

Analisi Comparativa di Architetture di Deep Learning per la Classificazione della Fase del Sonno

Francesco Pio Rossi¹

¹f.rossi51@campus.uniurb.it

Riassunto

In questo progetto si affronta un problema di classificazione multiclasse in cui, dato in input un segnale di elettroencefalogramma campionato a 100 Hz, il modello deve determinare la fase del sonno corrispondente (Wake, N1, N2, N3 o REM). Nel lavoro che segue vengono quindi implementate e confrontate quattro diverse architetture di reti neurali, con l'obiettivo di valutarne le prestazioni e individuare quella più efficace. Le architetture considerate sono un *percettore multistrato* (MLP), una *rete neurale convoluzionale* (CNN), una *Long-Short-Term Memory* (LSTM) e un'architettura ibrida *CNN + LSTM*.

1 Introduzione

Nel 2007 l'*American Academy of Sleep Medicine* (AASM) ha definito una classificazione del sonno in cinque stadi [3], che rappresenta tuttora lo standard di riferimento in ambito clinico e di ricerca:

- **Wake (W):** stato di veglia, caratterizzato da elevata attività cerebrale e muscolare.
- **Non-REM N1:** sonno leggero, fase di transizione tra la veglia e il sonno vero e proprio; rappresenta una piccola percentuale del tempo totale di sonno.
- **Non-REM N2:** sonno stabile, che costituisce lo stadio più abbondante durante una notte tipica.
- **Non-REM N3:** sonno profondo, associato al recupero fisico e caratterizzato da onde cerebrali lente.
- **REM (Rapid Eye Movement):** fase associata prevalentemente all'attività onirica, caratterizzata da rapidi movimenti oculari e da una quasi completa atonia muscolare.

Diversi segnali fisiologici possono contribuire a distinguere tra queste fasi. Tra questi, i più importanti sono l'*elettroencefalogramma* (EEG), l'*elettrooculogramma* (EOG) e l'*elettromiogramma* (EMG).

L'EEG registra l'attività elettrica cerebrale mediante elettrodi posizionati sul cuoio capelluto ed è il se-

gnale più informativo al fine di individuare lo stadio del sonno corretto. L'EOG misura l'attività elettrica associata ai movimenti oculari ed è particolarmente utile per l'identificazione della fase REM, caratterizzata da rapidi movimenti degli occhi. L'EMG, invece, registra l'attività elettrica dei muscoli scheletrici e consente di valutare il tono muscolare, che risulta elevato durante la veglia, ridotto nelle fasi non-REM e quasi completamente assente durante la fase REM.

Sebbene una discriminazione più accurata tra i cinque stadi del sonno richiederebbe idealmente la combinazione di più segnali fisiologici, in questo progetto viene utilizzato esclusivamente il segnale EEG, che fornisce comunque informazioni importanti per la corretta classificazione. Il lavoro si concentra quindi sull'utilizzo di segnali EEG etichettati per l'addestramento di diverse architetture di reti neurali, le cui prestazioni vengono successivamente valutate su un insieme di dati mai visto, al fine di analizzarne la capacità di generalizzazione e individuare l'architettura più efficace per il nostro scopo.

In particolare, vengono realizzati quattro modelli utilizzando la nota libreria di python PyTorch. Il confronto avviene sulla base delle metriche più utilizzate in un problema di classificazione: *accuratezza*, *precision*, *recall*, *f1-score* e *matrice di confusione*.

La prima architettura realizzata è un *percettore multistrato* (MLP). Sebbene questo modello non sia particolarmente adatto al compito in esame, in quanto non è in grado di catturare le dipendenze temporali

e spaziali presenti nei dati, viene comunque considerato a fini didattici e come termine di confronto.

La seconda architettura utilizzata è una *Long Short-Term Memory* (LSTM), una versione migliorata delle classiche *Recurrent Neural Network* (RNN), ideata proprio per gestire sequenze di dati. A differenza delle RNN, le LSTM sono in grado di catturare dipendenze temporali a lungo raggio, e risultano particolarmente adatte a serie temporali come l'EEG nel nostro caso. Viene poi sviluppata una *Convolutional Neural Network* (CNN), che utilizza filtri monodimensionali per estrarre automaticamente feature locali dal segnale EEG.

