

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Biomedica



Tesi di Laurea Magistrale

*Sviluppo di un algoritmo di deep learning
per la segmentazione dei tratti rettilinei e non-rettilinei
durante il cammino a partire dai soli segnali sEMG*

Relatori

Prof.ssa Valentina Agostini

Ing. Marco Ghislieri

Candidata

Ilaria Bellomo

s254287

Anno Accademico 2019/2020

Ottobre 2020

*Ai tratti ed alle transizioni del mio percorso,
alle impronte che hanno lasciato in me.*

Sommario

In analisi del cammino, i protocolli sperimentali che richiedono al soggetto di camminare lungo un percorso non tracciato permettono di riprodurre condizioni di cammino simili a quelle della vita quotidiana. Tipicamente, il soggetto cammina avanti e indietro in rettilineo, invertendo la propria direzione in prossimità di ciascuna delle due estremità del rettilineo. Tale percorso, di fatto, è costituito dalla successione alternata di un tratto rettilineo ed un tratto non-rettilineo. Quest'ultimo coincide con la traiettoria "a U" del cambio di direzione a 180°. Lo studio del segnale elettromiografico di superficie (sEMG), acquisito durante l'esecuzione di protocolli di analisi del cammino, incontra l'esigenza di distinguere i brani di segnale acquisiti nel percorrere i tratti rettilinei da quelli acquisiti nel percorrere i tratti non-rettilinei. Il problema, ed obiettivo, della presente tesi è segmentare i segnali sEMG, per il riconoscimento delle due tipologie di tratti percorsi, in modo automatico e senza l'ausilio di ulteriori segnali (ad esempio, basografici e goniometrici). La tesi sviluppa tale problema di segmentazione automatica in termini di problema di classificazione binaria. Di fatto, l'obiettivo proposto è etichettare il singolo campione del segnale sEMG con la corrispettiva classe di appartenenza: classe dei tratti rettilinei oppure classe dei tratti non-rettilinei. A tal fine, il metodo adottato si basa su un algoritmo di deep learning: la rete neurale Bidirectional Long Short-Term Memory (BiLSTM).

Il dataset è costituito dai segnali sEMG di 12 muscoli del tronco e di un arto inferiore, acquisiti da una popolazione di composizione mista, sana (9 soggetti giovani e 13 soggetti adulti) e patologica (20 soggetti affetti dal morbo di Parkinson). L'applicabilità delle reti neurali BiLSTM al problema in esame è analizzata mediante un dataset di 26 trials, eseguiti dai soggetti sani. Il metodo viene esplorato in termini di confronto tra due approcci di classificazione, il Sequence-to-Label (STL) ed il Sequence-to-Sequence

(STS). Complessivamente, sono stati implementati 160 modelli BiLSTM in modo parallelo tra i due approcci. Alla selezione del modello migliore segue la sua ottimizzazione. Il classificatore viene ottimizzato, innanzitutto, valutando alcune variazioni della sua architettura. Poi, il dataset viene ampliato con 5 ulteriori trials di soggetti sani e 20 di pazienti parkinsoniani. Su tale dataset di 51 trials, mediante la 10-fold cross-validation, viene esaminata la robustezza delle prestazioni del modello e, successivamente, la dipendenza dalla numerosità di canali sEMG considerati come input della rete neurale. In seguito, a partire dal classificatore ottimizzato, viene definito ed ottimizzato l'algoritmo di classificazione finale.

Tale algoritmo fornisce accuracy media (\pm deviazione standard) pari al $77.8\% \pm 13.1\%$, sul dataset complessivo dei 51 trials, e pari al $76.1\% \pm 10.9\%$, su un ulteriore dataset di validazione composto da 10 trials (5 di soggetti sani e 5 di pazienti parkinsoniani), non compresi tra i 51 su cui è stato allenato il modello. I due valori di accuracy indicati sono confrontabili. Lo sono anche i risultati medi (\pm deviazione standard) di altre metriche di classificazione considerate.

Concludendo, l'algoritmo sviluppato fornisce risultati mediamente buoni e si configura come una prima soluzione robusta al problema. Sviluppi futuri potrebbero indirizzarsi verso la validazione dell'algoritmo su soggetti con locomozione compromessa da altre patologie e verso la sua ulteriore ottimizzazione.

Indice

Capitolo 1 - Introduzione	8
1.1 Descrizione del problema e dell'obiettivo della tesi	8
1.2 Aspetti di analisi del cammino	12
1.2.1 Ciclo del passo.....	12
1.2.2 Stato dell'arte	14
1.3 Deep Learning per classificazione di serie temporali	18
1.3.1 Rete neurale Bidirectional Long Short-Term Memory	19
1.3.2 Applicazioni di rete neurale LSTM/ BiLSTM.....	21
Capitolo 2 - Materiali e Metodi	23
2.1 Descrizione della popolazione	23
2.2 Protocollo sperimentale.....	24
2.2.1 Sistema di acquisizione STEP32.....	26
2.3 Definizione del dataset	34
2.3.1 Segmentazione manuale.....	35
2.3.2 Pre-Processing dei segnali sEMG	39
2.4 Esplorazione dell'applicabilità delle reti neurali BiLSTM al problema in esame	43
2.4.1 Costruzione dei dataset	47
2.4.2 Definizione del modello BiLSTM.....	51
2.5 Ottimizzazione del modello BiLSTM scelto	55
2.5.1 Variazioni dell'architettura.....	55
2.5.2 Valutazione della robustezza delle prestazioni	58
2.5.3 Valutazione della dipendenza delle prestazioni dal numero di canali sEMG	59
2.6 Definizione del funzionamento dell'algoritmo finale.....	61
2.6.1 Ottimizzazione dell'algoritmo finale	64

2.7	Valutazione delle prestazioni.....	66
Capitolo 3 - Risultati		69
3.1	Esplorazione dell'applicabilità delle reti neurali BiLSTM.....	69
3.1.1	Definizione del modello BiLSTM.....	69
3.2	Ottimizzazione del modello BiLSTM scelto	72
3.2.1	Variazioni dell'architettura.....	72
3.2.2	Valutazione della robustezza delle prestazioni	73
3.2.3	Valutazione della dipendenza delle prestazioni dal numero di canali sEMG	75
3.3	Post-Processing.....	78
Capitolo 4 - Discussione e Conclusioni		85
Appendice A - Rete neurale BiLSTM		93
A.1	Unità LSTM.....	93
A.2	BiLSTM Layer.....	95
A.3	Opzioni di Training	96
Appendice B - Risultati		99
B.1	Definizione del modello BiLSTM.....	99
B.2	Variazioni dell'architettura.....	108
Bibliografia e Sitografia.....		109
Ringraziamenti.....		117

Capitolo 1

Introduzione

1.1 Descrizione del problema e dell'obiettivo della tesi

In analisi del cammino, i protocolli sperimentali che richiedono al soggetto di camminare lungo un percorso non tracciato permettono di riprodurre condizioni di cammino simili a quelle della vita quotidiana. Il cammino quotidiano è caratterizzato da ampia variabilità nei parametri spaziali e temporali e nei patterns di attivazione muscolare [1]. In laboratorio, in assenza delle limitazioni di un ambiente controllato (come nel caso di cammino su tapis roulant), condizioni simili al cammino quotidiano possono essere riprodotte lungo percorsi che inglobino traiettorie sia rettilinee che curvilinee. Inoltre, la mancanza di un percorso tracciato sul pavimento permette al soggetto che cammina di mantenere un'andatura naturale, senza forzature biomeccaniche specifiche. Tipicamente [2], [3], [4], [5], [6], [7], [8], [9], al soggetto è richiesto di descrivere il percorso mostrato in **Figura 1.1**. Tale percorso, sostanzialmente, è costituito dalla successione di tratti rettilinei e non-rettilinei in alternanza. Il soggetto cammina in rettilineo avanti e indietro, invertendo la propria direzione in prossimità di ciascuna estremità del rettilineo.

Di fatto, come illustrato in **Figura 1.2.**, lungo il percorso sono presenti due tipologie di tratti: rettilinei e non-rettilinei. Un tratto non-rettilineo coincide con la sequenza di decelerazione, curva ed accelerazione in coincidenza di un cambio di direzione a 180°

(curva "a U"). A rigore, la decelerazione e l'accelerazione, inglobate in un tratto non-rettilineo, si manifestano lungo il rettilineo, all'estremità.

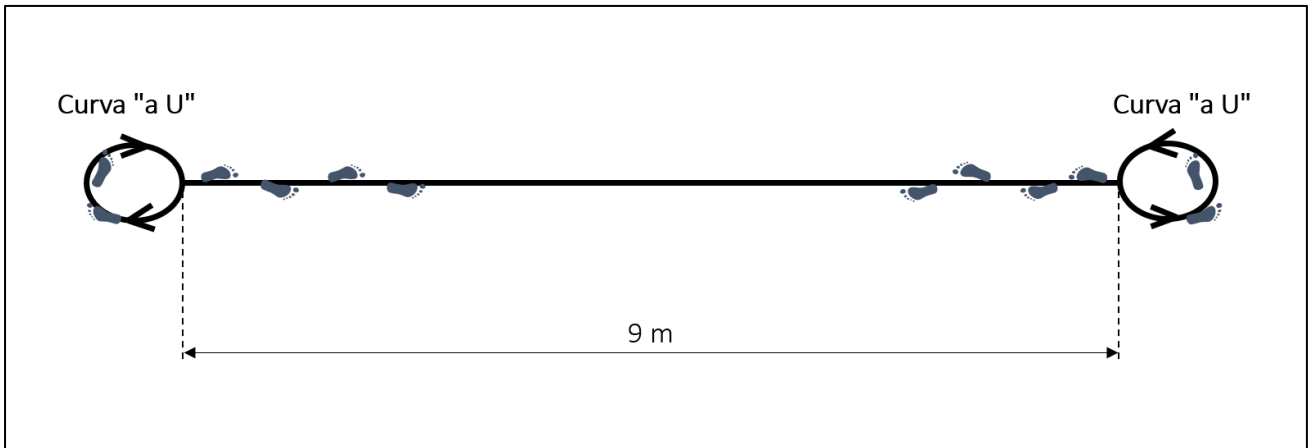


Figura 1.1. Percorso del protocollo sperimentale relativo ai trials esaminati nella presente tesi. Il soggetto cammina in rettilineo per 9 metri, avanti e indietro, invertendo la propria direzione alle estremità del percorso (curve "a U").

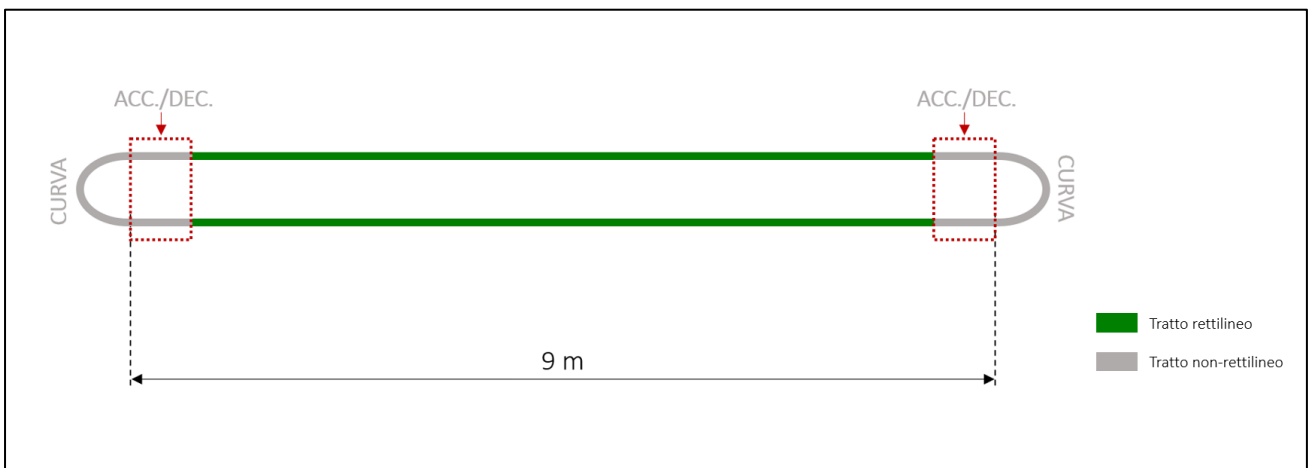


Figura 1.2. Individuazione delle due tipologie di tratti lungo il percorso: tratti rettilinei (in verde), tratti non-rettilinei (in grigio). Al fine di visualizzare più chiaramente le due curve "a U", il rettilineo di **Figura 1.1.** è rappresentato sdoppiato. Da protocollo, però, il soggetto cammina sempre (più o meno) lungo una stessa traiettoria rettilinea. Ovvero, i due tratti in verde, nella realtà, sono pressoché sovrapposti.

In analisi del cammino, si presenta l'esigenza di segmentare i segnali elettromiografici di superficie (sEMG), acquisiti mentre il soggetto cammina come in **Figura 1.1.**, in brani acquisiti lungo i tratti, rispettivamente, rettilinei e non-rettilinei del percorso. La presente tesi sviluppa il problema di effettuare tale segmentazione dei segnali sEMG in maniera automatica e senza l'ausilio di altri segnali (ad esempio, basografici e goniometrici). La tesi propone il problema di distinzione tra le due tipologie di tratti come problema di classificazione binaria. Al fine di individuare la classe dei tratti rettilinei e la classe dei tratti non-rettilinei, il metodo adottato consiste nell'implementazione di un algoritmo di deep learning: la rete neurale Bidirectional Long Short-Term Memory (BiLSTM). L'obiettivo posto, di fatto, è il seguente: etichettare il singolo campione dei segnali sEMG con la corrispondente classe di appartenenza.

La tesi guarda ad un duplice impatto in analisi del cammino:

- (1) fornire un metodo di segmentazione automatica dei segnali sEMG in tratti di rettilineo e tratti di non-rettilineo, parimenti sfruttabile sia per lo studio del cammino lungo percorsi rettilinei sia da gruppi di ricerca interessati ad isolare i cambi di direzione a 180°;
- (2) acquisire soltanto i segnali sEMG nei protocolli sperimentali per i quali i segnali dei sensori basografici e goniometrici servano esclusivamente per la selezione di specifiche tipologie di cicli del passo.

Rispetto a quanto puntualizzato in (1), in analisi del cammino, tradizionalmente, l'interesse è rivolto ai soli percorsi rettilinei. Tuttavia, l'analisi dei cambi di direzione a 180° permette di studiare la locomozione umana in condizioni di maggiore criticità. È osservato, infatti, che gli anziani ed i pazienti con disfunzioni nella deambulazione (ad esempio, affetti dal morbo di Parkinson oppure da emiparesi post-ictus) manifestano

difficoltà nel descrivere le curve “a U” [10], [11]. In particolare, per i soggetti patologici, l’analisi del cammino nei cambi di direzione a 180° permette di studiare l’evoluzione della patologia e di monitorarne il trattamento terapeutico [12].

Il vantaggio di (2) sarebbe, da una parte, la possibilità di ridurre l’ingombro del set-up sperimentale e di ottimizzare i tempi di pratica clinica [1] oppure, dall’altra, la possibilità di dedicare al segnale sEMG un maggior numero di canali del sistema di acquisizione.

1.2 Aspetti di analisi del cammino

1.2.1 Ciclo del passo

Il ciclo del passo è la sequenza di eventi biomeccanici tra due consecutivi appoggi-iniziali del piede di uno stesso arto inferiore. Ciascun ciclo è tipicamente suddiviso in due fasi principali: stance e swing. Come mostrato in **Figura 1.3.**, la fase di stance (della durata circa del 60% del ciclo di cammino) e la successiva fase di swing (della durata circa del 40% del ciclo di cammino) sono gli intervalli di tempo durante cui, rispettivamente, il piede è a contatto con il suolo ed il piede non è a contatto con il suolo per l'avanzamento dell'arto [13], [14]. Nei soggetti sani, la maggior parte dei cicli è costituita dalla successione di tre sotto-fasi di stance: heel contact (H), flat foot contact (F), push off (o heel off) (P); segue lo swing (S) dell'arto [13].

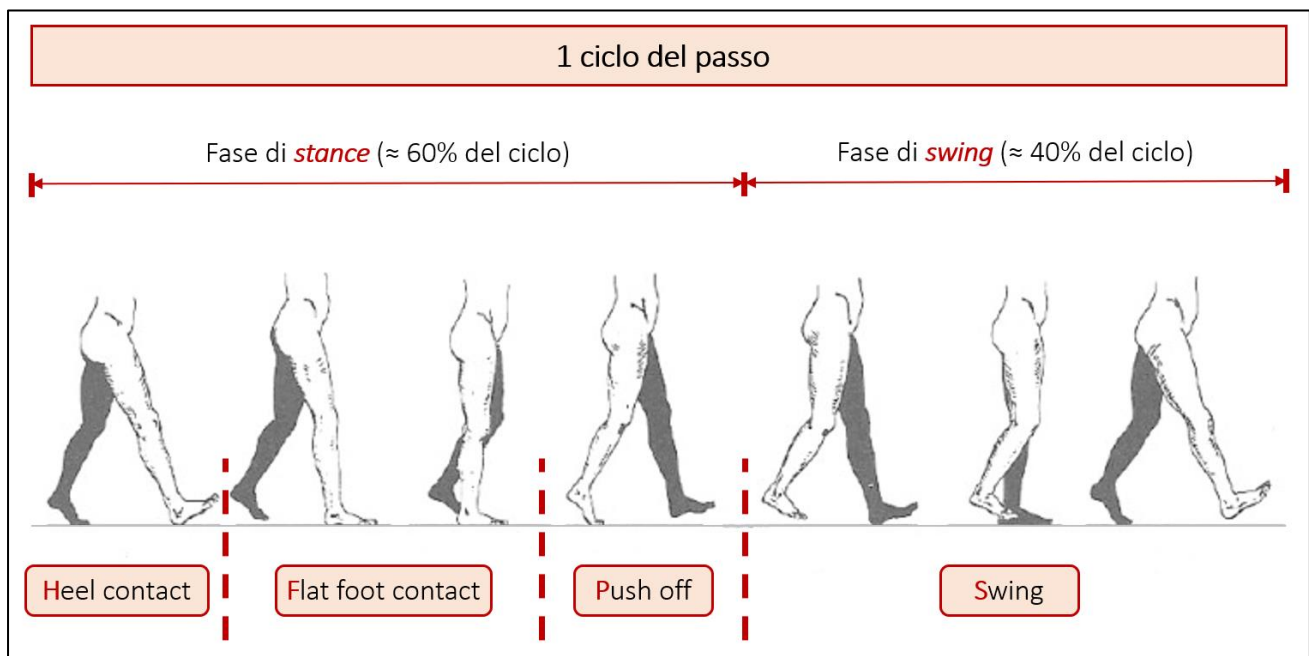


Figura 1.3. Fasi del ciclo del passo (gamba destra, in bianco): heel contact (H), flat foot contact (F), push off (o heel off) (P), swing (S). Adattata da [15].

Considerando la gamba destra, come dettagliato in [6], l'appoggio-iniziale del piede destro si verifica mentre il piede sinistro è ancora appoggiato al suolo: periodo di doppio supporto. Durante la fase di swing dell'arto sinistro, soltanto il piede destro è appoggiato al suolo: periodo di singolo supporto destro, che termina con l'appoggio-iniziale del piede sinistro. Segue un altro periodo di doppio supporto, fino allo stacco del piede destro. Durante la fase di swing dell'arto destro, soltanto il piede sinistro è appoggiato al suolo: periodo di singolo supporto sinistro. Il ciclo si conclude con il successivo appoggio-iniziale del piede destro.

Il ciclo del passo può essere caratterizzato da altre sequenze differenti da HFPS (ciclo tipico) [13], prevalenti nel cammino patologico. Ne sono esempio i soggetti con piede equino ed i pazienti affetti da malattie neuro-degenerative (quali ictus cerebrale, morbo di Parkinson, sclerosi multipla, distrofia muscolare). Nei soggetti patologici, inoltre, le fasi H, F, P, S anche dei cicli tipici possono possedere durata alterata (maggiore o minore) rispetto alle corrispondenti fasi dei cicli tipici di soggetti sani [16]. In letteratura [10], i cicli del passo che non presentano il pattern "standard" HFPS sono chiamati cicli atipici. Esempi di patterns più frequenti nei cicli atipici sono PS, PFPS, FPS, HFPSPS.

Come riportato in [13], l'analisi di "lunghe" camminate (almeno 100 - 250 passi consecutivi) mostra che il cammino (sia di soggetti sani che patologici) include sia cicli tipici che atipici, ovvero uno stesso soggetto presenta cicli del passo con sequenze di fasi differenti (variabilità intra-soggetto nel cammino). Nei soggetti patologici, a seconda della patologia, l'occorrenza percentuale di cicli atipici può significativamente aumentare, fino al 100% in soggetti gravemente compromessi.

1.2.2 Stato dell'arte

In letteratura [17], l'analisi del cammino è stata estensivamente sviluppata limitando l'interesse a traiettorie rettilinee. Numericamente inferiori sono gli studi pubblicati che analizzano il cammino lungo percorsi non-lineari, ad esempio curvilinei.

Come spiegato in [17], [18], durante il cammino in rettilineo, la simmetria dei movimenti facilita la coordinazione dei segmenti del corpo. Critica risulta invece la condizione di cammino in curva: essa, infatti, richiede di associare alla propulsione la ri-orientazione di testa, tronco, bacino e piedi, al fine di mantenere l'equilibrio, e il movimento asimmetrico degli arti inferiori, per cui la gamba nella parte interna della traiettoria percorre un tratto più breve della gamba all'esterno. Gli adattamenti del passo alla curva includono variazioni in durata della fase di stance e in lunghezza del passo e si riflettono in patterns muscolari arto-specifici. Anche l'analisi del cammino in curva possiede rilevanza clinica fondamentale. Studi recenti [17] mostrano, infatti, che soggetti affetti da patologie neurologiche (ad esempio, ictus cerebrale e morbo di Parkinson) manifestano difficoltà nel cammino più evidenti lungo traiettorie circolari piuttosto che rettilinee.

Relativamente al problema della presente tesi, in letteratura, sono pochi gli studi che propongono metodi di segmentazione dei tratti rettilinei e non-rettilinei durante il cammino [19]. Nella fattispecie, non sono stati trovati riferimenti a metodi che permettano di effettuare tale segmentazione a partire dai soli segnali sEMG, in maniera automatica e indipendente dall'ausilio di ulteriori segnali (ad esempio, basografici, goniometrici).

In [2], [3], [4], [5], [6], [7], [8], [9], sono considerati soltanto i cicli del passo con sequenza HFPS. A partire dai segnali basografici ed elettro-goniometrici acquisiti mediante il sistema elettromedicale STEP32 (Medical Technology, Italia), il filtro statistico multivariato (Hotelling t-test, livello di significatività $\alpha = 0.05$) del software di STEP32

rimuove i cicli outlier, ovvero cicli tipici ma con tempistica anomala delle fasi H, F, P, S. I cicli outlier possono includere quelli relativi ai cambi di direzione (comprese la decelerazione precedente e l'accelerazione seguente l'effettiva curva). Sostanzialmente, l'esclusione di tali cicli è effettuata in maniera automatica, però appoggiandosi ai segnali basografici ed elettro-goniometrici. In [3], è riportata la rappresentazione del percorso per analisi del cammino in rettilineo (**Figura 1.4.**): i soggetti sono istruiti a camminare avanti e indietro lungo il rettilineo di 15 metri, a superare i segni (a, b) sul pavimento, che identificano la fine del tratto rettilineo, prima di decelerare, cambiare direzione e accelerare.

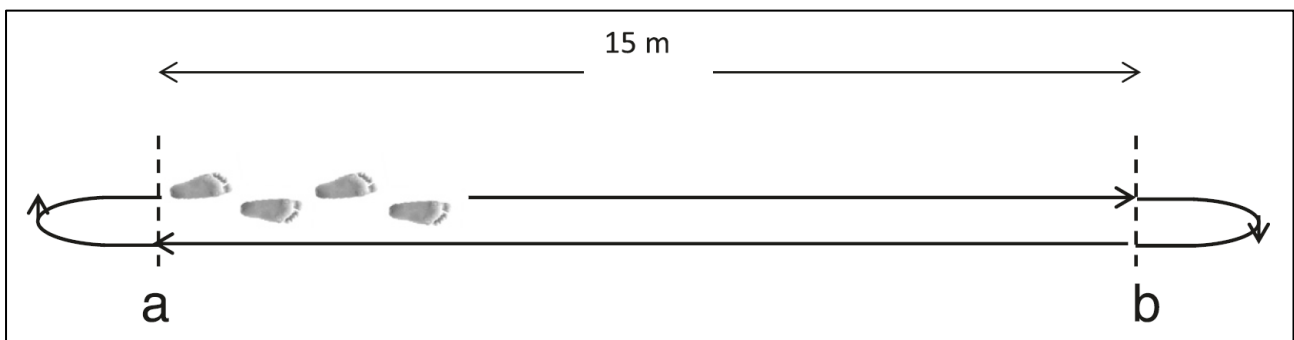


Figura 1.4. Percorso del protocollo sperimentale. I marks (a, b) alle estremità del rettilineo di 15 metri delimitano il percorso rettilineo lungo cui i soggetti sono istruiti a camminare ad una cadenza normale. Da [3].

In [10], sono riportati due metodi di identificazione delle curve “a U” durante il cammino. Un metodo, basato su segnali inerziali, consiste nel segmentare i singoli passi, definire una direzione di progressione (DoP, “Direction of Progression”) per ciascun passo e riconoscere come curve quei passi la cui DoP mostra un angolo non nullo con il passo precedente. L'altro metodo identifica una curva dall'analisi dei segnali acquisiti mediante IMUs (“Inertial Measurement Units”) e non dipende dall'individuazione dei singoli cicli del passo. In [19], sono esaminati due metodi di identificazione delle curve “a U” a partire dai segnali acquisiti mediante giroscopi. In [20], viene proposto un

metodo di segmentazione dei percorsi rettilinei e delle curve “a U” a partire dai segnali acquisiti mediante MIMUs (“Magneto-Inertial Measurement Units”).

In [21], i soggetti camminano lungo un rettilineo di 10 metri ed invertono la propria direzione alle due estremità, descrivendo curve a 180°. I segnali, acquisiti mediante IMUs, sono segmentati in tratti di rettilineo e di curva, utilizzando le videoregistrazioni dei trials. In seguito, le curve sono escluse dal dataset e, di conseguenza, lo studio si sviluppa considerando soltanto il percorso rettilineo. In [22], mentre i soggetti camminano lungo un rettilineo di 10 metri, vengono acquisiti dati RGB-D, mediante sensori Kinect v2. Soltanto i dati relativi al tratto intermedio di 6 metri vengono analizzati. Il tratto iniziale di 2 metri, in cui il soggetto accelera, e quello finale di 2 metri, in cui il soggetto decelera, sono esclusi utilizzando le videoregistrazioni acquisite.

In [23], i soggetti camminano lungo due percorsi costituiti da un tratto rettilineo e un quarto di cerchio con raggio, rispettivamente, di 0.5 metri e 1 metro. Il cammino è analizzato a partire dai dati acquisiti da un sistema di videoregistrazione 3D del movimento (MAC 3D system, Motion Analysis Corporation, U.S.). L’analisi è limitata alla sola sezione angolare (quarto di cerchio, evidenziato in **Figura 1.6.**) che viene “estratta” dal resto del trial, utilizzando la videoregistrazione acquisita.

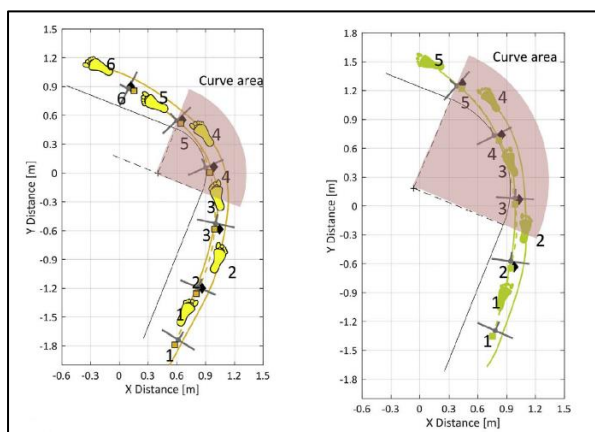


Figura 1.6. Percorsi del protocollo sperimentale. I soggetti camminano lungo un tratto rettilineo seguito da un tratto a quarto di cerchio (raggio di 0.5 metri, immagine a sinistra; raggio di 1 metro, immagine a destra). Da [23].

In [24], i soggetti camminano lungo un rettilineo di 10 metri (**Figura 1.7.**). Al fine di analizzare soltanto il tratto rettilineo, escludendo l'accelerazione iniziale e la decelerazione finale lungo il percorso, l'acquisizione dei dati RGB-D (sensori Kinect v2) è limitata ai soli 6 metri centrali del percorso. Di fatto, 2 metri per ciascuna estremità del rettilineo sono considerati come caratterizzati dai fenomeni di, rispettivamente, accelerazione e decelerazione e vengono rimossi non acquisendoli proprio.

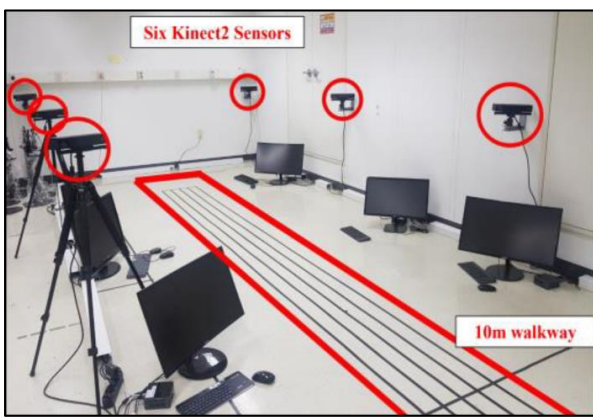


Figura 1.7. Percorso del protocollo sperimentale. I soggetti camminano lungo il rettilineo di 10 metri. I sei sensori sono posizionati in modo da coprire i 6 metri centrali cui è limitata la raccolta dei dati. Da [24].

1.3 Deep Learning per classificazione di serie temporali

Negli ultimi anni sono stati sviluppati nuovi metodi di intelligenza artificiale, tra cui il deep learning, con varie applicazioni in ricerca ed in pratica clinica [25]. Il deep learning include due principali algoritmi: Recurrent Neural Networks (RNNs), applicate a dati sequenziali, e Convolutional Neural Networks (CNNs), tipicamente applicate ad immagini [26]. Le reti neurali “deep” possono estrarre ed apprendere features di alto livello, non richiedendo necessariamente features engineering (ossia estrazione e successiva selezione delle features discriminanti) [25], [1].

Di fatto, i segnali sEMG sono serie temporali. In letteratura [27], il deep learning è ampiamente sfruttato per problemi di Time Series Classification (TSC), con performances superiori ad altri metodi. In particolare, le RNNs sono una variante del multilayer perceptron standard [28], capaci di apprendere le correlazioni tra data points che sono vicini nella sequenza [29], e possono essere applicate per la classificazione di serie temporali [30]. Le RNNs presentano due fondamentali limitazioni [31]:

- (1) Poiché processano gli input in ordine temporale, la predizione ad un certo time step si basa sulle informazioni degli input precedenti e non di quelli successivi. Sostanzialmente, apprendono dal contesto temporale passato ma non dal futuro, ovvero non sfruttano per intero l'informazione in input.
- (2) Presentano difficoltà nell'apprendere le dipendenze temporali in sequenze con elevato numero di time step.

Una soluzione a (1) è fornita dalle Bidirectional Recurrent Neural Networks (BRNNs) [29]. L'implementazione di tali reti neurali implica, però, la necessità di possedere, a monte, la totalità dei dati in input. Di conseguenza, esse non sono sfruttabili per applicazioni in real-time. Per risolvere (2) è stata introdotta una versione modificata di RNN, la Long Short-Term Memory (LSTM) [32].

1.3.1 Rete neurale Bidirectional Long Short-Term Memory

La rete neurale Bidirectional Long Short-Term Memory (BiLSTM) rappresenta una combinazione di BRNN e LSTM. Si tratta di una rete neurale bidirezionale nella cui architettura l'unità base di una RNN è sostituita dall'unità LSTM [33], [26], come illustrato in **Figura 1.8**. Mentre nella rete neurale LSTM l'informazione fluisce in avanti, la BiLSTM è stata proposta in [31] per processare entrambe le direzioni temporali contemporaneamente. In quanto bidirezionale [29], il sequence learning si sviluppa attraverso due hidden layers separati, connessi allo stesso output layer. Per ciascun punto di una sequenza in input, un hidden layer apprende da tutti i punti precedenti (direzione forward), l'altro hidden layer apprende da tutti i punti successivi (direzione backward). In quanto LSTM, i neuroni (unità base di una RNN) sono sostituiti da più complessi memory blocks (unità LSTM), al fine di affrontare il problema del vanishing gradient e apprendere lunghe dipendenze temporali.

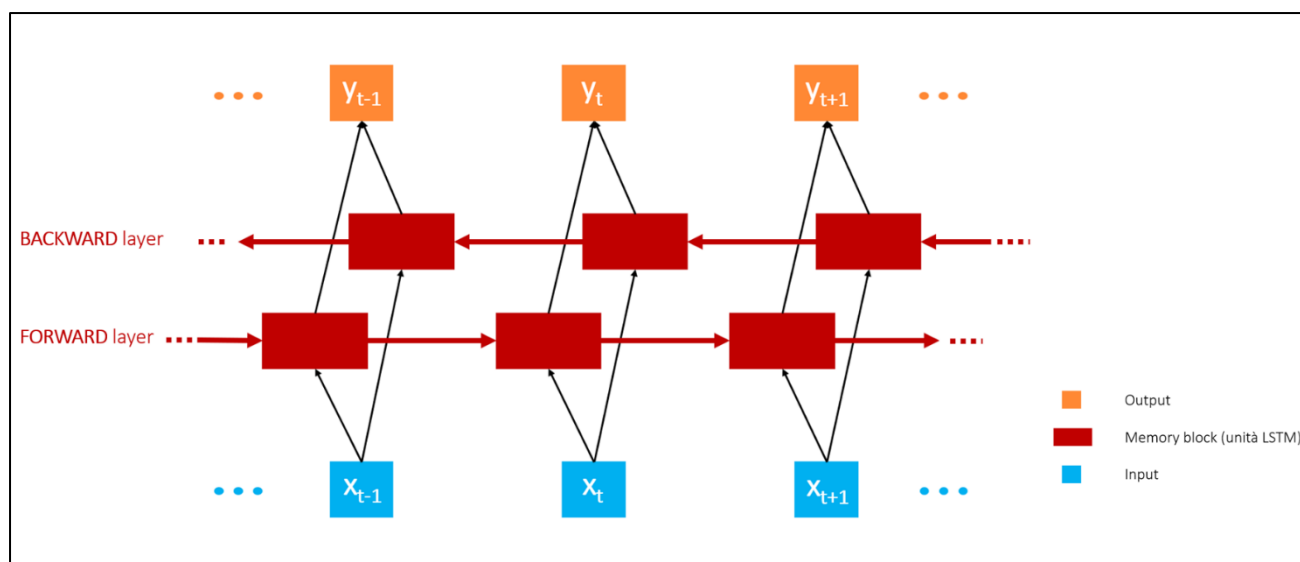


Figura 1.8. Schema a blocchi della BiLSTM. Ciascun time step va in input sia ad un memory block del forward layer sia ad uno del backward layer. L'output del forward layer e quello del backward layer confluiscono in un output unico della rete neurale per il singolo time step.

