

POLITECNICO DI TORINO

Corso di Laurea Magistrale in
Ingegneria Biomedica

Tesi di Laurea Magistrale

Deep neural approach for innovative analysis of electrocardiographic signals in patients with Brugada syndrome



**Politecnico
di Torino**

Relatori:

Prof. Filippo Molinari

Ing. Silvia Seoni

Candidato:

Francesca Maria Boero

Anno Accademico 2021/2022

SOMMARIO

ABSTRACT	6
CAPITOLO 1	7
INTRODUZIONE	7
CAPITOLO 2	9
SINDROME DI BRUGADA	9
2.1 DESCRIZIONE GENERALE	9
2.2 GENETICA	10
2.3 DIAGNOSI.....	10
2.4 TRATTAMENTO	12
CAPITOLO 3	13
RICERCA BIBLIOGRAFICA.....	13
CAPITOLO 4	15
MATERIALI E METODI.....	15
4.1 PRIMA FASE – SEGNALI PHYSIONET	16
4.1.1 Dataset	16
4.1.2 Target	17
4.1.3 Pre-processing.....	18
4.1.4 Divisione del dataset.....	18
4.1.5 Deep neural network.....	18
4.1.5.1 Architettura.....	18
4.1.5.2 Parametri di allenamento del modello	23
4.1.5.3 Allenamento del modello	27
4.1.6 Modello scelto: test con i segnali di Brugada	30
4.2 SECONDA FASE – SEGNALI DI BRUGADA	31

4.2.1 Dataset	31
4.2.2 Target	33
4.2.3 Pre-processing.....	34
4.2.4 Divisione del dataset.....	34
4.2.5 Deep neural network.....	35
4.2.5.1 Architettura.....	35
4.2.5.2 Parametri di allenamento del modello	36
4.2.5.3 Allenamento del modello	37
4.2.6 Algoritmo di post-processing: calcolo dell'isoelettrica e delle aree sottese ai tratti ST	37
4.2.7 Analisi delle aree sottese ai tratti ST	39
4.2.8 Algoritmo di validazione manuale	40
4.2.8.1 Validazione baseline.....	42
4.2.8.2 Validazione tratti ST.....	43
CAPITOLO 5.....	45
RISULTATI	45
5.1 RISULTATI PRIMA FASE	45
5.1.1 Risultati prove di allenamento	45
5.1.2 Risultati di riconoscimento ottenuti con il modello scelto	56
5.1.2.1 Esempi delineazione segnali Physionet.....	56
5.1.2.2 Esempi delineazione segnali di Brugada.....	58
5.2 RISULTATI SECONDA FASE	59
5.2.1 Risultati allenamento del modello con i segnali di Brugada	59
5.2.2 Risultati di riconoscimento ottenuti con i tre modelli allenati.....	61
5.2.3 Risultati calcolo isoelettrica e aree sottese ai tratti ST	65
5.2.4 Risultati validazione manuale	68
5.2.4.1 Risultati validazione baseline	68
5.2.4.2 Risultati validazione tratti ST	69

5.2.5 Risultati analisi aree sottese	72
CAPITOLO 6	75
DISCUSSIONE	75
CAPITOLO 7	77
CONCLUSIONI.....	77
RINGRAZIAMENTI	79
BIBLIOGRAFIA	80
LISTA DELLE FIGURE	84
LISTA DELLE TABELLE	88

ABSTRACT

Brugada Syndrome is a rare inherited disorder of cardiac electrophysiology that causes an increased risk of syncope and cardiac arrest in subjects with structurally normal hearts. Described as a new clinical entity in 1992, it is now widely recognized and is estimated to be the cause of at least 4% of all sudden deaths. Affected patients may be totally asymptomatic or exhibit symptoms due to arrhythmias. The syndrome presents a characteristic electrocardiographic pattern, with ST segment elevation in the right precordial leads (from V1 to V3). The only proven effective therapy is the placement of an implantable defibrillator, although alternatives such as drug therapy or epicardial ablation of the arrhythmic substrate are being studied. A timely diagnosis is therefore crucial, but this is sometimes difficult, since the first symptom is often sudden death, and the characteristic electrocardiographic pattern is highly variable over time. The aim of this thesis work is the development of an automatic algorithm for the detection of the typical pattern in ECG signals acquired in patients with Brugada Syndrome that could support the clinician in recognizing and analysing them. The ECG signals were acquired in three different conditions: before surgery, during administration of the ajmaline drug and after epicardial ablation of the right ventricle. To achieve the goal, a deep neural network was implemented, for the automatic detection of waves and segments in ECG signals, especially the ST and TP segments. These segments were subsequently used to compute the isoelectric and the area underlying the ST segment, that are useful features for diagnosing the pathology. Finally, the model was validated using a ground truth done by two operators. Furthermore, the areas under the ST segments were analysed to verify if there were statistically differences among the three conditions.

CAPITOLO 1

INTRODUZIONE

La Sindrome di Brugada è una patologia cardiaca descritta trent'anni fa, associata a disturbi dell'attività elettrica che provocano aritmie spesso fatali. I meccanismi alla base non sono ancora stati definiti completamente, ma è stata identificata un'anomalia genetica a carico dei canali del sodio cardiaci in un terzo dei pazienti affetti. I sintomi, quando presenti, sono correlati a manifestazioni aritmologiche, che in alcuni casi portano all'arresto cardiaco. I soggetti colpiti sono principalmente di sesso maschile, di età media tra i trenta e i quarant'anni. Dal punto di vista del segnale elettrocardiografico (ECG) la sindrome presenta un'estrema variabilità, anche durante la stessa giornata, ma è caratterizzata dal sopraelevamento del tratto ST nelle derivazioni precordiali V1 e V2. Per poter riconoscere il pattern tipico della malattia i medici necessitano una corretta formazione, senza considerare il fatto che a volte le opinioni sulla sua presenza nel tracciato ECG possono essere discordanti [1].

La patologia, per quanto appena discusso, spesso è difficilmente diagnosticabile, impedendo agli individui affetti di ottenere trattamenti adeguati e ponendoli a rischio della vita. I soggetti maggiormente colpiti sono ancora in giovane età, la comparsa di aritmie può pregiudicare il loro stile di vita, senza contare che il primo sintomo della patologia è spesso la morte improvvisa. Risulta quindi necessario svolgere ricerche per acquisire nuove conoscenze e migliorare la sensibilità e la specificità della diagnosi. Negli ultimi anni sono aumentati gli articoli relativamente all'utilizzo della tecnica del deep learning nel riconoscimento di aritmie nei segnali ECG, dove le prestazioni raggiunte sono promettenti, simili a quelle dei medici [2]. Recentemente, inoltre, è stato pubblicato un report che utilizza una deep neural network per la diagnosi dei pazienti affetti dalla Sindrome di Brugada [1].

In questo contesto si inserisce il seguente lavoro di tesi, il quale si propone l'obiettivo di implementare, proprio tramite deep learning, un algoritmo per l'identificazione e l'analisi quantitativa del pattern caratteristico della sindrome in pazienti patologici, per supportare i clinici durante la diagnosi. L'algoritmo, in particolare, è in grado di calcolare l'isoelettrica e le aree sottese ai tratti ST dei segnali elettrocardiografici. Lo studio è stato svolto in collaborazione con i cardiologi e gli ingegneri dell'A.O. Mauriziano di Torino, i quali hanno fornito i segnali

ECG di pazienti patologici in tre condizioni (situazione di base, durante la somministrazione di farmaco e subito dopo l'intervento di ablazione cardiaca).

Durante la prima parte del lavoro è stata implementata la rete neurale artificiale dell'articolo [3], la quale permette la delineazione delle onde nei segnali ECG, utilizzando i segnali del database QT contenuto nella banca dati Physionet. Dopo aver svolto diverse prove di allenamento, il modello migliore è stato utilizzato per il riconoscimento delle onde nei segnali dei pazienti affetti dalla sindrome di Brugada. Tuttavia, poiché i risultati non sono stati considerati soddisfacenti, la deep neural network è stata modificata e adattata all'allenamento con i segnali ECG forniti dall'ospedale. I risultati ottenuti sono stati utilizzati da un algoritmo di post-processing per il calcolo dell'isoelettrica, trovata come media delle mediane dei tratti TP dei vari battiti cardiaci, e delle aree sottese ai tratti ST identificati dalla rete. I valori di questi parametri sono poi stati validati, con una "ground truth" ricavata manualmente sui tracciati elettrocardiografici da due operatori esperti. A tal fine è stata creata una routine per permettere la selezione manuale dei tratti TP e ST sui segnali ECG e, poi, sono stati calcolati l'errore assoluto di riconoscimento dell'isoelettrica (baseline) e accuracy, precision e recall per l'identificazione dei segmenti ST. Infine, è stata svolta un'analisi quantitativa dei valori delle aree, calcolando i valori medi e la deviazione standard nelle tre condizioni, per evidenziare eventuali differenze statisticamente significative.

CAPITOLO 2

SINDROME DI BRUGADA

2.1 DESCRIZIONE GENERALE

La Sindrome di Brugada è una rara malattia cardiaca che provoca un aumento del rischio di sincope o morte improvvisa in soggetti con cuore strutturalmente normale [4]. Identificata inizialmente nel 1988, grazie alla ricerca di Andrea Nava, Bortolo Martini e colleghi [5][6], viene definita come nuova entità clinica solo nel 1992, dagli elettrofisiologi Pedro e Josep Brugada [7], dai quali prende il nome.

La patologia colpisce in modo più frequente gli uomini delle donne, con un rapporto di 8:1, specialmente tra la terza e la quarta decade di vita, mentre difficilmente si manifesta durante l'infanzia. Queste differenze potrebbero essere dovute a influenze di carattere ormonale [4].

Gli individui affetti dalla sindrome possono essere asintomatici o manifestare sintomi dovuti ad aritmie cardiache, come sincope, palpitazioni associate a malessere e respiro agonico notturno. In alcuni casi possono anche manifestarsi convulsioni ed episodi confusi con attacchi epilettici. In un'alta percentuale di soggetti il primo sintomo è la morte cardiaca improvvisa, causata da aritmie ventricolari fatali, ovvero tachicardia ventricolare polimorfa (PVT) e fibrillazione ventricolare (VF). L'insorgenza dei sintomi è maggiore a riposo, dopo pasti abbondanti, nella fase di recupero da uno sforzo fisico o durante il sonno notturno. Inoltre, esistono fattori in grado di innescare i sintomi e svelare la presenza della Sindrome di Brugada; tra questi troviamo febbre alta, alterazioni delle concentrazioni di potassio e calcio, consumo eccessivo di bevande alcoliche e farmaci (come beta-bloccanti e antiaritmici di classe I) [8].

La patologia è considerata la causa di più del 4% di tutte le morti improvvise e, tra queste, di più del 20% di decessi di individui con un cuore privo di anomalie strutturali. La sua prevalenza nella popolazione mondiale varia da 5 a 20 casi ogni 10,000 individui (cioè 0.05-0.2%); è diffusa con frequenza più bassa nei paesi occidentali e più alta in Asia sud-orientale, dove è chiamata SUNDS (sudden unexplained nocturnal death syndrome). L'incidenza della malattia è comunque considerata sottostimata, data l'estrema variabilità del tracciato

elettrocardiografico nel tempo e il fatto che la morte improvvisa è spesso la prima manifestazione clinica.

2.2 GENETICA

La Sindrome di Brugada è una malattia genetica a trasmissione autosomica dominante, ovvero ad ogni gravidanza di un paziente patologico la probabilità di trasmissione alla prole è del 50%. La prima mutazione, scoperta alla fine degli anni Novanta, è stata identificata nel gene SCN5A, che codifica le subunità α della proteina canale del sodio sulla superficie delle cellule cardiache. Tale anomalia determina uno squilibrio del flusso di sodio nei cardiomiociti, accorciando così il loro normale potenziale d'azione. Negli anni seguenti sono state scoperte più di 350 mutazioni patologiche a carico di diversi geni, codificanti le proteine costituenti i canali di sodio, potassio e calcio delle cellule cardiache. Tuttavia, l'origine genetica della patologia è stata individuata solo nel 35% dei pazienti affetti, di cui il 30% presenta un'alterazione nel gene SCN5A. I fattori che possono spiegare l'alto numero di pazienti affetti dalla malattia ma privi di alterazioni genetiche possono essere: variazione del numero di copie del gene SCN5A, mutazioni a carico di geni non considerati o fattori epigenetici (modifiche post-traduzionali o meccanismi legati all'RNA) [4].

Nonostante questo, lo screening genetico effettuato tramite prelievo ematico rimane molto importante in un sospetto caso clinico di Sindrome di Brugada, sia per la stratificazione del rischio nel paziente che per l'identificazione di altri individui patologici in famiglia.

2.3 DIAGNOSI

Il tracciato elettrocardiografico di un paziente affetto dalla sindrome di Brugada presenta un'enorme variabilità, durante il corso della giornata il segnale può passare molte volte da normale a patologico e viceversa.

La patologia è caratterizzata da tre quadri ECG, di cui solo il Tipo 1 viene considerato come diagnostico (Figura 1):

- Tipo 1 - sopraslivellamento del punto J superiore a 2 mm (morfologia detta "a tenda"), discesa graduale del segmento ST con onda T negativa in una o più delle derivazioni precordiali destre (V1, V2 e V3);

- Tipo 2 - presenta il cosiddetto “*saddle-back pattern*”, con sopraelevamento del punto J superiore a 2 mm e sopraslivellamento del tratto ST maggiore di 1 mm, con onda T che può essere positiva o bifasica;
- Tipo 3 - presenta il cosiddetto “*saddle-back pattern*”, con sopraelevamento del punto J inferiore a 2 mm, sopraslivellamento del tratto ST minore di 1 mm e onda T positiva.

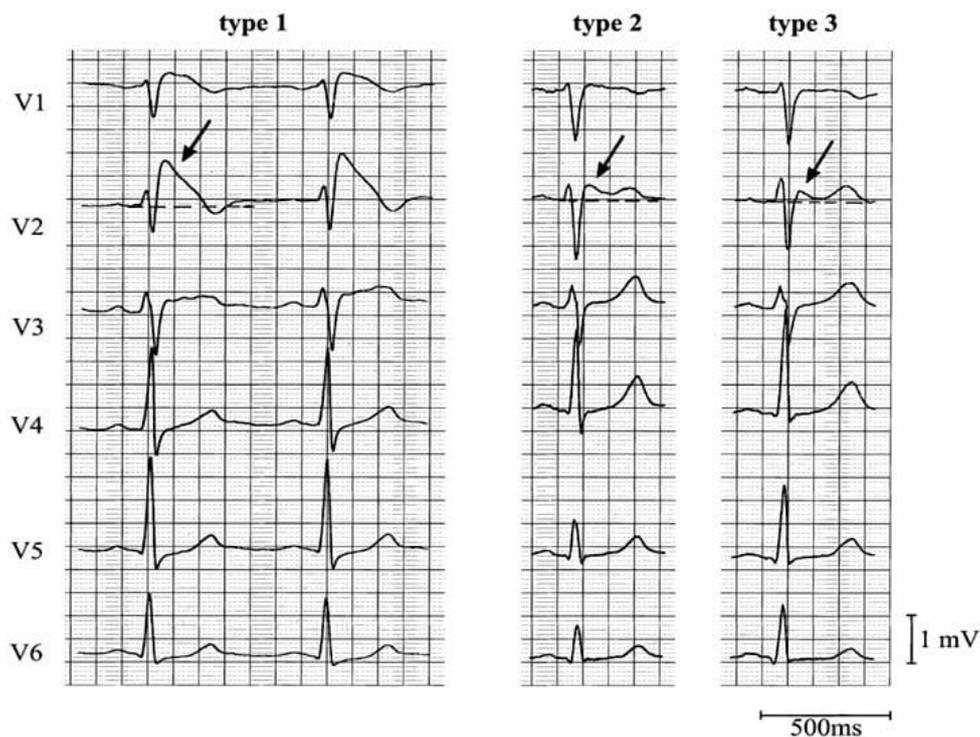


Figura 1. Pattern caratteristici (tipi 1, 2 e 3) della Sindrome di Brugada nelle derivazioni V1-V6.
Fonte: lifeinthefastlane.com

In alcuni casi, l'utilizzo delle derivazioni precordiali alte, cioè il posizionamento degli elettrodi nel 2° e 3° spazio intercostale invece che nel 4°, può aumentare la probabilità di individuare l'eventuale pattern caratteristico. In questo modo, infatti, gli elettrodi vengono posti esattamente al di sopra della regione del ventricolo destro interessata dal fenomeno di Brugada.

Quando è presente un quadro ECG di tipo 2 o 3 viene proposta l'esecuzione di un test provocativo, il quale prevede la somministrazione di farmaci antiaritmici come flecainide o ajmalina, per rivelare l'eventuale presenza del tipo 1 e arrivare alla diagnosi. Il test viene effettuato in presenza di personale medico preparato e tramite continua osservazione del tracciato ECG, poiché in alcuni casi ad alto rischio potrebbe determinare l'insorgenza di aritmie.

La sindrome è definita completamente quando il paziente presenta anche un membro della famiglia deceduto in modo improvviso in età inferiore a 45 anni o ECG di tipo 1, sintomi che possono essere correlati con una qualche aritmia o aritmie ventricolari (PVT o VF).

2.4 TRATTAMENTO

Dopo che il paziente è stato diagnosticato come affetto dalla Sindrome di Brugada, la scelta del successivo trattamento dipende dal suo livello di rischio. L'impianto di un defibrillatore cardioversore è l'unica strategia ad oggi considerata efficace per la prevenzione della morte improvvisa nei soggetti patologici. Tuttavia, il tasso di gravi complicazioni, come ad esempio degli shock inappropriati, è alto. I rischi, poi, sono maggiori per i bambini, a causa del piccolo diametro dei vasi sanguigni e di eventuali interruzioni dei cavi che possono verificarsi durante la crescita. Inoltre, i costi della terapia ICD non sono sostenibili per i pazienti residenti in zone del mondo più povere [9].

Altre opzioni di trattamento prevedono l'uso di agenti farmacologici, come la chinidina o l'idrochinidina, per ridurre il rischio di aritmie ventricolari. Gli svantaggi della somministrazione orale includono effetti collaterali gastrointestinali e predisposizione allo sviluppo di sindrome del QT lungo.

Un altro trattamento attuabile è l'ablazione epicardica transcatetere con radiofrequenza; quest'ultima consiste nell'individuare ed eliminare, tramite un catetere introdotto nel cuore, l'area responsabile dei potenziali elettrici anomali, solitamente posizionata sulla superficie esterna del tratto di efflusso del ventricolo destro. Tramite questa operazione si mira ad eliminare i battiti ventricolari prematuri che possono portare alla fibrillazione ventricolare e quindi al decesso del paziente.

In ogni caso, la progressione clinica della patologia non è definita, pertanto risulta necessario svolgere dei controlli cardiologici periodici, per verificare che non siano insorti nuovi parametri e valutare i trattamenti futuri.

CAPITOLO 3

RICERCA BIBLIOGRAFICA

Dalla sua definizione come nuova entità clinica, avvenuta trent'anni fa, le informazioni riguardanti la Sindrome di Brugada sono aumentate in modo esponenziale. L'impegno nella ricerca è stato promosso dalle caratteristiche della patologia, il cui primo sintomo è spesso la morte improvvisa, per cercare di prevenire e ridurre i casi con un drammatico epilogo.

Un primo Consensus Report pubblicato nel 2002 ha definito i criteri diagnostici per la sindrome [10], mentre un secondo, nel 2005, ha focalizzato l'attenzione sulla stratificazione del rischio e sull'approccio alla terapia [11]. Negli ultimi due decenni sono stati pubblicati molti articoli relativamente ai progressi svolti nella comprensione della presentazione clinica, della base genetica e molecolare e nei trattamenti della patologia [4] [8] [9][12]. Tuttavia, in letteratura, poco è presente a proposito dello studio e dell'analisi dei pattern tipici della malattia, anche in previsione di un aiuto al clinico nella diagnosi.

Degno di nota è un articolo pubblicato recentemente sul "Canadian Journal of Cardiology", riguardante lo sviluppo, tramite deep learning, di una rete neurale artificiale in grado di identificare in modo automatico i pazienti affetti da sindrome di Brugada a partire dal loro segnale elettrocardiografico [1]. I ricercatori Liu et al. hanno utilizzato un dataset costituito da un ugual numero di segnali ECG patologici e sani. La rete neurale sviluppata è stata prima allenata per la classificazione del blocco di branca destra (RBBB) e poi, tramite transfer learning, per identificare il pattern di Brugada di tipo 1. Successivamente, gli output della rete sono stati comparati con le diagnosi di cardiologi certificati e un dataset indipendente è stato utilizzato per la validazione delle performance.

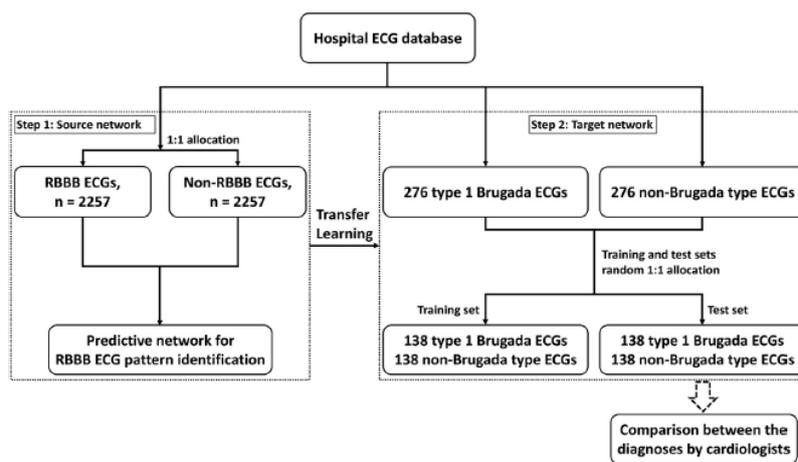


Figura 2. Schema della divisione del dataset e del transfer learning. Liu et al [1]

Per il test set del dataset originario, l'AUC (area under the receiver operating characteristic curve) è stata pari a 0.96, la sensibilità pari a 88.4% e la specificità pari a 89.1%. Nel confronto con i cardiologi la rete ha superato le loro performance e per il dataset indipendente le performance si sono ridotte ma sono rimaste soddisfacenti, dimostrando buona capacità di generalizzazione.

Oltre ad essere l'unico articolo di rilievo nel contesto citato precedentemente, poiché permette il riconoscimento dei segnali ECG patologici, il lavoro di Liu et al. mostra, in maniera innovativa, l'utilizzo del moderno deep learning. Tuttavia, esso non tiene in considerazione i casi apparentemente non sospetti, cioè quelli che possono essere svelati solo da un test farmacologico.

Il presente lavoro di tesi, invece, propone l'analisi quantitativa del pattern caratteristico di Brugada non solo in condizioni patologiche normali, ma anche durante il test farmacologico provocativo con ajmalina e immediatamente dopo l'intervento di ablazione, uno dei trattamenti sperimentati negli ultimi anni per normalizzare i tracciati ECG. In questo modo è possibile studiare le differenze nelle tre condizioni e scoprire nuove informazioni utili alla diagnosi della sindrome. Per raggiungere l'obiettivo è stato progettato un algoritmo automatico tramite deep learning; il lavoro risulta quindi innovativo rispetto a quanto pubblicato precedentemente.

CAPITOLO 4

MATERIALI E METODI

Il percorso generale seguito può essere schematizzato nel seguente modo:

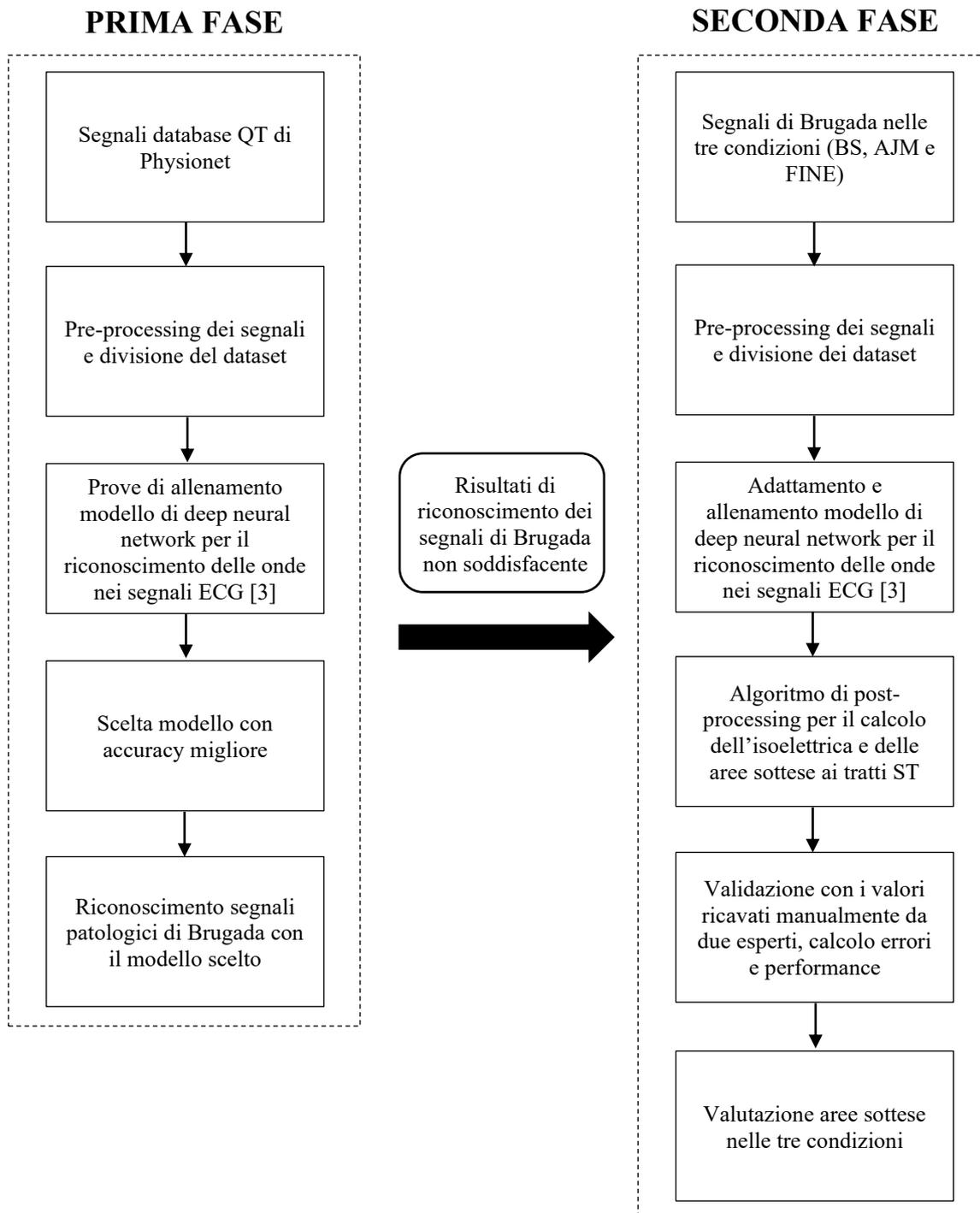


Figura 3. Schema del lavoro svolto

Inizialmente è stata svolta una ricerca bibliografica, prima sulla classificazione [2][13][14] e successivamente sulla detection di pattern [3][15][16][17][18][19] nei segnali biologici, di articoli incentrati sull'utilizzo di tecniche di machine e deep learning.

