	7,5	15,5	9,4	11,4	1,7	15,3	8,8
Jump sideways	4,2	6,9	9,8	14,7	11	10,4	5,7	3,6	11	8,5
Jump open-closed	16,2	6,9	3,8	2,5	8	7,5	5,2	6,9	8,7	7,3
Jump rope	8,4	7,5	9,7	2,1	17,5	7,7	10,3	4,9	14,2	9,1
Trunk twist 1	7	4,9	7,1	5,6	3,9	1,3	16,8	10,4	6,6	7,0
Trunk twist 2	5,2	4,6	6,4	3,9	3,2	1	10,7	5,2	0,7	4,5
Waist bends forward	8,7	2,1	3,7	3,1	17,2	7	1,5	0,5	3,5	5,2
Waist rotation	4,6	1,8	5,2	9,7	0,6	2,9	2,9	1	3,4	3,5
Waist bend	10,3	10,1	12,9	11,4	8,3	10,7	9,8	1,1	8,7	9,2
Reach heels	13,8	6,8	9,5	6,2	17,3	10,3	7	0,2	12,1	9,2
Lateral bend 1	7,2	6	38,1	28,7	33,5	34,5	1,8	4,7	19,2	19,3
Lateral bend 2	22,4	11,2	50,9	55,8	41,9	45,3	10,4	4,4	40,4	31,4
Repetitive stretch	10,8	4,2	1,1	7,5	10,3	4,8	3,1	5,4	7,3	6,0
Upper trunk	4,1	14	2,2	10,8	2,5	4,2	1,8	7,8	4	5,7
Arms lateral	0,4	1,6	4,6	4,3	3,9	1,4	1,3	0,4	3,9	2,4
Arms frontal	0,2	0,8	7,5	7,2	3,2	1,7	0,6	0,4	1,9	2,6
Frontal hand	0,9	1,1	6,9	2,4	2,7	3,4	0,3	1,4	3,1	2,4
Arms crossing	1,3	1,6	2,5	1,4	2,1	2,5	1,1	5,4	6	2,6
Shoulders rotation 1	1,6	1,6	1,6	8,4	0,8	1,7	10,8	3,4	3,9	3,7
Shoulders rotation 2	2,2	13,2	3,5	8,5	14,7	2,1	16	2,1	1,1	7,0
Arms rotation	0,5	9,2	1,8	5,9	1,3	2,3	7,3	8,5	1,7	4,2
Knees to breast	3,8	16	9,6	5,1	20,1	14,5	3,6	3,6	3,8	8,9
Heels to backside	10,7	3,3	8	3,5	3,5	7,1	4,5	30,3	8,1	8,7
Knees bending	4	16,2	6,4	12,2	11,8	8	0,6	0,1	19,5	8,7
Knees bend forward	14,4	10,5	16,6	15,1	14,7	27,5	8,9	6,4	16,6	14,5
Rotation on knees	13,2	10,5	9,8	7,7	14,7	4,8	6,3	4,2	6,4	8,6
Rowing	3,3	8	7,8	6,2	5,2	20,7	4,9	5,8	18,3	8,9
Elliptic bike	0,4	0,3	16,4	7	0,3	8,1	0,3	0,4	2,5	3,9
Cycling	0,1	0,1	9,3	8,2	0,3	5,9	0,05	1,2	0,5	2,8
Mean on all activity	6,085	6,388	9,839	9,133	10,724	8,782	5,214	4,242	8,445	7,6

Table 5.14: EER su dataset Reldisp, action dependent, caso Ideal.