Gli algoritmi di apprendimento per le RNNs, ad esempio il Back-Propagation Through Time (BPTT), sono tipicamente basati sul calcolo del gradiente di una funzione di costo rispetto ai pesi della rete [34]. Durante l'allenamento della rete, iterativamente l'errore tra output e target viene calcolato e propagato "all'indietro", attraverso le connessioni interne alla rete, per aggiornarne i pesi. All'aumentare del numero di time step della sequenza in input alla rete, tale flusso di errore tende a decadere. Di conseguenza, in cascata, i pesi tendono a non cambiare, il gradiente della funzione di costo ad annullarsi (problema del vanishing gradient) e la rete a smettere di apprendere. La struttura interna di un'unità LSTM (memory block), invece, forza il flusso di errore a rimanere costante [35]. Ciascun memory block, come mostrato in **Figura 1.9.**, contiene una o più memory cells e tre unità non-lineari chiamate gates (input gate, forget gate e output gate). Il nucleo fondamentale di una memory cell è il Constant Error Carrousel (CEC), un'unità lineare che immagazzina l'errore in memoria e lo fa "ricircolare" inalterato [36]. Durante l'allenamento i tre gates apprendono ad aggiornare l'informazione memorizzata. Rispettivamente, l'input gate impedisce che essa sia alterata da input irrilevanti, l'output gate impedisce che altre unità siano alterate da informazioni irrilevanti presenti in memoria, il forget gate resetta una memory cell quando il suo contenuto è obsoleto [35]. La rete può interagire con le memory cells esclusivamente tramite i gates [31]. In **Appendice A.1**, sono riportati i dettagli matematici dell'unità LSTM.

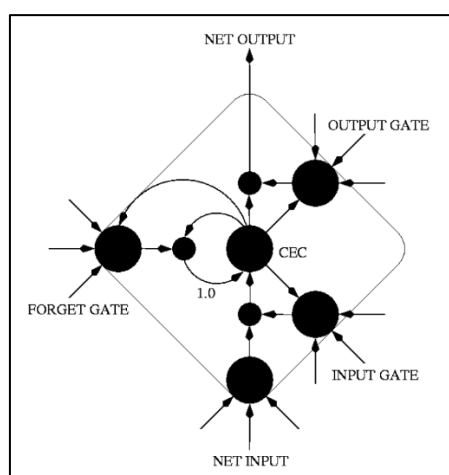


Figura 1.9. LSTM memory block con una memory cell. Il Constant Error Carrousel (CEC) mantiene costante lo stato della memory cell mediante una connessione recurrent con peso 1.0. I tre gates comunicano con l'interno e con l'esterno del memory block e controllano la memory cell tramite unità moltiplicative (cerchi piccoli). Da [35].

1.3.2 Applicazioni di rete neurale LSTM/ BiLSTM

La rete neurale LSTM, sia monodirezionale sia bidirezionale, è un metodo comunemente applicato per problemi di riconoscimento vocale, riconoscimento del testo, riconoscimento delle emozioni [37]. In letteratura, le reti neurali LSTM, sia feedforward che bidirezionali, sono anche applicate come metodo computazionale per analisi ed elaborazione di segnali biomedici [38]. Oltre ad esempi in ambito di Computer Vision [39], [40] oppure di Human Activity Recognition (HAR) [41], sono utilizzate per classificare anomalie cardiache a partire dal segnale elettrocardiografico (ECG) [26]. Rispetto a tale problema, una LSTM permette di soddisfare un duplice obiettivo: analizzare il segnale ECG più rapidamente in situazioni di emergenza e fornire allo specialista cardiologo uno strumento di supporto alla diagnosi [42]. Le applicazioni sul segnale elettroencefalografico (EEG) includono l'analisi automatica di registrazioni elettroencefalografiche intracraniche (iEEG) per l'individuazione di patterns epilettiformi [30], [25], la classificazione automatica delle fasi del sonno [26] e, anche a partire dal segnale elettro-oculografico (EOG), la rilevazione di sonnolenza durante la guida [38].

In analisi del cammino, permettono la classificazione di patterns del passo a partire dai segnali di reazione vincolare del terreno [43], il riconoscimento di patterns patologici a partire da video-frames acquisiti tramite sensori RGB-D [24], [44] e l'individuazione di cicli del passo da segnali accelerometrici [45].

Sono, infine, implementate per la classificazione del segnale EMG, sia intramuscolare (iEMG) che di superficie. In [46], permettono la classificazione di movimenti facciali in soggetti affetti da paralisi facciale, basata su segnali iEMG registrati dal lato paretico. Esempi di applicazione sul segnale sEMG, acquisito dagli arti sia superiori che inferiori, riguardano la realizzazione di protesi ed esoscheletri [47], [48], [49], il riconoscimento di gesti in ambito di Human-Computer Interface (HCI) [50], [51], il riconoscimento

dell'intenzione di movimento nella coordinazione uomo-macchina [52], [49]. In [53], sono implementate per elaborare i segnali sEMG acquisiti dagli arti superiori di soggetti affetti dal morbo di Parkinson, al fine di predire patterns di tremore in condizione di riposo. In [54], viene presentato un sistema hardware e software per l'acquisizione dei segnali sEMG dagli arti inferiori e la loro segmentazione, in real-time, rispetto a quattro fasi del ciclo del passo (fase iniziale, intermedia e finale di stance e fase di swing). Il software è basato su un classificatore, costituito da una combinazione di LSTM e multilayer perceptron, che riceve in ingresso i segnali sEMG grezzi.

Tipicamente, per un generico problema di classificazione di sequenze, una BiLSTM mostra prestazioni migliori rispetto ad una LSTM. Sostanzialmente, i risultati sperimentali confermano, come atteso, che l'apprendimento in entrambe le direzioni temporali, contemporaneamente, rende la rete neurale più robusta [30].

Capitolo 2

Materiali e Metodi

Globalmente, il lavoro svolto nella presente tesi si articola in quattro macrosezioni successive:

1. definizione del dataset;
2. esplorazione dell'applicabilità delle reti neurali BiLSTM al problema in esame;
3. ottimizzazione del modello BiLSTM scelto;
4. definizione del funzionamento ed ottimizzazione dell'algoritmo finale per la segmentazione dei tratti di rettilineo e non-rettilineo.

2.1 Descrizione della popolazione

Il dataset è costituito da segnali acquisiti da una popolazione di composizione mista, sana e patologica. La sottopopolazione sana (60.8%) include 9 soggetti giovani (sesso femminile; età: 24.2 ± 1.3 anni; altezza: 164.8 ± 7.2 cm; peso: 54.8 ± 6.1 kg) e 13 soggetti adulti (5 di sesso maschile; età: 52.5 ± 10.8 anni; altezza: 168.6 ± 10.4 cm; peso: 80.8 ± 25.9 kg). La sottopopolazione patologica (39.2%) è composta da 20 soggetti affetti da PD (Parkinson's Disease; 12 di sesso maschile; età: 58.1 ± 8.6 anni; altezza: 172.2 ± 10.5

cm; peso: 74.7 ± 12.9 kg). Nessuno dei soggetti sani presenta condizioni patologiche agli arti inferiori né disfunzioni neurologiche o muscolo-scheletriche che possano comprometterne il cammino.

La patologia scelta, al fine di disporre di un dataset di composizione mista (sana e patologica), è una delle malattie neurodegenerative più diffuse su scala mondiale. Essa è principalmente, ma non esclusivamente, associata ad una perdita di neuroni dopaminergici. Movimenti lenti, rigidità, tremore a riposo ed instabilità posturale sono le caratteristiche motorie distintive della malattia e hanno impatto negativo sulla qualità del movimento, sulla locomozione, sull'equilibrio durante il cammino e sul rischio di caduta [55].

2.2 Protocollo sperimentale

Il percorso, non tracciato sul pavimento, è costituito da un tratto rettilineo della lunghezza di 9 metri. Il rettilineo è delimitato sul pavimento da due linee rosse trasversali, una passante per il punto di partenza del soggetto e l'altra all'estremità opposta, come visibile in **Figura 2.1**. Il punto di partenza è lo stesso per tutti i soggetti. Il punto di arrivo, invece, dipende dalla cadenza e dalla velocità del soggetto. Al singolo soggetto viene chiesto di camminare, a piedi nudi, avanti e indietro lungo il rettilineo, invertendo la propria direzione non appena sia stata superata la linea rossa. La curva "a U" così percorsa è caratterizzata da un diametro molto variabile. Tale variabilità si manifesta principalmente inter-soggetto, ma anche intra-soggetto.

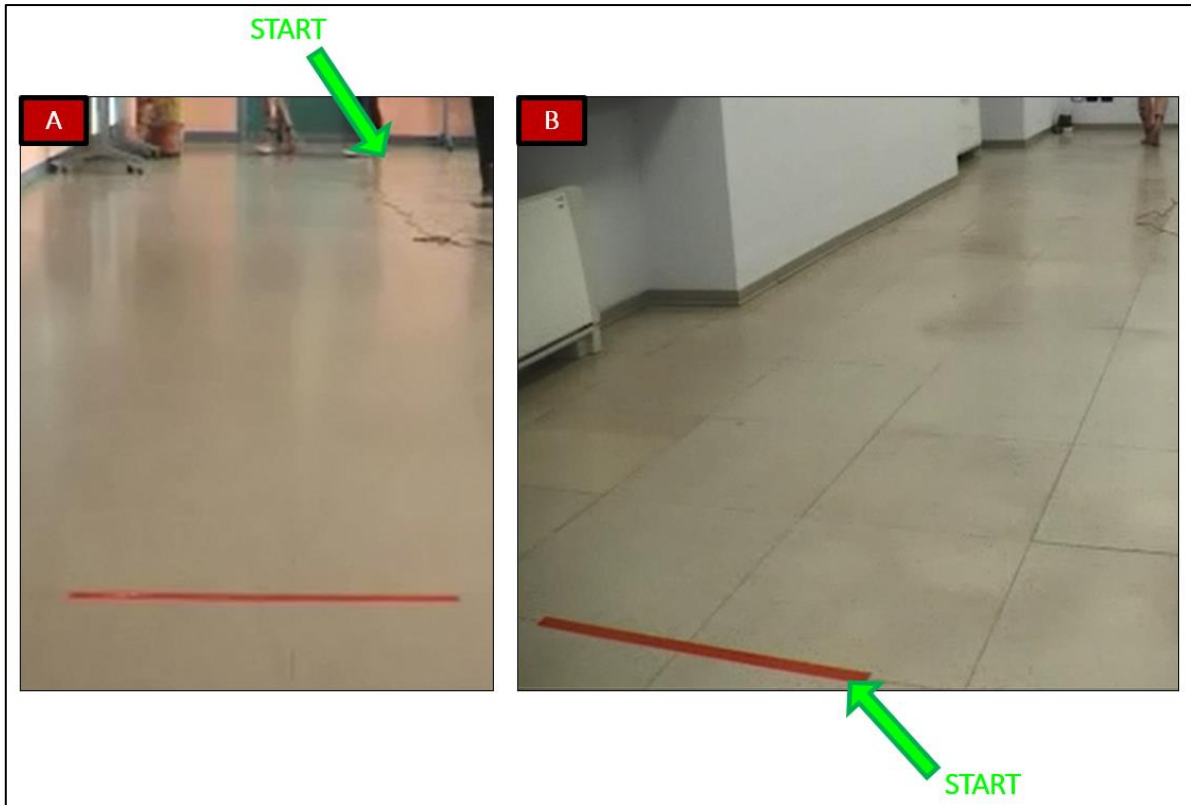


Figura 2.1. Ambienti di laboratorio. A sinistra (A) e a destra (B) sono mostrati i due ambienti di laboratorio relativi al dataset della tesi. In entrambi, il rettilineo è delimitato da due linee rosse trasversali, tracciate sul pavimento. Sulle due fotografie (A, B) è indicato il punto di partenza del soggetto (START). Il punto di arrivo, invece, non è indicato, in quanto dipendente dalla cadenza e dalla velocità del soggetto.

Durante l'esecuzione del protocollo sperimentale, tutti i soggetti hanno camminato con andatura normale, ininterrottamente, per circa 5 minuti. I soggetti sani giovani hanno camminato anche con andatura veloce, ininterrottamente, per circa 5 minuti.

La durata di circa 5 minuti del singolo trial permette di collezionare un elevato numero di passi (qualche centinaio), presupposto fondamentale per lo studio della variabilità intra-soggetto durante il cammino, in termini di parametri spazio-temporali del ciclo del passo e di patterns muscolari [13], [5].

2.2.1 Sistema di acquisizione STEP32

STEP32 (Medical Technology, Italia) è un sistema multicanale specificamente sviluppato per l'analisi clinica del cammino [8]. La configurazione base di STEP32 include i seguenti componenti [56]:

- l'unità paziente, tipicamente fissata alla vita del soggetto sottoposto all'esame mediante cintura elastica;
- i sensori di differente tipologia (basografici, goniometrici, elettromiografici, ecc.) da collegare all'unità paziente;
- la scheda proprietaria, da inserire in uno slot PCI del personal computer, che consente di acquisire fino a 16 segnali provenienti dall'unità paziente;
- il cavo di connessione della lunghezza di 12 m che collega la scheda PCI all'unità paziente;
- la videocamera standard digital video (DV), collegata ad una porta firewire, che registra il filmato del paziente durante l'esame, ed il suo applicativo software STEP 32-DV.

In particolare, utilizzato nella presente tesi per la segmentazione manuale, il software di STEP32 include: il modulo "cartella clinica", per memorizzare i dati anagrafici dei pazienti e l'elenco degli esami eseguiti; il modulo di "acquisizione", per visualizzare in tempo reale fino a 16 canali ed una finestra video in cui è riportata l'inquadratura della telecamera; il modulo di "rivisualizzazione", per visualizzare i segnali ad acquisizione terminata; il "Gait Analysis Wizard", che esegue l'analisi automatica del cammino; il modulo "digital video", per registrare e rivisualizzare il file video. La videocamera osserva il piano sagittale e acquisisce 25 frames/s [5]. Il filmato, registrato in formato AVI è sincronizzato ai segnali dei sensori collegati all'unità paziente. La visione del filmato, effettuabile a velocità diverse, permette di studiare l'andamento di tali segnali, ma non fornisce dati quantitativi [5].

Dal punto di vista metodologico [56], STEP32 analizza in modo automatico centinaia di passi in pochi secondi e fornisce una descrizione statistica dei parametri spazio-temporali del cammino di un soggetto. Il sistema elabora i segnali senza alterarne il contenuto informativo e non richiede all'utente di effettuare alcuna scelta. La ripetibilità dei risultati, che sono indipendenti dall'esperienza dell'operatore, permette di confrontare dati acquisiti in laboratori diversi.

Durante l'esecuzione del protocollo sperimentale sono contemporaneamente acquisiti i seguenti segnali:

- Basografici, bilateralmente, mediante foot-switches fissati sotto la pianta del piede con del nastro biadesivo, in corrispondenza del tallone, della prima e della quinta testa metatarsale. I tre interruttori basografici (dimensione: 10 mm x 10 mm x 0.5 mm; forza di attivazione: 3 N) sono collegati all'unità paziente mediante un convertitore D/A a 3 bit. Il connettore è fissato alla caviglia tramite un'apposita fascia in velcro.
- Goniometrici, da entrambe le articolazioni di ginocchio, mediante elettrogoniometri (accuracy: 0.5°) che misurano le escursioni angolari nel piano sagittale. L'elettrogoniometro è realizzato con un potenziometro lineare montato su una struttura a parallelogramma articolato. I due sensori sono posizionati lateralmente su ciascuna articolazione di ginocchio: un'estremità del goniometro è applicata sulla superficie laterale della coscia, allineata con la direzione del femore, e l'altra estremità al di sotto dell'articolazione, lungo il decorso del peroneo lungo, allineata con la direzione della tibia. Gli elettrogoniometri, fissati alla cute con del nastro biadesivo, non richiedono l'allineamento del potenziometro con il centro di rotazione articolare, perché il parallelogramma compensa la variabilità del posizionamento.

- Elettromiografici di superficie (sEMG), mediante sonde attive (configurazione: singolo differenziale; dimensione: 19 mm x 17 mm x 7 mm; diametro dei dischi in argento: 4 mm; distanza interelettrodica: 12 mm; guadagno: variabile nel range tra 60 dB e 86 dB). La cute del soggetto viene sottoposta a tricotomia (se necessario) e trattata con gel conduttivo. I sensori sono posizionati sui seguenti 12 muscoli del tronco e di un solo arto inferiore: Vastus Medialis (VM), Tensor Fasciae Latae (TFL), Gluteus Medius (GMD), Medial Hamstring (MH), right Longissimus Dorsii (LD_R), left Longissimus Dorsii (LD_L), Tibialis Anterior (TA), Lateral Gastrocnemius (LGS), Peroneus Longus (PL), Soleus (SOL), Rectus Femoris (RF), Lateral Hamstring (LH). L'arto inferiore scelto è il dominante (in riferimento al piede con cui iniziano a camminare), nei soggetti sani e, invece, il più compromesso dalla patologia, nei soggetti affetti da PD.

I 16 segnali sono acquisiti con una frequenza di campionamento di 2000 Hz, convertiti da un convertitore A/D a 12 bit e inviati al PC per la rappresentazione in real-time.

In **Figura 2.2.**, è mostrato il posizionamento dei tre interruttori basografici, dell'elettrogoniometro in corrispondenza dell'articolazione di ginocchio e di una sonda elettromiografica di superficie. In **Figura 2.3.**, è mostrato un soggetto sul quale sono montate le tre tipologie di sensori, come previsto dal protocollo sperimentale della presente tesi.

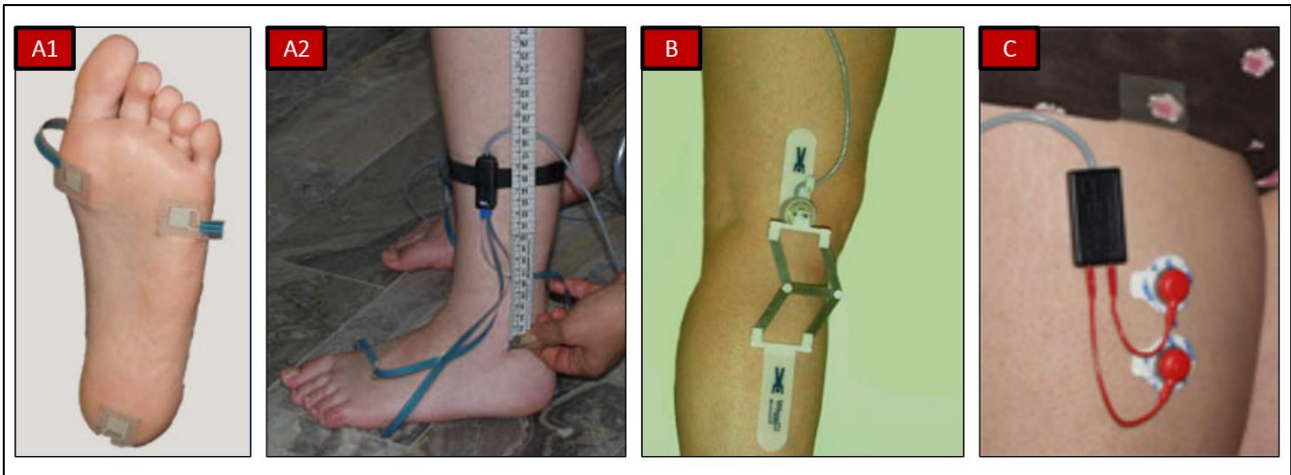


Figura 2.2. Sensori di STEP32 posizionati. In A1, i tre interruttori basografici su tallone, prima e quinta testa metatarsale. In A2, i codoli degli interruttori basografici collegati al connettore (scatoletta nera) che codifica i segnali provenienti dagli interruttori e li trasmette ad un unico canale dell'unità paziente. In B, l'elettrogoniometro sull'articolazione di ginocchio destro (lateralmente). In C, una sonda sEMG.

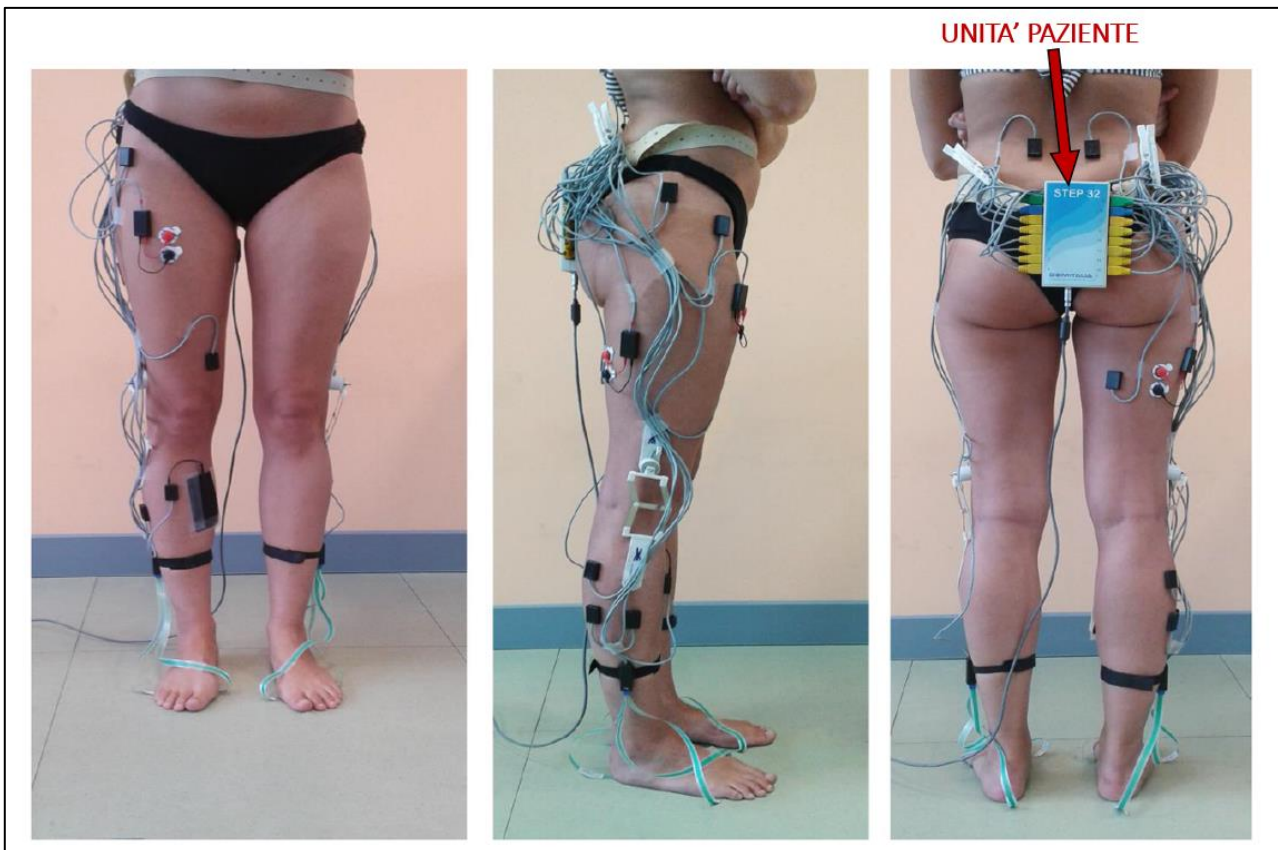


Figura 2.3. Soggetto con sensori basografici, goniometrici, elettromiografici di STEP32 posizionati. Bilateralmente, gli interruttori basografici e gli elettrogoniometri (articolazioni di ginocchio). Monolateralmente, le sonde sEMG. È indicata l'unità paziente fissata alla cinghia del soggetto. Adattata da [57].

Sono di seguito descritte alcune caratteristiche delle tre tipologie di segnali acquisiti.

- Segnali basografici

I segnali basografici descrivono il contatto piede-suolo, da cui sono ricavate la sequenza e la durata di ciascuna fase del ciclo del passo [56]. I tre segnali digitali (acquisiti rispettivamente dai tre foot-switches) sono combinati mediante un convertitore D/A per ottenere $2^3 = 8$ differenti condizioni di supporto (segnale a otto livelli). Il segnale a otto livelli è però troppo variabile e “dettagliato” per essere sfruttato in un’analisi statistica del cammino ed è, pertanto, tipicamente semplificato in un segnale a quattro livelli [13]. Nelle elaborazioni STEP32 adotta la basografia a quattro livelli [56], che corrispondono alle fasi H, F, P, S. Come dettagliato in [13] e illustrato in **Figura 2.4.**: durante la fase H soltanto l’interruttore sotto il tallone è chiuso; durante la fase F sono chiusi l’interruttore sotto il tallone e almeno uno degli interruttori sotto l’avampiede; durante la fase P l’interruttore sotto il tallone è aperto e almeno uno degli interruttori sotto l’avampiede è chiuso; durante la fase S tutti gli interruttori sono aperti.

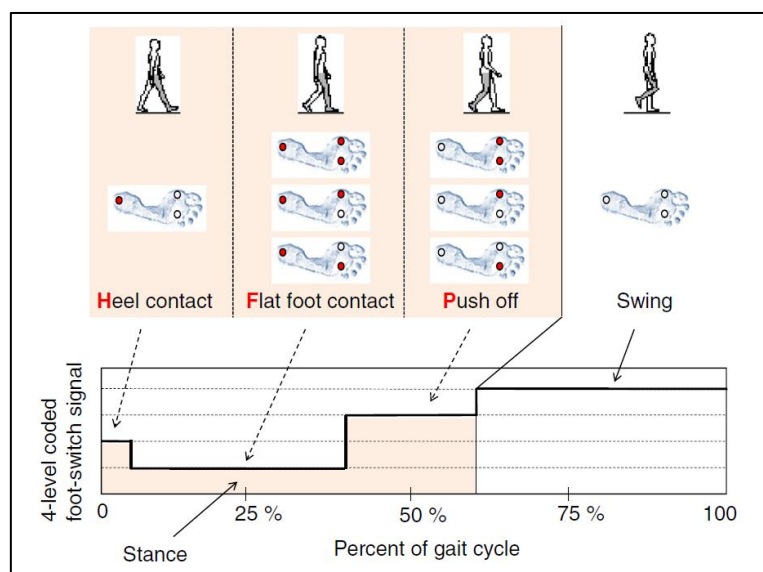


Figura 2.4. Combinazioni on/off dei tre foot-switches e corrispondenti fasi del ciclo del passo (piede destro, codifica a 4 livelli). Sotto la pianta del piede, il cerchio rosso rappresenta l'interruttore chiuso, il cerchio bianco l'interruttore aperto. Il segnale basografico possiede 4 livelli di quantizzazione: 1) interruttore del tallone chiuso (Heel contact); 2) interruttore del tallone e almeno uno dell'avampiede chiusi (Flat foot contact); 3) almeno un interruttore dell'avampiede chiuso (Push off); 4) nessun interruttore chiuso (Swing). Da [3].

- Segnali goniometrici

I segnali goniometrici acquisiti forniscono i dati relativi alla misura degli angoli articolari di ginocchio sul piano sagittale. Durante il cammino, il movimento dell'articolazione di ginocchio nel piano sagittale si sviluppa in termini di flessione ed estensione. La flessione-estensione del ginocchio si ripete in modo ciclico. Come visibile in **Figura 2.5.**, il ginocchio mostra due picchi di flessione e due picchi di estensione durante ciascun ciclo del passo [14]. In coincidenza dell'heel contact, il ginocchio è esteso. Dopo si flette per assorbire il carico che grava sull'arto inferiore. In seguito a questo primo picco di flessione, si estende fino alla quasi totale estensione. Poi inizia la seconda fase di flessione, mentre l'arto inferiore è in propulsione. Il ginocchio continua a flettersi in preparazione alla fase di swing. Il toe off si verifica quando la flessione di ginocchio individua un angolo di 40° . Il ginocchio continua a flettersi fino ad un massimo di $60^\circ - 70^\circ$ e poi si estende per prepararsi per il successivo heel contact.

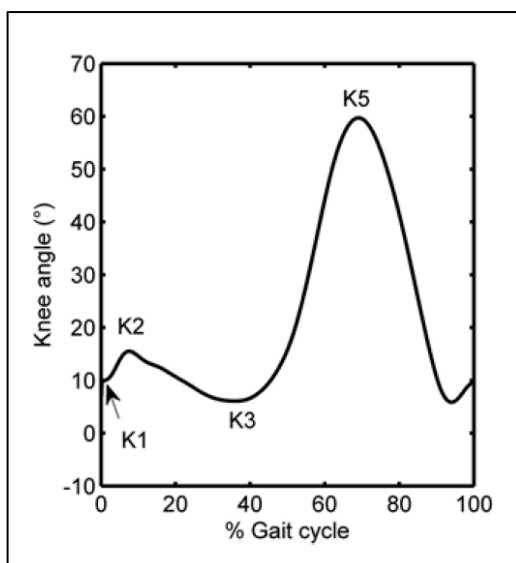


Figura 2.5. Parametrizzazione della curva della cinematica dell'articolazione di ginocchio. K1 indica l'angolo in coincidenza dell'heel contact; K2 e K3, rispettivamente, l'angolo massimo e l'angolo minimo durante la fase di stance; K5 la flessione massima (massimo assoluto). Da [62].

- Segnali elettromiografici di superficie (sEMG)

I segnali elettromiografici di superficie (sEMG) sono registrazioni dell'attività elettrica dei muscoli che si contraggono, acquisite tramite elettrodi di superficie. I segnali sEMG sono tipicamente caratterizzati da ampiezza dell'ordine delle centinaia di microvolt e banda in frequenza compresa tra 10 hertz e 500 hertz. L'elettromiografia di superficie rappresenta un approccio non invasivo, sfruttabile per monitorare l'attività muscolare durante il cammino [1]. I muscoli degli arti inferiori e del tronco svolgono funzioni biomeccaniche specifiche durante il ciclo del passo [58]. In coincidenza dell'appoggio-iniziale del piede, il bicipite femorale controlla il movimento di flessione dell'anca, mentre il tibiale anteriore rallenta il movimento verso il suolo del piede. Durante il primo periodo di doppio supporto, alcuni muscoli controllano le articolazioni di caviglia, ginocchio e anca per mantenere l'equilibrio durante l'avanzamento. Il retto del femore svolge la funzione di estensore al fine di rallentare la flessione del ginocchio. L'azione del bicipite femorale si riduce (per flettere il ginocchio) e il gluteo medio stabilizza il bacino. Alla fine della fase di stance, il tibiale anteriore prepara l'inizio della fase di swing. Durante la fase di swing, i muscoli flessori d'anca permettono l'avanzamento della coscia e la flessione del ginocchio. Il bicipite femorale incrementa la flessione del ginocchio e il tibiale anteriore prepara il raggiungimento del distacco dal suolo e mantiene la posizione della caviglia. Il gluteo medio controlaterale supporta la posizione del bacino. Alla fine della fase di swing, il bicipite femorale controlla le articolazioni di anca e ginocchio per rallentare il movimento in avanti della gamba. Il retto del femore estende il ginocchio e il tibiale anteriore posiziona la caviglia per assicurare il contatto con il suolo.

I 12 muscoli del tronco e di un unico arto inferiore da cui sono acquisiti i segnali sEMG, indicati in **Figura 2.6.**, svolgono funzioni biomeccaniche differenti [8], [14]: LD_L e LD_R (livello L4) controllano la posizione del tronco nel piano frontale in coincidenza dell'heel

contact del piede omolaterale e controlaterale; GMD e TFL stabilizzano l'articolazione d'anca durante l'heel contact e la fase di accettazione del carico; LH e MH decelerano la gamba alla fine della fase di swing; RF e VM estendono il ginocchio, rallentandone la flessione successiva all'heel contact; LGS, SOL e PL controllano la propulsione durante la fase di stance; TA controlla la distanza da terra dell'avampiede durante la fase di swing e il piede durante il primo contatto con il suolo.

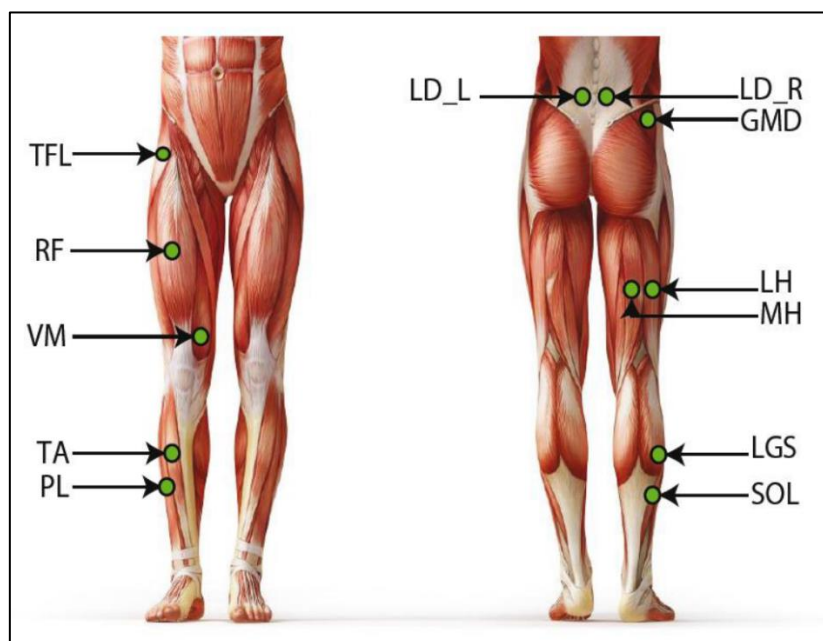


Figura 2.6. Individuazione della posizione dei 12 muscoli (tronco e un unico arto inferiore) da cui sono acquisiti i segnali sEMG. I muscoli sono i seguenti: left Longissimus Dorsii (LD_L), right Longissimus Dorsii (LD_R), Gluteus Medius (GMD), Tensor Fasciae Latae (TFL), Lateral Hamstring (LH), Medial Hamstring (MH), Rectus Femoris (RF), Vastus Medialis (VM), Lateral Gastrocnemius (LGS), Soleus (SOL), Tibialis Anterior (TA), Peroneus Longus (PL). Da [57].

2.3 Definizione del dataset

La raccolta dei trials, in termini di selezione dei trials da includere nel dataset, è caratterizzata da una fase iniziale di verifica della loro qualità. A tal fine, innanzitutto, viene visualizzata la videoregistrazione del singolo trial, su cui si basa la corrispettiva segmentazione manuale. In caso di videoregistrazione corrotta, il trial viene scartato. Altrimenti, vengono ispezionati visivamente i segnali sEMG, acquisiti durante l'esecuzione del trial stesso, per verificare che non siano eventualmente corrotti (ad esempio, per la presenza di canali con saturazione) e che presentino un rapporto segnale/rumore "soddisfacente". In seguito, viene effettuata la segmentazione manuale del trial selezionato.

Al fine di ridurre la complessità dell'informazione che la rete neurale deve gestire, i segnali sEMG non sono mantenuti grezzi: essi vengono pre-elaborati, estraendone i rispettivi involucri.

Complessivamente, vengono raccolti 51 trials, dei quali 31 eseguiti dai soggetti sani (18 trials dai soggetti giovani e 13 trials dai soggetti adulti) e 20 dai soggetti parkinsoniani (Figura 2.7.).

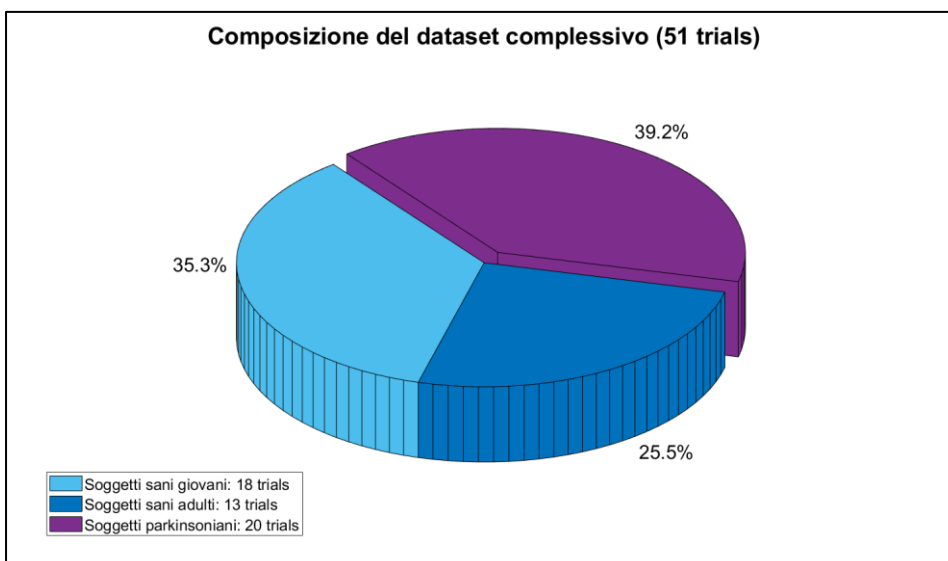


Figura 2.7. Visualizzazione della composizione del dataset complessivo di 51 trials rispetto alle tre "categorie" di soggetti (sani giovani, sani adulti, affetti dal morbo di Parkinson).

