

# Report Progetto Time Series

Marco Salmè, Giuliano Salvati, Martina Cudia, Alessandro Frisenda

2 ottobre 2023

## Sommario

Questo report presenta un progetto sviluppato in risposta alla pubblicazione dell'articolo [BT-PBea22], che ha reso disponibile un dataset contenente dati ECG e segnali da un accelerometro triassiale di 80 pazienti anziani fragili sottoposti a riabilitazione cardiaca post-operatoria. I dati sono stati raccolti durante test sotto sforzo utilizzando il cardiofrequenzimetro toracico Polar H10. L'obiettivo principale di questo progetto è sviluppare un modello di deep learning per effettuare previsioni sui tre segnali (asse x, y e z) registrati dall'accelerometro e identificare l'orizzonte di previsione ottimale per prevenire potenziali cadute dei pazienti anziani. Questo approccio innovativo nel campo della fragilità degli anziani mira a prevenire le conseguenze dannose delle cadute attraverso previsioni tempestive. Il report presenta dettagli sul dataset, la metodologia utilizzata per addestrare e ottimizzare i modelli di deep learning e i risultati empirici ottenuti. Questa ricerca rappresenta un importante passo avanti nella prevenzione delle cadute tra gli anziani fragili, aprendo la strada a ulteriori sviluppi futuri.

## 1 Introduzione

Lo sviluppo del progetto ha avuto luogo dall'interesse suscitato dalla pubblicazione dell'articolo [BTPBea22], attraverso cui è stato reso pubblico un dataset, contenente ECG e segnali rilevati da un accelerometro triassiale, appartenenti ad 80 pazienti anziani (over 65) fragili, inseriti in un programma di riabilitazione cardiaca dopo un intervento a cuore aperto. Nello specifico i dati sono stati raccolti utilizzando il cardiofrequenzimetro toracico Polar H10, indossato dai pazienti mentre venivano invitati ad eseguire una serie di test sotto sforzo per la valutazione del loro stato di salute. Dal punto di vista tecnico, la frequenza di campionamento dell'accelerometro è pari a 200 Hz, mentre quella dell'ECG è di 130 Hz. Gli esercizi svolti durante la registrazione sono stati i seguenti: Cicloergometria, Gait Analysis, TUG Test, Stair climbing e 6MWT. In aggiunta ai dati relativi alle registrazioni dei singoli esercizi, gli autori dell'articolo hanno fornito per ciascun paziente informazioni cliniche e demografiche che sono state catalogate all'interno di un file csv. Ai fini dell'implementazione del progetto, abbiamo dovuto però tener conto del fatto che la totalità del campione di pazienti non è stata sottoposta a tutti gli esercizi sopraelencati. L'obiettivo del nostro lavoro è stato quello di sviluppare un modello di Deep Learning che potesse effettuare un forecasting per ciascuno dei tre segnali, asse x, y e z rispettivamente, registrati dall'accelerometro e di individuare il prediction horizon ottimale per questo task. Questa applicazione rappresenta un'innovazione nell'ambito della ricerca sulla fragilità degli anziani, dal momento che permette di prevedere, attraverso la previsione fornita, possibili cadute nell'arco temporale considerato ed evitarne, così, le dannose conseguenze. Per fornire una visione chiara della struttura del presente lavoro, la sezione 2 presenterà in dettaglio il dataset utilizzato, comprese le caratteristiche dei dati, la loro raccolta e processamento. Successivamente, nei capitoli seguenti, esploreremo l'architettura dei diversi modelli di Deep Learning adottati, insieme alle metodologie e alle strategie di addestramento impiegate. Il resto del documento è organizzato come segue. La sezione 2 descrive i dati e i metodi utilizzati nello studio. La sezione 3 presenta i risultati empirici. Infine, le conclusioni sono riportate nella sezione 4.