I report considerati maggiormente utili sono stati quelli relativi alla delineazione del segnale ECG [3][17][18][19], ovvero all'identificazione di inizio, picco e fine delle diverse onde. Per la quantificazione dei pattern caratteristici della Sindrome di Brugada, infatti, risulta fondamentale il riconoscimento dei segmenti ST e TP. Tra gli articoli scelti è stato deciso di implementare e adattare al nostro caso la deep neural network sviluppata da Peimankar e Puthusserypady nell'articolo "DENS-ECG: A deep learning approach for ECG signal delineation" [3].

In una prima fase del lavoro è stata implementata la deep neural network sopra citata con l'utilizzo dei segnali della banca dati Physionet. Sono state eseguite diverse prove di allenamento ed è stato scelto il modello che ha fornito i risultati migliori. Tuttavia, questo modello non ha mostrato risultati soddisfacenti nella delineazione dei segnali ECG patologici di Brugada forniti dall'A.O. Mauriziano. Allora, in una seconda fase, la rete neurale è stata modificata ed allenata con questi segnali. I risultati sono poi stati utilizzati per il calcolo dell'isoelettrica e delle aree sottese ai tratti ST e sono stati validati con i riconoscimenti svolti manualmente. Infine, è stata svolta una valutazione quantitativa delle aree sottese.

In seguito, verranno prima descritti i materiali e i metodi della prima fase, e poi quelli della seconda.

4.1 PRIMA FASE – SEGNALI PHYSIONET

4.1.1 Dataset

Secondo quanto definito nell'articolo [3], il dataset per l'allenamento e la validazione della rete neurale è il QT database (QTDB) della banca dati Physionet. Esso è stato formato prelevando i records di altri sette databases, tra cui il MIT-BIH e lo European Society of Cardiology ST-T database, ottenendo così una certa varietà nella morfologia delle onde dei segnali ECG. I segnali sono patologici, provenienti da soggetti aritmici, un insieme di maschi e femmine di età compresa tra 20 e 88 anni. I records totali sono 105, ciascuno a 2 canali Holter ECG, ognuno dalla durata di 15 min e con frequenza di campionamento pari a 250 Hz. Ogni record possiede

diversi file: un file di testo “.hea” contenente le informazioni utili e il tipo di formato, un file binario “.dat” con i due canali e fino a nove file binari con le annotazioni di inizio, picco e fine delle onde nei segnali ECG [20]. I file utilizzati per l’allenamento sono il “.dat”, i file “.pu0” e “.pu1”, contenenti le annotazioni ottenute con l’algoritmo ecgpuwave in modo automatico e indipendente per i due canali ECG, il file “.pu”, simile ai precedenti ma con le annotazioni basate su entrambi i canali, e i file “.q1c” e “.q2c”, contenenti le annotazioni svolte da due cardiologi su battiti selezionati. Per leggere i file citati è stato utilizzato il pacchetto Python WFDB [21].

4.1.2 Target

In questo caso i target scelti sono le onde del segnale ECG: isoelettrica, onda P, complesso QRS e onda T. I target sono stati ricavati dai file di annotazioni. I file sono stati utilizzati in due modi, a seconda delle prove di allenamento del modello svolte:

- per ogni segnale ECG sono state scelte le annotazioni automatiche (file “.pu0” e “.pu1”) che mostravano il valore della metrica Dice Similarity maggiore per le varie onde quando confrontate con le annotazioni manuali (file “.q1c” e “.q2c”);
- per ogni segnale ECG sono state utilizzate le annotazioni automatiche contenute nel file “.pu”.

I target ricavati dalle annotazioni scelte sono poi stati codificati, associando a ciascuno un valore numerico. I target sono relativi ai campioni del segnale: ad ogni campione del segnale è stato associato il valore numerico che indica la sua appartenenza al tratto ECG corrispondente.

Onda/segmento	Codifica numerica
Isoelettrica (No wave, NW)	0
Onda P	1
Complesso QRS	2
Onda T	3

Tabella 1. Codifica numerica applicata ai target scelti

La Dice Similarity, sopra citata, è calcolata nel seguente modo:

$$Dice (Similarity) = 2 * \frac{X \cap Y}{X + Y} \quad (1)$$

Con X pari all'annotazione manuale e Y pari all'annotazione automatica. Si tratta di una metrica che, in questo caso, definisce un criterio generale di somiglianza tra le due annotazioni.

4.1.3 Pre-processing

I due segnali ECG contenuti in ogni record sono stati filtrati con un filtro passa-banda di Butterworth di ordine 3, con frequenze di taglio pari a 0.5 e 40 Hz, per attenuare la fluttuazione della linea base e i rumori ad alta frequenza. Il filtraggio è stato anticausale, per compensare la fase non lineare del filtro ed evitare la distorsione dei segnali ECG.

Ogni segnale è stato standardizzato, rimuovendo la media dei campioni del segnale e dividendo per la deviazione standard.

Infine, ogni segnale è stato suddiviso in segmenti da 1000 campioni l'uno, corrispondenti a circa cinque battiti cardiaci, per allenare il modello sulla continuità della serie temporale e ottenere un risultato più robusto. I segmenti così creati costituiscono l'input della rete neurale.

4.1.4 Divisione del dataset

Il dataset è stato diviso in training set (TRS), utilizzato per l'allenamento e l'ottimizzazione della rete neurale artificiale, e test set (TS), utilizzato per la validazione. La divisione è stata svolta a livello dei records, facendo in modo che questi fossero unici in entrambi i set: 84 records sono stati inseriti nel training e 21 nel test set.

	N° records	(N° segmenti, N° campioni)
TRS	84	(37800,1000)
TS	21	(9450,1000)

Tabella 2. Descrizione divisione del dataset

4.1.5 Deep neural network

4.1.5.1 Architettura

Il modello di deep neural network implementato da [3] è costituito da otto layers in tutto: un layer di input, tre layers convoluzionali monodimensionali, due layers BiLSTM, un layer di dropout e un layer dense time-distributed. Il modello è quindi costituito dalla combinazione di reti neurali convoluzionali (CNN) e reti di memoria a lungo termine (LSTM), due strutture che apprendono caratteristiche complesse differenti dai segnali di input durante l'allenamento, permettendo così migliori performance. Vengono ora definite in maniera più dettagliata le funzioni dei diversi layers.

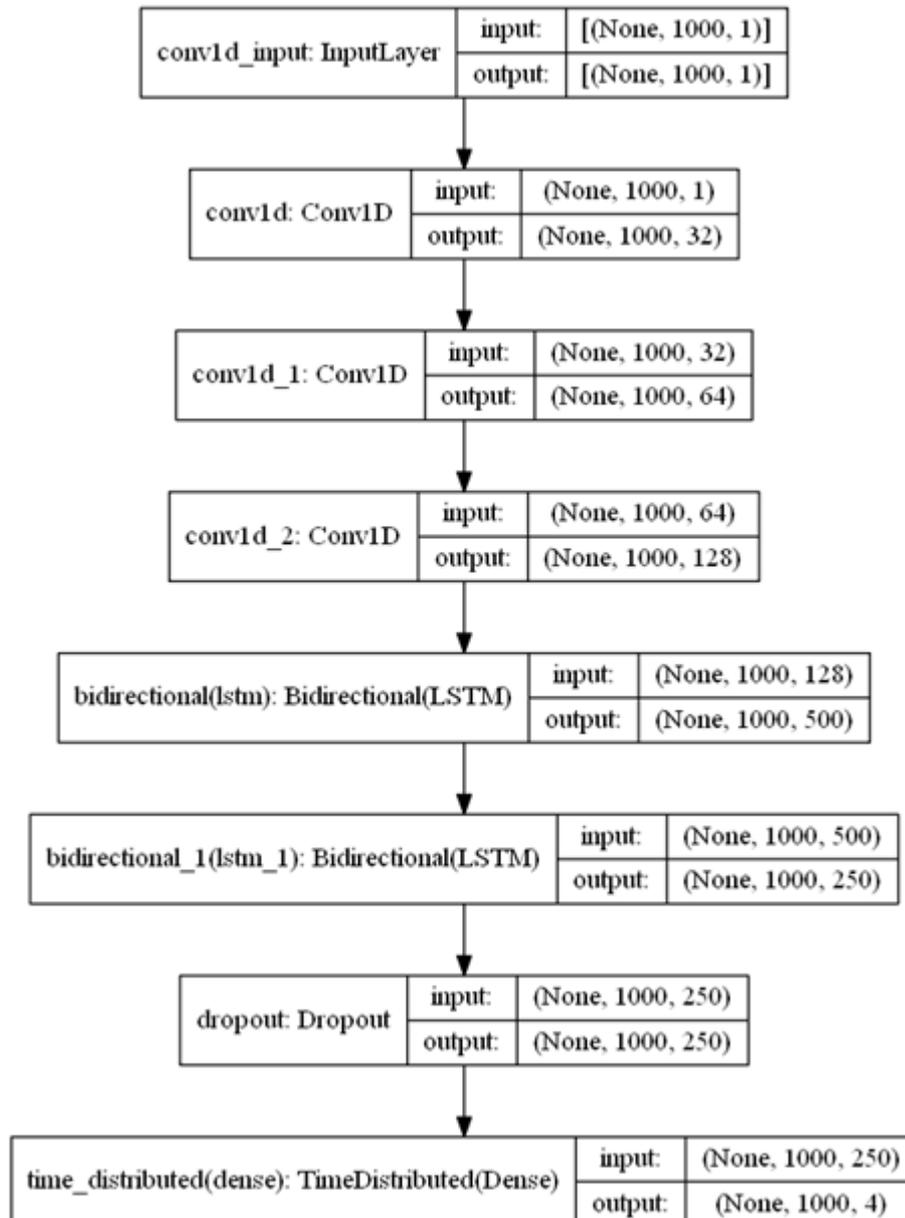


Figura 4. Architettura deep neural network implementata da [3]

Le 1D CNN costituiscono un caso particolare delle classiche CNN (Convolutional Neural Network), progettate per lavorare a due dimensioni e in grado di sfruttare qualsiasi pattern spaziale e temporale presente nelle immagini fornite in input [22]. In questo caso le reti vengono applicate a serie temporali (da qui 1D), risultando quindi ottimali per l'elaborazione dei segnali elettrocardiografici. Il vantaggio del loro utilizzo è che possono imparare direttamente dai dati a disposizione. Per svolgere le loro funzioni le CNN possiedono quattro attributi fondamentali:

1. connessioni locali: ogni neurone è connesso a quelli del livello precedente solo localmente, le sue elaborazioni sono perciò locali
2. pesi condivisi: i pesi vengono condivisi a gruppi, perciò neuroni diversi elaborano allo stesso modo parti diverse dell'input
3. elevato numero di layers e filtri
4. ridotta complessità della rete

L'architettura di una CNN è gerarchica, formata da un livello di input connesso ai dati, diversi livelli intermedi con connessioni locali e pesi condivisi e ultimi livelli fully-connected con funzione simile ad un Multilayer perceptron. L'operazione principale che viene svolta è quella di convoluzione: ogni valore di output è ottenuto dal prodotto scalare tra una matrice di pesi (kernel), fatta scorrere sui dati di input, e la porzione di input coperta. Nel caso delle 1D CNN il kernel scorre solo lungo la direzione temporale. Gli output ottenuti sono le features estratte dai dati di ingresso, utilizzate per creare le features maps, su cui opererà il livello successivo. La dimensione della matrice è detta "dimensione del kernel" e viene indicata come k_{size} . Il numero di features maps in output è pari al numero di filtri applicati, ognuno con k_{size} pesi condivisi e un bias, appresi durante la fase di allenamento tramite l'algoritmo di backpropagation. Il processo di convoluzione può essere spiegato con la seguente formula [23]:

$$a_{ij}^m = \varphi(b_i + \sum_{k=1}^{k_{size}} w_{ik} x_{j+k-1}) = \varphi(b_i + w_i^T x_j) \quad (2)$$

Con a_{ij}^m pari all'output del j-esimo neurone dell'i-esimo filtro del m-esimo layer convoluzionale, φ funzione di attivazione, b_i e w_i bias e vettore con i pesi condivisi dell'i-esimo filtro e x_j input in numero pari a k_{size} .

Nel nostro caso sono stati utilizzati tre layers convoluzionali, ciascuno con $k_{size} = 3$ e numero

di filtri pari, rispettivamente, a 32, 64 e 128. Per mantenere invariata la dimensione dei dati di input nei tre layers consecutivi è stato applicato lo zero-padding, ovvero sono stati aggiunti $k_{size}/2$ valori pari a 0 (o $(k_{size}-1)/2$ se k_{size} è dispari) ad ogni estremità del segnale. Perciò, l'output di ogni layer sarà dato da una sequenza di features pari al numero di filtri applicati, con la stessa dimensione dei segnali di input. L'ultimo livello fornisce in uscita 128 features maps, che verranno poi date in ingresso ai layers BiLSTM. Questa prima parte del modello è utile per l'estrazione dei pattern temporali (sotto forma di features complesse) dai segmenti ECG.

I due layers successivi sono reti LSTM (Long Short-Term Memory), le quali costituiscono un tipo particolare di RNN (Recurrent Neural Networks), adatto a processare dati di serie temporali anche in lunghe sequenze. Sono state introdotte per la prima volta nel 1997, da Hochreiter e Schmidhuber, per risolvere il problema del gradiente evanescente [24]. Le RNN sono reti neurali feedforward equipaggiate con una memoria interna per apprendere le dipendenze a lungo e breve termine nell'elaborazione di segnali sequenziali. Tuttavia, sebbene le reti abbiano successo con la memoria a breve termine, le dipendenze a lungo termine sono impedito dalla scomparsa del gradiente. Durante l'allenamento con backpropagation, infatti, i gradienti a lungo termine retro-propagati possono diventare sempre più piccoli fino a tendere a zero, portando così ad un apprendimento nullo [25]. Questo inconveniente è stato risolto con lo sviluppo delle LSTM, costituite da moduli ripetuti di reti neurali, ognuno formato da quattro layers interagenti. I moduli sono collegati tra loro dallo "stato della cella", contenente le informazioni. Ogni modulo può aggiornare e controllare lo stato della cella tramite tre strutture dette gates, utilizzando le funzioni di attivazione tangente iperbolica e sigmoide. Essi sono:

- forget gate: controlla quali informazioni devono essere eliminate dalla cella. Esso è costituito dalla funzione di attivazione sigmoide e svolge la seguente operazione

$$f_n = \varphi(b_f + u_f^T a_n + w_f^T h_{n-1}) \quad (3)$$

- input gate: controlla quali informazioni devono essere aggiunte alla cella. Il suo output è ottenuto dalla moltiplicazione puntuale degli output di due layers. Il primo sceglie quali valori aggiungere alla cella, ha funzione di attivazione sigmoide e svolge la seguente operazione

$$i_n = \varphi(b_i + u_i^T a_n + w_i^T h_{n-1}) \quad (4)$$

Il secondo layer crea un vettore con i valori da aggiungere, ha funzione di attivazione tangente iperbolica e svolge la seguente operazione

$$\tilde{c}_n = \tanh(b_c + u_c^T a_n + w_c^T h_{n-1}) \quad (5)$$

- output gate: controlla quali informazioni contenute nello stato della cella devono essere fornite in output. Esso è costituito dalla funzione di attivazione sigmoide e svolge la seguente operazione

$$o_n = \varphi(b_o + w_o^T a_n + u_o^T h_{n-1}) \quad (6)$$

Per aggiornare lo stato della cella basta mettere insieme i risultati dei primi due gates nel seguente modo

$$c_n = f_n c_{n-1} + i_n \tilde{c}_n \quad (7)$$

Le informazioni ottenute dall'output verranno fornite come input per lo step successivo in forma filtrata, svolgendo la seguente operazione

$$h_n = o_n \tanh(c_n) \quad (8)$$

Nelle formule riportate (equazioni 3 - 8) a_n è la sequenza in ingresso alla rete LSTM al passo n , in questo caso pari all'output dell'ultima CNN implementata precedentemente, e h_{n-1} è l'output al passo $n-1$; u_f, w_f, u_i, w_i, u_o e w_o sono i vettori dei pesi e b_f, b_i e b_o sono i termini di bias [26].

Le reti BiLSTM (Bidirectional LSTM) sono una variante delle reti appena spiegate, in grado di elaborare la sequenza temporale dei dati nelle due direzioni. Sono costituite da due LSTM, una che processa i dati da destra a sinistra e l'altra che li processa da sinistra a destra, e da un layer di output. In questo modo è possibile sfruttare le dipendenze passate e anche quelle future [27]. Il modello che è stato implementato è costituito da due layers BiLSTM, il primo con 250 unità nascoste di memoria e il secondo con 125 unità nascoste di memoria. Questa seconda parte del

modello è utile per processare le features complesse estratte dai precedenti layers convoluzionali.

Il successivo layer di Dropout viene utilizzato per prevenire il problema dell'overfitting durante l'allenamento della deep neural network, migliorando così le capacità di generalizzazione [28]. L'operazione che viene svolta è quella di rimuovere "apparentemente" alcune unità dalla rete insieme alle loro connessioni, cioè di non considerarle durante l'allenamento. Ogni unità ha associata una probabilità p di essere mantenuta: il processo di quali unità verranno rimosse è perciò casuale. Allenare una rete con n unità con l'utilizzo di un layer di Dropout è, quindi, come allenare 2^n reti con unità rimosse. Con questa premessa, il processo di calcolo della predizione media durante la fase di test risulta difficoltoso. Per evitare il problema, il Dropout layer non viene utilizzato durante il test, ma i pesi di ogni unità della rete sono moltiplicati per la p associata durante il training. Nel modello implementato la probabilità p è pari a 0.2.

Il Dense layer è un layer fully-connected, solitamente posto al termine del modello di rete neurale, dove fornisce i risultati di predizione. Esso viene utilizzato per cambiare la dimensione dei dati forniti dal layer precedente, svolgendo delle operazioni di moltiplicazione vettore-matrice [29]. Nel modello implementato esso è costituito da 4 unità nascoste, tante quante le classi da predire (onda P, complesso QRS, onda T e isoelettrica), e utilizza la funzione di attivazione "softmax", la quale fornisce, per ogni campione temporale, le probabilità di appartenenza alle 4 classi considerate. Poiché la somma delle probabilità è pari ad 1, esse possono essere interpretate come probabilità a posteriori. Ogni campione è associato alla classe con la probabilità più alta. In formule:

$$\hat{y}_i = c_j \quad \text{se} \quad P(y_i = c_j | x_i) = \operatorname{argmax} P(c_1, \dots, c_k | x_i) \quad (9)$$

Con \hat{y}_i la classe predetta dell' i -esimo campione x_i e P la probabilità a posteriori.

Il layer è TimeDistributed, per fare in modo che le trasformazioni siano applicate ad ogni passo temporale, mantenendo così la continuità temporale dei segnali ECG.

4.1.5.2 Parametri di allenamento del modello

Per poter utilizzare il modello di deep neural network appena definito è necessario che questo sia allenato, così da trovare la giusta combinazione di pesi e biases.

Il processo di allenamento consiste in un problema di ottimizzazione non convesso, nel nostro caso tradotto nella minimizzazione, attraverso un processo iterativo, di una “cost function”, funzione che fornisce l’errore di predizione del modello comparando gli output predetti con quelli veri previsti [23][30][31]. Poiché il nostro obiettivo è una classificazione multiclasse (ogni campione può appartenere ad una sola categoria su 4), è stata scelta come cost function la categorical cross-entropy (CCE), la quale quantifica la differenza tra due distribuzioni di probabilità tramite la seguente formulazione:

$$CCE = - \sum_{i=1}^{N_c} y_i \log(z_i) \quad (10)$$

Dove z_i è l’output della rete corrispondente alla classe reale, N_c è il numero totale delle classi e y_i è la probabilità di appartenenza alla classe i , diversa da zero solo per la classe reale. L’utilizzo di questa cost function è il motivo per il quale è stata scelta softmax come funzione di attivazione, poiché fornisce all’output le proprietà di cui abbiamo bisogno. In formule:

$$\sigma(z)_i = softmax = \frac{e^{z_i}}{\sum_{j=1}^{N_c} e^{z_j}}, \quad i = 1, \dots, N_c \quad (11)$$

$$CCE = - \sum_{i=1}^{N_c} \log(\sigma(z)_i) \quad (12)$$

Per poter applicare queste formule le classi devono essere codificate, associando un valore a ciascuna. In questo caso, poiché, per comodità, i valori associati sono numerici, la cost function è chiamata sparse categorical cross entropy.

Il metodo di ottimizzazione dei parametri della rete utilizzato durante l’allenamento è l’algoritmo Adam (Adaptive momentum estimation). Si tratta di un metodo di ottimizzazione del primo ordine che introduce dei miglioramenti rispetto all’algoritmo di discesa del gradiente stocastico (SGD) [30]. Il metodo memorizza la media mobile esponenziale (media) e la media mobile esponenziale dei quadrati (varianza) dei gradienti accumulati fino all’iterazione corrente. Questi valori vengono calcolati utilizzando 4 iperparametri:

- α , learning rate, per la fase di discesa del gradiente;

- $\beta_1 \in [0,1)$, decadimento del momento del primo ordine;
- $\beta_2 \in [0,1)$, decadimento del momento del secondo ordine;
- ϵ , parametro per la stabilità numerica.

Il metodo è di semplice implementazione, efficiente dal punto di vista computazionale ed è adatto per problemi che coinvolgono molti dati o parametri.

Nell'implementazione del nostro modello di deep neural network i quattro iperparametri sono stati utilizzati in due modi, a seconda delle prove successivamente svolte: con i valori di default presenti in Keras [32] ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-7$) o con i valori ricavati tramite l'algoritmo di Random Search.

Mentre i parametri del modello, ovvero i pesi e i bias interni alla deep neural network, vengono appresi in modo automatico durante l'allenamento, gli iperparametri devono essere impostati manualmente dall'utilizzatore. Per ricercare la giusta combinazione di iperparametri, ovvero quella che fornirà al modello le migliori prestazioni in termini di massima accuracy o minimo errore, viene svolta un'ottimizzazione (hyperparameter tuning). I due metodi di tuning più comuni sono:

- Grid Search: forniti gli iperparametri da ottimizzare, ognuno con i possibili valori che può assumere, viene creato e valutato un modello per ogni loro possibile combinazione;
- Random Search: ha lo stesso funzionamento della Grid Search, ma in questo caso vengono creati e valutati i modelli solo di un numero fisso di combinazioni, scelte in modo random tra tutte quelle possibili.

Tra i due approcci è stato scelto l'algoritmo Random Search, perchè è stato dimostrato essere più efficiente nel tuning degli iperparametri [33].

Il validation set (VS) viene utilizzato al termine di ogni epoca di allenamento del modello per valutare la loss e l'accuracy ottenute, è quindi utile per avere un'idea delle performance. Per alcune prove di allenamento è stato scelto come validation set il 20% dei segmenti appartenenti al training set originario (divisione percentuale), nel nostro caso una matrice con 7560 segmenti lunghi 1000 campioni (Tabella 3). I segmenti sono stati divisi tramite il "validation_split" implementato in Keras [32], il quale preleva gli ultimi elementi del dataset originario per creare il validation.

	(N° segmenti, N° campioni)
TRS	(30240,1000)
VS	(7560,1000)

Tabella 3. Creazione VS per divisione percentuale

Per altre prove di allenamento il validation è stato appositamente costruito con circa il 20% dei records appartenenti al training set originario (17 records), scelti in modo casuale. In questo modo i record dei due set sono unici. Il validation così creato è una matrice con 7650 segmenti lunghi 1000 campioni (Tabella 4).

	N° records	(N° segmenti, N° campioni)
TRS	67	(30150,1000)
VS	17	(7650,1000)

Tabella 4. Creazione VS per divisione dei records

Infine, in altre prove di allenamento è stata impiegata la tecnica del k-fold cross validation, utile quando i dati forniti in input non sono presenti in numero elevato [34]. Essa consiste nel suddividere il dataset in k parti e, in maniera iterativa, scegliere una parte (fold) come validation set e le altre k-1 come training set. Al termine di tutti gli allenamenti viene scelta la suddivisione che ha fornito i risultati migliori in termini di accuracy sul validation set. In particolare, è stato implementato lo stratified k-fold cross validation, il quale è applicato nello stesso modo, ma la suddivisione è operata in modo tale che in ogni fold sia inserito un numero di elementi appartenenti ad ogni classe proporzionale al numero totale di elementi della classe nel dataset. Questo metodo fornisce una migliore performance, in termini di bias e varianza, rispetto alla tecnica tradizionale. Il valore di k è stato scelto pari a 5, perché raccomandato come buon compromesso [35]. La tecnica è stata impiegata sia dividendo i segmenti ECG, sia provando a dividere i records.