Per ciascun soggetto, il dataset include il trial eseguito in condizione di cammino normale e, soltanto per i soggetti sani giovani, anche il trial eseguito in condizione di cammino veloce. Si tratta di 42 trials con andatura normale e 9 trials con andatura veloce.

Successivamente, ai fini dell'implementazione della rete neurale, le osservazioni sono ottenute a partire dai segnali sEMG pre-elaborati ed il target a partire dalle segmentazioni manuali.

2.3.1 Segmentazione manuale

Il singolo trial viene segmentato ed etichettato, in termini di ground truth, manualmente, al fine di disporre di un target adeguatamente robusto per l'implementazione della rete neurale. La segmentazione manuale si articola in due passi fondamentali:

1. posizionamento di markers di delimitazione dei tratti rettilinei/non-rettilinei;
2. costruzione del ground truth usato come input per l'allenamento della rete.

La segmentazione manuale è effettuata a partire dalla visualizzazione della videoregistrazione del trial mediante il software di STEP32. Contemporaneamente alla visualizzazione del video, il software permette il posizionamento manuale dei markers sulle tracce dei segnali acquisiti, come visibile in **Figura 2.8**.

Dopo aver posizionato un marker, viene associata manualmente una label che identifica la classe di appartenenza del tratto precedente.

Le labels sono codificate come di seguito indicato:

- la label 0 identifica la classe dei tratti non-rettilinei;

- la label 1 identifica la classe dei tratti rettilinei;
- la label 2 identifica tutti i tratti, rettilinei oppure non-rettilinei, “compromessi” (perché, ad esempio, nel percorrerli il soggetto si è fermato, per poi ripartire, a causa di un qualche interruttore basografico non perfettamente fissato).

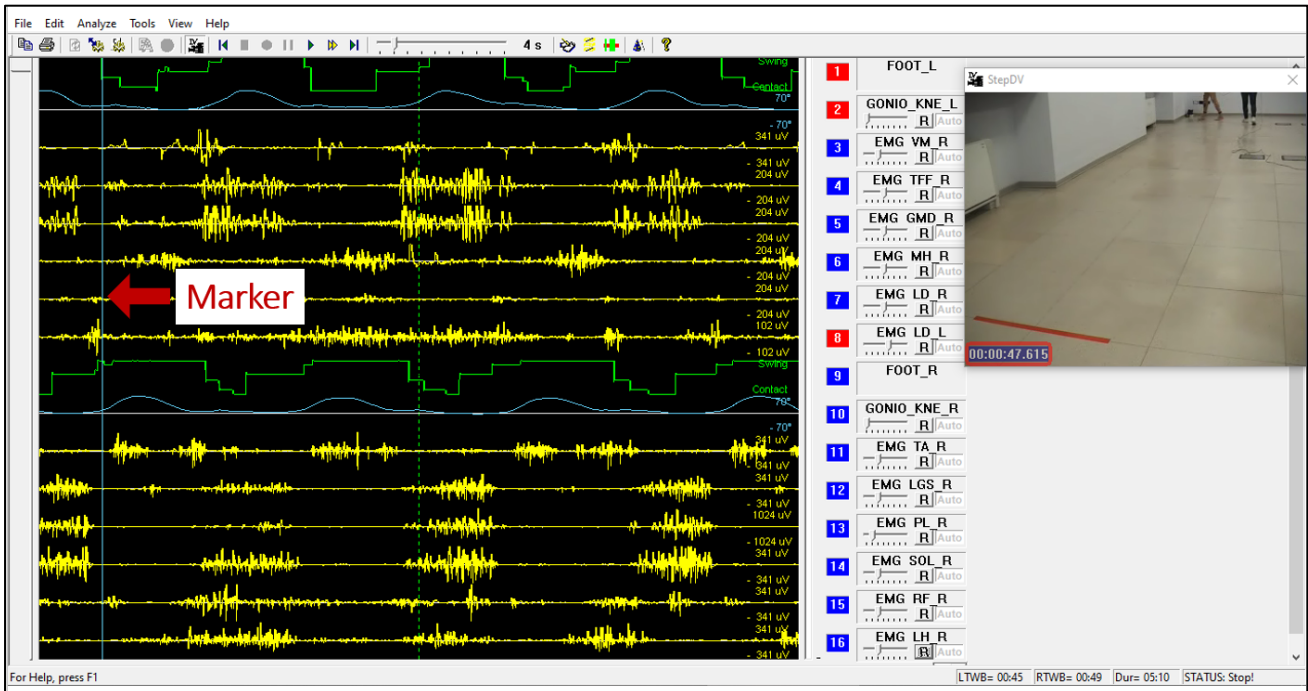


Figura 2.8. Software di STEP32. A sinistra, le tracce dei 16 segnali acquisiti (2 basografici + 2 goniometrici di ginocchio + 12 sEMG). A destra, la visualizzazione della videoregistrazione del trial. Come esempio, è indicato un marker posizionato durante la segmentazione manuale.

I tratti etichettati con la label 2 vengono esclusi dal dataset. Di fatto, la label 2 è soltanto uno strumento di filtraggio manuale al fine di costruire un ground truth robusto. La rete neurale rimane un classificatore binario rispetto alle classi 0 e 1 e, in termini di training set, validation set e testing set, non riceve in input osservazioni etichettate con la label 2.

Il ground truth è la sequenza di labels ottenuta alla fine della segmentazione manuale, includendo, però, soltanto le occorrenze di label 0 e di label 1 ed escludendo le occorrenze di label 2.

Lungo il percorso che il soggetto replica durante l'esecuzione di un trial, vengono posizionati quattro markers: rispettivamente, in A, B, C, D, come in **Figura 2.9**. Il primo marker della segmentazione manuale è posizionato in A, l'ultimo in B. In **Figura 2.10** è riportato il diagramma di flusso della segmentazione manuale del singolo trial.

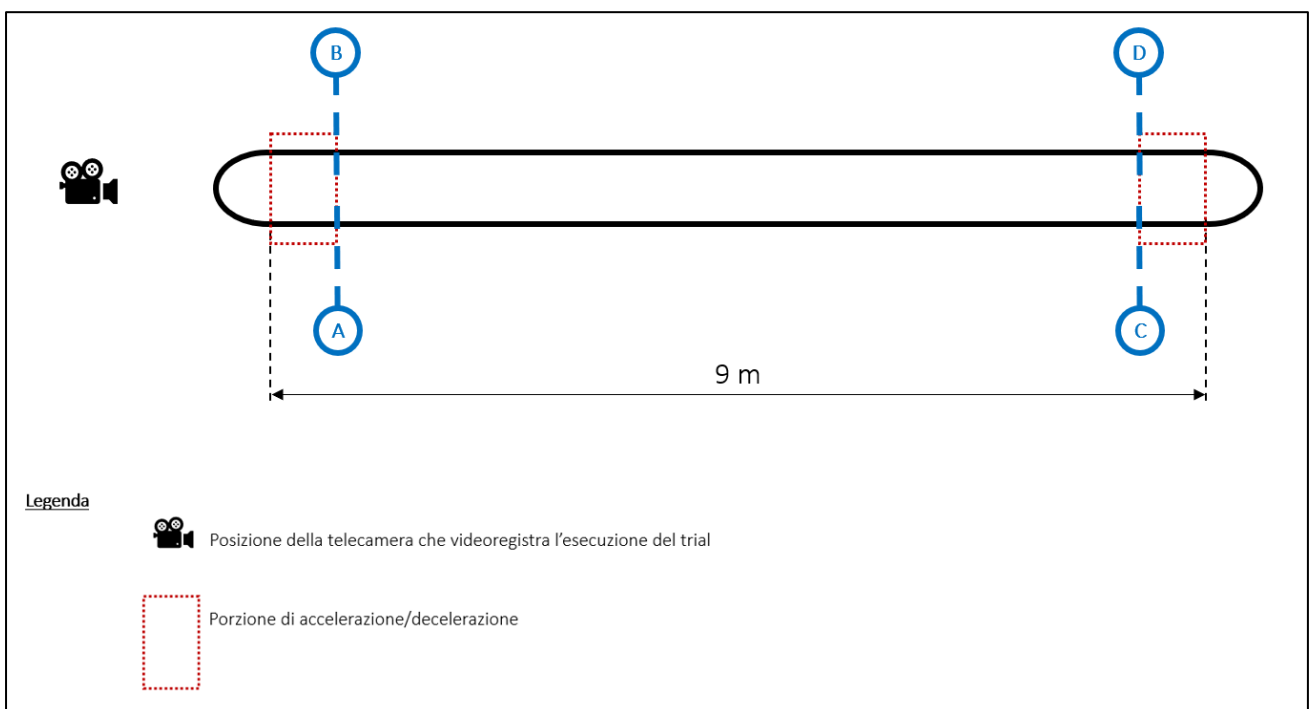
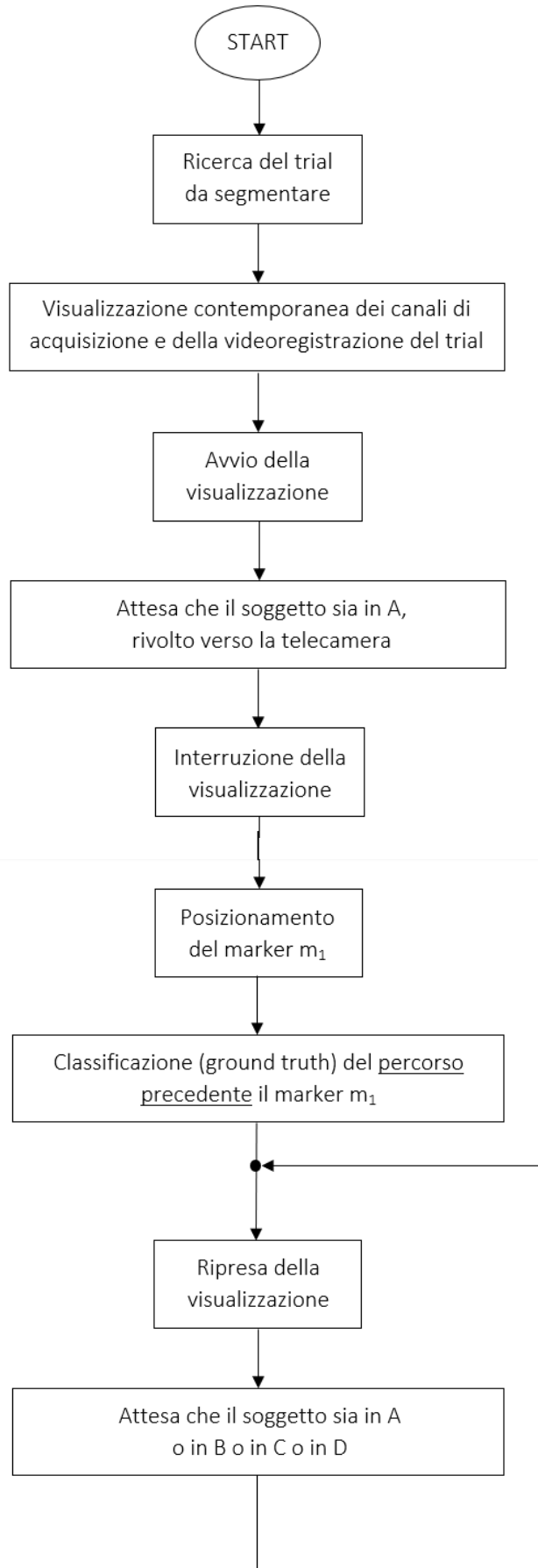


Figura 2.9. Posizionamento dei markers della segmentazione manuale lungo il percorso (in A, B, C, D). Il primo marker è posizionato in A e l'ultimo in B.



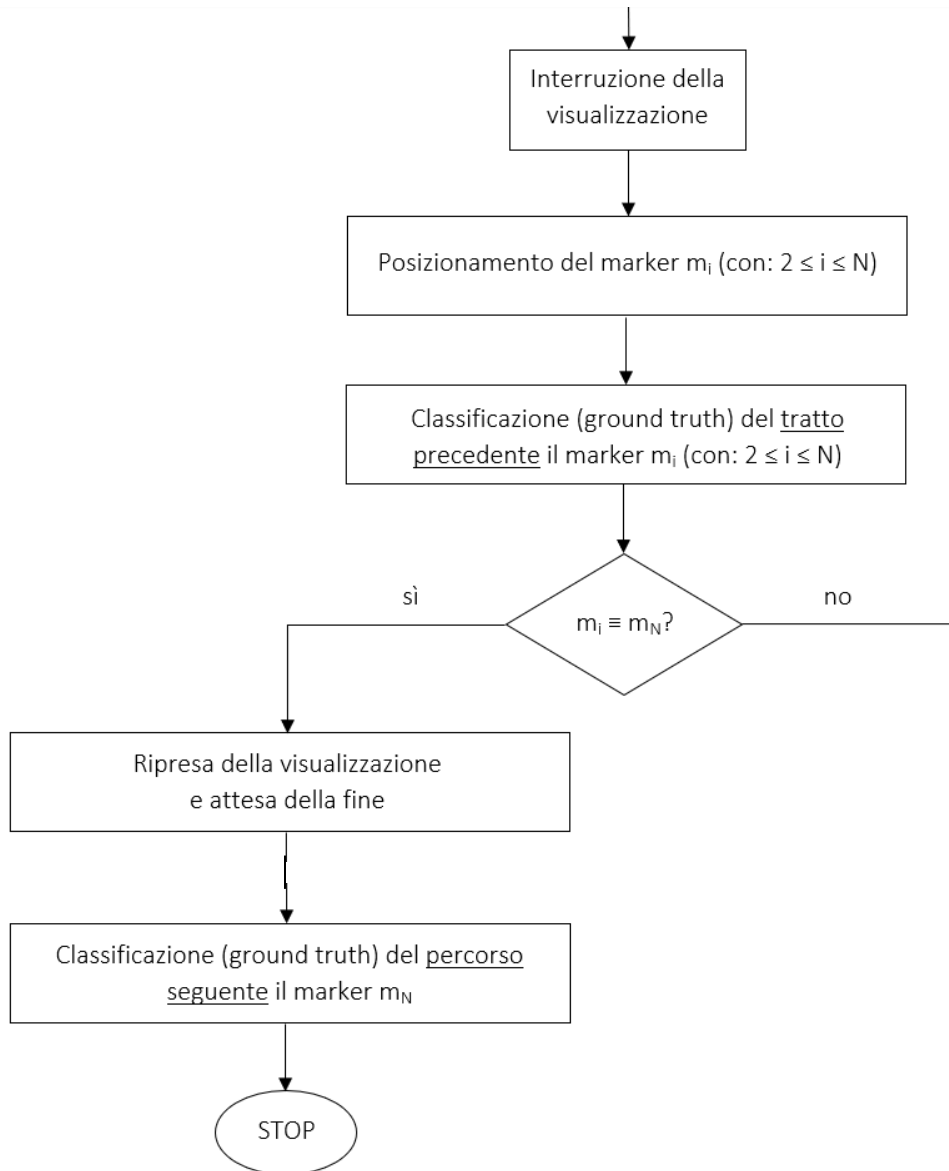


Figura 2.10. Diagramma di flusso della segmentazione manuale.

2.3.2 Pre-Processing dei segnali sEMG

I segnali sEMG acquisiti sono, innanzitutto, filtrati passa-banda mediante il filtro digitale implementato in STEP32 (banda passante tra 10 Hz e 400 Hz). Sono poi importati in MATLAB® release R2019b (The MathWorks Inc., Natick, MA, USA) e pre-elaborati al fine

di estrarre l'involuppo per ciascun canale sEMG. Tale algoritmo di pre-processing consta dei seguenti tre passi [57]:

1. Rettificazione, calcolando il valore assoluto del segnale sEMG, per ottenere un segnale che non abbia valore medio pari a 0.
2. Filtraggio passa-basso, ovvero estrazione dell'involuppo, per ottenere una curva "smooth" che evidenzi il trend del segnale rettificato.
3. Normalizzazione in ampiezza del singolo canale rispetto al suo valore massimo, così da evitare sbilanciamenti tra i canali.

L'ordine (pari a 2) e la frequenza di taglio (pari a 5 Hz) del filtro passa-basso Butterworth sono stati scelti confrontando visivamente le maschere di otto filtri Butterworth, considerando per coppie i valori riportati in **Tabella 2.1.**

Parametri	Valori
Ordine	2, 4, 6, 8
Frequenza di taglio (Hz)	5, 10

Tabella 2.1. Valori considerati per scegliere i parametri del filtro passa-basso Butterworth.

La coppia così selezionata è adottata anche in [1] per estrarre gli involuppi di segnali sEMG degli arti inferiori. In **Figura 2.11.**, è rappresentata la risposta in frequenza del filtro passa-basso, che è applicato a doppia passata (routine MATLAB® "filtfilt") al fine di minimizzare gli effetti della distorsione di fase.

In **Figura 2.12.**, per un soggetto sano e per un soggetto patologico, sono mostrati il segnale sEMG importato da STEP32 (filtrato passa-banda) e il risultato dell'algoritmo di pre-processing. Per chiarezza di visualizzazione, viene isolato, come esempio, un unico canale sEMG (tibiale anteriore).

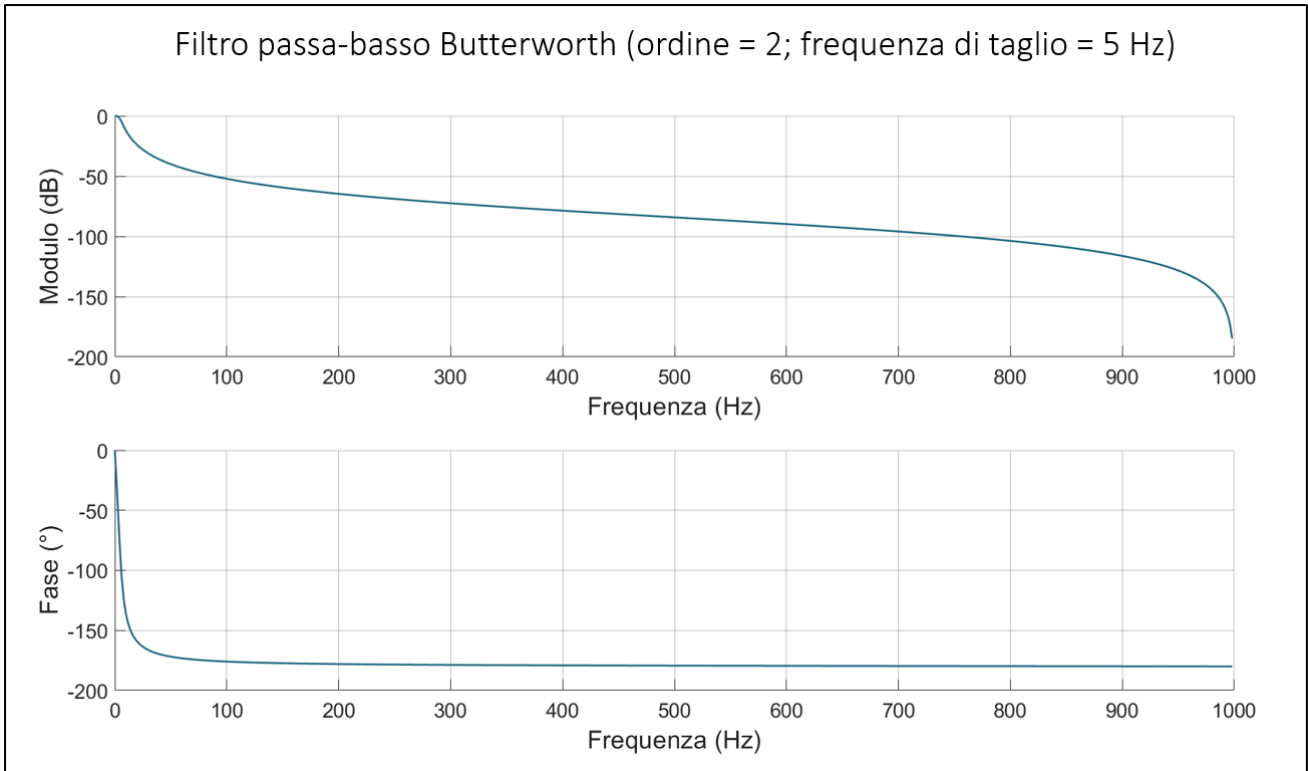
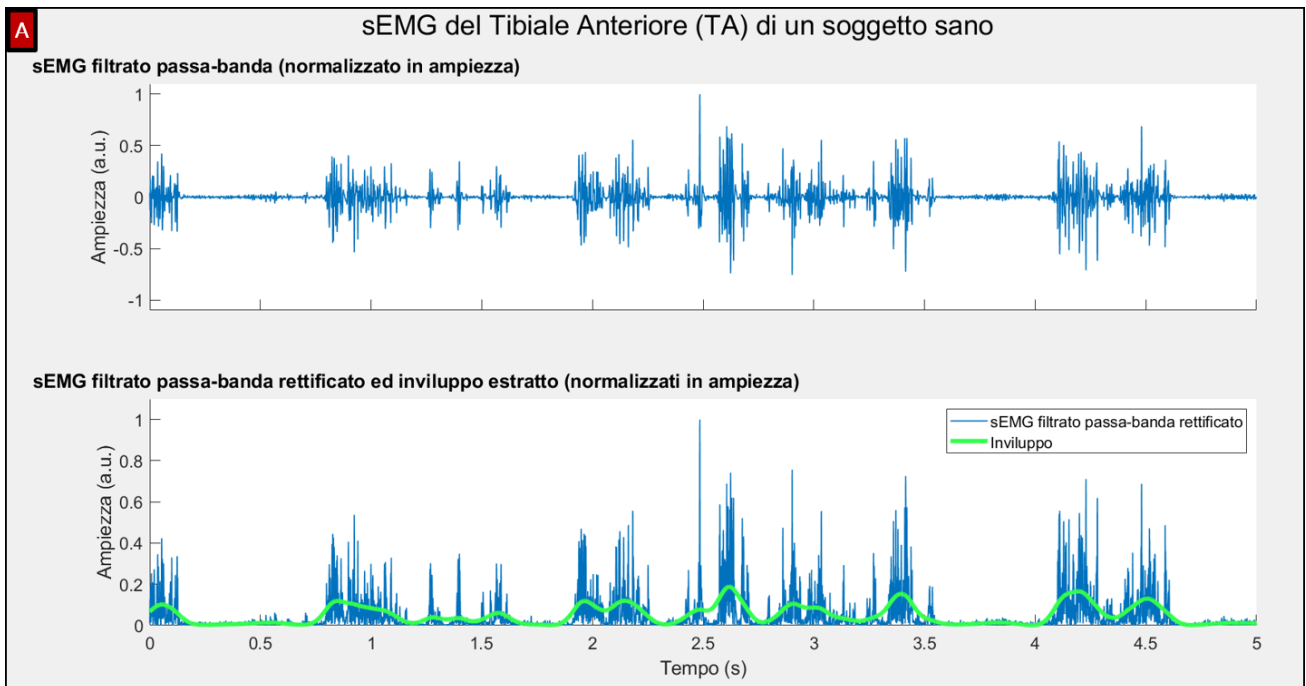


Figura 2.11. Risposta in frequenza del filtro passa-basso Butterworth del secondo ordine con frequenza di taglio 5 Hz. In alto, il modulo (in decibel); in basso, la fase (in gradi). Gli assi orizzontali non sono normalizzati.



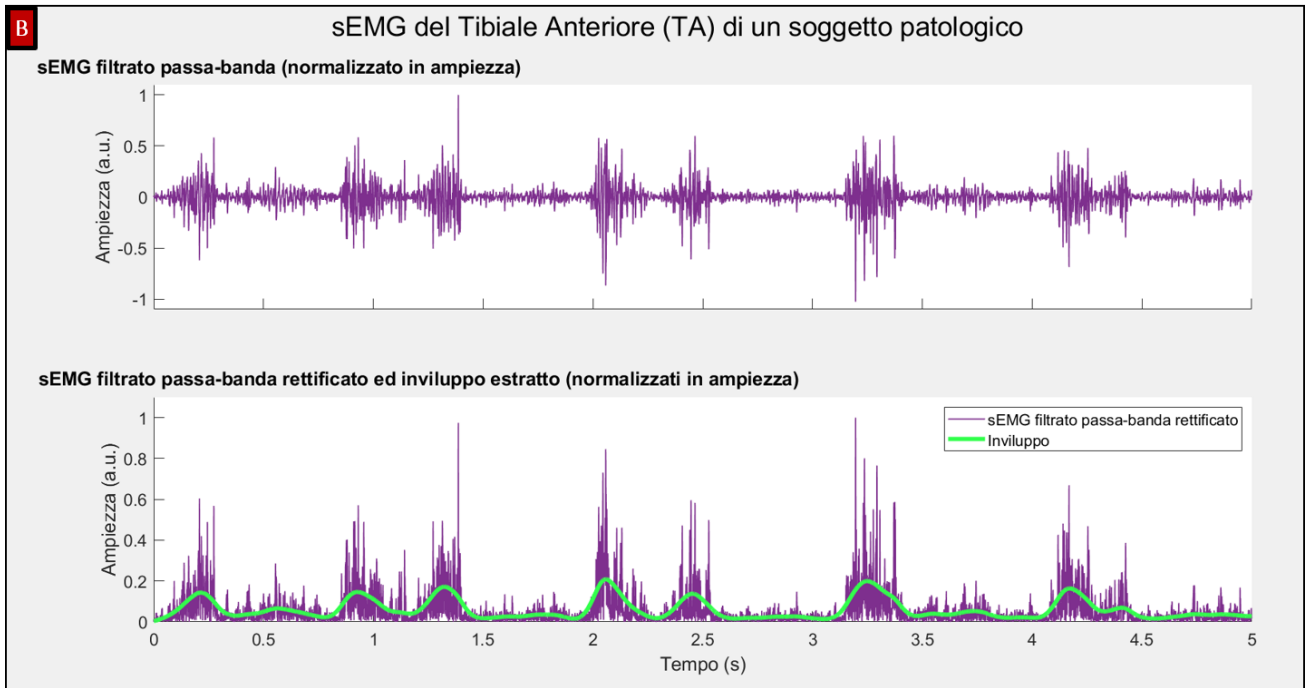


Figura 2.12. Pre-Processing del segnale sEMG. Esempio del Tibiale Anteriore (TA) di un soggetto sano (A) e di un soggetto patologico (B) su una finestra temporale di 5 s. Per ciascun soggetto sono mostrati, in alto, il segnale importato da STEP32 (filtrato passa-banda) normalizzato in ampiezza e, in basso, normalizzata in ampiezza, la sovrapposizione di segnale rettificato ed inviluppo estratto.

2.4 Esplorazione dell'applicabilità delle reti neurali BiLSTM al problema in esame

L'applicabilità delle reti neurali BiLSTM al problema in esame viene esplorata su un dataset ridotto, che comprende soltanto 26 trials di soggetti sani (83.9% dei trials della sottopopolazione sana). L'analisi viene sviluppata in termini di confronto tra due approcci di classificazione, il Sequence-to-Label (STL) ed il Sequence-to-Sequence (STS), implementati in ambiente MATLAB®. I due approcci permettono di elaborare il problema di classificazione binaria della tesi a partire dall'individuazione, lungo il percorso descritto dal soggetto che cammina (Figura 2.13.), di tratti e di transizioni.

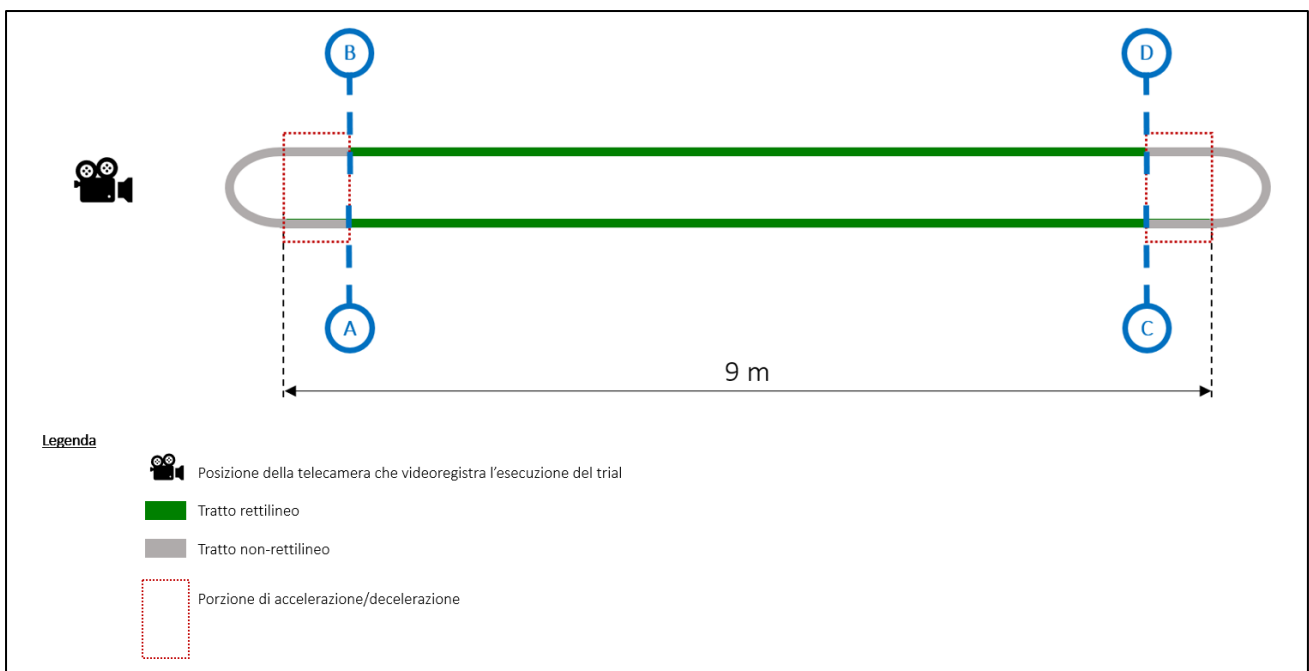


Figura 2.13. Percorso del protocollo sperimentale. I tratti sono individuati mediante due colori differenti: rettilinei in verde, non-rettilinei in grigio. Le quattro transizioni si sviluppano, rispettivamente, a cavallo di A, B, C, D. Ciascuna transizione include due tratti di classe differente.

Un tratto può essere rettilineo oppure non-rettilineo. Quest'ultimo coincide con la traiettoria "a U" del cambio di direzione a 180°. Sostanzialmente, un tratto non-rettilineo include una curva, la decelerazione che la precede e l'accelerazione che la segue. A rigore, la decelerazione e l'accelerazione, inglobate in un tratto non-rettilineo, si manifestano lungo il rettilineo, all'estremità. Una transizione è una porzione di percorso a cavallo tra due tratti consecutivi. Complessivamente, sono presenti quattro transizioni diverse: in A, B, C, D.

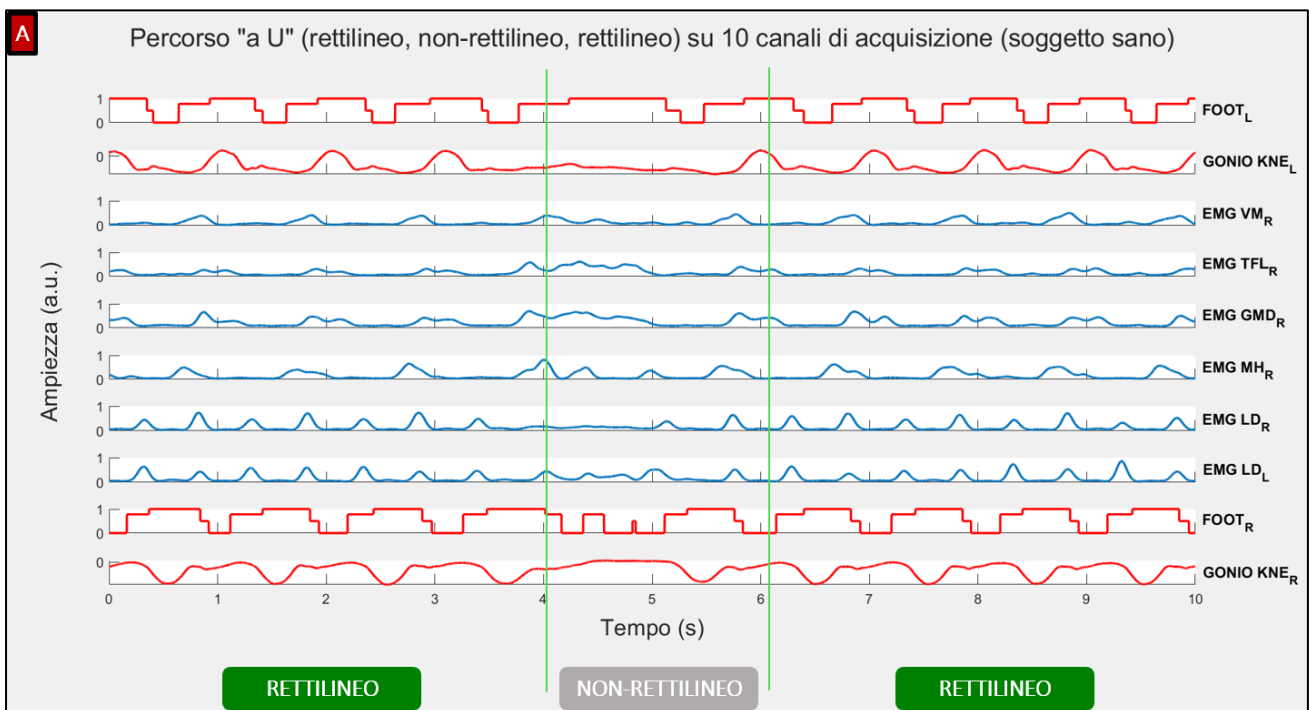
La differenza fondamentale tra le transizioni è rappresentata dal verso di percorrenza, orario oppure antiorario. Dal punto di vista biomeccanico, infatti, se essa viene percorsa in senso orario, il soggetto fa perno sulla gamba destra e attiva maggiormente i muscoli della gamba sinistra; viceversa, se in senso antiorario. Le quattro transizioni sono tali per cui:

- in A: passaggio da rettilineo a non-rettilineo in verso orario;
- in B: passaggio da non-rettilineo a rettilineo in verso orario;
- in C: passaggio da rettilineo a non-rettilineo in verso antiorario;
- in D: passaggio da non-rettilineo a rettilineo in verso antiorario.

Di fatto, mantenendo come obiettivo la distinzione tra la classe dei rettilinei e la classe dei non-rettilinei, il problema di classificazione binaria è scorporato in due sotto-problemi, sviluppati, rispettivamente, in termini di analisi dei tratti e di analisi delle transizioni. I tratti sono mono-classe, le transizioni sono bi-classe.

L'analisi dei tratti e l'analisi delle transizioni sono sviluppate in parallelo, in modo del tutto autonomo ed indipendente l'una dall'altra. Per l'analisi dei tratti è adottato l'approccio STL, per l'analisi delle transizioni è adottato l'approccio STS. Di conseguenza, anche il dataset viene costruito in maniera sdoppiata. Sostanzialmente, si tratta di due dataset distinti, uno per singolo approccio.

In entrambi gli approcci il classificatore viene implementato come una black box: la rete neurale estrae ed apprende features direttamente dai segnali sEMG pre-elaborati (cfr. 2.3.2), senza alcuna “features engineering” esterna ad essa. In **Figura 2.14.**, sono mostrati, rispettivamente per un trial eseguito da un soggetto sano e per uno eseguito da un soggetto patologico, due esempi di percorso “a U” (rettilineo, non-rettilineo, rettilineo), visualizzati rispetto ai segnali basografici e goniometrici di ginocchio (bilaterali) e a 6 canali sEMG (di un arto inferiore e del tronco). Sono rappresentati soltanto 6 dei 12 canali sEMG, selezionati casualmente, al fine di una migliore visualizzazione. Il classificatore apprende esclusivamente dagli involucri del segnale sEMG (un involuppo per singolo canale) e non conosce il contenuto informativo dei segnali basografici e goniometrici di ginocchio, in quanto non allenato su di essi.



