



Università Politecnica delle Marche

Facoltà di Ingegneria

Corso di laurea magistrale in ingegneria informatica e dell'automazione

---

# Il deep learning per la medicina preventiva: previsione di una crisi epilettica tramite rete neurale per serie temporali

Deep learning for preventive medicine: prediction  
of an epileptic seizure by neural network for time  
series

---

**Relatore:** Prof. Aldo Franco Dragoni **Tesi di laurea di:** Pietro Rignanese

**Anno Accademico:** 2020-2021

## INDICE

1. INTRODUZIONE.....	2
2. STATO DELL'ARTE.....	5
2.1. ELETTROENCEFALOGRAFIA: IL SISTEMA 10-20 STANDARD.....	6
2.2. PREVISIONE DI UNA CRISI: EVOLUZIONE STORICA.....	9
2.3. IL DEEP LEARNING NELLA NEUROSCIENZA.....	12
3. LAVORO SVOLTO.....	17
3.1. INCEPTION TIME: UNA RETE PER LA CLASSIFICAZIONE DELLE SERIE TEMPORALI.....	18
3.2. DATASET.....	22
3.3. INCEPTION TIME: POSA IN OPERA DELLA RETE.....	31
3.4. RISULTATI OTTENUTI E CONFRONTO CON ALTRE TECNICHE DI DEEP LEARNING.....	36
4. CONCLUSIONI.....	43
4.1. SVILUPPI FUTURI.....	44
5. BIBLIOGRAFIA.....	45

## 1. INTRODUZIONE

Il termine **'crisi epilettica'** descrive una varietà di sintomi neurologici dovuti a una scarica elettrica anomala, sincronizzata e prolungata di cellule nervose della corteccia o del tronco cerebrale. La diagnosi di epilessia implica una tendenza a crisi epilettiche ripetute che si trova nello 0.5% della popolazione. Crisi epilettiche sono favorite da fattori che aumentano l'eccitabilità elettrica delle cellule nervose e abbassano la naturale soglia alla loro scarica spontanea: l'uso o la sospensione improvvisa di certi farmaci, droghe o alcool; febbre, deficit di sonno, alterazioni degli elettroliti, e infine fattori genetici e metabolici. Per la diagnosi di epilessia è necessaria un'accurata valutazione dei sintomi e della storia clinica, che deve possibilmente comprendere anche le osservazioni dettagliate da parte di terzi, in quanto l'alterazione o la perdita di coscienza spesso precludono una descrizione dei sintomi da parte del paziente stesso. **L'elettroencefalogramma (EEG)** rileva l'attività elettrica del cervello ed è un'analisi fondamentale nella diagnosi dell'epilessia, perché le alterazioni elettriche, spesso molto indicative, possono essere presenti anche in assenza dei sintomi. (1)

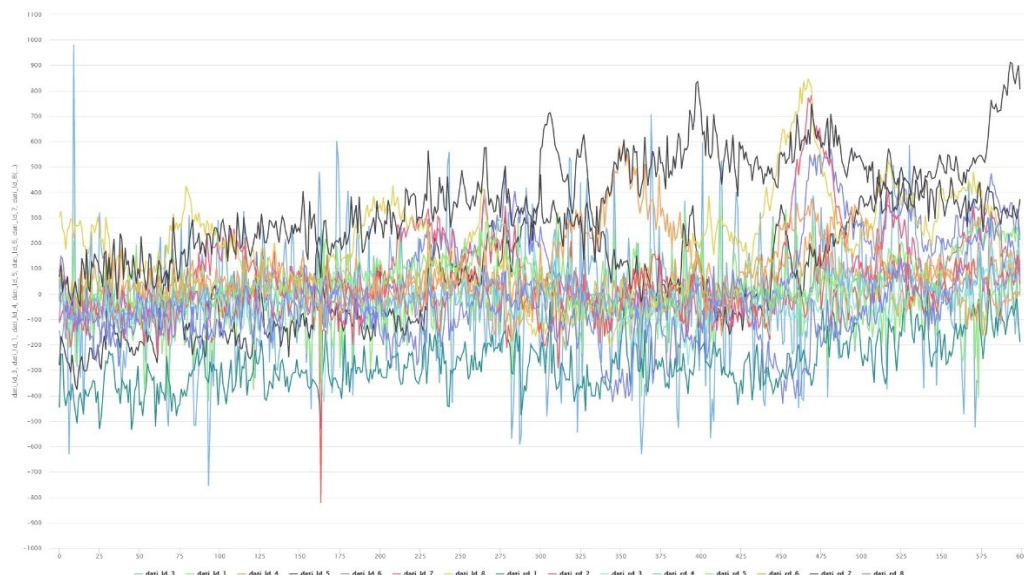
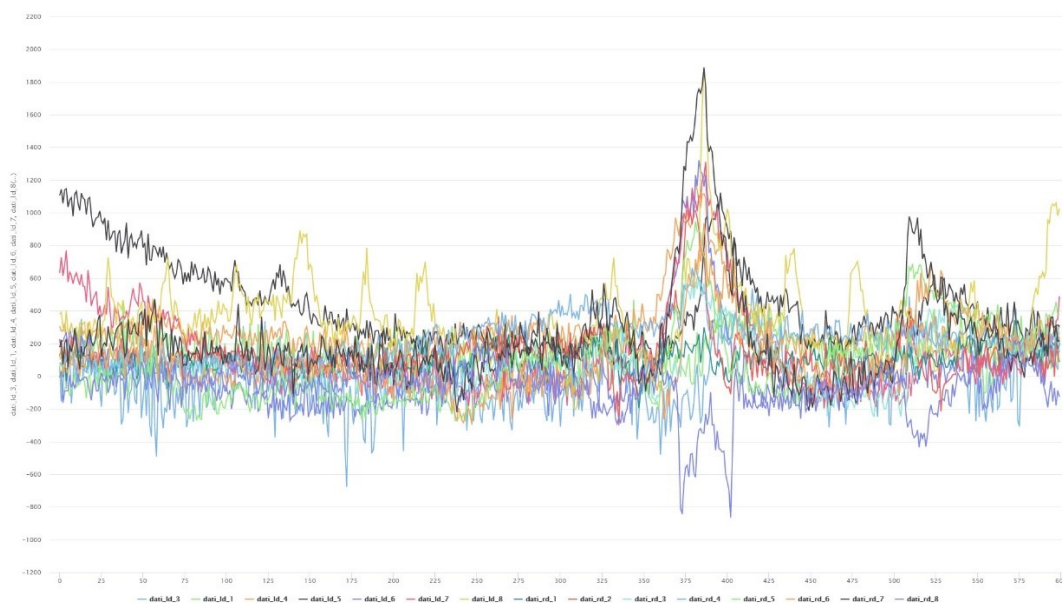


Figura 1 – Elettroencefalogramma a 15 elettrodi di un paziente epilettico in condizioni normali e lontano da attacchi

I soggetti affetti da epilessia trascorrono una vita relativamente normale quando non sono effettivamente colpiti da una crisi, tuttavia, l'incertezza su quando potrebbe

verificarsi un attacco epilettico pone per molti dei limiti nella vita di tutti i giorni. Inoltre, alcuni sintomi e alcune manifestazioni potrebbero sfociare in situazioni pericolose e possibilmente dannose per le persone soggette a questa malattia.

Con l'obiettivo principale di migliorare la vita dei pazienti epilettici (e anche quella delle persone vicino a loro) e grazie ai recenti sviluppi ottenuti sia dal punto di vista delle risorse tecnologiche sia delle conoscenze relative all'epilessia, negli ultimi anni molti ricercatori hanno promosso studi finalizzati alla realizzazione di un sistema per la predizione delle crisi epilettiche. La maggior parte di tali ricerche si basa sull'utilizzo di registrazioni elettroencefalografiche (EEG) e sulla detezione all'interno di essi dei tratti corrispondenti ai vari stati dell'epilessia.



*Figura 2 - Elettroencefalogramma a 15 elettrodi di un paziente epilettico in condizioni di pre-crisi*

Nella letteratura corrente, gli approcci più tradizionali ricorrono alla selezione ed estrazione di una o più features di interesse dagli EEG e ad algoritmi di classificazione che sulla base di esse possano distinguere le varie fasi ictaliche. Possiamo distinguere 3 tipologie di fasi ictaliche e sono le seguenti: fase pre-ictalica, pre-ictali o periodo pre-ictalico che definisce una fase transitoria prima di una crisi, tale fase può durare anche diversi minuti o ore e subito dopo tale fase subentra una crisi epilettica; fase inter-ictalica, inter-ictali o periodo inter-ictalico che definisce un periodo di quiete in

cui il paziente è lontano da una crisi o non avrà una crisi nel breve periodo; e infine, la fase ictalica che è la fase di crisi vera e propria in cui il paziente sta avendo una scarica e tale scarica può essere e avere natura di vario tipo.

Quindi, questi metodi si basano su molteplici ipotesi e richiedono sforzi per la selezione e l'estrazione delle features supposte più significative da differenti aree. Invece, più recentemente gli sforzi si sono concentrati sull'implementazione di **metodologie basate su reti neurali** in grado classificare i diversi stati ictalici partendo, però, da segnali EEG grezzi o sottoposti a poche pre-elaborazioni.

Le architetture di reti neurali utilizzate sono le **reti convoluzionali (CNN)** per le time series (cioè che lavorano con una sola dimensione, dette anche 1D-CNN), le **Long Short-Term Memory (LSTM)**, le **Bi-LSTM** (LSTM bidirezionali) e le **Multilayer Perceptron (MLP)**.

Questa tesi riguarda l'applicazione di una specifica rete neurale per le serie temporali, chiamata "Inception Time", usata per la prima volta nella classificazione degli stati pre-ictali e stati inter-ictali e un confronto con le principali reti utilizzate in ambito scientifico per la previsione delle crisi. A parità di dati è venuto fuori che la rete inception riesce a classificare con un'accuratezza dell'87% gli EEG pre-ictali e quelli inter-ictali mentre le altre reti non riescono a raggiungere tale accuratezza. La migliore tra le altre reti è la 1D-CNN che riesce a classificare i dati con un'accuratezza del 75%.

La struttura del documento sarà la seguente: si riporterà lo stato dell'arte e quindi quello che sono gli sviluppi delle attività di ricerca nella letteratura scientifica negli ultimi anni; verrà descritto il lavoro svolto facendo una piccola digressione su quella che è la rete Inception Time, verrà spiegato come è strutturato il dataset, la metodologia seguita e i risultati ottenuti; infine, le conclusioni e i possibili risvolti futuri.

## 2. STATO DELL'ARTE

La previsione precoce di una crisi in arrivo ha una grande influenza sulla vita dei pazienti epilettici e dei loro familiari. Nella letteratura scientifica vengono proposte molte tecniche per la previsione della crisi attraverso il deep learning e a che il machine learning. In questa tesi ci occuperemo di approfondire la parte inerente al deep learning e come queste tecniche siano differenti tra di loro, dandoci risultati differenti.

Tutte le tecniche utilizzate per la previsione, utilizzano i dati raccolti da un EEG a diversi campionamenti e diversi numeri di elettrodi. Prima di elencare le tecniche usate oggi vediamo nel dettaglio come vengono raccolti i dati sui pazienti e come e quanti sono gli elettrodi utilizzati per la raccolta.

## 2.1. ELETTOENCEFALOGRAFIA: IL SISTEMA 10-20 STANDARD

I biopotenziali elettroencefalografici, la cui rilevazione prende il nome di elettroencefalografia (EEG), sono una espressione dell'attività elettrica spontanea della corteccia cerebrale. Essi possono essere misurati attraverso elettrodi posti sullo scalpo del soggetto sotto forma di pattern temporali di potenziali elettrici. Questi ultimi sono classificabili in tre categorie:

- 1) *attività spontanea*;
- 2) *potenziali evocati*;
- 3) *eventi bioelettrici* provocati da singoli neuroni.

**L'attività spontanea** determina propriamente l'elettroencefalogramma ed è misurata sullo scalpo o, in casi sperimentali, direttamente sulla corteccia cerebrale. L'attività spontanea è continuamente presente nel cervello.

**I potenziali evocati** sono quelle componenti del segnale EEG che si presentano in risposta ad uno stimolo (visivo, tattile, uditivo, elettrico, ecc...). Questi segnali sono in genere di piccola ampiezza rispetto all'attività spontanea, divenendo confrontabili con il livello di rumore e risultando quindi non facilmente registrabili. Per indurre e analizzare i potenziali evocati si utilizzano in genere treni di impulsi di stimolazione, al fine di registrare segnali multipli e farne una media, così da aumentare il rapporto segnale-rumore.

**Gli eventi bioelettrici** provocati da singoli neuroni possono essere registrati utilizzando dei microelettrodi impiantati direttamente nelle cellule d'interesse. Lo studio di singole cellule risulta funzionale per la creazione di modelli delle rete cellulare che riflettano le reali proprietà del tessuto cerebrale (2).

Il sistema di registrazione dei segnali EEG, chiamato elettroencefalografo, è costituito da una unità di acquisizione dei segnali misurati sullo scalpo, da una unità di elaborazione dei segnali e da una unità di visualizzazione/memorizzazione dei dati. L'unità di acquisizione adotta elettrodi di misura tipicamente alloggiati su di una apposita cuffia, la quale viene adattata sulla testa del paziente. Gli elettrodi di superficie per EEG possono essere fissati alla cute con l'ausilio di collarini adesivi, cerotti o, come menzionato, di una cuffia apposita. Gli elettrodi possono essere di diversa tipologia, tra cui:

- elettrodi a coppetta, realizzati in stagno e in argento rivestito di cloruro d'argento (Ag/AgCl), del diametro di 10 mm;
- elettrodi in Ag/AgCl biadesivi monouso (2x3 cm e 1.5x2 cm);
- elettrodi ad anello in acciaio inossidabile.