Per prevenire l'overfitting del modello è stata anche impiegata la tecnica di EarlyStopping: il modello termina l'allenamento quando la loss calcolata sul validation set non diminuisce dopo un numero di epoche prefissato.

4.1.5.3 Allenamento del modello

Il modello così definito è costituito da un totale di 1,416,044 parametri allenabili, ovvero i pesi e bias che verranno ottimizzati. Il modello è stato implementato in Python 3.8.5 tramite il framework Keras [32], una libreria open-source progettata per il lavoro con le deep neural networks. L'allenamento è stato svolto su Google Colab [36], il quale permette l'utilizzo di GPU e quindi la riduzione dei tempi di allenamento. Durante l'allenamento è stata monitorata l'accuracy.

Per identificare la combinazione ottimale di iperparametri, ovvero quella con accuracy migliore, sono state svolte diverse prove di allenamento del modello. In particolare, sono state variate le seguenti condizioni:

- Numero di epoche, dove un'epoca si verifica presentando tutti gli elementi del dataset alla rete neurale;
- Annotazioni, in quanto sono state utilizzate le annotazioni automatiche (file “.pu0” e “.pu1”) con dice similarity maggiore rispetto alle annotazioni manuali (file “.q1c” e “.q2c”) oppure le annotazioni automatiche del file “.pu”;
- Standardizzazione del dataset, in quanto sono state svolte prove anche con il dataset non-standardizzato;
- Validation set, in quanto in ogni prova è stato scelto no validation o validation ottenuto per divisione percentuale, divisione dei records o stratified 5-fold cross validation (applicato ai segmenti o ai records);
- EarlyStopping, in quanto in ogni prova la tecnica è stata o no utilizzata;
- Iperparametri del metodo di ottimizzazione Adam, in quanto alcune prove li ricavano tramite la tecnica Random Search, altre utilizzano i loro valori di default.

Di tutte le prove svolte vengono ora elencate le più significative. Per ogni prova sono stati ricavati i grafici della loss e dell'accuracy per il training e il validation set. Inoltre, il modello allenato è stato validato con il test set ed è stata calcolata la confusion matrix relativa.

Nella prima prova è stato utilizzato un numero di epoche pari a 20, sono state utilizzate le annotazioni automatiche comparate con le annotazioni manuali, il dataset non è stato standardizzato, non è stato utilizzato il validation set, non è stato utilizzato l'EarlyStopping e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default.

In questo caso, quindi, l'allenamento è stato svolto con un training set contenente tutti i records.

Nella seconda prova è stato utilizzato un numero di epoche pari a 30, sono state utilizzate le annotazioni automatiche comparate con le annotazioni manuali, il dataset non è stato standardizzato, è stato utilizzato il validation set ottenuto tramite divisione percentuale, non è stato utilizzato l'EarlyStopping e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default.

Nella terza prova è stato utilizzato un numero di epoche pari a 30, sono state utilizzate le annotazioni automatiche comparate con le annotazioni manuali, il dataset non è stato standardizzato, è stato utilizzato il validation set ottenuto tramite divisione dei records, non è stato utilizzato l'EarlyStopping e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default.

Nella quarta prova è stato utilizzato un numero di epoche pari a 50, sono state utilizzate le annotazioni automatiche comparate con le annotazioni manuali, il dataset non è stato standardizzato, è stato utilizzato lo stratified 5-fold cross validation applicato ai segmenti ECG, è stata applicata la tecnica dell'EarlyStopping ponendo il numero di epoche di controllo pari a 12 e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default. In questo caso, quindi, sono stati allenati 5 modelli, tanti quanti i fold creati tramite il cross-validation. Al termine è stato scelto il modello con accuracy migliore sul validation set.

Nella quinta prova è stato utilizzato un numero di epoche pari a 15, sono state utilizzate le annotazioni automatiche comparate con le annotazioni manuali, il dataset non è stato standardizzato, è stato utilizzato il validation set ottenuto tramite divisione dei records, non è stata applicata la tecnica dell'EarlyStopping, l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri ricavati tramite Random Search. Relativamente a questo ultimo punto, sono stati fatti variare gli iperparametri α (range $[4e-4, 2e-3]$) e β_1 (da un valore minimo di 0.1 ad un massimo di 0.9, con step 0.1) svolgendo 10 trials, ricavando quindi 10 combinazioni in modo casuale tra tutte le combinazioni possibili.

Nella sesta prova è stato utilizzato un numero di epoche pari a 15, sono state utilizzate le annotazioni automatiche comparate con le annotazioni manuali, il dataset è stato standardizzato, è stato utilizzato il validation set ottenuto tramite divisione percentuale, non è stato utilizzato

l'EarlyStopping e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default.

Nella settima prova è stato utilizzato un numero di epoche pari a 15, sono state utilizzate le annotazioni automatiche comparate con le annotazioni manuali, il dataset è stato standardizzato, è stato utilizzato il validation set ottenuto tramite divisione dei records, non è stato utilizzato l'EarlyStopping e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default.

Nell'ottava prova è stato utilizzato un numero di epoche pari a 15, sono state utilizzate le annotazioni automatiche del file “.pu”, il dataset non è stato standardizzato, è stato utilizzato il validation set ottenuto tramite divisione percentuale, non è stato utilizzato l'EarlyStopping e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default.

Nella nona prova è stato utilizzato un numero di epoche pari a 15, sono state utilizzate le annotazioni automatiche del file “.pu”, il dataset è stato standardizzato, è stato utilizzato il validation set ottenuto tramite divisione percentuale, non è stato utilizzato l'EarlyStopping e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default.

Nella decima prova è stato utilizzato un numero di epoche pari a 15, sono state utilizzate le annotazioni automatiche del file “.pu”, il dataset è stato standardizzato, è stato utilizzato il validation set ottenuto tramite divisione dei records, non è stato utilizzato l'EarlyStopping e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default.

Nell'undicesima prova è stato utilizzato un numero di epoche pari a 30, sono state utilizzate le annotazioni automatiche del file “.pu”, il dataset è stato standardizzato, è stato utilizzato lo stratified 5-fold cross validation applicato ai segmenti ECG, è stata applicata la tecnica dell'EarlyStopping ponendo il numero di epoche di controllo pari a 6 e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default. In questo caso, quindi, sono stati allenati 5 modelli, tanti quanti i fold creati tramite il cross-validation. Al termine è stato scelto il modello con accuracy migliore sul validation set.

Nella dodicesima prova è stato utilizzato un numero di epoche pari a 15, sono state utilizzate le annotazioni automatiche comparate con le annotazioni manuali, il dataset non è stato

standardizzato, è stato utilizzato lo stratified 5-fold cross validation applicato ai records, non è stata applicata la tecnica dell'EarlyStopping e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default. In questo caso, quindi, sono stati allenati 5 modelli, tanti quanti i fold creati tramite il cross-validation. Ogni fold, però, è stato ottenuto dalla divisione dei records. Al termine è stato scelto il modello con accuracy migliore sul validation set.

Nella tredicesima prova è stato utilizzato un numero di epoche pari a 15, sono state utilizzate le annotazioni automatiche comparate con le annotazioni manuali, il dataset è stato standardizzato, è stato utilizzato lo stratified 5-fold cross validation applicato ai records, non è stata applicata la tecnica dell'EarlyStopping e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default. In questo caso, quindi, sono stati allenati 5 modelli, tanti quanti i fold creati tramite il cross-validation. Ogni fold, però, è stato ottenuto dalla divisione dei records. Al termine è stato scelto il modello con accuracy migliore sul validation set.

Nella quattordicesima prova è stato utilizzato un numero di epoche pari a 15, sono state utilizzate le annotazioni automatiche del file “.pu”, il dataset è stato standardizzato, è stato utilizzato lo stratified 5-fold cross validation applicato ai records, non è stata applicata la tecnica dell'EarlyStopping e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default. In questo caso, quindi, sono stati allenati 5 modelli, tanti quanti i fold creati tramite il cross-validation. Ogni fold, però, è stato ottenuto dalla divisione dei records. Al termine è stato scelto il modello con accuracy migliore sul validation set.

Nella quindicesima prova è stato utilizzato un numero di epoche pari a 15, sono state utilizzate le annotazioni automatiche del file “.pu”, il dataset non è stato standardizzato, è stato utilizzato lo stratified 5-fold cross validation applicato ai records, non è stata applicata la tecnica dell'EarlyStopping e l'algoritmo di ottimizzazione Adam è stato utilizzato con gli iperparametri di default. In questo caso, quindi, sono stati allenati 5 modelli, tanti quanti i fold creati tramite il cross-validation. Ogni fold, però, è stato ottenuto dalla divisione dei records. Al termine è stato scelto il modello con accuracy migliore sul validation set.

4.1.6 Modello scelto: test con i segnali di Brugada

Tra tutti i modelli testati durante le prove di allenamento è stato scelto quello con accuracy

migliore e, in generale, migliori performance nel riconoscimento dei tratti utili alla successiva valutazione della Sindrome di Brugada, quindi in particolare l'onda P. Il modello è stato utilizzato fornendo in input i segnali patologici messi a disposizione dall'ospedale, pre-processati come definito in 4.1.3 e in base alle caratteristiche della prova con cui il modello è stato allenato. I risultati ottenuti non sono stati considerati soddisfacenti, poiché il riconoscimento della rete neurale ha dato in output un'accuracy molto bassa

Si è così deciso di adattare la deep neural network e di allenarla direttamente con i segnali di Brugada. Inizia ora la seconda fase.

4.2 SECONDA FASE – SEGNALI DI BRUGADA

4.2.1 Dataset

I segnali ECG utilizzati per l'implementazione dell'algoritmo e la successiva validazione sono stati forniti dal S.S. Elettrofisiologia e Pace-maker, S.C. Cardiologia, Dipartimento Strutturale Medico dell'A.O. Ordine Mauriziano di Torino. I segnali sono stati acquisiti da pazienti affetti dalla Sindrome di Brugada in tre diverse condizioni:

- Baseline (BS);
- Durante la somministrazione del farmaco Ajmalina (AJM);
- Post-ablazione (FINE).

La condizione baseline è la situazione originaria del paziente, quando questo si trova all'inizio del percorso clinico e non è ancora stato sottoposto a nessun trattamento.

La seconda condizione è quella che si verifica quando al paziente viene somministrata Ajmalina, alcaloide che possiede proprietà bloccanti i canali del sodio cardiaci appartenente alla classe 1A dei farmaci antiaritmici. L'inibizione della corrente al sodio sbilancia l'equilibrio dei flussi ionici transmembrana portando ad una modifica dei potenziali di azione cellulari, e quindi al sopraslivellamento del tratto ST nel tracciato ECG dei soggetti patologici [37].

La condizione post-ablazione, infine, è la situazione immediatamente successiva all'intervento chirurgico di ablazione delle cellule associate alla comparsa dei potenziali elettrici anomali. Il substrato aritmico, localizzato sulla superficie epicardica del ventricolo destro e del suo tratto

d'efflusso, viene eliminato con rapide erogazioni a radiofrequenza. Questo trattamento, dopo qualche tempo, dovrebbe portare alla normalizzazione dell'elettrocardiogramma [38].

I pazienti totali sono 24, rappresentati in Tabella 5 con un identificativo numerico per motivi di privacy. Per ogni individuo è stato fornito l'ECG nelle tre condizioni con alcune eccezioni, principalmente dovute al dissenso di somministrazione del farmaco o dell'intervento chirurgico da parte del paziente.

ID paziente	BS	AJM	FINE
5	X	-	X
6	X	X	-
7	X	X	X
13	X	X	X
15	X	X	X
16	X	X	X
17	X	X	X
18	X	X	X
21	X	X	X
23	X	X	-
24	X	X	X
25	X	-	-
29	X	X	-
32	X	X	X
39	X	-	X
41	X	-	X
48	X	X	X
49	X	X	X
50	X	X	X
51	X	X	X
52	X	X	X
55	X	X	X
56	X	X	X
57	X	X	X
N° pazienti	24	20	20

Tabella 5. Descrizione dei pazienti nelle tre condizioni

Ogni segnale ECG è stato acquisito a 12 derivazioni, utilizzando però le derivazioni precordiali alte, cioè posizionando gli elettrodi V5 e V6 nel secondo spazio intercostale (V1_{IC2} e V2_{IC2}). Tuttavia, sono state utilizzate solo le derivazioni V1, V2, V1_{IC2} e V2_{IC2}, registranti l'attività elettrica della zona del ventricolo destro maggiormente interessata dalla manifestazione della Sindrome di Brugada. Quindi, con 4 tracciati elettrocardiografici per ogni paziente, risultano:

- 96 segnali ECG per BS;

- 80 segnali ECG per AJM;
- 80 segnali ECG per FINE.

Questi sono i tre dataset che verranno utilizzati separatamente per l'implementazione della deep neural network.

Ogni tracciato ECG è stato acquisito con una frequenza di campionamento pari a 1000 Hz e ha una durata pari a 62.5 s, risultando quindi formato da 62500 campioni. I segnali, inizialmente forniti in formato “.txt”, divisi in colonne e riportanti identificativo del paziente, data e condizione di acquisizione, sono poi stati convertiti in formato “.mat” tramite il software Matlab®, per facilitare la loro leggibilità.

4.2.2 Target

Per guidare l'allenamento di una rete neurale artificiale è necessario fornire la “ground truth”, ovvero i valori reali di cui si vuole ottenere il riconoscimento. La rete, infatti, modifica i pesi delle connessioni dei neuroni sulla base dell'errore compiuto tra il target e il valore di output predetto, cercando così di migliorare il suo apprendimento. Nel caso specifico in esame sono stati scelti come target le onde e i segmenti utili alla successiva valutazione della Sindrome di Brugada: isoelettrica, onda P, complesso QRS, segmento ST (tratto che verrà utilizzato per il calcolo dell'area) e segmento TP. Ognuno di essi è stato codificato con un valore numerico, come indicato in Tabella 6.

Onda/segmento	Codifica numerica
Isoelettrica (No wave, NW)	0
Onda P	1
Complesso QRS	2
Segmento ST (Tratto area)	3
Segmento TP	4

Tabella 6. Codifica numerica dei target

I target sono puntuali: ogni campione del segnale ha associato il valore numerico che indica la sua appartenenza al tratto ECG corrispondente. Ogni segnale dei tre dataset creati è stato visualizzato su plot grafico e i valori numerici sono stati inseriti manualmente da un operatore.

4.2.3 Pre-processing

I segnali di ciascuno dei tre dataset sono stati filtrati con un filtro passa-banda di Butterworth di ordine 3, con frequenze di taglio pari a 0.7 e 20 Hz, per attenuare la fluttuazione della linea base ed eventuali artefatti. Il filtraggio è stato anticausale, per compensare la fase non lineare del filtro ed evitare la distorsione dei segnali ECG.

Successivamente, ogni segnale è stato osservato attentamente e sono state eliminate le parti considerate troppo rumorose o non utili all'allenamento della rete neurale. In particolare, sono stati eliminati il paziente con numero identificativo 25 e, per la condizione AJM, le derivazioni V1 e V2 del paziente 7, V1 del paziente 15 e V2 del paziente 24.

Ogni segnale è stato standardizzato, rimuovendo la media dei campioni del segnale e dividendo per la deviazione standard, processo raccomandato nell'allenamento delle reti neurali profonde per migliorarne le prestazioni.

Infine, ogni segnale è stato suddiviso in segmenti da 2500 campioni l'uno, corrispondenti a circa tre battiti cardiaci, per allenare il modello sulla continuità della serie temporale e ottenere un risultato più robusto. I segmenti così creati costituiscono l'input della rete neurale.

4.2.4 Divisione del dataset

Per ogni dataset creato i segnali elettrocardiografici sono stati divisi in un training set (TRS), utilizzato per l'allenamento della rete e in un test set (TS), utilizzato per la validazione delle performance. I test set sono stati costituiti con circa il 20% del dataset corrispondente. La divisione è stata fatta in modo tale che i pazienti fossero unici nei due set, per validare la rete su dati mai elaborati ed evitare un bias. Dividendo i pazienti è stata ottenuta anche la divisione dei segmenti corrispondenti. Perciò, per ogni set, l'input che viene fornito alla rete è una matrice formata da segmenti lunghi 2500 campioni. Nel dettaglio in Tabella 7.

		N° pazienti	(N° segmenti, N° campioni)
Dataset 1 BS	TRS1	19	(1512,2500)
	TS1	5	(412,2500)
Dataset 2 AJM	TRS2	16	(1500,2500)
	TS2	4	(400,2500)
Dataset 3 FINE	TRS3	16	(1452,2500)
	TS3	4	(396,2500)

Tabella 7. Descrizione della divisione dei 3 dataset

4.2.5 Deep neural network

4.2.5.1 Architettura

Per poter riallenare la rete neurale con i segnali ECG dei pazienti affetti dalla Sindrome di Brugada e con i target creati sono state applicate delle piccole modifiche al modello:

- il layer di input è stato modificato per ricevere in ingresso sequenze da 2500 campioni;
- il Dense layer è stato implementato con 5 unità nascoste, pari al numero di onde e segmenti che si vogliono identificare.

Tutti gli altri layers e parametri definiti nella sezione 4.1.5.1 sono rimasti invariati.

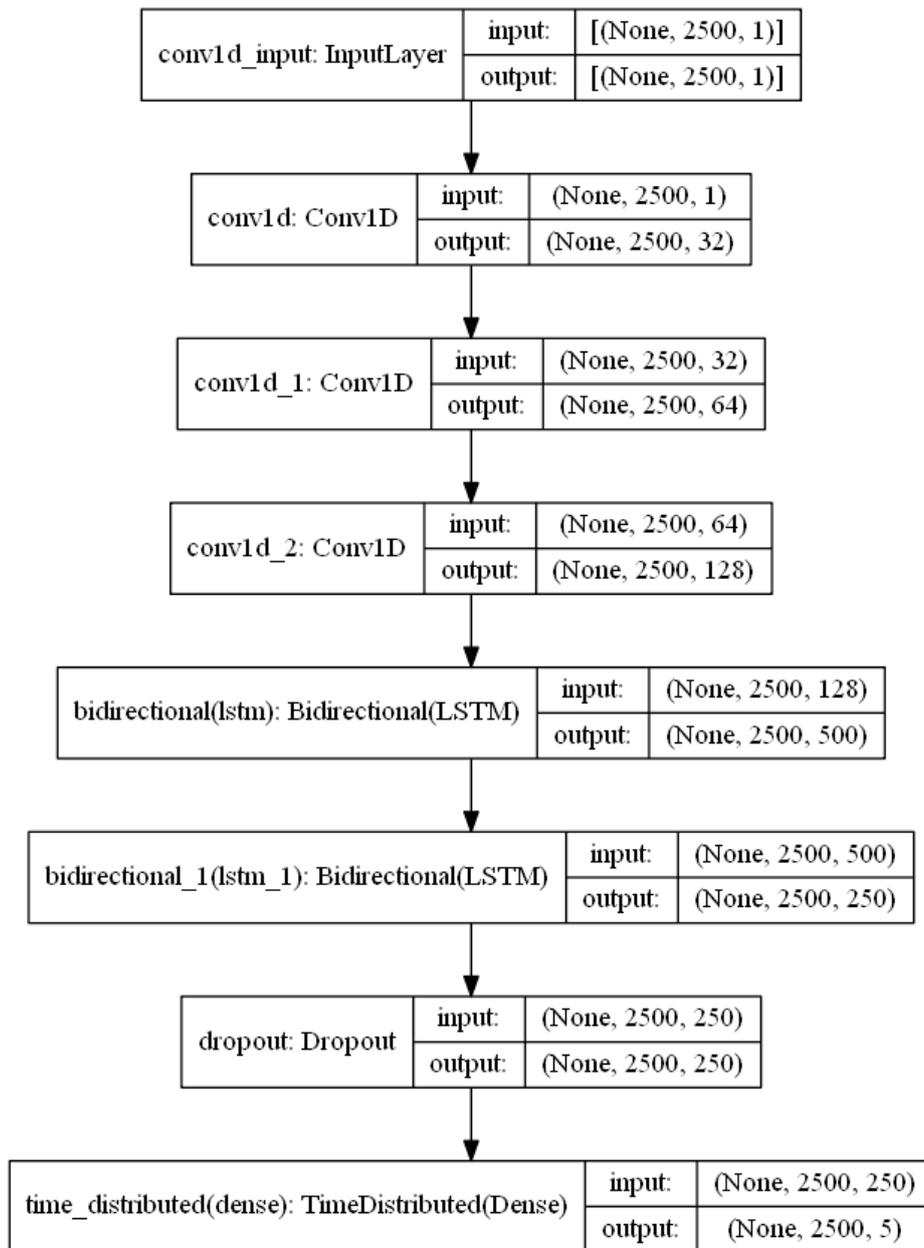


Figura 5. Architettura deep neural network modificata

4.2.5.2 Parametri di allenamento del modello

Anche in questo caso per il processo di allenamento è stata scelta come cost function la sparse categorical cross entropy, poiché ogni campione può essere classificato in una tra 5 classi possibili.

Come metodo di ottimizzazione è stato utilizzato l’algoritmo Adam, questa volta solo con i valori di default dei quattro iperparametri ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-7$).

Il validation set non è stato utilizzato, per non diminuire il numero di elementi utilizzati per l'allenamento, poiché i segnali a disposizione sono in numero ridotto.

La tecnica di EarlyStopping non è stata applicata.

4.2.5.3 Allenamento del modello

Il modello è costituito da un totale di 1,416,044 parametri allenabili, è stato implementato in Python 3.8.5 tramite il framework Keras [32] e il suo allenamento è stato svolto su Google Colab [36].

In questo caso non sono state svolte diverse prove, ma i TRS delle tre condizioni (BS, AJM, FINE) sono stati utilizzati in modo indipendente per allenare il modello. Al termine, quindi, sono stati ottenuti tre modelli, ognuno allenato per una delle tre condizioni.

Per ogni allenamento il numero di epoche utilizzate è stato pari a 50 e il validation set non è stato utilizzato. L'allenamento è quindi stato svolto con un training set contenente tutti gli elementi originari. Per ogni dataset sono stati ricavati i grafici della loss e dell'accuracy per il training set. Inoltre, ogni modello allenato è stato validato con il corrispondente test set ed è stata calcolata la confusion matrix relativa.

4.2.6 Algoritmo di post-processing: calcolo dell'isoelettrica e delle aree sottese ai tratti ST

I risultati di riconoscimento, ottenuti dall'applicazione del modello allenato di deep neural network, sono stati utilizzati per calcolare i valori dell'isoelettrica e delle aree sottese ai tratti ST tramite un algoritmo di post-processing. L'algoritmo è strutturato in tre fasi e viene applicato separatamente ad ognuno dei tre dataset (BS, AJM e FINE). L'applicazione dell'algoritmo è relativa, per ogni dataset, ad ogni singolo segmento ECG di 2500 campioni appartenente al test set; questo significa che per ogni segmento verranno calcolati un valore di isoelettrica e i valori di area sottesa ai tratti ST presenti.

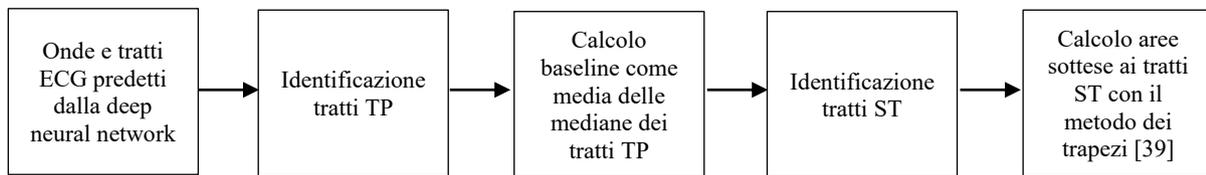


Figura 6. Flowchart algoritmo di post-processing

In seguito all'ispezione visiva, svolta mostrando a schermo il segmento ECG di 2500 campioni con le posizioni delle onde e dei tratti predette dalla rete neurale, ha inizio la prima fase, la quale ha come obiettivo il calcolo del valore dell'isoelettrica. In primo luogo vengono identificati tutti i campioni del segmento ECG che la rete ha predetto appartenenti ai tratti TP e, per ogni sequenza di campioni consecutivi, vengono memorizzati gli indici dei campioni di inizio e fine. Successivamente viene svolto un controllo sugli indici trovati, utile per eliminare piccole inesattezze di predizione da parte della deep neural network. Vengono eliminati gli indici di inizio e fine coincidenti (riconoscimento di un singolo punto isolato) e vengono unite le sequenze in cui l'indice di fine di una dista meno di 150 campioni dall'indice di inizio dell'altra (riconoscimento del tratto a pezzi). Inoltre, sono eliminate le sequenze il cui primo campione coincide con l'inizio del segmento ECG o il cui ultimo campione coincide con la fine del segmento ECG; questa operazione risulta necessaria per non utilizzare tratti TP incompleti, di cui non si è certi relativamente i veri inizio o fine. Le sequenze così rimaste costituiscono i tratti TP utilizzabili per il calcolo dell'isoelettrica. Per ogni tratto TP è calcolata la mediana dei campioni e poi, mediando il valore delle mediane di ogni tratto, è ricavato il valore di isoelettrica per il segmento ECG considerato. Al termine di questa fase viene mostrato a schermo il segmento ECG con i tratti TP trovati e l'isoelettrica calcolata.