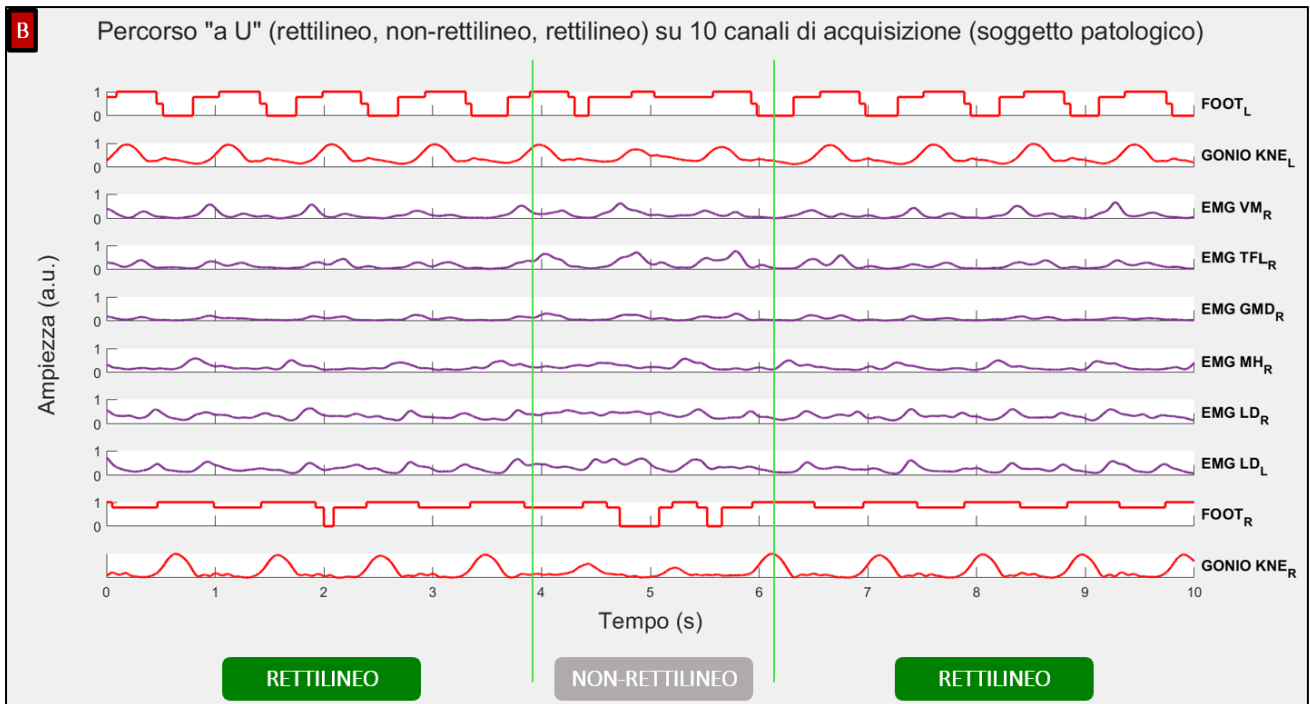


Figura 2.14. Esempio di percorso "a U" (rettilineo, non-rettilineo, rettilineo) su una finestra temporale di 10 s. Per un trial della sottopopolazione sana (A) e per un trial della sottopopolazione patologica (B), sono mostrati soltanto 10 dei 16 canali di acquisizione: 2 basografici (in rosso), 2 goniometrici di ginocchio (in rosso), 6 sEMG (inviluppi - A, in blu; B, in viola). I segnali sono normalizzati in ampiezza.

Sia approccio STL che in approccio STS, un'osservazione coincide con una "sequenza" (ad esempio, un brano di segnale sEMG, in quanto serie temporale di campioni). In termini di target e di output, a ciascuna sequenza è associata una label unica, che ne identifica la classe di appartenenza, in approccio STL e un vettore di labels (una label per ogni campione della sequenza) in approccio STS. Sostanzialmente, STL e STS differiscono in risoluzione temporale della classificazione: STL associa un'unica classe a tutta la sequenza temporale, STS il singolo campione di una sequenza.

2.4.1 Costruzione dei dataset

Sia in analisi dei tratti (approccio STL) che in analisi delle transizioni (approccio STS), la costruzione del dataset è sviluppata in termini di organizzazione delle osservazioni e definizione del target. Rispettivamente, le osservazioni a partire dagli involucri dei segnali sEMG ed il target a partire dal ground truth della segmentazione manuale. Il segnale sEMG acquisito da un soggetto è contenuto in una matrice, come in **Figura 2.15.**, avente numero di righe pari al numero di canali acquisiti ed un numero di colonne pari al numero di campioni acquisiti. Il numero di righe delle corrispettive matrici è pari a 12 (ovvero una riga per muscolo). Ciascuno dei 12 canali sEMG rappresenta una feature.

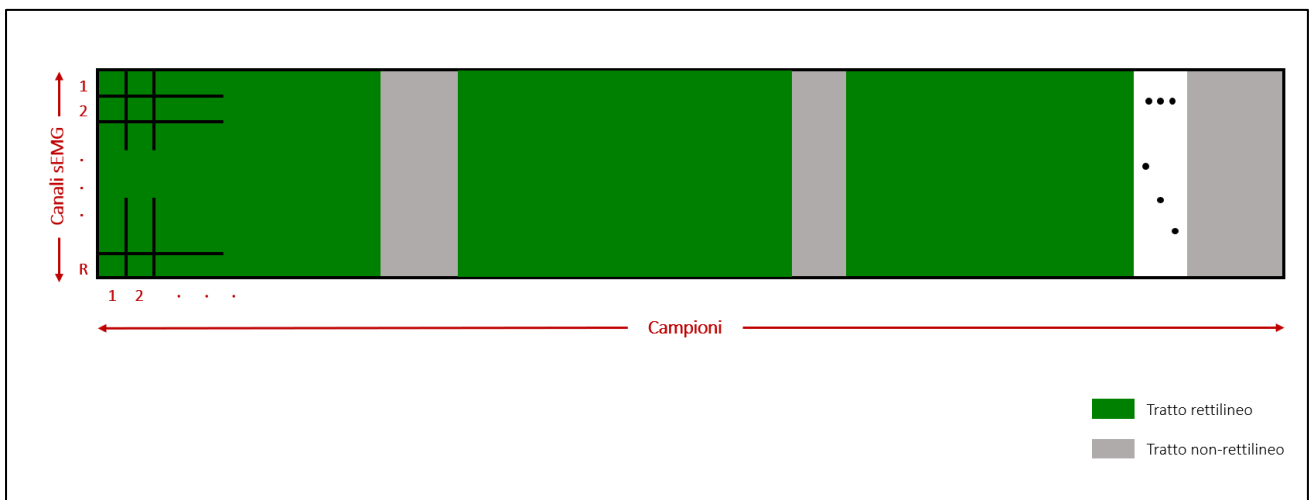


Figura 2.15. Matrice contenente i segnali sEMG. Il numero di righe coincide con il numero di canali sEMG di acquisizione, ovvero con il numero di features ($R = 12$). Il numero di colonne coincide con il numero di campioni acquisiti. I colori in alternanza indicano la successione delle due classi di tratti: rettilinei in verde, non-rettilinei in grigio. (Ovviamente, ciascun tratto, rettilineo oppure non-rettilineo, include un numero di colonne, almeno potenzialmente, diverso).

In approccio STL, il segnale sEMG, relativo al singolo trial, viene ritagliato in porzioni, ciascuna delle quali coincidente con il segnale acquisito nel percorrere un tratto

(rettilineo oppure non-rettilineo). Come illustrato in **Figura 2.16.**, una porzione di sEMG è di fatto una sottomatrice avente 12 righe e numero di colonne pari al numero di campioni acquisiti lungo il tratto stesso. Una sottomatrice costituisce un'osservazione per la rete neurale. Il target relativo ad una singola osservazione è una label che ne identifica la classe di appartenenza (0 per la classe dei tratti non-rettilinei, 1 per la classe dei tratti rettilinei).

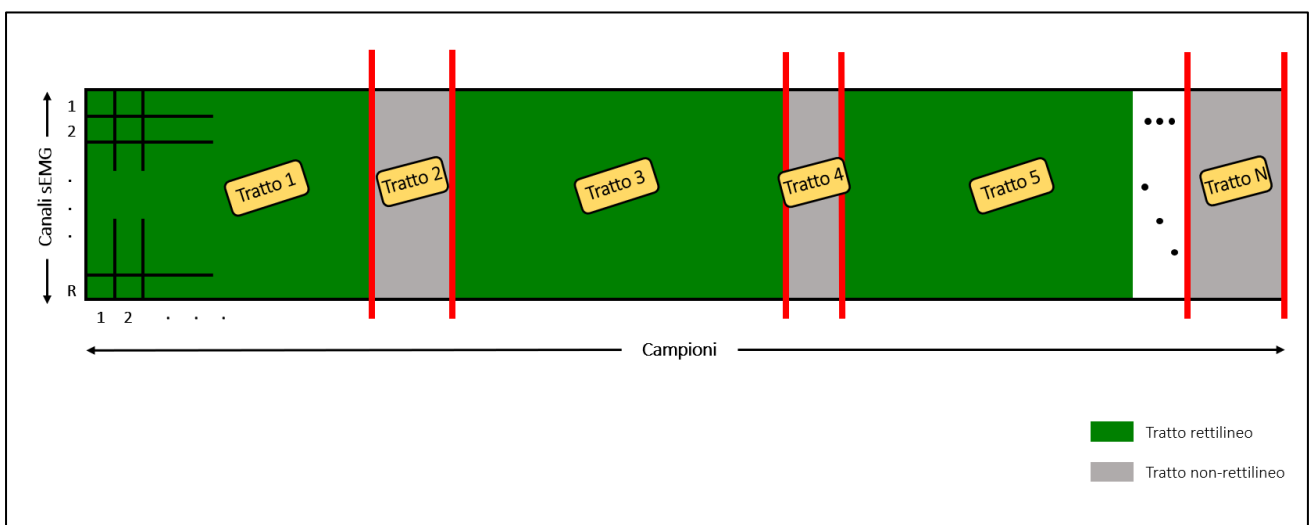


Figura 2.16. Matrice contenente il segnale sEMG con individuazione dei tratti (separati mediante le linee rosse). In approccio STL, un'osservazione coincide con la porzione di matrice corrispondente ad un tratto. (Ovviamente, ciascun tratto, rettilineo oppure non-rettilineo, include un numero di colonne, almeno potenzialmente, diverso).

In approccio STS, il segnale sEMG, relativo al singolo trial, viene ritagliato in porzioni, ciascuna delle quali coincidente con il segnale acquisito nel percorrere una transizione. Come illustrato in **Figura 2.17.**, una porzione di sEMG è di fatto una sottomatrice avente 12 righe e numero di colonne pari al numero di campioni acquisiti lungo una transizione. Una sottomatrice costituisce un'osservazione per la rete neurale. Il target relativo ad una singola osservazione è un vettore di labels ciascuna delle quali identifica la classe di appartenenza (0 per la classe dei tratti non-rettilinei, 1 per la classe dei tratti rettilinei) del singolo campione lungo una transizione. Sostanzialmente, in approccio STS il target

di un trial è costruito a partire dal ground truth della segmentazione manuale del trial stesso, replicando campione per campione le labels del ground truth.

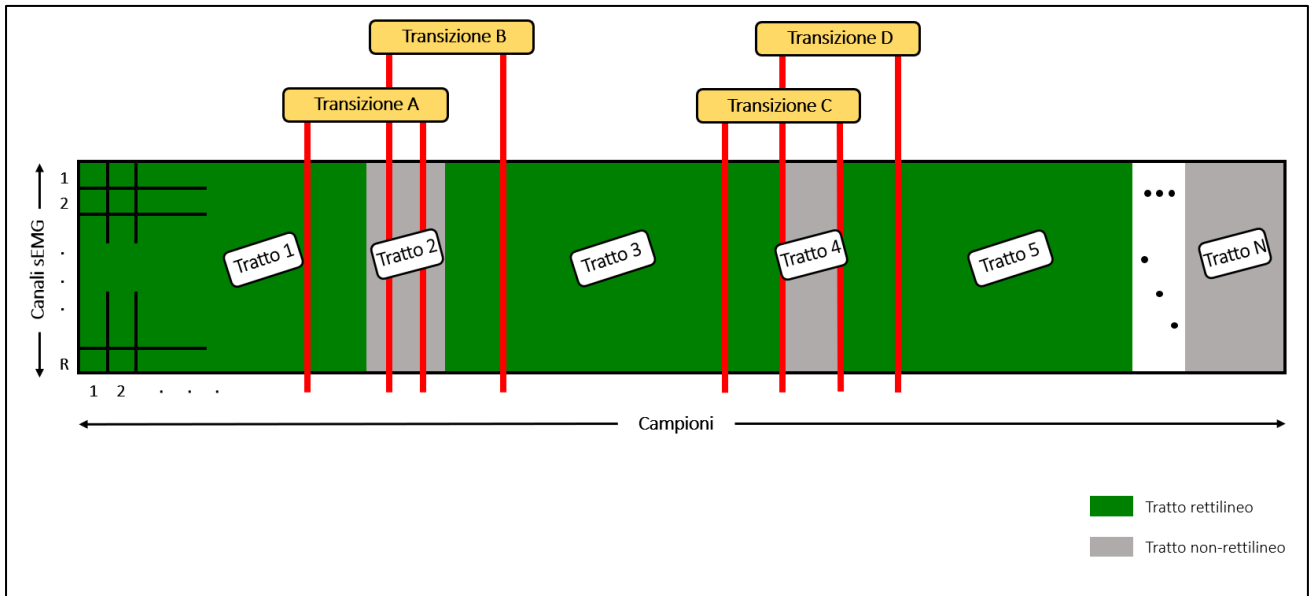


Figura 2.17. Matrice contenente il segnale sEMG con individuazione delle transizioni (separate mediante le linee rosse). In approccio STS, un'osservazione coincide con la porzione di matrice corrispondente ad una transizione. (Ovviamente, ciascun tratto, rettilineo oppure non-rettilineo, include un numero di colonne, almeno potenzialmente, diverso).

In **Figura 2.18.**, è schematizzato il confronto tra i due approcci rispetto ad osservazioni e target.

Ciascuno dei due dataset include l'insieme di tutte le osservazioni di tutti i trials e il corrispettivo target. Entrambi i dataset sono bilanciati rispetto alle due classi in maniera differente:

- In approccio STL, il bilanciamento è unico, garantendo che le osservazioni di ciascuna classe rappresentino il 50% dell'intero dataset.
- In approccio STS, il bilanciamento è doppio, garantendo, innanzitutto, che le osservazioni di ciascuna tipologia di transizione (in A, B, C, D) rappresentino il 25% dell'intero dataset e, poi, che ciascuna osservazione sia in sé bilanciata rispetto

alle due classi, ovvero che essa contenga il 50% del numero di colonne appartenente alla singola classe.

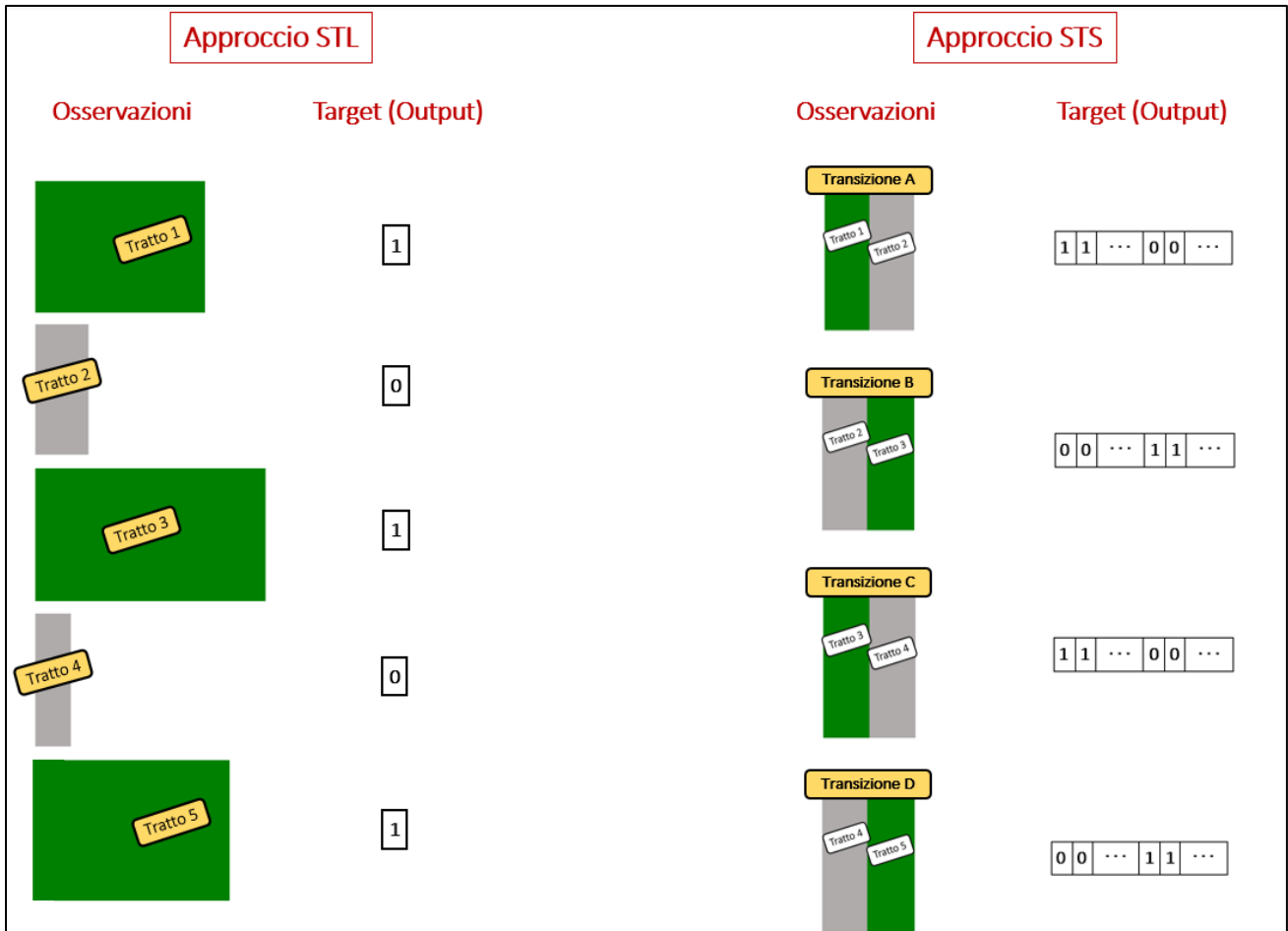


Figura 2.18. Confronto tra i due approcci (STL, STS) rispetto ad osservazioni e target. L'output della rete neurale possiede la stessa forma del target. In STL, un'osservazione è una porzione di matrice di sEMG ritagliata sul singolo tratto. Come target, ad essa è associata la label (0 oppure 1) che ne identifica la classe. In STS, un'osservazione è una porzione di matrice di sEMG ritagliata sulla singola transizione. Come target, ad essa è associato un vettore di labels (0 e 1), una per singola colonna della matrice.

2.4.2 Definizione del modello BiLSTM

L'applicabilità delle reti neurali BiLSTM al problema in esame è valutata implementando 80 modelli BiLSTM, in modo parallelo tra gli approcci STL e STS. Il parallelismo è mantenuto sia rispetto alla ripartizione del dataset sia rispetto all'architettura ed al tuning degli iperparametri delle reti.

Ciascuno dei due dataset, includendo soltanto 26 di 31 trials della sottopopolazione sana e nessuno della sottopopolazione patologica, è ripartito randomicamente in testing set (70%), validation set (15%) e testing set (15%). I tre sets non possiedono osservazioni in comune. Per ragioni di tempi computazionali elevati, viene adottata un'unica terna di ripartizione del dataset (70/15/15). Si tratta della terna intermedia rispetto a 60/20/20 e 80/10/10 [59].

Macroscopicamente, l'architettura della rete, schematizzata in **Figura 2.18.**, è caratterizzata dalla successione di cinque layers:

- Sequence Input Layer, che manda dati sequenziali in input alla rete;
- BiLSTM Layer, che apprende dipendenze bidirezionali a lungo termine tra time step di sequenze;
- Fully Connected Layer, che moltiplica l'input per una matrice di pesi e aggiunge un vettore di bias;
- Softmax Layer, che applica la funzione softmax al suo input;
- Classification Layer, che calcola la cross entropy loss per problemi di classificazione con classi mutuamente esclusive.

In particolare [26], la funzione softmax associa ad un'osservazione il valore di probabilità di appartenenza a ciascuna delle due classi. Sostanzialmente, per la singola osservazione, fornisce in output due valori compresi tra 0 e 1, la cui somma è pari a 1. Nell'attribuzione della classe, il Classification Layer applica una soglia pari a 0.5. La classe

assegnata alla singola osservazione, come output finale della rete neurale, è, cioè, quella che ha ricevuto dal Softmax Layer una probabilità maggiore o uguale al 50%.

In **Appendice A.2**, sono riportati dettagli relativi all'architettura interna del BiLSTM Layer.

Tutti gli iperparametri, eccetto tre, settabili in MATLAB® sono mantenuti fissi, con valori scelti oppure con i valori di default. I tre iperparametri che sono stati variati al fine di trovare la rete migliore sono i seguenti: numero di Hidden Units, Initial Learning Rate (α) e Learning Rate Drop Factor (δ). Sostanzialmente, il tuning degli iperparametri viene effettuato facendo variare il numero di hidden units del BiLSTM Layer e due parametri del learning rate (ovvero il suo valore iniziale ed il suo fattore di decremento). La **Tabella 2.2** riporta il valore o l'intervallo di valori scelti per gli iperparametri che non sono lasciati default. In **Appendice A.3**, sono indicate le effettive opzioni impostate in fase di allenamento ed i corrispettivi significati.

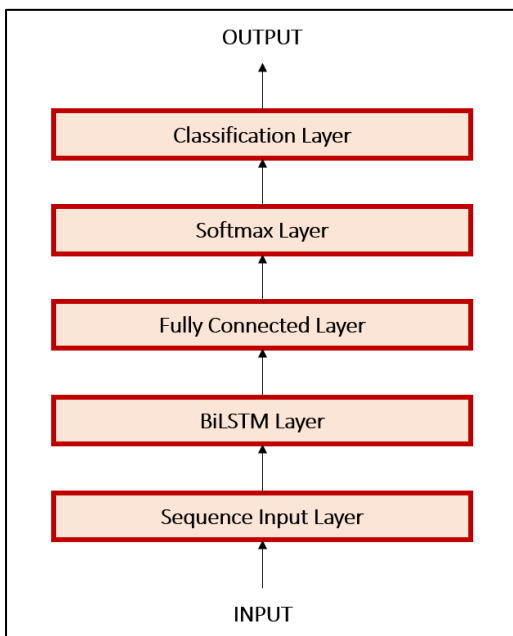


Figura 2.19. Architettura dei modelli BiLSTM implementati.

Dall'input verso l'output si susseguono cinque layers: Sequence Input Layer, BiLSTM Layer, Fully Connected Layer, Softmax Layer, Classification Layer.

Iperparametro	Valore/i
Numero di Features	12
Numero di Hidden Units	100, 150, 200, 250, 300
Numero di classi	2
Numero massimo di epoche	60
Dimensione del mini-batch	20% del training set
Initial Learning Rate	0.005, 0.01, 0.05, 0.1
Learning Rate Drop Factor	0.05, 0.1, 0.15, 0.2
Learning Rate Drop Period	15
Validation Patience	5
Gradient Threshold	1

Tabella 2.2. Valori scelti per gli iperparametri.

Complessivamente, combinando tutti i valori degli iperparametri, si sono ottenute 80 combinazioni differenti. Di conseguenza, vengono allenati, validati e testati 80 modelli BiLSTM per singolo approccio (in totale, 160 modelli). Successivamente, per singolo approccio, viene selezionato il modello migliore. La selezione di tale modello è basata sui seguenti due criteri in cascata:

- 1° criterio: selezione dei modelli tali per cui lo scostamento tra training accuracy e validation accuracy sia non superiore al 5%;
- 2° criterio: selezione del modello con testing accuracy massima tra quelli selezionati dal 1° criterio.

Le 160 reti neurali sono state implementate su una workstation con le seguenti specifiche: 3.20 GHz i7-8700 CPU; 32 GB di RAM; sistema operativo Windows 10 - versione a 64 bit.

Lo studio dell'applicabilità delle reti neurali BiLSTM al problema in esame culmina nel confronto tra i due modelli (STL, STS) selezionati. Tale confronto si basa, innanzitutto, sui corrispettivi valori di training accuracy, validation accuracy, testing accuracy. Di fatto, però, la scelta del modello, e quindi anche dell'approccio, migliore è dipendente, oltre che dai risultati numerici in termini di accuracy e di tempi computazionali richiesti (sulla stessa risorsa hardware utilizzata per l'implementazione delle 160 reti neurali), anche da una valutazione critica (non quantitativa) dei vantaggi e degli svantaggi di ciascun approccio. Infine, l'approccio scartato viene abbandonato e viene avviata la fase di ottimizzazione dell'altro modello (STL), scelto come migliore.

2.5 Ottimizzazione del modello BiLSTM scelto

Il modello BiLSTM scelto (STL) viene ottimizzato, innanzitutto, sul dataset ridotto dei 26 trials, valutando alcune variazioni della sua architettura. In seguito, sul dataset complessivo dei 51 trials, implementando la 10-fold cross-validation, viene esaminata la robustezza delle prestazioni del modello e, successivamente, la loro dipendenza dalla numerosità di canali sEMG considerati (ovvero di features in ingresso).

2.5.1 Variazioni dell'architettura

L'architettura del modello BiLSTM selezionato è sottoposta a variazioni nella successione dei layers. Complessivamente, sono effettuate 13 prove di variazione. Globalmente, vengono inseriti tre ulteriori layers:

- Dropout Layer, che randomicamente setta a zero, con un certo valore di probabilità, i dati che riceve in input;
- ReLU Layer, che forza a zero ogni elemento in input che sia minore di zero;
- tanh Layer, che applica la funzione tangente iperbolica (tanh) ai dati in input.

In **Tabella 2.3.**, sono elencate le 13 architetture con le quali il modello, sul dataset ridotto dei 26 trials, viene allenato, validato e testato.

Prova (#)	Architettura
1	SIL → BiLSTM Layer (200) → BiLSTM Layer (100) → FCL → SL → CL

2	SIL → BiLSTM Layer (200) → Dropout Layer (0.5) → FCL → SL → CL
3	SIL → BiLSTM Layer (200) → Dropout Layer (0.5) → BiLSTM Layer (200) → FCL → SL → CL
4	SIL → BiLSTM Layer (200) → Dropout Layer (0.5) → BiLSTM Layer (200) → Dropout Layer (0.5) → FCL → SL → CL
5	SIL → BiLSTM Layer (200) → BiLSTM Layer (200) → Dropout Layer (0.5) → FCL → SL → CL
6	SIL → BiLSTM Layer (200) → Dropout Layer (0.2) → FCL → SL → CL
7	SIL → BiLSTM Layer (200) → Dropout Layer (0.8) → FCL → SL → CL
8	SIL → BiLSTM Layer (200) → ReLU Layer → FCL → SL → CL
9	SIL → BiLSTM Layer (200) → tanh Layer → FCL → SL → CL
10	SIL → BiLSTM Layer (200) → FCL → ReLU Layer → SL → CL
11	SIL → BiLSTM Layer (200) → FCL → tanh Layer → SL → CL
12	SIL → BiLSTM Layer (200) → Dropout Layer (0.5) → ReLU Layer → FCL → SL → CL
13	SIL → BiLSTM Layer (200) → ReLU Layer → Dropout Layer (0.5) → FCL → SL → CL

Tabella 2.3. Architetture delle 13 prove di variazione (ciascuna prova è caratterizzata da un'architettura differente). Sono adottati i seguenti acronimi: SIL (Sequence Input Layer), FCL (Fully Connected Layer), SL (Softmax Layer), CL (Classification Layer). Il numero tra parentesi a destra di BiLSTM Layer ne indica il numero di hidden units. Il numero tra parentesi a destra di Dropout Layer ne indica il valore di probabilità.

Viene di seguito commentata la scelta delle 13 architetture considerate. Nella prova 1 è introdotto un secondo BiLSTM Layer, in cascata al primo e con numero di hidden units dimezzato (pari a 100). Nella prova 2 è inserito un Dropout Layer con probabilità pari a 0.5 (valore default in MATLAB®). Nella prova 3 è replicata l'architettura della prova 2 con un secondo BiLSTM Layer, identico al primo, a valle del Dropout Layer. Nella prova 4 è replicata l'architettura della prova 3, con un ulteriore Dropout Layer, identico al primo, a valle del secondo BiLSTM Layer. Nella prova 5 è mantenuta l'architettura della prova 3, scambiando le posizioni tra il Dropout Layer e il secondo BiLSTM Layer. Nelle prove 6 e 7 è mantenuta l'architettura della prova 2, cambiando il valore di probabilità del Dropout Layer: rispettivamente, in 0.2 e 0.8. Nelle prove 8 e 9 sono introdotti, rispettivamente, un ReLU Layer e un tanh Layer in cascata al BiLSTM Layer. Le prove 10 e 11 replicano, rispettivamente, la 8 e la 9, spostando il layer introdotto in avanti, dopo il Fully Connected Layer. Le prove 12 e 13 replicano la 8 e inseriscono un Dropout Layer (con valore di probabilità 0.5), rispettivamente, prima e dopo il ReLU Layer.

Per ciascuna prova, sono considerati lo scostamento tra i valori di training accuracy e validation accuracy e la testing accuracy. Le prove per le quali il valore dello scostamento è maggiore del 5% (cfr. 2.4.2) vengono trascurate. Delle prove rimanenti, viene confrontato il corrispettivo valore di testing accuracy con quello del modello scelto. Infine, se in presenza di prove con testing accuracy superiore a quella del modello scelto, viene valutato criticamente se mantenerne intatta l'architettura iniziale o modificarla di conseguenza.

2.5.2 Valutazione della robustezza delle prestazioni

Al fine di valutare la robustezza delle prestazioni del modello BiLSTM e la sua generalizzabilità, ottimizzato in architettura, esso viene allenato e testato in termini di 10-fold cross-validation, sul dataset complessivo dei 51 trials. Come rappresentato in **Figura 2.20.**, il dataset è ripartito randomicamente in 10 folds, aventi la stessa dimensione e nessuna osservazione in comune. Iterando 10 volte, progressivamente, 9 folds costituiscono il training set e 1 fold il testing set. Ciascuna fold è in sé bilanciata rispetto sia alle due sottopopolazioni che alle due classi. La sua composizione interna è tetrapartita come segue:

- 25% di osservazioni della sottopopolazione sana, classe 0;
- 25% di osservazioni della sottopopolazione sana, classe 1;
- 25% di osservazioni della sottopopolazione patologica, classe 0;
- 25% di osservazioni della sottopopolazione patologica, classe 1.

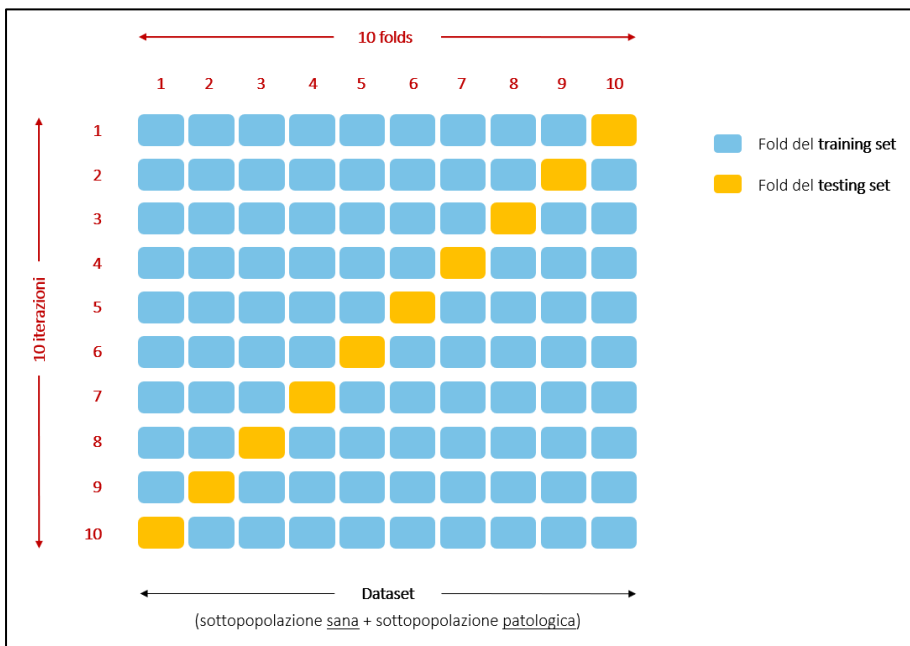


Figura 2.20. Implementazione della 10-fold cross-validation. Il dataset complessivo dei 51 trials (eseguiti dai soggetti sani e dai soggetti patologici) è splittato in 10 folds. Per ciascuna delle 10 iterazioni, progressivamente, 9 folds costituiscono il training set e 1 fold il testing set.

È così possibile valutare l'impatto della randomizzazione sulla capacità di generalizzazione del modello BiLSTM. Al fine di esaminare la robustezza delle prestazioni del modello, sono calcolati, rispetto al testing set, i risultati medi (\pm deviazione standard) delle seguenti metriche di classificazione: confusion matrix, accuracy, precision, recall, F_1 score ed indice di Jaccard (cfr. 2.7).

2.5.3 Valutazione della dipendenza delle prestazioni dal numero di canali sEMG

Il modello BiLSTM, ottimizzato in architettura, viene poi allenato e testato al variare della numerosità di canali sEMG considerati, ovvero di features in ingresso. L'obiettivo di tali implementazioni è la valutazione della dipendenza delle prestazioni del modello da tale numerosità. In **Tabella 2.4.**, sono indicati i set fisiologici di muscoli considerati.

Set (#)	Canali sEMG
1	LD _R , LD _L
2	TA, LGS, RF, LH
3	TA, LGS, RF, LH, VM
4	TA, LGS, RF, LH, LD _R , LD _L
5	TA, LGS, RF, LH, LD _R , LD _L , VM

Tabella 2.4. Elenco dei canali sEMG inclusi in ciascuno dei 5 set.

Viene di seguito commentata la scelta dei 5 set considerati. Il set 1 include gli unici due canali sEMG (del tronco) acquisiti bilateralmente. Il set 2 è costituito dai canali sEMG sempre acquisiti in analisi del cammino. Tale set permette infatti di analizzare almeno

una coppia di muscoli agonista-antagonista per ciascuna articolazione dell'arto inferiore (anca: RF/LH; ginocchio: LH-LGS/RF; caviglia: TA/LGS). Poiché LGS e RF sono muscoli bi-articolari, tale configurazione rende possibile ottenere informazioni biomeccaniche rilevanti mediante un set minimo di sonde sEMG [56]. Il set 3 replica il set 2 aggiungendo VM. Tale canale sEMG è stato scelto in quanto, ispezionando visivamente gli involucri dei segnali sEMG di alcuni trials, esso presentava tendenzialmente andamenti affini ai muscoli del set 2. Il set 4 replica il set 2 aggiungendo i due canali sEMG bilaterali. Il set 5 ripete il set 4 includendo VM.

Per ciascuna delle 5 prove, è implementata la 10-fold cross-validation, esattamente come descritto in **2.3.3.2**, (un'implementazione di 10-fold cross-validation per ciascun set). Le 10 folds ottenute in **2.3.3.2** (12 canali sEMG) sono mantenute intatte, rimuovendo però, per la singola prova, rispetto a tutte le osservazioni, i canali sEMG (righe delle sottomatrici) non considerati.

Al fine di esaminare il comportamento del modello a numerosità differenti di muscoli, per ciascuna prova sono calcolati, rispetto al testing set, i risultati medi (\pm deviazione standard) delle seguenti metriche di classificazione: confusion matrix, accuracy, precision, recall, F_1 score ed indice di Jaccard (cfr. **2.7**).