Infine viene adottata un'architettura ibrida, in cui alcuni strati convoluzionali sono utilizzati per l'estrazione delle feature dai segnali EEG, successivamente fornite in ingresso a una LSTM. Questa combinazione permette di integrare l'efficacia delle CNN nell'estrazione di feature locali con la capacità delle LSTM di analizzare le dipendenze temporali.

2 Dataset

Il dataset utilizzato in questo progetto – denominato *Sleep* e disponibile nel *Time Series Classification Repository* [4] – è una versione rielaborata del noto *Sleep-EDF Database Expanded* [1][2]. Quest'ultimo è un dataset pubblico ampiamente utilizzato nella ricerca clinica grazie alla sua ricchezza e varietà di dati: comprende 197 registrazioni polisonnografiche (PSG) contenenti segnali eterogenei come l'EEG, l'EOG, l'EMG e diversi marcatori di eventi. In alcuni casi, il database include anche informazioni sulla respirazione e sulla temperatura corporea dei soggetti monitorati.

Sleep raccoglie dati provenienti da 82 soggetti sani e, a differenza del dataset originale, è *univariato*, poiché contiene un singolo segnale EEG campionato a 100 Hz.

I dati sono suddivisi in 478 785 campioni per il training e 90 315 per il testing. Ogni campione rappresenta una serie temporale del segnale EEG, composta da 187 punti dati (quasi due secondi di segnale), che registrano l'ampiezza del potenziale elettrico sul cuoio capelluto. A ciascun campione è associata un'etichetta corrispondente a una fase del sonno secondo la classificazione standard in cinque stadi: veglia (W), sonno non-REM di tipo N1, N2 e N3, e sonno REM.

Il dataset presenta uno sbilanciamento marcato: la maggior parte dei campioni appartiene alla fase N2, mentre le fasi Wake e N1 sono le meno rappresentate. Questa distribuzione riflette la proporzione naturale

delle diverse fasi del sonno negli esseri umani, dal momento che durante una notte tipica il sonno N2 occupa gran parte del tempo, mentre i periodi di veglia e di sonno leggero (N1) sono relativamente brevi. La distribuzione percentuale dei dati per ciascuna classe nel training e nel test set è riportata nella Tabella 1.

Stadio del sonno	Train (%)	Test (%)
Wake (W)	13.84	4.05
N1	6.56	10.98
N2	42.73	48.93
N3	15.81	18.82
REM	21.06	18.82

Tabella 1: Distribuzione percentuale dei dati per classe nel training set e nel test set.

A titolo di riferimento, un classificatore *MiniRocket* applicato al dataset *Sleep* raggiunge un'accuratezza del 92.7%.

2.1 Preprocessing dei dati

Il dataset è stato processato prima dell'addestramento. In primo luogo, per far fronte allo sbilanciamento tra le classi e sfruttando l'ampia disponibilità di dati, sono stati selezionati solo 30 000 campioni per ciascuna classe (possibile visto che la classe meno frequente contiene circa 32 000 campioni). Questo approccio garantisce un bilanciamento perfetto dei dati, assicurando una fase di addestramento di qualità e prevenendo che il modello tenda a predire con maggior frequenza le classi più rappresentate.

Questa operazione comporta la rinuncia a una significativa quantità di dati. Tuttavia, ridurre il numero di campioni è utile e in questo caso necessario per limitare il consumo di memoria e la potenza computazionale, che, su un dataset di tali dimensioni, sarebbe altrimenti molto elevato.

Inoltre, poiché il progetto è stato eseguito su *Google Colab*, la quantità di memoria RAM disponibile era limitata.

Il dataset risultante contiene quindi 150 000 campioni ed è perfettamente bilanciato.

Prima di proseguire i dati sono stati standardizzati, ossia trasformati in modo che ogni campione avesse media 0 e deviazione standard 1.

La standardizzazione viene eseguita applicando la seguente formula:

$$x' = \frac{x - \mu}{\sigma}$$

dove x è il valore originale di un punto del segnale EEG, μ è la media dei punti, σ è la deviazione standard e x' è il valore standardizzato.

La Figura 1 mostra sei segnali EEG selezionati casualmente dal dataset dopo la fase di standardizzazione.

