PROGETTO DEEP LEARNING

Gianfranco Sapia
Matricola 223954

Andrea De Seta
Matricola 227755

Anno accademico 2021/22

CONTENTS

1. Business Understanding

1.1. Background

L'azienda Healthware, situata sul suolo italiano, ha raccolto dati riguardanti 40 pazienti e il loro stato di salute (battito cardiaco, stress, SPO2, qualità del sonno, ecc.) tramite dei dispositivi indossabili, come per esempio gli smartwatch. Questi 40 pazienti si suddividono in:

- 24 affetti da Parkinson
- 3 affetti da Parkinson in stato avanzato
- 13 senza essere affetti da Parkinson

Il dato più importante che andremo ad analizzare è quello riguardante ai **periodi OFF**, cioè tutti quei momenti in cui la cura per il Parkinson chiamata Levodopa non sta funzionando correttamente; questi momenti OFF possono presentare sia sintomi di natura motoria, come per esempio tremore e rigidità, e sia sintomi di natura non motoria, per esempio stati d'ansia.
Nel caso che si andrà ad analizzare, il focus sarà sui sintomi di natura motoria, precisamente sui tremori.

1.2. Business Reason

É importante identificare i sintomi OFF percepiti dal paziente analizzato in maniera accurata, tramite l'utilizzo dei dati raccolti affiché si possa migliorare la quantità di farmaco che il paziente dovrà assumere per cercare di evitare in futuro il verificarsi di questi eventi.

1.3. Business Goal

L'obiettivo è quello di identificare eventi anomali relativi ai sintomi OFF, in questo caso i tremori, analizzando anche quale siano le variabili più importanti al fine di risolvere il problema.

1.4. Business Success Criteria

Riuscire a predire quali possono essere dei futuri valori in una serie di dati e verificare la presenza di anomalie nelle misure fatte ad un paziente.

1.5. Inventory of resources

Al fine di risolvere il problema presentato, sono state utlizzate le seguenti tecnologie:

- Colab
- Jupyter Notebook
- Python
- Pandas
- NumPy
- Tensorflow
- Keras
- Sklearn
- matplotlib

1.6. DEEP LEARNING GOAL

Si trattano di due problemi riguardante le time series.

L'obiettivo del primo task è quello di predirre il prossimo valore in una sequenza di dati, risolvibile con una classificazione di tipo supervised.

Il goal del secondo problema è quello di trovare un modello che permetta di individuare anomalie nelle sequenze di dati di ogni paziente.

2. TASK 1 - NEXT VALUE PREDICTION

2.1. DATA UNDERSTANDING

I data relativi al primo task sono stati forniti tramite un dataset prontamente diviso in "*train.csv*" e "*test.csv*". Ogni riga di entrambi i dataset rappresentano un valore registrato ogni 10 secondi. Il training set presenta le seguenti caratteristiche:

- 144911 righe;
- 3 colonne;
- 434733 dati in totale.

É da sottolineare che tutti e tre gli attributi sono di tipo numerico, precisamente di tipo **intero**. Inoltre, si può notare dalla seguente tabella come l'intero dataset sia pulito in termini di Data Quality in quanto non si ha nessun dato nullo.

Attributo	**Valori non nulli**	**Tipo di dato**
x	144911 non-null	int64
y	144911 non-null	int64
z	144911 non-null	int64

Table 1: Info Training set

2.2. DATA VISUALIZATION

I dati relativi ad ogni feature del dataset sono stati visualizzati in dei grafici per mostrarne l'andamento lungo il tempo.