Activities	LT	RT	LLA	RLA	LUA	RUA	LC	RC	BACK	Mean
Walking	8,7	2,8	10,9	17,6	7	16,9	13,1	5,8	1	9,3
Jogging	4,8	0,7	9,4	18,8	7,1	13,2	1	0,2	0,1	6,1
Running	8,5	0,8	7	13,4	1,8	12,8	3	0,3	1	5,4
Jump up	15,5	5,1	2,5	10,8	11,3	2,3	6,6	2,9	5,3	6,9
Jump front-back	3,8	8,3	10,6	9,9	19,3	18	9	4,4	0,6	9,3
Jump sideways	6,4	7,6	9,7	12,5	19,7	14,2	8	7,5	2,8	9,8
Jump open-closed	8,8	6,7	7,4	10,9	5	4,5	5,6	3,1	6,2	6,4
Jump rope	8,1	1	14,8	13,7	6,1	24,7	9,1	4,3	8,6	10,0
Trunk twist 1	22,2	20,6	14,6	6,3	6,2	13,7	8,1	9,3	7,6	12,0
Trunk twist 2	10,1	11,7	3,7	8,7	7,1	18,1	8,9	5,1	3,6	8,5
Waist bends forward	6,2	14,9	9,1	2,8	22,1	13,1	7,1	4,5	0,8	8,9
Waist rotation	8,6	15,4	5,5	4,1	7,6	6,3	7,4	5,6	0,5	6,7
Waist bend	13,4	17,8	10,4	6,5	14,8	9,2	0,1	7,6	3,7	9,2
Reach heels	10,7	5,2	9,9	7	8,5	5,1	5,8	5,8	0,6	6,5
Lateral bend 1	5	20,6	19,2	16,3	30,4	49	5,7	3,9	6,9	17,4
Lateral bend 2	20,1	31,8	53,7	58	53,9	61,1	24,4	18,4	36,3	39,7
Repetitive stretch	10,4	12	8,1	5,7	5,2	8,5	0,8	2,9	1,8	6,1
Upper trunk	9,7	8,9	6,9	11,9	10,1	9	11,1	12,3	0,8	8,9
Arms lateral	4	4,6	7,8	6	3,9	9,9	9	0,1	2,3	5,2
Arms frontal	2,9	11,3	8,6	8,2	16,7	15,8	0,5	0,8	0,4	7,2
Frontal hand	2,3	0,5	3,1	10,2	5,5	12,4	0,6	0,3	2,5	4,1
Arms crossing	0,1	4	7,7	7,2	3,5	6	0,2	0,4	1,3	3,3
Shoulders rotation 1	1,6	1,2	2,8	6,7	5,6	9,3	0,8	2	0,7	3,4
Shoulders rotation 2	1,5	3,4	0,6	3,6	1,8	5,8	2	2,5	3,5	2,7
Arms rotation	1,7	2,1	11,5	11	5	4,6	0,7	0,7	10,4	5,3
Knees to breast	9,5	2,3	7,5	11,1	19	9,5	3,8	6,8	22,1	10,1
Heels to backside	4,9	10,8	12,2	10,8	4,1	7,5	4,9	10,8	23,3	9,9
Knees bending	4,1	5,4	10,8	18,1	15,4	11,3	2,4	7	1,7	8,4
Knees bend forward	38,4	33,8	5,1	15,6	32,9	8,6	0,7	37,1	1,8	19,3
Rotation on knees	7	4,3	7,6	0,7	38,5	0	0	6	4,3	7,6
Rowing	0,01	0,01	0,03	0,002	0,002	0	0	0	0	0,01
Elliptic bike	0,4	0,3	8,2	27,1	0,7	0,3	0,4	4,8	0,8	4,7
Cycling	0,1	0,3	0,2	0,8	6,8	4,4	0,2	0,4	0,2	1,4
Mean on all activity	7,864	8,370	9,307	11,273	12,200	12,276	4,879	5,564	4,955	8,5

Table 5.15: EER su dataset Reldisp, action dependent, caso Self.

	LT	RT	LLA	RLA	LUA	RUA	LC	RC	BACK
Ideal	40,4	39,4	40,9	44	41,9	42,1	36,7	39,8	27,9
Self	42,9	44,5	38,6	41,4	33,2	46	24,1	40,9	23,2

Table 5.16: EER action independent su dataset Realdisp.