Durante la seconda fase vengono identificati tutti i campioni del segmento ECG che la rete ha predetto appartenenti ai tratti ST, memorizzando gli indici dei campioni di inizio e fine di ogni sequenza di campioni consecutivi. Anche in questo caso viene svolto un controllo sugli indici trovati, eliminando gli indici di inizio e fine sequenza coincidenti, unendo le sequenze distanti meno di 150 campioni ed eliminando le sequenze in cui il primo o l'ultimo campione corrispondono, rispettivamente, all'inizio o alla fine del segmento ECG. Le sequenze così rimaste costituiscono i tratti ST utilizzabili.

Infine, ha inizio la terza fase, durante la quale vengono calcolati i valori delle aree sottese ai tratti ST. La funzione per il calcolo delle aree sottese utilizza i valori di baseline e i tratti ST trovati, svolgendo una comparazione tra i due. Gli indici di inizio e fine di ogni tratto ST vengono incrementati o decrementati fino a raggiungere il valore di baseline. Una volta trovati i nuovi valori degli indici, quindi i nuovi tratti ST, si procede con il calcolo delle aree sottese.

Queste vengono ricavate applicando il metodo dei trapezi [39] tra i nuovi tratti ST e la baseline. In analisi numerica il metodo dei trapezi è uno dei metodi utilizzati per approssimare il valore di un integrale definito $\int_a^b f(x) dx$. Nella sua formulazione più semplice il valore dell'integrale viene approssimato dall'area del trapezio avente vertici $(a, 0)$, $(a, f(a))$, $(b, f(b))$ e $(b, 0)$, la quale è pari a

$$\int_a^b f(x) dx \approx \frac{b-a}{2} [f(a) + f(b)] \quad (13)$$

Tuttavia, se la funzione integranda non è lineare, questa formula non è molto precisa. In questo caso l'intervallo di integrazione $[a, b]$ viene suddiviso in n sotto intervalli

$[x_i, x_{i+1}]$, con $i = 0, \dots, n-1$, dove $x_0 = a$ e $x_n = b$, ognuno con ampiezza pari a $h = \frac{b-a}{n}$.

Il valore approssimato dell'integrale è dato dalla somma delle aree dei trapezi costruiti in ogni intervallo, ognuno con base h e altezze pari ai valori di $f(x)$ agli estremi dell'intervallo. La formulazione è la seguente:

$$\int_a^b f(x) dx \approx h \left[\frac{f(a)}{2} + f(x_1) + \dots + f(x_{n-1}) + \frac{f(b)}{2} \right] \quad (14)$$

In uscita l'algoritmo di post-processing fornisce i nuovi indici dei tratti ST e i valori delle aree sottese calcolati.

4.2.7 Analisi delle aree sottese ai tratti ST

I valori delle aree ottenuti con l'algoritmo di post-processing sono stati memorizzati, per poter essere successivamente utilizzati per le analisi quantitative e i confronti tra i tre dataset. Per poter svolgere l'analisi nel modo corretto sono state calcolate e salvate, per tutte e tre le condizioni, anche le aree appartenenti ai corrispettivi training set. Quindi, per ogni dataset, i segmenti del training sono stati forniti alla rete e i risultati di riconoscimento sono stati utilizzati dall'algoritmo di post-processing; le aree così ricavate sono state unite a quelle del test set corrispondente.

Successivamente, per ogni dataset, tutte le aree calcolate sono state mediate ed è stata calcolata la deviazione standard. L'area totale di ogni dataset è stata espressa come valor medio \pm deviazione standard.

I valori delle aree sono stati ricavati anche divisi per derivazione (V1, V2, V1_{IC2}, V2_{IC2}). In questo caso le aree dei segmenti sono state raggruppate in base alla loro derivazione e poi sono state mediate in ogni gruppo. L'area totale di ogni derivazione, per ogni dataset, è stata espressa come valor medio \pm deviazione standard.

4.2.8 Algoritmo di validazione manuale

I risultati automatici sul test set, ottenuti tramite il modello di deep neural network e il successivo algoritmo di post-processing, sono stati validati da due operatori esperti. Per poter raggiungere l'obiettivo è stato sviluppato un algoritmo apposito, il quale permette la selezione dei tratti TP e ST nei segmenti ECG appartenenti al test set. Il procedimento è costituito da due fasi principali, ognuna delle quali viene applicata su tutti gli elementi del test set. Una volta avviata la routine, sviluppata su Python, il segmento ECG viene mostrato a video in una nuova finestra. Tramite il cursore del mouse è possibile interagire con il grafico in maniera interattiva: sul segmento ECG viene visualizzato un marker rosso, evidenziato da due rette (verticale e orizzontale) passanti per esso, il quale si muove lungo il segnale seguendo i movimenti del cursore.

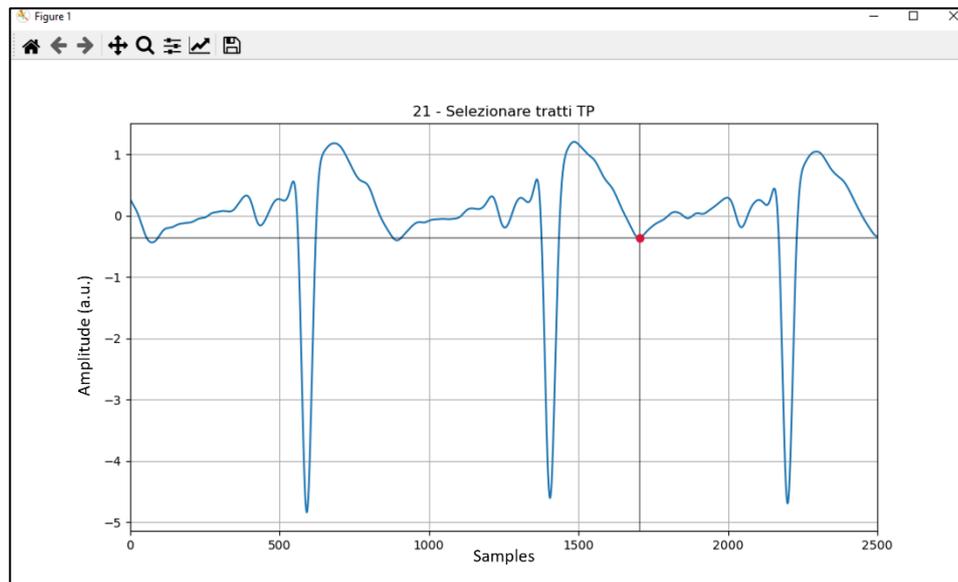


Figura 7. Esempio della finestra con il marker rosso per la selezione dei punti sul segmento ECG

In questa prima fase l'operatore deve selezionare il tratto o i tratti TP, compito indicato nel titolo del grafico insieme al numero identificativo del segmento visualizzato. Cliccando con il tasto destro del mouse, sulla posizione identificata dal marker viene applicato un label (punto rosso). L'operatore, in questo modo, deve applicare due label per ogni tratto TP che vuole

identificare: uno indicante l'inizio del tratto e l'altro la fine; risulta quindi sempre necessario un numero totale di punti pari.

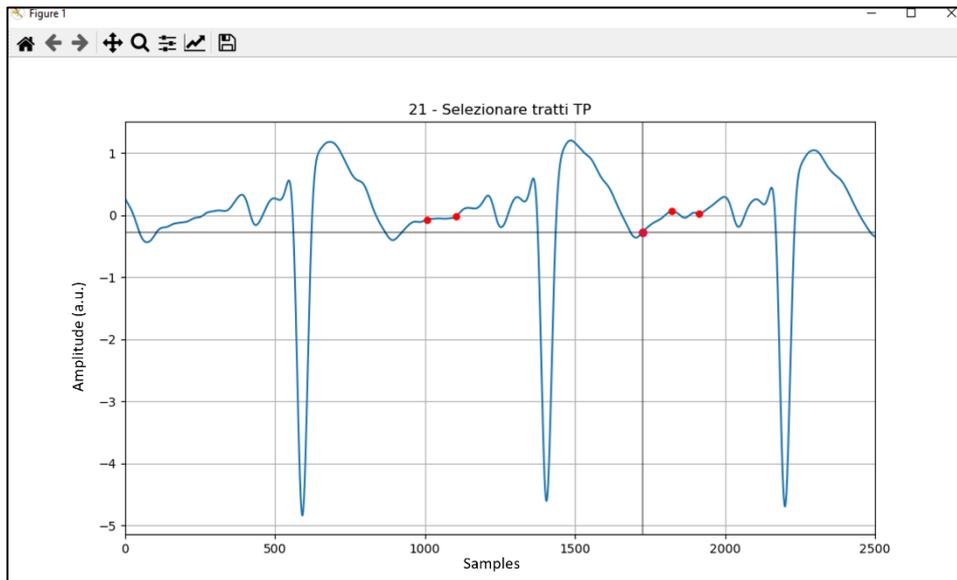


Figura 8. Esempio di selezione punti di inizio e fine tratti TP

Una volta terminata la selezione, cliccando la barra spaziatrice, è possibile visualizzare sul segmento ECG i tratti TP determinati dalla scelta dei punti e la baseline calcolata. Il calcolo è svolto applicando il metodo già utilizzato nell'algorithm di post-processing automatico, ovvero calcolando la mediana di ogni tratto TP e poi svolgendo la media delle mediane ottenute.

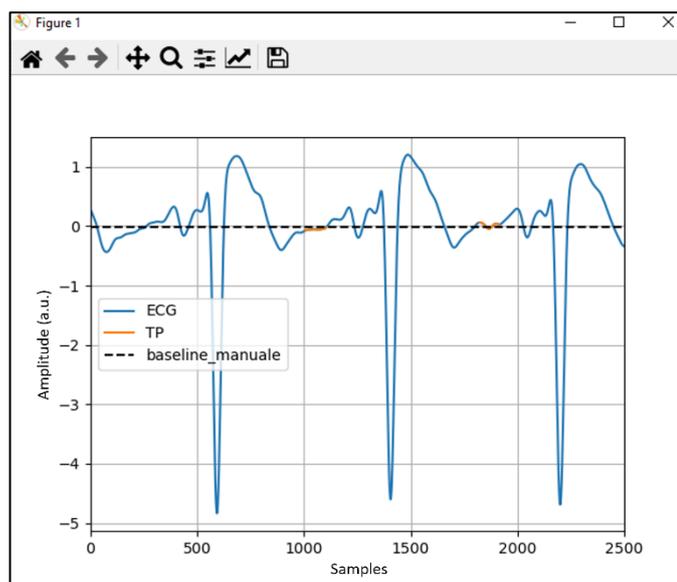


Figura 9. Esempio di tratti TP e baseline determinati con la selezione manuale

Cliccando nuovamente sulla barra spaziatrice si apre una nuova finestra, dando inizio alla seconda fase del processo. In questo caso l'operatore, sempre tramite marker in movimento sul grafico e tasto destro del mouse, deve selezionare inizio e fine del tratto o dei tratti ST che verranno successivamente utilizzati per il calcolo dell'area sottesa. Al termine della selezione anche in questo caso, cliccando la barra spaziatrice, vengono mostrati i tratti ST determinati insieme alla baseline manuale ricavata nella fase precedente.

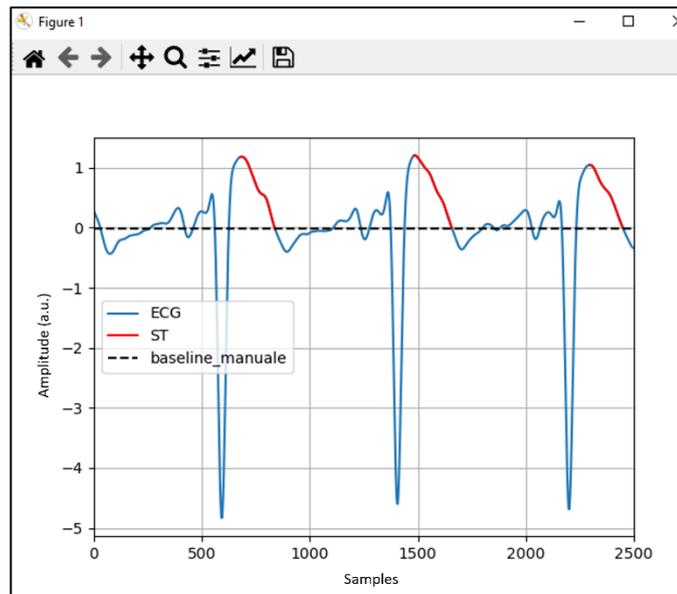


Figura 10. Esempio di tratti ST determinati con la selezione manuale

Per la successiva validazione sono stati salvati i valori di isoelettrica e inizio e fine dei tratti ST per ogni segmento ECG del test set. La routine così implementata è stata applicata, separatamente per ognuno, sui test set dei tre dataset a disposizione (BS, AJM e FINE). In tutto sono stati processati 1208 segmenti. Anche i processi di validazione spiegati in seguito sono stati svolti separatamente per i tre dataset.

4.2.8.1 Validazione baseline

Per la validazione della baseline è stato calcolato l'errore commesso dall'algoritmo automatico rispetto al valore ricavato manualmente. L'errore è stato espresso come errore assoluto, ricavato per ogni segmento ECG del test set nel seguente modo:

$$\text{Errore assoluto} = |\text{valore baseline manuale} - \text{valore baseline automatica}| \quad (15)$$

Tutti gli errori, di tutti i segmenti ECG, sono stati mediati ed è stata calcolata la deviazione standard. L'errore assoluto totale è stato espresso come valor medio \pm deviazione standard.

I valori di errore assoluto sono stati ricavati anche divisi per derivazione (V1, V2, V1_{IC2}, V2_{IC2}). In questo caso i segmenti sono stati raggruppati in base alla loro derivazione e poi gli errori trovati sono stati mediati in ogni gruppo. L'errore assoluto totale di ogni derivazione è stato espresso, anche in questo caso, come valor medio \pm deviazione standard.

4.2.8.2 Validazione tratti ST

Per la validazione dei tratti ST sono state utilizzate le metriche accuracy, precision e recall. Per ogni segmento del test set sono state create due maschere, una relativa al riconoscimento da parte dell'algoritmo automatico e l'altra relativa ai segmenti ST selezionati manualmente. Ogni maschera è costituita da 2500 campioni, tanti quanti sono contenuti in un segmento ECG; ogni campione assume valore 1 se è stato considerato appartenente al tratto ST e 0 altrimenti. Per ogni segmento ECG sono state comparate le due maschere calcolando:

- Accuracy: rappresenta la percentuale di elementi classificati correttamente rispetto al numero totale di elementi. Fornisce un indice delle prestazioni generali;

$$Accuracy = \frac{VP + VN}{VP + FP + FN + VN} \quad (16)$$

- Precision: rappresenta la percentuale di elementi positivi classificati correttamente rispetto al totale di elementi classificati come positivi. Indica la capacità di non includere campioni estranei nel tratto riconosciuto come ST;

$$Precision = \frac{VP}{VP + FP} \quad (17)$$

- Recall: rappresenta la percentuale di elementi positivi classificati correttamente rispetto al totale degli elementi positivi presenti. Indica la capacità di riconoscere correttamente il tratto ST.

$$Recall = \frac{VP}{VP + FN} \quad (18)$$

In queste formulazioni (equazioni 16 - 18) VP e VN sono i campioni con valore pari, rispettivamente, a 1 e 0 sia nella maschera manuale che in quella automatica, FP sono i campioni pari a 0 nella maschera manuale e 1 in quella automatica e FN sono i campioni pari a 1 nella maschera manuale e 0 in quella automatica.

Le metriche sono state calcolate per ogni segmento e poi, successivamente, i valori sono stati mediati ed è stata calcolata la deviazione standard. Ogni metrica totale è stata espressa come valor medio \pm deviazione standard.

I valori delle tre metriche sono stati ricavati anche divisi per derivazione (V1, V2, V1_{IC2}, V2_{IC2}). In questo caso i segmenti sono stati raggruppati in base alla loro derivazione e poi i valori delle metriche trovati sono stati mediati in ogni gruppo. I valori delle tre metriche totali di ogni derivazione sono stati espressi come valor medio \pm deviazione standard.

CAPITOLO 5

RISULTATI

5.1 RISULTATI PRIMA FASE

5.1.1 Risultati prove di allenamento

Per ogni prova vengono ora riportati i grafici della loss e dell'accuracy per il training e il validation set, quando questo è presente, durante le successive epoche di allenamento (Figure 11 - 25). Viene anche riportata la confusion matrix ottenuta mostrando alla rete i segmenti ECG appartenenti al test set, costruita ponendo le classi reali in riga e le classi predette dal modello di deep neural network in colonna (Tabelle 8 - 22). I valori classificati correttamente per le varie classi sono posti sulla diagonale.

Prova 1: sul training set, come valori medi, sono state ottenute loss pari a 0.123 e accuracy pari al 95.0%.

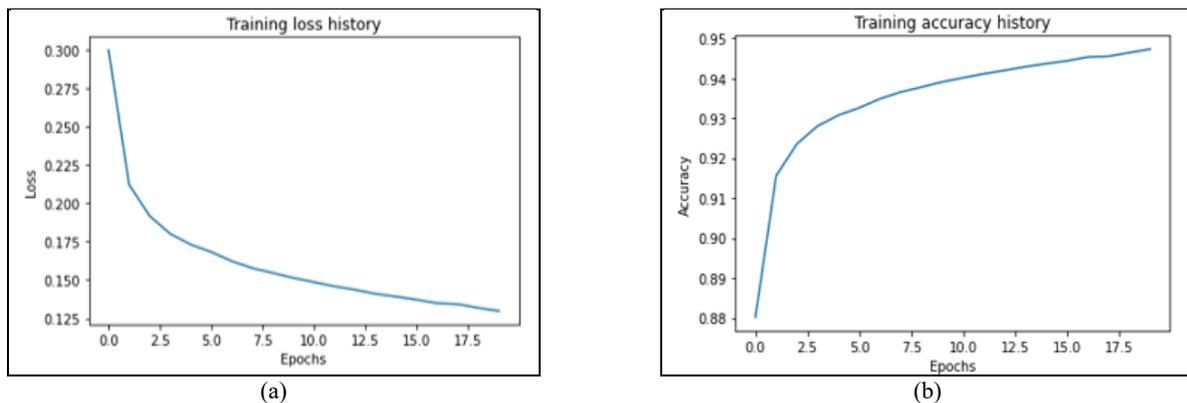


Figura 11. Loss training set (a) e accuracy training set (b) della prova 1

		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	90.4	2.2	2.1	5.2
	P	18.1	80.8	0.1	1.0
	QRS	13.6	0.1	86.2	0.2
	T	11.4	0.1	0.0	88.5

Tabella 8. Confusion matrix test set della prova 1

Prova 2: sul training set, come valori medi, sono state ottenute loss pari a 0.183 e accuracy pari al 93.3%.

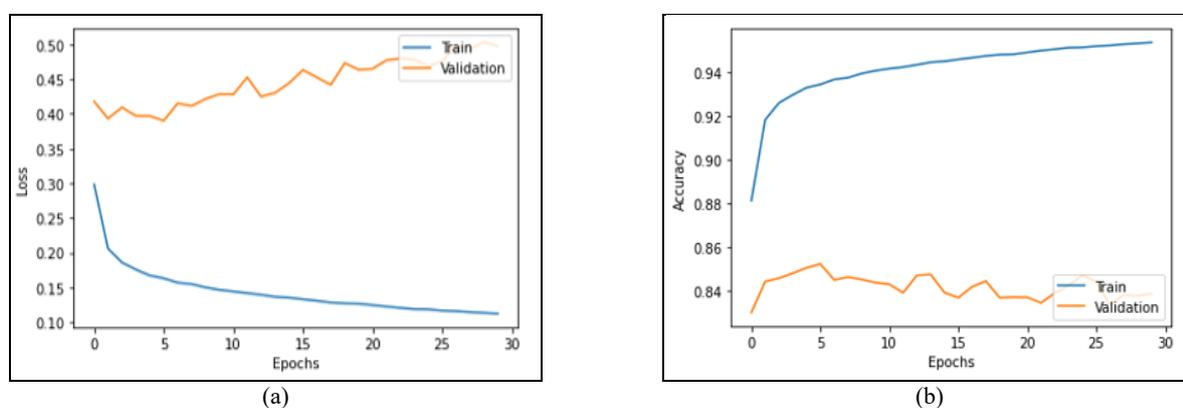


Figura 12. Loss training e validation set (a) e accuracy training e validation set (b) della prova 2

		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	89.8	2.5	2.2	5.5
	P	17.1	82.0	0.1	0.9
	QRS	13.2	0.1	86.5	0.2
	T	10.8	0.2	0.1	88.9

Tabella 9. Confusion matrix test set della prova 2

Prova 3: sul training set, come valori medi, sono state ottenute loss pari a 0.116 e accuracy pari al 95.2%; sul validation set, come valori medi, sono state ottenute loss pari a 0.359 e accuracy

pari al 89.4%.

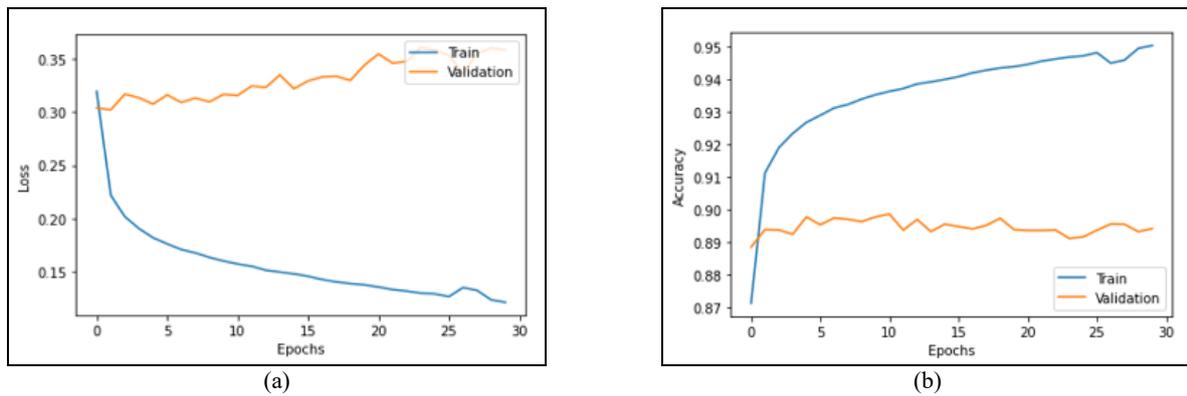


Figura 13. Loss training e validation set (a) e accuracy training e validation set (b) della prova 3

		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	90.1	2.3	2.5	5.1
	P	17.8	80.9	0.3	1.0
	QRS	11.8	0.1	87.9	0.1
	T	11.8	0.1	0.1	88.0

Tabella 10. Confusion matrix test set della prova 3

Prova 4: viene riportato il fold 1, ovvero quello che ha presentato i risultati migliori dall'applicazione dello stratified 5-fold cross-validation. Sul training set, come valori medi, sono state ottenute loss pari a 0.162 e accuracy pari al 93.3%; sul validation set, come valori medi, sono state ottenute loss pari a 0.352 e accuracy pari al 89.0%.

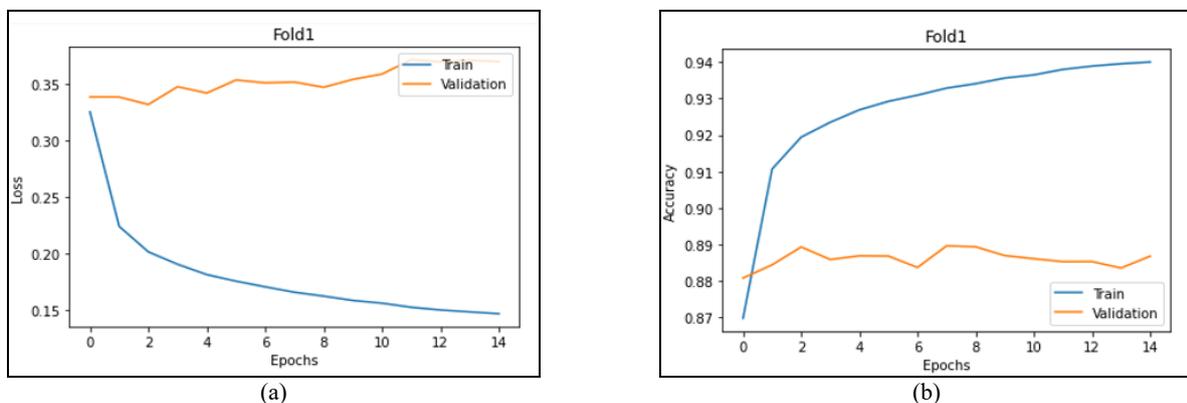
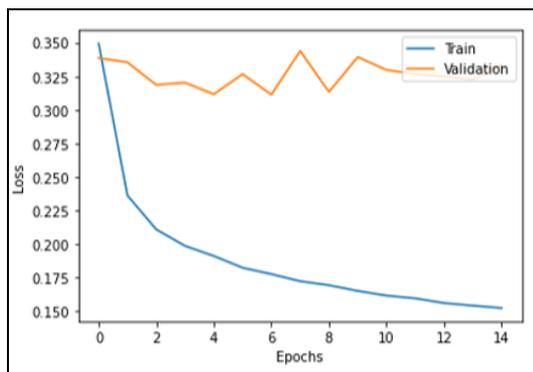


Figura 14. Loss training e validation set (a) e accuracy training e validation set (b) della prova 4

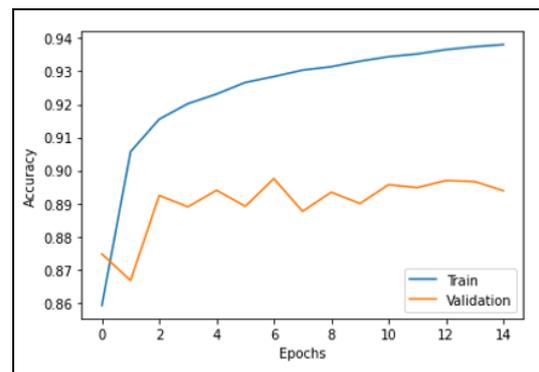
		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	90.3	2.0	2.2	5.5
	P	18.2	81.3	0.0	0.5
	QRS	13.5	0.1	86.2	0.1
	T	11.2	0.1	0.1	88.6

Tabella 11. Confusion matrix test set della prova 4

Prova 5: dall'utilizzo della tecnica Random Search per il tuning degli iperparametri del metodo di ottimizzazione Adam sono stati trovati un valore di α pari a 0.0006736 e un valore di β_1 pari a 0.3. Sul validation set, come valori medi, sono state ottenute loss pari a 0.359 e accuracy pari al 89.9%.