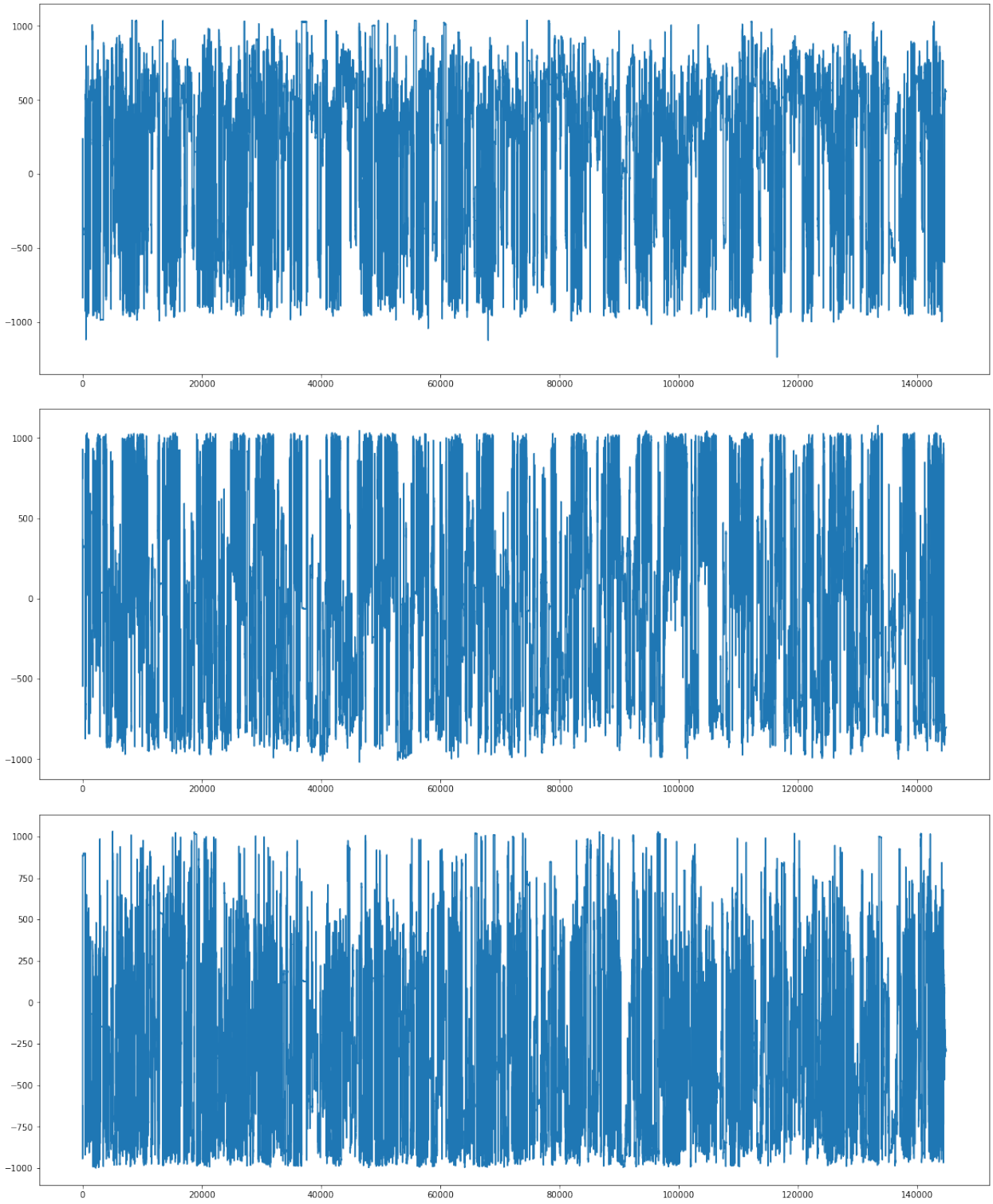

Figure 1: Plot relativi alle Time Series di ogni feature

Data l'analisi preliminare sui dati precedentemente effettuata e dai grafici mostrati si può evincere che il dataset non ha caratteristiche sul quale effettuare operazioni di Data Cleaning.

2.3. DATA PREPARATION

La prima operazione effettuata al fine di preparare il dataset per la sottomissione al modello è stata quella relativa al cambiare il tipo di dati per le tre features. È stata, quindi, effettuata una conversione dal tipo **int64** al tipo **float64**. Come si può evincere dalla seguente tabella, la conversione è stata effettuata con successo:

Attributo	Valori non nulli	Tipo di dato
x	144911 non-null	float64
y	144911 non-null	float64
z	144911 non-null	float64

Table 2: Info conversione di tipo nel Training set

Successivamente, è stata eseguita una normalizzazione dei dati; l'intero Training Set è stato normalizzato secondo la seguente formula:

$$\frac{trainData - mean}{std} \tag{1}$$

dove *mean* e *std* sono rispettivamente la media e la deviazione standard del Training Set secondo una distribuzione normale del dataset avente media 0 e varianza 1.

Come accennato prima, ogni riga del dataset rappresenta una sequenza di dati registrata ogni 10 secondi. È stata quindi creata una finestra (sequenza) di dati come richiesto dalla traccia: per ogni minuto (*Window Shift*) creare una sequenza che rappresenti 5 minuti di dati (*Windows Size*). La funzione che si occuperà della creazione di queste sequenze raccolgono i dati relativi a 30 righe spostandosi all'interno del dataset di 6 righe alla volta. Questo processo è stato effettuato per ogni feature.