Nelle tabelle 5.14, 5.15 e 5.16 sono mostrati i risultati dell'autenticazione sul dataset Realdisp, nel caso Ideal e nel caso più realistico Self. Dato il gran numero di attività, i pochi utenti considerati (solo 5) e la posizione più o meno fissa dei sensori, non è possibile effettuare una analisi approfondita delle azioni migliori. Dai risultati sul caso Self è possibile notare che in generale le performance migliori si hanno nelle posizioni LC, RC, BACK, un motivo potrebbe essere la presenza di meno fluttuazioni nel segnale rispetto alle posizioni legate alle braccia. Per quanto riguarda il caso action independent si ottengono buoni risultati nel caso Back con un EER di 27,9% e 23% nel caso Ideal e Self.

Walk

Per il dataset OUSIR dei 745 utenti, 522 sono utilizzati per il train del classificatore e 223 per testare l'autenticazione; gli utenti sono divisi anche in base al gender. Come Per la divisione gallery/probe si scelto di utilizzare la sequenza 0 come gallery e la sequenza 1 come probe, come fatto nella maggior parte dei lavori trovati in letteratura. Sulla base delle performance ottenute nel task di identificazione si è scelto di testare finestre di lunghezza 100, 128, con overlap del 50% e 75%, inoltre dato l'aumento di accuracy applicando data augmentation si è testata anche in fase di autenticazione. I risultati sono mostrati nella seguente tabella 5.17.

Sensors	Window Length	No data augmentation		Data augmentation	
		Overlap		Overlap	
		50%	75%	50%	75%
Accelerometer	100	10,08	10,30	10,10	9,8
	128	9,20	8,90	8,70	8,9
Accelerometer Gyroscope	100	9,80	9,10	8,40	8,50
	128	8,9	9,3	7,5	7,3

Table 5.17: EER su dataset OUISIR.

Nonostante le poche finestre disponibili per ogni utente, il gran numero di questi permette di creare un buon modello in grado di generalizzare su un gran numero di utenti. I risultati mostrano infatti un 7,3% di EER su 223 utenti, nel caso in cui vengono utilizzati sia accelerometro che giroscopio e vengano applicate alcune tecniche di data augmentation.

Chapter 6

Risultati

Nel capitolo saranno riassunti i risultati ottenuti, confrontandoli con lo stato dell'arte, e verranno elencati alcuni possibili sviluppi futuri.

6.1 ADL

Per i dataset ADL la tabella 6.1 evidenzia come il modello implementato sia migliore in termini di Accuracy e F1-score rispetto ad alcuni modelli dello stato dell'arte.

Study	SBHAR		UniMib-shar		REALDISP	
DeepSense	0.814	0.812	0.659	0.668	0.952	0.950
UANN	0.775	0.772	0.686	0.695	0.952	0.950
kNN18	0.770	0.767	0.674	0.684	0.950	0.946
IDNet	0.599	0.603	0.590	0.611	0.909	0.901
DCNN	0.470	0.512	0.505	0.522	0.798	0.759
METIER [33]	0.876	0.875	0.857	0.865	0.960	0.960
ResNet (our)	0.90	0.898	0.891	0.896	0.994	0.994

Table 6.1: ADL, confronto con lo stato dell'arte per il task di identificazione

Per il task di autenticazione non sono stati trovati lavori in letteratura, al massimo della nostra conoscenza, per operare un confronto, sono quindi riportati solamente i nostri risultati, tabelle 6.2, 6.3 e 6.4. Per il dataset Realdisp sono mostrati solo i risultati riguardo il caso action-independent, per i risultati rispetto alle 33 azioni si veda 5.14 e 5.15.

Activity	Session 0		Session 1		Extra session	
	Action dependent					
Walking	10,3	32,8	8,1	30,5	27,5	39,5
Running	11,1		15,5		26,4	
GoingUpS	11,9		10,1		28,9	
GoingDownS	11,7		13,5		33,8	
Jumping	12,1		14,0		32,9	
StandingUpFS	15,3		17,5		42,6	
StandingUpFL	23,7		23,5		47,1	
LyingDownFS	26,1		28,6		48,6	
SittingDown	19,2		20,1		42,0	
Mean all activities	15,7		16,8		36,6	