## 2 Metodologia

### 2.1 Obiettivo del team

Durante la prima fase di sviluppo del progetto, è stata effettuata una ricerca in letteratura sugli sviluppi e sulle applicazioni nel dominio di interesse; successivamente, sono stati analizzati i dati a disposizione, al fine di individuare quale fosse il miglior approccio per elaborarli e per definire le specifiche di implementazione del progetto. Dall'overview della letteratura è emersa una vasta produzione di pubblicazioni, sia accademiche che industriali, per lo più incentrate sul riconoscimento delle varie attività che il paziente anziano può svolgere, tra cui gli esercizi sopra citati, mentre, dalla nostra ricerca, risulta carente la diffusione di modelli di Intelligenza Artificiale, deputati alla previsione di valori critici dell'accelerometro, al fine di prevenire una possibile caduta del paziente in esame. Alla luce delle evidenze fornite dall'analisi dei dati e dalla letteratura, in un primo momento, è stato tentato un approccio che tenesse conto, nella previsione, della tipologia di esercizio svolto; l'idea è stata, in seguito, abbandonata sia per l'assenza delle registrazioni di tali esercizi per alcuni pazienti sia per la ristretta applicabilità che un modello simile potesse avere. Infatti, adottando questa tipologia di feature, ne sarebbe derivata la necessità di avere informazioni sugli esercizi svolti dai nuovi pazienti di cui si vuole prevenire la caduta, circoscrivendo l'applicabilità del modello a casi simili. Sulla base di queste valutazioni, si è stabilito come scopo del nostro modello la prevenzione di cadute di pazienti anziani fragili. L'obiettivo preposto ha determinato una serie di scelte progettuali, al fine di individuare la soluzione che fosse, allo stesso tempo, più precisa e generale possibile. In primo luogo, si è deciso di optare per 3 singole regressioni multivariate, così da poter ottenere una maggiore accuratezza sul singolo asse, piuttosto che predire tre dimensioni contemporaneamente. Inoltre, in questo modo, si sono potuti individuare, per ciascuna delle tre regressioni, i modelli e gli iperparametri ottimali. In seconda battuta, è stata effettuata un'analisi per decidere quale tipologia di forecasting implementare e, in particolare, si è dovuto scegliere tra un approccio many-to-many e uno many-to-one. Nei dati che avevamo a disposizione erano presenti delle corrispondenze tra le finestre temporali e gli specifici esercizi che venivano svolti dai pazienti in quegli istanti, però non vi era nessuna informazione riguardo una possibile caduta. Infatti, per prevenire una potenziale caduta a partire da un approccio many-to-many, sarebbe stata necessaria un'etichettatura sulle sequenze temporali che stabilissero un rischio caduta o meno, così da poter svolgere una classificazione a seguito della predizione della sequenza temporale. Per questo motivo abbiamo pensato che per il nostro scopo fosse meglio implementare un task many-to-one, in cui, data una finestra di input, venisse predetto un singolo valore nel futuro con un certo prediction horizon. In questo modo, infatti, il valore ottenuto può essere confrontato con una soglia stabilita a priori tramite studi sperimentali, per valutare se è critico o meno per una caduta. In tal caso, l'ipotesi di una previsione dell'istante temporale successivo (prediction horizon unitario) è stata scartata, perché, pur ottenendo buoni risultati, non era significativa per il nostro scopo che richiede una copertura temporale più ampia. Per questo motivo, è stato fissato un prediction horizon di 50 campioni, equivalenti a 5 secondi di segnale, con shift temporale della finestra pari a 1. Questa scelta si basa su due motivi fondamentali:

1. Salvaguardare la sicurezza dei pazienti, perché così è possibile monitorarli in ogni istante temporale, con un orizzonte di 5 secondi.
2. Superare il limite della previsione di un solo valore, perché con uno shift temporale pari a 1 il risultato è simile a quello di una previsione di un'intera sequenza, in cui però i valori vengono predetti volta per volta.