2.6 Definizione del funzionamento dell'algoritmo finale

L'algoritmo finale di classificazione binaria deve permettere di identificare la classe di appartenenza del singolo campione di un'intera acquisizione di segnali sEMG. La sua applicazione è esclusivamente off-line, in quanto la rete neurale LSTM bidirezionale (BiLSTM) classifica ciascun campione di una serie temporale (nella fattispecie, dei segnali sEMG) sfruttando tutti i campioni che lo precedono e tutti i campioni che lo seguono.

Il funzionamento dell'algoritmo si articola in tre passi fondamentali:

1. Segmentazione della matrice contenente i segnali sEMG in finestre da 1000 campioni ($f_s/2$) con "overlap" del 75%
2. Classificazione
3. Post-Processing

Il funzionamento dell'algoritmo si basa sulla finestratura dei segnali sEMG mediante una "sliding window" di 1000 campioni (500 ms), progressivamente sovrapposta per il 75% (**Figura 2.21.**). Ciascuna finestra viene classificata singolarmente (in approccio STL). Successivamente, la label assegnata alla singola finestra (label 0, se la finestra è classificata come appartenente alla classe 0; label 1, se la finestra è classificata come appartenente alla classe 1) viene replicata per tutti i 1000 campioni della finestra stessa, costruendo un vettore di 1000 labels identiche. Infine, tutti i vettori di 1000 labels sono concatenati in un vettore unico, così che a ciascun campione del segnale sia associata una label unica, che ne identifica la classe di appartenenza. La sovrapposizione del 75% tra due finestre consecutive permette che ogni campione del segnale venga classificato quattro volte. Per il singolo campione, la label presente nel vettore finale coincide con la moda delle quattro labels ottenute.

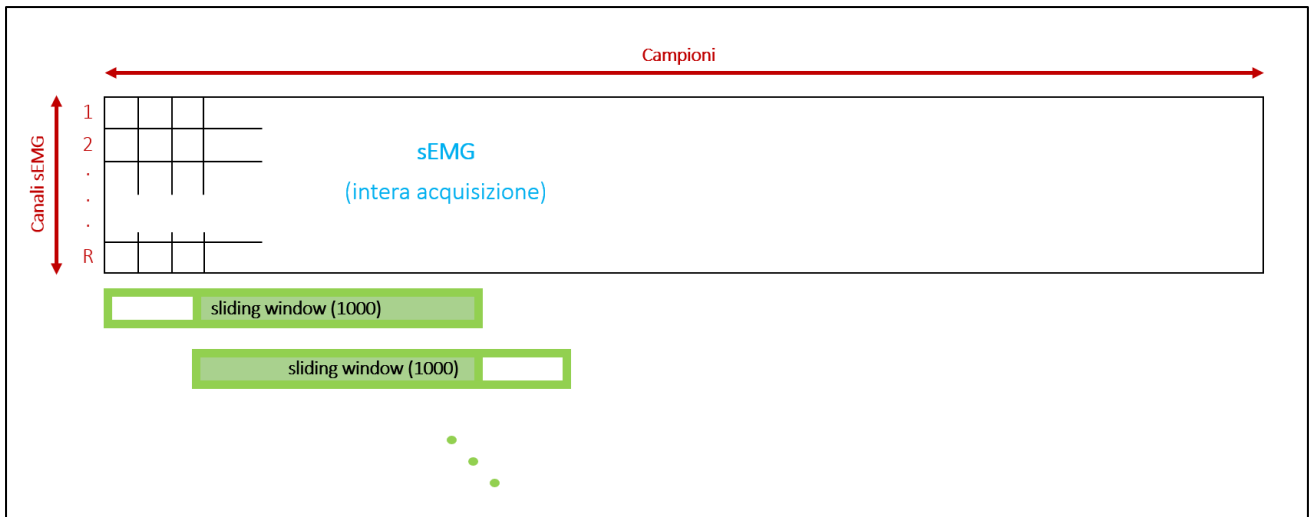


Figura 2.21. Rappresentazione del primo passo dell’algoritmo finale. La matrice contenente l’intera acquisizione di segnali sEMG viene finestrata mediante una “sliding window” di 1000 campioni, con sovrapposizione del 75%.

In **Figura 2.22.**, è riportato il diagramma di flusso relativo ai primi due passi dell’algoritmo, ovvero dalla finestratura dei segnali sEMG alla costruzione del vettore unico di labels. In **Figura 2.23.**, è illustrata la concatenazione delle labels in tale vettore unico, ottenuto a valle del secondo passo dell’algoritmo.

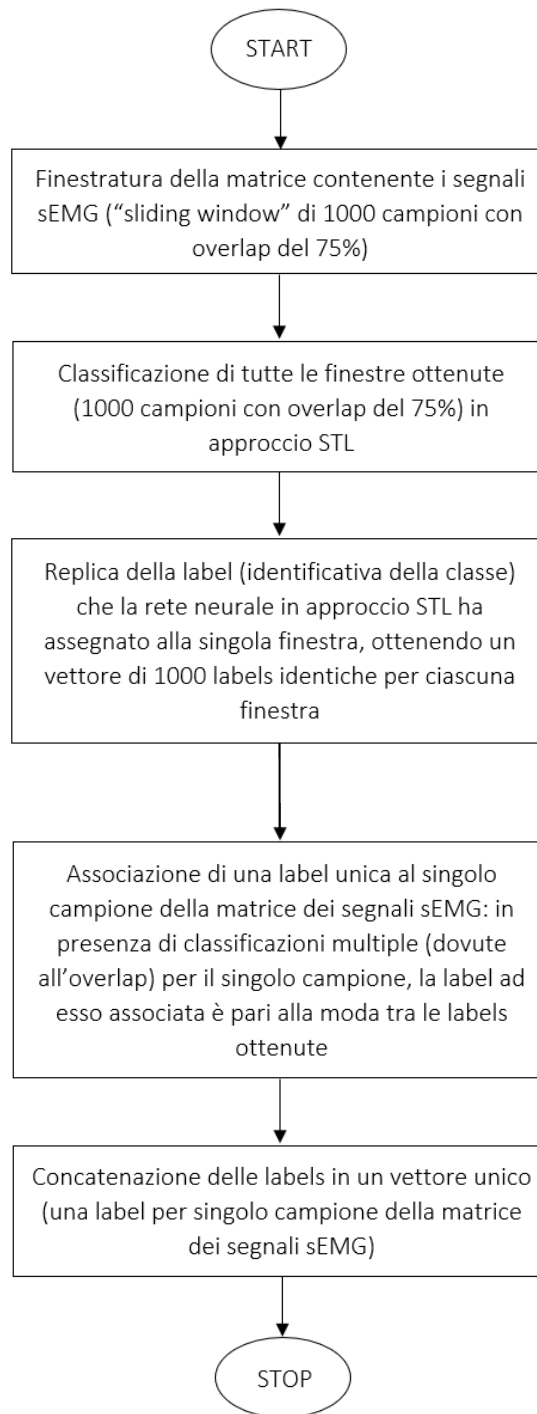


Figura 2.22. Diagramma di flusso dell'algorithmo fino al secondo passo.

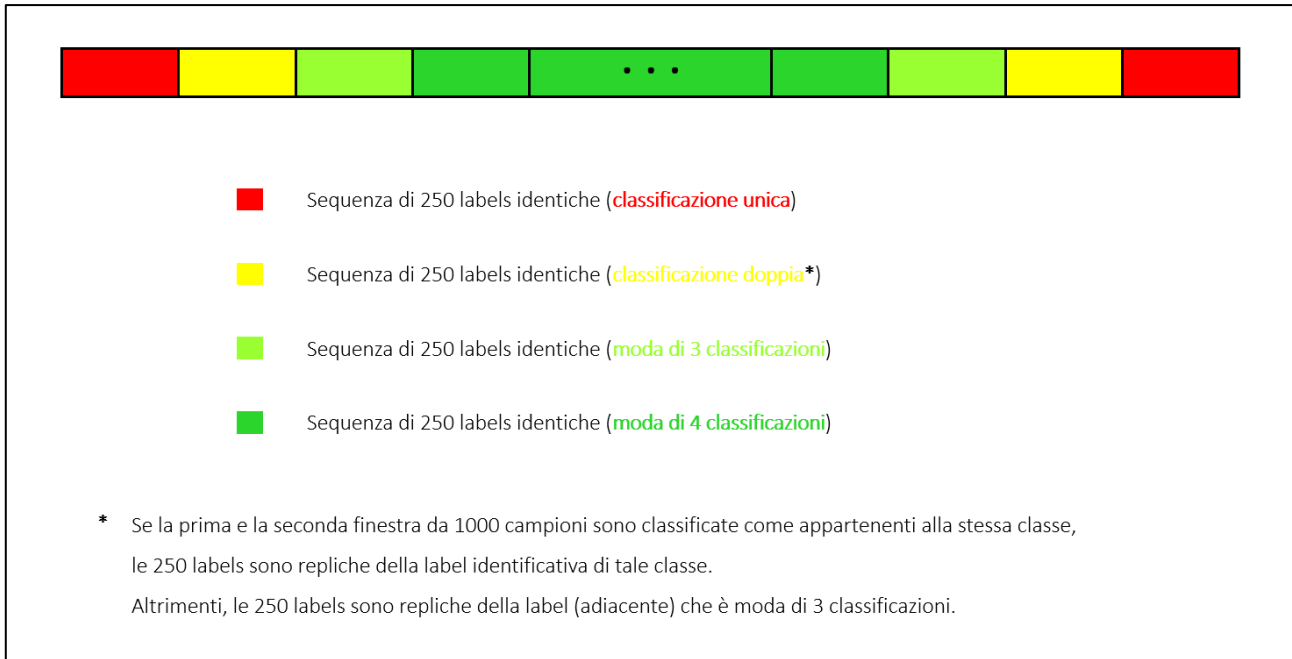


Figura 2.23. Costruzione del vettore di labels. Ciascuna label (0 oppure 1) identifica la classe di appartenenza (rispettivamente, non-rettilineo oppure rettilineo) della singola colonna della matrice dei segnali sEMG. Il vettore è costruito concatenando sequenze di 250 labels. Le 250 labels di una sequenza, identiche tra di esse, sono repliche di una label ottenuta da una classificazione unica o da una classificazione doppia o come moda di tre oppure di quattro classificazioni, a seconda della posizione della sequenza lungo il vettore. I tre punti al centro del vettore indicano la presenza di N sequenze di 250 labels.

2.6.1 Ottimizzazione dell'algorithmo finale

Il quarto passo dell'algorithmo finale consiste nella sua ottimizzazione, in termini di post-processing implementato sul vettore di labels (costruito nel terzo passo dell'algorithmo). La modalità di post-processing viene scelta confrontando visivamente le rappresentazioni grafiche del vettore di labels e del target per i 51 trials del dataset complessivo. Osservando una presenza significativa di falsi negativi (ovvero di campioni della classe 1 etichettati, erroneamente, come classe 0; cfr. 2.7), il post-processing ha come obiettivo la loro riduzione. Sostanzialmente, vengono ridotti i tratti classificati come non-rettilinei che sono considerati troppo "brevi". A tal fine, nel vettore di labels,

sequenze “brevi” di 0 sono forzate a 1. Per definire “breve” una porzione di segnale classificata come non-rettilineo, vengono valutate due lunghezze candidate-soglia: 2000 campioni e 4000 campioni. Entrambi i valori sono multipli della frequenza di campionamento del segnale (pari a 2 kHz) e sono stati scelti considerando la durata media (\pm deviazione standard) dei tratti non-rettilinei presenti nel dataset complessivo dei 51 trials. Tale durata è pari a $2.7 \text{ s} \pm 0.7 \text{ s}$. Al fine di adottare una soglia unica, l’algoritmo viene testato due volte (una volta con soglia 2000 campioni e una volta con soglia 4000 campioni) sul dataset complessivo dei 51 trials. L’individuazione della soglia migliore si basa sul confronto tra i risultati medi (\pm deviazione standard) delle due validazioni, in termini delle seguenti metriche di classificazione: confusion matrix, accuracy, precision, recall, F_1 score ed indice di Jaccard (cfr. 2.7).

In seguito, l’algoritmo, implementato con la soglia individuata, viene validato su un ulteriore dataset di 10 trials (5 di soggetti sani adulti e 5 di pazienti parkinsoniani), non compresi tra i 51. Confrontando i risultati medi (\pm deviazione standard), in termini di confusion matrix, accuracy, precision, recall, F_1 score ed indice di Jaccard (cfr. 2.7), ottenuti su tale dataset di validazione con i corrispettivi ottenuti sul dataset complessivo dei 51 trials (a parità di soglia), viene infine valutata la robustezza delle prestazioni dell’algoritmo sviluppato.

In termini di tempi computazionali, l’algoritmo finale viene testato sui 10 trials del dataset di validazione, mantenendo la stessa risorsa hardware utilizzata per l’implementazione delle 160 reti neurali (workstation con 3.20 GHz i7-8700 CPU; 32 GB di RAM; sistema operativo Windows 10 - versione a 64 bit).

2.7 Valutazione delle prestazioni

Per valutare le prestazioni delle reti neurali BiLSTM implementate, complessivamente, sono state adottate le seguenti metriche di classificazione [60]:

- Confusion Matrix, che mostra i valori dei veri positivi e negativi e dei falsi positivi e negativi, come in **Figura 2.24**. In termini di classificazioni corrette, i veri positivi (VP) sono le osservazioni della classe 1 (classe dei tratti rettilinei) che sono classificate come appartenenti a tale classe; i veri negativi (VN) sono le osservazioni della classe 0 (classe dei tratti non-rettilinei) che sono classificate come appartenenti a tale classe. In termini di classificazioni errate, i falsi positivi (FP) sono le osservazioni della classe 0 che sono classificate come appartenenti alla classe 1; i falsi negativi (FN) sono le osservazioni della classe 1 che sono classificate come appartenenti alla classe 0.

Classe Target	0	VN	FP
	1	FN	VP
		0	1
		Classe Predetta	

Figura 2.24. Confusion Matrix adottata nella tesi. Rispettivamente, gli acronimi riportati indicano: VN, i Veri Negativi; FN, i Falsi Negativi; FP, i Falsi Positivi; VP, i Veri Positivi.

- Accuracy, definita come in (1), che è il rapporto tra il numero di osservazioni classificate correttamente (cioè la somma di veri positivi e veri negativi) ed il

numero totale di osservazioni classificate (cioè la somma di veri positivi, falsi positivi, veri negativi e falsi negativi).

$$\text{accuracy} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{FP} + \text{VN} + \text{FN}} \quad (1)$$

Globalmente, sono stati considerati i seguenti tre valori di accuracy:

- training accuracy, in termini di accuracy di classificazione del training set nell'ultima iterazione di allenamento;
 - validation accuracy, in termini di accuracy di classificazione del validation set nell'ultima iterazione di allenamento;
 - testing accuracy, in termini di classificazione del testing set a valle dell'allenamento concluso.
-
- Precision, definita come in (2), che è il rapporto tra il numero di osservazioni della classe 1 classificate correttamente (cioè i veri positivi) ed il numero totale di osservazioni classificate come appartenenti alla classe 1 (cioè la somma di veri positivi e falsi positivi).

$$\text{precision} = \frac{\text{VP}}{\text{VP} + \text{FP}} \quad (2)$$

- Recall, definita come in (3), che è il rapporto tra il numero di osservazioni della classe 1 classificate correttamente (cioè i veri positivi) ed il numero totale di osservazioni appartenenti alla classe 1 (cioè la somma di veri positivi e falsi negativi).

$$\text{recall} = \frac{\text{VP}}{\text{VP} + \text{FN}} \quad (3)$$

- F₁ score, definito come in (4), che è la media armonica tra i valori di precision e recall.

$$F_1 \text{ score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (4)$$

- Indice di Jaccard, definito come in (5), che misura la similarità tra due insiemi, A e B, in termini di rapporto tra la dimensione della loro intersezione e la dimensione della loro unione.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (5)$$

L'indice di Jaccard è stato calcolato mediante la funzione MATLAB® "jaccard".

Tale funzione è stata implementata con i seguenti due argomenti in input:

- 1° argomento (ovvero l'insieme A): l'output della rete neurale;
- 2° argomento (ovvero l'insieme B): il target.

Capitolo 3

Risultati

Sono di seguito riportati i risultati ottenuti relativamente allo studio di applicabilità degli approcci STL e STS al problema di segmentazione dei tratti rettilinei e non-rettilinei, all'ottimizzazione del modello BiLSTM scelto ed all'implementazione dell'algoritmo finale, sviluppato a partire dal modello ottimizzato.

3.1 Esplorazione dell'applicabilità delle reti neurali BiLSTM

3.1.1 Definizione del modello BiLSTM

In **Appendice B.1**, sono riportati i risultati, in termini di training accuracy, validation accuracy e testing accuracy, relativi a ciascuno dei 160 modelli BiLSTM testati (80 modelli in approccio STL e 80 modelli in approccio STS). Applicando i due criteri indicati in **2.3.2.2**, viene selezionato un modello unico per ciascuno dei due approcci. Per i due modelli selezionati, in **Figura 3.1.**, sono rappresentati gli andamenti di accuracy e di loss durante il rispettivo allenamento. Globalmente, la sovrapposizione tra la curva di accuracy relativa al validation set e quella relativa al training set è più evidente per il modello STS. Ovvero, il modello STS riesce a generalizzare meglio. Per il modello STL, a partire dall'iterazione 200, si osserva maggiormente lo scostamento tra le due curve. Ovvero, il modello va in overfitting e si riduce la sua capacità di generalizzazione. Per

entrambi i modelli, la curva di loss relativa al training set e quella relativa al validation set si assestano intorno a 0.5.

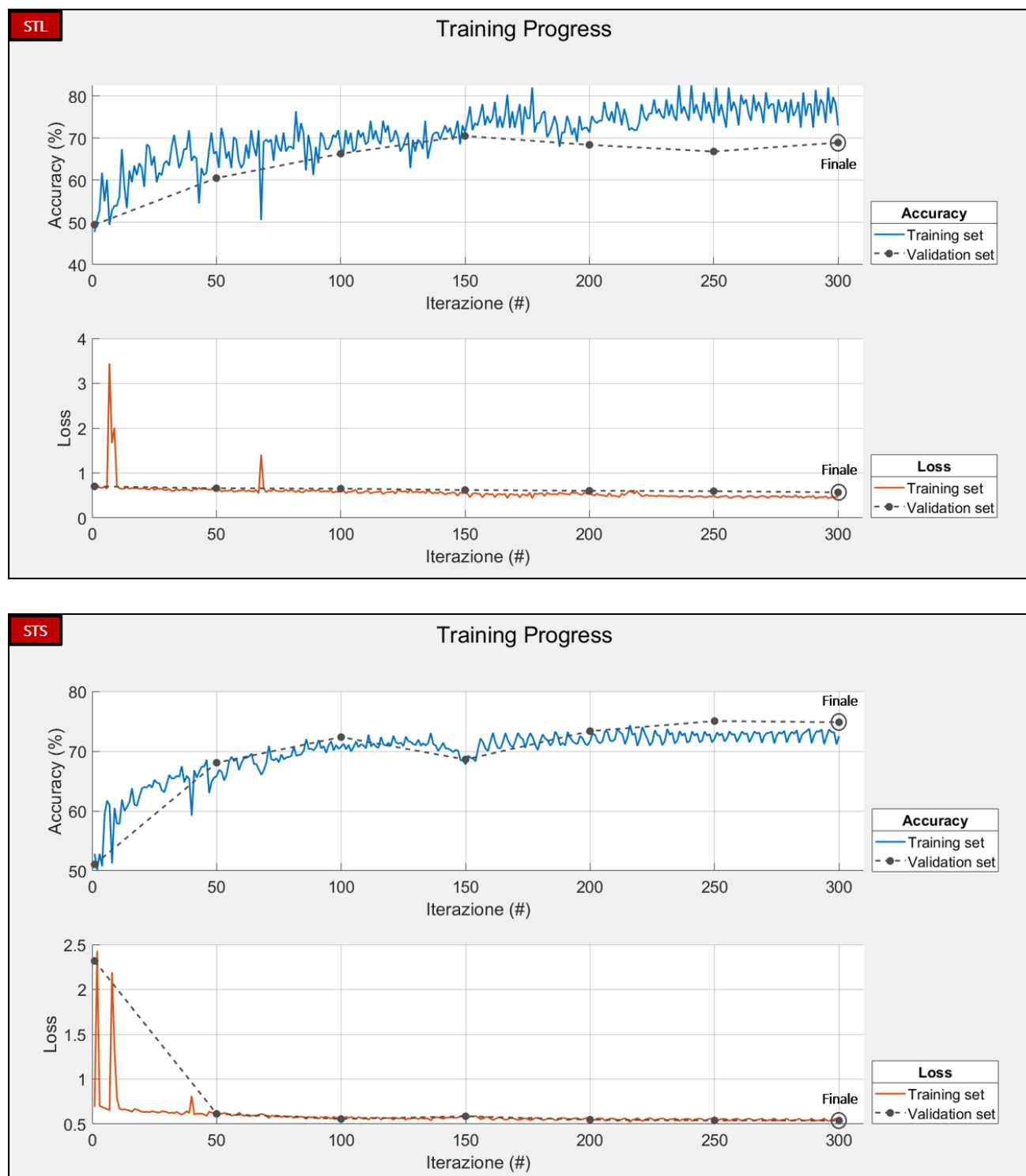


Figura 3.1. Evoluzione di accuracy e di loss durante l'allenamento dei due modelli selezionati. In alto, STL; in basso, STS.

Applicando i due criteri di selezione del modello specificati in 2.3.2.2, viene selezionato un modello unico per singolo approccio. Per i due modelli selezionati, in **Tabella 3.1.**, sono riportati i valori di accuracy e di scostamento tra training accuracy e validation accuracy, in percentuale, ed i valori degli iperparametri con variazione.

	Modello STL	Modello STS
Training Accuracy (%)	73.0	72.6
Validation Accuracy (%)	69.0	74.8
Scostamento (%)	4.0	2.2
Testing Accuracy (%)	71.6	73.1
Numero di Hidden Units	200	200
α	0.005	0.01
δ	0.2	0.2

Tabella 3.1. Valori di accuracy e di scostamento tra training accuracy e validation accuracy e valori degli iperparametri con variazione dei due modelli selezionati (STL, STS).

In riferimento a quanto riportato in **Tabella 3.1.**, considerando, da una parte, i due valori di testing accuracy e, dall'altra, i corrispettivi valori degli iperparametri, i due modelli selezionati (e dunque i due approcci) sono confrontabili. Tuttavia, rispetto ai tempi computazionali medi (\pm deviazione standard) richiesti per la classificazione del rispettivo training set e, in cascata, del rispettivo testing set, i due approcci, statisticamente, presentano differenza significativa (p -value = 0.000003; livello di significatività $\alpha = 0.05$): 19.4 s \pm 10.3 s (approccio STL) e 45.8 s \pm 47.7 s (approccio STS). Inoltre, per come sfruttato nella presente tesi, l'approccio STS implica anche maggiore complessità in termini di concettualizzazione del problema e di costruzione del dataset. Di conseguenza, l'approccio STS viene abbandonato e viene scelto esclusivamente il modello in approccio STL.

3.2 Ottimizzazione del modello BiLSTM scelto

3.2.1 Variazioni dell'architettura

Per ciascuna delle 13 prove di variazione dell'architettura del modello selezionato in approccio STL, la **Tabella 3.3.** riporta, in percentuale, i tre valori di accuracy e lo scostamento tra training accuracy e validation accuracy. Il numero che identifica ciascuna prova segue l'elenco in **2.3.3.1.**

Prova (#)	Training Accuracy (%)	Validation Accuracy (%)	Scostamento (%)	Testing Accuracy (%)
1	78.9	64.9	14.0	66.7
2	77.1	72.8	4.3	71.4
3	78.0	64.9	13.1	/
4	77.1	67.0	10.1	69.2
5	69.9	61.2	8.7	63.0
6	75.3	65.6	9.7	71.4
7	75.3	65.2	10.1	63.8
8	85.5	67.8	17.7	68.1
9	77.1	68.5	8.6	67.4
10	50.0	50.0	0.0	50.0
11	71.1	64.1	7.0	62.7
12	72.9	62.7	10.2	63.0
13	77.1	67.4	9.7	73.2

Tabella 3.3. Valori di accuracy e di scostamento tra training accuracy e validation accuracy relativi alle prove di variazione dell'architettura. Il valore dello scostamento è in rosso se supera la soglia del 5%, altrimenti è in verde.

Il valore di testing accuracy della prova 3 non è riportato in **Tabella 3.3.** perché, inavvertitamente, la prova è stata interrotta e, considerato il valore dello scostamento non soddisfacente, non è stata ripetuta.

In **Appendice B.2,** sono rappresentate le 13 architetture relative alle 13 prove di variazione, con indicazione di quali abbiano o meno soddisfatto il criterio di soglia sullo scostamento.

3.2.2 Valutazione della robustezza delle prestazioni

Sono di seguito riportati, in percentuale, i risultati medi (\pm deviazione standard) della 10-fold cross-validation, in termini di confusion matrix di classificazione del testing set (**Figura 3.2.**) e di metriche di classificazione relative al testing set (**Tabella 3.4.**).

Classe Target	0	33.5% \pm 3.6% (VN)	16.5% \pm 3.6% (FP)
	1	11.8% \pm 5% (FN)	38.2% \pm 5% (VP)
		0	1
		Classe Predetta	

Figura 3.2. Confusion Matrix di classificazione del testing set: media e deviazione standard (10-fold cross-validation con 12 canali sEMG).

Per ciascuna metrica di classificazione relativa al testing set, la **Tabella 3.4.** riporta anche i valori di minimo e massimo rispetto alle 10 iterazioni. I due valori estremi considerati, per ciascuna metrica di classificazione, individuano l'intervallo di variazione dei

corrispettivi risultati, al variare della fold utilizzata come testing set. In **Figura 3.3.**, sono mostrati, mediante grafico a barre, i valori percentuali di training accuracy e di testing accuracy di ogni iterazione ed è evidenziata l'iterazione per la quale lo scostamento tra training accuracy e testing accuracy non supera la soglia del 5% e la testing accuracy è la massima. Tale iterazione corrisponde alla rete neurale implementata nell'algoritmo finale di segmentazione automatica dei segnali sEMG.

	Accuracy (%)	Precision (%)	Recall (%)	F_1 score (%)	Indice di Jaccard (%)
	71.7 ± 7.2	69.8 ± 6.1	76.3 ± 10.0	72.8 ± 7.4	57.7 ± 9.1
Range	61.1 – 80.0	59.4 - 78.5	61.1 - 92.2	62.5 - 82.2	45.5 - 69.8

Tabella 3.4. Metriche di classificazione relative al testing set: media e deviazione standard (riga in alto) e valori minimo e massimo (riga in basso) rispetto alle 10 iterazioni (10-fold cross-validation con 12 canali sEMG).

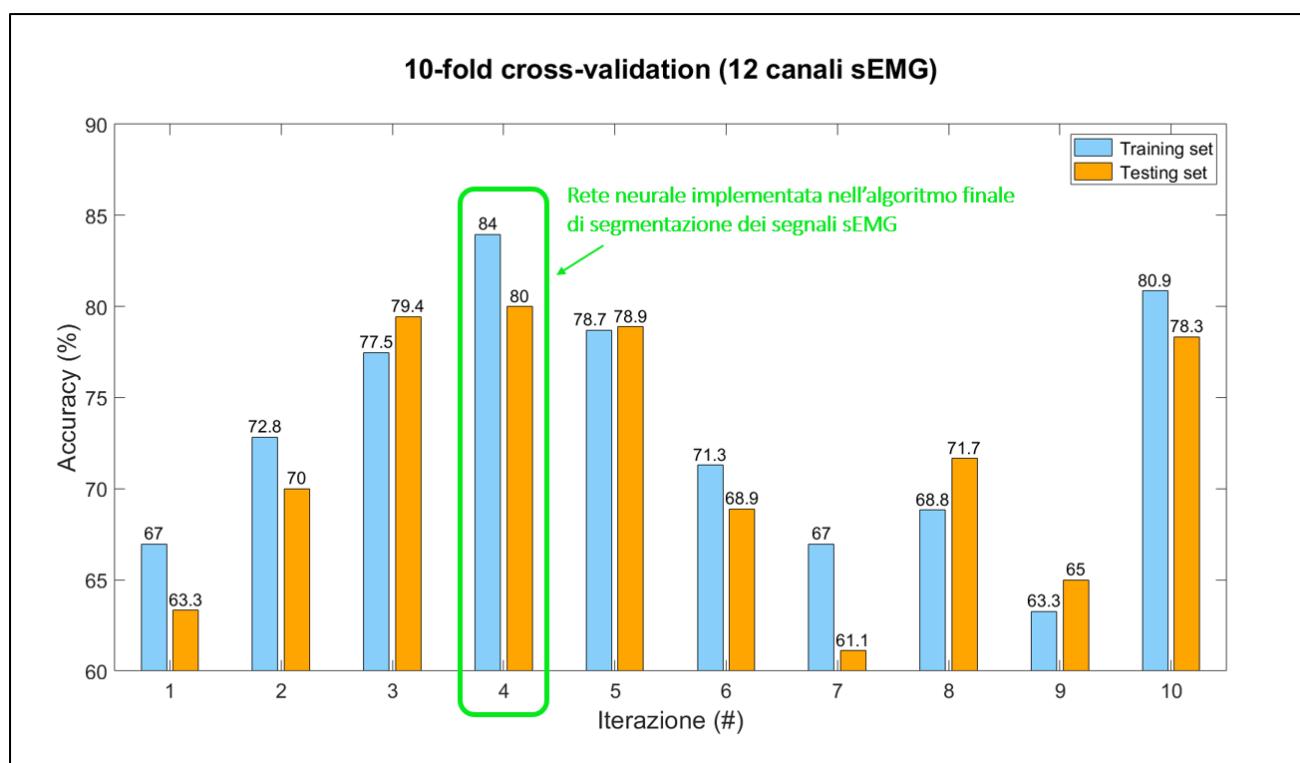
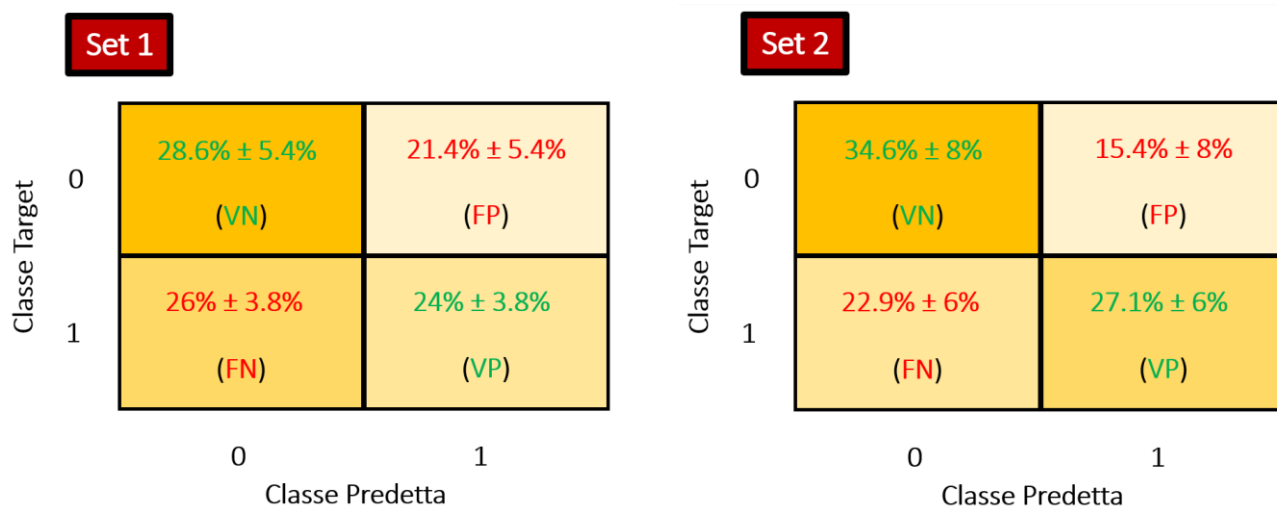


Figura 3.3. Valori di training accuracy e testing accuracy delle 10 iterazioni (10-fold cross-validation con 12 canali sEMG). La rete neurale dell'iterazione 4 è quella implementata nell'algoritmo finale di segmentazione dei segnali sEMG.

3.2.3 Valutazione della dipendenza delle prestazioni dal numero di canali sEMG

Per ciascun set di canali sEMG, sono di seguito riportati, in percentuale, i risultati medi (\pm deviazione standard) della corrispettiva 10-fold cross-validation, in termini di confusion matrix di classificazione del testing set (**Figura 3.4.**) e di metriche di classificazione relative al testing set (**Tabella 3.5.**). In **Figura 3.4.** e in **Tabella 3.5.**, il numero che identifica ciascun set segue l'elenco in **2.3.3.3.** Per ciascuna metrica di classificazione relativa al testing set, la **Tabella 3.6.** riporta i valori di minimo e massimo rispetto alle 10 iterazioni. Come precisato in **3.2.2.**, i due valori estremi considerati, per ciascuna metrica di classificazione, individuano l'intervallo di variazione dei corrispettivi risultati, al variare della fold utilizzata come testing set.



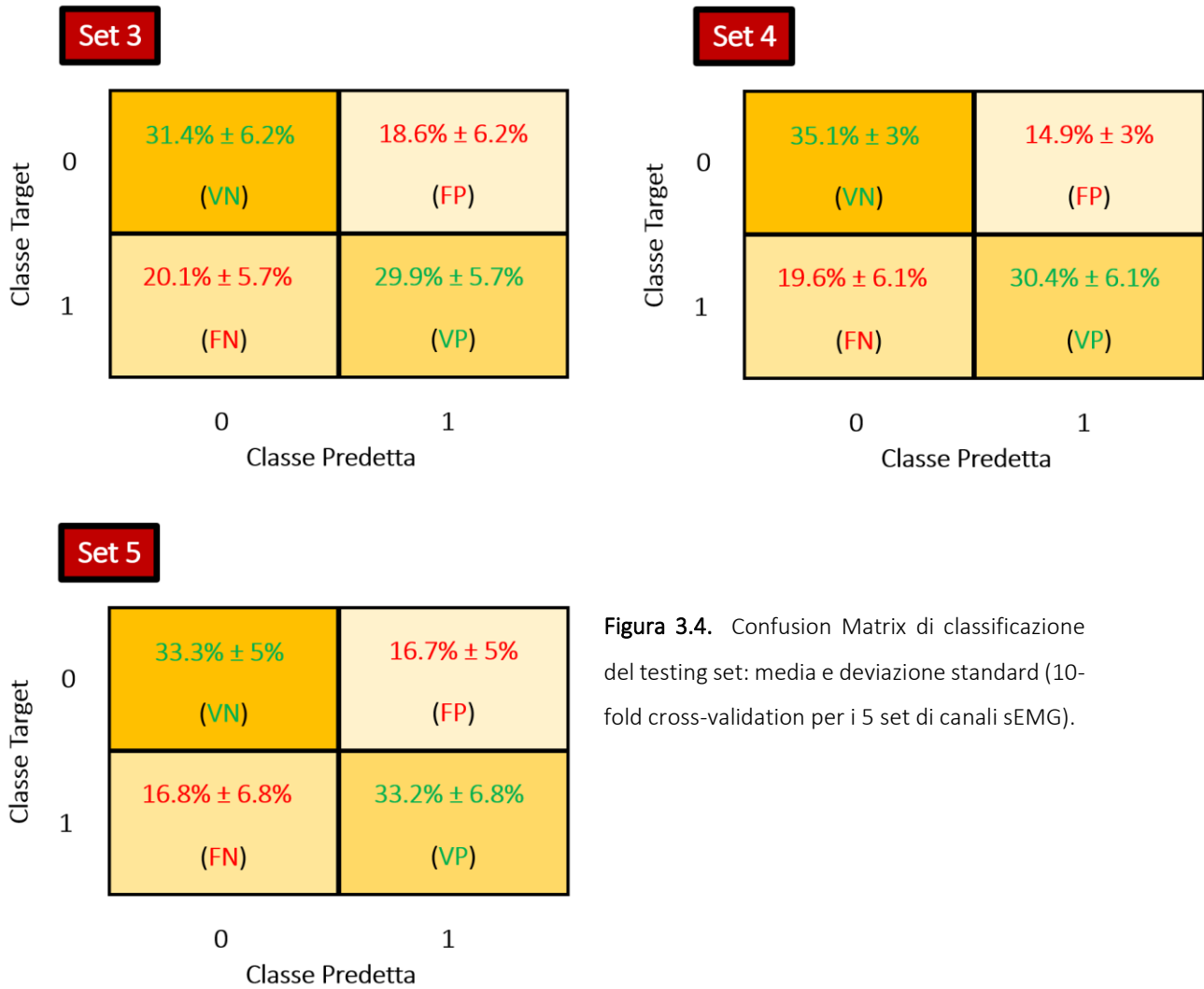


Figura 3.4. Confusion Matrix di classificazione del testing set: media e deviazione standard (10-fold cross-validation per i 5 set di canali sEMG).