Per eseguire registrazioni da un numero considerevole di punti dello scalpo, si ricorre spesso all'utilizzo di cuffie in tessuto elasticizzato (per garantire una opportuna aderenza) con elettrodi incorporati. L'esatta disposizione degli elettrodi sullo scalpo è regolata dallo standard internazionale 10/20.

Per quanto riguarda l'apposizione degli elettrodi, si ricorre a due derivazioni standard, a seconda degli specifici requisiti sperimentali: nella derivazione monopolare (o unipolare) un elettrodo viene posto in un sito attivo, mentre l'altro (elettrodo di riferimento) in un sito elettricamente neutro (ad es. la punta del naso, il lobo dell'orecchio, il mento). Con una lettura unipolare il potenziale di ogni elettrodo viene misurato rispetto all'elettrodo neutro o alla media di tutti gli elettrodi. Questo tipo di registrazione evidenzia pertanto il livello assoluto di attività elettrica sottostante il sito attivo. Nella derivazione bipolare, invece, entrambi gli elettrodi sono posti su siti attivi dell'area di interesse e il segnale rilevato corrisponde alla differenza che emerge tra le attività dei due siti.

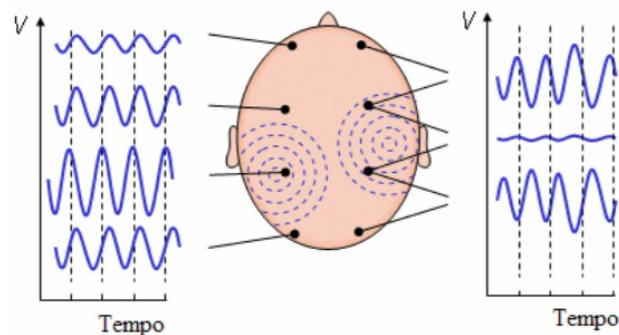


Figura 3 - Esempi di derivazione unipolari (sinistra) e bipolari (destra) (2)

Convenzionalmente, gli elettrodi sono posizionati sulla cute avendo cura di rispettare alcune corrispondenze tra aree corticali e sito di apposizione; a questo scopo si segue la dislocazione standardizzata nota come Sistema Internazionale Standard 10/20. Esso



prevede il posizionamento di 21 elettrodi sulla superficie dello scalpo, secondo la modalità convenzionale mostrata nella figura sottostante (3). Le posizioni sono determinate come di seguito descritto. Si considerano due punti di riferimento: il nasion, ovvero l'avvallamento tra naso e fronte a livello degli occhi, e l'inion, ovvero la protuberanza alla base del cranio sulla linea mediana della testa. Da questi punti viene misurata la lunghezza del cranio sui piani trasversale e mediano. Le posizioni degli elettrodi sono determinate dividendo queste lunghezze in intervalli del 10% e 20% del valore totale.

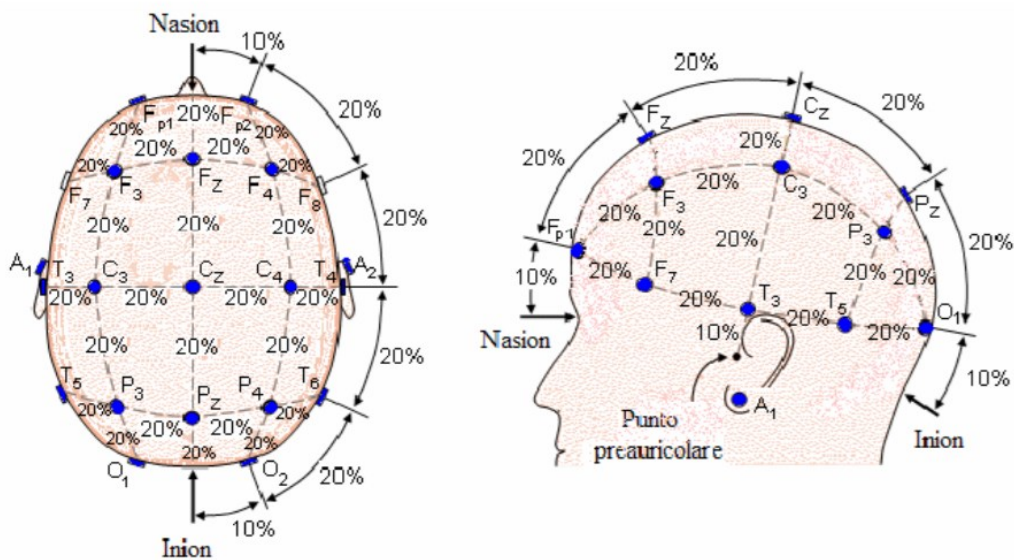


Figura 4 - Standard internazionale per la posizione degli elettrodi

I dati sono raccolti da questi elettrodi posti sulla cute del paziente e possono variare di numero anche, nel nostro caso (poi lo approfondiremo nei prossimi capitoli) gli elettrodi posizionati sul cuoio capelluto sono 15 e raccolgono i dati con un campionamento molto frequente (circa 1 dato ogni 0,0012 secondi).

## 2.2. PREVISIONE DI UNA CRISI: EVOLUZIONE STORICA

Negli anni '70 sono state condotte le prime ricerche sulla previsione delle crisi epilettiche utilizzando approcci lineari di estrazione delle caratteristiche (4). Nel frattempo negli anni '80, lo sviluppo di metodi non lineari ha aiutato i ricercatori a impiegare queste tecniche per l'estrazione delle caratteristiche a causa della natura non lineare dei segnali EEG (5) (6). Con il riconoscimento dei modelli EEG dell'epilessia, ovvero modelli preictali, ictali e interictali, l'uso dello stadio pre-ictale per il rilevamento di una crisi è stato applicato anche in questo decennio. Nel 1998 la previsione di crisi avveniva quasi 6 secondi prima dell'inizio dell'attacco e la ricerca sull'argomento era condotta da Salant (7) che è stato ulteriormente sviluppata e migliorata da Drogenlen nel 2003 (8). Tali approcci utilizzavano l'entropia di Kolmogorov-Sinai in cui consideriamo una sequenza  $x^{\tau_1}, x^{\tau_2}, x^{\tau_3}, \dots$ , con  $x^{\tau_\zeta} = x(j\tau)$ , ottenuta campionando una traiettoria generata da una condizione iniziale  $x^{\tau_0} = x(0)$  e si introduce una partizione  $A$  dividendo lo spazio delle fasi  $\Omega$  in elementi disgiunti  $a_1, a_2, \dots, a_M$  tali che l'unione di tutti gli elementi sia l'intero spazio delle fasi. In questo modo per ogni condizione iniziale si ha una successione di numeri interi  $i_1, i_2, \dots$  corrispondenti alle celle occupate dalla traiettoria al tempo  $\tau, 2\tau, \dots$ , ossia se  $x^{\tau_\zeta} \in a_k$  allora  $i_\zeta = k$ . In analogia con la teoria dell'informazione, si può definire l'entropia delle successioni  $S_n$  di lunghezza  $n$  (per es., se  $M=2$  allora per  $n=2$  si hanno le quattro possibili  $S_2$  [1;1]; [1;2]; [2;1] e [2;2]) (9)

Il primo seminario internazionale sulla previsione della crisi epilettica si è tenuto nel 2002 nel quale veniva proposto un set di dati di registrazioni di più giorni di EEG fornito da diversi centri per l'epilessia. Successivamente, sono stati condotti diversi studi su questo set di dati (10).

Nel 2003 Mormann ha notato il fatto che l'attivazione ipersincrona dei neuroni nel cervello è una causa di crisi epilettica e ha scoperto che la sincronizzazione di fase dei diversi canali EEG diminuisce prima dell'inizio delle crisi (11).

All'inizio del nostro secolo, studi basati su dati EEG estesi hanno sollevato dubbi sulle prestazioni di misure calcolate nel secolo precedente. Alcuni ricercatori hanno scoperto che i risultati di studi precedenti basati su un selezionato e inadeguato set di dati non può essere riprodotta su dati estesi. Si è deciso, quindi, di condurre concorsi sulla previsione degli attacchi epilettici in workshop internazionali condotti su tale argomento. Lo scopo di queste

competizioni era quello di standardizzare il confronto delle prestazioni di algoritmi addestrati su insiemi di dati comuni. La prima competizione di previsione delle crisi è stata tenuta in collaborazione con International Workshop on Seizure Pronostico 3 (IWSP3) nel 2007 mentre la seconda gara, condotta nel 2009, è stata in collaborazione con IWSP4. In entrambe le gare ai concorrenti erano state fornite le registrazioni di EEG continui da tre pazienti epilettici. Tuttavia, i risultati delle prestazioni degli algoritmi non erano affatto soddisfacenti. Fu tenuta un'altra sfida, nel 2014, tenuta dell'American Epilepsy Society, ha messo a disposizione una serie di EEG contenente 942 convulsioni registrate in più di 500 giorni e registrazioni EEG a lungo termine di cani con epilessia. A tutti i concorrenti è stata fornita la stessa durata di 10 minuti di dati di formazione e test. (12)

I concorsi, che abbiamo elencato poco fa, erano aperti a qualsiasi tipo algoritmo che calcoli le caratteristiche di base dei segnali EEG per le crisi e crei dei modelli di previsione o di apprendimento automatico addestrati sul dataset base fornito. In ogni caso ancora non sappiamo bene quali caratteristiche o gli algoritmi siano i migliori. Nei concorsi, i vari team hanno presentato algoritmi che erano troppo complicati dal punto di vista computazione. Quindi è difficile dire quale caratteristica o algoritmo di machine learning era da considerare il migliore.

Recenti lavori di Matias Maturana (13) hanno presentato una soluzione che potrebbe funzionare bene su tutti i pazienti. Tali lavori hanno identificato il rallentamento critico dei segnali cerebrali come un indicatore per la previsione della crisi.

Di seguito, una linea temporale sui lavori discusso poco fa.

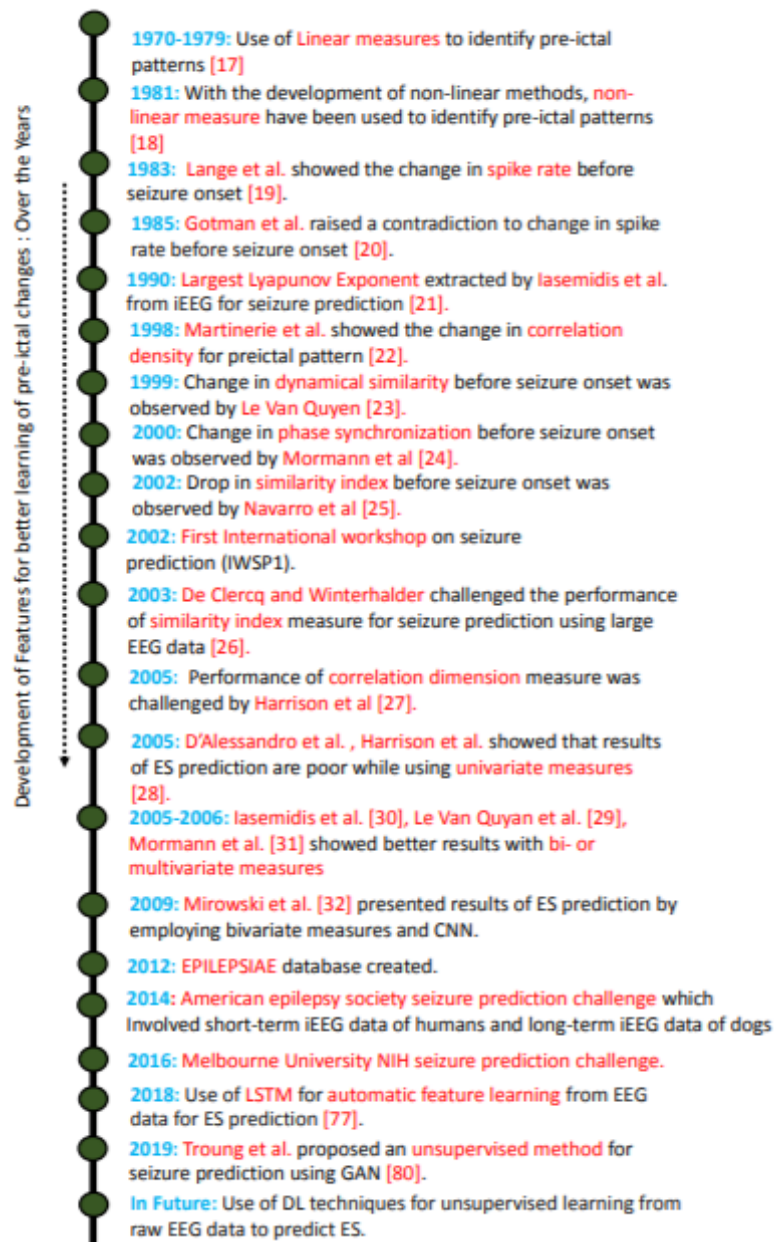


Figura 5 - Timeline sulla predizione delle crisi epilettiche (14)

### 2.3. IL DEEP LEARNING NELLA NEUROSCIENZA

Il deep learning studia modelli composti da molteplici strati di computazione con l'obiettivo di apprendere rappresentazioni dei dati a diversi livelli di astrazione. Le tecniche basate su deep learning hanno permesso di migliorare lo stato dell'arte in numerose applicazioni, dal riconoscimento automatico del parlato alla segmentazione e classificazione di oggetti nelle immagini, dal riconoscimento facciale al campo biomedicale (15). Questi strati multipli aiutano nell'estrazione di appropriate caratteristiche e sono elementi costitutivi di reti neurali profonde (DNN). Tra le più usate ci sono quelle convoluzionali, chiamate anche **Convolutional Neural Network** (CNN) e **Recurrent Neural Network** (RNN) (15). La struttura della CNN è ispirata a quella della connessione fra neuroni nel cervello e l'operazione di convoluzione è proprio come un filtro con i pesi per estrarre le caratteristiche da dati di input multidimensionali. Mentre vengono utilizzati gli RNN per trovare sequenze logiche nei dati di input. L'output di ciascun livello nascosto passato al livello successivo e riportato anche a sé. La differenza fondamentale tra l'architettura CNN e RNN è che le CNN considerano solo l'ingresso corrente mentre RNN considera l'ingresso corrente e anche l'ingresso precedente, cioè contiene una logica di memoria mediante un meccanismo di retrazione tra i livelli della rete. RNN ha prestazioni significativamente migliori sui dati delle serie temporali mentre le CNN sono particolarmente indicate per il riconoscimento di pattern in immagini. Le architetture deep learning sono state utilizzate in molti domini medici, ad esempio, nell'imaging clinico, genomica e proteomica, biologia computazionale e previsione della malattia (16).

