UNIVERSITÀ DI PISA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Laurea Triennale in Ingegneria Informatica

Interpretazione dei segnali polisonnografici basata su eXplainable Artificial Intelligence

Relatori:

Prof. Mario G. C. A. Cimino
Ing. Guido Gagliardi

Candidato:

Edoardo Cremente

ANNO ACCADEMICO 2022/2023

Indice

1 Introduzione **1**

2 Materiali e metodi **4**
2.1 Modello e dati 5
2.2 Strumenti e risorse 5

3 Esperimenti **7**
3.1 Stage 1: Importanza delle bande per le classi del sonno 7
3.1.1 Pseudocodice stage 1 8
3.2 Stage 2: Considerazione sull'effetto dell'influenza delle bande sulle altre bande 9
3.2.1 Pseudocodice stage 2 10
3.3 Stage 3: Introduzione del concetto dell'importanza nel tempo 12
3.3.1 Pseudocodice stage 3 13

4 Risultati **16**
4.1 Stage 1 16
4.1.1 Alpha 17
4.1.2 Beta 17
4.1.3 Delta 18
4.1.4 Gamma 19
4.1.5 Sigma 20
4.1.6 Theta 21
4.2 Stage 2 22
4.2.1 Alpha 23
4.2.2 Beta 24
4.2.3 Delta 25
4.2.4 Gamma 26
4.2.5 Sigma 27
4.2.6 Theta 28
4.3 Stage 3 29
4.3.1 Heatmap delle classi 31

5 Conclusione **34**

Elenco delle figure

1.1 Una visualizzazione grafica delle bande che compongono un'onda cerebrale [8] 3

4.1 Importanza dell'onda Alpha rispetto alle classi del sonno predette dalla rete neurale 17
4.2 Importanza dell'onda Beta 18
4.3 Importanza dell'onda Delta 19
4.4 Importanza dell'onda Gamma 20
4.5 Importanza dell'onda Sigma 21
4.6 Importanza dell'onda Theta 22
4.7 Importanza dell'onda Alpha calcolata con la media pesata 24
4.8 Importanza *pesata* dell'onda Beta 25
4.9 Importanza *pesata* dell'onda Delta 26
4.10 Importanza *pesata* dell'onda Gamma 27
4.11 Importanza *pesata* dell'onda Sigma 28
4.12 Importanza *pesata* dell'onda Theta 29
4.13 Importanza nel tempo per la classe Wake con valori normalizzati tra 0 e 1 31
4.14 Importanza nel tempo per la classe NREM1 con valori normalizzati tra 0 e 1 31
4.15 Importanza nel tempo per la classe NREM2 con valori normalizzati tra 0 e 1 32
4.16 Importanza nel tempo per la classe DeepSleep con valori normalizzati tra -1 e 1 32
4.17 Importanza nel tempo per la classe DeepSleep con valori normalizzati tra 0 e 1 33
4.18 Importanza nel tempo per la classe REM con valori normalizzati tra 0 e 1 33

Sommario

L'analisi del sonno riveste un ruolo fondamentale nel campo della salute e del benessere umano. Tradizionalmente, tale analisi si basa sulla registrazione di segnali fisiologici complessi attraverso la polisonnografia (PSG), seguita dalla valutazione manuale di un esperto del sonno. Questo studio propone un approccio innovativo attraverso lo sviluppo di applicazioni basate su explainable artificial intelligence (XAI) per l'interpretazione automatica delle onde cerebrali in relazione alle diverse fasi del sonno.

Lo studio proposto si pone come continuazione di una ricerca nella quale i dati polisonnografici sono stati utilizzati per addestrare una rete neurale profonda. Il focus è quello di interpretare i risultati prodotti dalla rete neurale attraverso metodi di XAI che vedremo come implementare. La funzione implementata consente di calcolare e restituire valori di *importanza* per determinati set di bande, senza l'influenza del tempo e con l'influenza del tempo, evidenziando l'associazione tra onde cerebrali e classi di sonno, seguendo la nomenclatura delle regole dell'American Academy of Sleep Medicine (AASM). L'obiettivo è quello di fornire una comprensione trasparente e interpretabile delle previsioni della rete neurale.

L'approccio XAI ha permesso di esplorare in dettaglio le decisioni del modello, contribuendo a garantire la fiducia degli utenti e a promuovere una maggiore comprensione delle dinamiche del sonno.

In conclusione, questa ricerca delinea una metodologia innovativa per l'analisi del sonno, unendo gli avanzamenti delle reti neurali profonde all'interpretabilità offerta dalle tecniche XAI. Tale approccio potrebbe avere impatti significativi nel migliorare la precisione delle diagnosi per pazienti che soffrono di disturbi del sonno e nell'agevolare la collaborazione tra professionisti della salute e intelligenza artificiale.

Capitolo 1

Introduzione

Il sonno, una componente essenziale della vita umana, riveste un ruolo fondamentale nella promozione della salute fisica e mentale. Il suo studio approfondito non solo svela i misteri dell'incoscienza notturna, ma apre la strada a una comprensione più profonda dei meccanismi biologici e delle dinamiche fisiologiche che caratterizzano il nostro corpo durante il riposo. Nell'era dell'intelligenza artificiale (IA), la convergenza tra analisi del sonno e explainable artificial intelligence (XAI) offre un'opportunità unica per svelare e interpretare le complesse interazioni tra le onde cerebrali e le diverse fasi del sonno.

Questa tesi si propone di esplorare il territorio inesplorato della relazione tra le onde cerebrali e le fasi del sonno, abbracciando il potenziale rivoluzionario dell'XAI nel rendere trasparenti e interpretabili tali associazioni. Attraverso lo sviluppo di applicazioni dedicate, ci si propone di superare le sfide legate alla comprensibilità delle previsioni dei modelli di intelligenza artificiale, aprendo la strada a un nuovo paradigma nell'analisi del sonno.

Attraverso l'integrazione di tecniche avanzate di analisi delle onde cerebrali e l'applicazione di modelli di machine learning, questa ricerca si propone di gettare luce su territori finora poco esplorati, aprendo nuove prospettive per la diagnosi e il trattamento dei disturbi del sonno. La tesi si svilupperà attraverso una sequenza di capitoli dedicati ai materiali e metodi, esperimenti, risultati e conclusioni, guidando il lettore attraverso un viaggio di scoperta nell'intersezione tra sonno, intelligenza artificiale e comprensibilità dei modelli.

Sarà attraverso questo approccio innovativo e interdisciplinare che ci si proporrà di contribuire alla crescente comprensione degli intricati meccanismi del sonno umano, puntando a tradurre la complessità delle onde cerebrali in immagini chiare e comprensibili.

Citando [3], tradizionalmente, l'attribuzione di onde cerebrali a una classe del sonno viene svolta nel seguente modo. Inizialmente, un soggetto dorme con un dispositivo medico che esegue una polisonnografia (PSG), registrando segnali di elettroencefalografia (EEG) in diverse posizioni sulla testa, segnali di elettrooculografia (EOG), segnali di elettromiografia (EMG) e, eventualmente, altri segnali. Successi-

vamente, un esperto del sonno umano esamina le diverse serie temporali registrate durante la notte e assegna a ciascun segmento temporale di 30 secondi una fase del sonno seguendo una nomenclatura di riferimento, come le regole dell'American Academy of Sleep Medicine (AASM) o le regole di Rechtschaffen and Kales (RK). Per quanto riguarda le regole dell'AASM (che sono quelle utilizzate per gli esperimenti che affronteremo), vengono identificate 5 fasi: Sveglio (Wake), Movimenti Rapidi degli Occhi (Rapid Eye Movements, o REM), Non REM1 (NREM1), Non REM2 (NREM2) e Non REM3 (NREM3, o DeepSleep), noto anche come sonno a onde lente o sonno profondo.

Non è difficile immaginare come il lavoro di un esperto, se distribuito su dati che coprono un intero ciclo di sonno (ad esempio 8 ore), diventi lento ed estenuante. Oltre a questo, il rischio di commettere potenziali errori dopo un numero così elevato di analisi è alto. È qui che entrano in gioco modelli di reti neurali in grado di attribuire i segmenti temporali di 30 secondi, denominati epoche, a classi del sonno. Un modello accurato potrebbe semplificare enormemente questi lunghi e tediosi processi, aiutando gli esperti a concentrarsi su altri aspetti come l'analisi dei risultati, o in caso di necessità, prescrizione di terapie a pazienti che ne hanno bisogno. Il problema più grande di queste considerazioni, è il fatto che nell'ambito dell'intelligenza artificiale e delle reti neurali, queste operazioni sono associate a delle *black box*.

Il concetto della *black box* può essere spiegato come segue: nonostante una rete neurale venga allenata tramite un processo di tranining con un dataset apposito, e sia successivamente in grado di fornirci risultati con un certo livello di accuratezza, noi esseri umani non siamo comunque in grado di capire quali informazioni la rete ha appreso durante il training per poter essere capace di assegnare i dati di input a un target, ad esempio un'immagine all'animale che raffigura, o un'epoca di onde cerebrali alla classe del sonno a cui appartiene. Prendiamo l'esempio dell'immagine, di più facile e immediata comprensione. Dopo aver mostrato alla rete neurale, durante la fase di training, un enorme numero di immagini di gatti, la rete è ora in grado, durante la fase di testing, di identificare se un'immagine raffigura effettivamente un gatto. Noi osservatori e analizzatori dei risultati, siamo in grado di stabilire se la rete neurale ha effettuato una giusta predizione o no, ma non possiamo andare oltre. *Come* ha fatto la rete a capire che quella foto raffigura un gatto? *Quali* sono gli aspetti e i dettagli delle immagini fornite durante il training che hanno permesso alla rete di capire effettivamente quali fossero gli elementi principali che contraddistinguono un gatto? Oltre a queste problematiche, riflettiamo anche sul fatto che se non siamo in grado di capire come la rete prende le sue decisioni, come potremmo migliorarne le prestazioni nel caso le sue previsioni siano di accuratezza non sufficientemente adeguata? Nel caso contrario invece, ossia che la rete abbia un'accuratezza elevatissima, tendente al 100% dei casi, cosa sta valutando la rete che gli esperti del settore non stanno considerando? Non trascurabile è infine un problema di natura etica e legale, ossia che un esperto, ad esempio un dottore, è impossibilitato ad effettuare una diagnosi o una prescrizione ad un paziente, se non è in grado di capire a fondo come la rete neurale ha effettuato le sue decisioni.

È proprio di questo che si occupa il campo dell'*eXplainable artificial intelligence*, ed è ciò di cui andremo a parlare in questa tesi, ovviamente in ambito dell'analisi del sonno. Nel dettaglio, andremo a vedere come si può capire in base a cosa la rete ha stabilito che una particolare epoca appartenga a una determinata classe del sonno. Per cominciare a fare una breve grafica introduzione di ciò che andremo a vedere, viene mostrata un'immagine delle principali onde, o bande, o range di frequenze, che compongono un'onda cerebrale.

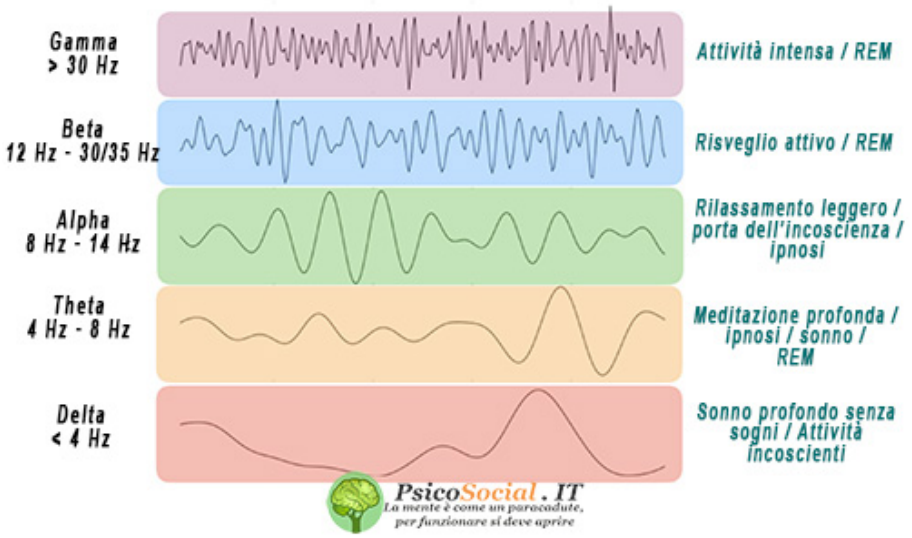

Figura 1.1: Una visualizzazione grafica delle bande che compongono un'onda cerebrale [8]

La figura 1.1 rappresenta 5 delle frequenze che fanno parte di un'onda cerebrale e che fanno parte degli input della rete neurale che sono stati analizzati e su cui si è lavorato. Vedremo più avanti che per la realizzazione delle nostre ricerche ed esperimenti sono state usate 6 onde anziché 5. Oltre a quelle presentate in questo capitolo, è stata analizzata anche l'onda Sigma, suddividendo i range di frequenze in modo tale da farli diventare leggermente più ristretti. Questa scelta è stata effettuata per rimanere il più vicino possibile ad altri set di esperimenti già effettuati e consultabili in [3].