Set (#)	Accuracy (%)	Precision (%)	Recall (%)	F_1 score (%)	Indice di Jaccard (%)
1	52.6 ± 4.3	53.3 ± 5.5	48.1 ± 7.7	50.2 ± 5.1	33.6 ± 4.4
2	61.6 ± 5.5	65.2 ± 8.2	54.1 ± 12.0	58.0 ± 6.8	41.2 ± 6.7
3	61.3 ± 2.6	62.5 ± 4.5	59.8 ± 11.4	60.2 ± 5.6	43.3 ± 5.5
4	65.5 ± 5.4	67.0 ± 4.4	60.8 ± 12.1	63.3 ± 7.9	46.7 ± 8.6
5	66.4 ± 8.3	66.5 ± 8.6	66.3 ± 13.7	66.0 ± 10.1	50.0 ± 10.8

Tabella 3.5. Metriche di classificazione relative al testing set: media e deviazione standard (10-fold cross-validation per i 5 set di canali sEMG).

Range	Accuracy (%)	Precision (%)	Recall (%)	F ₁ score (%)	Indice di Jaccard (%)	
Set (#)	1	46.1 - 60.6	46.2 - 64.2	35.6 - 56.7	41.0 - 54.8	25.8 - 37.8
	2	51.1 - 70.6	50.9 - 82.5	38.9 - 71.1	48.0 - 67.0	31.5 - 50.4
	3	57.8 - 66.7	56.9 - 69.6	35.6 - 72.2	47.1 - 65.5	30.8 - 48.7
	4	58.9 - 73.3	59.8 - 73.6	46.7 - 75.6	53.9 - 73.6	36.8 - 58.3
	5	52.8 - 75.6	53.7 - 80.6	40.0 - 83.3	45.9 - 77.3	29.8 - 63.0

Tabella 3.6. Metriche di classificazione relative al testing set: valori minimo e massimo rispetto alle 10 iterazioni (10-fold cross-validation per i 5 set di canali sEMG).

Riscontrando mediamente, sul testing set, il peggioramento delle prestazioni del modello al diminuire del numero di canali sEMG considerati, è necessario mantenere tutti i 12 canali sEMG nell'implementazione dell'algoritmo finale. Per tale ragione, le 5 implementazioni di 10-fold cross-validation per i 5 set di canali sEMG non sono ulteriormente esplorate in termini di valori di training accuracy e di testing accuracy delle 10 iterazioni (cfr. **Figura 3.3.**). In particolare, rispetto ai 5 set di canali sEMG considerati, non subentra l'esigenza di individuare alcuna iterazione (e dunque rete neurale) migliore.

3.3 Post-Processing

Inizialmente, l'algoritmo di segmentazione dei segnali sEMG, sviluppato a partire dal modello BiLSTM ottimizzato, è stato implementato su ciascun trial del dataset complessivo dei 51 trials senza alcun post-processing, per valutarne le performance. Sono di seguito riportati, in percentuale, i risultati medi (\pm deviazione standard) in termini di confusion matrix (**Figura 3.5.**) e di metriche di classificazione (**Tabella 3.7.**).

Classe Target	0	19.8% \pm 4.4% (VN)	4.6% \pm 2.1% (FP)
	1	46.9% \pm 8.4% (FN)	28.7% \pm 8.5% (VP)
		0	1
		Classe Predetta	

Figura 3.5. Confusion Matrix di classificazione dei 51 trials del dataset complessivo: media e deviazione standard (algoritmo di segmentazione implementato senza post-processing).

Accuracy (%)	Precision (%)	Recall (%)	F_1 score (%)	Indice di Jaccard (%)
48.6 \pm 7.4	85.8 \pm 6.9	37.9 \pm 11.2	51.6 \pm 12.8	35.6 \pm 10.2

Tabella 3.7. Metriche di classificazione relative ai 51 trials del dataset complessivo: media e deviazione standard (algoritmo di segmentazione implementato senza post-processing).

Rispetto ai valori in **Tabella 3.7.**, le prestazioni del modello (perlomeno, in termini di accuracy, recall, F_1 score ed indice di Jaccard) sono insoddisfacenti. In particolare, la confusion matrix (**Figura 3.5.**) mostra una percentuale elevata di falsi negativi. Al fine di (almeno) ridurre tale valore, l'algoritmo di segmentazione viene ottimizzato,

testando, prima, entrambi i valori candidati-soglia (2000 campioni e 4000 campioni) su ciascuno dei 51 trials del dataset complessivo. Sono di seguito riportati, in percentuale, i risultati medi (\pm deviazione standard) in termini di confusion matrix (**Figura 3.6.**) e di metriche di classificazione (**Tabella 3.8.**).

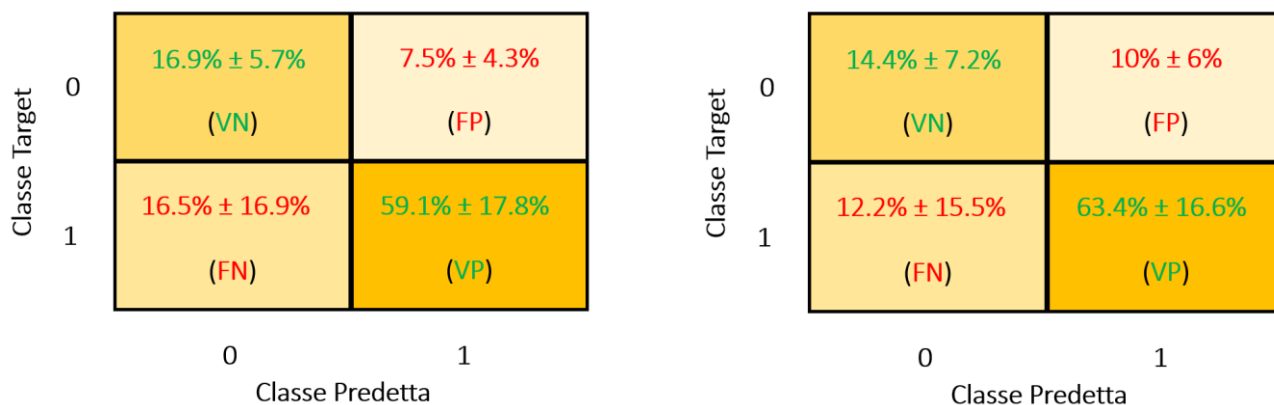


Figura 3.6. Confusion Matrix di classificazione dei 51 trials del dataset complessivo: media e deviazione standard (algoritmo di segmentazione implementato con post-processing: a sinistra, soglia 2000 campioni e, a destra, 4000 campioni).

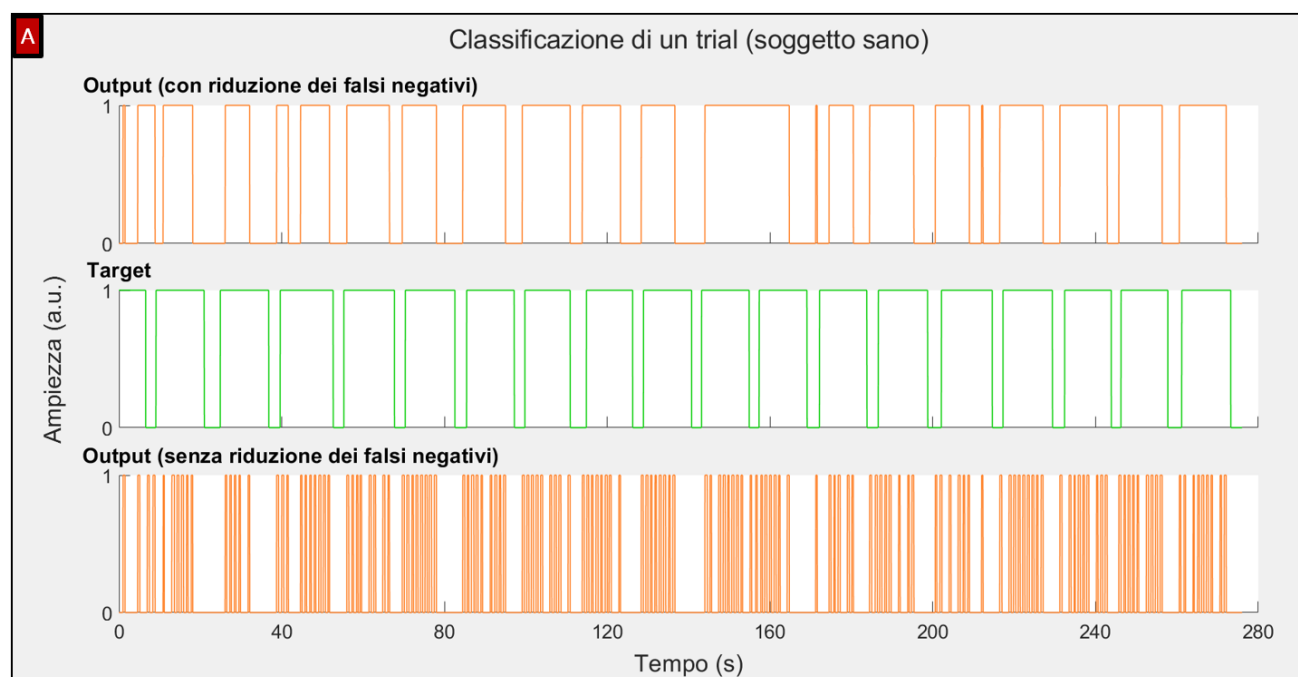
Soglia (campioni)	Accuracy (%)	Precision (%)	Recall (%)	F_1 score (%)	Indice di Jaccard (%)
2000	76.0 \pm 15.1	88.3 \pm 7.1	78.0 \pm 23.0	80.6 \pm 19.2	70.6 \pm 20.2
4000	77.8 \pm 13.1	86.4 \pm 7.5	83.6 \pm 21.1	83.0 \pm 17.5	73.5 \pm 17.8

Tabella 3.8. Metriche di classificazione relative ai 51 trials del dataset complessivo: media e deviazione standard (algoritmo di segmentazione implementato con post-processing: soglia 2000 campioni e 4000 campioni).

Dal confronto dei risultati ottenuti per i due valori-candidati (**Tabella 3.8.**), la soglia adottata in fase di post-processing dell'algoritmo di segmentazione è pari a 4000 campioni. Confrontando la confusion matrix di **Figura 3.5.** con quella di **Figura 3.6.**

relativa alla soglia di 4000 campioni (a destra), la percentuale di falsi negativi diminuisce in modo significativo (passando dal $46.9\% \pm 8.4\%$ al $12.2\% \pm 15.5\%$); invece, l'aumento di quella dei falsi positivi non è significativo (passando dal $4.6\% \pm 2.1\%$ al $10\% \pm 6\%$). Inoltre, la percentuale dei veri positivi aumenta significativamente (passando dal $28.7\% \pm 8.5\%$ al $63.4\% \pm 16.6\%$) e quella dei veri negativi diminuisce in modo non significativo (passando dal $19.8\% \pm 4.4\%$ al $14.4\% \pm 7.2\%$). Tutti i valori delle metriche di classificazione in **Tabella 3.7.** migliorano in seguito all'applicazione del post-processing (**Tabella 3.8.**). In particolare, l'accuracy del $48.6\% \pm 7.4\%$ (**Tabella 3.7.**) diventa pari al $77.8\% \pm 13.1\%$ con la soglia scelta (**Tabella 3.8.**).

In **Figura 3.7.**, sono mostrati, rispettivamente per un trial eseguito da un soggetto sano e per uno eseguito da un paziente affetto da PD, due esempi di confronto tra il target, l'output dell'algoritmo senza post-processing e l'output dell'algoritmo corretto mediante post-processing.



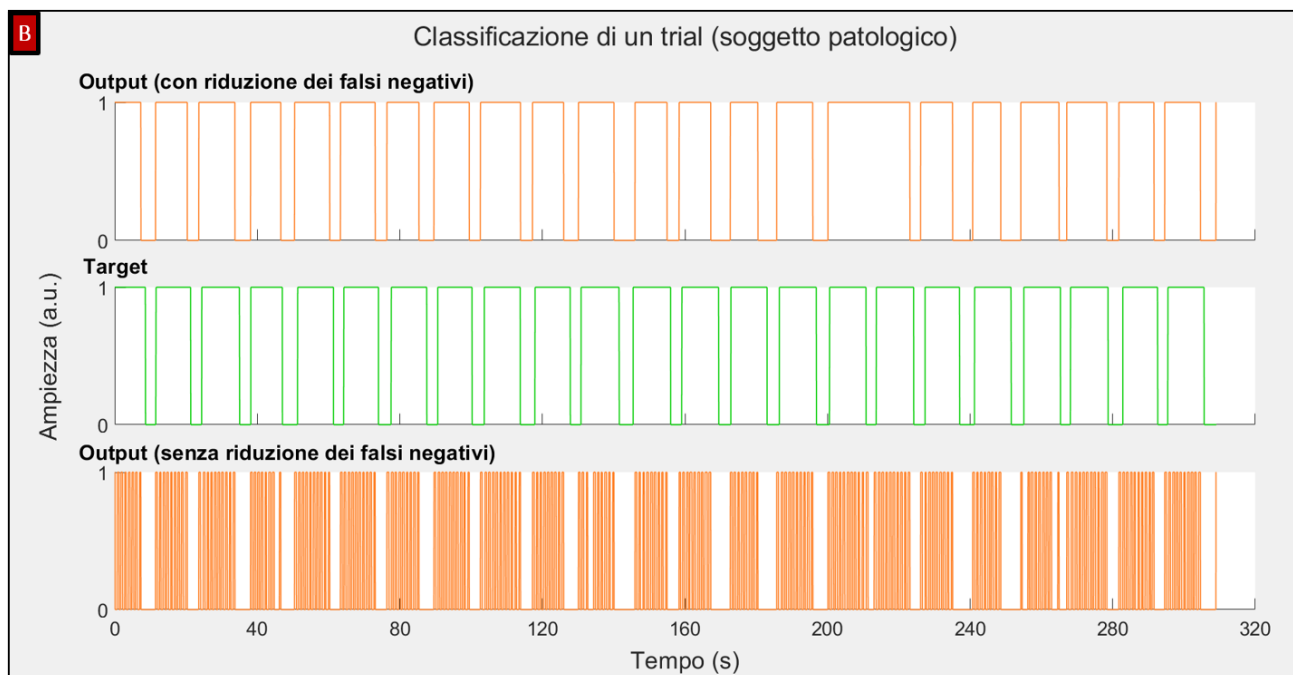


Figura 3.7. Esempio di classificazione di un intero trial (durata di circa 5 minuti). Per un trial di un soggetto sano (A) e per un trial di un soggetto patologico (B), entrambi compresi tra i 51 trials del dataset complessivo, sono mostrati il target, l'output dell'algoritmo senza post-processing e l'output dell'algoritmo con post-processing.

Infine, testando l'algoritmo sui 10 trials del dataset di validazione (non compresi tra i 51 trials del dataset complessivo), sono ottenuti i risultati medi (\pm deviazione standard) riportati in **Figura 3.8.**, in termini di confusion matrix e, in **Tabella 3.9.**, in termini di metriche di classificazione. In **Figura 3.9.**, sono mostrati, rispettivamente per un trial di un soggetto sano e per uno di un soggetto patologico, due esempi di funzionamento dell'algoritmo di segmentazione dei segnali sEMG in tratti rettilinei e non-rettilinei. Entrambi i trials appartengono al dataset di validazione e, di conseguenza, la rete neurale implementata nell'algoritmo finale non li ha mai "visti".

Classe Target	0	14.7% ± 7.7% (VN)	10.4% ± 6.2% (FP)
	1	13.5% ± 15.3% (FN)	61.4% ± 15.5% (VP)
		0	1
		Classe Predetta	

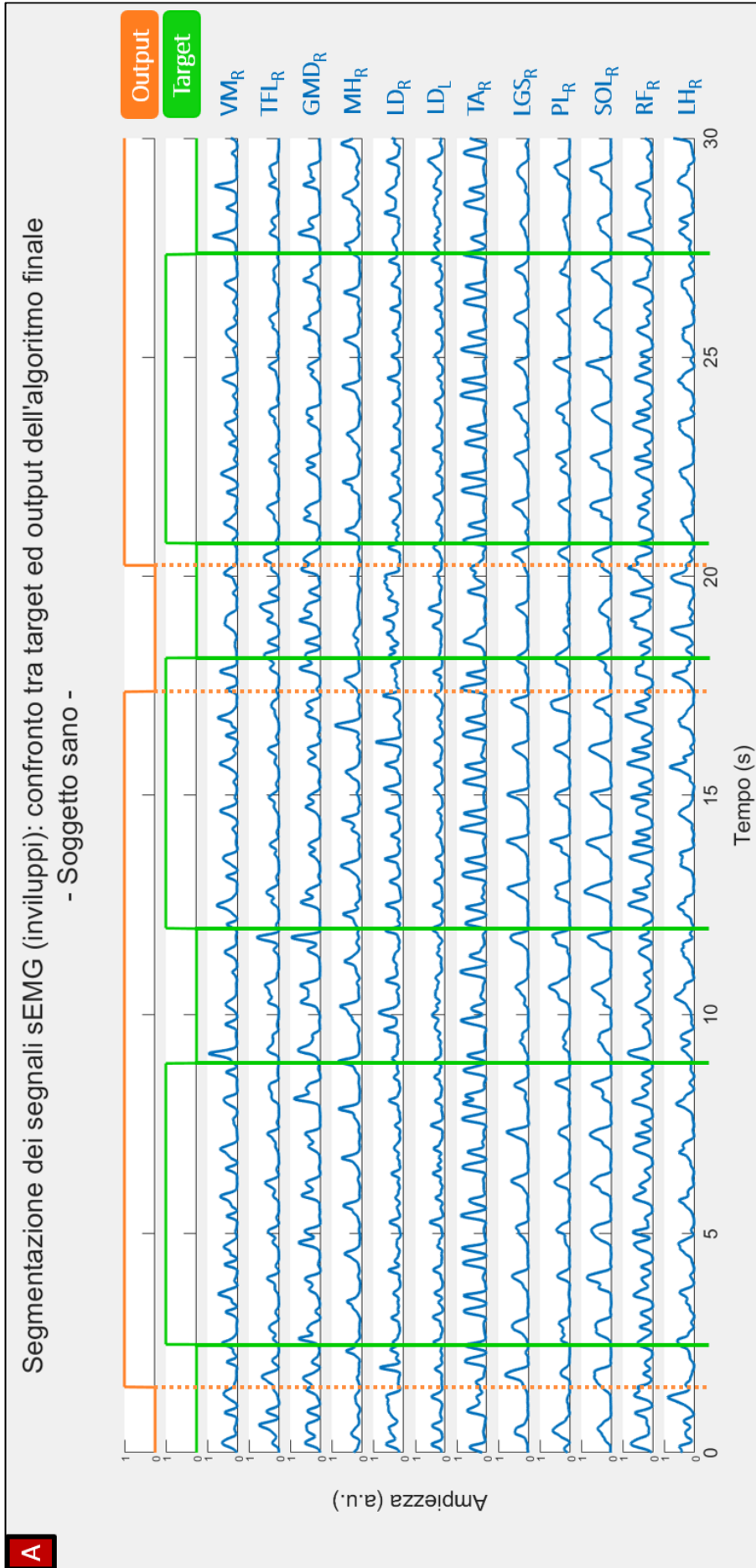
Figura 3.8. Confusion Matrix di classificazione dei 10 trials del dataset di validazione: media e deviazione standard.

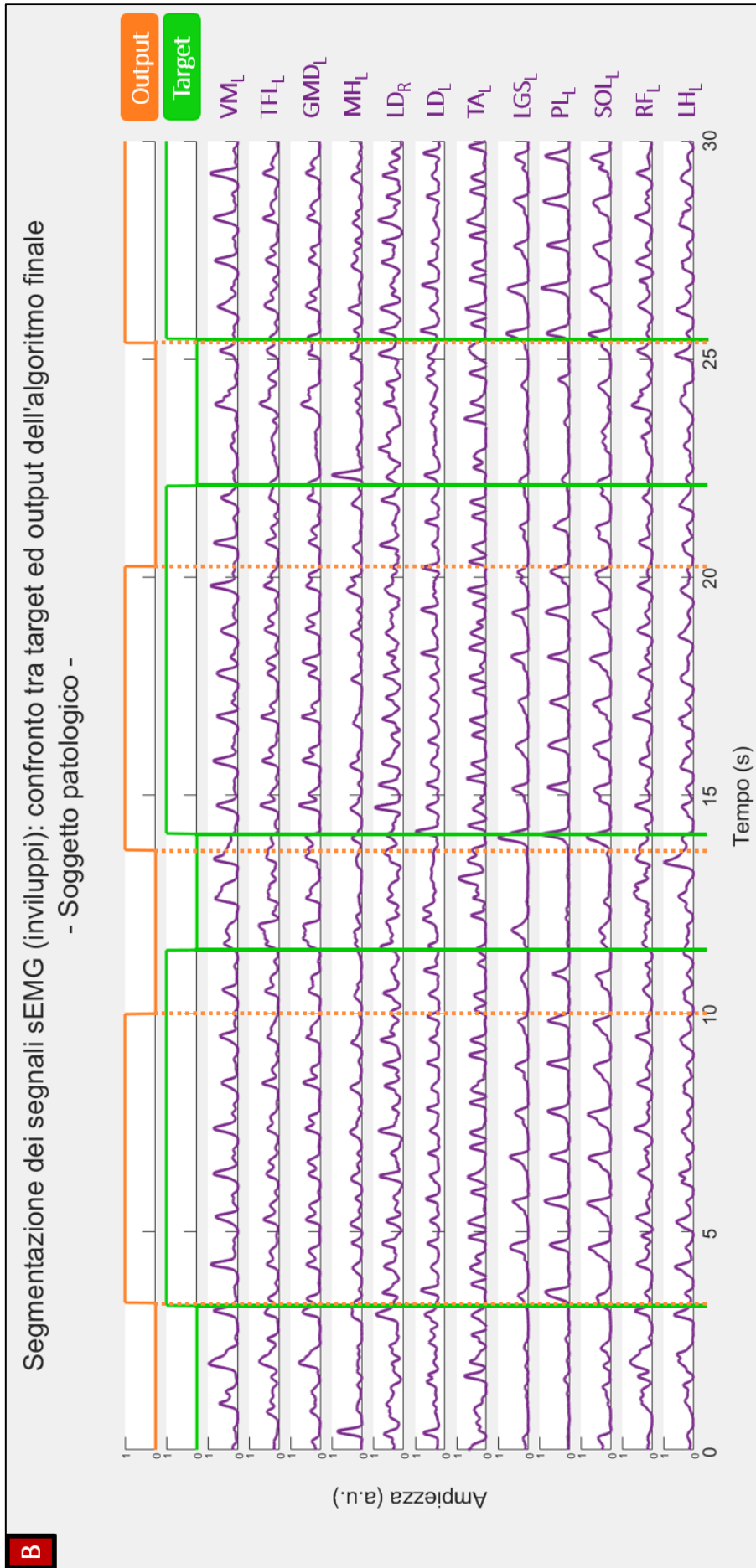
Accuracy (%)	Precision (%)	Recall (%)	F_1 score (%)	Indice di Jaccard (%)
76.1 ± 10.9	86.8 ± 5.5	82.0 ± 20.1	82.4 ± 11.6	71.4 ± 14.7

Tabella 3.9. Metriche di classificazione relative ai 10 trials del dataset di validazione: media e deviazione standard.

L'algoritmo finale, implementato su tale dataset di validazione, richiede un tempo computazionale medio (\pm deviazione standard) pari a 15.1 s \pm 1.0 s.

Figura 3.9. Involuppi dei segnali sEMG, target e output dell'algoritmo finale. Per un trial di un soggetto sano (A) e per un trial di un soggetto patologico (B), entrambi compresi tra i 10 trials del dataset di validazione, è mostrata soltanto una finestra temporale di 30 s. Gli involuppi dei segnali sEMG sono normalizzati in ampiezza. Il target e l'output valgono 0, se non-rettilineo, oppure 1, se rettilineo.





Capitolo 4

Discussione e Conclusioni

L'analisi del cammino, basata sui segnali sEMG acquisiti in condizione di locomozione libera (ad esempio, non su tapis roulant), incontra l'esigenza di distinguere i brani di segnale acquisiti nel percorrere un tratto rettilineo da quelli acquisiti nell'effettuare un cambio di direzione a 180°, descrivendo una "U", per ripercorrere il rettilineo in direzione opposta. La segmentazione dei segnali sEMG in tratti rettilinei e tratti non-rettilinei, obiettivo della presente tesi, deve essere automatica e svincolata dall'ausilio di segnali basografici e goniometrici, eventualmente acquisiti. L'indipendenza da tali segnali eviterebbe la necessità di acquisirli qualora fossero utilizzati esclusivamente come strumento per la segmentazione dei tratti di rettilineo. Inoltre, sarebbe possibile segmentare i segnali sEMG nei casi di relative acquisizioni basografiche o goniometriche corrotte oppure mancanti e qualora, in tali casi, non si disponga neppure della videoregistrazione del trial per effettuare manualmente la segmentazione. In letteratura, non sono stati trovati studi che affrontino direttamente il problema di automatizzare la segmentazione dei tratti rettilinei e non-rettilinei durante il cammino, sfruttando i soli segnali sEMG.

La presente tesi propone l'identificazione della tipologia dei tratti percorsi come un problema di classificazione binaria. Ovvero, al singolo campione dei segnali sEMG viene associata la corrispondente classe di appartenenza: classe dei tratti rettilinei (classe 1) oppure classe dei tratti non-rettilinei (classe 0). Il problema di classificazione viene

sviluppato in ambito di deep learning, mediante l'implementazione della rete neurale BiLSTM.

Globalmente, dopo la definizione del dataset, la tesi si articola in tre macrosezioni consecutive: (1) esplorazione dell'applicabilità delle reti neurali BiLSTM al problema in esame; (2) ottimizzazione del modello BiLSTM scelto; (3) definizione del funzionamento ed ottimizzazione dell'algoritmo finale per la segmentazione dei segnali sEMG in tratti rettilinei e non-rettilinei.

L'applicabilità delle reti neurali BiLSTM al problema in esame è analizzata mediante il dataset ridotto dei 26 trials, eseguiti dai soggetti sani. A tal fine, sono implementati e valutati entrambi gli approcci di classificazione (STL, STS), presenti in ambiente MATLAB®. L'approccio STL e l'approccio STS permettono di scorporare questo problema di classificazione binaria in due sottoproblemi: l'analisi dei tratti (approccio STL) e l'analisi delle transizioni (approccio STS). L'obiettivo di entrambe le analisi rimane la distinzione tra la classe dei tratti rettilinei e la classe dei tratti non-rettilinei. Rispetto al percorso del protocollo sperimentale (cfr. 2.2), le due analisi presuppongono una diversa costruzione del dataset (di fatto, vengono costruiti due dataset differenti) e permettono di classificare i segnali sEMG con risoluzione differente (classificazione unica dell'intero brano di segnale acquisito lungo un tratto, in approccio STL; classificazione del singolo campione del segnale, acquisito lungo una transizione, in approccio STS). Per singolo approccio, viene selezionato il modello migliore tra gli 80 modelli BiLSTM implementati. Si tratta del modello (unico per singolo approccio) che soddisfa entrambi i criteri di selezione (cfr. 2.4.2), ovvero è il modello con scostamento tra training accuracy e validation accuracy non superiore al 5% e con testing accuracy massima. I due modelli così selezionati sono poi valutati criticamente. Nella fattispecie, essi vengono confrontati, quantitativamente, in termini di prestazioni e di valori degli iperparametri e, qualitativamente, rispetto ai vantaggi e agli svantaggi degli approcci

STL e STS (per come sfruttati nella presente tesi). Innanzitutto, si riscontra che i due modelli sono confrontabili in termini di testing accuracy (e, anche, di training accuracy e di validation accuracy). Sono, altresì, confrontabili rispetto ai valori degli iperparametri: differiscono, infatti, soltanto per il valore iniziale del learning rate. I due valori sono però “vicini” tra di essi, in quanto si tratta, rispettivamente, del primo e del secondo valore dell’intervallo di variazione considerato per α (0.005 per STL; 0.01 per STS). Tuttavia, relativamente ai tempi computazionali medi (\pm deviazione standard), perlomeno nella classificazione del rispettivo training set e, in cascata, del rispettivo testing set, i due approcci non sono statisticamente confrontabili. L’approccio STS richiede tempo computazionale medio superiore al doppio di quello necessario in approccio STL. Inoltre, l’approccio STS implica anche una maggiore complessità in termini di concettualizzazione del problema e di costruzione del dataset. Esso infatti viene sfruttato per la classificazione binaria dei tratti però a partire dalle transizioni. L’analisi delle transizioni richiede di distinguere le quattro transizioni presenti lungo il percorso del soggetto e di garantire che ciascuna transizione sia presente per il 25% del dataset (primo bilanciamento) e sia in sé bilanciata rispetto alle due classi (secondo bilanciamento). Poiché l’approccio STS classifica con risoluzione del campione, il secondo bilanciamento impone che tutte le transizioni siano opportunamente “ritagliate” dalla matrice dei segnali sEMG per avere lo stesso numero di campioni (tale numero di campioni è pari al doppio di quello del tratto non-rettilineo più breve). L’approccio STL, invece, permette una maggiore semplicità, classificando direttamente i tratti. L’analisi dei tratti richiede esclusivamente di garantire la loro equa presenza nel dataset (ciascuna tipologia per il 50%), indipendentemente dall’effettivo numero di campioni del singolo tratto. Di conseguenza, risulta di gran lunga preferibile l’approccio STL. Il modello STL selezionato viene scelto come migliore tra i due e a partire da esso viene sviluppato l’algoritmo finale di segmentazione automatica dei segnali sEMG.

La macrosezione di esplorazione dell'applicabilità delle reti neurali BiLSTM al problema si conclude con la decisione di trascurare il modello STS (e dunque l'approccio STS) e di procedere all'ottimizzazione del solo modello STL.

Il modello BiLSTM "vincente" viene ottimizzato innanzitutto provando a variarne l'architettura. Le 13 prove di variazione dell'architettura si configurano come tentativi di alterare la successione dei 5 layers. Delle 13 prove effettuate soltanto due (prova 2 e prova 10) presentano scostamento tra training accuracy e validation accuracy non superiore alla soglia del 5% (prova 2: 4.3%; prova 10: 0%). Di fatto, però, la prova 10 è del tutto insoddisfacente, in quanto i tre valori di accuracy sono pari a 50%. A rigore, il valore di testing accuracy del modello STL selezionato (71.6%) è superiore a quello del modello STL della prova 2 (71.4%). Inoltre, anche giudicando trascurabile la differenza tra i due valori (0.2%), l'architettura del modello "vincente" (cfr. **2.4.2**) viene comunque mantenuta intatta perché più semplice (cfr. **2.5.1**). Sostanzialmente, la successione dei 5 layers rimane inalterata, senza introdurre (come nella prova 2) il Dropout Layer in cascata al BiLSTM Layer.

Successivamente, al fine di valutare la robustezza delle prestazioni del modello selezionato e la sua capacità di generalizzazione, il dataset ridotto viene ampliato inglobando tutti i 51 trials delle sottopopolazioni sana e patologica e su tale dataset complessivo viene implementata la 10-fold cross-validation. Rispetto ai risultati sul testing set, il modello presenta accuracy media (\pm deviazione standard) pari al $71.7\% \pm 7.2\%$, con valore minimo pari al 61.1% e massimo pari all'80.0%. A fronte del valore originario di testing accuracy (71.6%), nel contesto degli 80 allenamenti, il modello STL selezionato si presenta robusto. Tale robustezza risulta ancora più evidente considerando che il valore di accuracy è pari al 71.6%, sul dataset ridotto (26 trials dei soli soggetti sani), ed è pari al $71.7\% \pm 7.2\%$, sul dataset complessivo (51 trials dei

soggetti sani e dei soggetti patologici). Inoltre, per 9 delle 10 iterazioni lo scostamento tra training accuracy e testing accuracy non supera la soglia del 5% (iterazione 7: 5.9%). Infine, il modello STL selezionato viene ottimizzato valutando l'impatto che numerosità differenti di canali sEMG hanno sulle prestazioni del classificatore. A tal fine, viene implementata la 10-fold cross-validation sul dataset complessivo dei 51 trials. Rispetto ai 5 set considerati (cfr. **2.5.3**), i risultati medi (\pm deviazione standard) delle metriche di classificazione sul testing set mostrano che, al diminuire del numero di canali sEMG considerati, le prestazioni del classificatore, mediamente, peggiorano. In particolare, l'accuracy media (\pm deviazione standard), pari al $52.6\% \pm 4.3\%$, del set 1 (che include 2 canali sEMG: LD_R e LD_L) aumenta progressivamente all'aumentare del numero di canali sEMG considerati, fino al $66.4\% \pm 8.3\%$ del set 5 (che include 7 canali sEMG: TA, LGS, RF, LH, LD_R, LD_L, VM). Confrontando tali risultati con l'accuracy media (\pm deviazione standard) in presenza dei 12 canali sEMG, pari al $71.7\% \pm 7.2\%$, è evidente che le prestazioni del modello, mediamente, peggiorano. Sostanzialmente, la riduzione del numero di canali sEMG comporta perdita di informazione. Verosimilmente, il problema della perdita di contenuto informativo è da ricondurre anche all'acquisizione monolaterale dei segnali sEMG. Ovvero, nell'effettuare un cambio di direzione a 180°, i muscoli di un arto inferiore manifestano patterns di attivazione differenti a seconda che l'arto stesso sia quello interno oppure esterno rispetto alla curva. In particolare, si attivano maggiormente quando l'arto è mantenuto esternamente alla curva. Di fatto, rispetto al percorso del protocollo sperimentale (cfr. **2.2**), per uno stesso muscolo, in coincidenza dei tratti non-rettilinei del percorso, la rete neurale apprende due comportamenti differenti. Di conseguenza, il classificatore, allenato sui 12 canali sEMG di un solo arto inferiore e del tronco, viene messo in crisi nel riconoscimento dei tratti non-rettilinei. Invece, i tratti rettilinei non sono interessati da tale fenomeno. Allo stato della presente tesi, la criticità dei tratti non-rettilinei non può essere superata, in quanto sarebbe necessario possedere un dataset di segnali sEMG bilaterali, rispetto al singolo

trial. Tale variazione nella “natura” del dataset rappresenta, di fatto, un interessante sviluppo futuro per il metodo proposto. Inoltre, a partire dagli stessi trials considerati nella tesi, si potrebbero però costruire due dataset differenti per verso di percorrenza (orario oppure antiorario) dei tratti non-rettilinei. Su tali dataset potrebbero essere ripetuti i 160 allenamenti negli approcci STL e STS per confrontarne i risultati con quelli ottenuti nella presente tesi.