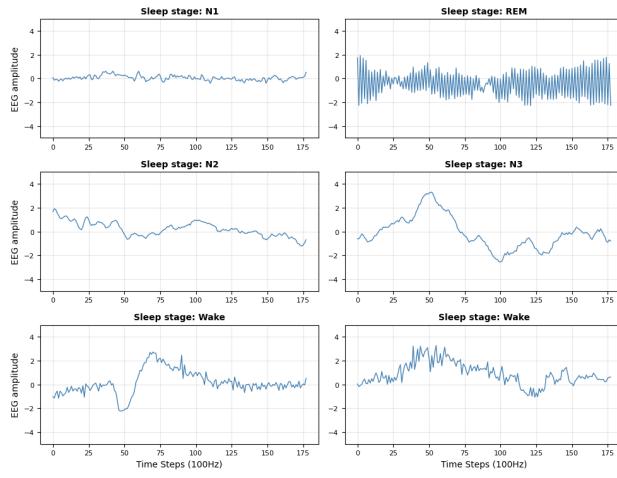


Figura 1: Alcuni campioni dal dataset dopo la standardizzazione

3 Modelli

In questa sezione vengono descritte brevemente le quattro architetture implementate. Tutti i modelli sono stati addestrati per un totale di 10 epoche, utilizzando un *batch size* pari a 64; ciò significa che, durante la fase di addestramento, vengono elaborati simultaneamente 64 segnali EEG. Il valore del *learning rate* è stato invece adattato in base all'architettura considerata.

Al termine di ciascuna epoca sono stati inoltre registrati sia il valore della funzione di perdita (*loss*) sia l'accuratezza sul training set, al fine di monitorare l'andamento delle prestazioni del modello durante l'addestramento.

Per l'addestramento di tutti i modelli è stata utilizzata la funzione di perdita *CrossEntropyLoss*, comunemente impiegata nei problemi di classificazione multiclassa.

3.1 Modello 1: MLP

La MLP è la più semplice architettura di rete neurale ed è poco adatta al compito in esame, poiché gli input della rete – i punti della serie temporale del segnale EEG – perdono qualsiasi informazione spaziale e temporale. È stata comunque implementata a scopo comparativo e didattico.

La rete MLP utilizzata è composta da uno strato di input con 178 nodi, corrispondenti ai punti dati di un segnale EEG, e da uno strato nascosto densamente connesso con 512 neuroni. La funzione di attivazione adottata è la ReLU, mentre il learning rate è 0.0001. La Figura 2 schematizza l'architettura MLP.

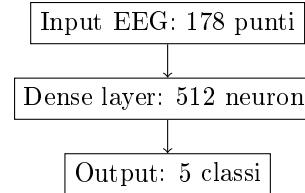


Figura 2: Architettura MLP

3.2 Modello 2: CNN

Il secondo modello implementato è una CNN composta da due livelli convoluzionali seguiti da uno strato denso, il cui obiettivo è mappare il vettore delle feature estratte in una delle cinque classi previste. Questa architettura estrae automaticamente feature rilevanti dal segnale EEG mediante filtri convoluzionali monodimensionali, vista la natura del dataset.

Il primo livello convoluzionale opera direttamente sul segnale EEG in ingresso e utilizza 32 filtri di dimensione 7, producendo altrettante feature map. A queste vengono successivamente applicate la funzione di attivazione ReLU e un'operazione di *max pooling*, che consente di ridurre la dimensionalità del segnale mantenendo però le informazioni più significative. Il secondo livello convoluzionale impiega 64 filtri di dimensione 5 ed è anch'esso seguito da ReLU e *max pooling*, e ha l'obiettivo di estrarre dai segnali EEG pattern via via più complessi.

Ogni operazione di convoluzione è preceduta dall'applicazione di un opportuno *padding*, con l'obiettivo di preservare la dimensionalità del segnale dopo l'applicazione del filtro. Nel caso di convoluzioni monodimensionali con passo unitario (*stride* pari a 1), il padding simmetrico da applicare all'inizio e alla fine del vettore è calcolato come:

$$p = \frac{k - 1}{2}$$

dove k rappresenta la dimensione del kernel.

Le feature estratte vengono infine appiattite in un unico vettore e fornite in ingresso alla seconda parte dell'architettura, costituita da uno strato denso con 256 neuroni, incaricato della classificazione del segnale EEG.

Al pari del modello MLP, il learning rate è stato fissato a 0.0001.

La Figura 3 mostra lo schema dell’architettura CNN adottata, insieme alle dimensioni dei tensori dopo l’applicazione di ciascuna operazione.