Capitolo 2

Materiali e metodi

Il presente capitolo costituisce il fondamento metodologico della ricerca, delineando gli strumenti e le procedure adottate nell'ambito dello sviluppo di applicazioni basate su *eXplainable artificial intelligence* (XAI) per l'analisi del sonno. L'obiettivo primario di questa indagine è la creazione di una funzione di XAI in grado di restituire al chiamante una spiegazione di come il modello utilizzato prende le proprie decisioni, nel nostro caso analizzando l'importanza che diversi range di frequenze corrispondenti a un determinato set di bande hanno avuto nella scelta finale della classe target, consentendo una comprensione approfondita di come le onde cerebrali siano correlate a specifiche fasi del sonno.

Di seguito, verranno esaminate le tappe chiave del processo di ricerca. Si inizierà con la descrizione dei dati utilizzati, delineando la loro provenienza e le caratteristiche principali. Successivamente, saranno illustrati i dettagli della preparazione dei dati e l'elaborazione necessaria per garantire l'affidabilità delle analisi condotte.

Sarà introdotto il modello che definisce l'architettura della rete neurale impiegata nell'analisi delle onde cerebrali e nella previsione delle fasi del sonno per gli esperimenti svolti. Allo stesso modo, saranno presentate le metodologie implementate per rendere interpretabili i risultati ottenuti.

Infine, la sezione sarà completata con una panoramica sugli strumenti e le risorse utilizzate che hanno plasmato la metodologia di questa ricerca.

Attraverso questo esame dei materiali e dei metodi impiegati, si mira a fornire una base solida per la comprensione dei risultati ottenuti e per consentire la replicabilità e l'evoluzione di questa ricerca nel contesto dell'analisi del sonno assistita dall'XAI. Si fa inoltre presente che questo progetto e questa ricerca sono stati effettuati su basi solide già costruite. Il modello con cui sono stati elaborati i dati, così come i dati stessi, sono stati già forniti in partenza e non sono stati reperiti durante gli esperimenti, per cui non ci si soffermerà a lungo su questi aspetti, seppur verranno menzionati.

2.1 Modello e dati

Per condurre gli esperimenti che andremo a vedere nel prossimo capitolo, abbiamo usufruito del modello ideato da Stanislas Chambon e spiegato nel dettaglio in [3]. Per brevità, durante la ricerca abbiamo soprannominato il modello *chambon2018*. Nonostante questo, l'applicazione è stata accuratamente sviluppata in modo tale da poter funzionare con qualsiasi modello adottato in questo ambito, e continuare a restituire risultati coerenti.

I dati utilizzati sono composti da onde cerebrali catturate attraverso elettroencefalogrammi (EEG). Più dettagli su come possono essere ottenuti dati per la ricerca possono essere trovati in [3]. Affinché le analisi condotte siano affidabili e la rete neurale possa avere un dataset di training equilibrato, i dati sono stati suddivisi in modo tale da avere una equa distribuzione di onde cerebrali appartenenti a tutte le classi del sonno. Classi come NREM1 sono decisamente più rare e transitorie, e avere una mole di dati troppo poco consistente porterebbe ad un addestramento della rete non adeguato con conseguenti inconsistenze di valutazione.

Durante gli esperimenti svolti, abbiamo accompagnato la creazione dell'applicazione XAI a una funzione che potesse leggere i risultati ottenuti e crearne dei plot leggibili facilmente a occhio nudo. Questi grafici rendono i risultati facilmente comprensibili da chiunque, anche una persona non esperta del settore. Una volta appresi i concetti di *confidenza* della rete e *importanza* di una banda, non si avrà alcuna difficoltà nella lettura. D'altro canto, un esperto del settore potrebbe usufruire della lettura dei risultati di questi esperimenti per elaborare nuovi studi o ricerche nel settore, migliorare il modo in cui i modelli vengono implementati.

2.2 Strumenti e risorse

Per condurre gli esperimenti è stato usato il linguaggio Python (in particolare, Pytorch Lightning [1]) e si è fatto ampio uso di alcune delle sue librerie, tra cui, per menzionare le più importanti:

- NumPy [4], fondamentale per eseguire operazioni matematiche su array e matrici, anche multidimensionali.
- SciPy [9], grazie alla quale abbiamo potuto usufruire di filtri da applicare sui segnali di input, e al calcolo della funzione di Z Score per la normalizzazione di un set di valori.
- Pandas [7, 11], per costruire, salvare, caricare e manipolare Dataframe.
- Seaborn [10], per generare boxplot e heatmap.
- Matplotlib [6], per generare le immagini (in combinazione con Seaborn) che sono state salvate e che andremo a visualizzare nei prossimi capitoli.

- Itertools, per poter lavorare con tutte le possibili combinazioni di un set di dati, nel nostro caso, le bande che andremo ad analizzare.

Per la potenza di calcolo necessaria a condurre gli esperimenti è stato utilizzato Google Colab (o Colaboratory), una piattaforma gratuita basata su cloud offerta da Google che fornisce un ambiente di sviluppo basato su Jupyter Notebook. Per scrivere il codice invece, si è usufruito della funzionalità di codespace remoto offerta da GitHub, piattaforma su cui la ricerca e gli esperimenti svolti sono attualmente consultabili nella relativa repository [5], nella quale possiamo inoltre trovare ulteriori richiami al modello utilizzato (e ulteriori modelli utilizzabili), e al dataset fornito per gli esperimenti effettuati (nel nostro caso, è stato utilizzato il dataset Dreem).

Capitolo 3

Esperimenti

Gli esperimenti condotti sono stati suddivisi in tre diverse fasi, o stage, con l'obiettivo di cercare di capire a fondo quale sia l'importanza di un un determinato set di bande nella categorizzazione dei sample di input come appartenenti a una specifica classe del sonno. Le bande scelte per la conduzione degli esperimenti sono (prese come riferimento le bande in [3]):

- **Delta**: range di frequenze di [0.5 - 4 Hz]
- **Theta**: [4 - 8 Hz]
- **Alpha**: [8 - 11.5 Hz]
- **Sigma**: [11.5 - 15.5 Hz]
- **Beta**: [15.5 - 30 Hz]
- **Gamma**: [30 - 49.5 Hz]

3.1 Stage 1: Importanza delle bande per le classi del sonno

Durante il primo stage degli esperimenti è stata calcolata, per ogni banda, la sua importanza rispetto alle varie classi del sonno. La metodologia con cui è stato effettuato questo calcolo è stata ottenuta attraverso il seguente algoritmo:

1. Si ottiene la confidenza con la quale la rete neurale associa un determinato batch di input alle classi del sonno.
2. Si filtra l'input attraverso l'utilizzo di un filtro elimina banda, per escludere dall'input la banda della quale si vuole calcolare l'importanza.
3. Si ottiene nuovamente la confidenza con la quale la rete neurale associa il batch di input già esaminato in precedenza, ma questa volta filtrato, alle classi del sonno.

4. Si sottraggono le due confidenze ottenute prima e dopo aver filtrato l'input.

Il risultato della sottrazione tra le due confidenze ci permette di ottenere l'importanza della banda che è stata usata per filtrare l'input, rispetto alle varie classi del sonno.

$$Importanza = Confidenza_p - Confidenza_d \tag{3.1}$$

dove $Confidenza_p$ è la confidenza della rete calcolata prima di applicare il filtro, e $Confidenza_d$ è quella calcolata dopo.

3.1.1 Pseudocodice stage 1

Di seguito è fornita un'idea di pseudo codice che esplica come è stato eseguito lo Stage 1 degli esperimenti. Logicamente la funzione viene richiamata per ogni banda, in modo tale da ottenere l'importanza di tutte le bande.

```
# includere librerie di cui si ha bisogno...

def compute_band_importance(range_frequenza_banda_di_interesse, model, dataloader, ...):
    y_true = []
    y_pred = []
    importance = []

    for batch in dataloader:
        inputs, y_true_batch = batch

        # mi salvo la classe di appartenenza originaria dell'input
        y_true.append(y_true_batch.numpy())

        # lascio che il modello calcoli la confidenza
        pred_proba = model(...)
        # applico funzione di softmax in modo tale che l'array abbia valori che sommati diano 1

        # salvo la classe predetta
        y_pred.append(risultato della rete)

        # applico filtro elimina banda all'input tramite funzioni fornite dalla libreria scipy

        for index in range(batch_size):
            inputs[index] = filtro(inputs[index])

        # calcolo nuovamente la confidenza con l'input filtrato
        filtered_pred_proba = model(...)

```

```
29          # l'importanza della banda filtrata corrisponde con la
        differenza tra le due confidenze
30          batch_importance = pred_proba - filtered_pred_proba
31          importance.append(batch_importance)
32
33      return importance, y_pred, y_true
```

3.2 Stage 2: Considerazione sull'effetto dell'influenza delle bande sulle altre bande

I risultati dello stage 1 degli esperimenti, che analizzeremo nel prossimo capitolo, sono sicuramente interessanti, ma purtroppo non abbastanza accurati per poterci ritenere soddisfatti. Cerchiamo brevemente di capire il perché, con un semplice esempio matematico. Supponiamo di avere due funzioni: $f_1(x, y) = x + y$, e $f_2(x, y) = x \cdot y$. Per calcolare l'importanza delle variabili d'ingresso di queste due funzioni, dobbiamo calcolarne il gradiente. Proviamo a farlo. Il gradiente, ossia il vettore delle derivate parziali rispetto alle variabili d'ingresso, verrà rispettivamente: $\nabla f_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ e $\nabla f_2 = \begin{bmatrix} y \\ x \end{bmatrix}$. Supponiamo ora, per ricollegarci all'esempio fatto in precedenza, di decidere di "occludere" una delle variabili d'ingresso (come prima abbiamo filtrato una delle bande appartenenti all'input), per esempio y, e di calcolare l'importanza della variabile x. Nel primo caso è semplice, poiché dal momento che la variabile x non dipende in alcun modo da y, la derivata parziale della funzione rispetto a x rimane sempre 1, perciò l'importanza è 1, ma nel secondo caso? Nel secondo caso, possiamo osservare come l'importanza della variabile x era esattamente y, ma se y è stata occlusa, siamo adesso impossibilitati a calcolarla, e questo può causare dei problemi nei calcoli dell'esperimento. Per porre il problema in termini prettamente matematici, se la funzione presa in considerazione ha la matrice Hessiana (ossia la matrice delle derivate seconde) non diagonale, allora siamo impossibilitati a calcolare l'importanza delle variabili d'ingresso attraverso il semplice uso del gradiente. Verifichiamo infatti che le matrici Hessiane delle nostre due funzioni di partenza sono, rispettivamente: $Hf_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, e $Hf_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Come si può facilmente verificare, la seconda Hessiana non è diagonale, per cui la funzione f_2 sarebbe fonte di problemi nel calcolo dell'importanza.

Lo stage 2 degli esperimenti effettuati si basa proprio su questo problema e cerca di risolverlo. Per farlo, cambiamo l'algoritmo con il quale calcoliamo l'importanza di una banda, in modo tale da considerare che nel calcolo dell'importanza di una banda, è presente anche l'influenza di tutte le altre. Più nel dettaglio, i passi sono i seguenti:

1. Si individua la banda di cui vogliamo ottenere l'importanza, e si costruiscono tutte le possibili combinazioni di bande in cui la banda individuata sia presente.

2. Si ripetono i seguenti passaggi esaustivamente per ogni combinazione:

   (a) Si ottiene la confidenza con la quale la rete neurale associa un determinato batch di input alle classi del sonno.

   (b) Si filtra l'input attraverso l'utilizzo di uno o più filtri elimina banda, per escludere dall'input le bande che fanno parte della combinazione presa in considerazione.

   (c) Si ottiene nuovamente la confidenza con la quale la rete neurale associa il batch di input già esaminato in precedenza, ma questa volta filtrato, alle classi del sonno.

   (d) Si sottraggono le due confidenze ottenute prima e dopo aver filtrato l'input.

3. Una volta esaurite tutte le combinazioni e ottenuto quindi un ampio set di importanze relative a ognuna di queste, si effettua una media per ottenere l'importanza della banda scelta inizialmente.