Table 6.2: EER su dataset Unimib

Activity	Session 0		Session 1		Extra session	
	Action dependent					
Walking	2,7	47,2	2,4	44,9	14,6	45,7
Walking upstairs	4,0		6,4		16,8	
Walking downstairs	8,0		7,7		23,0	
Sitting	48,1		38,1		45,9	
Standing	25,0		36,3		40,4	
Laying	24,0		28,5		26,6	
Mean all activities	18,6		19,9		27,9	

Table 6.3: EER su dataset Shar

	LT	RT	LLA	RLA	LUA	RUA	LC	RC	BACK
Ideal	40,4	39,4	40,9	44	41,9	42,1	36,7	39,8	27,9
Self	42,9	44,5	38,6	41,4	33,2	46	24,1	40,9	23,2

Table 6.4: EER action independent su dataset Realdisp.

6.2 Walking

Riguardo il dataset OUISIR sono stati ritrovati in letteratura molti approssimi, con poca coerenza nell'uso dei dati (preprocessing, divisione utenti per identification e authentication) e quindi un vero e proprio confronto non è semplice. Nonostante questo problema i risultati sono in linea con lo stato dell'arte, in alcuni casi migliori, come si può vedere in tabella 6.5. Per esempio un possibile confronto può essere fatto con il lavoro mostrato in [78], nel quale la porzione di dati usati (100 campioni) è simile al lavoro presentato nella tesi e i risultati sono migliori.

Study	Training portion	Testing portion	Identification	Verification
[60]	50 % dataset	9m long	-	5.6 %
[79]	-	1.4s	-	10.07 %
[80]	50 % dataset	3m long	83.8 %	-
[78]	75 % dataset	100 signals	96.84 %	10.43 %
[45]	50 % dataset	1 cycle	-	> 6 %
[64]	50 % dataset	9m long	94.8 %	1.1 %
[48]	80 % dataset	1 cycle	-	7.55 %
[44]	8 cycles/user	1 cycle	89.99 %	4.49 %
ResNet (our)	70 % of users	100-128 signals	85-89 %	7.3-8.7 %

Table 6.5: Risultati autenticazione OUISIR

Chapter 7

Conclusioni e sviluppi futuri

L’obiettivo della tesi era quello di studiare e implementare modelli di Deep Learning per i task di identificazione e autenticazione, basandosi sui dati inerziali. Sono stati affrontati entrambi i task su due categorie di dataset differenti: dataset dedicati al task di autenticazione con la sola attività di camminata e dataset creati per Activity Recognition nei quali sono presenti più azioni svolte dall’utente. Dopo aver analizzato i principali approcci in letterature si è scelto un preprocessing basato su sliding window. Come modello ci si è ispirati all’architettura ResNet, creando un’architettura poco onerosa ma che ottiene buoni risultati come mostrato nel capitolo 5. Durante le sperimentazioni effettuate sono state implementate alcune tecniche di data augmentation. L’uso di queste permette un aumento delle performance in alcuni casi (Unimib-Shar, Ouisir) mentre su altri dataset si nota un degrado dei risultati, per questo sarà sicuramente oggetto di indagini future. Nel complesso l’approccio implementato è in grado di riconoscere (identificazione) gli utenti con buone performance, indipendentemente dall’attività svolta. Le attività sulle quali il modello mostra i risultati minori sono quelle definite statiche, dove il segnale presenta poche caratteristiche distintive. Per il task di autenticazione le attività migliori, in base ai risultati ottenuti, risultano essere la camminata e la corsa. In fase di autenticazione è stato anche sperimentato un approccio action-independent, in cui il segnale viene confrontato in modo indipendente dall’azione svolta. Come ci si aspettava le performance degrado di molto, anche se i risultati di alcuni test (Realdisp-Self-Ideal-BACK) incoraggiano un ulteriore approfondimento e possibile miglioramento.

Dai risultati ottenuti i possibili sviluppi futuri possono essere i seguenti:

- ricerca di tecniche di preprocssing che permettano una rappresentazione invariante rispetto la posizione del sensore;
- approfondimento e implementazione di nuove tecniche di data augmentation;
- studio di modelli RNN, da usare in solo oppure in sequenza alla CNN implementata;
- date le basse performance nei casi di autenticazione action-independent, studio dell'integrazione di framework di activity recognition per migliorare le performance.