(a)



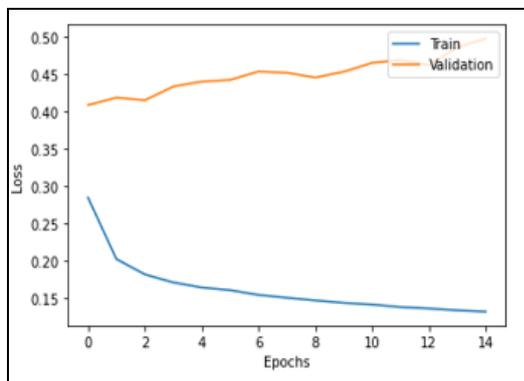
(b)

Figura 15. Loss training e validation set (a) e accuracy training e validation set (b) della prova 5

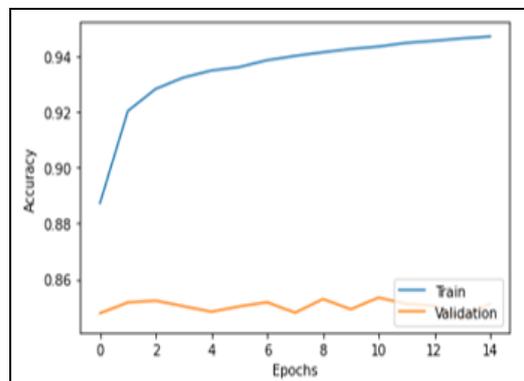
		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	90.0	2.4	2.4	5.2
	P	17.3	82.0	0.1	0.6
	QRS	14.1	0.1	85.6	0.2
	T	12.4	0.2	0.0	87.3

Tabella 12. Confusion matrix test set della prova 5

Prova 6: sul training set, come valori medi, sono state ottenute loss pari a 0.199 e accuracy pari al 93.0%.



(a)



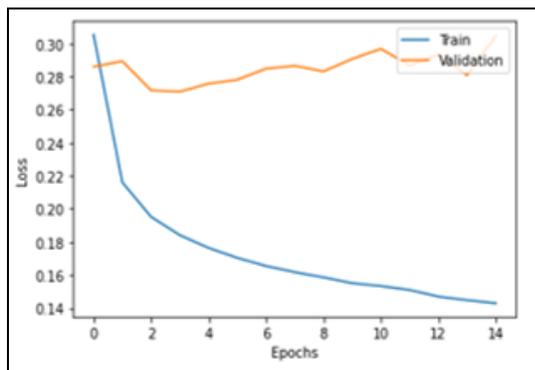
(b)

Figura 16. Loss training e validation set (a) e accuracy training e validation set (b) della prova 6

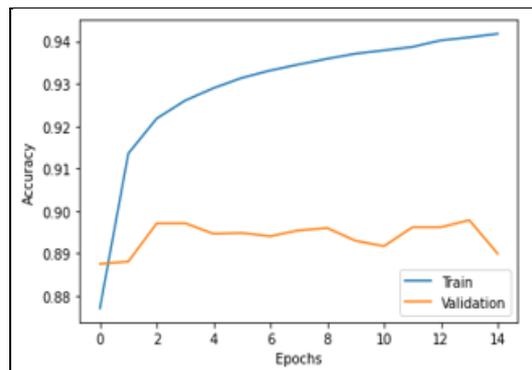
		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	91.3	2.1	2.2	4.4
	P	14.7	84.5	0.1	0.7
	QRS	12.3	0.1	87.2	0.3
	T	11.5	0.2	0.0	88.3

Tabella 13. Confusion matrix test set della prova 6

Prova 7: sul training set, come valori medi, sono state ottenute loss pari a 0.138 e accuracy pari al 94.4%; sul validation set, come valori medi, sono state ottenute loss pari a 0.304 e accuracy pari al 89.0%.



(a)



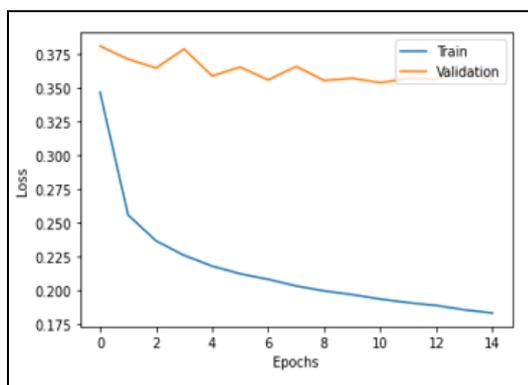
(b)

Figura 17. Loss training e validation set (a) e accuracy training e validation set (b) della prova 7

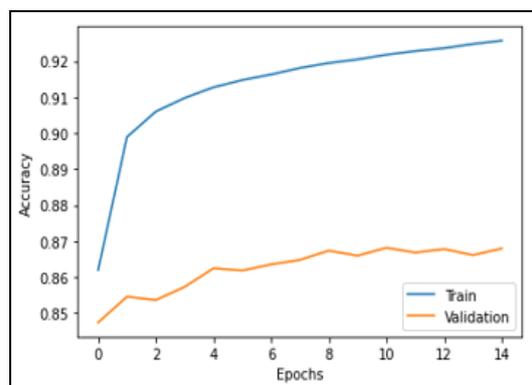
		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	90.9	2.2	2.4	4.5
	P	17.6	81.5	0.1	0.8
	QRS	12.6	0.1	87.2	0.1
	T	11.6	0.1	0.1	88.2

Tabella 14. Confusion matrix test set della prova 7

Prova 8: sul training set, come valori medi, sono state ottenute loss pari a 0.213 e accuracy pari al 91.6%.



(a)



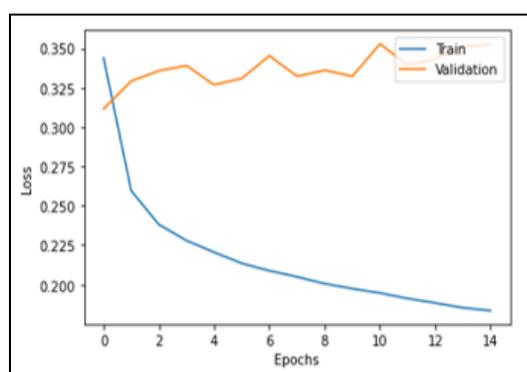
(b)

Figura 18. Loss training e validation set (a) e accuracy training e validation set (b) della prova 8

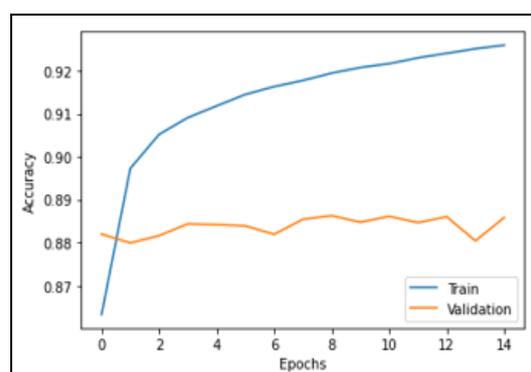
		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	88.4	3.1	1.9	6.5
	P	12.2	86.0	0.1	1.8
	QRS	11.1	0.2	88.5	0.2
	T	15.1	0.4	0.2	84.4

Tabella 15. Confusion matrix test set della prova 8

Prova 9: sul training set, come valori medi, sono state ottenute loss pari a 0.210 e accuracy pari al 92.1%.



(a)



(b)

Figura 19. Loss training e validation set (a) e accuracy training e validation set (b) della prova 9

		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	90.6	2.9	2.2	4.3
	P	12.5	86.1	0.1	1.2
	QRS	10.8	0.1	89.0	0.1
	T	16.3	0.2	0.1	83.4

Tabella 16. Confusion matrix test set della prova 9

Prova 10: sul training set, come valori medi, sono state ottenute loss pari a 0.169 e accuracy pari al 93.2%; sul validation set, come valori medi, sono state ottenute loss pari a 0.357 e accuracy pari al 88.1%.

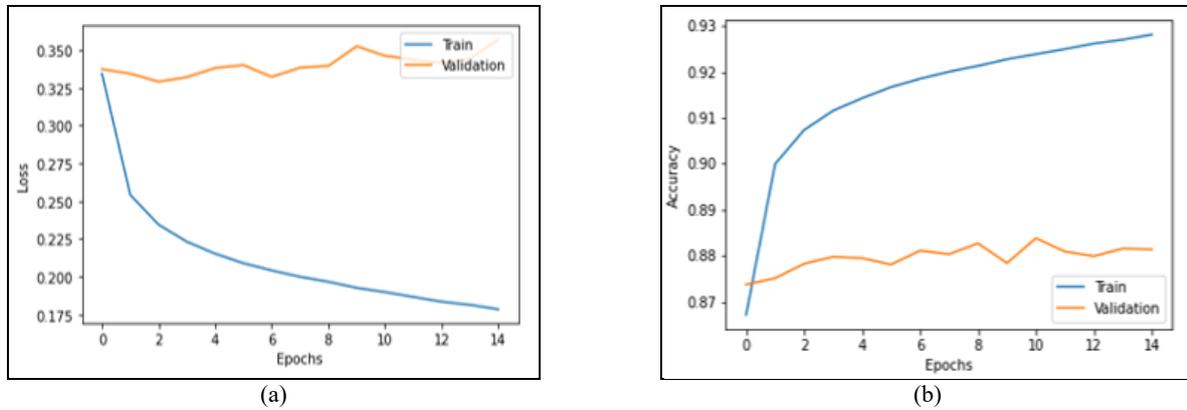
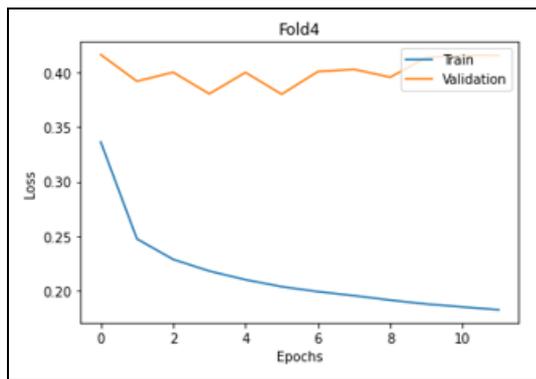


Figura 20. Loss training e validation set (a) e accuracy training e validation set (b) della prova 10

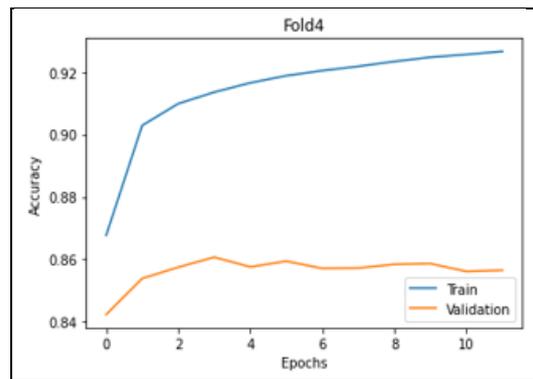
		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	89.0	3.1	2.3	5.7
	P	13.4	85.9	0.1	0.6
	QRS	10.1	0.1	89.7	0.2
	T	15.6	0.4	0.1	83.9

Tabella 17. Confusion matrix test set della prova 10

Prova 11: viene riportato il fold 4, ovvero quello che ha presentato i risultati migliori dall'applicazione dello stratified 5-fold cross-validation. Sul validation set, come valori medi, sono state ottenute loss pari a 0.380 e accuracy pari al 85.9%.



(a)



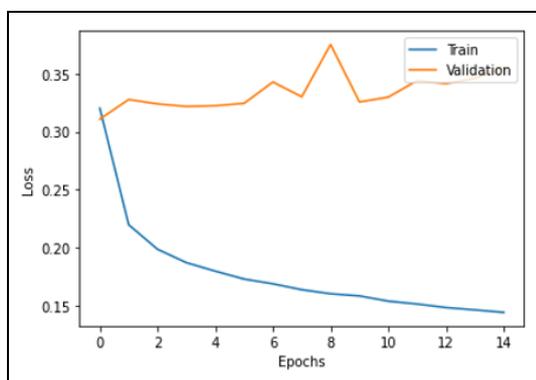
(b)

Figura 21. Loss training e validation set (a) e accuracy training e validation set (b) della prova 11

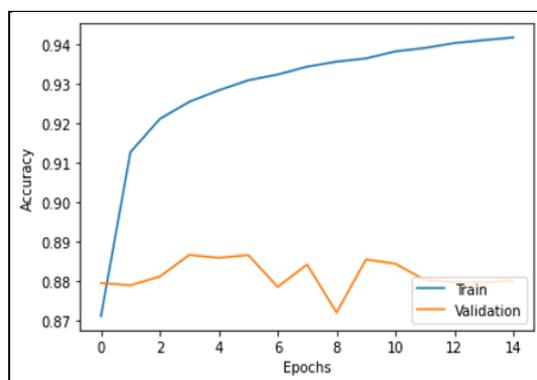
		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	89.5	2.7	2.1	5.8
	P	13.3	85.4	0.2	1.1
	QRS	11.9	0.1	88.0	0.1
	T	16.3	0.2	0.1	83.4

Tabella 18. Confusion matrix test set della prova 11

Prova 12: viene riportato il fold 5, ovvero quello che ha presentato i risultati migliori dall'applicazione dello stratified 5-fold cross-validation. Sul training set, come valori medi, sono state ottenute loss pari a 0.138 e accuracy pari al 94.4%; sul validation set, come valori medi, sono state ottenute loss pari a 0.355 e accuracy pari al 88.0%.



(a)



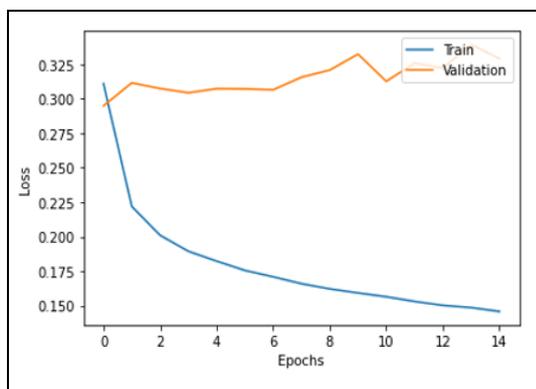
(b)

Figura 22. Loss training e validation set (a) e accuracy training e validation set (b) della prova 12

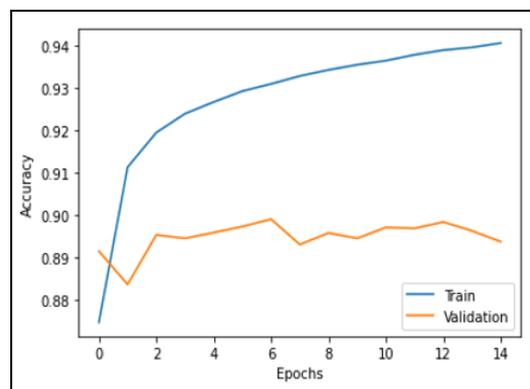
		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	91.6	2.1	2.1	4.3
	P	18.0	81.2	0.1	0.7
	QRS	14.0	0.1	85.7	0.2
	T	11.8	0.2	0.0	87.9

Tabella 19. Confusion matrix test set della prova 12

Prova 13: viene riportato il fold 4, ovvero quello che ha presentato i risultati migliori dall'applicazione dello stratified 5-fold cross-validation. Sul training set, come valori medi, sono state ottenute loss pari a 0.139 e accuracy pari al 94.3%; sul validation set, come valori medi, sono state ottenute loss pari a 0.329 e accuracy pari al 89.4%.



(a)



(b)

Figura 23. Loss training e validation set (a) e accuracy training e validation set (b) della prova 13

		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	90.7	2.1	2.3	4.9
	P	16.6	82.4	0.2	0.8
	QRS	13.0	0.1	86.8	0.2
	T	11.8	0.1	0.1	87.9

Tabella 20. Confusion matrix test set della prova 13

Prova 14: viene riportato il fold 4, ovvero quello che ha presentato i risultati migliori dall'applicazione dello stratified 5-fold cross-validation. Sul training set, come valori medi, sono state ottenute loss pari a 0.179 e accuracy pari al 92.8%; sul validation set, come valori medi, sono state ottenute loss pari a 0.298 e accuracy pari al 89.2%.

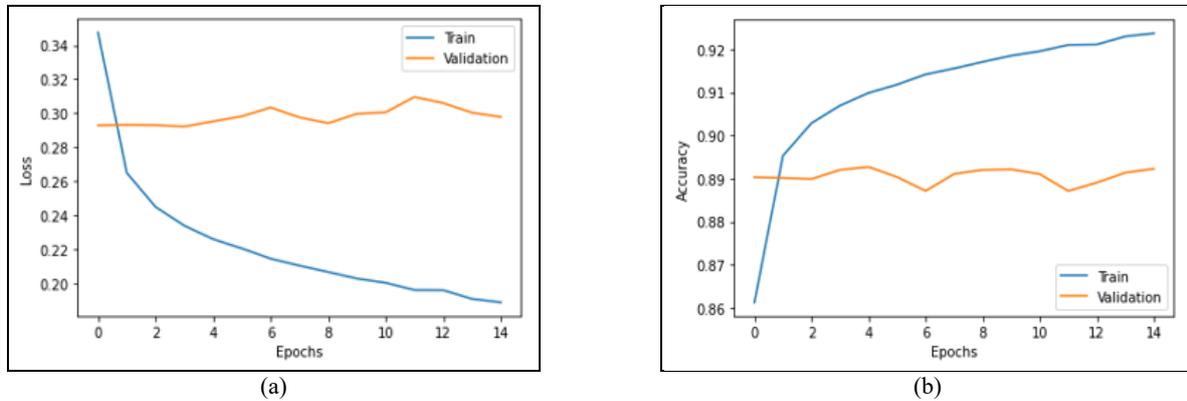
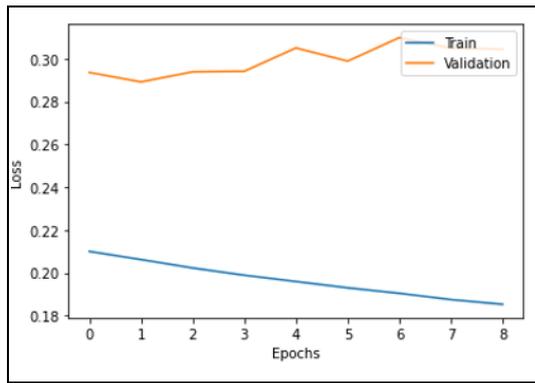


Figura 24. Loss training e validation set (a) e accuracy training e validation set (b) della prova 14

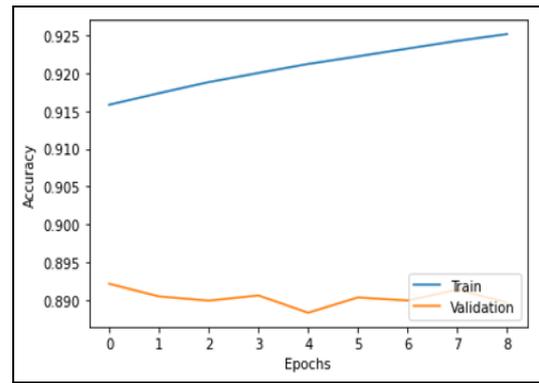
		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	90.0	2.8	2.3	4.9
	P	14.2	85.0	0.1	0.7
	QRS	10.5	0.1	89.3	0.2
	T	15.9	0.3	0.1	83.8

Tabella 21. Confusion matrix test set della prova 14

Prova 15: viene riportato il fold 5, ovvero quello che ha presentato i risultati migliori dall'applicazione dello stratified 5-fold cross-validation. Sul training set, come valori medi, sono state ottenute loss pari a 0.177 e accuracy pari al 92.9%; sul validation set, come valori medi, sono state ottenute loss pari a 0.304 e accuracy pari al 89.0%.



(a)



(b)

Figura 25. Loss training e validation set (a) e accuracy training e validation set (b) della prova 15

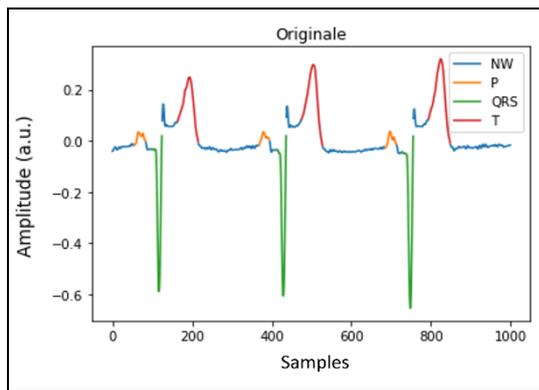
		Classe predetta			
		NW	P	QRS	T
Classe reale	NW	90.6	2.9	2.0	4.5
	P	11.6	87.7	0.1	0.6
	QRS	11.4	0.1	88.4	0.2
	T	15.3	0.4	0.1	84.2

Tabella 22. Confusion matrix test set della prova 15

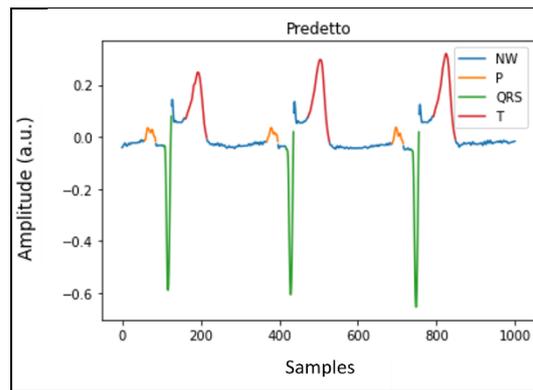
5.1.2 Risultati di riconoscimento ottenuti con il modello scelto

Il modello della quindicesima prova è stato scelto come modello migliore di tutte le prove svolte, poiché presenta la percentuale di corretto riconoscimento dell'onda P più alta (Tabella 22), importante per trovare la baseline nel calcolo dell'area sottesa. Vengono ora riportati alcuni esempi di delineazione dei segnali ECG del test set ottenuti con tale modello, fornendo in input prima i segnali del QT database (Figure 26 - 28) e poi quelli patologici di Brugada (Figure 29 - 31).