Una volta ottenuta l'importanza della banda scelta, gli esperimenti sono stati condotti anche sul resto delle bande, e sono stati poi confrontati i risultati ottenuti in questo stage, con quelli ottenuti alla fine dello stage 1. Negli esperimenti effettuati, si è tenuto conto di due diverse tipologie di calcolo della media. La prima, che è stata denominata *media semplice* o *simple average*, è stata ottenuta sommando tra loro tutte le diverse importanze delle varie combinazioni delle bande, e poi dividendo il risultato per il numero di elementi considerati. La seconda, denominata *media pesata* o *weighted average*, è stata ottenuta sommando tra loro tutti i contributi delle diverse importanze, come nella prima tipologia, ma ognuno dei contributi è stato moltiplicato per un peso relativo, o per meglio dire un *fattore moltiplicativo*, corrispondente a $1/B$, dove B è il numero di bande presenti nella combinazione corrispondente al contributo che si sta sommando. Per esempio, se sto sommando l'importanza del set di bande [Alpha, Beta], questo contributo è stato moltiplicato per 1/2. La somma di tutti i contributi moltiplicati per il peso relativo, è stata poi divisa per la somma dei pesi relativi, in modo tale da normalizzare i pesi e rendere il risultato una media pesata a tutti gli effetti.

Anche i risultati dello stage 2 degli esperimenti sono interessanti, e verranno analizzati nel prossimo capitolo e confrontati con quelli dello stage 1. Vedremo che in linea di massima le considerazioni più importanti rimangono le medesime, ma le immagini mostreranno dei comportamenti che fanno capire che stiamo imboccando la strada giusta.

3.2.1 Pseudocodice stage 2

Nello pseudocodice dello stage 2 mostriamo come viene introdotto il filtraggio di molteplici bande all'input, e come si può implementare il calcolo della media delle

importanze appartenenti a tutte le possibili combinazioni in cui una banda target compare (logicamente le operazioni vanno ripetute per ogni banda, se si vogliono conoscere tutte)

```
# includere librerie di cui si ha bisogno...

def _compute_cross_band_importance(bands, ...):
    # bands contiene il set di bande che fanno parte della combinazione da filtrare

    ...

    # come nello stage 1, ma con la seguente differenza
    for band in bands:
        # applico filtro elimina banda all'input tramite funzioni fornite dalla libreria scipy
        for index in range ( batch_size ) :
            inputs [ index ] = filtro ( inputs [ index ])

    ...

    return importance, y_pred, y_true

def compute_band_importance(...):

    # band_freq_combinations e' la lista in cui finiranno le varie combinazioni. in particolare e' una lista di liste. ogni elemento della lista e' una lista di combinazioni di bande. il primo elemento e' la lista di combinazioni di 1 banda, il secondo elemento e' la lista di combinazioni di 2 bande, e cosi' via
    band_freq_combinations = []

    for i in range(num_bande):
        combination_list = lista_combinazioni_i+1_bande
        for elem in combination_list:
            band_freq_combinations.append(elem)

    for cross_band in band_freq_combinations:
        permuted_bands = [0 0 ... 0]

        for i, band in enumerate( bands ):
            if band in cross_band:
                permuted_bands [i] = 1

        # abbiamo messo a 1 gli elementi di permuted_bands corrispondenti alle bande della combinazione attuale

        importance, y_pred, y_true = _compute_cross_band_importance(cross_band, model, dataloader, model_device, sampling_rate)
```

```
38
39      importances_matrix = []
40      permuted_bands_importance = # lista di importanze dei set
    di bande contenuti in band_freq_combinations..
41      permutations_array = # lista che contiene tutte le
    possibili combinazioni di 0 e 1 composte da 6 elementi,
    esclusa quella di tutti 0
42
43      for i in range(num_bande):
44          # per ogni banda, calcolo la media semplice e la media
     pesata di tutte le importanze di quelle combinazioni dove
    la banda compare. Per farlo, specifico la banda selezionata
     (i) come parametro della funzione che calcola la media,
    cosi' da sapere quali sono le importanze da selezionare
45
46          # l'average type e' da passare come parametro alla
    funzione
47          # simple_average
48          if average_type == 0:
49              band_importance = get_simple_importance(
    permuted_bands_importance, permutations_array, i)
50          #weighted_average
51          elif average_type == 1:
52              band_importance = get_weighted_importance(
    permuted_bands_importance, permutations_array, i)
53
54          importances_matrix.append(band_importance)
55
56      return importances_matrix, y_pred, y_true
```

3.3 Stage 3: Introduzione del concetto dell'importanza nel tempo

Nei primi due stage degli esperimenti abbiamo visto come determinare in maniera accurata l'importanza di una banda nei confronti delle determinate classi del sonno. In relazione al nostro problema, ossia quello di stabilire in che modo la rete neurale prende le sue decisioni, i risultati ottenuti sono sicuramente di grande aiuto per capire quali sono le informazioni che la rete prende in considerazione per la decisione finale. Nonostante questo, possiamo effettuare ancora un passo successivo, e andare ad introdurre il concetto di importanza nel tempo. Nel determinare l'importanza delle bande fino a questo momento, abbiamo sempre preso dei batch di input che, nel nostro caso, sono composti da 3 epoche di 30 secondi l'una, campionate a una frequenza di 100Hz. In altre parole, questo comporta che fino ad ora abbiamo sempre associato l'importanza di una banda ad un determinato periodo di tempo di 90 secondi. L'obbiettivo dello stage 3 degli esperimenti è quello di tentare di risolvere il problema di capire, ad ogni istante di tempo, quale sia l'importanza delle 6 bande

prese in considerazione, sapendo che quell'istante di tempo fa parte di un batch di input la cui analisi dei sample ha prodotto una determinata predizione. In altre parole: presa una classe, che può essere la classe predetta dalla rete o una classe target, prendiamo un batch di input che sappiamo per certo corrispondere alla classe selezionata (nel caso la classe sia quella predetta dalla rete, allora bisogna sempre tenere a mente che l'input che viene preso è un batch del quale la rete ha predetto una determinata classe, ma la vera classe di appartenenza potrebbe essere un'altra, in quanto è sempre possibile che la rete abbia commesso un errore). Una volta preso il nostro input, attraverso l'uso della funzione IntegratedGradients fornita dalla libreria Captum [2], con un algoritmo simile a quello utilizzato durante lo stage 2 andiamo a trovarci, per ogni banda (considerando tutte le varie combinazioni...), la sua importanza rispetto a ogni classe del sonno, per ogni singolo istante di tempo contenuto nell'input considerato. Al termine di questo processo, otteniamo 5 matrici, una per ogni classe del sonno, corrispondenti all'importanza nel tempo di tutte le bande. In particolare, le righe delle matrici corrisponderanno al nostro set di bande [3], mentre le colonne corrisponderanno ai sample dell'input preso in considerazione, per ognuno dei quali le nostre bande avranno una determinata importanza nei confronti della classe considerata.

Per poter visualizzare al meglio questi risultati, sono stati generati dei subplot attraverso la libreria matplotlib, all'interno dei quali possiamo osservare delle heatmap che ci permettono di vedere in modo grafico l'importanza delle bande nel tempo, affiancate all'immagine dell'input non filtrato, così da poter associare a ogni sample della matrice delle importanze, lo stesso sample di input nella sua forma originale. Anche questi risultati verranno visualizzati e discussi nel prossimo capitolo. Per avere una più ampia visione dei risultati, sono stati adottati due approcci differenti nella creazione dell'heatmap. Dal momento che analizzando l'importanza nel tempo delle bande, in alcuni istanti di tempo è possibile che l'importanza sia negativa, il primo approccio è stato quello di usare una colormap che andasse dal blu (per le importanze negative) al bianco (tendente allo 0) al rosso (per le importanze positive). I valori della matrice sono stati normalizzati (attraverso l'uso di Z Score [12]) tra -1 e 1 per rendere meglio questa distribuzione di colori. Il secondo approccio è stato quello di applicare il valore assoluto alla matrice delle importanze, rendendo quindi positivi i valori negativi, e applicare una normalizzazione che andasse da 0 a 1. La scala di colori utilizzata va dal bianco (tendente allo 0) al rosso (tendente a 1).

3.3.1 Pseudocodice stage 3

Riprendendo dallo pseudocodice dello stage 2, in quest'ultimo stage aggiungiamo il calcolo dell'importanza nel tempo e quindi poi della media tra le varie importanze nel tempo di combinazioni di bande in cui sia presente la banda target di cui vogliamo calcolare l'importanza nel tempo finale. Anche qui, se vogliamo conoscere

l'importanza nel tempo di tutte le bande, l'operazione va ripetuta per ognuna di esse.

```
# includere librerie di cui si ha bisogno...

def _compute_cross_band_importance(bands, ...):
    # bands contiene il set di bande che fanno parte della combinazione da filtrare
    time_importance = []
    ...

    # come nello stage 2, ma oltre a calcolare l'importanza del set di bande, calcoliamo anche l'importanza nel tempo attraverso la funzione IntegratedGradients di Captum
     ig = IntegratedGradients(model)

        partial_time_importance = []
        for c in num_classi:
            partial_time_importance.append(ig.attribute(inputs, inputs_filtrati, target=c))

        time_importance.append(partial_time_importance)

    ...

    return time_importance, importance, y_pred, y_true

def compute_band_importance(...):
    # per lo piu' simile allo stage 2, ma ora dalla chiamata alla funzione _compute_cross_band_importance riceveremo anche la time_importance, della quale andremo a calcolarci la media. La funzione finale dovra' restituire sia le importanze come alla fine dello stage 2, sia le importanze nel tempo.

    ...

    time_importance, band_importance, y_pred, y_true = _compute_cross_band_importance(cross_band, model, dataloader, model_device, sampling_rate)

    ...

    importances_matrix = []
    time_importances_matrix = []
    permuted_bands_importance = # lista di importanze dei set di bande contenuti in band_freq_combinations..
    permutations_array = # lista che contiene tutte le possibili combinazioni di 0 e 1 composte da 6 elementi, esclusa quella di tutti 0

```

```
35      for i in range(num_bande):
36          # oltre al calcolo fatto durante lo stage 2, aggiungo
    anche il calcolo della time_importance come media di tutte
    le time_importance delle combinazioni in cui compare la
    banda i
37
38          # l'average type e' da passare come parametro alla
    funzione
39          #simple_average
40          if average_type == 0:
41              band_importance = get_simple_importance(
    permuted_bands_importance, permutations_array, i)
42              band_time_importance = get_simple_importance(
    permuted_bands_time_importance, permutations_array, i)
43          #weighted_average
44          elif average_type == 1:
45              band_importance = get_weighted_importance(
    permuted_bands_importance, permutations_array, i)
46              band_time_importance = get_weighted_importance(
    permuted_bands_time_importance, permutations_array, i)
47
48          importances_matrix.append(band_importance)
49          time_importances_matrix.append(band_time_importance)
50
51      return time_importances_matrix, importances_matrix, y_pred
    , y_true, importances_df
```

Capitolo 4

Risultati

Dopo aver discusso la metodologia con cui sono stati affrontati gli esperimenti e aver mostrato un possibile pseudocodice per ognuno degli stage, mostrando una possibile idea di implementazione, andiamo ora ad analizzare i risultati che le varie fasi degli esperimenti hanno prodotto. In particolare, per i primi due stage, andremo a vedere dei boxplot che ci mostreranno l'importanza delle bande nei confronti delle cinque classi del sonno, e faremo delle considerazioni per compararli e confrontarli tra loro, mettendo in luce la possibile presenza di similitudini o differenze. In linea di massima, per stabilire l'importanza della banda nei confronti di una particolare classe, menzioneremo il valore della mediana. Con il termine *overlapping* intenderemo invece quel fenomeno in cui il boxplot di una particolare classe si sovrappone con quello di un'altra. Nell'analisi dello stage 3, mostreremo invece alcune specifiche immagini prodotte per ogni classe, che ci mostrano i possibili approcci discussi nel precedente capitolo. È importante dire che il dataset su cui gli esperimenti sono stati condotti era di dimensioni notevoli, ed è stato diviso in 10 *fold*, per ognuna delle quali sono state prodotte lo stesso numero di immagini per calcolare le importanze. Per brevità, di seguito verranno mostrati i risultati solamente degli esperimenti condotti sulla *fold* 0, che sono in ogni caso coerenti e comparabili senza notevoli differenze con quelli delle altre *fold*, ad eccezione di un singolo particolare caso di cui discuteremo poco più avanti.

4.1 Stage 1

Di seguito andremo ad analizzare le immagini prodotte dallo stage 1 degli esperimenti, ossia le importanze delle bande rispetto alle classi del sonno, calcolate come la differenza tra la confidenza della rete con l'input non filtrato, e quella con l'input filtrato, dove con filtrato intendiamo che è stata esclusa, attraverso un filtro elimina banda, la banda per la quale vogliamo calcolare l'importanza.

4.1.1 Alpha

In figura 4.1 possiamo notare come la classe Wake è quella per la quale la banda Alpha è più importante, per un valore che tocca lo 0.4. Si può notare un overlapping con la classe NREM1, ma in generale non è raro che classi adiacenti abbiano overlapping, poiché durante il sonno, le transizioni da una classe a un'altra avvengono sempre tra classi adiacenti tra loro, per cui è normale trovare tracce di una banda in entrambe le classi (in questo caso, Wake e NREM1).