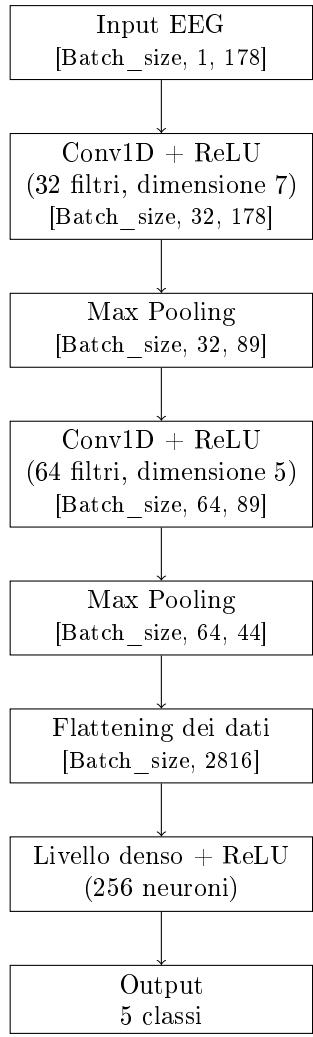


Figura 3: Architettura della CNN, con le dimensioni dei tensori in gioco dopo ogni operazione

3.3 Modello 3: LSTM

Il terzo modello implementato è una LSTM. A differenza delle MLP e delle CNN, le LSTM nascono proprio per l’elaborazione di serie temporali e dati sequenziali, risultando quindi particolarmente adatte agli obiettivi di questo progetto.

L’architettura realizzata consiste in una LSTM a 2 strati, con uno stato nascosto (la “memoria”) di dimensione 64. La sequenza in ingresso, composta da

178 punti temporali, viene processata passo dopo passo. Per la classificazione si adotta una strategia *Many-to-One*: viene considerato soltanto lo stato nascosto dell’ultimo step temporale ($t = 178$), che funge da vettore riassuntivo dell’intera sequenza. Questa rappresentazione finale viene passata a un singolo livello lineare che ha l’obiettivo di calcolare i logits finali corrispondenti alle cinque classi del problema.

Per questo terzo modello le migliori prestazioni sono state ottenute utilizzando un *learning rate* più elevato, pari a 0.001.

La Figura 4 schematizza il flusso dei dati e le dimensioni dei tensori dopo ogni operazione.

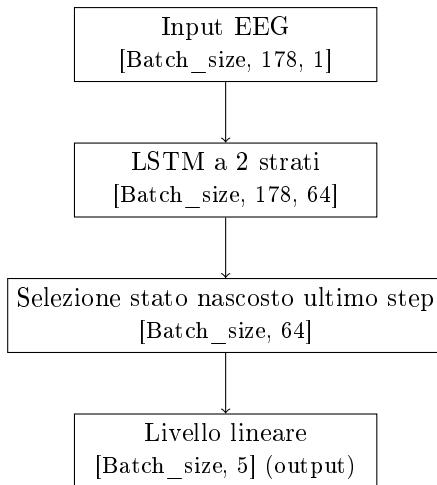


Figura 4: Architettura della LSTM, con dimensioni dei tensori in gioco dopo ogni operazione.

3.4 Modello 4: Modello Ibrido CNN + LSTM

L’ultimo modello è un’architettura ibrida che combina i punti di forza delle due precedenti reti. Il flusso dei dati attraversa tre blocchi:

1. *CNN 1D*: estrae feature rilevanti dal segnale EEG. Il segnale passa attraverso due blocchi convoluzionali, ognuno seguito da ReLU e Max Pooling. Viene applicato un padding opportuno per mantenere la dimensione temporale invariata dopo la convoluzione. Ogni operazione di pooling, invece, dimezza la lunghezza temporale.
2. *LSTM*: riceve in ingresso l’output dell’ultimo livello convoluzionale, che ha dimensione $[T, F]$, dove T è la lunghezza temporale residua del segnale filtrato (ridotta dalle operazioni di pool-

ling) e F è il numero di filtri convoluzionali usati. Questo tensore viene interpretato come una sequenza di T passi temporali: a ciascun passo t la LSTM riceve in ingresso un vettore di F valori, che rappresenta l'attivazione simultanea di tutti i filtri convoluzionali in quell'istante del segnale. La LSTM elabora la sequenza scorrendo lungo la dimensione temporale T , scovando le dipendenze temporali tra i diversi segmenti del segnale EEG e producendo un risultato che tiene conto dell'evoluzione delle feature nel tempo. Questo aspetto distingue il modello ibrido dal modello basato esclusivamente su CNN, nel quale le feature map estratte venivano utilizzate direttamente per la classificazione finale, senza considerare esplicitamente le relazioni temporali.