5.1.2.1 Esempi delineazione segnali Physionet

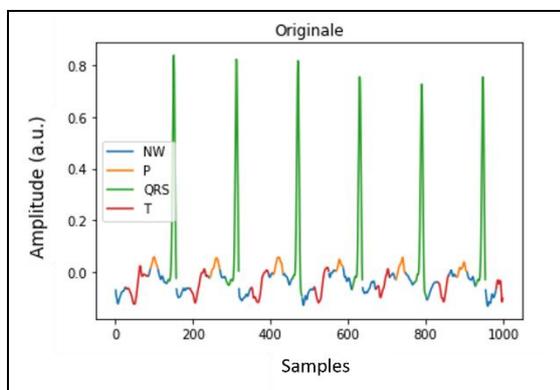


(a)

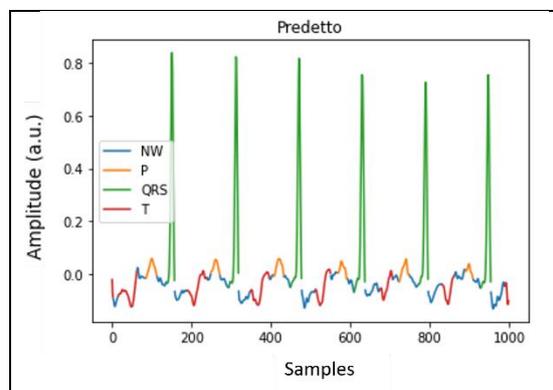


(b)

Figura 26. Primo esempio di riconoscimento: comparazione segnali originale (a) e predetto(b)

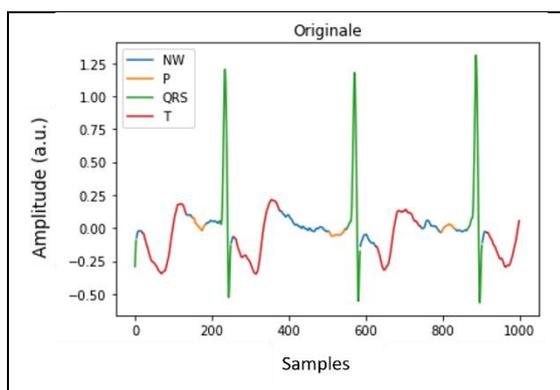


(a)

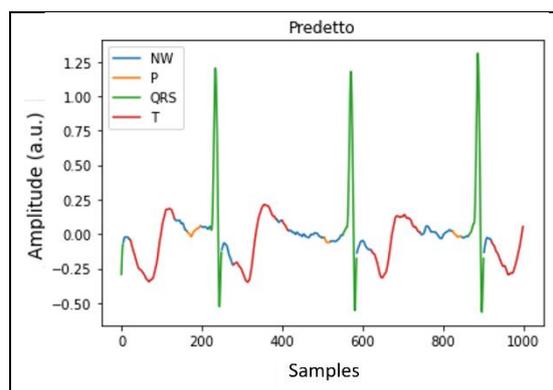


(b)

Figura 27. Secondo esempio di riconoscimento: comparazione segnali originale (a) e predetto (b)



(a)



(b)

Figura 28. Terzo esempio di riconoscimento: comparazione segnali originale (a) e predetto (b)

5.1.2.2 Esempi delineazione segnali di Brugada

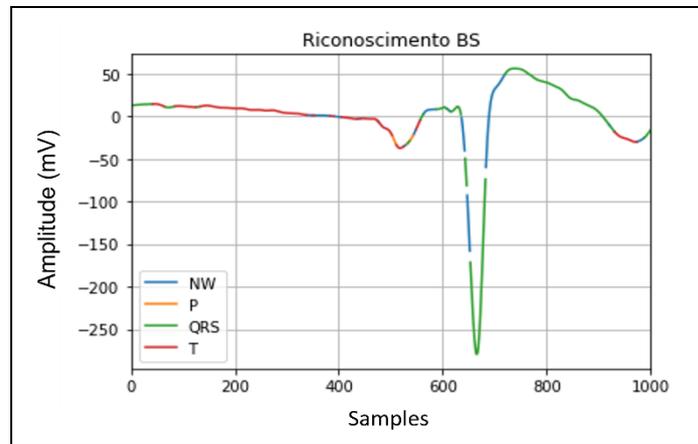


Figura 29. Esempio riconoscimento di un segnale del dataset BS

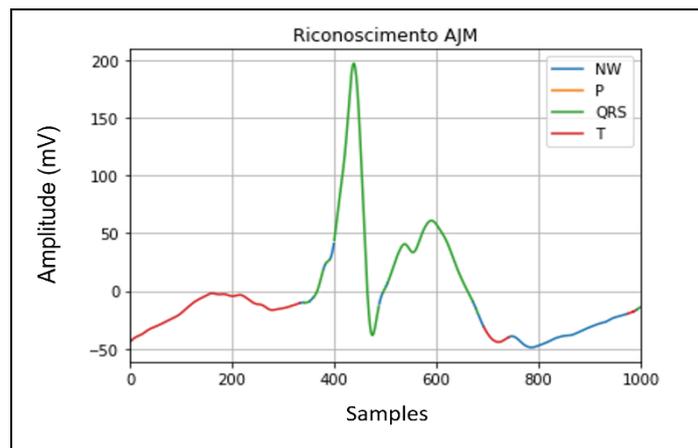


Figura 30. Esempio riconoscimento di un segnale del dataset AJM

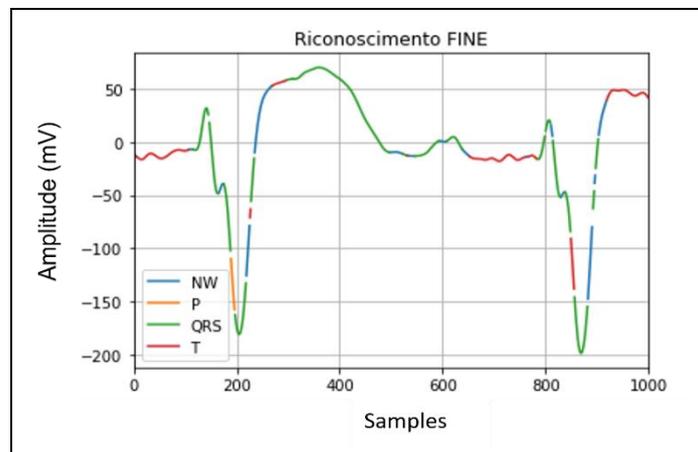


Figura 31. Esempio riconoscimento di un segnale del dataset FINE

5.2 RISULTATI SECONDA FASE

5.2.1 Risultati allenamento del modello con i segnali di Brugada

Vengono ora riportati i grafici della loss e dell'accuracy per i training set dei tre dataset, ovvero BS, AJM e FINE (Figure 32 - 34), durante le successive epoche di allenamento. Vengono anche riportate le confusion matrices ottenute mostrando ai tre modelli allenati i segmenti ECG appartenenti al test set (Tabelle 23 - 25). Ognuna è costruita ponendo le classi reali in riga e le classi predette dal modello di deep neural network in colonna. I valori classificati correttamente per le varie classi sono posti sulla diagonale.

Dataset BS: sul training set, come valori medi, sono state ottenute loss pari a 0.140 e accuracy pari al 94.4%.

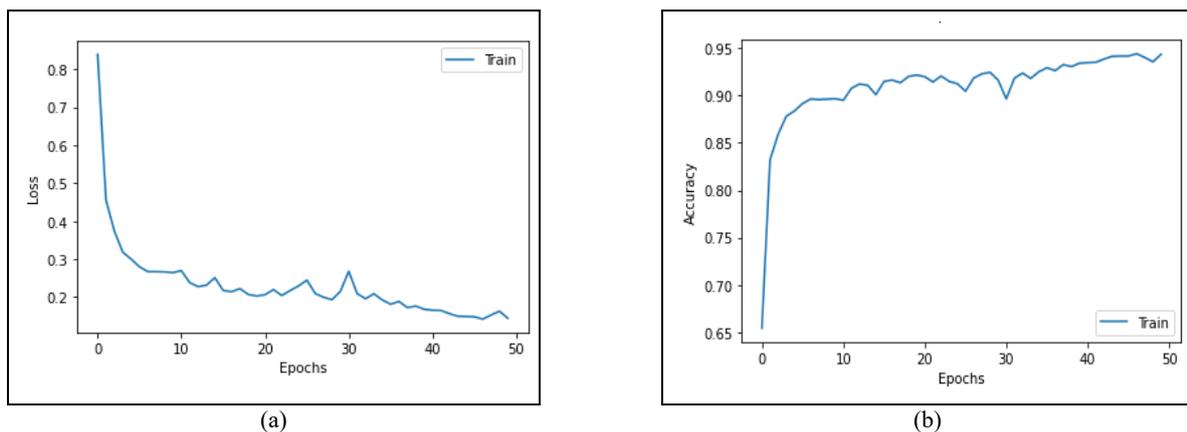


Figura 32. Loss (a) e accuracy (b) training set del dataset BS

		Classe predetta				
		NW	P	QRS	ST	TP
Classe reale	NW	84.3	4.1	0.6	1.0	10.0
	P	3.3	90.6	0.1	0.0	6.0
	QRS	5.6	0.2	93.5	0.8	0.0
	ST	7.5	0.0	0.3	89.2	3.0
	TP	12.0	1.5	0.0	0.0	86.5

Tabella 23. Confusion matrix test set del dataset BS

Dataset AJM: sul training set, come valori medi, sono state ottenute loss pari a 0.102 e accuracy pari al 95.5%.

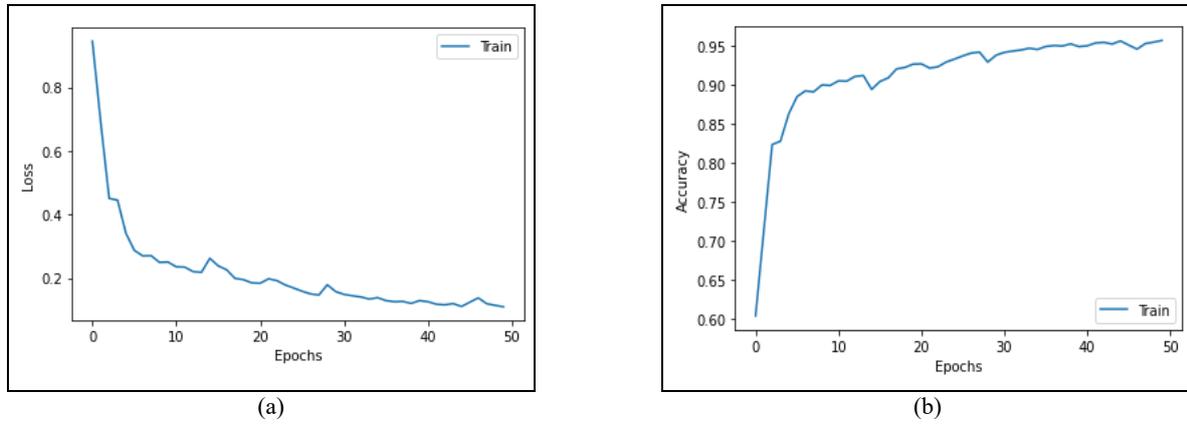


Figura 33. Loss (a) e accuracy (b) training set del dataset AJM

		Classe predetta				
		NW	P	QRS	ST	TP
Classe reale	NW	82.6	8.9	0.7	2.5	5.4
	P	2.6	88.4	0.0	0.0	8.9
	QRS	3.7	0.2	95.8	0.3	0.0
	ST	4.1	0.0	0.9	94.9	0.0
	TP	8.9	5.9	0.0	0.1	85.0

Tabella 24. Confusion matrix test set del dataset AJM

Dataset FINE: sul training set, come valori medi, sono state ottenute loss pari a 0.153 e accuracy pari al 94.1%.

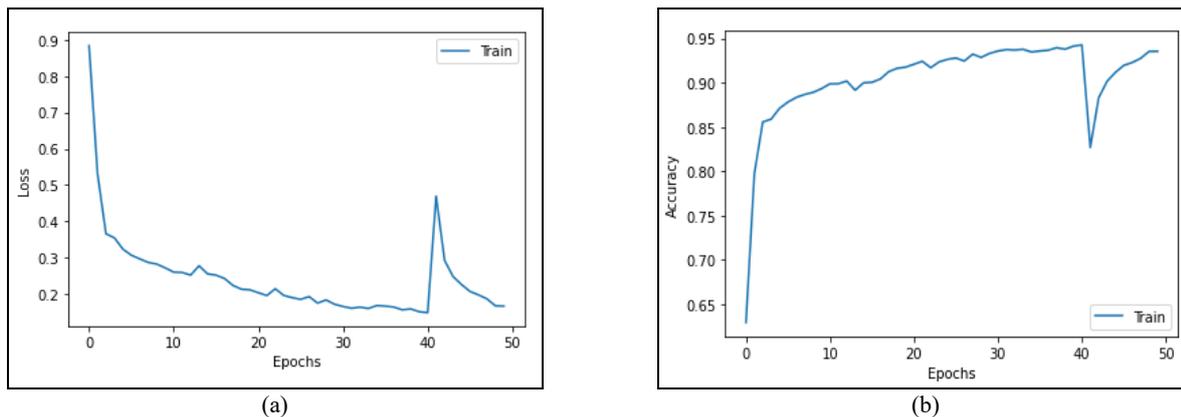


Figura 34. Loss (a) e accuracy (b) training set del dataset FINE

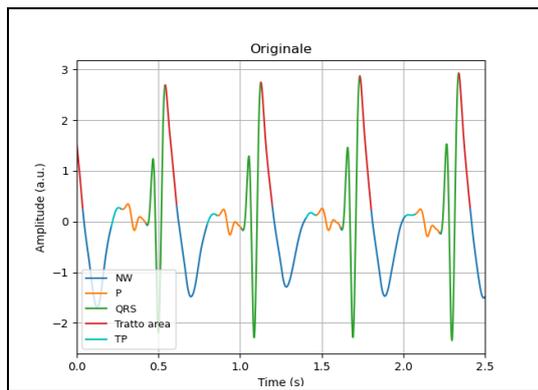
		Classe predetta				
		NW	P	QRS	ST	TP
Classe reale	NW	82.8	3.1	4.6	2.4	7.1
	P	3.9	89.2	1.1	0.0	5.8
	QRS	1.6	0.2	95.6	2.6	0.0
	ST	4.0	0.0	0.9	95.1	0.0
	TP	12.2	4.6	0.1	0.1	83.1

Tabella 25. Confusion matrix test set del dataset FINE

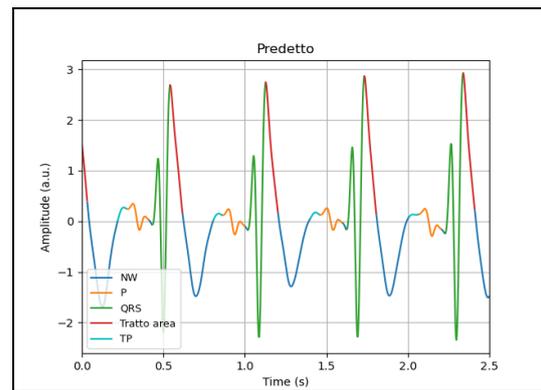
5.2.2 Risultati di riconoscimento ottenuti con i tre modelli allenati

Vengono ora mostrati degli esempi di riconoscimento delle onde e dei tratti ottenuti con i tre modelli allenati, rispettivamente, con i dataset BS (Figure 35 - 37), AJM (Figure 38 - 40) e FINE (Figure 41 - 43). Per ogni dataset sono mostrati tre esempi, dove i segnali utilizzati sono appartenenti al corrispondente test set. In ogni esempio vengono comparati il segnale con i target veri e quello con i target ottenuti dalla deep neural network.

Esempi di riconoscimento per il dataset BS:

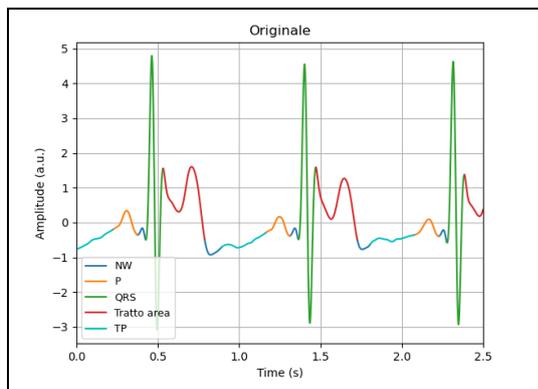


(a)

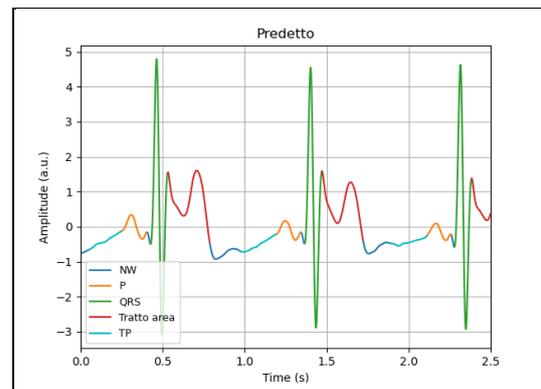


(b)

Figura 35. Primo esempio di riconoscimento dataset BS: comparazione segnali originale (a) e predetto (b)

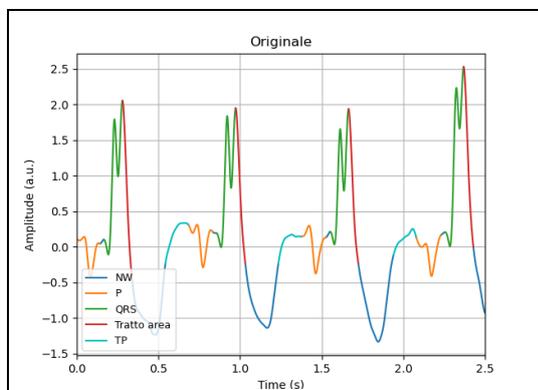


(a)

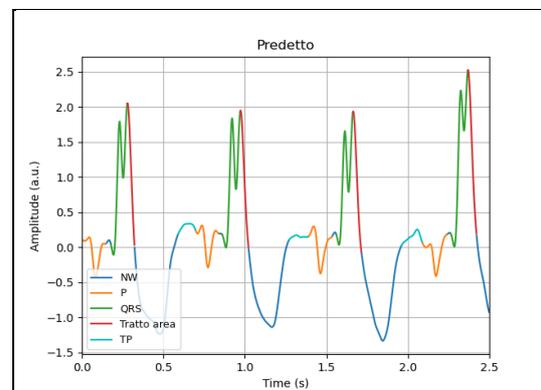


(b)

Figura 36. Secondo esempio di riconoscimento dataset BS: comparazione segnali originale (a) e predetto (b)



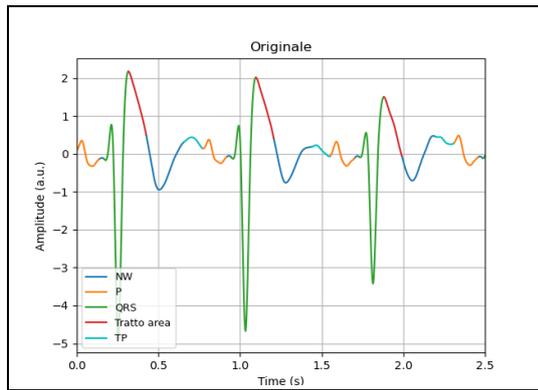
(a)



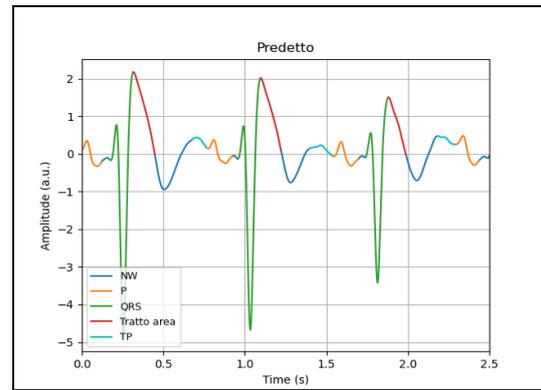
(b)

Figura 37. Terzo esempio di riconoscimento dataset BS: comparazione segnali originale (a) e predetto (b)

Esempi di riconoscimento per il dataset AJM:

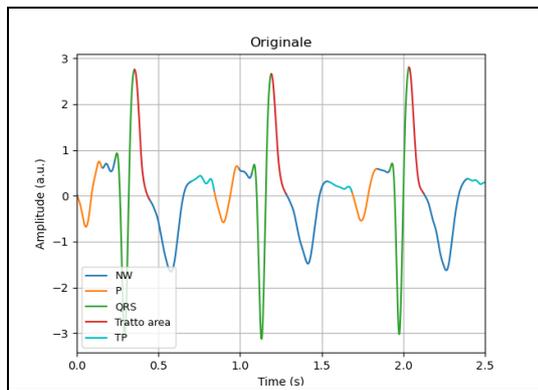


(a)

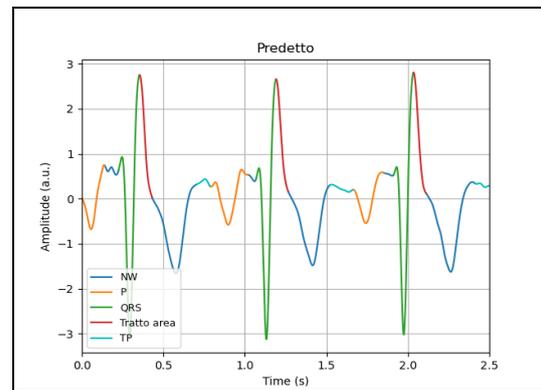


(b)

Figura 38. Primo esempio di riconoscimento dataset AJM: comparazione segnali originale (a) e predetto (b)

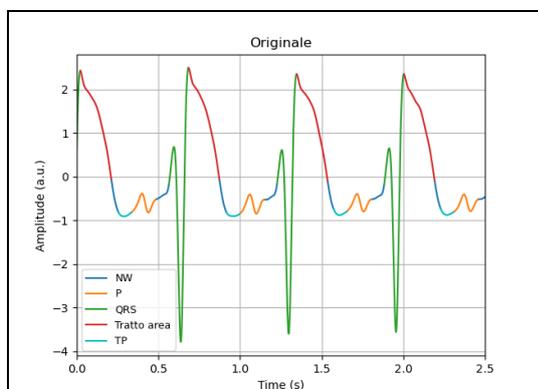


(a)

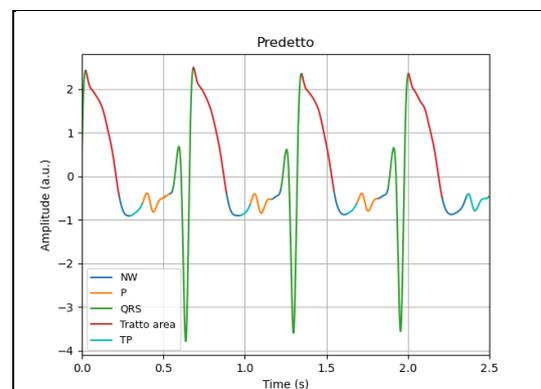


(b)

Figura 39. Secondo esempio di riconoscimento dataset AJM: comparazione segnali originale (a) e predetto (b)



(a)



(b)

Figura 40. Terzo esempio di riconoscimento dataset AJM: comparazione segnali originale (a) e predetto (b)

Esempi di riconoscimento per il dataset FINE:

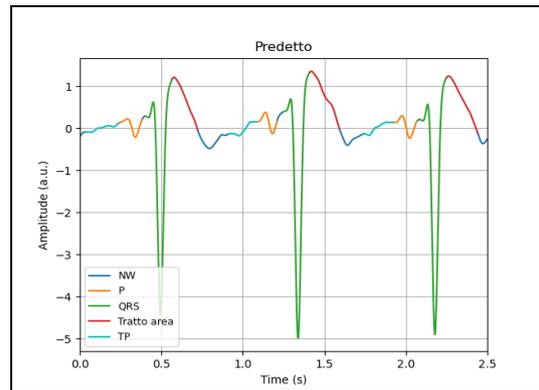
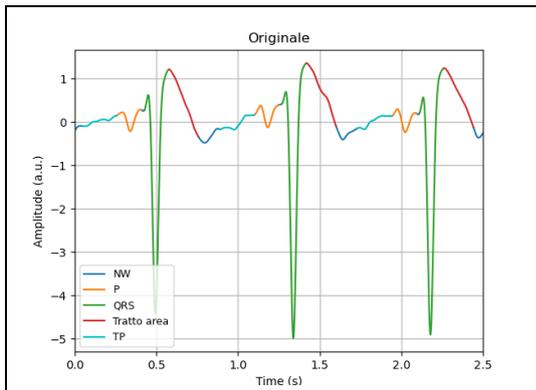
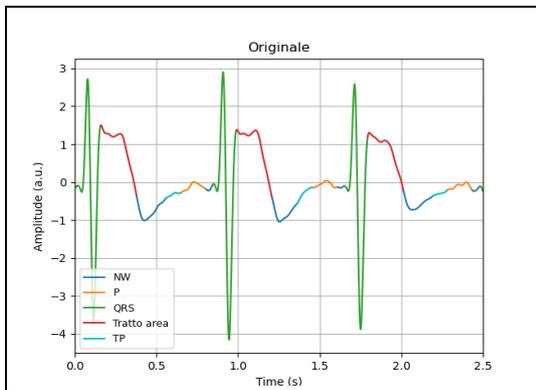
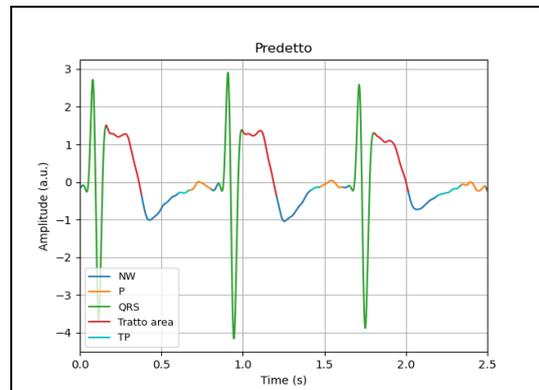


Figura 41. Primo esempio di riconoscimento dataset FINE: comparazione segnali originale (a) e predetto (b)

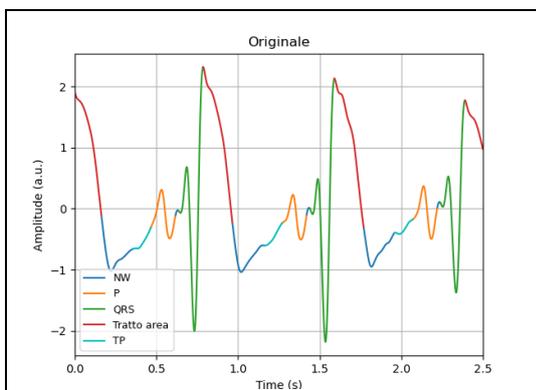


(a)

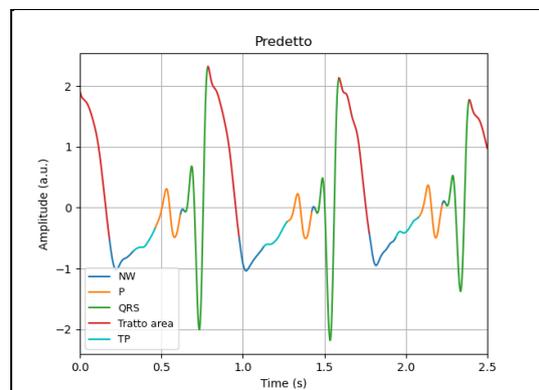


(b)

Figura 42. Secondo esempio di riconoscimento dataset FINE: comparazione segnali originale (a) e predetto (b)



(a)



(b)

Figura 43. Terzo esempio di riconoscimento dataset FINE: comparazione segnali originale (a) e predetto (b)

5.2.3 Risultati calcolo isoelettrica e aree sottese ai tratti ST

Vengono ora mostrati degli esempi di baseline e aree sottese ai tratti ST ottenuti fornendo i risultati della deep neural network all’algoritmo di post-processing. Per ogni dataset sono mostrati tre esempi, dove i segnali utilizzati sono appartenenti al corrispondente test set. In ogni esempio vengono mostrati il segnale con i tratti TP evidenziati e l’isoelettrica calcolata (baseline) e poi il segnale con l’area sottesa indicata da inizio (ST_i) e fine (ST_f).