Bibliography

- [1] C. Wang, Y. Wang, Y. Chen, H. Liu, and J. Liu, “User authentication on mobile devices: Approaches, threats and trends,” *Computer Networks*, vol. 170, p. 107118, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128618312799>
- [2] L. O’Gorman, “Comparing passwords, tokens, and biometrics for user authentication,” *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2021–2040, 2003.
- [3] R. Clarke, “Human identification in information systems: Management challenges and public policy issues’ info. technology & people 7, 4 (december 1994),” 1994.
- [4] M. Abuhamad, A. Abusnaina, D. H. Nyang, and D. Mohaisen, “Sensor-based continuous authentication of smartphones’ users using behavioral biometrics: A contemporary survey,” *IEEE Internet of Things Journal*, 2020.
- [5] C. Tsikos, “Capacitive fingerprint sensor,” Oct. 5 1982, uS Patent 4,353,056.
- [6] J. K. Schneider, “Ultrasonic fingerprint sensors,” in *Advances in Biometrics*. Springer, 2008, pp. 63–74.
- [7] F. Liu, Q. Zhao, and D. Zhang, “3d fingerprint authentication,” in *Advanced Fingerprint Recognition: From 3D Shape to Ridge Detail*. Springer, 2020, pp. 33–57.
- [8] C.-C. Han, H.-L. Cheng, C.-L. Lin, and K.-C. Fan, “Personal authentication using palm-print features,” *Pattern Recognition*, vol. 36, no. 2, pp. 371 – 381, 2003, biometrics. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0031320302000377>

- [9] A. K. Jain, A. A. Ross, and K. Nandakumar, *Introduction to biometrics*. Springer Science & Business Media, 2011.
- [10] S. Yadav and V. P. Vishwakarma, “Extended interval type-ii and kernel based sparse representation method for face recognition,” *Expert Systems with Applications*, vol. 116, pp. 265–274, 2019.
- [11] K. Brigham and B. V. K. V. Kumar, “Subject identification from electroencephalogram (eeg) signals during imagined speech,” in *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2010, pp. 1–8.
- [12] P. S. Teh, A. B. J. Teoh, and S. Yue, “A survey of keystroke dynamics biometrics,” *The Scientific World Journal*, vol. 2013, 2013.
- [13] D. Cilia and F. Inguanez, “Multi-model authentication using keystroke dynamics for smartphones,” in *2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*. IEEE, 2018, pp. 1–6.
- [14] J.-S. Wu, W.-C. Lin, C.-T. Lin, and T.-E. Wei, “Smartphone continuous authentication based on keystroke and gesture profiling,” in *2015 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2015, pp. 191–197.
- [15] C. Shen, Y. Zhang, X. Guan, and R. A. Maxion, “Performance analysis of touch-interaction behavior for active smartphone authentication,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 498–513, 2015.
- [16] H. Khan, A. Atwater, and U. Hengartner, “Itus: an implicit authentication framework for android,” in *Proceedings of the 20th annual international conference on Mobile computing and networking*, 2014, pp. 507–518.
- [17] H. Khan and U. Hengartner, “Towards application-centric implicit authentication on smartphones,” in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, 2014, pp. 1–6.
- [18] C. L. Vaughan, B. L. Davis, J. C. O’connor *et al.*, *Dynamics of human gait*. Human Kinetics Publishers, 1992.
- [19] W. Pirker and R. Katzenbach, “Gait disorders in adults and the elderly: A clinical guide,” *Wiener klinische Wochenschrift*, vol. 129, 10 2016.