3. *Classificazione (Many-to-One)*: viene adottata una strategia *Many-to-One*: si considera esclusivamente lo stato nascosto prodotto dell'ultimo passo temporale della LSTM. Questo vettore viene dato in input a un singolo layer lineare che produce le predizioni sulle 5 classi di stadio del sonno.

Analogamente al modello LSTM è stato utilizzato un learning rate pari a 0.001.

La Figura 5 illustra schematicamente il flusso dei dati all'interno del modello e le dimensioni dei tensori dopo ciascuna operazione, evidenziando anche i valori degli iperparametri utilizzati.

4 Metriche

I vari modelli vengono confrontati utilizzando le metriche più comuni nei problemi di classificazione multiclasse. In particolare saranno considerate l'accuratezza, la recall, la precision, l'F1-score e la matrice di confusione.

Tutte queste metriche sono state ottenute tramite il metodo `classification_report` della libreria `scikit-learn`. L'accuratezza rappresenta una misura generale delle prestazioni complessive del modello, mentre recall, precision e F1-score possono essere valutate sia globalmente (facendo ad esempio una media) sia rispetto a una classe specifica.

4.1 Accuratezza

L'accuratezza fornisce una prima indicazione delle prestazioni complessive del modello. Si calcola come il rapporto tra il numero di predizioni corrette e il numero totale di predizioni effettuate:

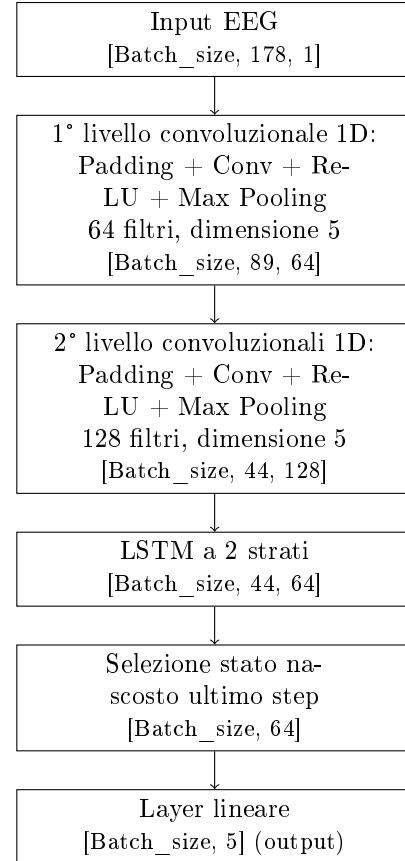


Figura 5: Architettura ibrida CNN + LSTM, con dimensioni dei tensori in gioco dopo ogni operazione

$$\text{Accuratezza} = \frac{\text{predizioni corrette}}{\text{predizioni totali}}$$

Questa metrica può risultare poco affidabile quando il dataset è sbilanciato, poiché il modello potrebbe ottenere un'accuratezza elevata predicendo sempre la classe più frequente. Nel nostro caso il dataset è stato bilanciato perfettamente, quindi l'accuratezza può essere considerata una misura attendibile delle prestazioni complessive del modello.

4.2 Recall e Precision

La recall misura la capacità del modello di individuare correttamente tutti i campioni di una determinata classe. Si calcola come il rapporto tra i veri positivi e la somma di veri positivi e falsi negativi:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

La precision indica quanto le predizioni attribuite a una classe siano effettivamente corrette. Si calcola come il rapporto tra i veri positivi e la somma di veri positivi e falsi positivi:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

4.3 F1-score

L'F1-score si calcola come la media armonica tra precision e recall:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Come per precision e recall, l'F1-score può essere calcolato per ciascuna classe o aggregato per riassumere la prestazione su tutte le classi tramite medie macro o ponderate.

4.4 Matrice di confusione

La matrice di confusione è uno strumento che permette di capire quali classi il modello tende a confondere. Si tratta di una tabella in cui le righe corrispondono alle etichette reali e le colonne alle predizioni del modello. I valori lungo la diagonale rappresentano i campioni classificati correttamente, mentre gli elementi fuori diagonale indicano errori di classificazione. La matrice ha dimensione $n \times n$, dove n è il numero di classi.

5 Risultati

In questa sezione vengono presentati i risultati ottenuti dai quattro modelli su dati mai visti, valutati secondo le metriche descritte nella Sezione 4.