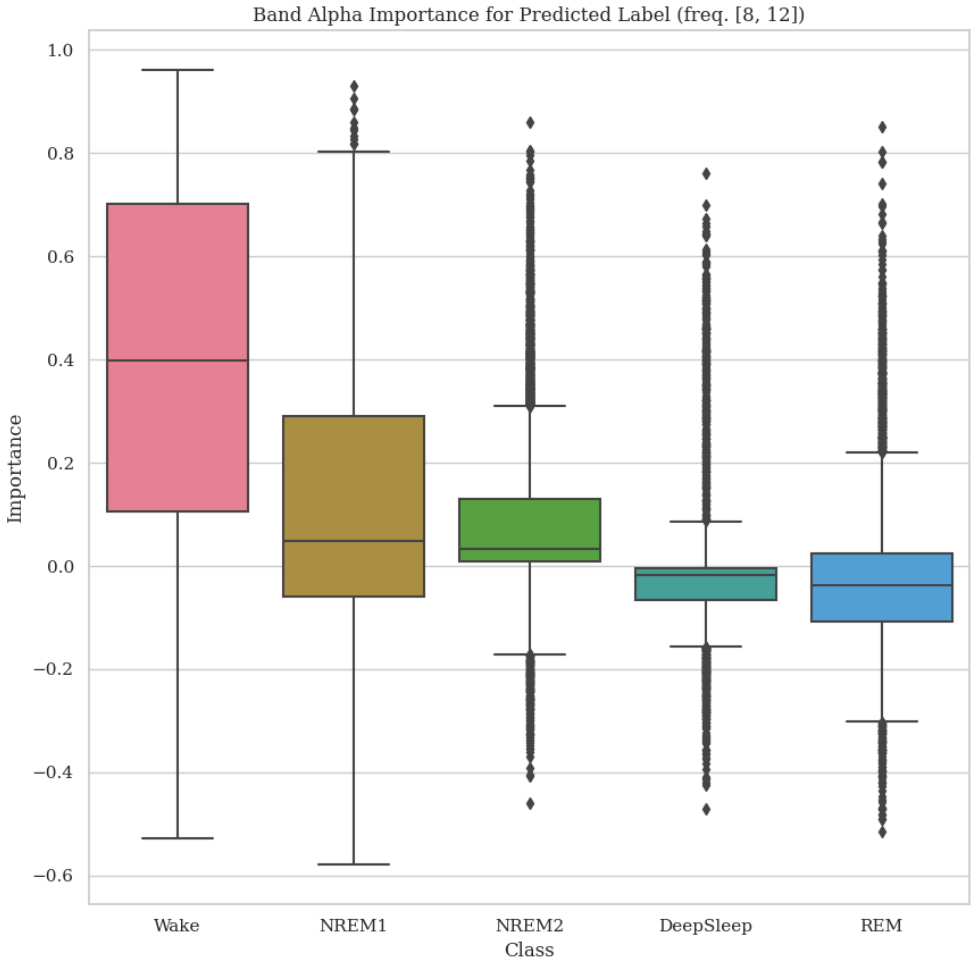

Figura 4.1: Importanza dell'onda Alpha rispetto alle classi del sonno predette dalla rete neurale

4.1.2 Beta

Nell'analizzare l'onda Beta (figura 4.2) ci accorgiamo di come non ci sia nessuna classe del sonno per cui la sua importanza appaia elevata. Potrebbe sembrare un'a-

nomalia ma in realtà non è un comportamento inaspettato, dal momento che le frequenze dell'onda Beta sono elevate. Evidentemente il modello non ha basato le sue decisioni su questo range di frequenze, ed a supportare questa ipotesi ci sarà l'immagine delle importanze dell'onda Gamma, che presenta frequenze ancora più alte.

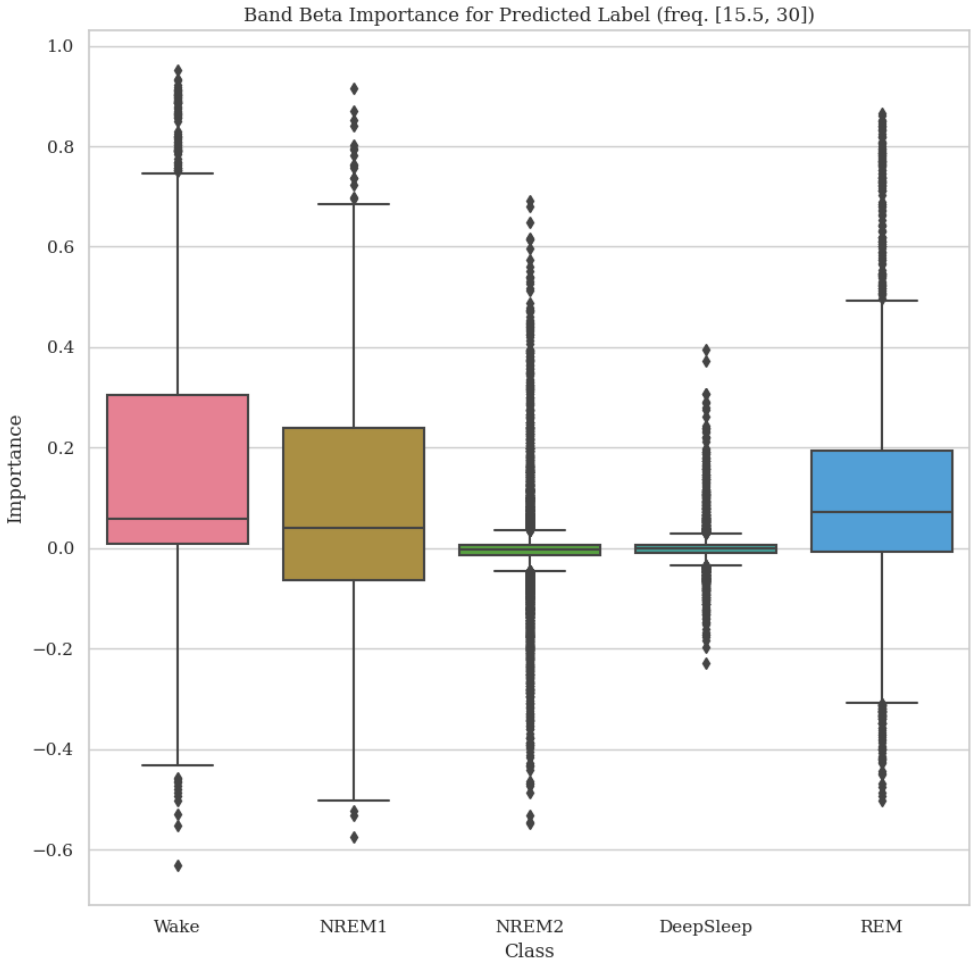

Figura 4.2: Importanza dell'onda Beta

4.1.3 Delta

Fondamentale è l'onda Delta per la classe DeepSleep. Il fatto che sia risaputo in ambito scientifico che le onde Delta sono studiate ed analizzate per riconoscere se, ad esempio, un paziente si trova in uno stadio di sonno profondo, è molto rassicurante quando notiamo che la rete con la quale abbiamo condotto gli esperimenti reputa che l'onda Delta abbia un'importanza di valore sostanzialmente tendente a 1 per

la classe DeepSleep (figura 4.3). Notiamo come anche l'importanza per le classi NREM2 e REM siano alte, dal momento che ci sono transitori in cui, nonostante in uno stadio del sonno diverso da DeepSleep, le onde Delta sono comunque presenti.

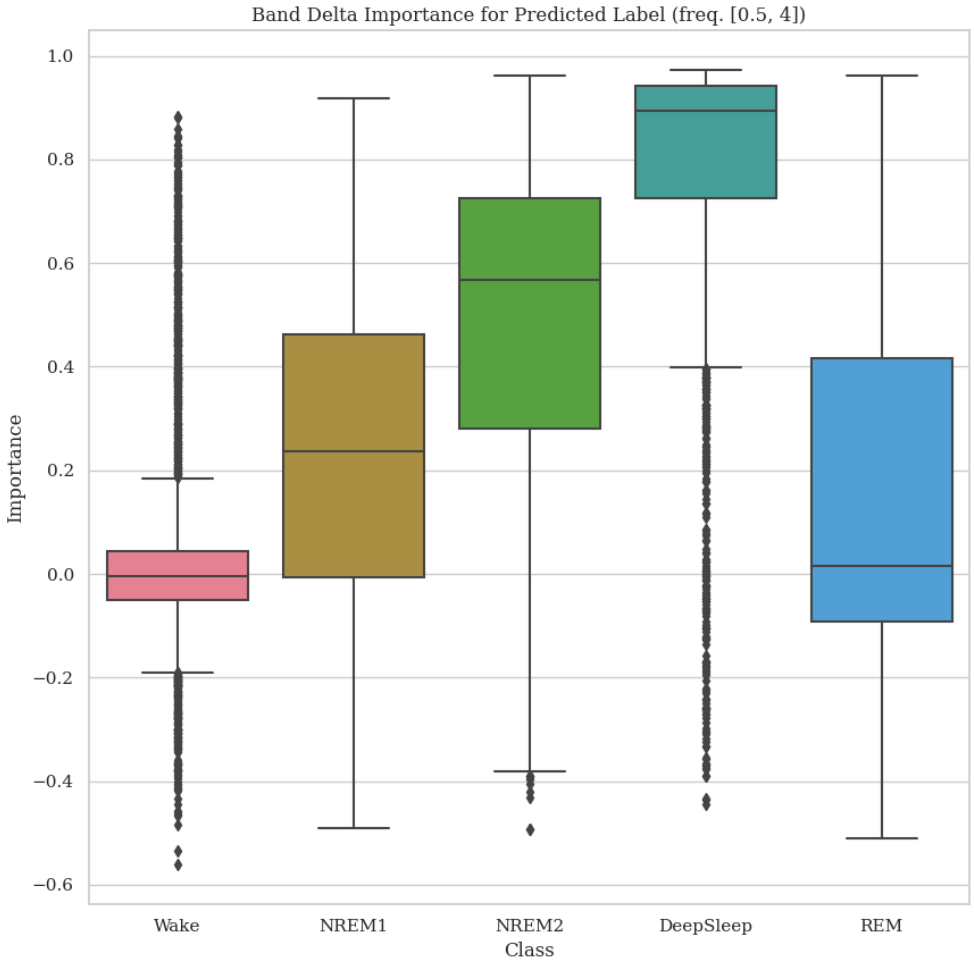

Figura 4.3: Importanza dell'onda Delta

4.1.4 Gamma

Come abbiamo ipotizzato prima mentre analizzavamo le onde Beta, possiamo qui verificare (figura 4.4) che anche per le onde Gamma l'importanza per qualsiasi classe è sostanzialmente nulla. Possiamo attribuire la causa alle elevate frequenze dell'onda.

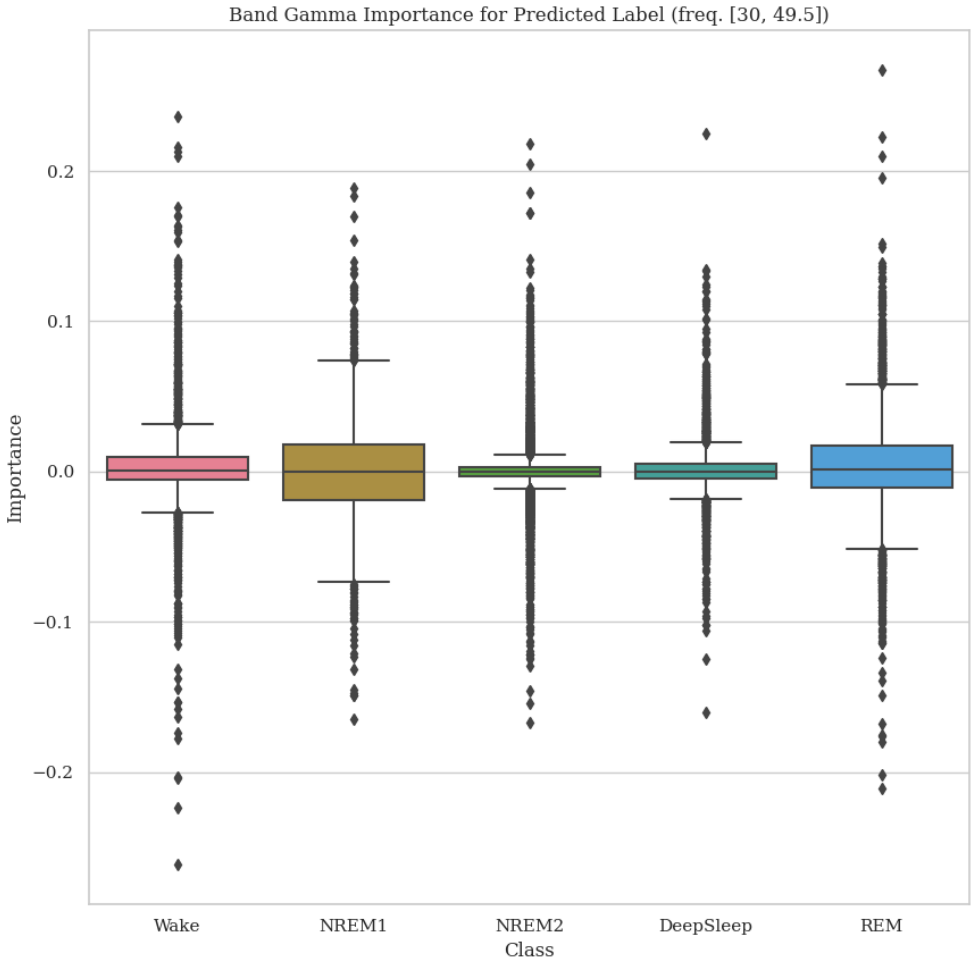

Figura 4.4: Importanza dell'onda Gamma

4.1.5 Sigma

L'onda Sigma, per le sue frequenze che cominciano ad essere elevate, presenta comportamenti riconducibili a quelli di cui abbiamo discusso per l'onda Beta (figura 4.5). Nonostante questo, è comunque visibile un'importanza leggermente più alta per la classe NREM2, anche se in generale, le classi più studiate per l'analisi del sonno sono Wake, DeepSleep e REM, mentre NREM1 e NREM2 sono considerate prevalentemente come stadi transitori. È sicuramente degno di nota riportare come nelle immagini prodotte dalle altre *fold*, l'importanza per la classe NREM2 sia in generale più alta di quella della *fold* 0. Questo aspetto varrà anche per i risultati che analizzeremo nello stage 2. Prendendo nota di ciò, si può affermare anche se non osservabile in figura in modo particolarmente evidente, la classe NREM2 è quella per cui le onde Sigma hanno un'importanza maggiore.