Questa operazione di sequenzializzazione dei dati è stata applicata su entrambi i dataset forniti, quindi sia sul Test set e sia sul Training Set. Quest'ultimo è stato diviso per il 70% in training set e per il restante 30% in validation set.

2.4. MODELLING

Il problema da risolvere si tratta di un problema supervisionato. Il modello che verrà costruito sarà un modello di tipo sequenziale che presenta i seguenti layer:

- Un layer **LSTM** (*Long Short Time Memory*), molto utile nel caso delle Time Series in quanto consentono di avere una "traccia del passato"
- Un layer **Dense** con un solo nodo, in quanto interessati nel voler predire un solo elemento successivo ad ogni sequenza.

Il numero di nodi che si è interessati ad applicare al primo layer variano da 16 a 256 con una step-size di 16. Per quanto riguarda il parametro di learning rate anche qui la scelta varia tra 1e-2, 1e-3 ed 1e-4. L'ottimizzatore scelto per questo modello è "Adam". Come *loss function* verrà utilizzata la **Mean Absolute Error (MAE)** al fine di avere un valore medio di errore tra il

valore predetto e quello reale.

Questa scelta di non utilizzare un numero fissato per quanto riguarda il numero di neuroni relativo al layer LSTM e al learning rate non è stata del tutto ”casuale”. Il motivo di ciò è dovuto al fatto che verrà utilizzato un hypertuner parameters per utilizzare, una volta finita la ricerca, i migliori parametri che performano meglio sul modello.

Effettuata questa ricerca sui miglior hyperparameters verrà ricostruito un modello che li applichi. A seguito si riporta il variare della ”loss” su un training set basato su 10 epoche.

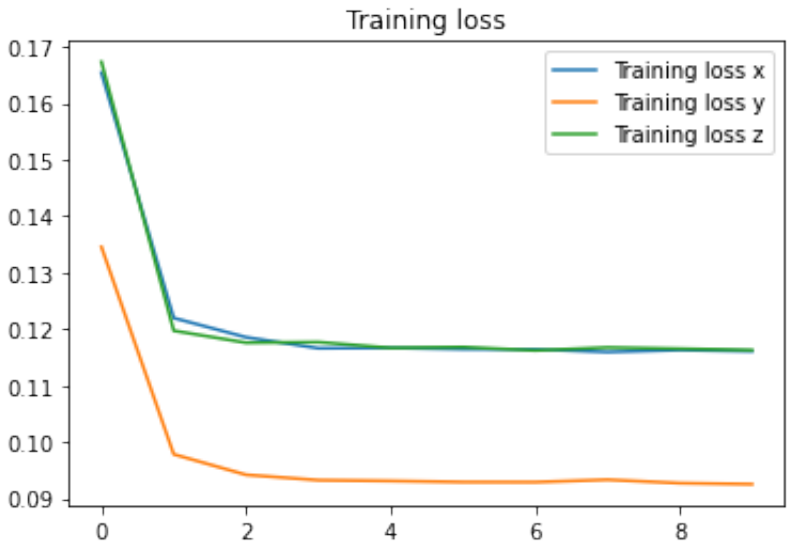

Figure 2: Plot relativo alla funzione di loss del modello creato

L’ultima fase sarà quella di predizione effettuata sulle sequenze di dati del test set per ogni feature. A seguito della predizione si riporta il valore di ”MAE” ottenuto per ogni feature:

Feature	MAE
x	73.58875112590657
y	80.48110151229177
z	74.20807178783062

Table 3: MAE ottenuti

2.5. EVALUATION

In quest’ultima fase si è valutato il comportamento del modello nel caso in cui venissero create delle sequenze con diversa ”*Window Size*” e ”*Windows Shift*”. Considerando la sola feature ”Y” sono state create diverse combinazioni di sequenze che variavano con ”Windows Size” tra 2, 5 e 10 minuti (rispettivamente 12, 30 e 60 righe) e con ”Window Shift” 30 secondi, 1 minuto ed 1 minuto e 30 secondi (uno shift rispettivamente di 3, 6, 9 righe). Per ogni combinazione di Windows Size e Windows Shift considerate viene prima effettuata una ricerca dei miglior

hyperparameters, attraverso un hypertuner parameters, al fine di ricostruire il miglior modello per le sequenze di dati create. A seguito di questa ricerca e delle predizioni effettuate, viene riportato un grafico che mostra l'indice MAE calcolato per ogni combinazione.