5.1 Risultati Modello 1: MLP

La MLP è stata addestrata per un numero variabile di epochhe al fine di osservare come le prestazioni evolvono con l'aumentare delle iterazioni. Inizialmente si nota un miglioramento delle metriche, ma oltre un certo numero di epochhe le performance cominciano a peggiorare. Questo comportamento è tipico dell'**overfitting**, in cui il modello si adatta troppo ai dati di training e fatica a generalizzare su dati mai visti.

La Tabella 2 mostra chiaramente che, aumentando le epochhe da 20 a 30, l'accuratezza e l'F1-score diminuiscono. Inoltre, dalla Figura 6 si osserva come

Epochs	Accuracy (%)	F1-Score
5	44.12	0.41
10	49.60	0.46
20	52.52	0.48
30	51.15	0.47

Tabella 2: Performance della MLP

la loss diminuisca e l'accuratezza aumenti con l'avanzare delle epochhe, confermando che il modello continua a migliorare sui dati di addestramento. Tuttavia, i progressi sono più significativi nelle prime epochhe e diventano progressivamente meno marcati, come evidenziato dall'appiattirsi delle due curve.

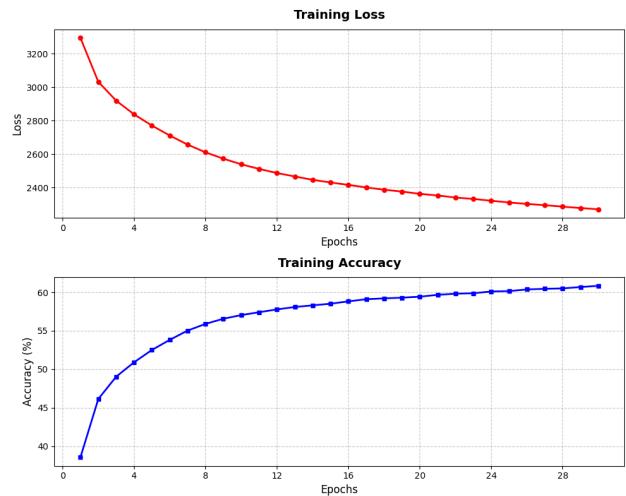


Figura 6: Loss e accuratezza durante l'addestramento della MLP

A prescindere dal numero di epochhe impiegato, l'utilizzo di una MLP produce prestazioni limitate per questo specifico problema, con valori di accuratezza che superano di poco il 50%.

5.2 Risultati Modello 2: CNN

Le performance della CNN sono riportate nella Tabella 3. Si ottiene un'accuratezza del 56% e un F1-score del 52%, rappresentando un passo in avanti rispetto alla MLP.

5.3 Risultati Modello 3: LSTM

Le performance della LSTM sono riportate nella Tabella 4. Si ottiene un'accuratezza del 58% e un F1-score del 53%, rappresentando un passo in avanti rispetto alla CNN.

Classe	Precision	Recall	F1-score
Wake (W)	0.38	0.52	0.44
N1	0.25	0.56	0.35
N2	0.84	0.51	0.64
N3	0.68	0.81	0.74
REM	0.41	0.43	0.42
Macro avg	0.51	0.57	0.52
Weighted avg	0.66	0.56	0.58

Tabella 3: Classification report del modello CNN sul test set

Classe	Precision	Recall	F1-score
Wake (W)	0.36	0.63	0.46
N1	0.27	0.55	0.36
N2	0.83	0.57	0.67
N3	0.66	0.83	0.74
REM	0.48	0.36	0.41
Macro avg	0.52	0.59	0.53
Weighted avg	0.66	0.58	0.60

Tabella 4: Classification report del modello LSTM sul test set

5.4 Risultati Modello 4: CNN + LSTM

Il modello ibrido si è rivelato il più performante, raggiungendo un'accuratezza superiore al 61% e un F1-score del 56%. Tutti i risultati delle metriche sono riportati nella Tabella 5.

Classe	Precision	Recall	F1-score
Wake (W)	0.41	0.63	0.50
N1	0.33	0.41	0.36
N2	0.83	0.58	0.68
N3	0.66	0.84	0.74
REM	0.47	0.60	0.53
Macro avg	0.54	0.61	0.56
Weighted avg	0.67	0.61	0.62

Tabella 5: Classification report del modello ibrido CNN + LSTM sul test set

5.5 Confronto tra i vari modelli

Dall'osservazione della Tabella 3, della Tabella 4 e della Tabella 5, emerge che la classe N3 presenta il valore di recall più alto, mentre la classe N2 ha il valore di precision più elevato.