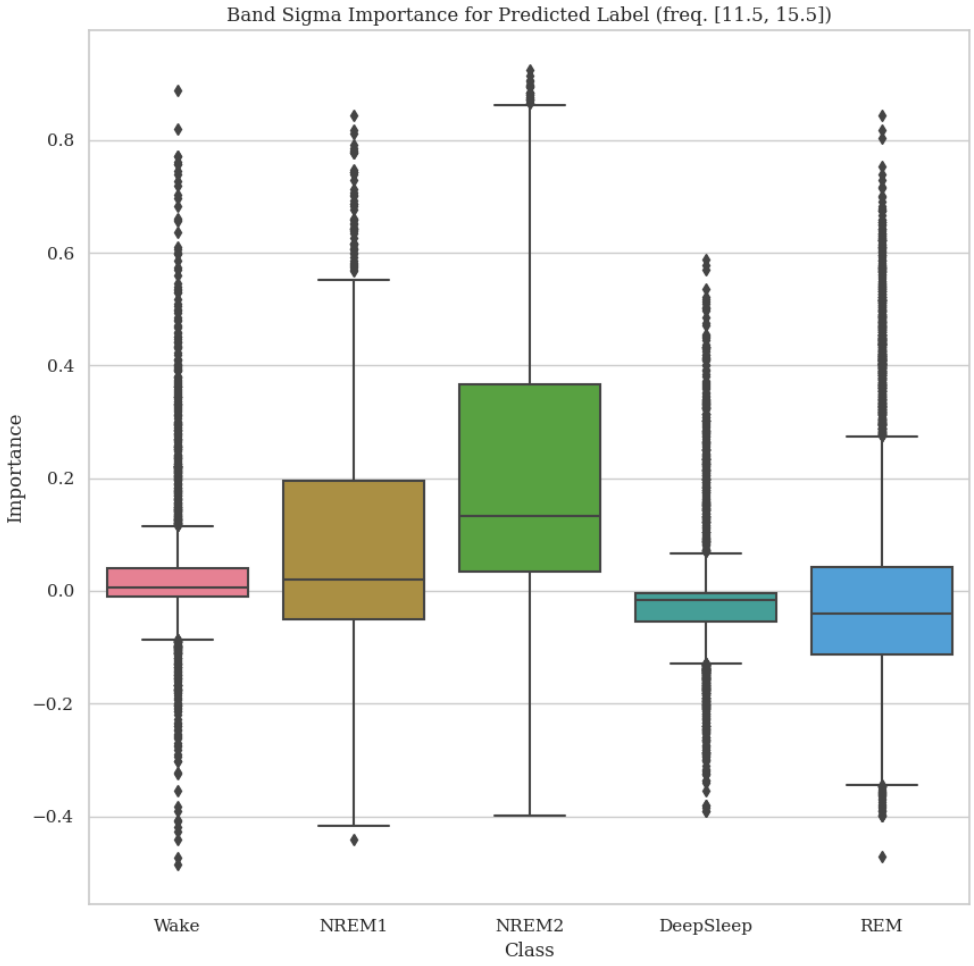

Figura 4.5: Importanza dell'onda Sigma

4.1.6 Theta

Sono le classi NREM1 e REM ad essere quelle infuenzate di più dall'importanza delle onde Theta secondo le immagini prodotte dallo stage 1 dei nostri esperimenti (figura 4.6).

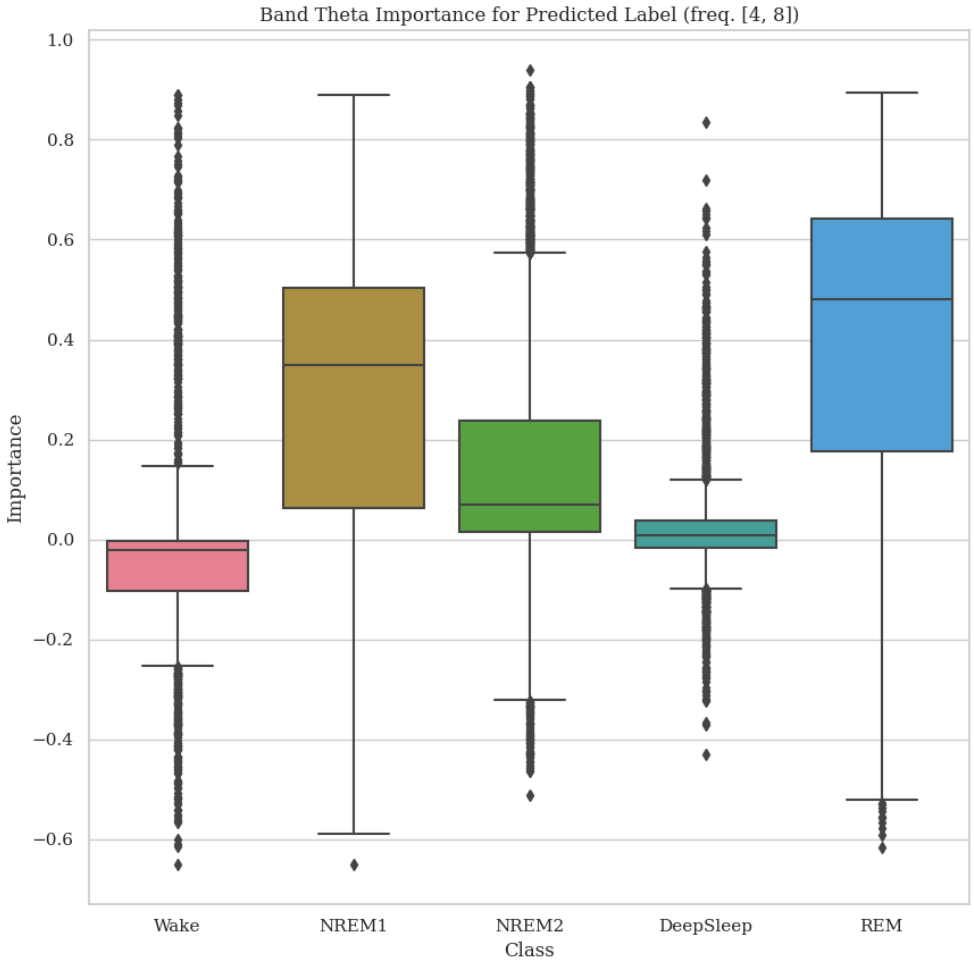

Figura 4.6: Importanza dell'onda Theta

4.2 Stage 2

Nello stage 2 degli esperimenti viene cambiata la modalità di calcolo dell'importanza delle bande, introducendo le medie fra le importanze dei set di bande che fanno parte delle possibili combinazioni di bande di cui il range di frequenze di cui vogliamo calcolare l'importanza fa parte. Di seguito verranno mostrate, per brevità, le immagini corrispondenti al calcolo della *media pesata*, tenendo a mente che i risultati corrispondenti alla *media semplice* sono coerenti e comparabili senza grosse differenze degne di nota. In continuità con quanto mostrato fino ad ora, continueranno ad essere mostrati i risultati ottenuti elaborando il dataset della *fold* 0.

4.2.1 Alpha

Il primo aspetto che possiamo notare di questi nuovi risultati, a differenza di quelli dello stage 1, è che nessuna, o quasi nessuna, delle classi del sonno, hanno importanza pari (o comunque molto tendente) a 0. Questo aspetto sarà osservabile anche nelle immagini che analizzeremo a breve per le altre onde, e il motivo è proprio quello che ora, nello stage 2, per calcolare l'importanza di una banda abbiamo preso in considerazione anche l'influenza di tutte le altre. È quindi normale aspettarsi che le importanze per le altre classi salgano, seppur di poco. Con questa nuova metodologia di calcolo, vediamo come (figura 4.7) l'importanza delle onde Alpha per la classe Wake è ancora comparabile ai livelli precedenti (intorno a 0.4), con la classe NREM1 che rimane vicina in quanto transitorio. Vediamo però ora alzarsi anche le classi NREM2 e DeepSleep, causando un overlapping esteso un po' per tutte le classi e non permettendoci bene di stabilire le onde Alpha come prevalentemente importanti per una delle classi in particolare

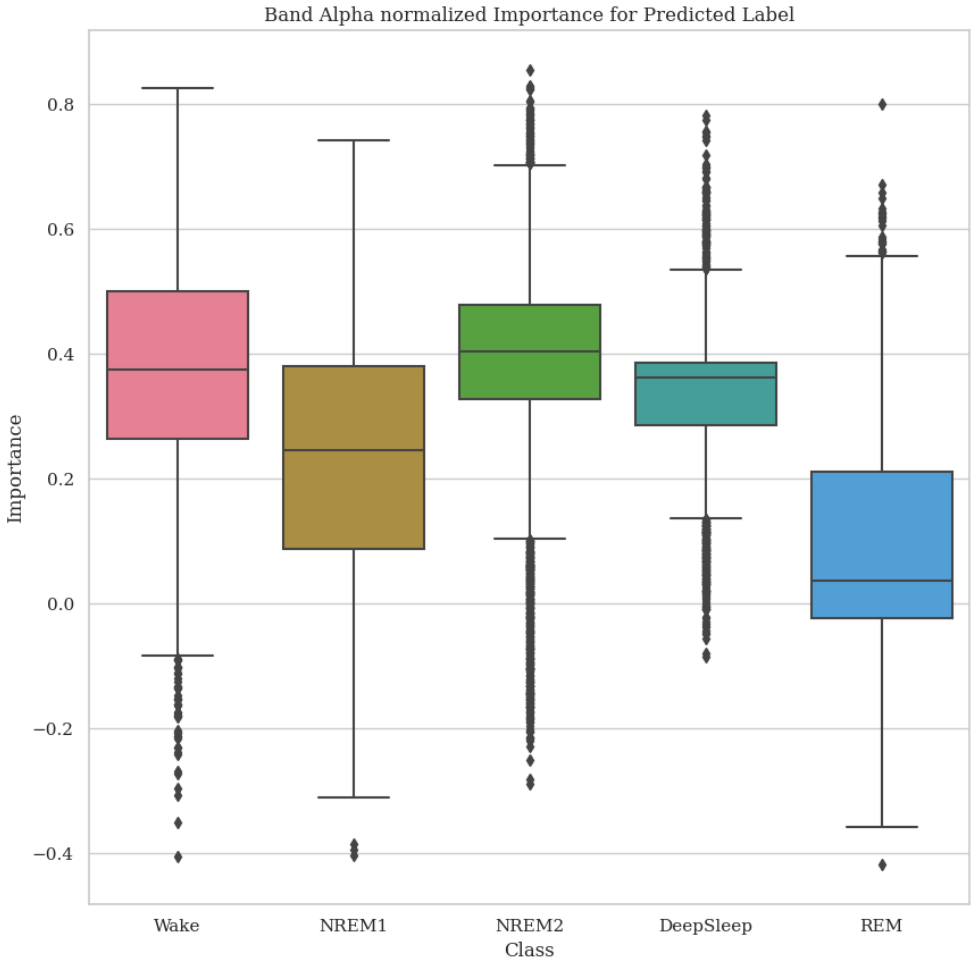

Figura 4.7: Importanza dell'onda Alpha calcolata con la media pesata

4.2.2 Beta

Come per il risultato prodotto durante lo stage 1, l'onda Beta non sembra avere una preponderante importanza verso una classe in particolare (figura 4.8). Come accennato in precedenza però, anche per le onde Beta tutte le importanze si alzano leggermente dal valore 0, in quanto l'influenza di tutte le altre bande si mostra sull'importanza di una sola. Da questo momento in poi eviteremo di specificarlo, ma questo è un effetto che continuerà a ripetersi anche nei successivi boxplot. Anche per lo stage 2, risultati simili si osserveranno nell'onda Gamma, a causa delle elevate frequenze.