Figure 3: I vari MAE calcolati per ogni combinazione messi a confronto

2.6. Conclusion

Il modello si è rilevato molto efficiente in questo tipo di problema supervised. In particolare, la fase di "Evaluation" ha mostrato come diverse combinazioni di "Window Size" e "Window Shift" non hanno influenzato notevolmente i risultati. Infatti la configurazione iniziale può ritenersi soddisfacibile per ulteriori analisi simili.

3. TASK 2 - ANOMALY DETECTION

3.1. DATA UNDERSTANDING

Come per il primo task, anche i dati dei pazienti da analizzare per questo problema sono stati forniti tramite un dataset diviso in due file, precisamente in "*ad_train.csv*" e "*ad_test.csv*".
I dati, in entrambi i file, sono stati ordinati per paziente; il training set presenta dati di persone non affette dalla Sindrome di Parkinson, mentre il test set presenta dati da pazienti affetti da questa sindrome. La seguente tabella descrive caratteristiche e descrizione del training set:

Feature	Description
patient	Identification of patient
x	Accelerometer readings in x
y	Accelerometer readings in y
z	Accelerometer readings in z
heartRate	Heart Rate
timestamp	Timestamp
tsDate	Date

Table 4: Descrizione features Task 2

Il training set presenta:

- 943522 righe;
- 7 colonne;
- 6604654 dati in totale.

Quest'ultimo è composto da righe che registrano ogni secondo i dati relativi alle feature menzionate prima ottenuti da volontari non affetti dalla malattia di Parkinson durante la vita quotidiana. All'interno di questo dataset tutti i dati mancanti che riguardano l'attributo *heartRate* sono registrati con il valore di -1. Di seguito, una tabella riassuntiva dei dati:

Feature	Valori non nulli	Tipo di dato
patient	943522 non-null	int64
x	943522 non-null	int64
y	943522 non-null	int64
z	943522 non-null	int64
heartRate	943522 non-null	int64
timestamp	943522 non-null	int64
tsDate	943522 non-null	object

Table 5: Descrizione features

Il test set è composto da dati ottenuti ogni 10 secondi unicamente da pazienti affetti dalla malattia di Parkinson.

3.2. DATA VISUALIZATION

I valori mancanti riguardano la colonna *heartRate*, pertanto è stata effettuata un'operazione di plotting divisa per ogni paziente presente nel training set e ciò è quanto ne è emerso:

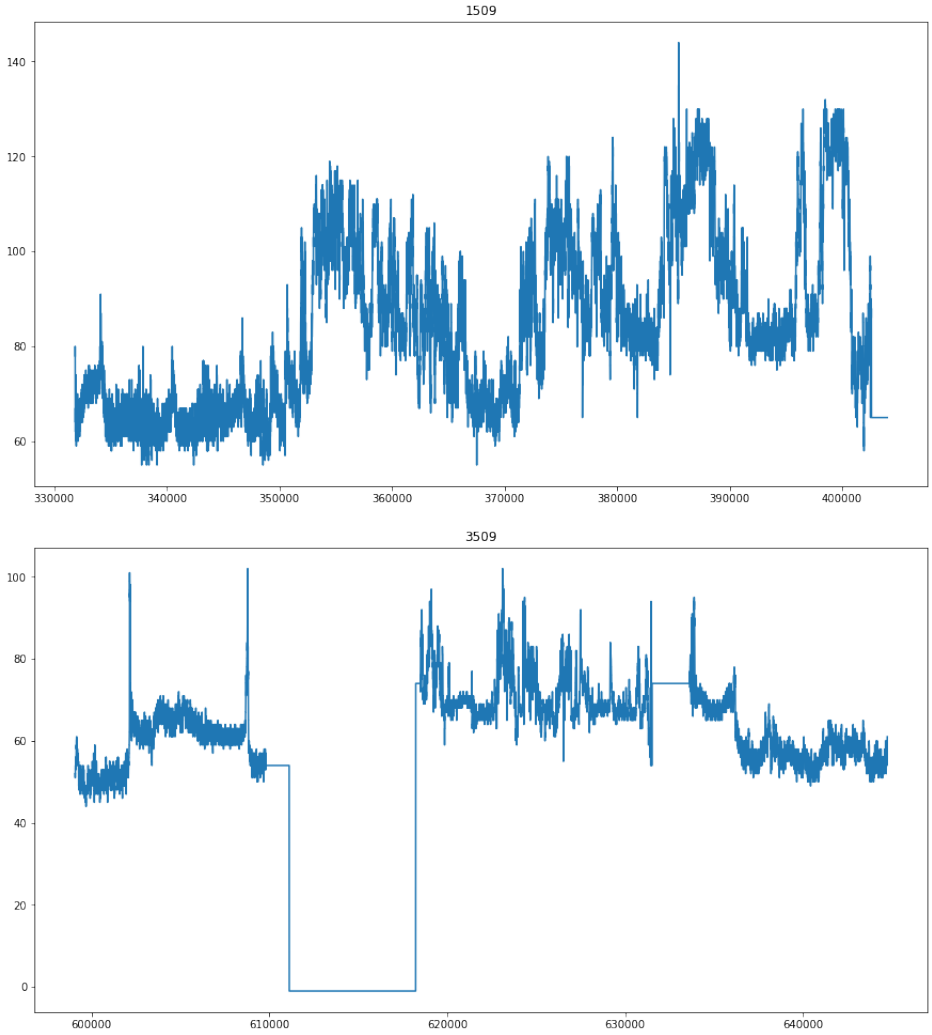

Figure 4: Plot relativi all'attributo *heartRate* di ogni feature

I grafici riportati sono quelli più significativi: uno in cui non sono presenti valori a "-1" e l'altro in cui sono presenti valori a "-1". Dai grafici si può notare che i valori di "-1" presenti in alcune series sulla feature di *heartRate* siano costanti per un periodo continuo di tempo e andando a creare delle series ai lati di questo periodo di inattività.

Sulla base di questa conclusione viene calcolato la lunghezza di questo periodo di tempo per ogni paziente in cui sono assenti i valori. Di seguito è riportato per ogni paziente il momento in cui si è iniziato a registrare valori di "-1", il momento in cui si registra l'ultimo valore "-1" e infine l'intervallo di tempo passato.

```
Initial time :2019-09-10 00:00:00.003000
Last time :2019-09-10 01:59:58.395000
Time passed :1:59:58.392000

Initial time :2019-09-10 14:00:00.007000
Last time :2019-09-10 14:59:58.542000
Time passed :0:59:58.535000

Initial time :2019-09-10 19:00:00.016000
Last time :2019-09-10 20:59:58.105000
Time passed :1:59:58.089000

Initial time :2019-10-30 06:00:00.005000
Last time :2019-10-30 07:59:58.754000
Time passed :1:59:58.749000

Initial time :2019-10-30 15:00:00.012000
Last time :2019-10-30 17:59:58.767000
Time passed :2:59:58.755000

Initial time :2019-10-30 20:00:00.002000
Last time :2019-10-30 20:59:58.327000
Time passed :0:59:58.325000
```

Figure 5: Report tempi in cui il dispositivo ha registrato valori nulli

Da questa analisi si può arguire come l'assenza di questi valori variano per un'intervallo di qualche ora. Quindi, da ciò deduciamo che i valori per questi pazienti sono assenti per questo lasso di tempo in quanto quest'ultimi non indossavano oppure non indossavano correttamente il braccialetto che registra i dati che vengono raccolti.

3.3. Data Preparation

Una prima modifica apportata al dataset al fine di preparare esso alla sottomissione del modello che verrà creato successivamente è quella riguardante la conversione delle colonne interessate da sottoporre al modello dal tipo **int64** al tipo **float64**, come dimostrato nella seguente tabella:

Feature	Valori non nulli	Tipo di dato
patient	943522 non-null	int64
x	943522 non-null	float64
y	943522 non-null	float64
z	943522 non-null	float64
heartRate	943522 non-null	float64
timestamp	943522 non-null	int64
tsDate	943522 non-null	object

Table 6: Descrizione features

Successivamente, come già fatto nel task 1, l'obiettivo è creare sequenze, questa volta però separate per ogni paziente. Nella creazione di quest'ultime va precisato che i pazienti che presentano dati con valori mancanti nella colonna *heartRate*, sono stati a loro volta divisi in due parti come menzionato prima: la serie di dati a sinistra dei valori di "-1" e la serie di dati a destra dei valori di "-1", in modo tale che possano essere considerate come se appartenessero a due paziente diversi. La motivazione di questa scelta è dovuta in merito al fatto che considerare come serie unica, a meno dei valori di "-1", potrebbe portare alla creazione di false sequenze che conterrebbero battiti cardici di momenti diversi della giornata del paziente visto il lasso di tempo trascorso nel momento in cui non vengono registrati battiti.

In seguito, è stato adattato il training set al test set, in quanto ogni riga presente nel training set rappresenta un record ogni secondo mentre ogni riga presente del test set rappresenta un record ogni 10 secondi. Quindi, sui dati di ogni paziente verrà effettuato un'operazione di *rolling* sulla media di 10 valori. Inoltre, vengono eliminati gli attributi **patient**, **timestamp** e **tsDate** in quanto essi non rappresentano dati significativi per l'operazione di training che verrà effettuata in seguito.