- [20] M. P. Murray, A. B. Drought, and R. C. Kory, “Walking patterns of normal men,” *JBJS*, vol. 46, no. 2, pp. 335–360, 1964.
- [21] M. D. Marsico and A. Mecca, “A survey on gait recognition via wearable sensors,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 4, pp. 1–39, 2019.
- [22] Z. Xue, D. Ming, W. Song, B. Wan, and S. Jin, “Infrared gait recognition based on wavelet transform and support vector machine,” *Pattern recognition*, vol. 43, no. 8, pp. 2904–2910, 2010.
- [23] M. Singh, R. Singh, and A. Ross, “A comprehensive overview of biometric fusion,” *Information Fusion*, vol. 52, pp. 187–205, 2019.
- [24] J. Frank, S. Mannor, J. Pineau, and D. Precup, “Time series analysis using geometric template matching,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 3, pp. 740–754, 2012.
- [25] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, “The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication,” *Pattern Recognition*, vol. 47, no. 1, pp. 228–237, 2014.
- [26] Y. Zhang, G. Pan, K. Jia, M. Lu, Y. Wang, and Z. Wu, “Accelerometer-based gait recognition by sparse representation of signature points with clusters,” *IEEE transactions on cybernetics*, vol. 45, no. 9, pp. 1864–1875, 2014.
- [27] M. Gadaleta and M. Rossi, “Idnet: Smartphone-based gait recognition with convolutional neural networks,” *Pattern Recognition*, vol. 74, pp. 25–37, 2018.
- [28] M. De Marsico, D. De Pasquale, and A. Mecca, “Embedded accelerometer signal normalization for cross-device gait recognition,” in *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, 2016, pp. 1–5.
- [29] Q. Zou, Y. Wang, Q. Wang, Y. Zhao, and Q. Li, “Deep learning-based gait recognition using smartphones in the wild,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3197–3212, 2020.
- [30] A. Vajdi, M. R. Zaghdan, S. Farahmand, E. Rastegar, K. Maroofi, S. Jia, M. Pomplun, N. Haspel, and A. Bayat, “Human gait database for

- normal walk collected by smart phone accelerometer,” *arXiv preprint arXiv:1905.03109*, 2019.
- [31] G. Santos, P. H. Pisani, R. Leyva, C.-T. Li, T. Tavares, and A. Rocha, “Manifold learning for user profiling and identity verification using motion sensors,” *Pattern Recognition*, vol. 106, p. 107408, 2020.
 - [32] J. Permatasari, T. Connie, and O. T. Song, “The mmuisd gait database and performance evaluation compared to public inertial sensor gait databases,” in *Computational Science and Technology*. Springer, 2020, pp. 189–198.
 - [33] L. Chen, Y. Zhang, and L. Peng, “Metier: A deep multi-task learning based activity and user recognition model using wearable sensors,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, pp. 1–18, 2020.
 - [34] D. Micucci, M. Mobilio, and P. Napoletano, “Unimib shar: A dataset for human activity recognition using acceleration data from smartphones,” *Applied Sciences*, vol. 7, no. 10, p. 1101, 2017.
 - [35] K. Gusain, A. Gupta, and B. Popli, “Transition-aware human activity recognition using extreme gradient boosted decision trees,” in *Advanced Computing and Communication Technologies*. Springer, 2018, pp. 41–49.
 - [36] O. Baños, M. Damas, H. Pomares, I. Rojas, M. A. Tóth, and O. Amft, “A benchmark dataset to evaluate sensor displacement in activity recognition,” in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 1026–1035.
 - [37] M. O. Derawi, C. Nickel, P. Bours, and C. Busch, “Unobtrusive user-authentication on mobile phones using biometric gait recognition,” in *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE, 2010, pp. 306–311.
 - [38] D. Gafurov and E. Snekkenes, “Towards understanding the uniqueness of gait biometric,” in *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*. IEEE, 2008, pp. 1–8.
 - [39] D. Gafurov, E. Snekkenes, and P. Bours, “Improved gait recognition performance using cycle matching,” in *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 2010, pp. 836–841.