### 2.2 Esplorazione Dataset

Considerando la natura temporale dei dati a disposizione, sono state applicate le tecniche principali per individuare informazioni rilevanti nei segnali. Abbiamo, quindi, esaminato i risultati ottenuti attraverso l'auto correlation function (ACF) e la partial auto correlation function (PACF) che, rispettivamente, decretavano: la significatività statistica dei 300 lags precedenti con l'istante temporale considerato e l'influenza su quest'ultimo di ciascuno dei precedenti 10 lags. Si tenga presente come tali valori si riferiscano al segnale lungo l'asse x ma risultano essere validi anche per i restanti assi. In aggiunta, si è fatto ricorso al pacchetto STATSMODELS di Python, con l'obiettivo di evidenziare trend, stagionalità e ciclicità considerevoli, ma tale analisi non ha condotto a risultati rilevanti, mentre dalla matrice di

correlazione redatta sui segnali lungo i tre assi , va sottolineata una forte correlazione, pari allo 0.84, tra l'asse x e l'asse z. In seguito, sono state valutate la seguenti possibilità :

- Introduzione dei segnali relativi all'ECG come feature aggiuntiva, ipotesi ignorata per via dell'assenza di influenza di tali segnali su quelli rilevati dall'accelerometro.
- Trasformate di Wavelet dei vari segnali come feature aggiuntive, idea che non ha avuto seguito alla luce degli scarsi risultati ottenuti nella fase preliminare di sviluppo.
- Integrazione, a livello del singolo paziente, delle feature statiche a disposizione. Dal momento che la maggior parte di questw facevano riferimento agli esercizi svolti e, pertanto, non erano disponibili per la totalità dei pazienti, sono state incluse, esclusivamente, feature generiche come età, sesso, peso, altezza e l'EFS score, che sta per Edmonton Frail Scale score ed è un valore che varia tra 0 e 17; tanto più questo score è elevato e maggiore è la fragilità (e quindi il rischio di caduta) del paziente.
- Potenza del segnale come feature aggiuntiva, effettivamente inclusa e definita come segue per l'i-esimo istante temporale.  $f_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$

## 2.3 Pre-Processing

Le acquisizioni a disposizione presentavano segnali discreti, in particolare per quanto riguarda l'accelerometro, i segnali erano caratterizzati da una frequenza di campionamento pari a 200 campioni al secondo. Dato che la singola serie temporale è, pertanto, costituita da un numero elevato di campioni, si è deciso di andare a considerare come unità minima di essa il decimo di secondo, espresso come media dei sottostanti 20 valori. A tale processamento, ha fatto seguito un'analisi nel dominio della frequenza, che ha rilevato come per i vari segnali si avevano ampiezze considerevoli della trasformata per frequenze pressoché comprese tra  $[0, 0.5]$  Hz. Pertanto, si è applicato un filtro di Butterworth di ordine 4 con frequenza di taglio pari a 1Hz, vedi Figura 1. In seguito, è stata effettuata una normalizzazione per ottenere una scala comune evitando di distorcere differenze negli intervalli di valori o di perdere informazioni. Normalizzazione dei segnali rispetto, come da prassi, alla media e alla deviazione standard del training set. Infine, prima di passare all'addestramento dei modelli, abbiamo ragionato su uno split adeguato dei dati che potesse sfruttare al meglio le informazioni presenti nel dataset formato dagli 80 pazienti. Si è optato per una suddivisione hold-out del tipo 70-20-10, in cui il 70% dei pazienti apparteneva al training set, il 20% al validation set e il 10% al test set. Dopo aver effettuato questa suddivisione, vengono generate le finestre temporali da dare in input alla rete a partire dalla serie temporale di ciascun paziente. Per effettuare questa operazione, vengono stabilite la lunghezza delle singole finestre, lo shift temporale per passare alla finestra successiva e il prediction horizon. Una volta che tali sequenze risultano essere generate, queste vengono concatenate con quelle dei pazienti appartenenti al medesimo set. Questa tipologia di configurazione permette di non avere all'interno della stessa finestra temporale segnali provenienti da pazienti diversi.

## 3 Modelli

### 3.1 Overview modelli

Nel seguito, vengono presentati le tipologie dei modelli sperimentati all'interno del progetto con la rispettiva architettura implementata. Il Benchmark di riferimento utilizzato è rappresentato da una Mean Baseline ovvero un modello che fornisce come previsione la semplice media dei valori all'interno della finestra di input.