Baseline e aree sottese calcolate per il dataset BS:

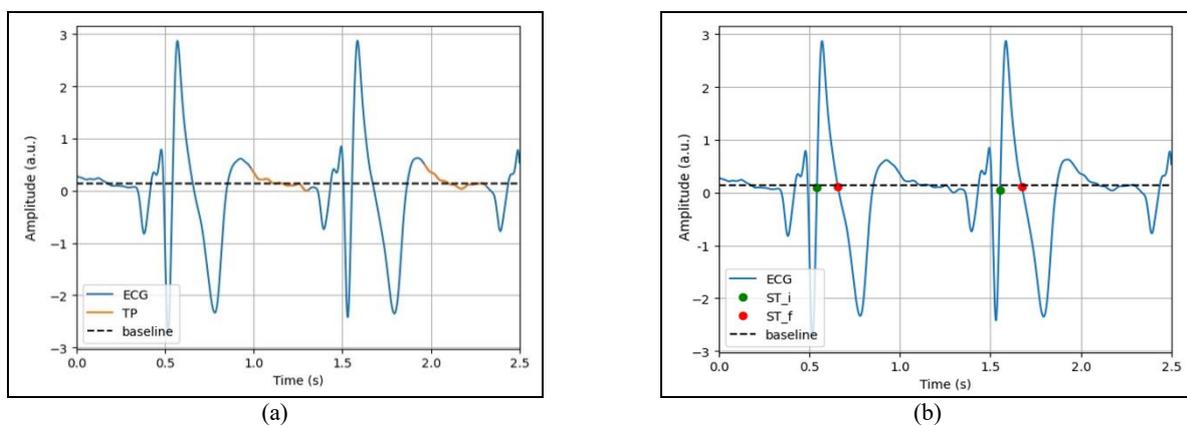


Figura 44. Primo esempio dataset BS: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)

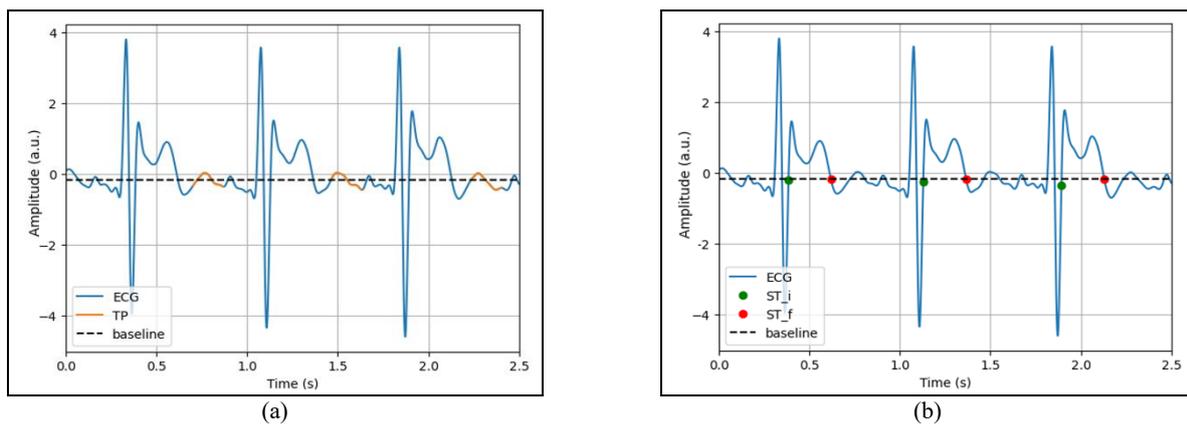
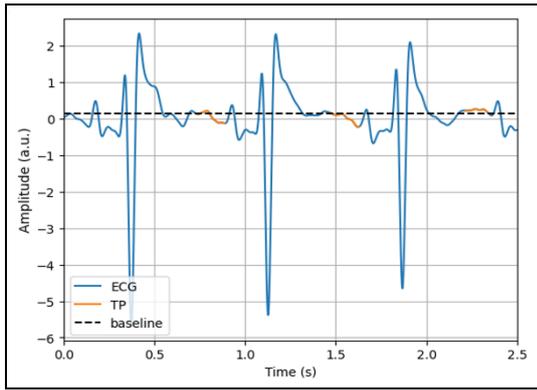
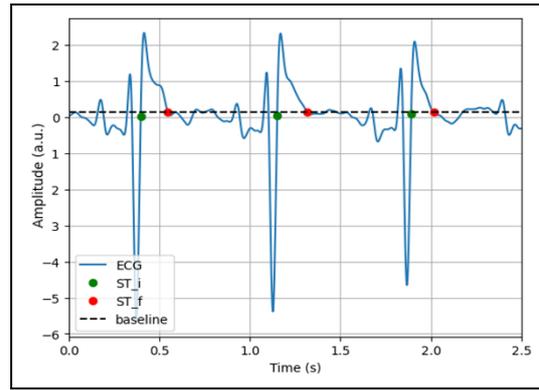


Figura 45. Secondo esempio dataset BS: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)



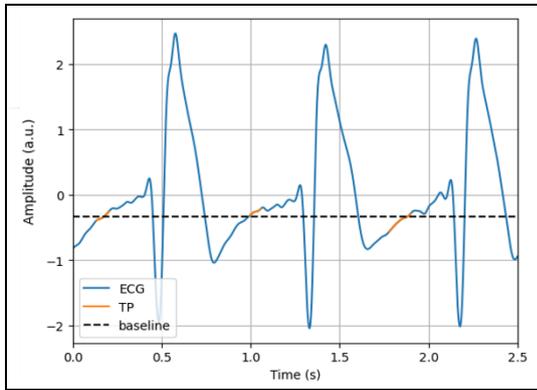
(a)



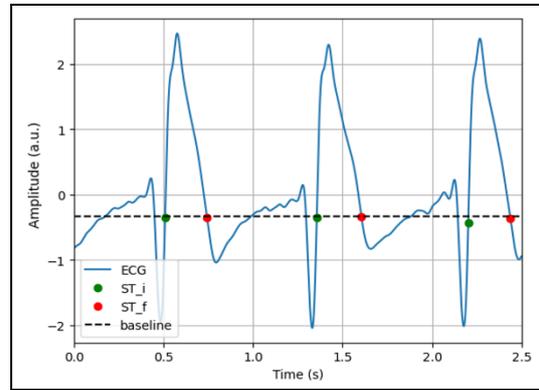
(b)

Figura 46. Terzo esempio dataset BS: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)

Baseline e aree sottese calcolate per il dataset AJM:

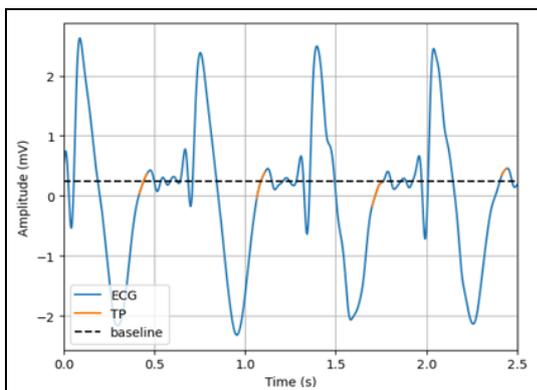


(a)

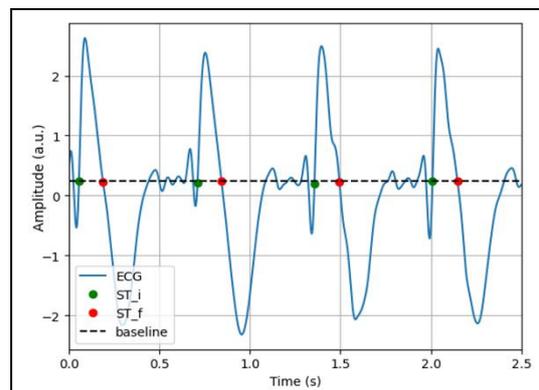


(b)

Figura 47. Primo esempio dataset AJM: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)

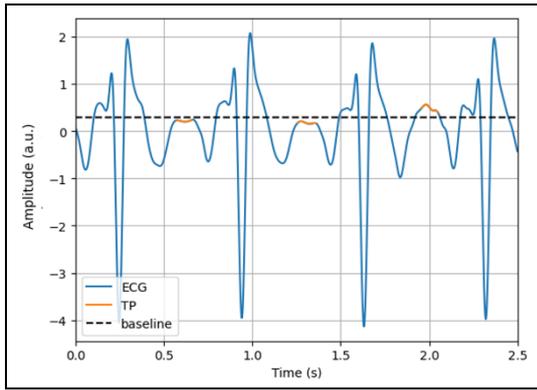


(a)

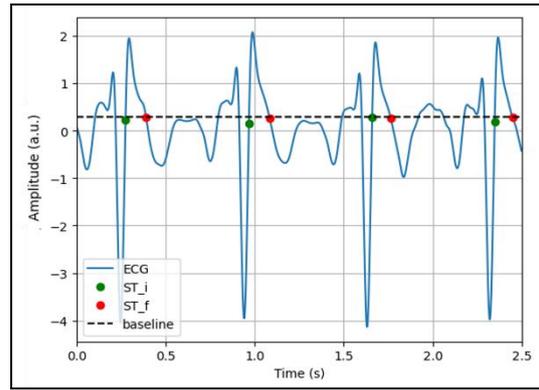


(b)

Figura 48. Secondo esempio dataset AJM: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)



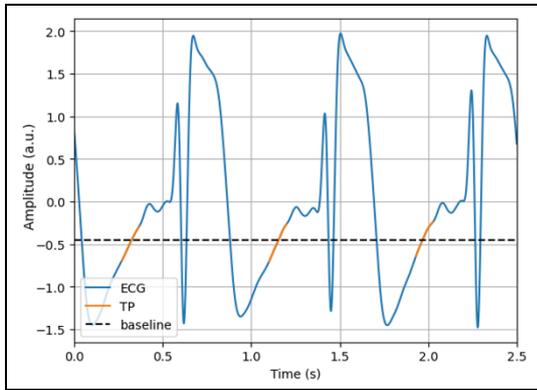
(a)



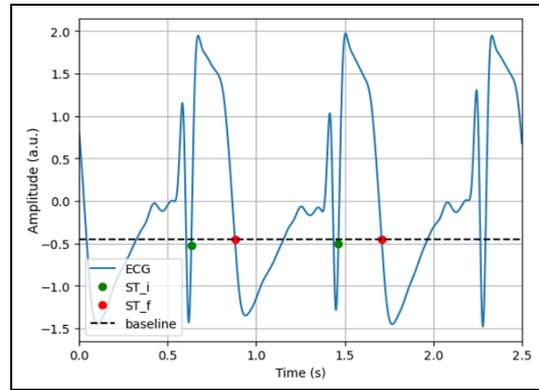
(b)

Figura 49. Terzo esempio dataset AJM: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)

Baseline e aree sottese calcolate per il dataset FINE:

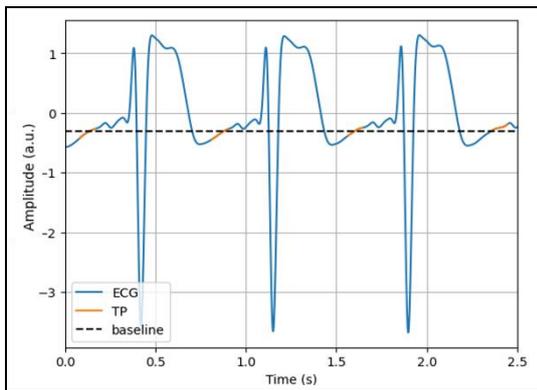


(a)

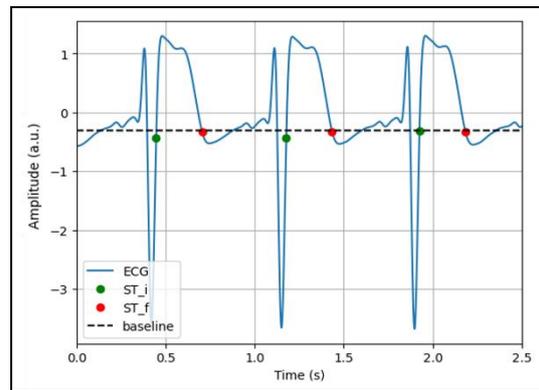


(b)

Figura 50. Primo esempio dataset FINE: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)



(a)



(b)

Figura 51. Secondo esempio dataset FINE: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)

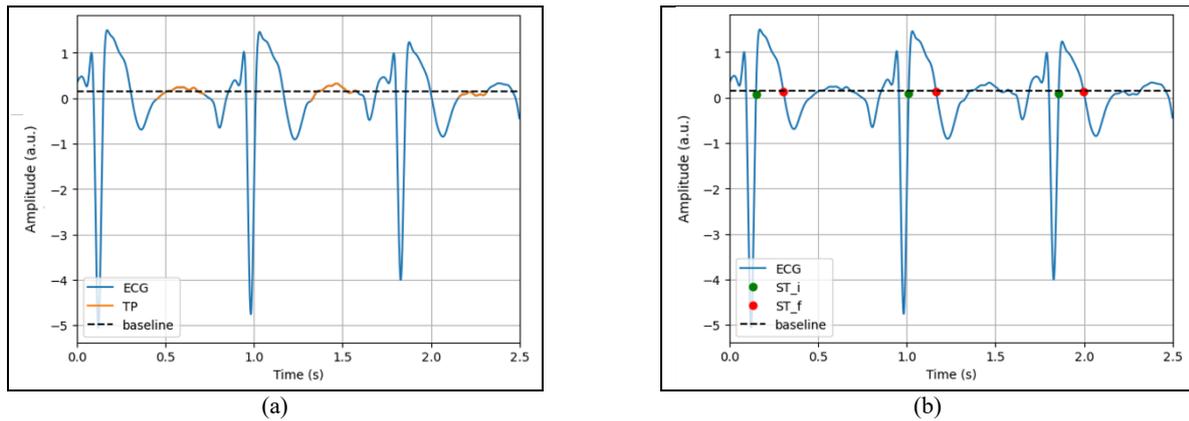


Figura 52. Terzo esempio dataset FINE: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)

5.2.4 Risultati validazione manuale

5.2.4.1 Risultati validazione baseline

Vengono riportati gli errori assoluti calcolati per la validazione della baseline nei tre dataset, sia divisi per derivazione che totali. I valori sono espressi come media \pm deviazione standard.

Derivazione	Errore assoluto (a.u.)		
	BS	AJM	FINE
V1	0,06 \pm 0,05	0,04 \pm 0,05	0,05 \pm 0,05
V2	0,06 \pm 0,06	0,05 \pm 0,05	0,06 \pm 0,05
V1 _{IC2}	0,07 \pm 0,08	0,04 \pm 0,05	0,04 \pm 0,05
V2 _{IC2}	0,06 \pm 0,06	0,05 \pm 0,05	0,05 \pm 0,04

Tabella 26. Errori assoluti di ogni derivazione di ogni dataset

Errore assoluto (a.u.) totale		
BS	AJM	FINE
0,06 \pm 0,06	0,04 \pm 0,05	0,05 \pm 0,05

Tabella 27. Errori totali di ogni dataset

5.2.4.2 Risultati validazione tratti ST

Vengono riportati i valori di accuracy, precision e recall calcolati per la validazione dei tratti ST nei tre dataset, sia divisi per derivazione che totali. I valori sono espressi come media \pm deviazione standard. I valori sono riportati sia in tabella che tramite istogramma.

I valori delle tre metriche per le derivazioni V1, V2, V1_{IC2} e V2_{IC2} per il dataset BS sono:

Derivazione	BS		
	Accuracy (%)	Precision (%)	Recall (%)
V1	94,33 \pm 8,57	93,68 \pm 9,52	82,59 \pm 20,39
V2	99,17 \pm 1,22	97,81 \pm 2,48	98,01 \pm 4,17
V1 _{IC2}	98,86 \pm 0,95	94,51 \pm 8,37	93,03 \pm 6,96
V2 _{IC2}	98,17 \pm 2,49	87,92 \pm 13,50	97,42 \pm 5,35

Tabella 28. Valori di accuracy, precision e recall per le diverse derivazioni per il dataset BS

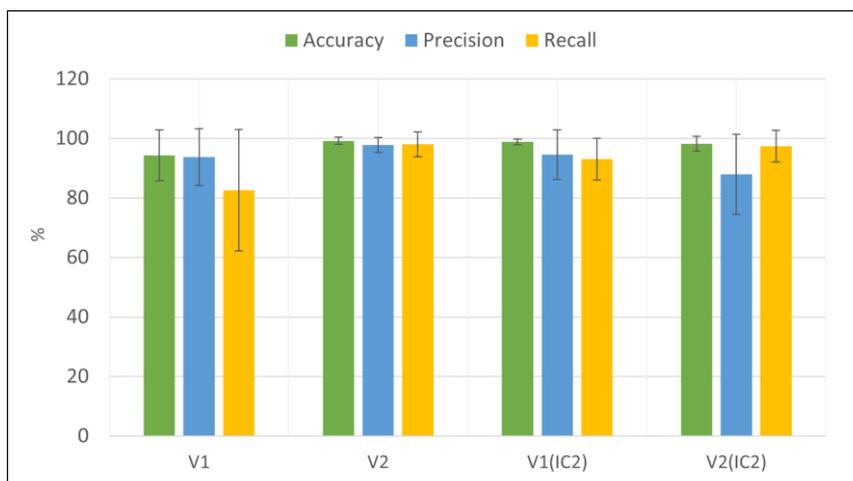


Figura 53. Istogramma di accuracy, precision e recall per le diverse derivazioni per il dataset BS

I valori delle tre metriche per le derivazioni V1, V2, V1_{IC2} e V2_{IC2} per il dataset AJM sono:

Derivazione	AJM		
	Accuracy (%)	Precision (%)	Recall (%)
V1	98,58 ± 0,77	94,70 ± 5,49	94,92 ± 5,11
V2	97,80 ± 1,65	93,38 ± 8,93	95,76 ± 4,43
V1 _{IC2}	98,37 ± 0,94	91,50 ± 8,47	94,83 ± 4,63
V2 _{IC2}	98,16 ± 1,44	93,91 ± 5,49	96,16 ± 5,72

Tabella 29. Valori di accuracy, precision e recall per le diverse derivazioni per il dataset AJM

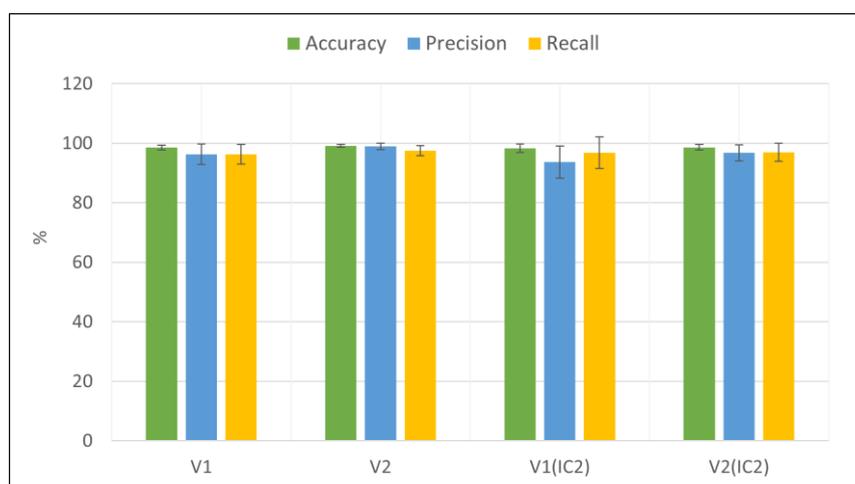


Figura 54. Istogramma di accuracy, precision e recall per le diverse derivazioni per il dataset AJM

I valori delle tre metriche per le derivazioni V1, V2, V1_{IC2} e V2_{IC2} per il dataset FINE sono:

Derivazione	FINE		
	Accuracy (%)	Precision (%)	Recall (%)
V1	98,50 ± 0,79	96,28 ± 3,45	96,22 ± 3,31
V2	99,14 ± 0,49	98,90 ± 1,02	97,45 ± 1,69
V1 _{IC2}	98,29 ± 1,36	93,70 ± 5,37	96,79 ± 5,28
V2 _{IC2}	98,62 ± 0,98	96,75 ± 2,75	96,95 ± 2,97

Tabella 30. Valori di accuracy, precision e recall per le diverse derivazioni per il dataset FINE

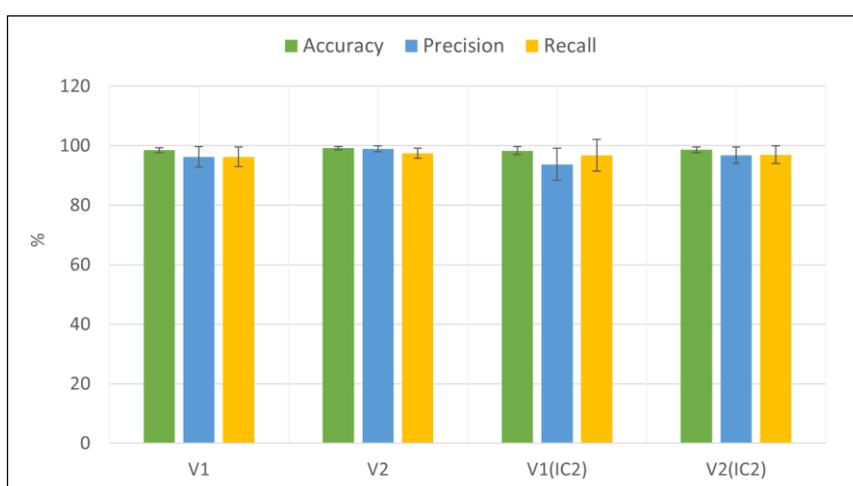


Figura 55. Istogramma di accuracy, precision e recall per le diverse derivazioni per il dataset FINE

I valori delle tre metriche totali per i dataset BS, AJM e FINE sono:

Dataset	Totale (tutte le derivazioni insieme)		
	Accuracy (%)	Precision (%)	Recall (%)
BS	97,63 ± 4,93	93,48 ± 10,00	92,76 ± 12,87
AJM	98,23 ± 1,28	93,37 ± 7,37	95,42 ± 5,03
FINE	98,63 ± 1,01	96,41 ± 3,97	96,85 ± 3,58

Tabella 31. Valori totali di accuracy, precision e recall per i dataset BS, AJM e FINE

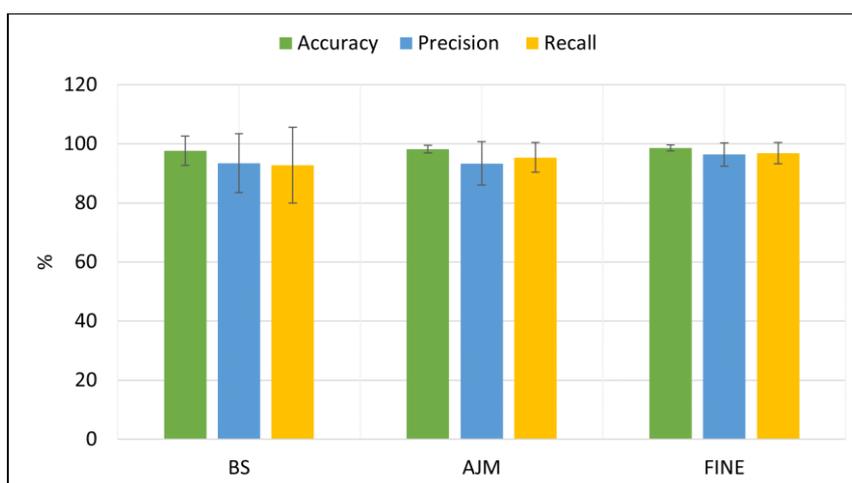


Figura 56. Istogramma con i valori totali di accuracy, precision e recall per i dataset BS, AJM e FINE

5.2.5 Risultati analisi aree sottese

Vengono ora riportati i valori medi delle aree sottese ai tratti ST ottenuti per i tre dataset sia divisi per derivazione (Tabella 32 e Figura 57) che totali (Tabella 33 e Figura 58). I valori sono espressi come media ± deviazione standard. I valori sono riportati sia in tabella che tramite istogramma.