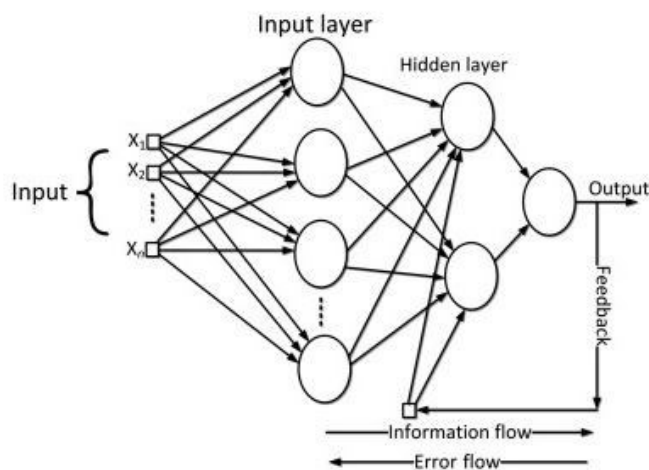


Figura 6 - Esempio di una rete RNN (Recurrent Neural Network)

Gli algoritmi di deep learning si sono rivelati adeguati nel rilevare intricati modelli in dati ad alta dimensionalità per la classificazione, in particolare nei dati EEG. La CNN è una rete neurale ampiamente utilizzata per allenarsi utilizzando i dati EEG perché può essere molto efficace per ridurre il rumore (17).

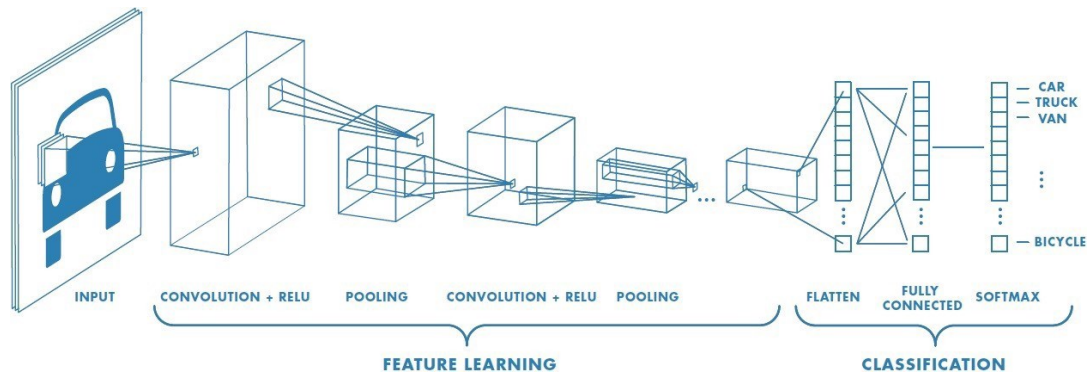


Figura 7 - Esempio di una CNN (Convolutional Neural Network)

Gli algoritmi di classificazione che usano il machine learning utilizzano vettori di caratteristiche, derivati dai tradizionali metodi di elaborazione del segnale per l'addestramento di classificatori e forniscono una buona accuratezza, ma da queste tecniche non può essere previsto un modello generalizzato (14). Gli algoritmi di classificazione basati sul machine learning utilizzano vettori di caratteristiche, derivati dai tradizionali metodi di elaborazione del segnale per l'allenamento e forniscono una buona accuratezza, ma da queste tecniche non può essere previsto un modello generalizzato. Per la previsione delle crisi attraverso un approccio machine learning, la scrittura di script richiede l'estrazione delle funzionalità, fase che richiede molto tempo. La presenza di rumore e gli artefatti nei dati rendono l'estrazione delle caratteristiche molto complessa da maneggiare. Quindi è un problema impegnativo produrre un sistema automatico generalizzato con prestazioni fedeli soprattutto anche quando sono disponibili campioni di addestramento limitati. D'altra parte, gli algoritmi di deep learning apprendono automaticamente le caratteristiche e mostrano risultati incoraggianti nella previsione di una crisi epilettica. Le caratteristiche apprese attraverso i modelli di deep learning sono più distintivi e robusti (18).

L'uso della CNN per la previsione di una crisi introduce un metodo che può essere applicato a tutti i pazienti con una preelaborazione minima dei dati EEG, come proposto da Troung (19). Hanno usato il dataset dell'ospedale di Friburgo e il dataset del cuoio capelluto CHBMIT ( Children's Hospital Boston ) per l'addestramento e test del modello CNN. Hanno

utilizzato la Short-Term Fourier Transform (STFT) per trasformare i dati EEG grezzi in una matrice bidimensionale. Questa immagine viene poi trasmessa alla CNN per l'apprendimento e la classificazione di stati di tipo pre-ictale e inter-ictale.

Pur seguendo una cross-validation, hanno raggiunto il 79,7% di sensibilità con 0,24 di false predizioni su EEG grezzo e una sensibilità dell'89,8% con 0,17 di false predizioni su dati standardizzati. Il tutto lavora su tempi molto vicini a una eventuale crisi epilettica (sotto ai 5 minuti).

Per l'uso clinico in real time per la previsione di una crisi, il tempo di previsione deve essere abbastanza lungo da permettere al paziente di uscire da uno stato di pericolo o adottare una serie di misure precauzionali. Il lavoro di Haider (16) ha eseguito meglio il lavoro precedente dando una sensibilità dell'87,8% e 0,142 di false predizioni con 10 minuti di anticipo da una crisi.

Motivato dal lavoro di Cock (20), Ramy Hussain (21) ha lavorato con una parte dei dati di questi pazienti. Hanno applicato una tecnica di **downsampling** per ridurre la dimensione dei dati di un fattore 4. Hanno spiegato che alcune caratteristiche non sono adatte per prevedere le crisi epilettiche perché i dati EEG non solo variano tra i pazienti ma varia anche per lo stesso paziente nel tempo. Hanno trasformato i dati EEG applicando STFT e alimentato la CNN. Per imparare le funzionalità locali che hanno usato Strati convoluzionali 1x1 e per l'apprendimento delle funzionalità astratte hanno usato convoluzioni più grandi. Hanno ottenuto l'87,85% di sensibilità.

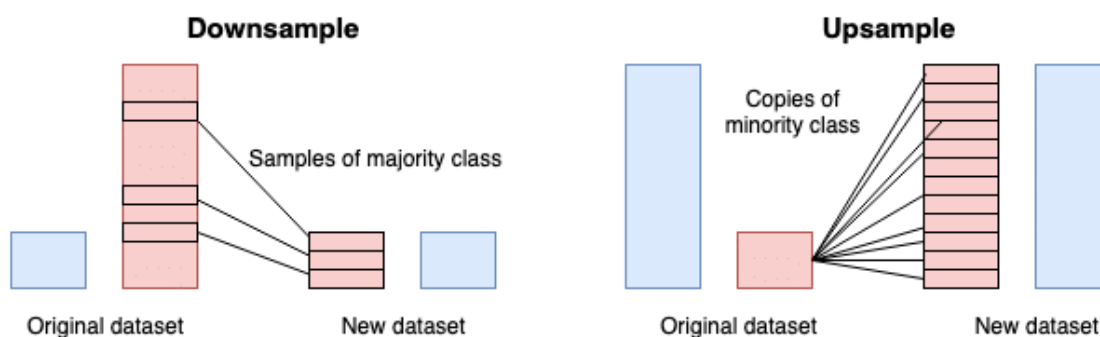


Figura 8 - Esempio grafico di un downsampling e un upsampling su un dataset

Hanno anche spiegato che le ragioni delle prestazioni limitate degli algoritmi di previsione delle crisi sono perdite di dati, mancata corrispondenza dei dati, distribuzione dello squilibrio e valori anomali dovuti a rumore o altro.

L'uso di RNN per la previsione delle crisi è presentato nell'articolo di Tsiouris (22) che ha utilizzato per la prima volta una Long-Short Term Memory (LSTM) per la previsione di una crisi epilettica. Hanno confrontato le prestazioni di diverse architetture di LSTM utilizzando vettori di segmenti EEG come input per LSTM. L'architettura LSTM-1 consisteva in un singolo strato con 32 unità di memoria, mentre il numero di unità di memoria è aumentato a 128 in architettura LSTM-2. Il numero di unità di memoria è portato a 256 per la LSTM-3 (aggiungendo due strati da 128). La performance di LSTM-3 è stata la migliore tra le tre architetture considerate. In media, LSTM-3 ha funzionato meglio con una sensibilità del 99,28% per un periodo pre-ictale di 15 minuti, 99,35% di recall per un periodo pre-ictale di 30 minuti, sensibilità del 99,63% con periodo pre-ictale di 60 minuti e sensibilità del 99,84% con 120 minuti di periodo pre-ictale.

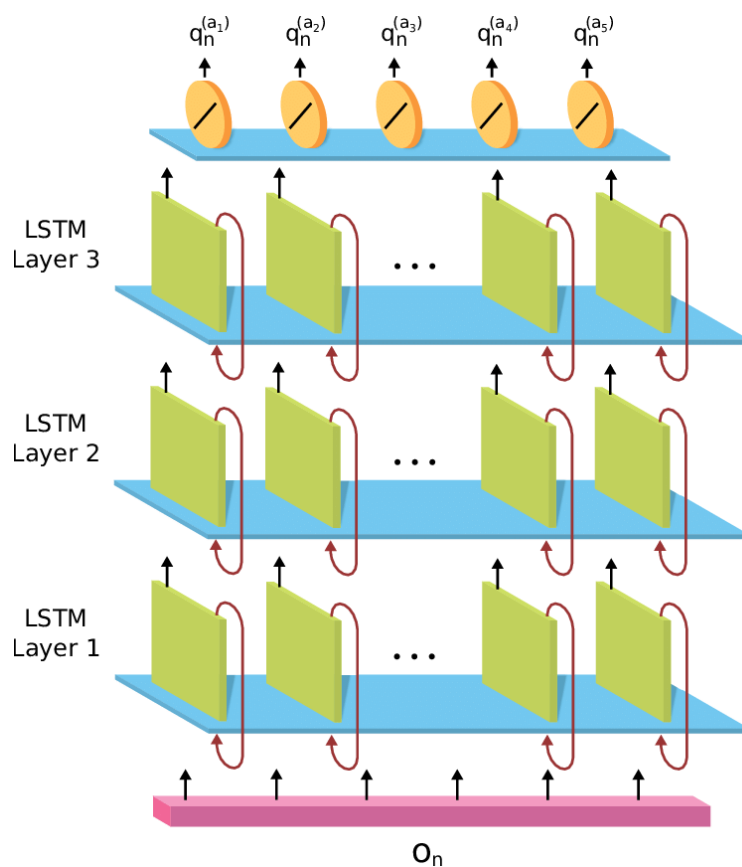


Figura 9 - Esempio di una rete LSTM



Tuttavia, ci sono altri approcci che hanno avuto un'accuratezza degna di nota come la Multilayer Perceptron (MLP) descritta nell'articolo di Hishan (23) dove mette a confronto una serie di reti neurali tra cui anche la CNN e la Bi-LSTM. Vengono creati anche degli ibridi tra le varie reti ma la MLP ha una risposta molto più elevata delle altre, a parità di dati, e riesce ad avere una sensibilità che supera il 99%.

<b>Tipo Rete</b>	<b>Referente</b>	<b>Anno</b>	<b>Sensibilità</b>
CNN	Haider	2017	87,8%
CNN	Ramy Hussian	2019	87,8%
LSTM-3	Tsiouris	2018	99,28%
Bi-LSTM	Hisham	2019	99,72%
MLP	Hisham	2019	99,6%

Il motivo per cui si utilizza la sensibilità, in tali ricerche, è il seguente:

sensibilità è la capacità di un test di identificare correttamente le persone che hanno una determinata malattia o disturbo. Ad esempio, un determinato test potrebbe aver dimostrato di essere sensibile al 90%. Cioè, se 100 persone note per avere una certa malattia vengono testate con quel metodo, il test identificherà correttamente 90 di quei 100 casi di malattia. Anche le altre 10 persone che sono state testate hanno la malattia, ma il test non riuscirà a rilevarla. Per quel 10%, la scoperta di un risultato "normale" è un risultato falso negativo fuorviante. La sensibilità di un test diventa particolarmente importante quando si cerca di escludere una malattia pericolosa.

Più un test è sensibile, meno risultati "falsi negativi" produce. Un risultato falso negativo non riesce a identificare gli stati patologici anche se sono presenti. (24)

### 3. LAVORO SVOLTO

Si potrebbe utilizzare lo stesso approccio studiato e migliorato, presente in letteratura scientifica, nel giro di anni, ma il nostro scopo porta a sperimentare una nuova rete da utilizzare in tale campo e confrontarla con le principali reti esposte nello stato dell'arte.