Successivamente, è stata effettuta una operazione di normalizzazione sui dati di ogni paziente considerando come *media* e *std* quelle relative ai dati raggruppati. Verrà utilizzata la formula a seguire, dove media e deviazione standard seguono una distribuzione normale avente media 0 e varianza 1:

$$\frac{data_{idPaziente} - mean_{idPaziente}}{std_{idPaziente}} \tag{2}$$

Anche il test set è stato sottoposto all'operazione di normalizzazione. Su quest'ultimo verranno applicati, alla formula sopraccitata, come *media* una media dei valori di media relativa ai dati di ogni paziente e come *std* una media dei valori di std relativa ai dati di ogni paziente.

Dal training set, poi, è stato ricavato il validition set utilizzando le seguenti percentuali: 70% per il training set mentre il restante 30% per il validition set.

Come ultima fase, training set, validation set e test set sono stati sottoposti all'operazione di Data Windowing utilizzando due funzioni diverse: *data_windowing* per training set e validation set e *data_windowing_test* per il test set. Queste due funzioni avranno il seguente comportamento:

- **Training set e Validation Set** - il windowing effettuato avrà come risultato all'interno di una collezione le sequenze raccolte da ogni paziente;
- **Test set** - il windowing effettuato sul test set, invece, avrà come risultato una collezione di sequenze di un determinato paziente.

3.4. MODELLING

Per questa fase, essendo un problema di tipo *unsupervised* è stato utilizzato un autoencoder in quanto è stato ritenuto più adeguato per il tipo di task. Questo tipo di rete neurale fa in modo che il suo input viene copiato in output. Inizialmente i dati passeranno per un **encoder** che verrano poi ricostruiti in una fase successiva da un **decoder**. Il modello che sarà utilizzato è composta dai seguenti layer:

- un layer LSTM avente una forma di input pari a quella delle sequenze;
- un primo layer Dropout avente ratio pari a 0.2;
- un RepeatVector che esegue una replica dell'output del layer; precedente in modo da essere passata al layer successivo
- un layer LSTM con parametro *"return_sequences=True"*;
- un secondo layer Dropout avente ratio pari a 0.2;
- un layer TimeDistributed che permette di applicare un layer ad ogni fetta temporale di un input.

Anche per questo task, come nel primo, è stata effettuata una ricerca dei best hyperparameters attraverso il tuner messo a disposizione dal framework Keras. L'output ottenuto da questa ricerca è il seguente:

```
{'units1': 112,
 'learning_rate': 0.01,
 'tuner/epochs': 15,
 'tuner/initial_epoch': 5,
 'tuner/bracket': 2,
 'tuner/round': 2,
 'tuner/trial_id': '0012'}
```

Figure 6: Output dato dalla ricerca tramite Hyperparameters

Da qui verrà ricostruito un nuovo modello basato su questi hyperparameters che verrà poi nuovamente rieseguito. Un'analisi che è stata possibile eseguire su questo modello è quella della convergenza sulla loss function. Partendo dai best hyperparameters trovati dalla ricerca è stato eseguito un *fit* su numero di epoche pari a 25 ottenendo quanto ne segue:

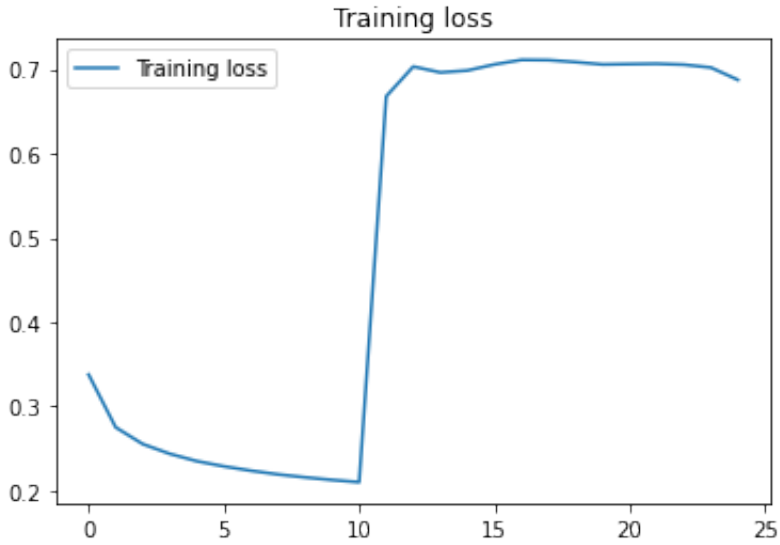

Figure 7: Convergenza della Loss

Dal grafico ottenuto si può osservare come la misura di 'loss' converga dopo 10 epoche. Questa analisi è stata effettuata in un file differente (**"Anomaly Detection Loss Convergence.ipynb"**) in modo da essere ottimizzata coi tempi di esecuzione grazie allo sfruttamento della piattaforma *Google Colab*.