1. **Multistep Dense:** Il modello "Multistep Dense" è basato su reti neurali artificiali composte da strati densi. In termini matematici, una rete densa è costituita da una serie di neuroni collegati tra loro con pesi sinaptici. Il modello apprende una funzione non lineare che mappa i dati di input, spesso rappresentati come vettori, a una serie di previsioni future. La previsione a più passi coinvolge la progettazione di un output layer che può generare una sequenza di uscite temporali.

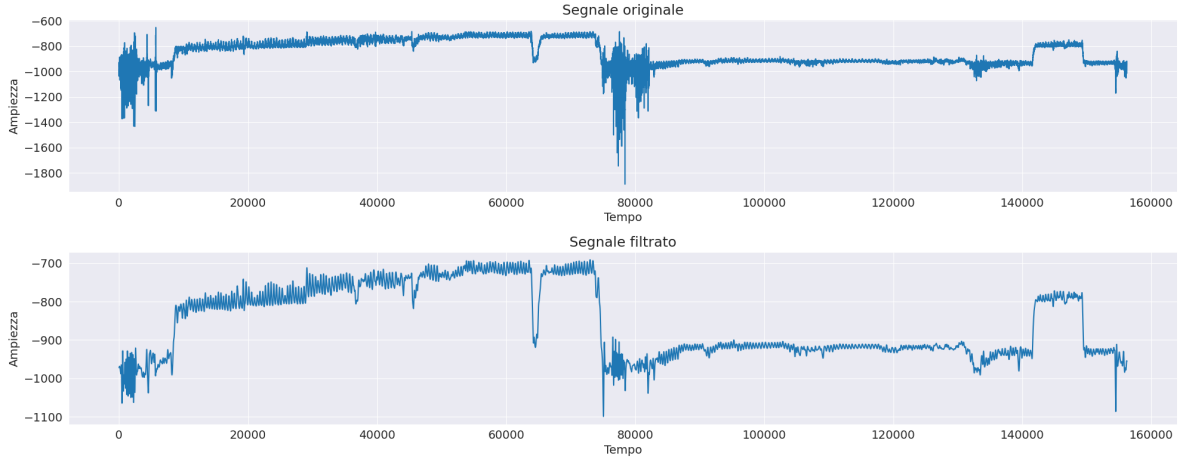


Figura 1: Original sig. vs Filtered sig. Patient 1 .

Layers	Type
Layer1	Dense
Layer2	Dense

Tabella 1: Multi Step Dense Architecture

- LSTM (Long Short-Term Memory):** Le reti LSTM sono basate sulla cella LSTM, una unità di memoria che può mantenere informazioni a lungo termine e prevenire il problema della scomparsa del gradiente. In termini matematici, una cella LSTM include porte di input, di output e di forget, ciascuna governata da funzioni di attivazione e pesi sinaptici. Queste equazioni matematiche permettono alla cella LSTM di memorizzare e aggiornare le informazioni in ingresso, creando una memoria a lungo termine della sequenza temporale.

Layers	Type
Layer1	LSTM
Layer2	LSTM
Layer3	Dense
Layer4	Dense (units=1)

Tabella 2: LSTM Architecture

- ConvLSTM (Convolutional LSTM):** Il modello "ConvLSTM" è una combinazione di reti neurali convoluzionali (CNN) e LSTM. In termini matematici, le operazioni di convoluzione sono integrate nelle unità LSTM, consentendo al modello di acquisire informazioni spaziali tramite operazioni di filtraggio convoluzionale. Ciò viene eseguito in parallelo con le operazioni LSTM tradizionali, che gestiscono le dipendenze temporali.
- GRU (Gated Recurrent Unit):** Le reti GRU sono basate sulla cella GRU, una variante semplificata delle LSTM. In termini matematici, una cella GRU include porte di reset e di aggiornamento che regolano il flusso delle informazioni attraverso la cella. Le equazioni matematiche delle celle GRU sono più leggere rispetto a quelle delle LSTM, il che le rende più efficienti dal punto di vista computazionale.
- Simple RNN (Recurrent Neural Network):** Il modello "Simple RNN" è una rete neurale ricorrente di base. In termini matematici, le unità RNN sono caratterizzate da una funzione di attivazione che consente loro di mantenere uno stato interno. Tuttavia, le equazioni matematiche delle RNN standard possono soffrire del problema della scomparsa del gradiente in sequenze lunghe, poiché l'aggiornamento dello stato interno è semplice e potrebbe causare perdita di informazioni.