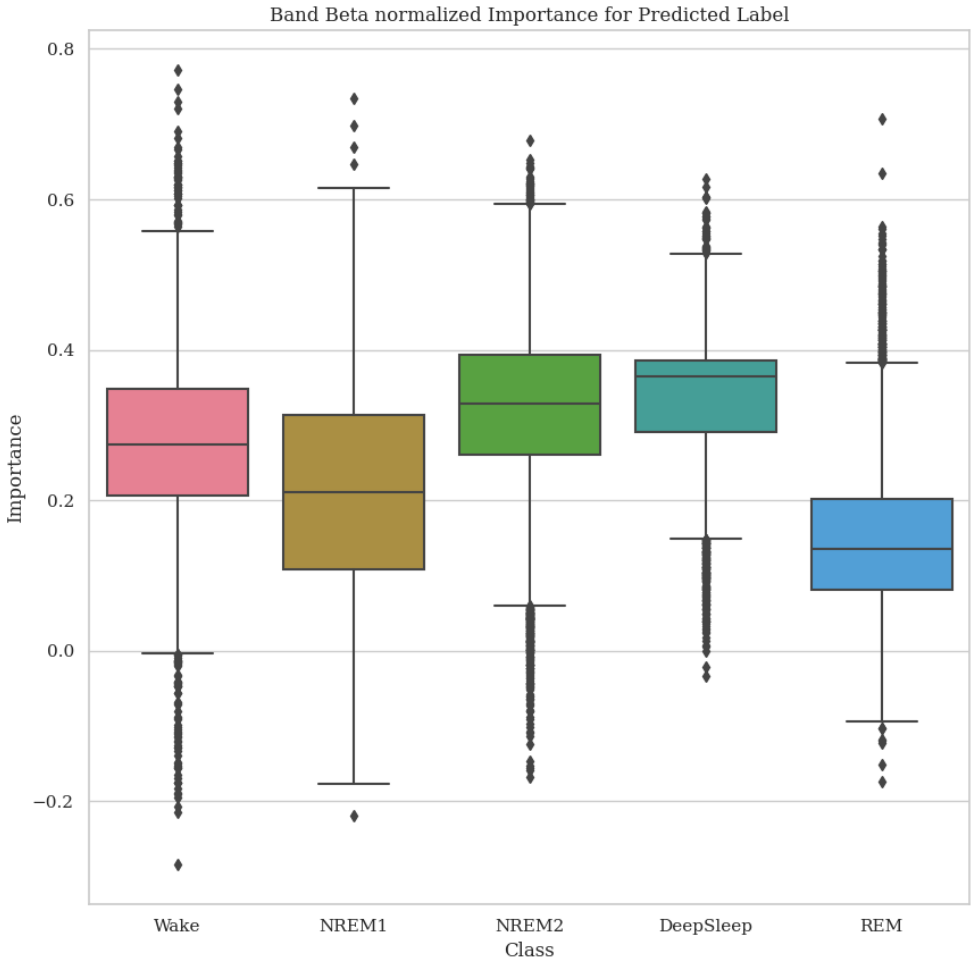

Figura 4.8: Importanza *pesata* dell'onda Beta

4.2.3 Delta

La banda delta rimane la banda fondamentale per riuscire a capire se ci troviamo nella classe DeepSleep. Con un'importanza essenzialmente tendente a 1 (figura 4.9), non ci sono dubbi su quale sia il range di onde che il modello cerca per capire se ci troviamo in uno stato di sonno profondo. Molto alta comunque, in quanto stadio transitorio, anche l'importanza di NREM2, con un leggero overlapping sulla classe DeepSleep.

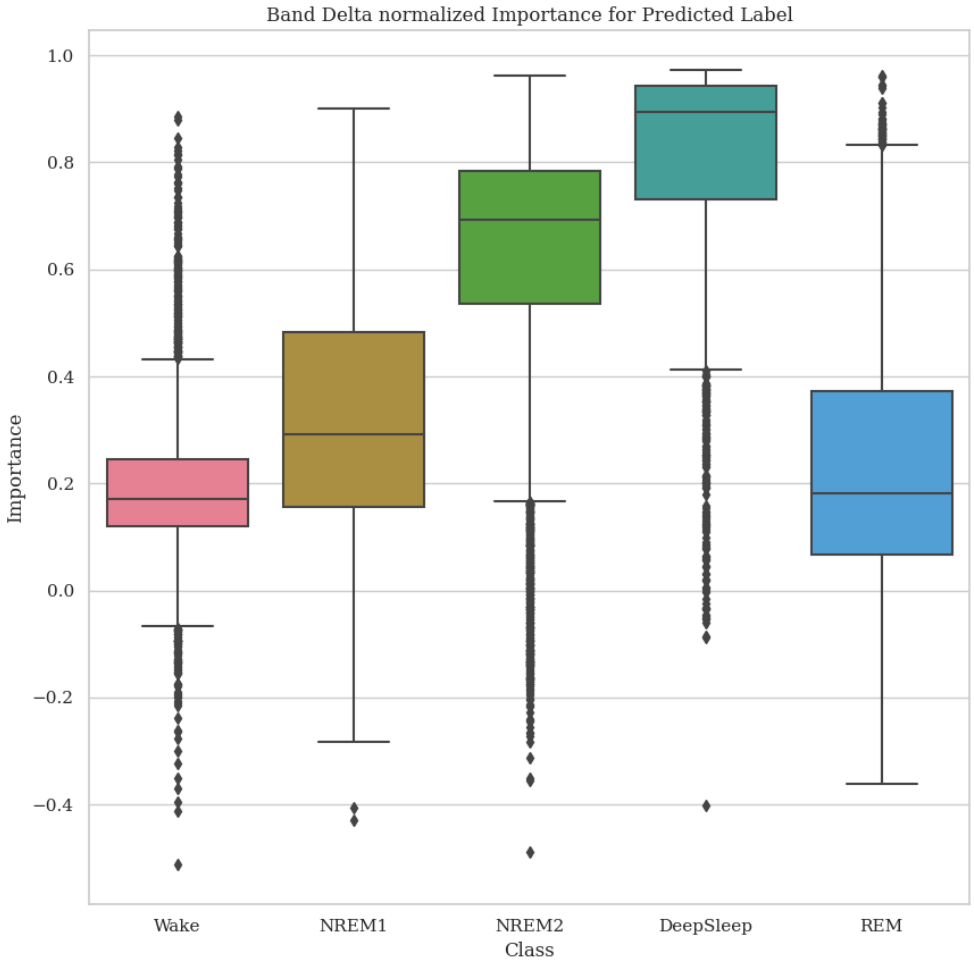

Figura 4.9: Importanza *pesata* dell'onda Delta

4.2.4 Gamma

Come già anticipato, le alte frequenze delle onde Gamma impediscono l'attribuzione della sua importanza a una classe del sonno in particolare, anche se notiamo come le classi NREM2 e DeepSleep si elevino leggermente sulle altre (figura 4.10).

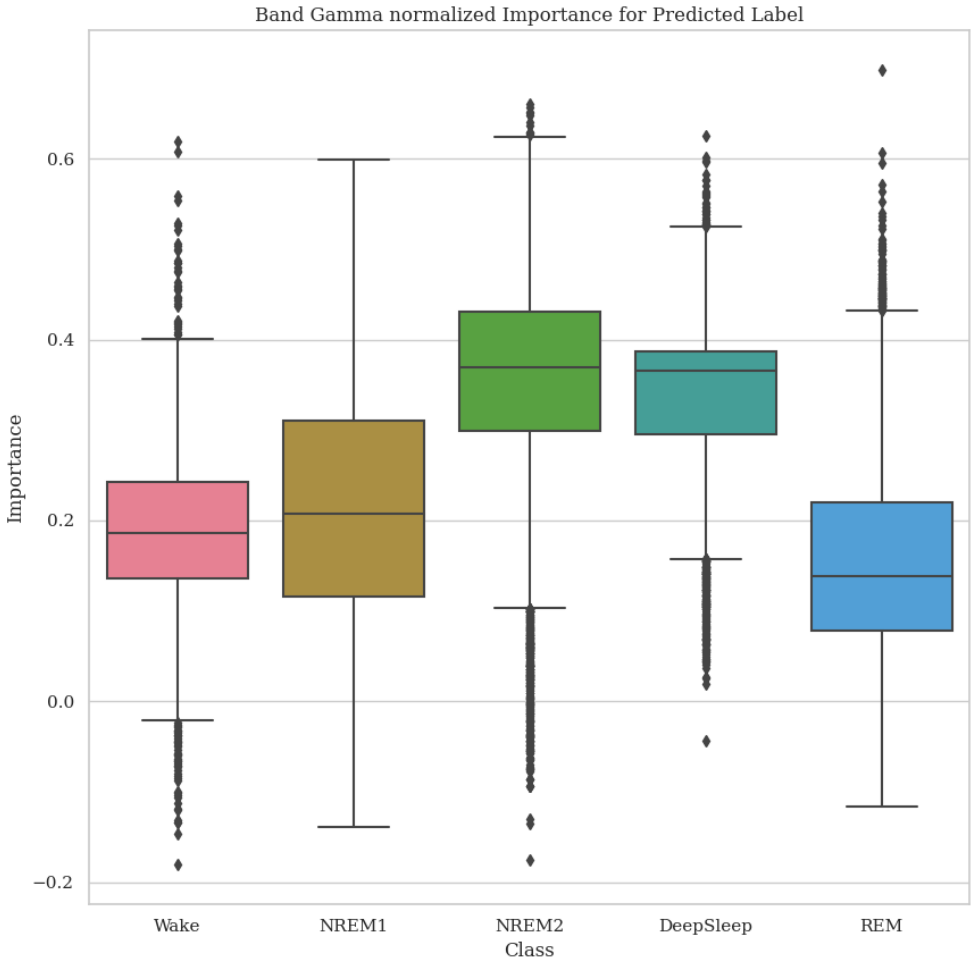

Figura 4.10: Importanza *pesata* dell'onda Gamma

4.2.5 Sigma

Coerentemente ai risultati dello stage 1, anche per lo stage 2 l'onda Sigma rimane importante per la classe NREM2 (figura 4.11), con le stesse considerazioni sulle *fold* fatte sull'immagine dello stage 1. In quanto transitori, NREM1 e DeepSleep in questo caso sono leggermente più alti di Wake e Rem, ma non si presenta overlapping tra i vari box, il che rende NREM2 la classe per la quale l'onda Sigma presenta una maggiore importanza, in maniera piuttosto ben definita.