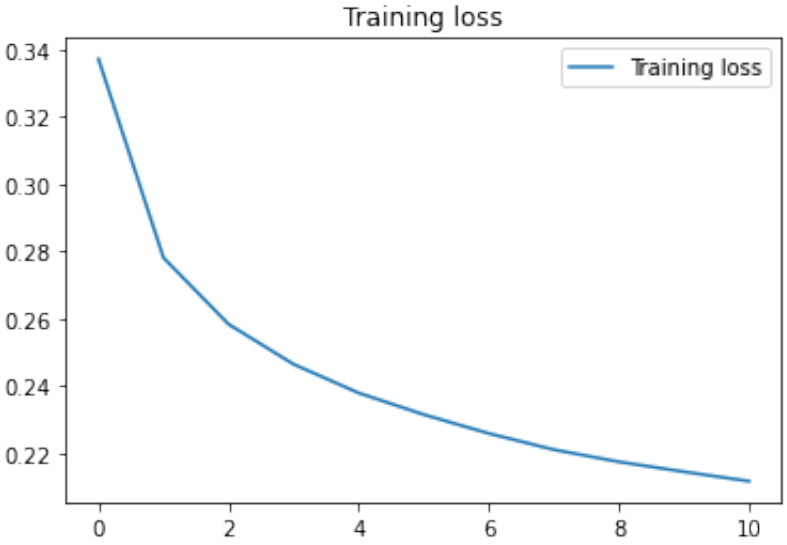

Figure 8: Convergenza della Loss fino a 10 epoche

L'ultima fase da eseguire è quella di ***Prediction***. Partendo dal test set che è stato precedentemente raggruppato per pazienti, sequenzializzato e normalizzato verrà lanciata la funzione di predizione *predict* su ogni sequenza di ogni paziente. Quest'ultima è stata un'operazione abbastanza costosa in termini di tempo. A seguire è riportato il tempo di esecuzione in secondi per le

sequenze di ogni paziente e il tempo totale trascorso.

```
Ended seq 1 in 6986.851551294327
Ended seq 1 in 6826.48596906662
Ended seq 1 in 8695.8202521801
Ended seq 1 in 5076.885046482086
Ended seq 1 in 6183.212693452835
33769.26610541344
```

Figure 9: Tempo di predizione di ogni sequenza per ogni paziente

L'ultima operazione in questa fase è quella del calcolo del MAE (**Mean Absolute Error**) per ogni sequenza, che però presenta un problema. Calcolando il MAE per ogni riga di ogni sequenza risulteranno valori duplicati. Questo è dovuto perché una riga può apparire in più sequenze e, quindi, il MAE di ogni riga potrebbe risultare essere calcolato più volte. Per risolvere questo problema, i valori che vengono calcolati più volte sono stati compressi per somma in modo da avere per ogni riga un valore di MAE unico. Infine, questa somma viene divisa per il numero delle volte che è stata calcolata in modo da ottenerne una media.

3.5. EVALUATION

Una volta ottenuti tutti i valori medi del MAE per ogni riga si può costruire un grafico che ne descrive l'andamento per ogni paziente in modo da verificarne la presenza di eventuali anomalie.