- [40] M. Derawi and P. Bours, “Gait and activity recognition using commercial phones,” *computers & security*, vol. 39, pp. 137–144, 2013.
- [41] S. Sprager and D. Zazula, “A cumulant-based method for gait identification using accelerometer data with principal component analysis and support vector machine,” *WSEAS Transactions on Signal Processing*, vol. 5, no. 11, pp. 369–378, 2009.
- [42] L. Rong, Z. Jianzhong, L. Ming, and H. Xiangfeng, “A wearable acceleration sensor system for gait recognition,” in *2007 2nd IEEE Conference on Industrial Electronics and Applications*. IEEE, 2007, pp. 2654–2659.
- [43] L. Rong, D. Zhiguo, Z. Jianzhong, and L. Ming, “Identification of individual walking patterns using gait acceleration,” in *2007 1st international Conference on Bioinformatics and Biomedical Engineering*. IEEE, 2007, pp. 543–546.
- [44] L. Tran and D. Choi, “Data augmentation for inertial sensor-based gait deep neural network,” *IEEE Access*, vol. 8, pp. 12 364–12 378, 2020.
- [45] R. Subramanian and S. Sarkar, “Evaluation of algorithms for orientation invariant inertial gait matching,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 304–318, 2018.
- [46] G. Giorgi, F. Martinelli, A. Saracino, and M. Sheikhalishahi, “Try walking in my shoes, if you can: Accurate gait recognition through deep learning,” in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2017, pp. 384–395.
- [47] F. Sun, W. Zang, R. Gravina, G. Fortino, and Y. Li, “Gait-based identification for elderly users in wearable healthcare systems,” *Information Fusion*, vol. 53, pp. 134–144, 2020.
- [48] P. Fernandez-Lopez, J. Liu-Jimenez, K. Kiyokawa, Y. Wu, and R. Sanchez-Reillo, “Recurrent neural network for inertial gait user recognition in smartphones,” *Sensors*, vol. 19, no. 18, p. 4054, 2019.
- [49] D. Gafurov, K. Helkala, and T. Søndrol, “Biometric gait authentication using accelerometer sensor.” *JCP*, vol. 1, no. 7, pp. 51–59, 2006.
- [50] ——, “Gait recognition using acceleration from mems,” in *First International Conference on Availability, Reliability and Security (ARES’06)*. IEEE, 2006, pp. 6–pp.

- [51] K. Holien, “Gait recognition under non-standard circumstances,” Master’s thesis, 2008.
- [52] D. Gafurov, “Performance and security analysis of gait-based user authentication,” 2008.
- [53] M. Muaaz and R. Mayrhofer, “Smartphone-based gait recognition: From authentication to imitation,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 11, pp. 3209–3221, 2017.
- [54] M. Derawi and P. Bours, “Gait and activity recognition using commercial phones,” *Computers Security*, vol. 39, pp. 137 – 144, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167404813000953>
- [55] T. Hoang, D. Choi, and T. Nguyen, “On the instability of sensor orientation in gait verification on mobile phone,” in *2015 12th International Joint Conference on e-Business and Telecommunications (ICETE)*, vol. 4. IEEE, 2015, pp. 148–159.
- [56] M. De Marsico and A. Mecca, “Biometric walk recognizer,” *Multimedia Tools and Applications*, vol. 76, no. 4, pp. 4713–4745, 2017.
- [57] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, “Orientation-compensative signal registration for owner authentication using an accelerometer,” *IEICE TRANSACTIONS on Information and Systems*, vol. 97, no. 3, pp. 541–553, 2014.
- [58] C. Nickel, H. Brandt, and C. Busch, “Classification of acceleration data for biometric gait recognition on mobile devices,” *BIOSIG 2011—Proceedings of the Biometrics Special Interest Group*, 2011.
- [59] H. Lu, J. Huang, T. Saha, and L. Nachman, “Unobtrusive gait verification for mobile phones,” in *Proceedings of the 2014 ACM international symposium on wearable computers*, 2014, pp. 91–98.
- [60] Y. Zhong and Y. Deng, “Sensor orientation invariant mobile gait biometrics,” in *IEEE international joint conference on biometrics*. IEEE, 2014, pp. 1–8.
- [61] C. Nickel, M. O. Derawi, P. Bours, and C. Busch, “Scenario test of accelerometer-based biometric gait recognition,” in *2011 Third International Workshop on Security and Communication Networks (IWSCN)*. IEEE, 2011, pp. 15–21.