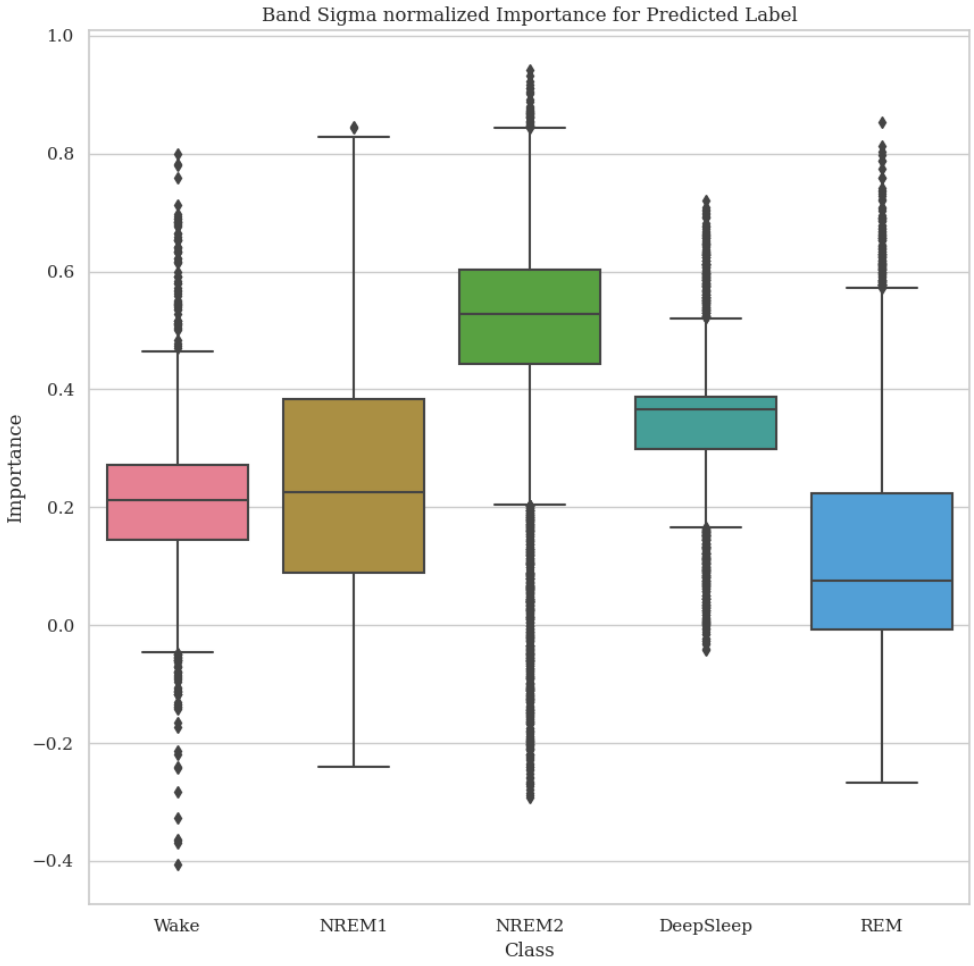

Figura 4.11: Importanza *pesata* dell'onda Sigma

4.2.6 Theta

L'onda Theta sembra essere quella per la quale si presentano più differenze tra lo stage 1 e lo stage 2 degli esperimenti. Notiamo come in figura 4.6 siano le classi NREM1 e REM ad essere quelle con i boxplot di valore più elevato, mentre ora, come è possibile osservare in figura 4.12, è NREM2 ad essere la classe per la quale Theta ha importanza più elevata, ma in ogni caso mantiene all'incirca gli stessi valori anche per tutte le altre classi, esclusa Wake. È curioso notare come, in un certo senso, questo plot sia simile a quello che mostravano le onde Alpha e Beta, ma "ribaltate", poiché mentre nei primi casi era la classe REM ad essere quella più vicino allo 0, in quest'ultimo caso, è la classe Wake a subire lo stesso trattamento.

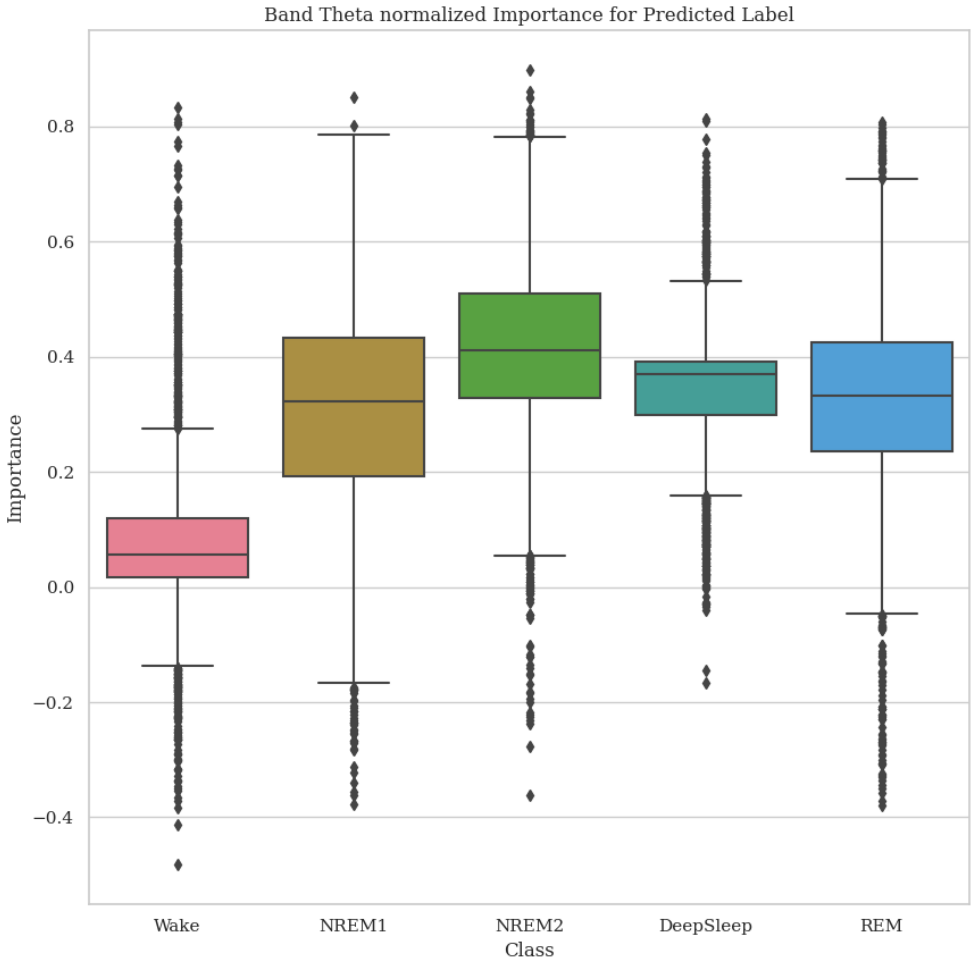

Figura 4.12: Importanza *pesata* dell'onda Theta

4.3 Stage 3

Lo stage 3 degli esperimenti ha introdotto il concetto di importanza nel tempo. Come discusso nel capitolo precedente, gli esperimenti condotti hanno portato alla creazione di 5 subplot composti da una heatmap che raffigura le bande, e da un input per poter capire in ogni istante di tempo, quale punto di input stiamo considerando, e che importanza hanno le bande in quel punto. Come già anticipato, negli esperimenti condotti è stata creata una funzione che permette di scegliere classi "target", interscambiabili tra classi scelte dall'utente, oppure le classi direttamente predette dalla rete. In base a questa scelta, le heatmap vengono prodotte andando a estrapolare dall'input la prima serie di 3 epoche che corrispondono alla classe target, e analizzando l'importanza delle bande per quelle 3 epoche. Di seguito andremo a

mostrare 5 immagini, una per ogni classe, che si riferiscono alle classi predette dalla rete neurale. Verranno mostrate solamente le immagini che si riferiscono all'epoca "centrale" (la seconda delle 3), per evitare problemi di dimensioni. Nonostante nel precedente capitolo abbiamo introdotto due approcci con i quali gli esperimenti sono stati condotti, verranno mostrate solo le immagini appartenenti alla categoria del dati normalizzati tra 0 e 1, con l'eccezione della classe DeepSleep, per la quale verranno mostrati entrambi gli approcci (puramente per rendere l'idea di entrambi, sfruttando una classe per la quale siamo certi esserci una banda che avrà un'importanza riconoscibile e non indifferente). Andremo inoltre a mostrare, per brevità come per lo stage 2, solo le immagini per le quali l'importanza è stata calcolata attraverso la *media pesata*, sempre tenendo a mente che i risultati sono coerenti e paragonabili a quelli ottenuti attraverso la *media semplice*.

4.3.1 Heatmap delle classi

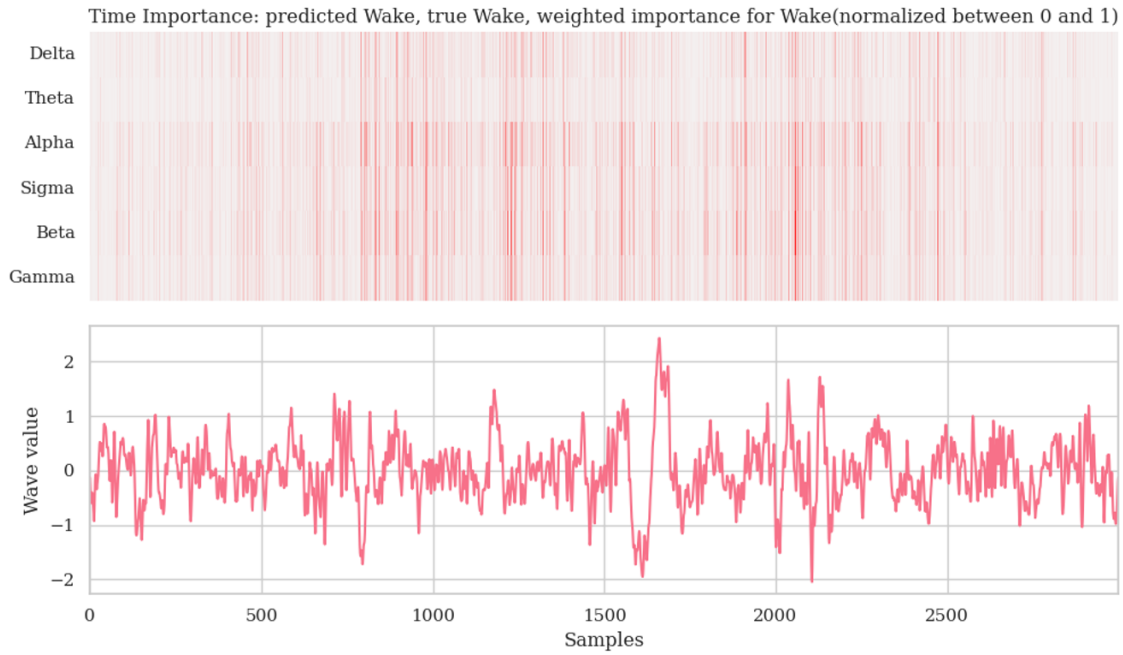

Figura 4.13: Importanza nel tempo per la classe Wake con valori normalizzati tra 0 e 1

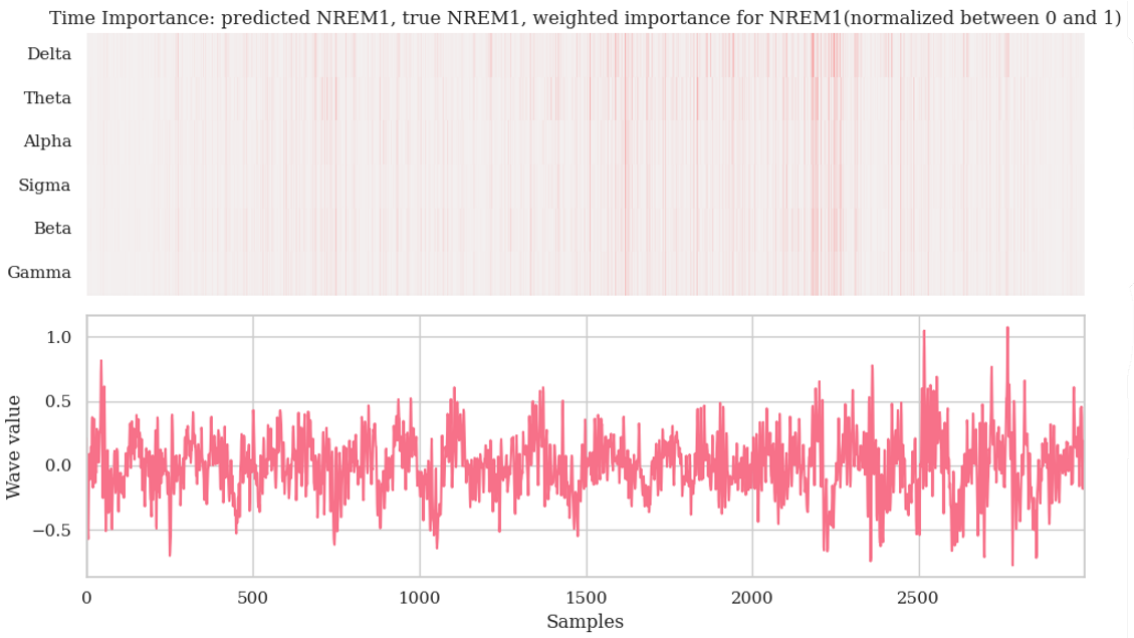

Figura 4.14: Importanza nel tempo per la classe NREM1 con valori normalizzati tra 0 e 1