Il fatto che la recall sia maggiore per N3 indica che questa classe è la più facilmente distinguibile: pochi campioni di sonno profondo vengono confusi con altri stadi. Questo riflette il fatto che i segnali EEG appartenenti a N3 presentano caratteristiche distintive ben definite, come la presenza di onde lente ad alta

ampiezza.

D'altra parte, la precision elevata per N2 significa che, tra tutti i campioni che il modello predice come N2, la maggior parte è effettivamente corretta. In altre parole, le predizioni di N2 sono molto affidabili, anche se alcuni veri campioni di N2 possono essere confusi con altre fasi del sonno.

La Figura 7 confronta i valori di accuratezza e F1-score ottenuti dai diversi modelli. Si nota come CNN e LSTM abbiano prestazioni simili mentre il modello ibrido raggiunge risultati superiori, grazie alla combinazione della capacità della CNN di estrarre feature rilevanti dai segnali EEG e della LSTM di cogliere le dipendenze temporali all'interno del segnale.

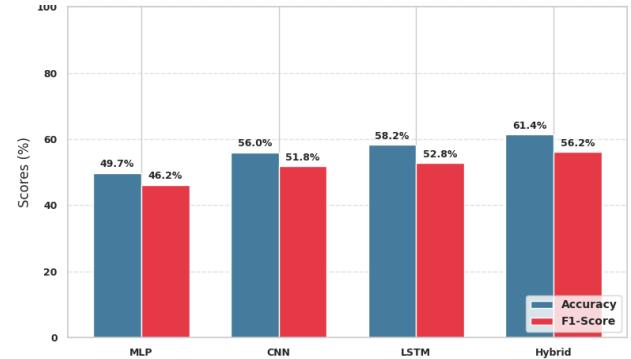


Figura 7: Confronto dei valori di accuratezza e F1-score dei 4 modelli

La Figura 8 mostra la matrice di confusione del modello ibrido. Sulla diagonale principale è presente la recall per ciascuna classe, ovvero la percentuale di campioni classificati correttamente rispetto al totale reale di quella classe. Osservando i risultati emergono

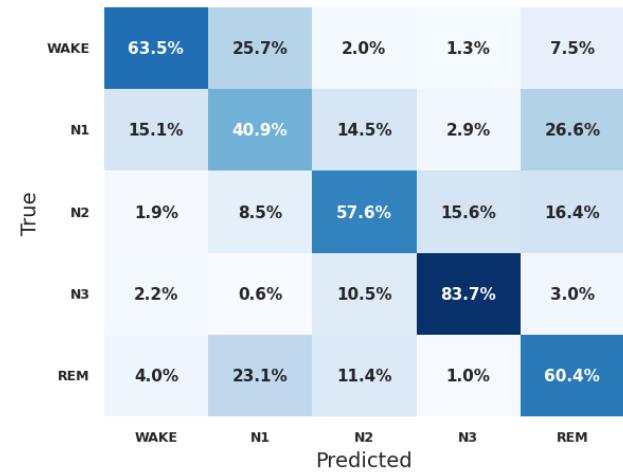


Figura 8: Matrice di confusione del modello ibrido

le seguenti osservazioni:

- *Miglior Classificazione (N3)*: come già anticipato, la classe che il modello riconosce con maggiore accuratezza è N3 (sonno profondo), con un tasso di successo dell'83.7%.
- *Peggior Classificazione (N1)*: la classe più problematica risulta essere N1 (sonno leggero), che viene classificata correttamente solo nel 40.9% dei casi.

Lo stadio N1 viene spesso confuso con la fase REM, e mostra una notevole ambiguità anche nei confronti della fase WAKE.

Si osserva che, in generale, i segnali EEG delle fasi N1, REM e WAKE presentano caratteristiche simili, rendendo la loro distinzione più complessa. Ciò è dovuto al fatto che la fase N1, essendo sonno leggero, presenta un'attività cerebrale ancora piuttosto elevata, così come anche la fase REM è caratterizzata da un'attività cerebrale intensa, simile a quella dello stato di veglia.

Per migliorare le prestazioni del modello sarebbe utile integrare altri segnali fisiologici nello studio: ad esempio l'EOG potrebbe facilitare la distinzione tra la fase N1 e lo stato WAKE.