A partire dal modello STL selezionato ed ottimizzato, confermando l’architettura iniziale e mantenendo tutti i 12 canali sEMG acquisiti, viene, in ultimo, definito l’algoritmo finale per la segmentazione automatica dei segnali sEMG. Il classificatore implementato nell’algoritmo è la rete neurale individuata come migliore tra le dieci della 10-fold cross-validation. Si tratta di quella corrispondente all’iterazione 4, per la quale lo scostamento tra training accuracy e testing accuracy non supera la soglia del 5% e la testing accuracy è la massima (80.0%). L’algoritmo, implementato sui 51 trials del dataset complessivo, fornisce prestazioni insoddisfacenti, in termini di confusion matrix, accuracy, precision, recall, F_1 score ed indice di Jaccard. In particolare, l’accuracy media (\pm deviazione standard) è pari al $48.6\% \pm 7.4\%$ (mentre il valore di testing accuracy della rete neurale su cui l’algoritmo si basa è pari all’80%). A seguito di tali risultati, l’algoritmo viene ottimizzato in termini di post-processing implementato sul suo output. L’obiettivo del post-processing è la riduzione dei falsi negativi, applicando una soglia in campioni, scelta testando due valori candidati-soglia (2000 campioni e 4000 campioni).

La confusion matrix media (\pm deviazione standard) relativa alla classificazione dei 51 trials del dataset complessivo, senza implementazione di alcun post-processing, presenta il valore dei falsi negativi pari al $46.9\% \pm 8.4\%$. Tale valore si riduce al $16.5\% \pm 16.9\%$, se la soglia è di 2000 campioni, e al $12.2\% \pm 15.5\%$, se la soglia è di 4000 campioni. La riduzione dei falsi negativi, per come sviluppata in fase di post-processing, potrebbe comportare un aumento (magari significativo) dei falsi positivi. Il valore medio

(\pm deviazione standard) dei falsi positivi pari al $4.6\% \pm 2.1\%$ diventa pari al $7.5\% \pm 4.3\%$, con soglia di 2000 campioni, e pari al $10.0\% \pm 6.0\%$, con la soglia di 4000 campioni. Rispetto a tali risultati, che sono confrontabili, mediamente, con la soglia di 4000 campioni è maggiore la riduzione dei falsi negativi, ma è maggiore anche l'aumento dei falsi positivi; viceversa, con la soglia di 2000 campioni, la riduzione dei falsi negativi è minore, ma è minore anche l'aumento dei falsi positivi. Confrontando, per i due valori candidati-soglia, i risultati medi (\pm deviazione standard) delle metriche di classificazione sull'intero dataset, la soglia scelta è 4000 campioni (soltanto il valore medio di precision è maggiore nel caso di 2000 campioni). L'algoritmo finale, con imposizione della soglia di 4000 campioni, viene testato anche su un ulteriore dataset di validazione, avente dimensione pari al 19.6% della dimensione del dataset complessivo (10 trials vs 51 trials). I 10 trials adottati per la validazione dell'algoritmo non sono compresi nel dataset dei 51 trials. Le metriche di classificazione presentano risultati medi (\pm deviazione standard) confrontabili tra i due casi. In particolare, l'accuracy media (\pm deviazione standard) è pari al $77.8\% \pm 13.1\%$, sul dataset dei 51 trials, ed è pari al $76.1\% \pm 10.9\%$, sul dataset dei 10 trials.

L'algoritmo di segmentazione automatica dei segnali sEMG, sviluppato nella tesi, si configura come un metodo "general purpose" applicabile in analisi del cammino, nella condizione di locomozione libera, ogni qualvolta sia di interesse la distinzione tra percorsi rettilinei e non-rettilinei a partire dai soli segnali sEMG. Allo stato della presente tesi, l'algoritmo finale rappresenta una prima soluzione robusta a tale problema di segmentazione automatica. Inoltre, perlomeno sulla risorsa hardware utilizzata, esso permette di segmentare un'intera acquisizione di segnali sEMG della durata di circa 5 minuti richiedendo, mediamente, poco più di una decina di secondi.

Sviluppi futuri potrebbero includere la validazione dell'algoritmo in presenza di soggetti con locomozione compromessa da altre patologie neurologiche di interesse in analisi

del cammino (ad esempio, NPH (Normal Pressure Hydrocephalus), atassia e paralisi cerebrale) [8]. Al fine di migliorarne le prestazioni attuali, l'algoritmo potrebbe essere ottimizzato in termini di post-processing implementato sul suo output, ad esempio rivalutandone il criterio-base (la riduzione dei falsi negativi). Sarebbe però opportuno sostituire al criterio attuale o associare ad esso un metodo altrettanto generalizzabile e indipendente dalla conformazione del percorso. Inoltre, potrebbe essere interessante sviluppare l'analisi dei tratti anche in approccio STS, valutando le prestazioni delle reti neurali BiLSTM nel caso di classificazione con risoluzione del campione. In ultimo, si potrebbero valutare le prestazioni in caso di allenamento con features, rilevanti per il segnale sEMG, opportunamente ingegnerizzate.

Il metodo presentato è applicabile off-line, in quanto la rete neurale BiLSTM classifica l'*i*-esimo campione sfruttando sia l'informazione passata che quella futura, ovvero, rispettivamente, campioni ad esso precedenti e successivi. L'applicazione in real-time sarebbe possibile soltanto a partire da una rete neurale feedforward. In particolare, per convertire l'algoritmo attuale in algoritmo in real-time sarebbe opportuno adottare un modello LSTM (feedforward). Considerate le affinità tra LSTM e BiLSTM, tale modello potrebbe essere selezionato ed ottimizzato ripetendo i passi del flusso di lavoro della presente tesi. Di fatto, sfruttando gli stessi dataset costruiti nella tesi, il punto di partenza potrebbe essere direttamente l'implementazione degli stessi 80 modelli in approccio STL e in approccio STS.

Appendice A

Rete neurale BiLSTM

A.1 Unità LSTM

Sono di seguito riportati dettagli matematici che caratterizzano l'unità LSTM (memory block) implementata in MATLAB® release R2019b (The MathWorks Inc., Natick, MA, USA).

In una rete neurale BiLSTM l'unità LSTM presenta due strutture speculari (**Figura A.1.**): una appartiene ai memory blocks della direzione forward, l'altra ai memory blocks della direzione backward. Al time step t , lo stato dell'unità LSTM si compone di cell state (c_t) e hidden state (h_t). Il cell state contiene l'informazione appresa dai precedenti time step. L'hidden state, anche chiamato output state, contiene l'output dell'unità LSTM per il time step t . Per ciascun time step, l'unità LSTM aggiunge informazione al cell state o rimuove informazione dal cell state, controllando questi aggiornamenti mediante i tre gates (input gate, forget gate, output gate). Il cell state e l'hidden state sono controllati dai seguenti componenti: cell candidate (g), che aggiunge informazione al cell state, input gate (i), che controlla il livello di aggiornamento del cell state, forget gate (f), che controlla il livello di azzeramento del cell state, output gate (o), che controlla il livello del cell state aggiunto all'hidden state.

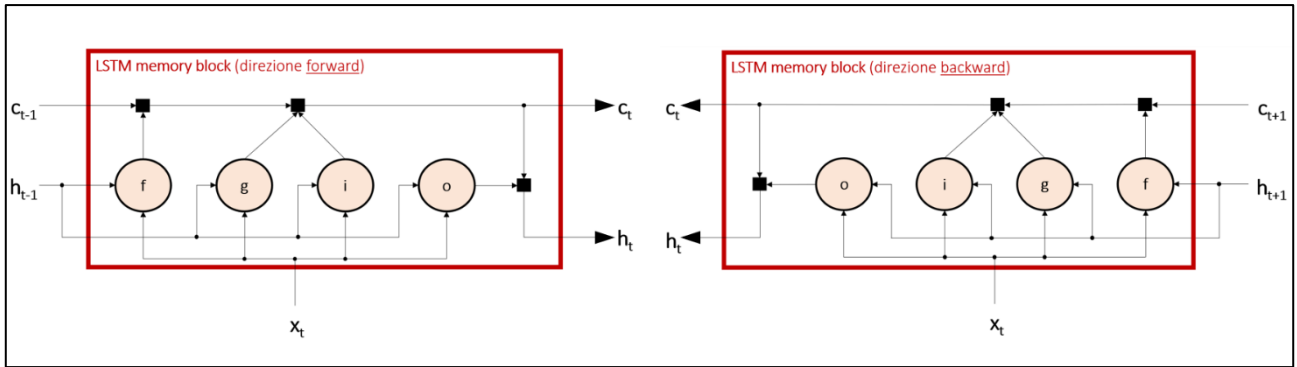


Figura A.1. LSTM memory blocks. Al time step t , il memory block della direzione forward (a sinistra) e quello della direzione backward (a destra) processano sia l'input attuale (x_t) che l'informazione del time step, rispettivamente, precedente e successivo e danno in output il cell state c_t e l'hidden state h_t . Ciascun memory block ingloba quattro componenti fondamentali: cell candidate (g), input gate (i), forget gate (f) e output gate (o).

I pesi che l'unità LSTM apprende sono gli input weights W , i recurrent weights R e il bias b . Le matrici W , R e b sono concatenazioni, rispettivamente, degli input weights, dei recurrent weights e del bias di ciascun componente. Tali matrici sono concatenate come in (1):

$$W = \begin{bmatrix} W_i \\ W_f \\ W_g \\ W_o \end{bmatrix}, \quad R = \begin{bmatrix} R_i \\ R_f \\ R_g \\ R_o \end{bmatrix}, \quad b = \begin{bmatrix} b_i \\ b_f \\ b_g \\ b_o \end{bmatrix} \quad (1)$$

In (1), i , f , g e o indicano, rispettivamente, l'input gate, il forget gate, il cell candidate e l'output gate.

Il cell state al time step t è pari a:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (2)$$

In (2), \odot indica il prodotto di Hadamard (moltiplicazione di vettori elemento per elemento).

L'hidden state al time step t è pari a:

$$\mathbf{h}_t = \mathbf{o}_t \odot \sigma_c(\mathbf{c}_t) \quad (3)$$

In (3), σ_c indica la funzione di attivazione dello stato. In MATLAB®, di default, tale funzione è calcolata mediante la funzione tangente iperbolica (\tanh).

Al time step t i componenti sono calcolati come in (4) - (7):

$$i_t = \sigma_g(W_i \mathbf{x}_t + R_i \mathbf{h}_{t-1} + b_i) \quad (4)$$

$$f_t = \sigma_g(W_f \mathbf{x}_t + R_f \mathbf{h}_{t-1} + b_f) \quad (5)$$

$$g_t = \sigma_c(W_g \mathbf{x}_t + R_g \mathbf{h}_{t-1} + b_g) \quad (6)$$

$$o_t = \sigma_g(W_o \mathbf{x}_t + R_o \mathbf{h}_{t-1} + b_o) \quad (7)$$

In (4), (5), (7), σ_g indica la funzione di attivazione del gate. In MATLAB®, di default, tale funzione è calcolata mediante la funzione sigmoide, data da $\sigma(x) = (1 + e^{-x})^{-1}$.

A.2 BiLSTM Layer

L'architettura interna del BiLSTM Layer, schematizzata in **Figura A.2.**, è caratterizzata da due sequenze di unità LSTM, una in direzione forward e l'altra in direzione backward. In presenza di una serie temporale in input X , di lunghezza S , con C features, ciascun time step viene processato da un'unità LSTM per singola direzione. Per il time step t , vengono calcolati l'output \mathbf{h}_t (anche noto come hidden state) e il cell state \mathbf{c}_t .

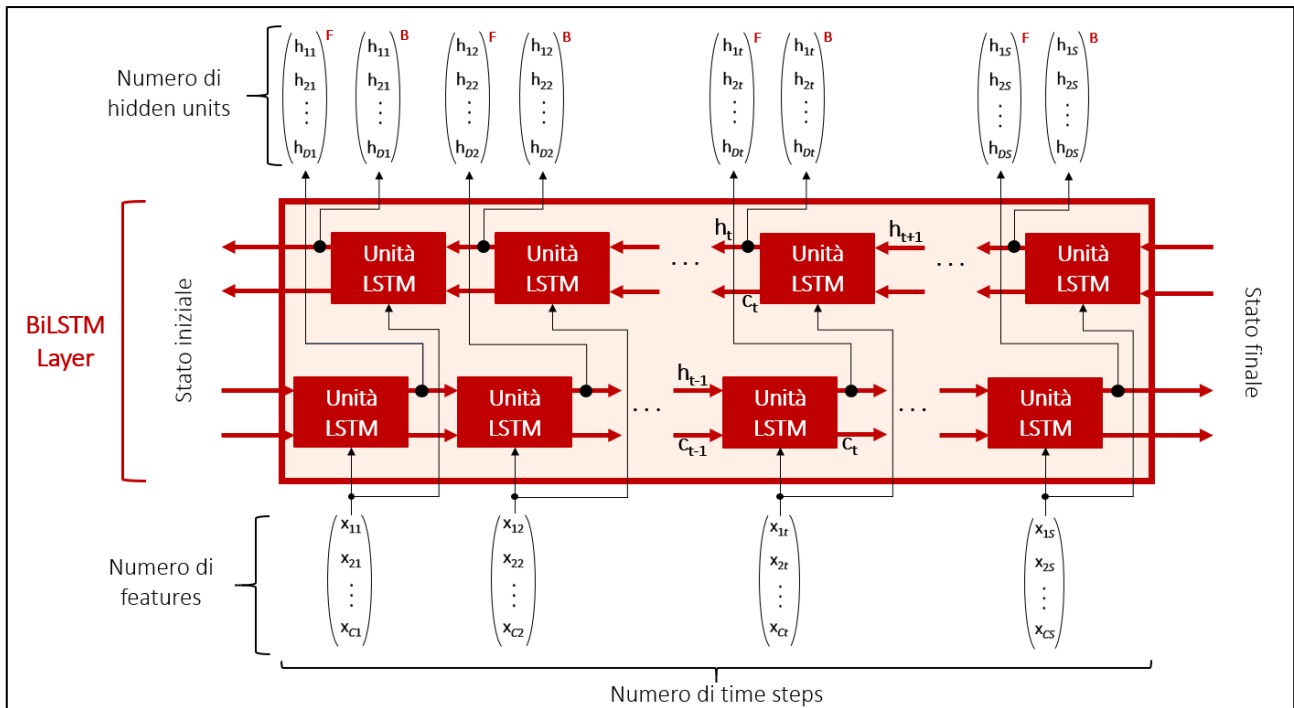


Figura A.2. BiLSTM Layer. La struttura interna è caratterizzata dalle unità LSTM in sequenza lungo le due direzioni temporali (forward e backward). In input riceve una serie temporale X , di lunghezza S e con C features. Al time step t , x_t viene processato da un'unità LSTM per singola direzione e vengono calcolati l'output h_t (hidden state) e il cell state c_t .

In direzione forward, la prima unità LSTM utilizza lo stato iniziale della rete neurale e il primo time step della sequenza per calcolare il primo output e il cell state aggiornato. Al time step t , l'unità utilizza lo stato attuale della rete (c_{t-1} , h_{t-1}) e il successivo time step della sequenza per calcolare l'output e il cell state aggiornato c_t . In direzione backward, il processing si sviluppa in modo speculare a partire dall'ultima unità LSTM.

A.3 Opzioni di Training

Al fine di selezionare il modello BiLSTM migliore, le 80 reti neurali, in approccio sia STL che STS, sono allenare settando le opzioni di training come di seguito riportato:


```
(1) 'adam'  
(2) 'MaxEpochs', 60  
(3) 'MiniBatchSize', round(0.2*numel(XTrain))  
(4) 'InitialLearnRate', [0.005 0.01 0.05 0.1]  
(5) 'LearnRateSchedule', 'piecewise'  
(6) 'LearnRateDropFactor', [0.05 0.1 0.15 0.2]  
(7) 'LearnRateDropPeriod', 15  
(8) 'SequenceLength', 'shortest'  
(9) 'Shuffle', 'once'  
(10) 'ValidationData', {XValidation, YValidation}  
(11) 'ValidationPatience', 5  
(12) 'GradientThreshold', 1  
(13) 'ExecutionEnvironment', 'cpu'
```

(1) Ottimizzatore Adam (adaptive moment estimation). Si tratta di un metodo di ottimizzazione di funzioni obiettivo, che, a partire dalla stima dei momenti del primo e del secondo ordine dei gradienti, calcola learning rates adattivi per parametri differenti [61].

(2) Massimo numero di epoche.

(3) Dimensione del mini-batch per ciascuna iterazione di training. Un mini-batch è un subset del training set utilizzato per valutare il gradiente della funzione di loss e aggiornare i pesi.

(4) Valore iniziale del learning rate.

(5) Opzione per il decremento del learning rate. Con 'piecewise', il learning rate viene aggiornato ad intervalli regolari, moltiplicandolo per un certo fattore. Il valore di tale fattore è specificato da 'LearnRateDropFactor'. Il numero di epoche tra moltiplicazioni è specificato da 'LearnRateDropPeriod'.

- (6) Fattore per decrementare il learning rate. È un fattore moltiplicativo da applicare al learning rate ogni volta che sia passato un certo numero di epoche.
- (7) Numero di epoche per il decremento del learning rate, che viene moltiplicato per il drop factor ogni volta che sia passato il numero specificato di epoche.
- (8) Opzione per gestire la lunghezza delle sequenze in input. Con 'shortest', le sequenze in ciascun mini-batch sono troncate per avere la stessa lunghezza della sequenza più breve. Non viene aggiunto padding, al costo di scartare i dati.
- (9) Opzione per il mescolamento dei dati. Con 'once', i dati di allenamento e di validazione vengono mescolati una sola volta prima dell'allenamento.
- (10) Dati da utilizzare per la validazione durante l'allenamento.
- (11) Pazienza nel fermare la validazione dell'allenamento della rete. È il numero di volte che la loss sul validation set può essere maggiore o uguale alla loss minima precedente prima che l'allenamento si interrompa.
- (12) Soglia sul gradiente. Se il gradiente supera tale valore, allora viene scalato.
- (13) Risorsa hardware per allenare la rete. Con 'cpu', è utilizzata la CPU.

Appendice B

Risultati

B.1 Definizione del modello BiLSTM

Relativamente agli 80 modelli di rete neurale BiLSTM implementati in approccio STL e agli 80 modelli implementati in approccio STS, sono di seguito riportati (**Figura B.1.**), in forma di grafico a barre, i risultati in termini di training accuracy, validation accuracy e testing accuracy. Al fine di una migliore visualizzazione, le 80 terne di accuracy, per singolo approccio, sono rappresentate in 5 grafici a barre distinti. Ciascun grafico rappresenta quindi le performance dei modelli BiLSTM testati aventi lo stesso numero di Hidden Units, facendo invece variare i valori di Initial Learning Rate (α) e di Learning Rate Drop Factor (δ)¹. I 16 modelli sono indicati lungo l'asse orizzontale di ciascun grafico mediante un numero progressivo da 1 a 16. Le 16 corrispondenze tra numero progressivo e combinazione di valori sono raccolte in **Tabella B.1.**

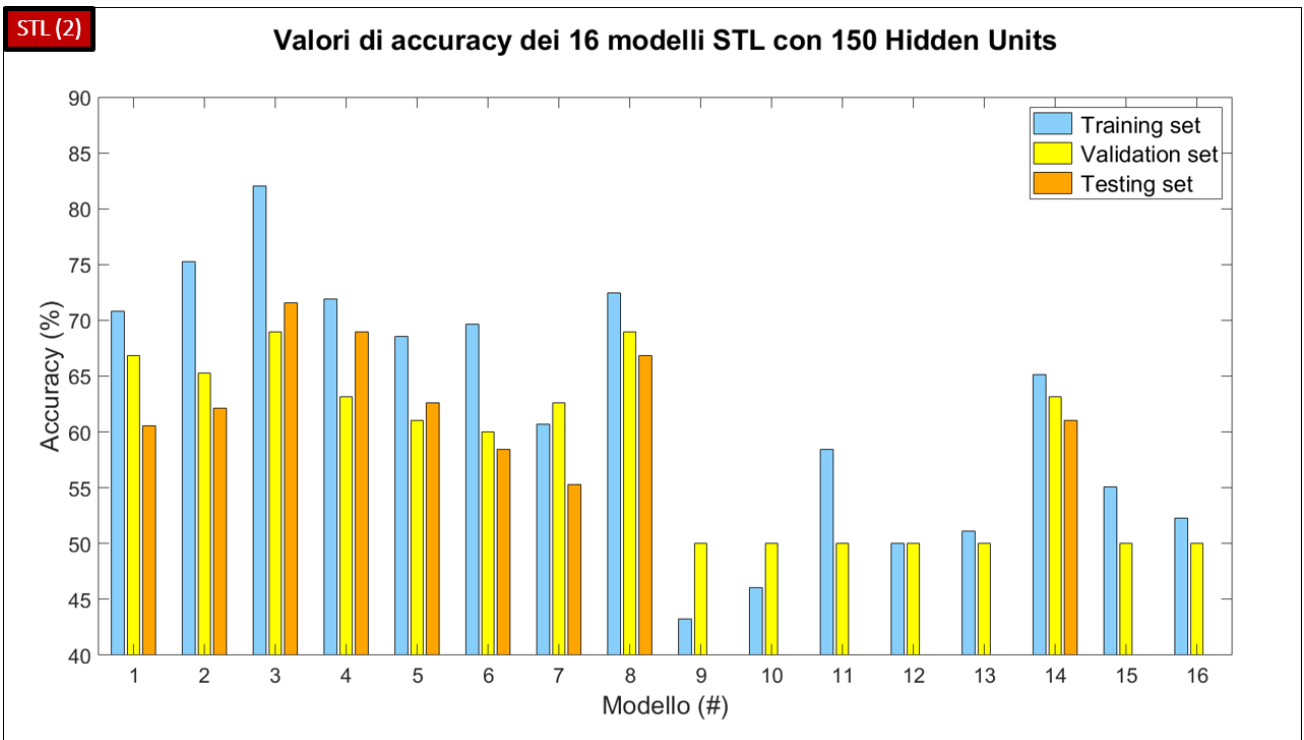
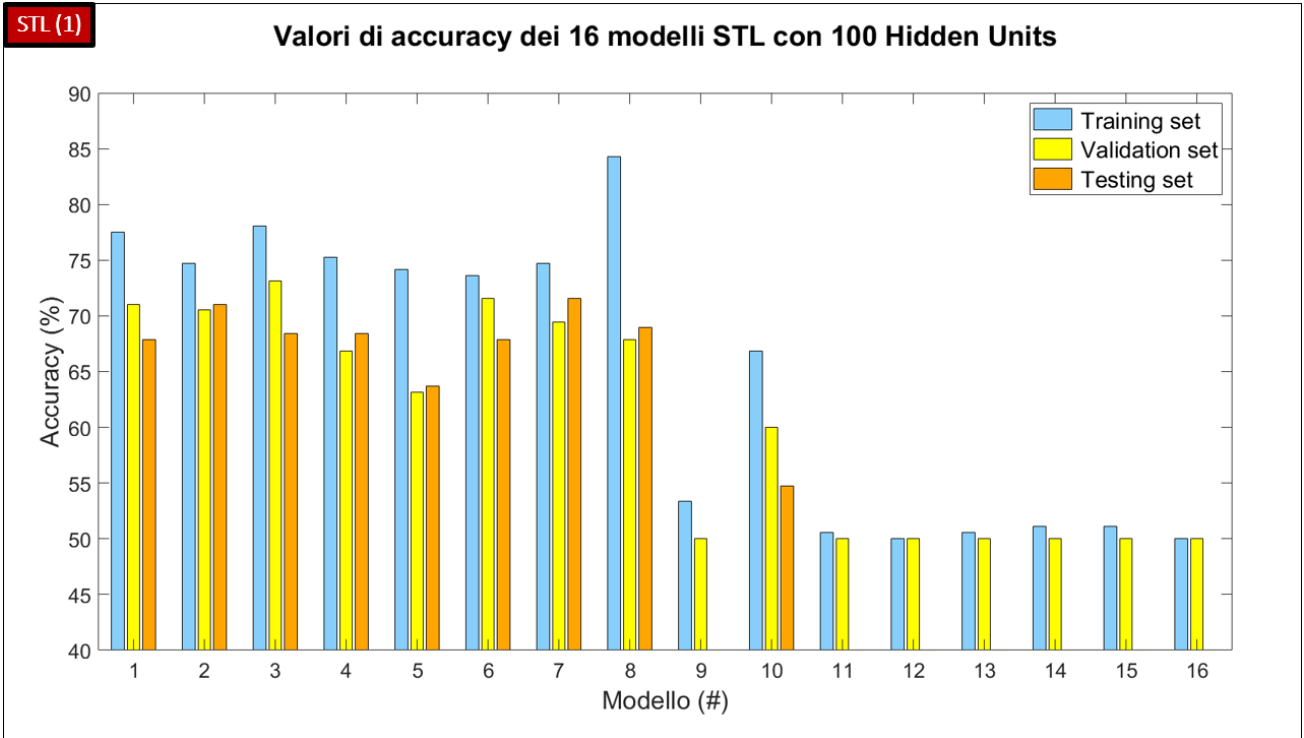
Rispetto ai risultati in **Figura B.1.**, per alcuni modelli la barra relativa alla testing accuracy non è visibile nel grafico. Tali modelli presentano, infatti, testing accuracy pari allo 0%.

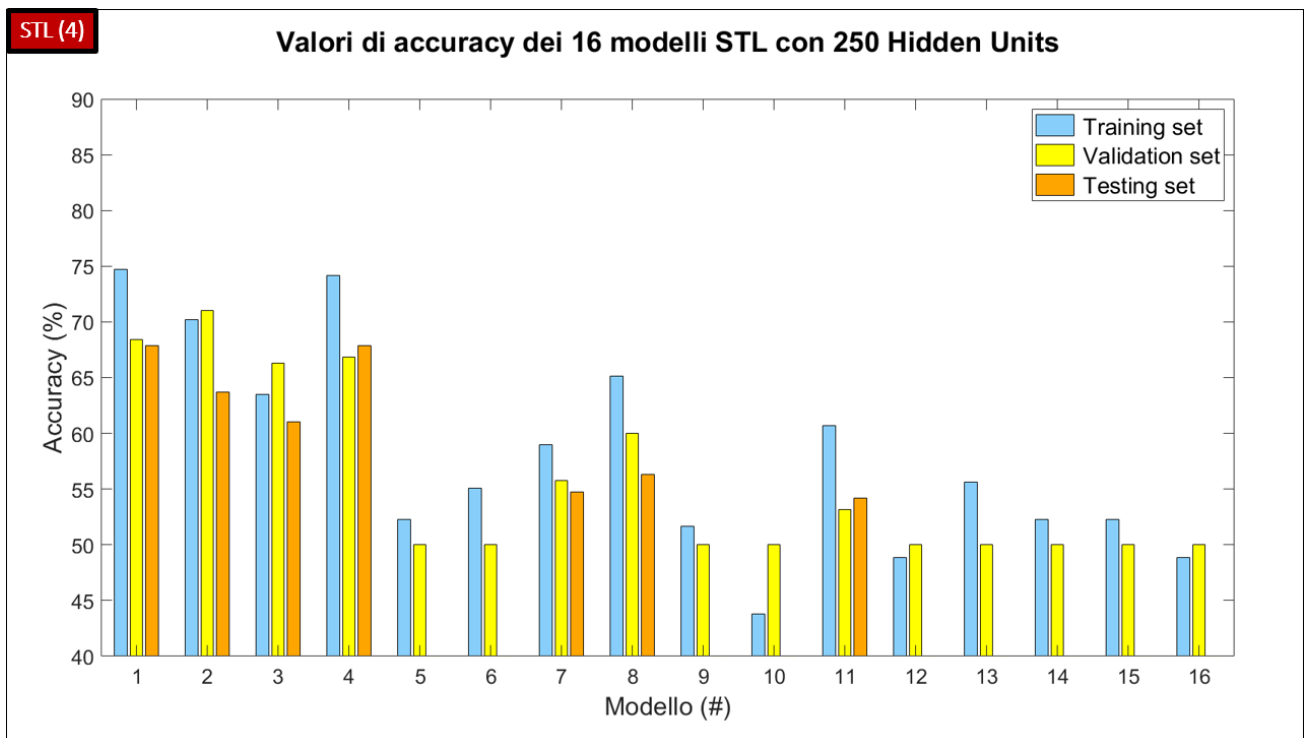
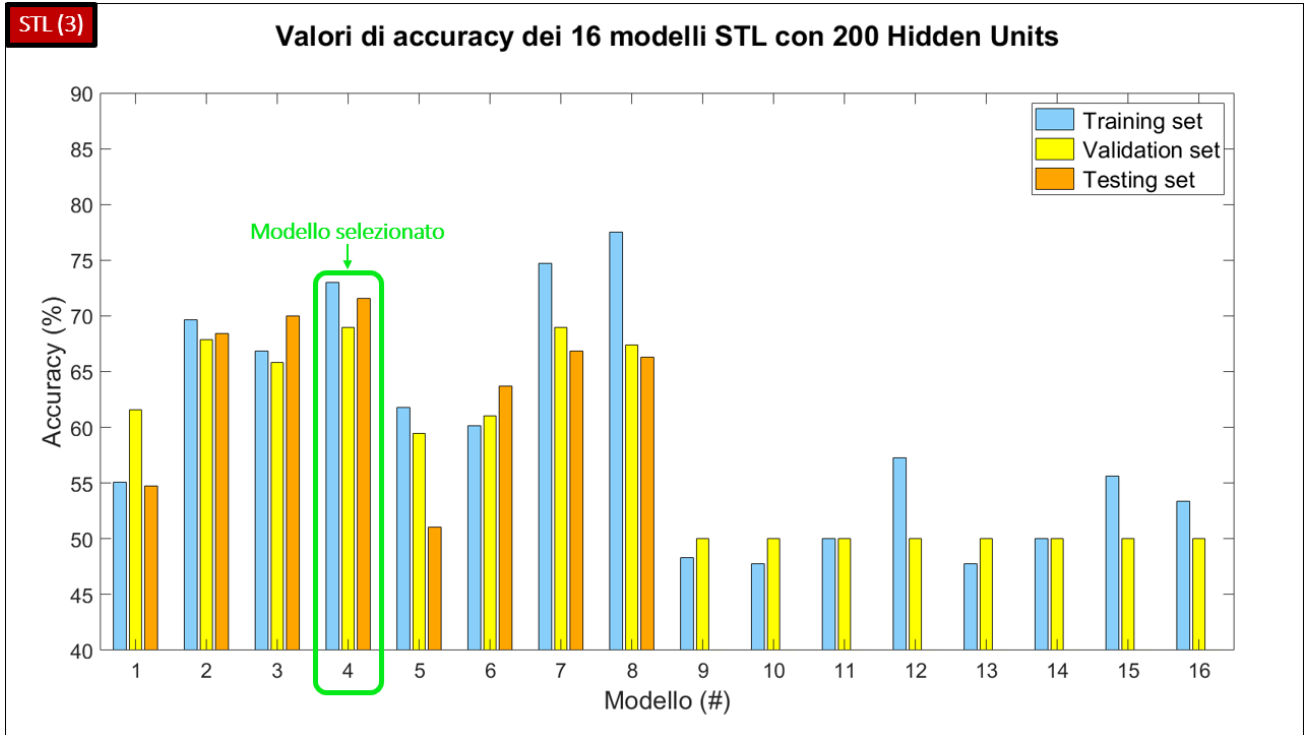
¹ Il numero di Hidden Units, l'Initial Learning Rate (α) e il Learning Rate Drop Factor (δ) sono gli iperparametri con variazione in fase di allenamento degli 80 modelli BiLSTM (per singolo approccio). Combinandone i rispettivi valori, sono infatti ottenute 80 combinazioni. Ciascuna combinazione costituisce un modello BiLSTM differente.