Layers	Type
Layer1	CONV1D (kernelsize=2)
Layer2	CONV1D (kernelsize=2)
Layer3	LSTM
Layer4	Dense (units=1)

Tabella 3: CONV-LSTM Architecture

Layers	Type
Layer1	GRU
Layer2	GRU
Layer3	Dense
Layer4	Dense
Layer5	Dense (units=1)

Tabella 4: GRU Architecture

### 3.2 Iperparametri

Nel corso di questo progetto, l'ottimizzazione dei modelli preposti ha rivestito un ruolo cruciale al fine di ottenere prestazioni ottimali. Gli iperparametri, che definiscono la struttura e il comportamento di un modello, sono elementi fondamentali per raggiungere tale obiettivo. Tuttavia, trovare la combinazione ottimale di iperparametri può essere un compito complesso e intensivo in termini computazionali.

Per affrontare questa sfida, abbiamo fatto uso della tecnica conosciuta come Grid Search. Questo approccio sistematico ci ha permesso di esplorare una gamma diversificata di possibili configurazioni degli iperparametri. L'obiettivo principale era identificare le combinazioni che massimizzano le prestazioni dei nostri modelli. Sebbene, come indicato nella Tabella 6, alcuni di questi siano stati fissati in partenza, nella Tabella 7 sono state riportate le varie combinazioni che si sono esplorate. Infine nelle tabelle 8, 9, 10 sono riportati le combinazioni finali, ottenuti per ciascuno dei modelli, rispettivamente, per l'asse x, y,z, su 5 epoche.

### 3.3 Risultati

Una volta ottenuti le combinazioni migliori, ciascuno dei modelli è stato addestrato su 10 epoche, con il Mean Absolute Error come loss. Questa scelta è stata dettata dal fatto che i segnali a disposizione presentavano valori, sia positivi che negativi, prossimi allo zero dopo la normalizzazione. Pertanto, con valori di questo tipo, il MAE non risulta avere problemi e ne è così giustificato il suo utilizzo. Il tempo di addestramento di ciascun modello, per ciascun asse, utilizzando la GPU 'T4' di Google Colab, è stato, in media, di circa 1h sulle 10 epoche. I risultati finali possono essere visionati nelle Tabelle 11, 12, 13,. Mentre con la GRU si ottengono previsioni più accurate sugli assi y e z, la LSTM risulta essere la migliore per l'asse x. Inoltre, è da evidenziare la differenza, in termini di performance, tra le due precedenti e reti più classiche come la Simple-RNN e la Multi Dense Step. Mentre, è doveroso riflettere sulla simile performance tra LSTM/GRU e la Mean Baseline per l'asse x, in particolare sul trade-off tra complessità e performance soprattutto per l'asse x. Al contrario di quanto inizialmente ipotizzato, la CONV-LSTM non ha fornito risultati in linea con le reti migliori.

## 4 Conclusioni

L'obiettivo del lavoro era quello di implementare una rete neurale in grado di effettuare il forecasting di segnali provenienti da un accelerometro triassiale, così da utilizzare quest'ultimo per la previsione di potenziali future cadute in pazienti anziani fragili. In seguito ad un'attenta esplorazione del dataset a disposizione, si è effettuata un'accurata analisi su quelli che potessero essere i modelli migliori per ciascuno dei tre assi. Dalla fase di addestramento, a cui ha fatto seguito quella di test, sono state registrate le migliori performance sull'asse x per la LSTM, su y e z per la GRU. Questo progetto rappresenta un'interessante base di partenza, a cui può seguire l'integrazione di un modello in cui si espliciti la relazione tra valore del segnale dell'accelerometro e caduta del paziente, così da implementare