Per concludere, la Figura 9 mostra l'andamento della funzione di perdita e dell'accuratezza durante le epoche di addestramento dei quattro modelli. Sebbene questi valori siano calcolati sul training set e non riflettano pienamente la capacità di generalizzazione, è interessante osservare come ciascun modello apprenda in modo diverso. Il modello ibrido ottiene già dalla prima epoca risultati superiori rispetto agli altri, mentre la LSTM parte con prestazioni inizialmente inferiori persino rispetto alla MLP, per poi migliorare significativamente con il procedere delle epoche, stabilizzandosi intorno alla quarta epoca. Invece, per la MLP, la CNN e il modello ibrido i valori si stabilizzano già a partire dalla seconda epoca, variando sostanzialmente molto poco nelle epoche successive.

6 Conclusioni

In questo progetto sono state sviluppate quattro diverse architetture di reti neurali con l'obiettivo di classificare correttamente la fase del sonno a partire dai segnali EEG in input. Il modello che ha ottenuto le migliori prestazioni, sia in termini di accuratezza sia di F1-score, è risultato il modello ibrido, che combina le capacità delle CNN di estrarre feature rilevanti dai dati con la capacità delle LSTM di cogliere le dipendenze temporali. La MLP è risultata il modello

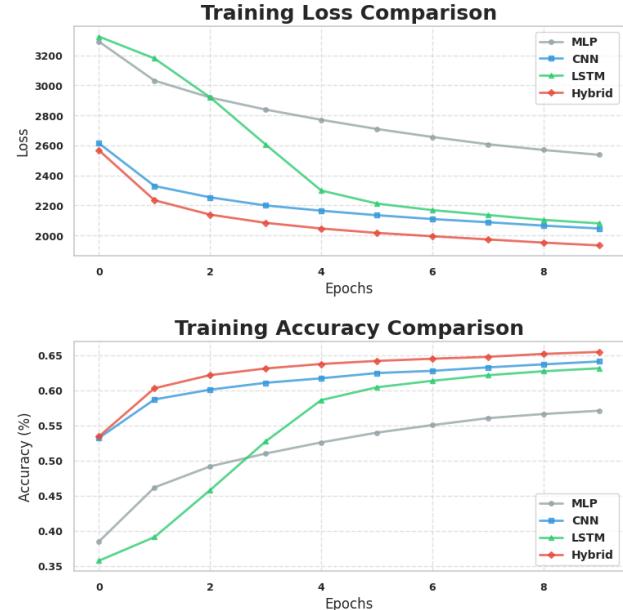


Figura 9: Accuratezza e Loss durante l'addestramento dei 4 modelli

meno performante, come ci si poteva aspettare, poiché la sua struttura non sfrutta le relazioni spaziali e temporali presenti nei dati. La CNN e la LSTM, invece, hanno mostrato prestazioni simili tra loro.

Dall'analisi dei risultati si è osservato che i segnali EEG permettono di distinguere più facilmente la fase di sonno profondo N3, caratterizzata da onde lente e di alta ampiezza, mentre le fasi N1, WAKE e REM risultano molto simili tra loro, rendendo la loro classificazione più complessa.

Per migliorare ulteriormente le prestazioni sarebbe utile includere segnali aggiuntivi come quelli già presenti nel dataset *Sleep-EDF Database Expanded* (cioè EOG, EMG, respirazione e temperatura corporea). Inoltre, in questo progetto è stato utilizzato solo circa il 30% dei dati disponibili. Disporre di maggiore potenza di calcolo e memoria permetterebbe di sfruttare l'intero dataset, ottenere risultati migliori e realizzare modelli più complessi con un numero maggiore di parametri.

Riferimenti bibliografici

- [1] Ary L. Goldberger et al. "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals". In: *Circulation* 101.23 (2000). RRID:SCR_007345, e215–e220.

- [2] Bob Kemp et al. “Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG”. In: *IEEE Transactions on Biomedical Engineering* 47.9 (2000), pp. 1185–1194.
- [3] (American Academy of Sleep Medicine). “The AASM manual for the scoring of sleep and associated events: rules, terminology, and technical specification”. In: (2007).
- [4] Time Series Classification Repository. *Sleep Dataset*. <https://www.timeseriesclassification.com/description.php?Dataset=Sleep>. Accessed: 2026-01-20. 2018.