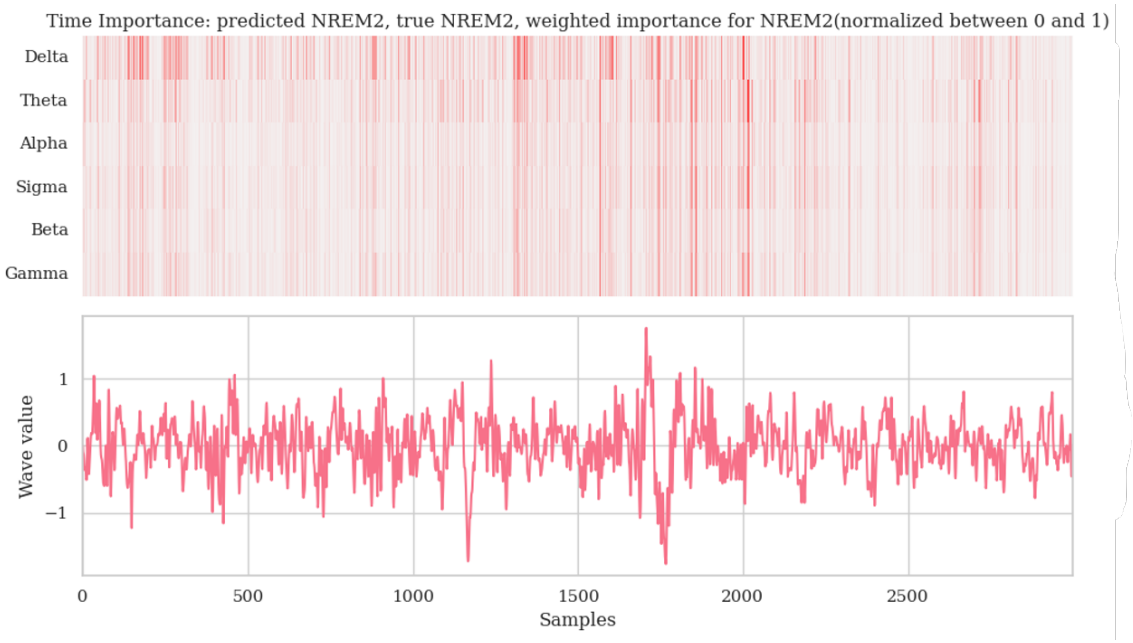

Figura 4.15: Importanza nel tempo per la classe NREM2 con valori normalizzati tra 0 e 1

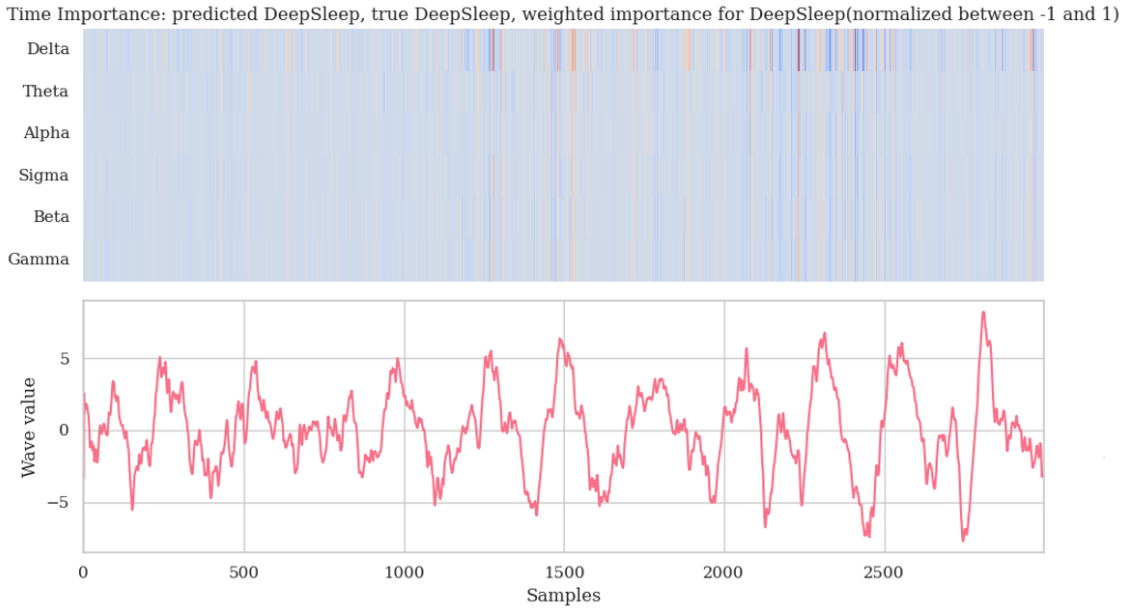

Figura 4.16: Importanza nel tempo per la classe DeepSleep con valori normalizzati tra -1 e 1

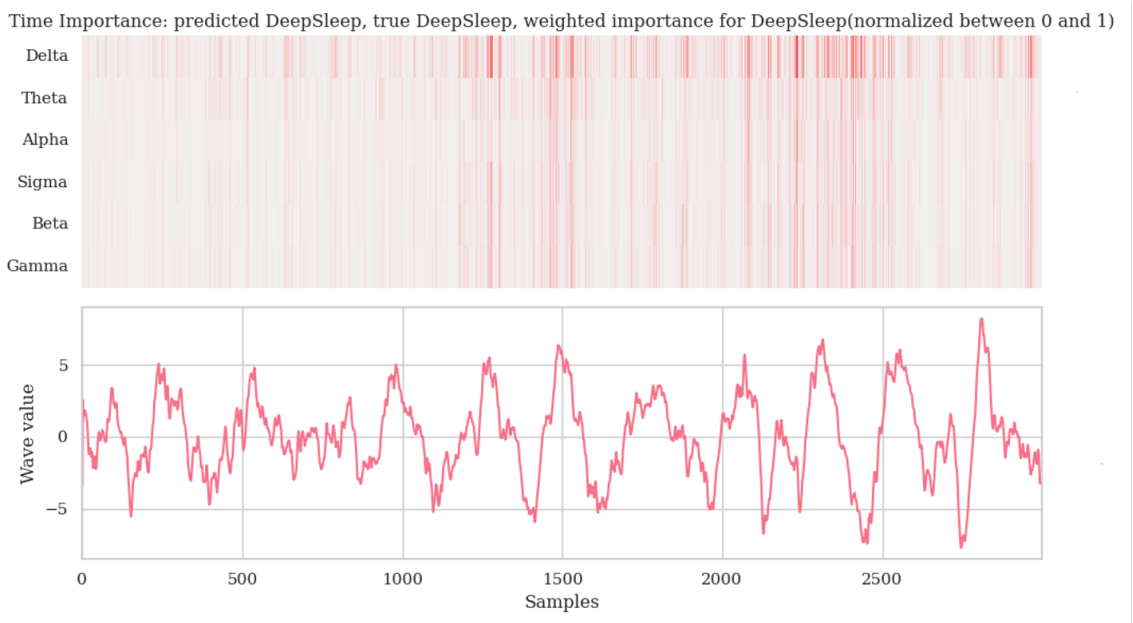

Figura 4.17: Importanza nel tempo per la classe DeepSleep con valori normalizzati tra 0 e 1

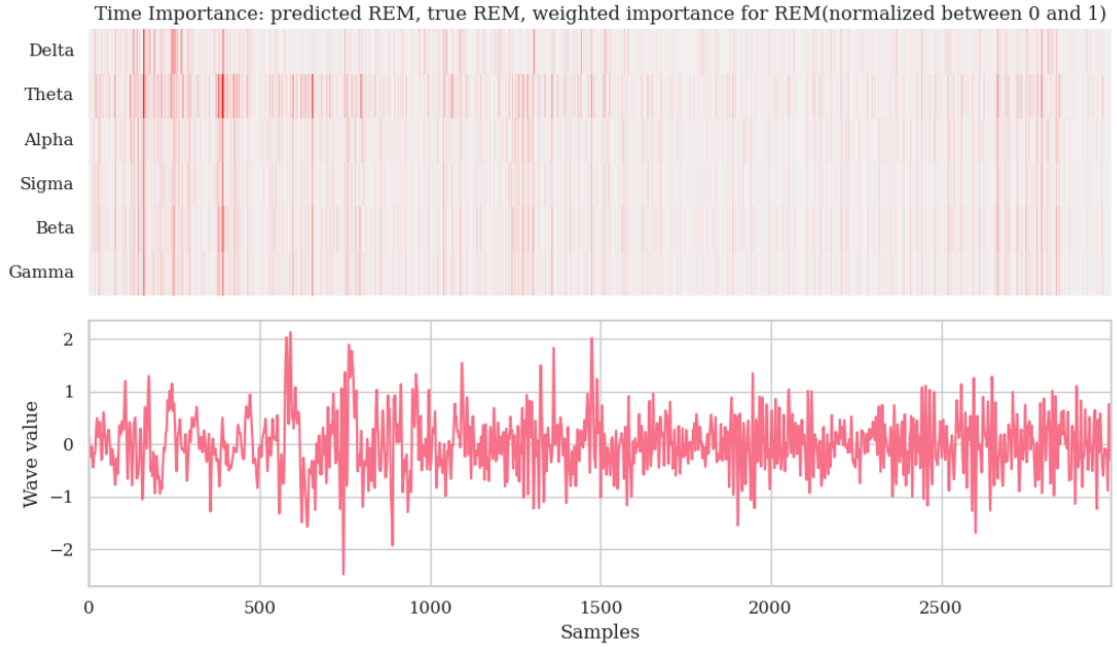

Figura 4.18: Importanza nel tempo per la classe REM con valori normalizzati tra 0 e 1

Capitolo 5

Conclusione

Nel corso di questa ricerca, abbiamo inaugurato una prospettiva innovativa nell'analisi del sonno, con l'intento di superare le limitazioni delle metodologie tradizionali attraverso l'integrazione di reti neurali profonde e tecniche di *eXplainable artificial intelligence* (XAI). La nostra ambizione era quella di sviluppare applicazioni capaci di interpretare automaticamente le complesse dinamiche delle onde cerebrali in relazione alle fasi del sonno, offrendo un'interpretazione trasparente e interpretabile delle previsioni del modello.

I risultati ottenuti hanno dimostrato la validità di questo approccio, permettendo di riconoscere e capire le modalità con cui la rete neurale prende le sue decisioni associando epoche di tempo di onde cerebrali alle diverse fasi del sonno rispetto alle metodologie tradizionali. La combinazione di una rete neurale profonda addestrata su dati polisonnografici, la creazione di una funzione in grado di calcolare l'importanza di un set di bande con e senza l'influenza del tempo e la generazione di immagini esplicative attraverso tecniche XAI ha contribuito a migliorare significativamente la comprensione delle relazioni complesse tra onde cerebrali e classi di sonno.

La trasparenza del modello, resa possibile dall'utilizzo di tecniche XAI, non solo aumenta la fiducia degli utenti nelle previsioni del sistema, ma apre anche nuove opportunità di collaborazione tra professionisti della salute e intelligenza artificiale nel contesto dell'analisi del sonno. Questa sinergia potrebbe portare a progressi significativi nella diagnosi precoce di disturbi del sonno e nell'adattamento personalizzato delle terapie.

Tuttavia, è importante sottolineare che questo lavoro rappresenta solo un punto di partenza. Futuri sviluppi potrebbero esplorare ulteriormente le potenzialità di modelli più complessi, l'espansione del dataset per una maggiore diversità, e la validazione su un ampio spettro di pazienti. L'analisi del sonno assistita da tecniche XAI promette di rimanere una direzione di ricerca cruciale, con impatti significativi nel migliorare la qualità della diagnosi e del trattamento per individui affetti da disturbi del sonno.

In definitiva, questa ricerca non solo offre una solida base per la comprensione delle metodologie impiegate, ma sottolinea anche il ruolo cruciale che l'integrazione

tra intelligenza artificiale e medicina potrebbe svolgere nell'ambito dell'analisi del sonno. Questa sinergia promette di essere una forza trainante nell'evoluzione della diagnosi e del trattamento dei disturbi del sonno, contribuendo a migliorare la qualità della vita di individui affetti da tali patologie. Concludiamo con la consapevolezza che la strada davanti a noi è ricca di opportunità e sfide, e attendiamo con ansia i futuri progressi che questa intersezione tra scienza del sonno e intelligenza artificiale potrà portare.

Bibliografia

[1] Lightning AI. Pytorch lightning website. https://lightning.ai/docs/pytorch/stable/, 2024. Acceduto nel febbraio 2024.

[2] Captum. Captum api reference - integrated gradients. https://captum.ai/api/integrated_gradients.html, 2024. Acceduto nel febbraio 2024.

[3] Stanislas Chambon, Mathieu N. Galtier, Pierrick J. Arnal, Gilles Wainrib, and Alexandre Gramfort. A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(4):758–769, 2018.

[4] NumPy Developers. Numpy documentation. https://numpy.org/doc/stable/index.html, 2022. Acceduto nel febbraio 2024.

[5] Guido Gagliardi. Physioex (physiological signal explainer). https://github.com/guidogagl/physioex, 2024. Acceduto nel febbraio 2024.

[6] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[7] The pandas development team. pandas-dev/pandas: Pandas. https://doi.org/10.5281/zenodo.3509134, February 2020.

[8] PsicoSocial.it. Delta, theta, alfa, beta e gamma: stati di coscienza e onde cerebrali. https://www.psicosocial.it/onde-cerebrali/, 2024. Acceduto nel febbraio 2024.

[9] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[10] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[11] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.

[12] Wikipedia. Standardizzazione (statistica). https://it.wikipedia.org/wiki/Standardizzazione_(statistica), Modificata l'ultima volta il 18/01/2024. Acceduto nel febbraio 2024.