L'obiettivo principale è dare un contributo, se pur piccolo, alla scienza e alla ricerca per verificare che ci possano essere altre metodologie di approccio al problema o che quelle già presenti sono da considerare, per il momento, le migliori e che quindi sono da protrarre e utilizzarle in tale ambito.

Per eseguire la classificazione degli EEG sui pazienti epilettici ho deciso di utilizzare una rete molto moderna, strutturata a fine 2019, che si occupa di classificare, sulla base di una rete convoluzionale a una dimensione, le serie temporali.

Tale rete prende il nome di ***Inception Time*** e di seguito descriverò la sua architettura, i risultati ottenuti e i raffronti con altre reti neurali.

### 3.1. INCEPTION TIME: UNA RETE PER LA CLASSIFICAZIONE DELLE SERIE TEMPORALI

*“La **classificazione delle serie storiche (TSC)** è l'area del Machine Learning interessata all'apprendimento di come assegnare etichette alle serie storiche. Gli ultimi decenni di lavoro in questo settore hanno portato a progressi significativi nell'accuratezza dei classificatori, con lo stato dell'arte ora rappresentato dall'algoritmo **HIVE-COTE**. Sebbene estremamente accurato, HIVE-COTE è impossibile da utilizzare in molte applicazioni a causa della sua **altissima complessità del tempo di addestramento** in  $O(N^2 \cdot T^4)$  per un set di dati con  $N$  serie temporali di lunghezza  $T$ . Ad esempio, HIVE-COTE impiega più di 72.000s per imparare da un piccolo set di dati con  $N = 700$  serie temporali di breve durata  $T = 46$ .*

*Il deep learning, d'altra parte, ha ora ricevuto un'enorme attenzione grazie alla sua elevata scalabilità e accuratezza all'avanguardia nelle attività di computer vision e di elaborazione del linguaggio naturale. Il deep learning per TSC è stato esplorato solo di recente, con le prime architetture sviluppate solo negli ultimi 3 anni (tenendo presente il 2019 come data in cui viene scritto l'articolo a cui si fa riferimento). L'accuratezza del deep learning per TSC è stata portata a un livello competitivo, ma non ha ancora raggiunto il livello di HIVE-COTE. Questo è ciò che questo documento vuole raggiungere: sovraperformare l'accuratezza di HIVE-COTE insieme alla scalabilità. Facciamo un passo importante verso la ricerca della rete AlexNet per TSC presentando Inception Time, un insieme di modelli di rete neurale convoluzionale profonda (CNN), ispirato all'architettura Inception-v4.*

*I nostri esperimenti dimostrano che **Inception Time supera leggermente HIVE-COTE** con una vittoria / pareggio / perdita nell'archivio UCR del 40 / 6 / 39. Non solo Inception Time è **più preciso**, ma è molto **più veloce**: Inception Time impara dallo stesso set di dati con 700 serie temporali in 2.300s, ma può anche imparare da un set di dati con serie temporali 8M in 13 ore, una quantità di dati che è completamente fuori portata di HIVE-COTE.” (25)*

Sono queste le parole con cui si apre l'abstract dell'articolo inerente alla Inception Time. Tale rete risulta avere degli ottimi risultati, confrontandola con una delle principali reti utilizzate per la classificazione delle serie temporali. Questo ci fa capire come questa rete abbia delle potenzialità anche per lo studio degli EEG, essendo che quest'ultimi siano a loro volta delle serie temporali. Detto ciò, andiamo a vedere l'architettura di tale rete.

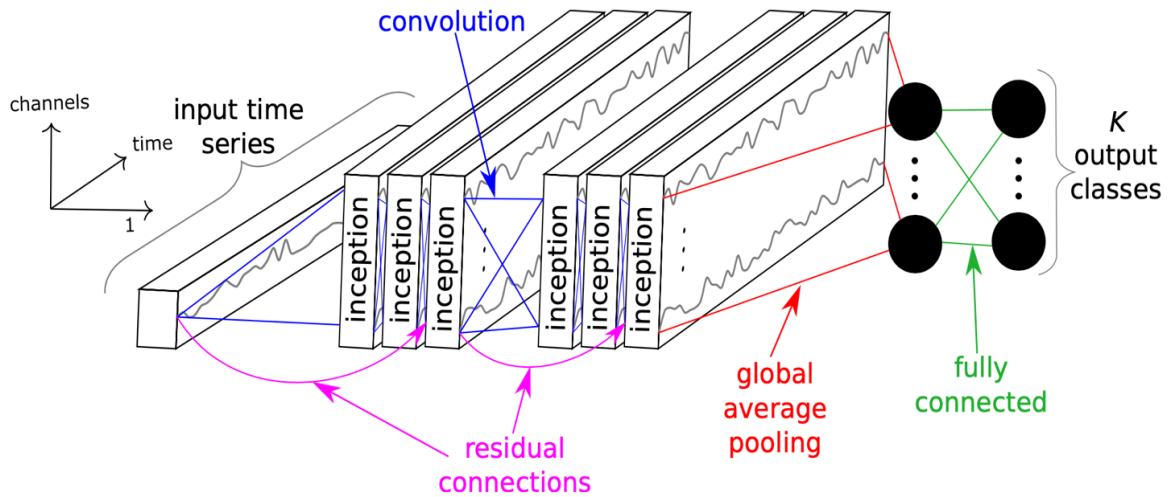


Figura 10 - Rete Inception Time

La **composizione di un classificatore di rete Inception** contiene **due diversi blocchi residuali**, a differenza di ResNet, che è composto da tre.

Per la rete Inception, **ogni blocco è composto da tre moduli Inception** piuttosto che da livelli tradizionalmente completamente convoluzionali.

*I blocchi residuali sono delle variazioni dei classici blocchi convoluzione-attivazione, in questi ultimi, abbiamo un input che entra in un layer e quest'ultimo genera, attraverso una funzione di attivazione, un output che alimenta il blocco successivo. Un blocco residuale oltre a costituire l'input del layer successivo, costituisce anche l'input di altri layer più avanti nella catena dei vari blocchi che la compongono grazie alle 'skip connection'.*

L'input di ciascun blocco residuale viene trasferito tramite una 'shortcut linear connection' da aggiungere all'ingresso del blocco successivo, mitigando così il problema del 'vanishing gradient' o **scomparsa del gradiente** consentendo un flusso diretto del gradiente. (26)

**Il problema della scomparsa del gradiente** è un fenomeno che crea difficoltà nell'addestramento delle CNN tramite **backpropagation** mediante **discesa stocastica del gradiente**. In tale metodo, ogni parametro del modello riceve a ogni iterazione un aggiornamento proporzionale alla derivata parziale della funzione di costo rispetto al parametro stesso. Una delle principali cause è la presenza di funzioni di attivazione non lineari classiche (tanh) che hanno gradiente a valori nell'intervallo (0,1). Poiché nell'algoritmo di retropropagazione i gradienti ai vari livelli vengono moltiplicati tramite la 'regola della

catena' il prodotto di  $n$  numeri in  $(0,1)$  decresce esponenzialmente rispetto alla profondità  $n$  della rete. Quando invece il gradiente delle funzioni di attivazione può assumere valori elevati, un problema analogo che può manifestarsi è quello dell'esplosione del gradiente.

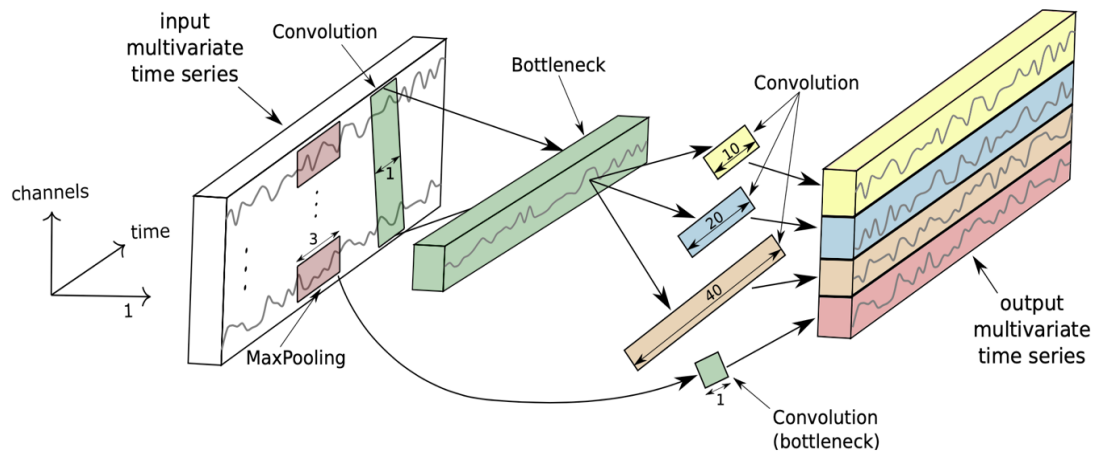


Figura 11 - Modulo Inception nel dettaglio

Come da immagine, consideriamo un input come un MTS (Multivariant Time Series) con dimensioni  $M$ . Il primo componente principale del modulo Inception è chiamato livello **'bottleneck'**. Questo strato esegue **un'operazione di scorrimento sull'input attraverso  $m$  filtri di lunghezza 1 con un passo uguale a 1**. Questo trasformerà le serie temporali da un MTS con dimensioni  $M$  a un MTS con dimensioni  $m \ll M$ , **riducendo così significativamente la dimensionalità delle serie temporali nonché la complessità del modello, mitigando problemi di overfitting per piccoli set di dati**.

Si noti che ai fini della visualizzazione, la Figura 11 illustra uno strato **'bottleneck'** con  $m = 1$ . Infine, dovremmo menzionare che questa tecnica di **'bottleneck'** consente alla rete Inception di avere filtri molto più lunghi di ResNet (quasi dieci volte) con approssimativamente lo stesso numero dei parametri da apprendere, poiché, senza lo strato **'bottleneck'** non avremmo una riduzione di dimensionalità e i filtri avrebbero dimensioni pari a  $M$ . Grazie a questo filtro, la dimensione degli input viene tagliata ed è possibile lavorare su grandi moli di dati e con velocità di apprendimento elevate rispetto alla concorrenza.

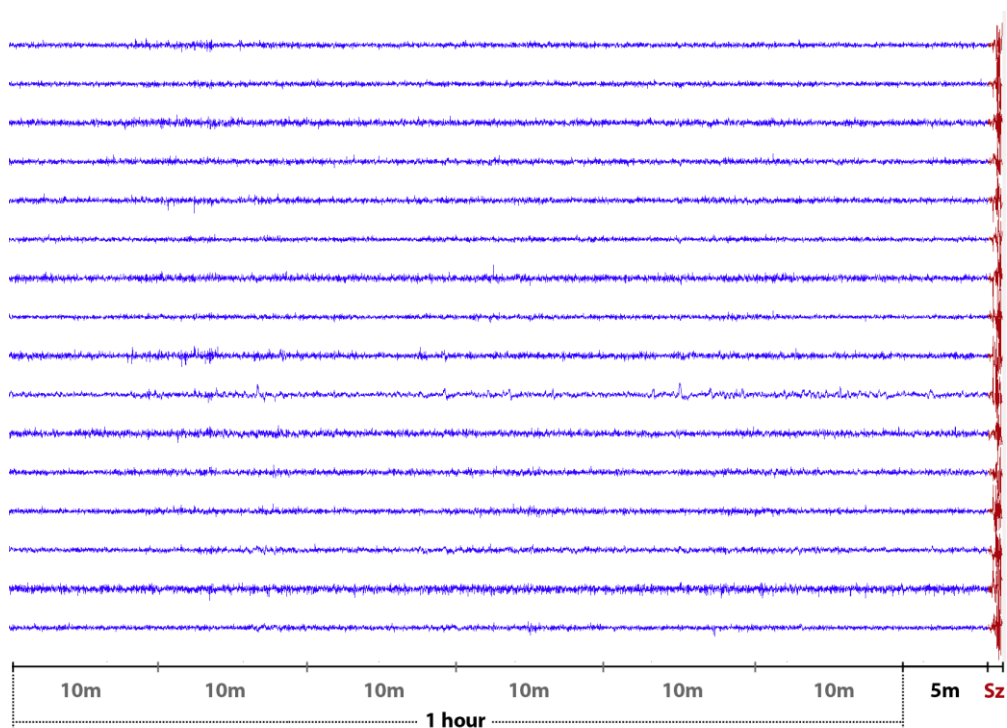
Il **secondo componente principale** del modulo Inception è lo **scorrimento simultaneo di più filtri con diverse lunghezze sulla stessa serie temporale di input**. Ad esempio, nella Figura 11, tre diverse convoluzioni con lunghezza  $\in \{10, 20, 40\}$  sono applicate all'ingresso MTS, che è tecnicamente l'uscita del livello *'bottleneck'*. Inoltre, per introdurre i vantaggi di avere un modello invariante a piccole perturbazioni, introduciamo un MaxPooling parallelo, seguito da un livello *'bottleneck'* per ridurre la dimensionalità.

L'output della finestra scorrevole MaxPooling viene calcolato prendendo il valore massimo in questa finestra di serie storiche. Infine, gli output di ogni convoluzione parallela indipendente e del MaxPooling vengono concatenati per formare l'output MTS del modulo Inception corrente.