Layers	Type
Layer1	SimpleRNN
Layer2	Dense
Layer3	Dense (units=1)

Tabella 5: Simple RNN Architecture

Fixed Hyperparameters						
temporal average	input dow	win-	shift	label width	batch size	epochs
40	60		50	1	256	10

Tabella 6: Fixed Hyperparameters.

in maniera totale il task preposto. Infine, con una maggiore disponibilità di risorse computazionali, sarebbe possibile:

1. migliorare i modelli presentati, addestrandoli su un numero maggiore di epoche.
2. addestrare più volte gli stessi modelli e presentare la media delle performance.
3. sperimentare ulteriori combinazioni in fase di GridSearch.
4. aggiungere un livello di complessità ulteriore a tali modelli, ad esempio, introducendo layer.Transformer in grado di effettuare un’analisi di ordine superiore dei segnali stessi.
5. ottenere un modello più robusto optando, al posto del semplice hold-out, per una k-fold cross validation.

## Riferimenti bibliografici

- [BTPBea22] M. Butkuvienė, E. Tamuleviciute-Prasciene, A. Beigiene, and et al. Wearable-based assessment of frailty trajectories during cardiac rehabilitation after open-heart surgery. *IEEE Journal of Biomedical and Health Informatics*, 26(9):4426–4435, 2022.

Parameters	Options
N.Layers	8-16-32-64-128-256
Learning Rate	1e-2;1e-3;1e-4
Activation Fun.	'relu'; 'tanh'

Tabella 7: Grid Search Parameters.

Asse X			
Model	N.Layers	Learning Rate	Activation Fun.
Simple RNN	256-32-1	0.0001	'relu'
LSTM	256-8-32-1	0.0001	'relu'
GRU	32-8-8-32-1	0.0001	'relu'
CONV-LSTM	32-8-16-1	0.0001	'tanh'
Multi Dense Step	8-1	0.0001	'relu'

Tabella 8: Risultati GridSearch AsseX.

Asse Y			
Model	N.Layers	Learning Rate	Activation Fun.
Simple RNN	32-32-1	0.001	'relu'
LSTM	256-8-32-1	0.001	'relu'
GRU	16-32-64-16-1	0.001	'relu'
CONV-LSTM	32-4-32-1	0.001	'relu'
Multi Dense Step	8-1	0.001	'relu'

Tabella 9: Risultati GridSearch AsseY.

Asse Z			
Model	N.Layers	Learning Rate	Activation Fun.
Simple RNN	64-16-1	0.0001	'tanh'
LSTM	128-64-16-1	0.0001	'relu'
GRU	64-16-16-32-1	0.0001	'tanh'
CONV-LSTM	32-8-8-1	0.0001	'relu'
Multi Dense Step	8-1	0.0001	'relu'

Tabella 10: Risultati GridSearch AsseZ.

Asse X		
Model	Val <sub>Loss</sub>	Test Performance
Simple RNN	0.13411	0.10365
<b>LSTM</b>	<b>0.09047</b>	<b>0.0765</b>
GRU	0.18600	0.08550
CONV-LSTM	0.12209	0.10992
Multi Dense Step	0.27953	0.28824
Mean Baseline	—	0.08980

Tabella 11: Risultati Forecasting AsseX.

Asse Y		
Model	Val <sub>Loss</sub>	Test Performance
Simple RNN	0.27482	0.27659
LSTM	0.09110	0.12167
<b>GRU</b>	<b>0.08512</b>	<b>0.08953</b>
CONV-LSTM	0.19503	0.22678
Multi Dense Step	0.21514	0.22459
Mean Baseline	—	0.22015

Tabella 12: Risultati Forecasting AsseY.

Asse Z		
Model	Val <sub>Loss</sub>	Test Performance
Simple RNN	0.20884	0.17912
LSTM	0.14199	0.12167
<b>GRU</b>	<b>0.071632</b>	<b>0.07132</b>
CONV-LSTM	0.16856	0.13820
Multi Dense Step	0.21514	0.22459
Mean Baseline	—	0.17405

Tabella 13: Risultati Forecasting AsseZ.