Derivazione	Valori aree sottese (a.u. ²)		
	BS	AJM	FINE
V1	124,37 ± 79,81	156,16 ± 97,78	281,15 ± 120,46
V2	233,15 ± 116,25	281,48 ± 139,62	391,21 ± 95,57
V1 _{IC2}	100,57 ± 42,51	157,45 ± 74,93	277,68 ± 132,98
V2 _{IC2}	169,45 ± 74,51	244,97 ± 136,01	416,70 ± 108,96

Tabella 32. Valori medi aree sottese di ogni derivazione di ogni dataset

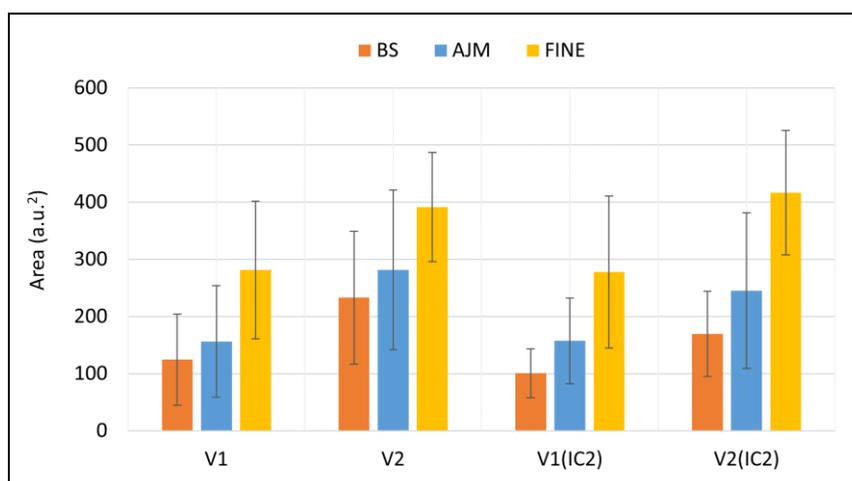


Figura 57. Istogramma valori medi aree sottese di ogni derivazione di ogni dataset

Valori aree sottese (a.u. ²) totale		
BS	AJM	FINE
156,22 ± 96,27	208,65 ± 126,73	341,54 ± 131,45

Tabella 33. Valori aree sottese totali di ogni dataset

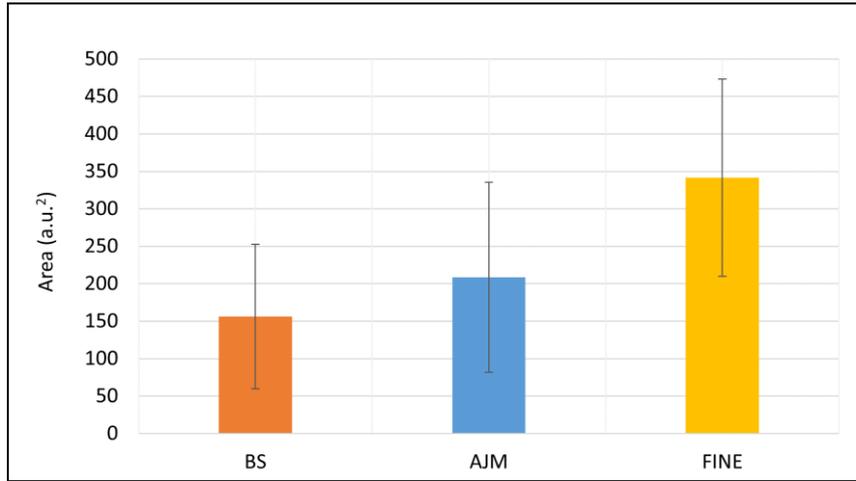


Figura 58. Istogramma valori aree sottese totali di ogni dataset

CAPITOLO 6

DISCUSSIONE

Relativamente ai risultati ottenuti nella prima fase del metodo, le confusion matrices del test set per le diverse prove di allenamento mostrano tutte prestazioni simili. Le funzioni di loss del training set hanno un andamento monotono decrescente, corrisposto ad un accuracy in costante aumento, mentre le prestazioni del validation set sono molto variabili tra le diverse epoche di allenamento. Molte prove mostrano un andamento circa costante per l'accuracy sul validation set, indice che il modello ha dei problemi di overfitting. Tra le varie prove di allenamento svolte è stato scelto il modello della prova 15, perché, sebbene non mostri un netto miglioramento, fornisce però il valore massimo relativamente alla percentuale di corretto riconoscimento dell'onda P nel test set. In questa prima fase del lavoro la detection dell'onda P è stata considerata importante perché si pensava di poterla utilizzare, tramite un apposito algoritmo, per il calcolo del tratto TP corrispondente. I tratti TP trovati sarebbero poi stati utilizzati per il calcolo della baseline, uno dei parametri chiave per lo studio della Sindrome di Brugada. Il modello scelto fornisce un riconoscimento ottimale dei segnali elettrocardiografici del database QT, ma quando questo è utilizzato per la valutazione dei segnali patologici di Brugada le prestazioni calano drasticamente. La rete non è in grado di riconoscere correttamente nessuna onda e sbaglia anche nella presentazione temporale di queste. Il motivo è dovuto, presumibilmente, ad un diverso range di ampiezza dei segnali, pari a poche unità nei segnali Physionet e a qualche centinaio nei segnali di Brugada, e ad una diversa frequenza di campionamento.

L'adattamento e il riallenamento della rete neurale hanno però mostrato risultati soddisfacenti. Sebbene sia stata svolta un'unica prova, i modelli allenati nelle tre condizioni dimostrano di saper riconoscere le onde e i tratti target con una percentuale di corretti classificati sempre superiore all'80%. I modelli sono in grado di riconoscere bene la sequenza temporale delle varie onde e, anche nel caso di inesattezze, queste sono facilmente risolte dal successivo algoritmo di post-processing. I modelli allenati sbagliano principalmente nel riconoscimento del tratto TP, il quale la maggior parte delle volte viene sovrastimato. Tuttavia, il successivo utilizzo delle mediane per il calcolo della baseline permette di mantenere una buona robustezza dei risultati. L'algoritmo di post-processing realizzato è in grado di calcolare la baseline in modo ottimale; i

risultati di validazione mostrano un errore assoluto in termini di unità arbitrarie molto piccolo per tutti e tre i set. La deviazione standard è molto grande per la presenza di alcuni segnali non riconosciuti correttamente. Gli errori assoluti totali sono simili nei tre set, l'errore più basso è ottenuto nel caso del dataset AJM, mentre l'errore più grande è relativo al dataset BS.

La comparazione dei tratti ST individuati dalla rete con quelli ricavati manualmente mostra un'accuracy superiore al 94% in tutti i set, indice che le prestazioni generali sono molto buone. I valori di precision e recall sono leggermente inferiori, indice che in alcuni casi il riconoscimento delle varie onde viene sovrastimato o sottostimato, ma rimangono comunque apprezzabili. Le prestazioni migliori delle tre metriche sono state ottenute per la derivazione V2 in tutti e tre i set. Il BS nella totalità presenta le prestazioni peggiori, ma i valori delle sue metriche rimangono superiori al 92%.

Nel caso dell'analisi delle aree sottese è possibile notare come le aree aumentino in modo progressivo passando dal dataset BS ad AJM fino ad arrivare a FINE. Questo rapporto viene mantenuto invariato sia nella divisione per derivazione che nei risultati totali.

I risultati ottenuti in quest'ultimo caso sono coerenti con quanto può essere osservato dal punto di vista clinico. L'aumento dell'area sottesa ai tratti ST che si verifica passando dal dataset BS al dataset AJM è spiegato dall'effetto dell'ajmalina. Il farmaco somministrato innesca il pattern di Brugada tipo 1, caratterizzato da un sopraslivellamento del tratto ST, portando così ad un incremento dell'area sottesa. L'aumento dell'area che si verifica passando ai segnali del dataset FINE, acquisiti subito dopo l'intervento di ablazione chirurgica del substrato aritmico, è dovuto alla formazione della cosiddetta "onda di lesione" nel segnale ECG, ovvero il sopraslivellamento del segmento ST che si osserva più frequentemente negli stadi iniziali di un infarto miocardico. I potenziali anomali, rilevabili a livello del tracciato elettrocardiografico, sono dovuti al tessuto cardiaco lesionato dalle emissioni a radiofrequenza dell'intervento chirurgico. La normalizzazione dell'ECG, e quindi la diminuzione delle aree sottese dovuta alla scomparsa dell'onda di lesione e del pattern tipico di Brugada, è osservabile solo dopo qualche tempo dall'intervento.

Gli incrementi nelle tre condizioni variano a seconda della derivazione considerata, fatto dovuto principalmente alle diverse morfologie dei tracciati elettrocardiografici.

CAPITOLO 7

CONCLUSIONI

In conclusione, è possibile affermare che la Sindrome di Brugada è un argomento le cui informazioni a riguardo sono aumentate negli ultimi anni. La ricerca in questo campo è di fondamentale importanza, per supportare i clinici durante la diagnosi e garantire che sempre più pazienti vengano sottoposti alle cure necessarie e adeguate.

Lo scopo di questo lavoro è stato lo sviluppo di un algoritmo automatico per lo studio e l'analisi dei parametri caratteristici della patologia, rilevabili a livello del tracciato elettrocardiografico, in tre condizioni (baseline, somministrazione di ajmalina e subito dopo l'intervento di ablazione). A tal fine è stata implementata una deep neural network per il riconoscimento dei tratti TP e ST, i cui risultati sono stati utilizzati da un algoritmo di post-processing per il calcolo della baseline e delle aree sottese. Questi parametri sono poi stati analizzati per evidenziare eventuali differenze tra le tre condizioni.

L'algoritmo creato funziona in modo automatico: se venissero forniti nuovi segnali ECG questi sarebbero pre-processati, la posizione delle onde e dei tratti sarebbe identificata dal modello allenato e i risultati sarebbero utilizzati per il calcolo delle baseline e delle aree sottese. Il calcolo della baseline svolto in maniera completamente automatica costituisce una novità rispetto a quanto presente in letteratura; la sua identificazione manuale non sempre è semplice e il suo utilizzo è fondamentale nella diagnosi di molte patologie.

Il lavoro svolto è innovativo poiché, non solo offre un supporto alla diagnosi tramite il calcolo dei parametri caratteristici, ma svolge anche un'analisi in ben tre condizioni differenti.

I risultati ottenuti sono incoraggianti e costituiscono il primo passo per altri studi futuri. Utilizzando quando svolto durante questo lavoro potrebbe essere possibile evidenziare nuove conoscenze sulla Sindrome di Brugada in condizioni diverse da quelle analizzate. Ad esempio, si potrebbe valutare come varia il segnale ECG dei pazienti durante il follow-up dell'intervento di ablazione epicardica, per evidenziare il ritorno ad una condizione normale, priva del pattern tipico di Brugada. Altre possibili strade potrebbero essere l'analisi del tracciato elettrocardiografico dopo cure farmacologiche non ancora sperimentate o in seguito a variazioni di abitudini e stile di vita.

I risultati ottenuti fanno ben sperare, offrendo un contributo nella prevenzione dei pazienti

affetti dalla sindrome di Brugada e fornendo nuovi sbocchi a possibili ricerche future, sempre nell'ottica del miglioramento della salute delle persone.

RINGRAZIAMENTI

Il mio primo ringraziamento va al mio relatore, Prof. Filippo Molinari, per aver proposto questa tesi, permettendomi di dare un contributo in prima persona ad un argomento non molto discusso ma estremamente importante. Ringrazio il professore anche per le conoscenze che mi ha fornito in questi anni, con le sue lezioni chiare e coinvolgenti.

Un importante ringraziamento va alla mia correlatrice, Ing. Silvia Seoni, che mi ha accompagnato durante tutti questi mesi, sempre disponibile a fornire consigli e a risolvere i miei dubbi, aiutandomi così a migliorare il mio lavoro e ad arrivare fino alla fine.

Vorrei ringraziare i cardiologi e gli ingegneri dell'A. O. Mauriziano di Torino, che mi hanno aiutato a svolgere il lavoro nel modo corretto fornendomi le giuste valutazioni.

Un grazie infinito va alla mia famiglia, fondamentale nella mia formazione personale, che mi ha supportato e incoraggiato durante tutto il percorso universitario. Innanzitutto ai miei genitori, che mi sono stati vicini nei successi e nelle delusioni, con affetto e instancabile sostegno, credendo in me anche quando io non sono riuscita a farlo. E poi alle mie sorelle, Federica e Caterina, che considero le mie migliori amiche e compagne di vita.

Ringrazio immensamente Maurizio, mia roccia durante tutti questi anni, che mi ha sempre ascoltato e confortato e con il quale ho condiviso tutto. Il suo affetto e le sue parole mi hanno dato la giusta motivazione per continuare il mio percorso e migliorarmi come persona.

Un pensiero speciale va ai miei nonni, miei angeli custodi che vegliano su di me ogni giorno, con cui avrei voluto condividere la gioia di questo traguardo.

Ringrazio i miei amici e le mie amiche per tutti i ricordi e le belle esperienze, per aver condiviso sorrisi e lacrime, aiutandomi a vivere questi anni in modo piacevole e divertente.

BIBLIOGRAFIA

- [1] C. M. Liu, C. L. Liu, K. W. Hu, V. S. Tseng, S. L. Chang, Y. J. Lin, L. W. Lo, F. P. Chung, T. F. Chao, T. C. Tuan, J. N. Liao, C. Y. Lin, T. Y. Chang, C. S. J. Fann, S. Higa, N. Yagi, Y. F. Hu, S. A. Chen. A Deep Learning–Enabled Electrocardiogram Model for the Identification of a Rare Inherited Arrhythmia: Brugada Syndrome. *Canadian Journal of Cardiology*, Volume 38, Issue 2, 152-159 (2022).
- [2] A.Y. Hannun, P. Rajpurkar, M. Haghpanahi, et al. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nat Med* 25, 65–69 (2019).
- [3] A. Peimankar, S. Puthusserypady. DENS-ECG: A deep learning approach for ECG signal delineation. *Expert Systems with Applications*, Volume 165, 113911 (2021).
- [4] R. Brugada, O. Campuzano, G. Sarquella-Brugada, J. Brugada, P. Brugada. Brugada syndrome. *Methodist DeBakey Cardiovasc J.*, 10(1):25-8 (2014).
- [5] A. Nava, B. Canciani, B. Martini, et al. La ripolarizzazione precoce nelle precordiali destre. Correlazioni ECG-VCGelettrofisiologia. *Giornale Italiano di Cardiologia*, 18 supplemento 1, 118 (1988).
- [6] B. Martini, A. Nava, G. F. Buja, et al. Fibrillazione ventricolare in apparente assenza di cardiopatia. Descrizione di 6 casi. *Giornale Italiano di Cardiologia*, 18 supplemento 1, 136 (1988).
- [7] P. Brugada, J. Brugada. Right bundle branch block, persistent ST segment elevation and sudden cardiac death: a distinct clinical and electrocardiographic syndrome. A multicenter report. *J Am Coll Cardiol*, 20(6):1391-6 (1992).
- [8] J. M. Gehshan, D. Rizzolo. Understanding Brugada syndrome. *JAAPA*, 28(6):32-6. (2015).
- [9] C. Antzelevitch, N. Eyal. Brugada syndrome: recent advances and controversies. *Current cardiology reports vol. 10*, 5, 376-83 (2008).
- [10] A.A. Wilde, C. Antzelevitch, M. Borggrefe, et al. Proposed diagnostic criteria for the Brugada syndrome: consensus report. *Circulation*, 106: 2514-2519 (2002).

- [11] C. Antzelevitch, P. Brugada, M. Borggrefe, et al. Brugada Syndrome. *Report of the second consensus conference: endorsed by the Heart Rhythm Society and the European Heart Rhythm Association*, 659-670 (2005).
- [12] D. G. Iglesias, J. Rubín, D. Pérez, C. Morís, D. Calvo. Insights for Stratification of Risk in Brugada Syndrome. *Eur Cardiol*, 14(1):45-49 (2019).
- [13] O. S. Lih, V. Jahmunah, T. R. San, E. J. Ciaccio, T. Yamakawa, M. Tanabe, M. Kobayashi, O. Faust, U. R. Acharya. Comprehensive electrocardiographic diagnosis based on deep learning. *Artif Intell Med*, 103:101789 (2020).
- [14] J. Huang, B. Chen, B. Yao, W. He. ECG Arrhythmia Classification Using STFT-Based Spectrogram and Convolutional Neural Network. *IEEE Access*, vol. 7, 92871-92880 (2019).
- [15] L. Maršánová, A. Němcová, R. Smíšek, et al. Advanced P Wave Detection in Ecg Signals During Pathology: Evaluation in Different Arrhythmia Contexts. *Sci Rep* 9, 19053 (2019).
- [16] S. Mousavi, F. Afghah, A. Razi, U. R. Acharya. ECGNET: Learning Where to Attend for Detection of Atrial Fibrillation with Deep Visual Attention. *IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, 1-4 (2019).
- [17] G. Jimenez-Perez, A. Alcaine, O. Camara. U-Net Architecture for the Automatic Detection and Delineation of the Electrocardiogram. *2019 Computing in Cardiology (CinC)*, 1-4 (2019).
- [18] V. A. Moskalenko, N. Y. Zolotykh, G. Osipov. Deep Learning for ECG Segmentation. *ArXiv*, abs/2001.04689 (2019).
- [19] G. Jimenez-Perez, A. Alcaine, O. Camara. Delineation of the electrocardiogram with a mixed-quality-annotations dataset using convolutional neural networks. *Sci Rep* 11, 863 (2021).
- [20] P. Laguna, R. G. Mark, A. L. Goldberger, G. B. Moody. A Database for Evaluation of Algorithms for Measurement of QT and Other Waveform Intervals in the ECG. *Computers in Cardiology*, 24:673-676 (1997).
- [21] A.L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, H. E. Stanley. Physiobank, Physiokit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101, e215–e220 (2000).
- [22] Y. LeCun, Y. Bengio, G. Hinton. Deep learning. *Nature*, 521, 436–444 (2015).

- [23] I. Goodfellow, Y. Bengio, A. Courville. *Deep learning*. MIT press (2016).
- [24] S. Hochreiter, J. Schmidhuber. Long short-term memory. *Neural Computation*, 9, 1735–1780 (1997).
- [25] Y. Bengio, P. Simard, P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5, 157–166 (1994).
- [26] J. Chung, C. Gulcehre, K. Cho, Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv*, preprint arXiv, 1412.3555 (2014).
- [27] A.Graves, N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. *International conference on machine learning*, 1764–1772 (2014).
- [28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, Volume 15, Issue 1, 1929–1958 (2014).
- [29] Y. Verma. A Complete Understanding of Dense Layers in Neural Networks. *Developers Corner* (2021).
- [30] D. P. Kingma, J. Ba. A method for stochastic optimization. *ArXiv*, preprint arXiv, 1412.6980. (2014).
- [31] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in neural information processing systems*, 2933–2941 (2014).
- [32] F. Chollet, et al. Keras (2015).
- [33] J. Bergstra, Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*. 13, 281–305 (2012).
- [34] J. Friedman, T. Hastie, R. Tibshirani. The elements of statistical learning volume 1. *Springer series in statistics* (2001).
- [35] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *14th International joint conference on artificial intelligence (IJCAI)*, Vol. 2, 1137–1143 (1995).
- [36] E. Bisong. Google Colaboratory. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Apress, Berkeley, CA (2019)

- [37] C. Antzelevitch, B. Patocskaï. Ajmaline-Induced Slowing of Conduction in the Right Ventricular Outflow Tract Cannot Account for ST Elevation in Patients With Type I Brugada ECG. *Circulation: Arrhythmia and Electrophysiology* (2017).
- [38] J. Brugada, C. Pappone, A. Berruezo, G. Vicedomini, F. Manguso, G. Ciconte, L. Giannelli, V. Santinelli. Brugada Syndrome Phenotype Elimination by Epicardial Substrate Ablation. *Circulation: Arrhythmia and Electrophysiology*, 8:1373–1381 (2015).
- [39] Trapezium formula. *Encyclopedia of Mathematics*. URL: http://encyclopediaofmath.org/index.php?title=Trapezium_formula&oldid=12696

LISTA DELLE FIGURE

Figura 1. Pattern caratteristici (tipi 1, 2 e 3) della Sindrome di Brugada nelle derivazioni V1-V6. Fonte: lifeinthefastlane.com	11
Figura 2. Schema della divisione del dataset e del transfer learning. Liu et al [1]	14
Figura 3. Schema del lavoro svolto	15
Figura 4. Architettura deep neural network implementata da [3]	19
Figura 5. Architettura deep neural network modificata	36
Figura 6. Flowchart algoritmo di post-processing	38
Figura 7. Esempio della finestra con il marker rosso per la selezione dei punti sul segmento ECG.....	40
Figura 8. Esempio di selezione punti di inizio e fine tratti TP	41
Figura 9. Esempio di tratti TP e baseline determinati con la selezione manuale.....	41
Figura 10. Esempio di tratti ST determinati con la selezione manuale.....	42
Figura 11. Loss training set (a) e accuracy training set (b) della prova 1	45
Figura 12. Loss training e validation set (a) e accuracy training e validation set (b) della prova 2.....	46
Figura 13. Loss training e validation set (a) e accuracy training e validation set (b) della prova 3.....	47
Figura 14. Loss training e validation set (a) e accuracy training e validation set (b) della prova 4.....	47
Figura 15. Loss training e validation set (a) e accuracy training e validation set (b) della prova 5.....	48
Figura 16. Loss training e validation set (a) e accuracy training e validation set (b) della prova 6.....	49
Figura 17. Loss training e validation set (a) e accuracy training e validation set (b) della prova 7.....	50
Figura 18. Loss training e validation set (a) e accuracy training e validation set (b) della prova 8.....	50
Figura 19. Loss training e validation set (a) e accuracy training e validation set (b) della prova 9.....	51
Figura 20. Loss training e validation set (a) e accuracy training e validation set (b) della prova	

10.....	52
Figura 21. Loss training e validation set (a) e accuracy training e validation set (b) della prova 11.....	53
Figura 22. Loss training e validation set (a) e accuracy training e validation set (b) della prova 12.....	53
Figura 23. Loss training e validation set (a) e accuracy training e validation set (b) della prova 13.....	54
Figura 24. Loss training e validation set (a) e accuracy training e validation set (b) della prova 14.....	55
Figura 25. Loss training e validation set (a) e accuracy training e validation set (b) della prova 15.....	56
Figura 26. Primo esempio di riconoscimento: comparazione segnali originale (a) e predetto(b)	57
Figura 27. Secondo esempio di riconoscimento: comparazione segnali originale (a) e predetto (b)	57
Figura 28. Terzo esempio di riconoscimento: comparazione segnali originale (a) e predetto (b)	57
Figura 29. Esempio riconoscimento di un segnale del dataset BS.....	58
Figura 30. Esempio riconoscimento di un segnale del dataset AJM.....	58
Figura 31. Esempio riconoscimento di un segnale del dataset FINE.....	58
Figura 32. Loss (a) e accuracy (b) training set del dataset BS	59
Figura 33. Loss (a) e accuracy (b) training set del dataset AJM	60
Figura 34. Loss (a) e accuracy (b) training set del dataset FINE	61
Figura 35. Primo esempio di riconoscimento dataset BS: comparazione segnali originale (a) e predetto (b)	62
Figura 36. Secondo esempio di riconoscimento dataset BS: comparazione segnali originale (a) e predetto (b)	62
Figura 37. Terzo esempio di riconoscimento dataset BS: comparazione segnali originale (a) e predetto (b)	62
Figura 38. Primo esempio di riconoscimento dataset AJM: comparazione segnali originale (a) e predetto (b)	63
Figura 39. Secondo esempio di riconoscimento dataset AJM: comparazione segnali originale (a) e predetto (b).....	63
Figura 40. Terzo esempio di riconoscimento dataset AJM: comparazione segnali originale (a)	

e predetto (b)	63
Figura 41. Primo esempio di riconoscimento dataset FINE: comparazione segnali originale (a) e predetto (b)	64
Figura 42. Secondo esempio di riconoscimento dataset FINE: comparazione segnali originale (a) e predetto (b).....	64
Figura 43. Terzo esempio di riconoscimento dataset FINE: comparazione segnali originale (a) e predetto (b)	64
Figura 44. Primo esempio dataset BS: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)	65
Figura 45. Secondo esempio dataset BS: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)	65
Figura 46. Terzo esempio dataset BS: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)	66
Figura 47. Primo esempio dataset AJM: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)	66
Figura 48. Secondo esempio dataset AJM: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)	66
Figura 49. Terzo esempio dataset AJM: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)	67
Figura 50. Primo esempio dataset FINE: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)	67
Figura 51. Secondo esempio dataset FINE: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)	67
Figura 52. Terzo esempio dataset FINE: tratti TP e baseline calcolata (a) mostrati con le aree sottese (b)	68
Figura 53. Istogramma di accuracy, precision e recall per le diverse derivazioni per il dataset BS	69
Figura 54. Istogramma di accuracy, precision e recall per le diverse derivazioni per il dataset AJM.....	70
Figura 55. Istogramma di accuracy, precision e recall per le diverse derivazioni per il dataset FINE	71
Figura 56. Istogramma con i valori totali di accuracy, precision e recall per i dataset BS, AJM e FINE	72
Figura 57. Istogramma valori medi aree sottese di ogni derivazione di ogni dataset	73

Figura 58. Istogramma valori aree sottese totali di ogni dataset..... 74

LISTA DELLE TABELLE

Tabella 1. Codifica numerica applicata ai target scelti	17
Tabella 2. Descrizione divisione del dataset	18
Tabella 3. Creazione VS per divisione percentuale	26
Tabella 4. Creazione VS per divisione dei records	26
Tabella 5. Descrizione dei pazienti nelle tre condizioni	32
Tabella 6. Codifica numerica dei target	33
Tabella 7. Descrizione della divisione dei 3 dataset	35
Tabella 8. Confusion matrix test set della prova 1	46
Tabella 9. Confusion matrix test set della prova 2	46
Tabella 10. Confusion matrix test set della prova 3	47
Tabella 11. Confusion matrix test set della prova 4	48
Tabella 12. Confusion matrix test set della prova 5	48
Tabella 13. Confusion matrix test set della prova 6	49
Tabella 14. Confusion matrix test set della prova 7	50
Tabella 15. Confusion matrix test set della prova 8	51
Tabella 16. Confusion matrix test set della prova 9	51
Tabella 17. Confusion matrix test set della prova 10	52
Tabella 18. Confusion matrix test set della prova 11	53
Tabella 19. Confusion matrix test set della prova 12	54
Tabella 20. Confusion matrix test set della prova 13	54
Tabella 21. Confusion matrix test set della prova 14	55
Tabella 22. Confusion matrix test set della prova 15	56
Tabella 23. Confusion matrix test set del dataset BS	59
Tabella 24. Confusion matrix test set del dataset AJM	60
Tabella 25. Confusion matrix test set del dataset FINE	61
Tabella 26. Errori assoluti di ogni derivazione di ogni dataset	68
Tabella 27. Errori totali di ogni dataset	68
Tabella 28. Valori di accuracy, precision e recall per le diverse derivazioni per il dataset BS	69
Tabella 29. Valori di accuracy, precision e recall per le diverse derivazioni per il dataset AJM	70
.....	70

Tabella 30. Valori di accuracy, precision e recall per le diverse derivazioni per il dataset FINE	71
Tabella 31. Valori totali di accuracy, precision e recall per i dataset BS, AJM e FINE	72
Tabella 32. Valori medi aree sottese di ogni derivazione di ogni dataset	73
Tabella 33. Valori aree sottese totali di ogni dataset.....	73