Impilando più moduli Inception e allenando i pesi (valori dei filtri) tramite **backpropagation**, la rete è in grado di estrarre e riconoscere pattern non proprio facili da scovare grazie all'utilizzo di filtri di varie lunghezze. Per completezza, specifichiamo il numero esatto di filtri per il nostro modulo Inception proposto: 3 set di filtri ciascuno con 32 filtri di lunghezza  $l \in \{10, 20, 40\}$  con MaxPooling aggiunto al mix, facendo così il numero totale di filtri per strato uguale a  $32 \times 4 = 128 = M$  - la dimensionalità dell'uscita MTS.

### 3.2. DATASET

È stato preso un dataset utilizzato per una competizione tenuta, dall'American Epilepsy Society Seizure Prediction Challenge, nel 2014 e messo a disposizione della piattaforma Kaggle (27). Il dataset mette a disposizione delle sequenze di EEG, di un'ora l'uno, fino a 5 minuti prima di una crisi, di 3 pazienti in condizioni normali (inter-ictal) e di pre-ictal (pre-ictal), campionati ogni 0,0002 secondi, il tutto raccolto in una sequenza di 6 file (EEG di 10 minuti a file) con estensione *.MAT*. I file a loro volta sono etichettati in pre-ictal o inter-ictal, in modo da dare delle informazioni sul tipo di EEG raccolto e poterlo organizzare al meglio.



*Figura 12 – Esempio di EEG di un paziente su 1 ora,*

*suddiviso in 15 starti dovuti ai 15 elettrodi fino a 5 minuti prima di una crisi epilettica*

Come possiamo vedere dalla *Figura 12* l'EEG in questione è suddiviso in 15 linee di raccolta dati (dovuto ai 15 elettrodi con cui sono stati catturati i potenziali elettrici) e ognuna di queste sequenze viene impilata e raccolta ogni 10 minuti in un unico file.

3 pazienti con:

- *Paziente 1: pre-ictal 3 ore e 8 ore di inter-ictal*
- *Paziente 2: pre-ictal 7 ore e 10 ore di inter-ictal*
- *Paziente 3: pre-ictal 10 ore e 20 ore di inter-ictal*

Per poter organizzare al meglio il dataset e utilizzarlo per fare classificazione con la rete Inception ho previsto di utilizzare l'ultimo minuto della sequenza in modo che la finestra di controllo fosse di un solo minuto ma comunque distante da una eventuale crisi, in modo da poter intervenire prontamente qualora si dovesse verificarne una. Ricordiamo che il dataset è raccolto in un lasso temporale di un'ora fino a 5 minuti prima che si verifichi una crisi, quindi, prendendo come finestra di controllo l'ultimo minuto della sequenza, stiamo considerando un EEG, di 1 minuto, 5 minuti prima di una eventuale crisi epilettica. In questo modo diamo largo anticipo ai pazienti o ai loro cari di provvedere alla somministrazione di una pillola qualora fosse necessario.

Come dicevamo, i file hanno una particolare estensione per essere aperti su Matlab, ma nel nostro caso abbiamo provveduto diversamente e abbiamo utilizzato una specifica libreria per Python (chiamata Scipy.io), per aprirli, leggerli e scrivere la parte fondamentale un file con estensione CSV.

Il comando per leggere un file con estensione MAT è il seguente:

```
data = scipy.io.loadmat(nome_file + ".mat")
```

La pagina viene letta tutta e la variabile *data* prende l'intera informazione delle pagine come fosse un array di elementi. Infatti, la pagina stessa è caratterizzata da un array di array di elementi che compongono le informazioni dell'EEG.

Le informazioni contenute nella pagina sono le seguenti:

- **data**: una matrice di informazioni in cui ogni riga rappresenta l'elettrodo e la colonna la sequenza temporale delle informazioni raccolte dall'elettrodo stesso;



- **data\_length\_sec**: la durata in secondi di ogni sequenza, nel nostro caso per ogni file avremo 600 secondi come informazione (perché ogni file raccoglie 10 minuti di informazioni);
- **sampling\_frequency**: il numero dei dati che vengono raccolti ogni secondo (5000 nel nostro caso);
- **channel**: la lista degli elettrodi che compongono l'EEG;
- **sequence**: l'indice della sequenza di file considerata, per esempio se stiamo trattando il file 5 di 6 della sequenza di una persona, troveremo come dato il numero 5 e così via.

Come possiamo notare le informazioni contenute nel file sono tante e non tutte sono essenziali al nostro scopo. Le informazioni necessarie sono contenute nella matrice **data**, che contiene la sequenza temporale dell'elettroencefalogramma raccolto ogni 0,0002 secondi.

Per poter estrarre quel particolare valore presente nella matrice dobbiamo utilizzare quanto segue all'interno di un ciclo for:

```
data[segment][0][0][0][counter][num]
```

- I blocchi in cui è presente '0' sono semplicemente dei valori da dare all'array per poter andare a leggere solamente la matrice interessata;
- La variabile *segment* è una stringa e si riferisce alla sequenza di file a cui stiamo lavorando, nel nostro caso lavoreremo su tutte le sequenze finali di un EEG di un'ora e quindi tale variabile avrà sempre il valore *preictal\_segment\_6*, perché appunto rappresenterà il file 6 di 6 della sequenza;
- La variabile *counter* si riferisce al numero dell'elettrodo di riferimento a cui si sta attingendo il valore;
- L'ultimo blocco si riferisce al valore ennesimo della sequenza raccolta dall'elettrodo preso in questione e la variabile *num*, quindi, avrà un numero che crescerà ad ogni iterazione, ma nel nostro caso partirà da un punto ben preciso della sequenza, un minuto prima della fine, per poter prendere l'ultimo minuto della sequenza stessa; quindi, il codice sarà implementato in questo modo:

```
data[segment][0][0][0][counter][len(data[segment][0][0][0][counter]) - num]
```

L'ultimo blocco è strutturato dalla lunghezza del blocco informativo stesso in cui li viene sottratto un numero che crescerà di 1 ad ogni iterazione e tale *num* sarà settato a 2.700.000, questo perché la sequenza intera di 10 minuti è lunga 3 milioni e noi abbiamo bisogno delle ultime 300 mila informazioni (300mila corrisponde a 1 minuto esatto). Da quest'ultime informazioni andremo a raccogliere solo una parte del contenuto informativo, cercando di avere un campionamento più ridotto e più precisamente andremo a raccogliere i dati una volta ogni 500 step, in modo da abbassare il numero del campionamento alzando il numero del tempo di raccolta ogni 0,1 secondi al posto di 0,0002 secondi. Tutto ciò determinerà una raccolta dati meno frequente e più leggera senza perdere il contenuto informativo.

Motivazione di raccolta dati ogni 500 step:

$$3000000 = 10 \text{ min}$$

$$\frac{3000000}{10} = 300 \text{ mila /min} \rightarrow \text{Dati raccolti ogni minuto}$$

$$\frac{300000}{60} = 5000/\text{sec} \rightarrow \text{Dati raccolti al secondo (frequenza di campionamento)}$$

$$\frac{1\text{sec}}{5000} = 0,0002 \text{ sec} \rightarrow \text{Numero del tempo in cui viene raccolto un dato}$$

$$3000000 - 300000 = 2700000 \rightarrow \text{Sequenza partenza di raccolta}$$

Abbassando i campioni da 300 mila a 600, in modo da alleggerire il contenuto informativo risultando anche più facile da manipolare, possiamo determinare il numero esatto di secondi in cui verrà raccolto ogni dato:

$$\frac{600}{60} = 10/\text{sec} \rightarrow \text{Dati raccolti al secondo}$$

$$\frac{1}{10} = 0,1 \text{ sec} \rightarrow \text{Numero del tempo in cui viene raccolto un dato}$$

Ovviamente per raccogliere 600 campioni in una sequenza di 300 mila, il calcolo è il seguente:

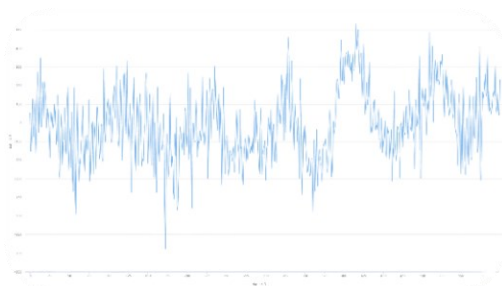
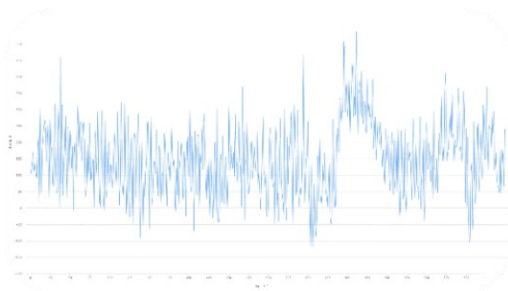
$$\frac{300000}{600} = 500 \rightarrow \text{Numero degli step da saltare nella sequenza per campionare ogni 0,1 secondo}$$

La variabile *num* crescerà di 500 ad ogni iterazione, in modo da avere il campionamento sperato e avere una sequenza dati di un minuto su 600 campioni.

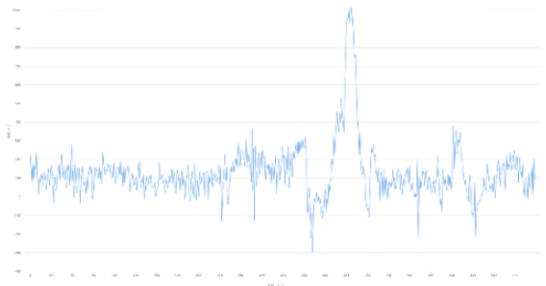
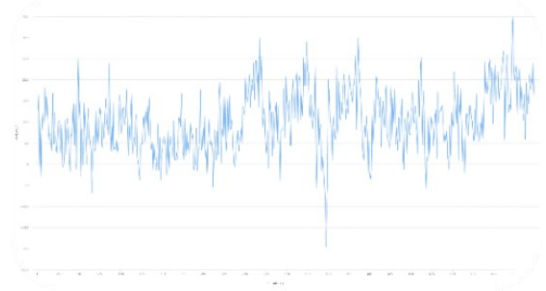
Una volta trovato il giusto compromesso si è potuto scrivere le informazioni raccolte in file con estensione CSV.

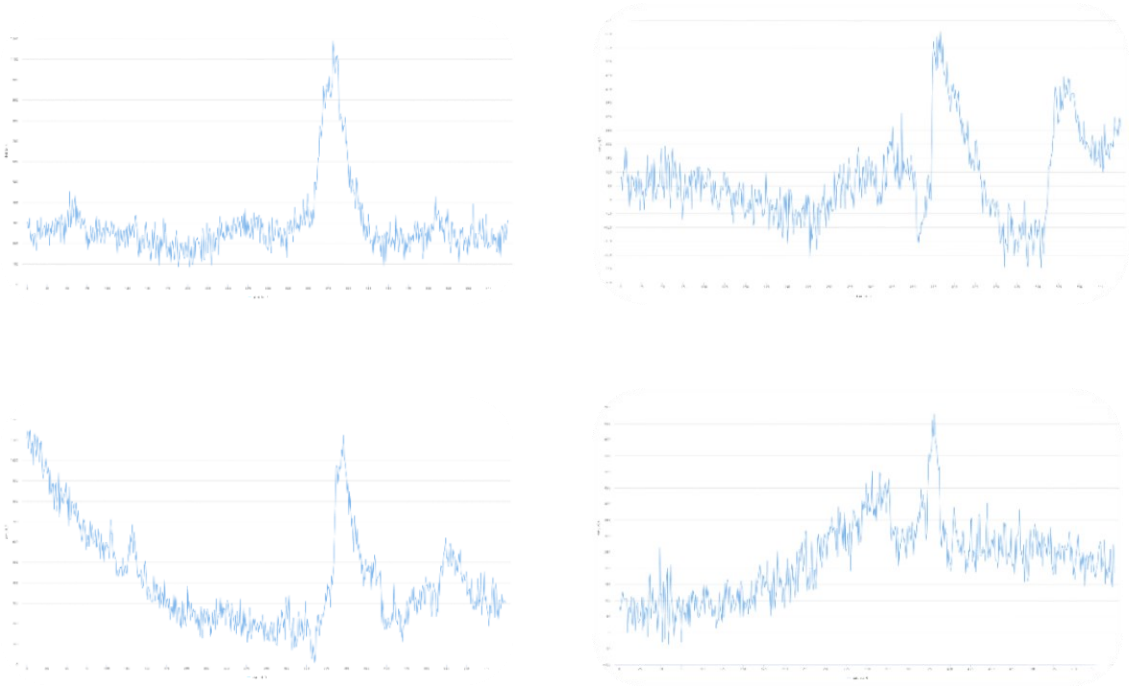
Fatto ciò, avremo una serie di file in formato CSV che possiamo tranquillamente plottare con Rapidminer o Python. Ecco alcuni degli EEG raccolti su un paziente in una finestra di 1 minuto con campionamento 0,1 secondi, suddivisi per emisfero destro e sinistro:

***Elettrodi Emisfero Sinistro***

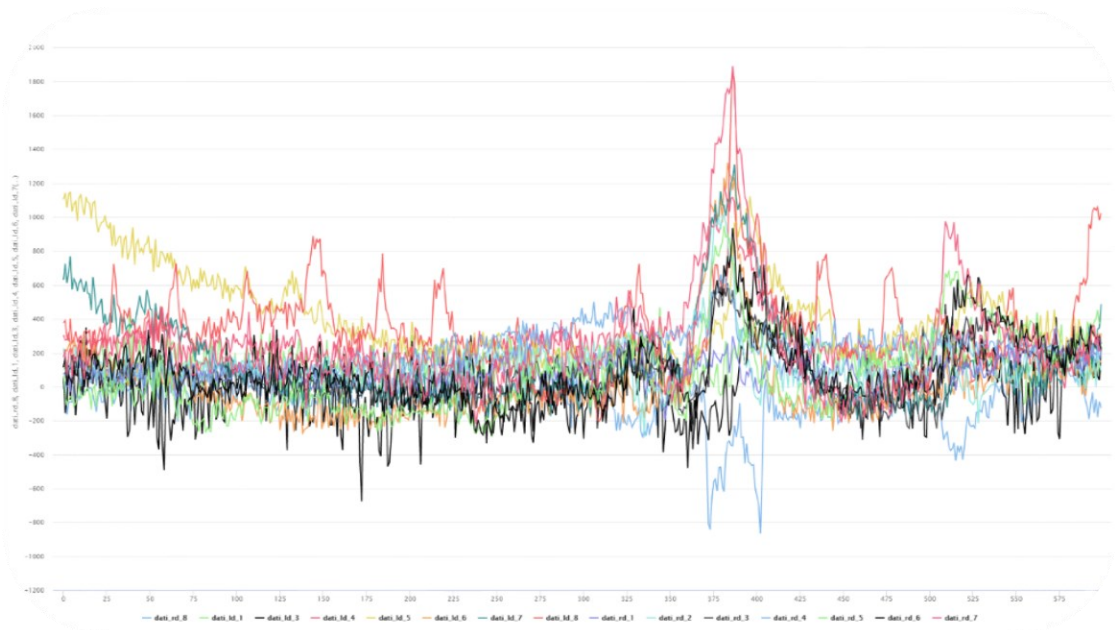


***Elettrodi Emisfero Destro***





Andando a posizionare tutti gli EEG singoli, in un unico grafico otterremmo questo:



Il dataset raccolto è suddiviso in 3 pazienti in cui ognuno di essi ha una serie di EEG di pre-ictal e inter-ictal. In totale abbiamo una sequenza di 70 dati che dovranno essere ripartiti in TRAIN SET e TEST SET attraverso una 5-Cross-Validation. La percentuale di suddivisione del dataset è la seguente:

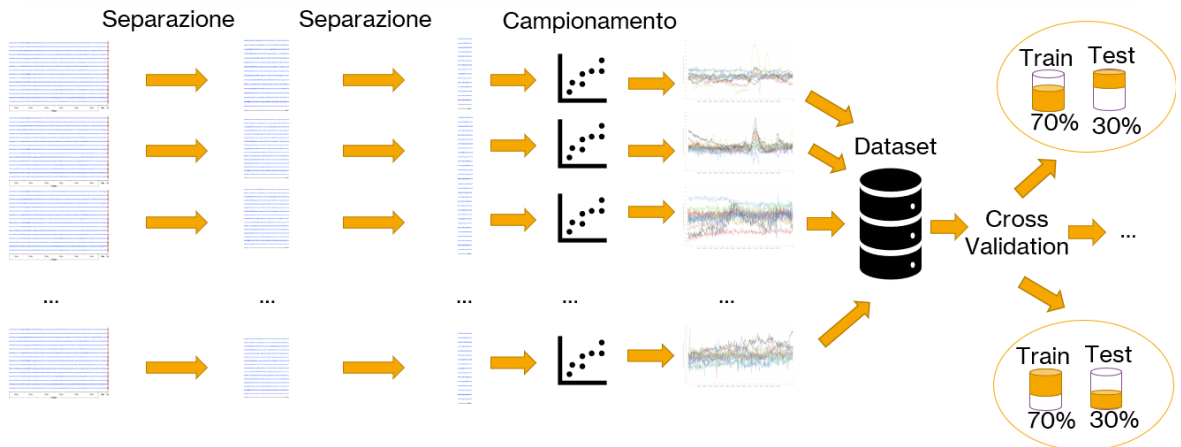


Figura 13 - Preprocessing Dataset con Cross Validation in breve

Una volta raccolti i vari EEG ho sviluppato un algoritmo, sempre in Python per organizzare i dati in modo tale da poter essere letti dalla rete Inception e quindi addestrare un modello sulla base dei dati raccolti fino adesso.

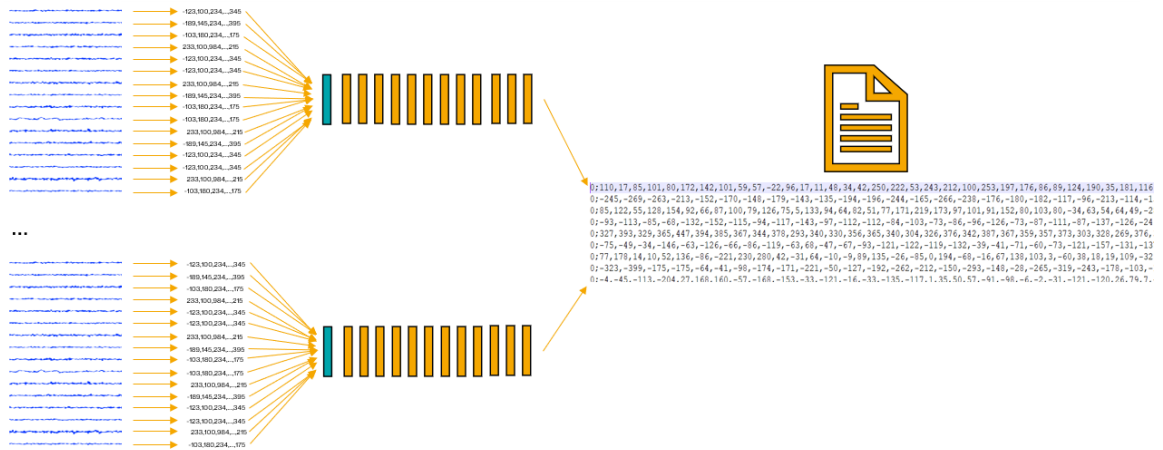


Figura 14 - Schema di creazione per il file di lettura per rete Inception

Come possiamo vedere dalla *Figura 14* si è realizzato un algoritmo per la lettura dei singoli elettrodi e riposizionamento di questi in un file di lettura per la Inception. Il file che la rete Inception dovrà andare a leggere dovrà essere strutturato come segue:

- Le righe rappresentano una sequenza di dati con la seguente struttura:

- [Label ; Seq\_1 ; Seq\_2 ; Seq\_3 ; ... ; Seq\_15]

Il primo campo sarà assegnato al tipo di classificazione a cui fa riferimento l'intera sequenza di informazioni, la seconda rappresenta la sequenza di dati dell'elettrodo 1, la terza rappresenta la sequenza di dati dell'elettrodo 2 e così via fino ad arrivare all'ultimo blocco che farà riferimento alla sequenza dell'elettrodo 15. Le varie sequenze sono separate dal punto e virgola e all'interno di ogni sequenza la distinzione avviene attraverso delle semplici virgole come separatori.

Ecco un esempio:

```

0;110,17,85,101,80,172,142,101,59,57,-22,96,17,11,48,34,42,250,222,53,243,212,100,253,197,176,86,89,124,190,35,181,116
0;-245,-269,-263,-213,-152,-170,-148,-179,-143,-135,-194,-196,-244,-165,-266,-238,-176,-180,-182,-117,-96,-213,-114,-1
0;85,122,55,128,154,92,66,87,100,79,126,75,5,133,94,64,82,51,77,171,219,173,97,101,91,152,80,103,80,-34,63,54,64,49,-2
0;-93,-113,-85,-68,-132,-152,-115,-94,-117,-143,-97,-112,-112,-84,-103,-73,-86,-96,-126,-73,-87,-111,-87,-137,-126,-24
0;327,393,329,365,447,394,385,367,344,378,293,340,330,356,365,340,304,326,376,342,387,367,359,357,373,303,328,269,376,
0;-75,-49,-34,-146,-63,-126,-66,-86,-119,-63,68,-47,-67,-93,-121,-122,-119,-132,-39,-41,-71,-60,-73,-121,-157,-131,-13
0;77,178,14,10,52,136,-86,-221,230,280,42,-31,64,-10,-9,89,135,-26,-85,0,194,-68,-16,67,138,103,3,-60,38,18,19,109,-32
0;-323,-399,-175,-175,-64,-41,-98,-174,-171,-221,-50,-127,-192,-262,-212,-150,-293,-148,-28,-265,-319,-243,-178,-103,-
0;-4,-45,-113,-204,27,168,160,-57,-168,-153,-33,-121,-16,-33,-135,-117,1,35,50,57,-91,-98,-6,-2,-31,-121,-120,26,79,7.

```

Questo approccio va fatto sia per i dati di train che per quelli di test, tenendo presente sempre la suddivisione di 70% per il train e 30% per il test in base al numero di dati di partenza, oppure può essere organizzato una sola volta e poi andranno separati a mano i valori di train da quelli di test.

In questo modo il dataset risulta essere organizzato e pronto per essere utilizzato per l'addestramento e il test.

La sequenza raccontata va fatta su ogni dataset raccolto, perché attraverso la Cross-Validation abbiamo strutturato, riorganizzando gli stessi dati in modo differente, 5 dataset.

N.B: Il file con i dati di train e test per la rete Inception Time non deve avere estensioni.

### 3.3. INCEPTION TIME: POSA IN OPERA DELLA RETE

Abbiamo già parlato molto dettagliatamente della rete Inception Time, ma quello che faremo in questo capitolo sarà qualcosa di diverso: affronteremo l'uso della rete su Google Colab, scaricando la rete già pronta all'uso sulla repository ufficiale del team di ricerca che ci ha lavorato. (28)

I passaggi sono pochi e richiede poca esperienza di lavoro. Per prima cosa scarichiamo la repository attraverso i comandi di git oppure possiamo scaricarla direttamente dal sito e cliccare sul bottone “Code” e successivamente su “Download ZIP”. Una volta scaricato il file lo si potrà decomprimere in una cartella dedicata. All'interno della cartella controllate se è presente una directory chiamata ‘dataset’, se non è presente createla e al suo interno create un'altra directory chiamata a ‘archives’. Una volta creata questa cartella dovete creare un'altra cartella al suo interno chiamata ‘TSC’ e al suo interno dobbiamo metterci il dataset che abbiamo creato nel capitolo precedente.

Per inserire il dataset nella directory TSC dobbiamo seguire altri pochi passaggi, il primo è che dobbiamo rinominare la directory con lo stesso nome dei file, e al suo interno devono essere presenti i due file di train e di test. Ad esempio: i due file devono avere la prima parte del nome uguale e tutto deve essere scritto in UPPER CASE (cioè in maiuscolo): EEGEPILESSY\_TRAIN e EEGEPILESSY\_TEST. La directory che li conterrà dovrà essere rinominata, obbligatoriamente, EEGEPILESSY, perché corrisponde alla parte del nome che hanno in comune.

Fatto ciò, possiamo mettere la nostra rete Inception (quasi pronta all'uso) su Google Drive, semplicemente copiando l'intera cartella in cui contiene tutti i file della nostra rete estratta e importarli all'interno del nostro spazio di archiviazione. Una volta che l'intera cartella, con tutti i file, è stata copiata con successo, possiamo passare alla configurazione di alcuni campi attraverso Google Colab.

Google Colab è un ambiente che permette di scrivere, eseguire e memorizzare notebook Jupyter in cloud. Basta avere un account Google o account di Gmail per potervi direttamente accedere. Una volta fatto l'accesso potrete aprire un nuovo foglio di lavoro o blocco note (possiamo chiamarlo in entrambi i modi). Inizialmente si presenterà spoglio e saremo noi ad aggiungere degli script che ci permetteranno prima di collegarci



alla directory su Google Drive e poi a settare dei parametri alla nostra rete Inception per far sì che lavori con il nostro Dataset.

```
%tensorflow_version 1.x
%pip install h5py == 2.8.0
import h5py
import tensorflow as tf
device_name = tf.test.gpu_device_name()
if device_name != '/device:GPU:0':
    raise SystemError('GPU device not found')
print('Found GPU at: {}'.format(device_name))
```

Ogni cella, contenente il codice sorgente, può essere opportunamente eseguito come sezione separata dagli altri; quindi, possiamo già far partire questa parte di codice attraverso il bottone di RUN presente al blocco di script.