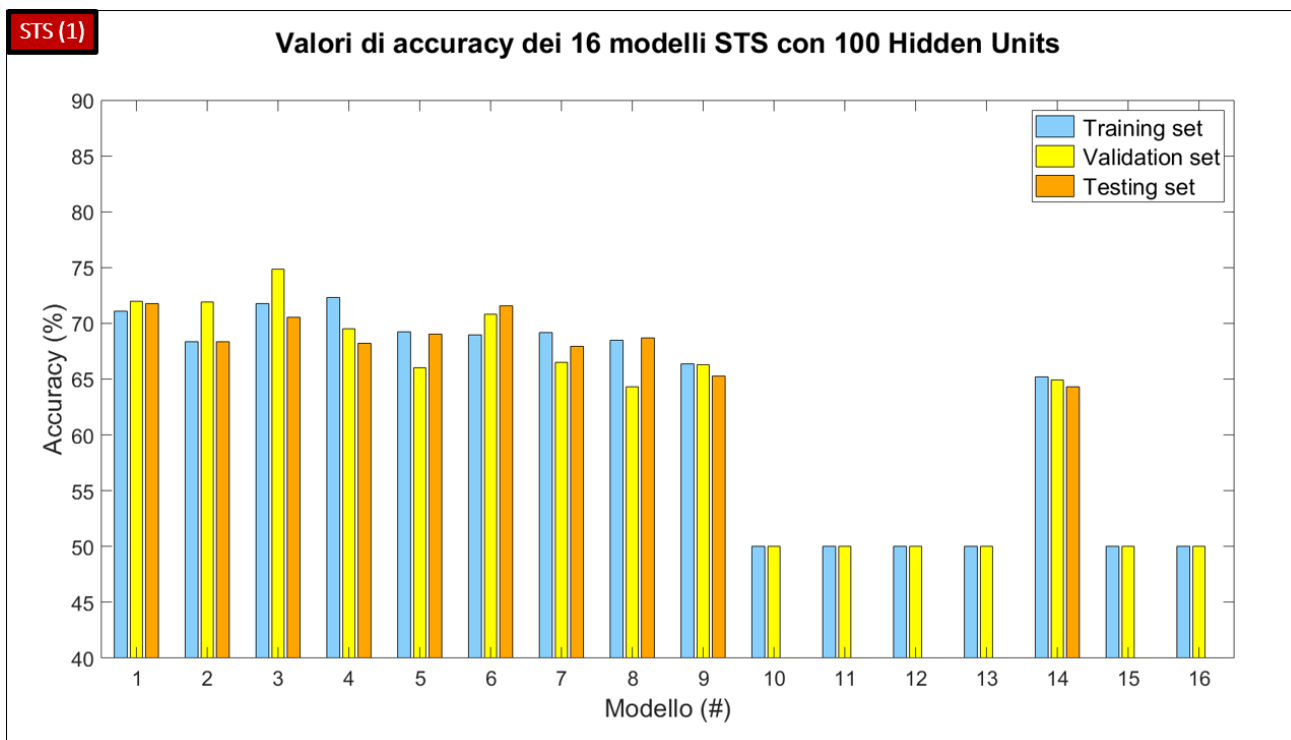
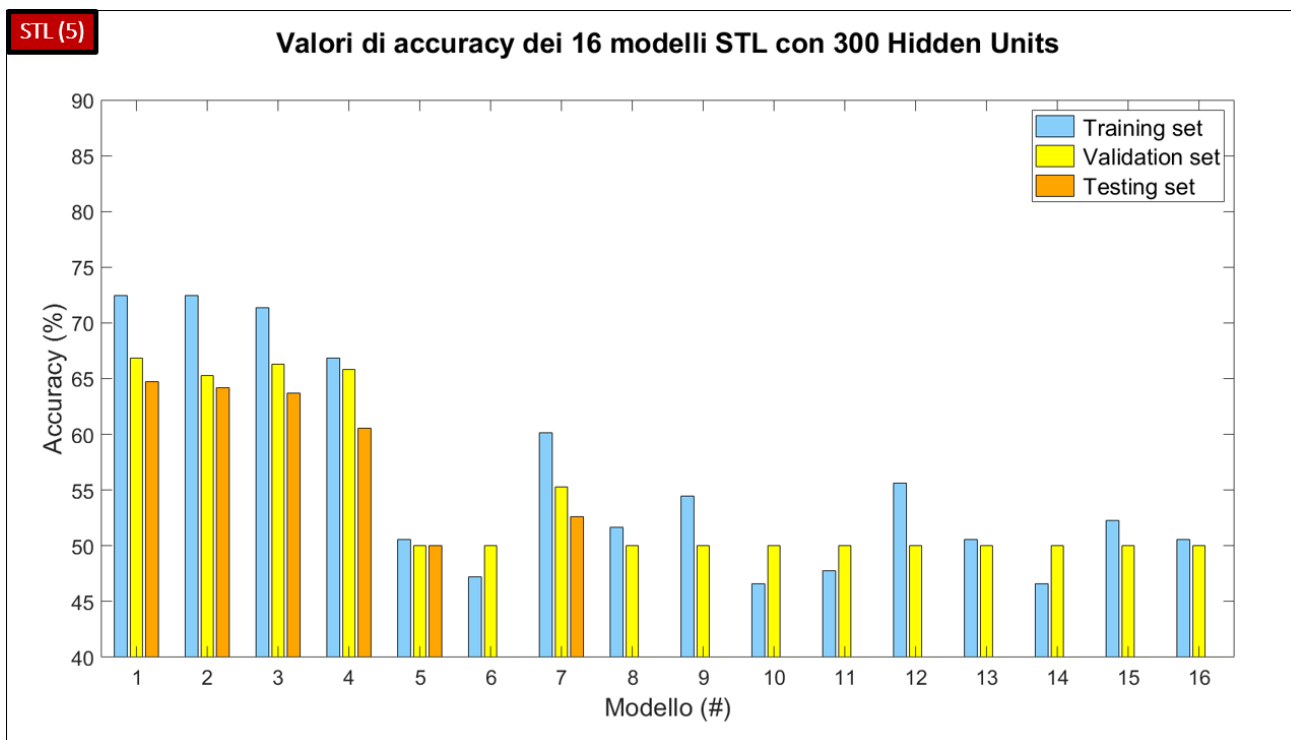
Modello (#)	Initial Learning Rate (α)	Learning Rate Drop Factor (δ)
1	0.005	0.05
2	0.005	0.1
3	0.005	0.15
4	0.005	0.2
5	0.01	0.05
6	0.01	0.1
7	0.01	0.15
8	0.01	0.2
9	0.05	0.05
10	0.05	0.1
11	0.05	0.15
12	0.05	0.2
13	0.1	0.05
14	0.1	0.1
15	0.1	0.15
16	0.1	0.2

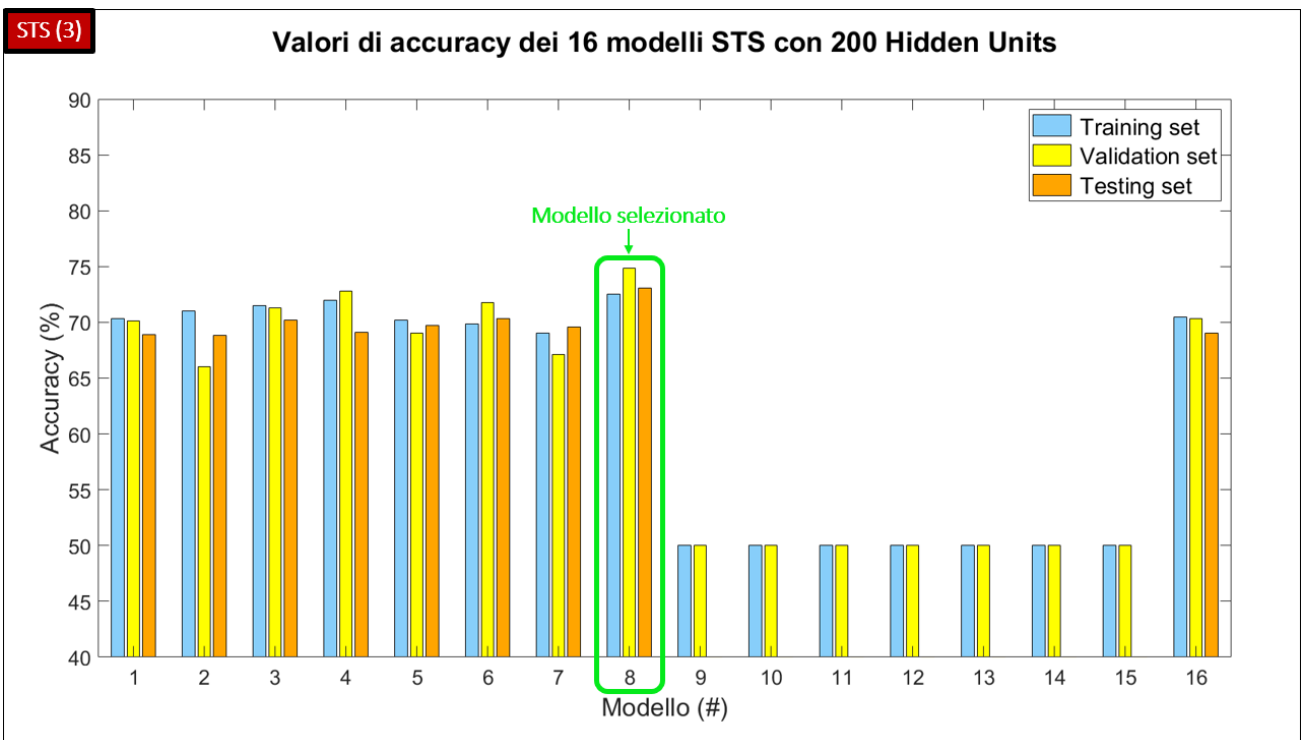
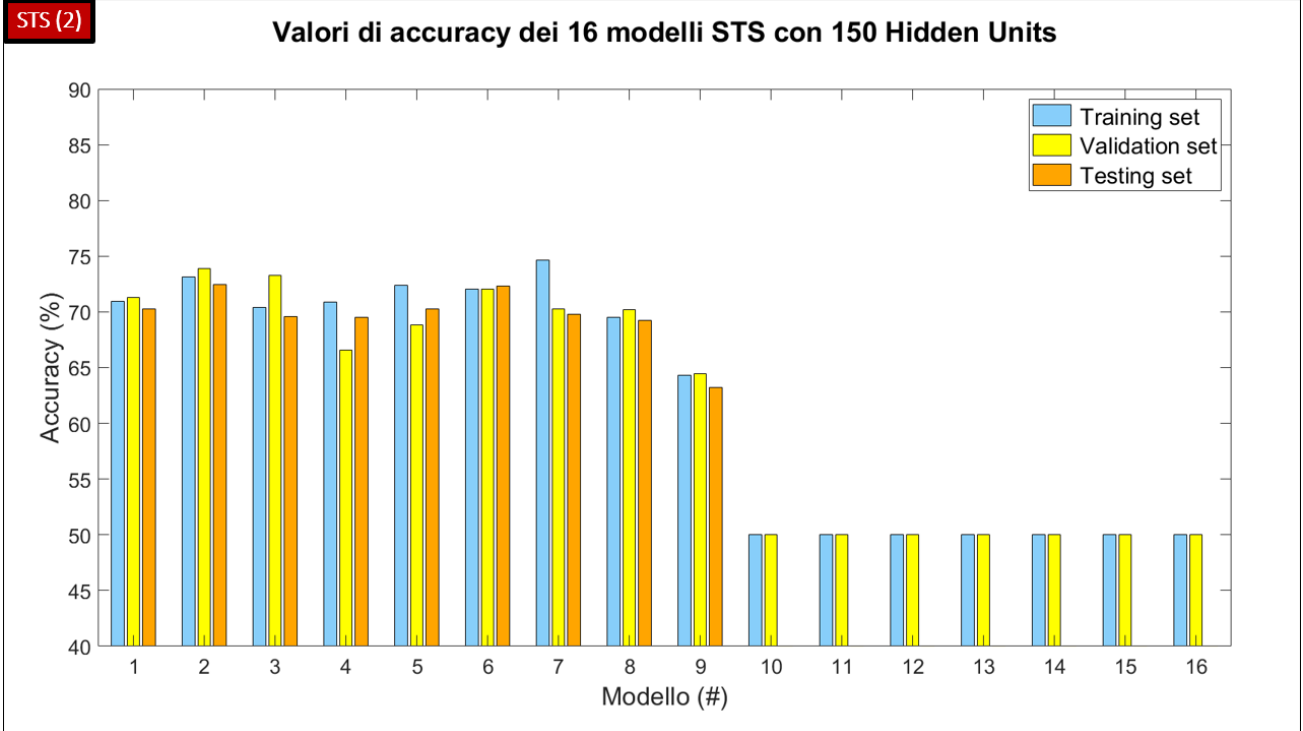
Tabella B.1. Corrispondenze tra numeri, che identificano le combinazioni di α e δ , e combinazioni stesse.

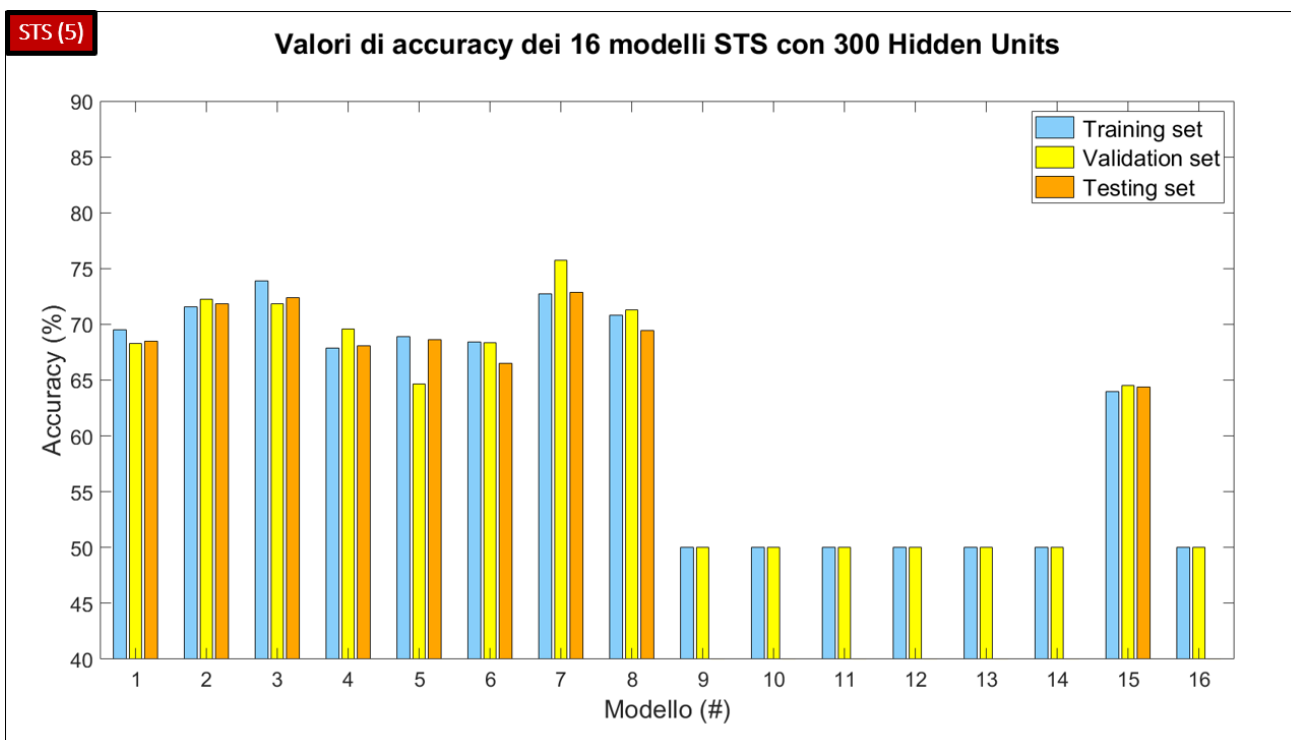
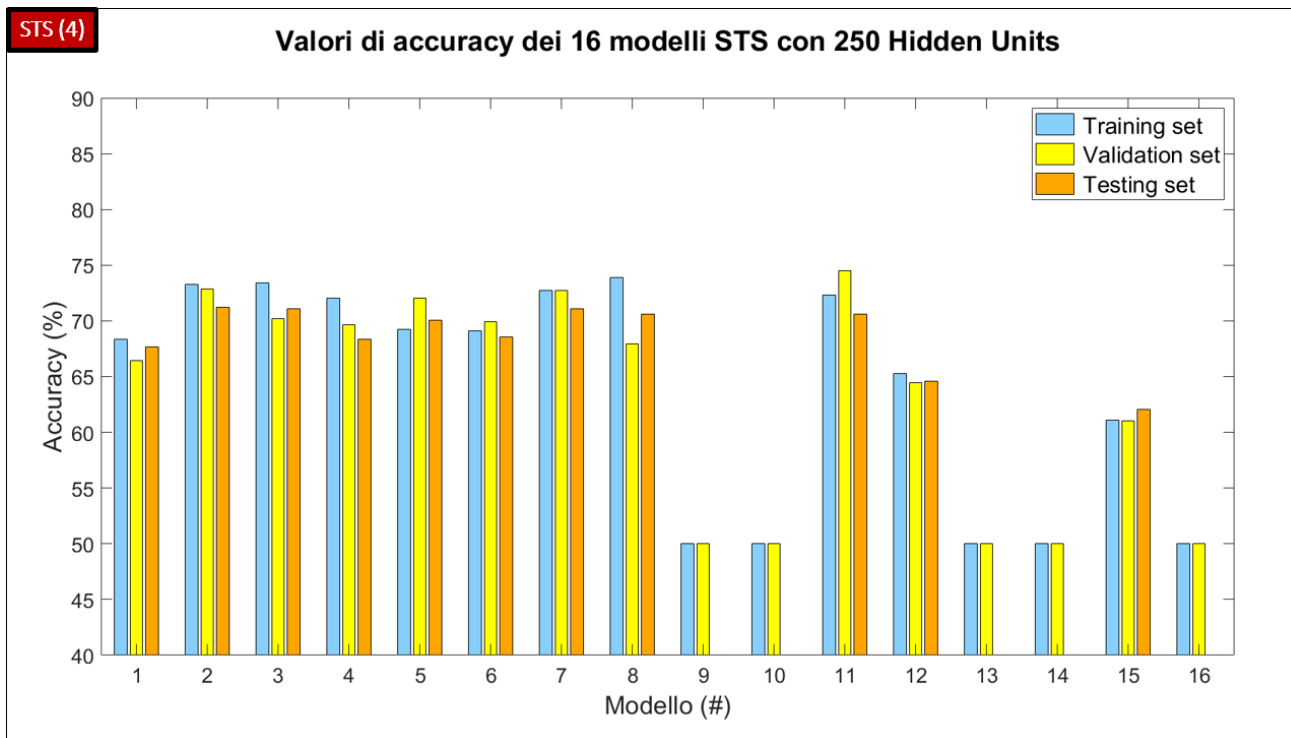
Figura B.1. Valori di training accuracy, validation accuracy e testing accuracy degli 80 modelli BiLSTM in approccio STL e degli 80 modelli BiLSTM in approccio STS. Per singolo approccio, i risultati sono raccolti in 5 grafici. Ciascuno dei 5 grafici è associato ad un numero di Hidden Units e mostra 16 terne di accuracy, una terna per singolo modello di **Tabella B.1.** Sono evidenziate le due terne dei due modelli selezionati, rispettivamente STL e STS.











Sia in approccio STL che in approccio STS, la testing accuracy è pari allo 0% prevalentemente per i modelli dal #9 al #16 (cfr. **Tabella B.1.**), a prescindere dal numero di Hidden Units, come mostrato in **Tabella B.2.** Tali modelli con testing accuracy nulla presentano validation accuracy pari al 50% (e training accuracy variabile). Inoltre, la corrispettiva validation loss, eccetto che per l'iterazione #1, possiede valore "NaN".

Modello (#)	Numero di Hidden Units				
	100	150	200	250	300
5	/	/	/	STL	/
6	/	/	/	STL	STL
7	/	/	/	/	/
8	/	/	/	/	STL
9	STL	STL	STL - STS	STL - STS	STL - STS
10	STS	STL - STS	STL - STS	STL - STS	STL - STS
11	STL - STS	STL - STS	STL - STS	/	STL - STS
12	STL - STS	STL - STS	STL - STS	STL	STL - STS
13	STL - STS	STL - STS	STL - STS	STL - STS	STL - STS
14	STL	STS	STL - STS	STL - STS	STL - STS
15	STL - STS	STL - STS	STL - STS	STL	STL
16	STL - STS	STL - STS	STL	STL - STS	STL - STS

Tabella B.2. Indicazione dei modelli con testing accuracy nulla. I modelli da #1 a #4 ed i modelli sbarrati (/) presentano testing accuracy non nulla. Gli acronimi STL e STS indicano, rispettivamente, se si tratta di un modello STL e/o STS. I due colori (verde per STL, rosso per STS) sono adottati esclusivamente ai fini di una migliore visualizzazione dei dati riportati.

Complessivamente, sia in approccio STL che in approccio STS, gli 80 allenamenti degli 80 modelli sono tali per cui, con mini-batch size pari al 20% del training set, ciascuna

epoca include 5 iterazioni. In totale, 300 iterazioni per le 60 epoche (numero massimo). Con validation frequency pari a 50, la validazione si verifica in coincidenza delle seguenti iterazioni: #1, #50, #100, #150, #200, #250, #300. Con validation patience pari a 5, a seguito di 5 validazioni con validation loss maggiore o uguale al valore minimo precedente, l'allenamento termina (anche senza raggiungere il numero massimo di epoche). In particolare, i modelli con testing accuracy nulla (cfr. **Tabella B.2.**) presentano validation loss pari a "NaN" per 5 validazioni (iterazioni: #50, #100, #150, #200, #250) e, di conseguenza, il corrispettivo allenamento termina all'iterazione #250 (non raggiunge il numero massimo di epoche, si conclude all'epoca #50). Verosimilmente, il valore nullo della testing accuracy, per tali modelli, è riconducibile alla validation loss non definita. Sostanzialmente, la funzione obiettivo non viene "adeguatamente" calcolata sul validation set (rimanendo non definita) e la rete neurale risulta incapace nel generalizzare.

B.2 Variazioni dell'architettura

In Figura B.2., sono rappresentate le 13 architetture di rete neurale BiLSTM delle 13 prove di variazione effettuate.

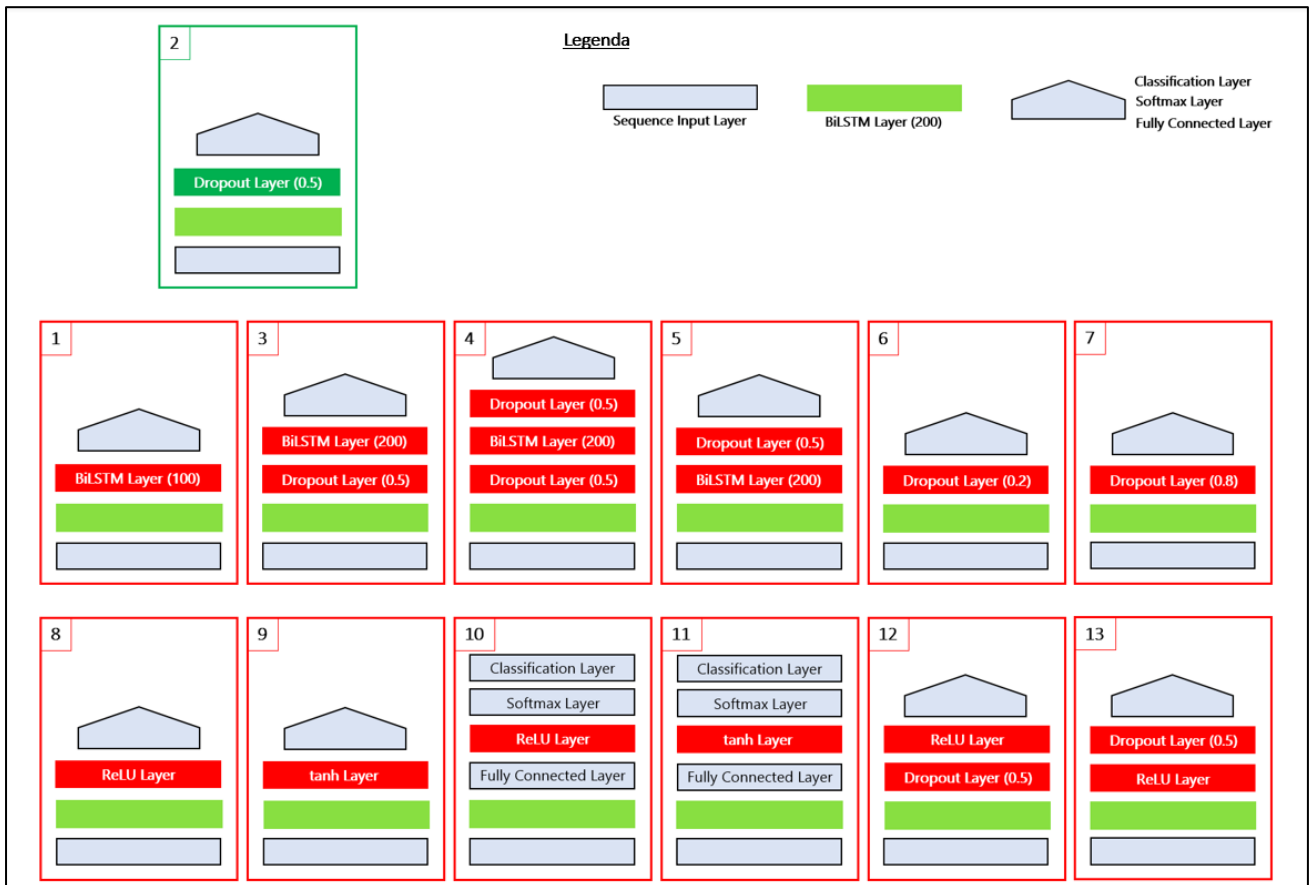


Figura B.2. Architetture delle 13 prove di variazione (un'architettura per singola prova). L'architettura della prova 2, per la quale lo scostamento tra training accuracy e validation accuracy non supera la soglia del 5% e, a differenza della prova 10, la testing accuracy non è nulla, è illustrata entro il riquadro verde. Le altre 12 sono illustrate entro i riquadri rossi.

Bibliografia e Sitografia

- [1] C. Morbidoni, A. Cucchiarelli, S. Fioretti and F. Di Nardo, "A Deep Learning Approach to EMG-Based Classification of Gait Phases during Level Ground Walking," *Electronics*, vol. 8, p. 894, 2019.
- [2] V. Agostini, D. Ganio, K. Facchin, L. Cane, S. M. Carneiro and M. Knaflitz, "Gait parameters and muscle activation patterns at 3, 6 and 12 months after Total Hip Arthroplasty," *The Journal of Arthroplasty*, vol. 29, no. 6, pp. 1265-1272, 2014.
- [3] V. Agostini, F. Lo Fermo, G. Massazza and M. Knaflitz, "Does texting while walking really affect gait in young adults?," *Journal of Neuroengineering and Rehabilitation*, vol. 12, no. 86, 2015.
- [4] V. Agostini, A. Nascimbeni, A. Gaffuri, P. Imazio, M. G. Benedetti and M. Knaflitz, "Normative EMG activation patterns of school-age children during gait," *Gait Posture*, vol. 32, no. 3, pp. 285-289, Jul. 2010.
- [5] V. Agostini, A. Nascimbeni, A. Gaffuri and M. Knaflitz, "Multiple gait patterns within the same Winters class in children with hemiplegic cerebral palsy," *Clinical Biomechanics*, vol. 30, no. 9, pp. 908-914, Nov. 2015.
- [6] F. Di Nardo, A. Mengarelli, A. Strazza, V. Agostini, M. Knaflitz, L. Burattini and S. Fioretti, "A new parameter for quantifying the variability of surface electromyographic signals during gait: The occurrence frequency," *Journal of Electromyography and Kinesiology*, vol. 36, pp. 25-33, 2017.
- [7] M. Ghislieri, V. Agostini and M. Knaflitz, "How to Improve Robustness in Muscle Synergy Extraction," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2019.
- [8] M. Ghislieri, V. Agostini and M. Knaflitz, "Muscle Synergies Extracted Using Principal Activations: Improvement of Robustness and Interpretability," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 2, pp. 453-460, 2020.

- [9] D. Rimini, V. Agostini and M. Knaflitz, "Intra-Subject Consistency during Locomotion: Similarity in Shared and Subject-Specific Muscle Synergies," *Frontiers in Human Neuroscience*, vol. 11, no. 586, 2017.
- [10] M. Bertoli, A. Cereatti, D. Trojaniello, A. Ravaschio and U. Della Croce, "The identification of multiple U-turns in gait: comparison of four trunk IMU-based methods," in *BODYNETS*, Turin, Italy, 2016, December 15-16.
- [11] M. Kobayashi and S. Usuda, "Development of a clinical assessment test of 180-degree standing turn strategy (CAT-STTS) and investigation of its reliability and validity," *Journal of Physical Therapy Science*, vol. 28, pp. 646-653, 2016.
- [12] A. Salarian, C. Zampieri, F. B. Horak, P. Carlson-Kuhta and J. G. Nutt, "Analyzing 180° Turns Using an Inertial System Reveals Early Signs of Progression of Parkinson's Disease," in *31st Annual International Conference of the IEEE EMBS*, Minneapolis, Minnesota, USA, September 2-6, 2009.
- [13] V. Agostini, G. Balestra and M. Knaflitz, "Segmentation and classification of gait cycles," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, pp. 946-952, 2014.
- [14] M. W. Whittle, *Gait Analysis: An Introduction*, Fourth edition ed., Elsevier, 2007.
- [15] L. N. M. Gommans, A. T. Smid, M. R. M. Scheltinga, E. Cancrinus, F. A. M. Brooijmans, K. Meijer and J. A. W. Teijink, "Prolonged stance phase during walking in intermittent claudication," *Journal of Vascular Surgery*, vol. 66, no. 2, pp. 515-522, 2017.
- [16] S. Rosati, V. Agostini, G. Balestra and M. Knaflitz, "Basographic Gait Impairment score: a fuzzy classifier based on foot-floor contact parameters," in *Proceeding of the 9th IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2014.
- [17] A. M. Turcato, M. Godi, A. Giordano, M. Schieppati and A. Nardone, "The generation of centripetal force when walking in a circle: insight from the distribution of ground reaction forces recorded by plantar insoles," *Journal of NeuroEngineering and Rehabilitation*, vol. 12, p. 4, 2015.

- [18] G. Courtine and M. Schieppati, "Tuning of a Basic Coordination Pattern Constructs Straight-Ahead and Curved Walking in Humans," *Journal of Neurophysiology*, vol. 91, pp. 1524-1535, 2004. First published December 10, 2003.
- [19] M. Bertoli, A. Cereatti and U. Della Croce, "Identification of multiple U-turns using gyroscopes: comparative assessment of two methods," in *GNB2016*, Naples, Italy, June 20th-22nd 2016.
- [20] M. Bertoli, A. Cereatti, U. Della Croce, A. Pica and F. Bini, "Can MIMUs positioned on the ankles provide a reliable detection and characterization of U-turns in gait?," in *2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, Rome, Italy, 2018.
- [21] N. H. Ghassemi, J. Hannink, C. F. Martindale, H. Gaßner, M. Müller, J. Klucken and B. M. Eskofier, "Segmentation of Gait Sequences in Sensor-Based Movement Analysis: A Comparison of Methods in Parkinson's Disease," *Sensors*, vol. 18, p. 145, 2018.
- [22] D. J. Geerse, B. H. Coolen and M. Roerdink, "Kinematic Validation of a Multi-Kinect v2 Instrumented 10-Meter Walkway for Quantitative Gait Assessments," *PloS one*, Oct. 2015.
- [23] Y. Akiyama, H. Toda, T. Ogura, S. Okamoto and Y. Yamada, "Classification and analysis of the natural corner curving motion of humans based on gait motion," *Gait Posture*, vol. 60, pp. 15-21, 2018.
- [24] D.-W. Lee, K. Jun, S. Lee, J.-K. Ko and M. S. Kim, "Abnormal Gait Recognition Using 3D Joint information of Multiple Kinects System and RNN-LSTM," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019.
- [25] A. V. Medvedev, G. I. Agoureeva and A. M. Murro, "A Long Short-Term Memory neural network for the detection of epileptiform spikes and high frequency oscillations," *Scientific Reports*, vol. 9, 2019.
- [26] N. Michielli, U. R. Acharya and F. Molinari, "Cascaded LSTM recurrent neural network for automated sleep stage classification using single-channel EEG signals," *Computers in Biology and Medicine*, vol. 106, pp. 71-81, 2019.
- [27] D. Smirnov and E. M. Nguifo, "Time Series Classification with Recurrent Neural Networks," *Advanced Analytics and Learning on Temporal Data*, p. 8, 2018.

- [28] R. Pascanu, T. Mikolov and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA, 2013.
- [29] K. K. Paliwal and M. Schuster, "Bidirectional Recurrent Neural Networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, 1997.
- [30] M. Geng, W. Zhou, G. Liu, C. Li and Y. Zhang, "Epileptic Seizure Detection Based on Stockwell Transform and Bidirectional Long Short-Term Memory," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 3, pp. 573-580, 2020.
- [31] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, pp. 602-610, 2005.
- [32] S. Schmidhuber and J. Hochreiter, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [33] A. Graves, A.-r. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.
- [34] Y. Bengio, P. Simard and P. Frasconi, "Learning Long-Term Dependencies with Gradient Descent is Difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, Mar. 1994.
- [35] A. Graves, N. Beringer and J. Schmidhuber, "A Comparison Between Spiking and Differentiable Recurrent Neural Networks on Spoken Digit Recognition," in *The 23rd IASTED International Conference on modelling, identification, and control*, Grindelwald, 2004.
- [36] F. A. Gers, N. N. Schraudolph and J. Schmidhuber, "Learning Precise Timing with LSTM Recurrent Networks," *Journal of Machine Learning Research*, vol. 3, pp. 115-143, 2002.
- [37] A. Ray, S. Rajeswar and S. Chaudhury, "Text Recognition using Deep BLSTM Networks," in *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR)*, 2015.
- [38] Y. Jiao, Y. Deng, Y. Luo and B.-L. Lu, "Driver sleepiness detection from EEG and EOG signals using GAN and LSTM networks," *Neurocomputing*, 2020.

- [39] D. Thapar, G. Jaswal, A. Nigam and C. Arora, "Gait metric learning siamese network exploiting dual of spatio-temporal 3D-CNN intra and LSTM based inter gait-cycle-segment features," *Pattern Recognition Letters*, vol. 125, pp. 646-653, 2019.
- [40] K. Nkabiti, Y. Chen, K. Sultan and B. Armand, "A deep bidirectional LSTM recurrent neural networks for identifying humans indoors using channel state information," in *2019 28th Wireless and Optical Communications Conference, WOCC 2019*, 2019.
- [41] S. Norgaard, R. Saeedi and A. H. Gebremedhin, "Multi-Sensor Time-Series Classification for Activity Tracking Under Variable Length," *IEEE Sensors Journal*, vol. 20, no. 5, pp. 2701-2709, 2020.
- [42] N. Elsayed, A. S. Maida and M. Bayoumi, "An Analysis of Univariate and Multivariate Electrocardiography Signal Classification," in *2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2019.
- [43] A. S. Alharthi and K. B. Ozanyan, "Deep Learning for Ground Reaction Force Data Analysis: Application to Wide-Area Floor Sensing," in *The 28th International Symposium on Industrial Electronics (ISIE)*, 2019.
- [44] M. Khokhlova, C. Migniot, A. Morozov, O. Sushkova and A. Dipanda, "Normal and pathological gait classification LSTM model," *Artificial Intelligence in Medicine*, vol. 94, pp. 54-66, 2019.
- [45] H. X. Tan, N. N. Aung, J. Tian, M. C. H. Chua and Y. O. Yang, "Time series classification using a modified LSTM approach from accelerometer-based data: A comparative study for gait cycle detection," *Gait & Posture*, vol. 74, pp. 128-134, 2019.
- [46] L. Leistriz, J. Hochreiter, G. F. Volk and F. Bachl, "Classification of facial movements in chronic facial palsy based on intramuscular EMG signals recorded from the paretic side," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Montreal, QC, Canada, 2020.
- [47] D. Huang and B. Chen, "Surface EMG Decoding for Hand Gestures Based on Spectrogram and CNN-LSTM," in *Proceedings of 2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*, Xi'an, China, 21-22 September 2019.

- [48] J. Song, A. Zhu, Y. Tu, H. Huang, M. A. Arif, Z. Shen and X. Z. a. G. Cao, "Effects of Different Feature Parameters of sEMG on Human Motion Pattern Recognition Using Multilayer Perceptrons and LSTM Neural Networks," *Applied Sciences (Switzerland)*, vol. 10, no. 10, 2020.
- [49] X. Ma, Y. Liu, Q. Song and C. Wang, "Continuous Estimation of Knee Joint Angle Based on Surface Electromyography Using a Long Short-Term Memory Neural Network and Time-Advanced Feature," *Sensors (Switzerland)*, vol. 20, no. 17, pp. 1-18, 2020.
- [50] I. Ketykó, F. Kovács and K. Z. Varga, "Domain Adaptation for sEMG-based Gesture Recognition with Recurrent Neural Networks," in *IJCNN 2019. Proceedings of the International Joint Conference on Neural Networks*, Budapest, Hungary, 14-19 July 2019.
- [51] Y. Hu, Y. Wong, W. Wei, Y. Du, M. Kankanhalli and W. Geng, "A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition," *PloS one*, vol. 13, no. 10, 2018.
- [52] L. Zhang, G. Liu, B. Han, Z. Wang and T. Zhang, "sEMG Based Human Motion Intention Recognition," *Journal of Robotics*, vol. 2019, 2019.
- [53] R. A. Zanini, E. L. Colombini and M. C. Ferrari de Castro, "Parkinson's disease EMG signal prediction using Neural Networks," in *Conference Proceedings - 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Bari, Italy, October 6-9, 2019.
- [54] R. Luo, S. Sun, X. Zhang, Z. Tang and W. Wang, "A Low-Cost End-to-End sEMG-Based Gait Sub-Phase Recognition System," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 1, pp. 267-276, Jan. 2020.
- [55] K. Dockx, E. M. J. Bekkers, V. Van den Bergh, P. Ginis, L. Rochester, J. M. Hausdorff, A. Nieuwboer and A. Mirelman, "Virtual reality for rehabilitation in Parkinson's disease," *Cochrane Database of Systematic Reviews*, no. 12, 2016.
- [56] V. Agostini, M. Ghislieri, S. Rosati, G. Balestra and M. Knaflitz, "Surface Electromyography Applied to Gait Analysis: How to Improve Its Impact in Clinics?," *Front. Neurol.*, vol. 11:994, 2020.
- [57] V. Agostini, *Integration of CNS and PNS*, Anno Accademico 2018/2019.

- [58] A. Bonnefoy-Mazure and S. Armand, "NORMAL GAIT," in *Orthopedic Management of Children with Cerebral Palsy*, Nova Science Publishers, Inc., 2015.
- [59] [Online]. Available: <https://glassboxmedicine.com/2019/09/15/best-use-of-train-val-test-splits-with-tips-for-medical-data>.
- [60] M. Hossin and M. Sulaiman, "A REVIEW ON EVALUATION METRICS FOR DATA CLASSIFICATION EVALUATIONS," *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, vol. 5, no. 2, Mar. 2015.
- [61] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [62] V. Agostini, M. Knaflitz, L. Antenucci, G. Lisco, L. Gastaldi and S. Tadano, "Wearable sensors for gait analysis: Comparison between a MIMUs system and a gold standard electromechanical system," in *2015 IEEE International Symposium on Medical Measurements and Applications (MeMeA) Proceedings*, IEEE: Torino, Italy, 2015.

Ringraziamenti

Desidero innanzitutto ringraziare la Prof.ssa Valentina Agostini per l'accoglienza nel progetto di tesi e per ogni prezioso suggerimento davanti agli aspetti più critici. Rivolgo un ringraziamento speciale all'Ing. Marco Ghislieri per la costante disponibilità e per la cura nel seguire il lavoro svolto, impedendo che la comunicazione virtuale del periodo di lockdown potesse rappresentare un ostacolo.

Un immenso ringraziamento ai miei genitori, che sono stati il motore di questi anni, sempre pronti a sostenermi e a smorzare i miei momenti difficili. Non basterebbero le parole a ringraziare Enrico, per la complicità, le cose non dette eppure capite e, soprattutto, per l'ironia, che ne ha cambiate di cose!

Fabiola, Salvo, Angelica, Ciccio, Marco, Vito, come sarebbe stato senza di Voi? Grazie per la leggerezza, mai come superficialità ma sempre come approccio ottimistico ai problemi. Grazie per ogni emozione, grazie per ogni risata. Grazie per tutto quello che mi mancherà. Dal 2020 voleremo così: distanti ma vicini.

A Te, alla Tua pacatezza e alla Tua determinazione. Al valore che, nel mio futuro, non perderai.