- [62] Y. Zhao and S. Zhou, “Wearable device-based gait recognition using angle embedded gait dynamic images and a convolutional neural network,” *Sensors*, vol. 17, no. 3, p. 478, 2017.
- [63] O. Dehzangi, M. Taherisadr, and R. ChangalVala, “Imu-based gait recognition using convolutional neural networks and multi-sensor fusion,” *Sensors*, vol. 17, no. 12, p. 2735, 2017.
- [64] R. Delgado-Escano, F. M. Castro, J. R. C  zar, M. J. Mar  n-Jim  nez, and N. Guil, “An end-to-end multi-task and fusion cnn for inertial-based gait recognition,” *IEEE Access*, vol. 7, pp. 1897–1908, 2018.
- [65] R. Dama  vi  cius, R. Maskeli  nas, A. Ven  kauskas, and M. Wo  niak, “Smartphone user identity verification using gait characteristics,” *Symmetry*, vol. 8, no. 10, p. 100, 2016.
- [66] M. Ehatisham-ul Haq, M. A. Azam, U. Naeem, Y. Amin, and J. Loo, “Continuous authentication of smartphone users based on activity pattern recognition using passive mobile sensing,” *Journal of Network and Computer Applications*, vol. 109, pp. 24–35, 2018.
- [67] M. N. Malik, M. A. Azam, M. Ehatisham-Ul-Haq, W. Ejaz, and A. Khalid, “Adlauth: Passive authentication based on activity of daily living using heterogeneous sensing in smart cities,” *Sensors*, vol. 19, no. 11, p. 2466, 2019.
- [68] G. M. Weiss, K. Yoneda, and T. Hayajneh, “Smartphone and smartwatch-based biometrics using activities of daily living,” *IEEE Access*, vol. 7, pp. 133 190–133 202, 2019.
- [69] N. Neverova, C. Wolf, G. Lacey, L. Fridman, D. Chandra, B. Barbelli, and G. Taylor, “Learning human identity from motion patterns,” *IEEE Access*, vol. 4, pp. 1810–1820, 2016.
- [70] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, “Deepsense: A unified deep learning framework for time-series mobile sensing data processing,” in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 351–360.
- [71] M. P. Centeno, A. van Moorsel, and S. Castruccio, “Smartphone continuous authentication using deep learning autoencoders,” in *2017 15th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2017, pp. 147–1478.

- [72] M. P. Centeno, Y. Guan, and A. van Moorsel, “Mobile based continuous authentication using deep features,” in *Proceedings of the 2nd International Workshop on Embedded and Mobile Deep Learning*, 2018, pp. 19–24.
- [73] Z. Qin, L. Hu, N. Zhang, D. Chen, K. Zhang, Z. Qin, and K.-K. R. Choo, “Learning-aided user identification using smartphone sensors for smart homes,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7760–7772, 2019.
- [74] R. Delgado-Escáño, F. M. Castro, J. R. Cázar, M. J. Marín-Jiménez, N. Guil, and E. Casilar, “A cross-dataset deep learning-based classifier for people fall detection and identification,” *Computer methods and programs in biomedicine*, vol. 184, p. 105265, 2020.
- [75] M. Abuhamad, T. Abuhmed, D. Mohaisen, and D. H. Nyang, “Autosen: Deep learning-based implicit continuous authentication using smartphone sensors,” *IEEE Internet of Things Journal*, 2020.
- [76] T. T. Um, F. M. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fitzek, and D. Kulic, “Data augmentation of wearable sensor data for parkinson’s disease monitoring using convolutional neural networks,” in *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, 2017, pp. 216–220.
- [77] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [78] K.-T. Nguyen, T.-L. Vo-Tran, D.-T. Dinh, and M.-T. Tran, “Gait recognition with multi-region size convolutional neural network for authentication with wearable sensors,” in *International Conference on Future Data and Security Engineering*. Springer, 2017, pp. 197–212.
- [79] S. Sprager and M. B. Juric, “An efficient hos-based gait authentication of accelerometer data,” *IEEE transactions on information forensics and security*, vol. 10, no. 7, pp. 1486–1498, 2015.
- [80] Z. Wei, W. Qinghui, D. Muqing, and L. Yiqi, “A new inertial sensor-based gait recognition method via deterministic learning,” in *2015 34th Chinese Control Conference (CCC)*. IEEE, 2015, pp. 3908–3913.