L'output che produrrà colab sarà il seguente:

```
TensorFlow 1.x selected.
Collecting h5py==2.8.0
  Downloading h5py-2.8.0-cp37-cp37m-manylinux1_x86_64.whl (2.8 MB)
    |████████████████████████████████████████| 2.8 MB 9.2 MB/s
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-
packages (from h5py==2.8.0) (1.15.0)
Requirement already satisfied: numpy>=1.7 in
/usr/local/lib/python3.7/dist-packages (from h5py==2.8.0) (1.19.5)
Installing collected packages: h5py
  Attempting uninstall: h5py
    Found existing installation: h5py 3.1.0
    Uninstalling h5py-3.1.0:
      Successfully uninstalled h5py-3.1.0
Successfully installed h5py-2.8.0
Found GPU at: /device:GPU:0
```

Fatto ciò, possiamo passare a collegarci al nostro Google Drive attraverso il seguente comando:

```
from google.colab import drive
drive.mount('/content/drive', force_remount=True)
```

Ci verrà mostrato un link che ci porterà ad una pagina di sicurezza per acconsentire al controllo del nostro spazio di archiviazione tramite colab. Una volta che avremo

acconsentito ci verrà fornito un codice di sicurezza che dovremo copiare e incollare nell'apposito spazio che ci viene mostrato su colab e se tutto è andato a buon fine avremo il seguente messaggio:

```
Mounted at /content/drive
```

Adesso dobbiamo utilizzare un comando per spostarci all'interno della cartella che contiene la rete Inception. Il comando dovrà essere pressappoco così (dipende da dove sarà presente sul drive):

```
%cd /content/drive/MyDrive/Rete/InceptionTime-master
```

Manca poco per far partire la nostra rete, per ultima cosa dobbiamo modificare due file:

- **Utils.py:**
  - Dobbiamo modificare il codice della funzione *readucr* in questo modo:

```
def readucr(filename, delimiter=';'):
    data = np.loadtxt(filename, dtype="str", delimi
ter=delimiter)
    Y = data[:, 0]
    X = data[:, 1:]
    Xn = []

    """
    for i in range(0, len(X)):
        X1 = np.fromstring(X[i][0], dtype=float, sep=
',')
        X2 = np.fromstring(X[i][1], dtype=float, sep=
',')
        X3 = np.fromstring(X[i][2], dtype=float, sep=
',')

        Xn = np.append(np.asarray(Xn),X1)
        Xn = np.append(np.asarray(Xn),X2)
        Xn = np.append(np.asarray(Xn),X3)
    """
    for i in range(0, len(X)):
        X1 = np.fromstring(X[i][0], dtype=float, sep=
',')
        X2 = np.fromstring(X[i][1], dtype=float, sep=
',')
```

```

X3 = np.fromstring(X[i][2], dtype=float, sep=
',')
X4 = np.fromstring(X[i][3], dtype=float, sep=
',')
X5 = np.fromstring(X[i][4], dtype=float, sep=
',')
X6 = np.fromstring(X[i][5], dtype=float, sep=
',')
X7 = np.fromstring(X[i][6], dtype=float, sep=
',')
X8 = np.fromstring(X[i][7], dtype=float, sep=
',')
X9 = np.fromstring(X[i][8], dtype=float, sep=
',')
X10 = np.fromstring(X[i][9], dtype=float, sep=
',')
X11 = np.fromstring(X[i][10], dtype=float, se
p=',')
X12 = np.fromstring(X[i][11], dtype=float, se
p=',')
X13 = np.fromstring(X[i][12], dtype=float, se
p=',')
X14 = np.fromstring(X[i][13], dtype=float, se
p=',')
X15 = np.fromstring(X[i][14], dtype=float, se
p=',')

raw = []
support = []
for j in range(0, len(X1)):
    support.append(X1[j])
    support.append(X2[j])
    support.append(X3[j])
    support.append(X4[j])
    support.append(X5[j])
    support.append(X6[j])
    support.append(X7[j])
    support.append(X8[j])
    support.append(X9[j])
    support.append(X10[j])
    support.append(X11[j])
    support.append(X12[j])
    support.append(X13[j])
    support.append(X14[j])
    support.append(X15[j])

    raw.append(np.asarray(support))
    support = []

raw = np.asarray(raw)

```

```

Xn.append(raw)

raw = []

Xn = np.asarray(Xn)

Y = Y.astype(np.int)

return Xn, Y

```

La modifica va a catalogare i 15 elettrodi presenti nel file di lettura.

- **Constants.py:**

- Alla prima riga bisogna aggiungere il nome della directory da utilizzare alla lista di dataset.

Esempio: `UNIVARIATE_DATASET_NAMES = ['50words', 'EEG', ...`

E poi bisogna aggiungere tale nome anche alla raccolta successiva:

`UNIVARIATE_DATASET_NAMES = ['EEG']`

Siamo pronti per far partire la rete Inception con il nostro dataset:

```
!python main.py InceptionTime
```

Una volta che l'inception sarà partita, comincerà ad addestrare un modello sul dataset di train e poi, una volta completato il modello, lo testerà con il dataset di test riportando i dati nella directory: `'dataset/sersults/nne'`.

### 3.4. RISULTATI OTTENUTI E CONFRONTO CON ALTRE TECNICHE DI DEEP LEARNING

Una volta che l'Inception Time finisce di addestrare un modello, questo viene testato con i dati di test forniti e i risultati vengono generati in una cartella specifica che è possibile consultare direttamente su Google Drive.

Lavorando con gli EEG dei pazienti epilettici e classificando i vari dai in due grandi categorie, il risultato finale che abbiamo ottenuto è il seguente:

Cross-Validation	Precision	Accuracy	Recall
1	89%	86%	86%
2	95%	95%	95%
3	96%	95%	95%
4	93%	95%	97%
5	81%	86%	91%
<b>MEDIA</b>	91%	91%	93%

Ricordiamo come vengono calcolate le varie metriche:

Classi Effettive	Classi Previste	
	Vero Positivo (TP)	Falso Negativo (FN)
	Falso Positivo (FP)	Vero Negativo (TN)

$Precision = \frac{TP}{TP+FP}$  → Una precision alta indica un modello che sbaglia raramente perché i falsi positivi sono pochi.

$Recall = \frac{TP}{TP+FN}$  → la recall indica la sensibilità del modello ed è, infatti, il rapporto tra le classi predette in modo corretto e sul totale dei casi in cui si verifica effettivamente. Quando un modello è sensibile per una classe, la prevede ogni volta che si verifica.

$Accuracy = \frac{TP+TN}{TP+FN+FP+TN}$  → L'accuratezza non è altro che il numero di campioni classificati in modo corretto su tutti i campioni totali.

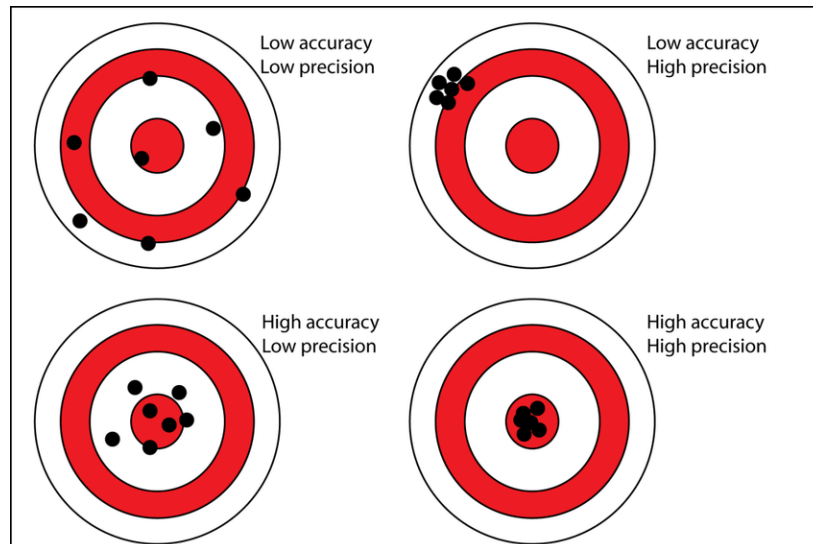


Figura 15 - Accuracy e Precision

Il nostro modello ha una buona accuratezza, precisione e recall, questo significa che riesce a classificare le due tipologie di classi in modo corretto. L'Inception Time può essere utilizzata per la classificazione degli EEG, che precedono fino a 5 minuti l'arrivo di una scarica, per la previsione dei possibili attacchi epilettici con una buona accuratezza.

Fatto ciò, si sono confrontati i risultati ottenuti con altre tecniche di deep learning, quindi, ho strutturato delle reti neurali e le ho fatte addestrare dei modelli sugli stessi dati della Inception, e su quelli sono andato a fare i vari confronti.

Di seguito riporto le reti che ho utilizzato con la loro struttura e l'accuratezza ottenuta:

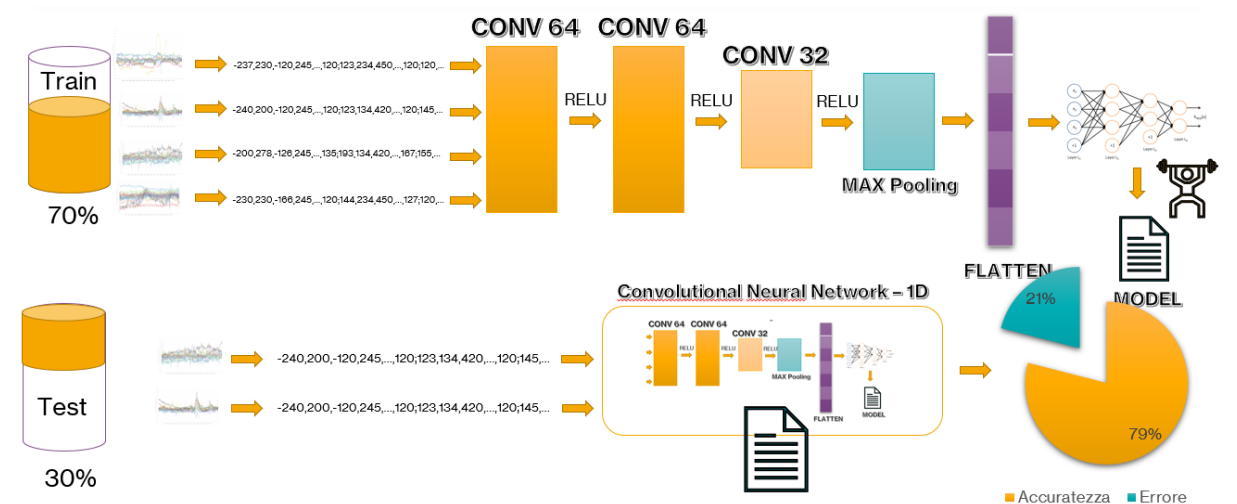
### 1D-CNN:

*Summary:*

Layer (type)	Output Shape	Param #
conv1d_67 (Conv1D)	(None, 598, 64)	2944
conv1d_68 (Conv1D)	(None, 596, 64)	12352
conv1d_69 (Conv1D)	(None, 594, 32)	6176
dropout_23 (Dropout)	(None, 594, 32)	0

max_pooling1d_23 (MaxPooling (None, 297, 32))		0
flatten_23 (Flatten)	(None, 9504)	0
dense_67 (Dense)	(None, 100)	950500
dense_68 (Dense)	(None, 50)	5050
dense_69 (Dense)	(None, 2)	102

### Architettura:



Rete	Accuratezza	Recall
1D-CNN	79%	81%

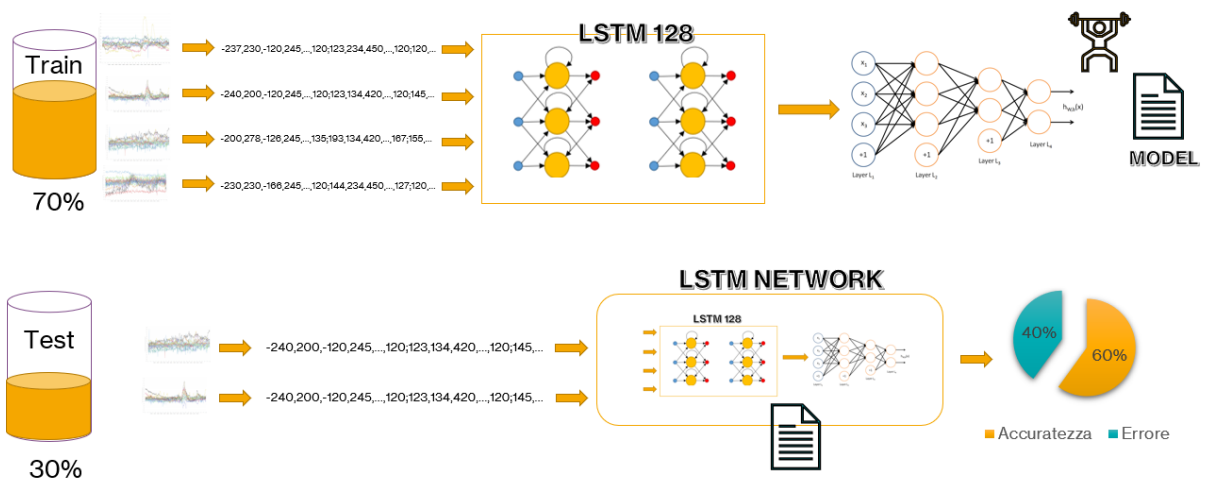
N.B. Per ottenere i valori dalla rete 1D-CNN non è stata eseguita una 5-cross validation, ma solamente due esecuzioni (quindi una 2-cross validation)

**LSTM:**

*Summary:*

Layer (type)	Output Shape	Param #
lstm_91 (LSTM)	(None, 128)	46400
dropout_82 (Dropout)	(None, 128)	0
dense_220 (Dense)	(None, 128)	10100
dense_221 (Dense)	(None, 50)	3945
dense_222 (Dense)	(None, 2)	79