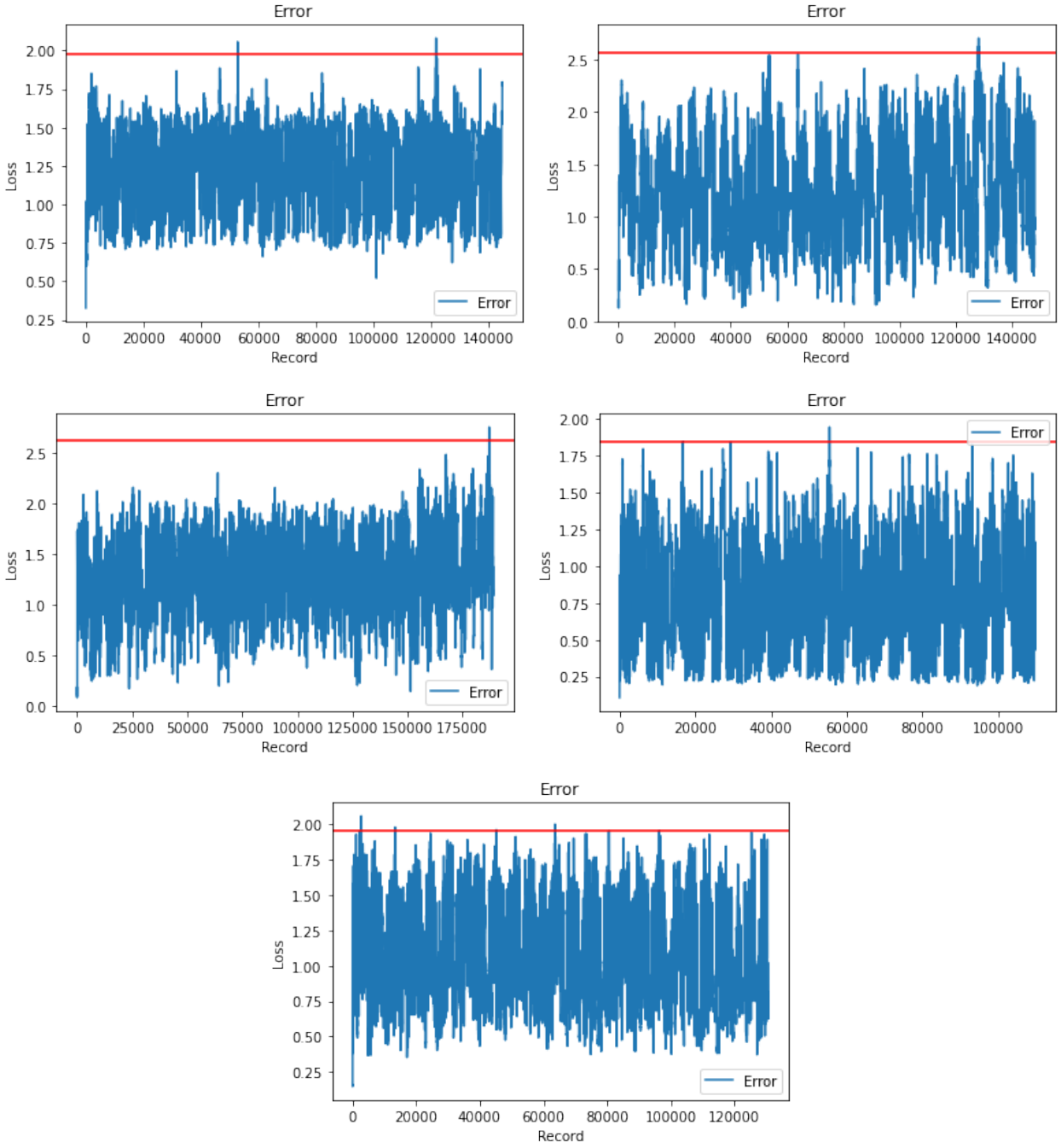

Figure 10: Plot relativi all'attributo *heartRate* di ogni feature

Già da subito si può notare la presenza di anomalie dovuta a picchi molto elevati lungo la serie degli errori calcolati. Sono state ritenute anomalie tutti quei momenti in cui l'errore calcolato oltrepassa la linea rossa indicata sul grafico. Questa soglia è stata calcolata come il 95% rispetto all'errore massimo di ogni serie.

3.6. CONCLUSION

Tutti i periodi considerati come OFF period sono stati salvati all'interno del file ***off_period.csv***. Da tale foglio si può notare come siano stati effettivamente rilevati i momenti con un errore più alto rispetto ad altri considerabili "non anomali". Si può quindi dedurre che il modello abbia funzionato perfettamente affinché potesse trovare queste anomalie ma, purtroppo, i tempi di esecuzione e di calcolo dell'errore sono risultati piuttosto elevati; infatti, soprattutto nella fase di prediction, sono trascorsi all'incirca 9 ore. Ulteriori miglioramenti futuri possono essere un'ottimizzazione sul calcolo del MAE ed un'esecuzione basata sullo sfruttamento della GPU per velocizzare i tempi di esecuzioni in fase di fit e di prediction.

Questo task era concentrato sul riconoscimento di anomalie basato sui sintomi motori, in particolare i tremori; uno studio futuro potrebbe essere basato su un altro tipo di sintomi motori che possono verificarsi nei pazienti affetti da sindrome di Parkinson che assumono il farmaco Levedopa: la rigidità.