*Architettura:*



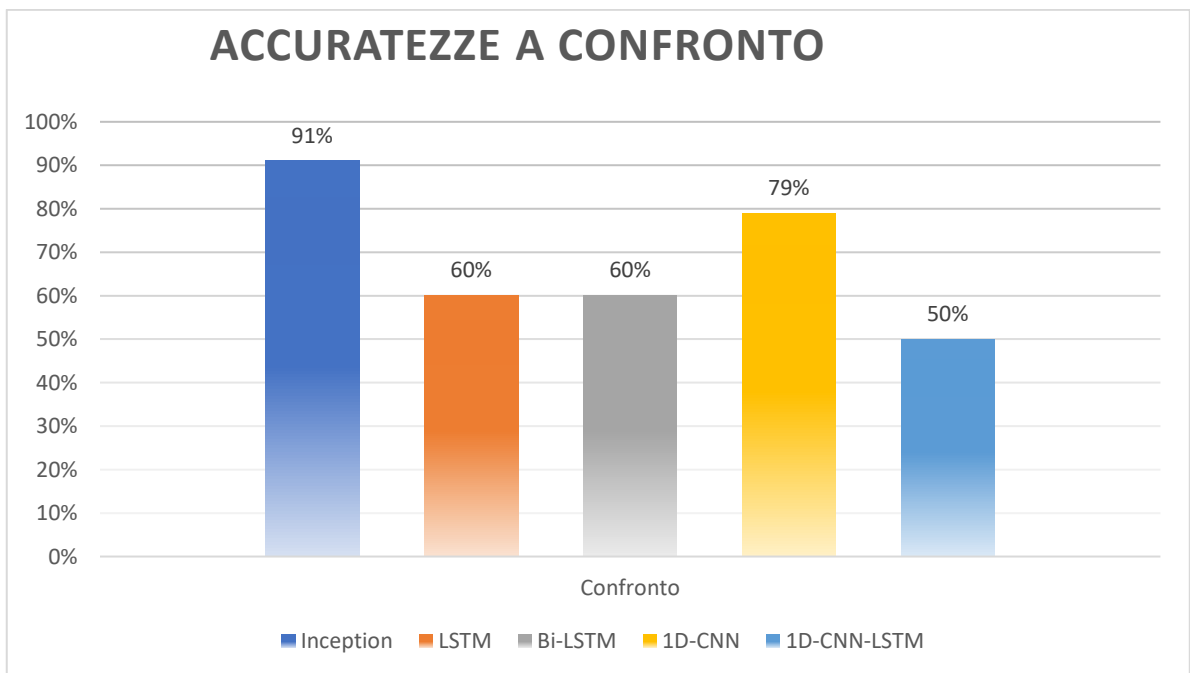


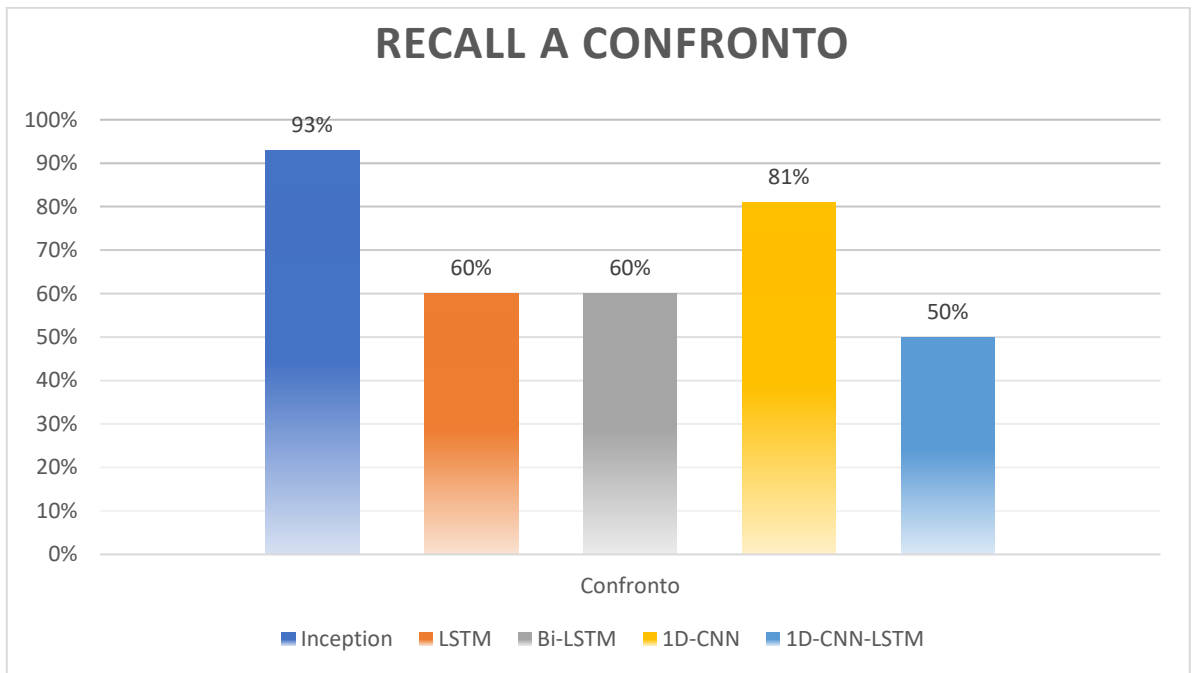
<b>Rete</b>	<b>Accuratezza</b>	<b>Recall</b>
LSTM	60%	60%

N.B. Per ottenere i valori dalla rete LSTM non è stata eseguita una 5-cross validation, ma solamente due esecuzioni (quindi una 2-cross validation)

**Accuratezze e Recall a confronto:**

Rete	Accuratezza	Recall
LSTM	60%	60%
1D-CNN	79%	81%
1D-CNN + LSTM	50%	50%
INCEPTION TIME	91%	93%





## 4. CONCLUSIONI

La rete Inception si è dimostrata essere la migliore rete di classificazione per gli EEG di un paziente epilettico, riuscendo a totalizzare un'accuratezza superiore al 90% e superando di gran lunga il suo diretto competitor (1D-CNN che ha totalizzato 79%).

La LSTM, che in letteratura si è dimostrata più performante nel classificare le fasi pre-ictali, non si è dimostrata all'altezza e la motivazione è abbastanza ovvia: non ha potuto addestrare un modello con una buona quantità di informazioni. Questo ha creato uno squilibrio nella classificazione non riuscendo a differenziare le varie tipologie di EEG. Infatti, nel caso di questo studio, abbiamo lavorato con pochissimi dati creando dei problemi su alcune di queste reti, come la LSTM e la Bi-LSTM; questo ha anche dimostrato che la Inception Time e la CNN funzionano molto bene su pochi dati, riuscendo a classificare con accuratezze abbastanza alte. La migliore in tutto ciò è ovviamente la Inception che ha superato il 90% di accuratezza dovuto sicuramente al suo filtro bottleneck che oltre a diminuire la dimensionalità degli input, diminuisce anche la complessità e mitiga i problemi di overfitting dovuto a piccoli dataset.

Indipendentemente dai dataset utilizzati, ha assolutamente senso utilizzare la Inception per la classificazione di serie temporali di ogni tipo, visto i risultati promettenti ottenuti in letteratura e con i confronti avute con le altre reti.

Sicuramente non possiamo dire, in generale, che l'Inception time sia la migliore rete in assoluto per la classificazione degli EEG per la previsione delle crisi, ma può essere presa in considerazione come eventuale rete di controllo e per poter avere un confronto con i risultati ottenuti con altre tipologie di reti.

#### 4.1. SVILUPPI FUTURI

Tra gli sviluppi futuri, bisogna considerare assolutamente un confronto, con le reti maggiormente utilizzate in letteratura, con dataset molto più corposi, con molti più dati e molti più pazienti.

In questo caso, avendo approcci differenti, potremmo ulteriormente valutare la bontà della rete Inception e confrontare i risultati sapremmo se la rete continua ad avere comportamenti al di sopra della media di altre tipologie di reti o se, in alcuni casi, si comporta peggio.

Le reti neurali sono una specie di “black box” e quindi non possiamo sapere a prescindere se il comportamento vero un tipo di dataset possa essere buono oppure no; quello che possiamo fare è sicuramente provare e testare con tutti i mezzi possibili.

Sicuramente la rete Inception è un’ottima rete da tenere sempre in considerazione, anche come eventuale rete di confronto, e potrebbe avere delle applicazioni in cui potrebbe dimostrarsi essere la migliore, anche perché essendo che è una rete nata a fine 2019, non è stata ancora testata con tutti i possibili dataset e potrebbe nascondere delle potenzialità che vale la pena essere utilizzata.

## 5. BIBLIOGRAFIA

1. **Prior, Dr. Reinhard.** Epilessia - Dr. Reinhard Prior - Specialista e Docente di Neurologia. *Neurologia*. [Online] [https://www.neurologia.it/epilessia/epilessia\\_info.html](https://www.neurologia.it/epilessia/epilessia_info.html).
2. *Bioelectromagnetism: Principles and Applications of Bioelectric.* **Malmivuo J., Plonsey R.** 1995.
3. *American Electroencephalographic Society Guidelines for Standard Electrode Position Nomenclature.* **J. Sharbrough F, Chatrian G-E, Lesser RP, Lüders H, Nuwer M, Picton TW.** 1991.
4. *Proceedings: Epileptic seizure prediction.* **Walsh., SS Viglione and GO.** 1975.
5. *On the prediction of epileptic.* **Z Rogowski, I Gath, and E Bental.** 1981.
6. *Correlation dimension and integral do not predict epileptic seizure.* **Mary Ann F Harrison, Ivan Osorio, Mark G Frei, Srividhya Asuri, and Ying-Cheng Lai.** 2005.
7. *Prediction of epileptic seizures from two channel EEG.* **Y Salant, I Gath, and O Henriksen.** 1998.
8. *Seizure anticipation in pediatric epilepsy: use of kolmogorov entropy.* **Wim van Drongelen, Sujatha Nayak, David M Frim, Michael H, Kohrman, Vernon L Towle, Hyong C Lee, Arnetta B McGee, Maria S, Chico, and Kurt E Hecox.** 2003.
9. **Vulpani, Angelo.** *Entropia di Kolmogorov-Sinai: Enciclopedia della scienza e della tecnica.* s.l. : Treccani, 2008.
10. *The first international collaborative workshop on seizure prediction: summary and data description.* **Litt, Klaus Lehnertz and Brian.** 2005.
11. *Epileptic seizures are preceded by a decrease in synchronization.* **Florian Mormann, Thomas Kreuz, Ralph G Andrzejak, Peter David, Klaus Lehnertz, and Christian E Elger.** 2003.
12. *Crowd-sourcing reproducible seizure prediction with long-term human intracranial eeg.* **Levin Kuhlmann, Philippa Karoly, Dean R Freestone, Benjamin H, Brinkmann, Andriy Temko, Alexandre Barachant, Feng Li, Gilberto, Titericz Jr, Brian W Lang, Daniel Lavery, et al.** *Epilepsyecosystem.org.* 2018.
13. *Critical slowing as a biomarker for seizure susceptibility.* **Matias I Maturana, Christian Meisel, Katrina Dell, Philippa J Karoly, Wendyl D'Souza, David B Grayden, Anthony N Burkitt, Premysl, Jiruska, Jan Kudlacek, Jaroslav Hlinka.** 2008.
14. *Machine Learning for Predicting Epileptic Seizures.* **Khansa Rasheed, Adnan Qayyum, Junaid Qadir, Shobi Sivathamboo, Patrick Kwan, Terence O'Brien, Levin Kuhlmann and Adeel Razi.** 2020.
15. *Deep learning.* **Yann LeCun, Yoshua Bengio, and Geoffrey Hinton.** 2015.

16. *Focal onset seizure prediction using convolutional networks.* **Haidar Khan, Lara Marcuse, Madeline Fields, Kalina Swann, and Bulent Yener.** 2017.
17. *Deep learning for eeg data analytics: A survey.* **Gen Li, Chang Lee, Jason Jung, Young Chul Youn, and David Camacho.** 2019.
18. *Deep learning for epileptic intracranial eeg data.* **Andreas Antoniadis, Loukianos Spyrou, Clive Cheong Took, and Saeid Sanei.** 2016.
19. *A generalised seizure prediction with convolutional neural networks for intracranial and scalp electroencephalogram data analysis.* **Nhan Duy Truong, Anh Duy Nguyen, Levin Kuhlmann, Mohammad Reza Bonyadi, Jiawei Yang, and Omid Kavehei.** 2017.
20. *Prediction of seizure likelihood with a long-term, implanted seizure advisory system in patients with drug-resistant epilepsy: a first-in-man study.* **Mark J Cook, Terence J O'Brien, Samuel F Berkovic, Michael Murphy, Andrew Morokoff, Gavin Fabinyi, Wendyl D'Souza, Raju Yerra, John Archer, Lucas Litewka.** 2013.
21. *Human intracranial eeg quantitative analysis and automatic feature learning for epileptic seizure prediction.* **Ramy Hussein, Mohamed Osama Ahmed, Rabab Ward, Z Jane Wang, Levin Kuhlmann, and Yi Guo.** 2019.
22. *A long short-term memory deep learning network for the prediction of epileptic seizures using eeg signals.* **Kostas M Tsiouris, Vasileios C Pezoulas, Michalis Zervakis, Spiros Konitsiotis, Dimitrios D Koutsouris, and Dimitrios I Fotiadis.** 2018.
23. *Efficient epileptic seizure prediction based on deep learning.* **Doud, Hisham, and Magdy A. Bayoumi.** 2019.
24. **UK, LabTest Online.** Lab Test Online UK. [Online] <https://labtestsonline.org.uk/articles/accuracy-precision-specificity-sensitivity>.
25. *InceptionTime: Finding AlexNet for Time Series Classification.* **Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F. Schmidt, Jonathan Weber, Geoffrey I. Webb, Lhassane Idoumghar, Pierre-Alain Muller, François Petitjean.** s.l. : Researchgate, 2020.
26. *Deep residual learning for image recognition.* **He K, Zhang X, Ren S, Sun J.** 2016.
27. **Kaggle.** American Epilepsy Society Seizure Prediction Challenge. *Kaggle.* [Online] 2014. <https://www.kaggle.com/c/seizure-prediction/data?select=sampleSubmission.csv>.
28. **Ismail Fawaz, Hassan and Lucas, Benjamin and Forestier, Germain and Pelletier, Charlotte and Schmidt, Daniel F. and Weber, Jonathan and Webb, Geoffrey I. and Idoumghar, Lhassane and Muller, Pierre-Alain and Petitjean, François.** Inception Time. *GitHub.* [Online] 2019. <https://github.com/hfawaz/InceptionTime>.
29. *Epileptic seizure prediction using big data and deep learning: toward a mobile system.* **Isabell Kiral-Kornek, Subhrajit Roy, Ewan Nurse, Benjamin Mashford, Philippa Karoly, Thomas Carroll, Daniel Payne, Susmita Saha, Steven Baldassano and Terence O'Brien.** 2018.

