

ECG CNN Diagnostic System Report

Infante Martina, Lacatena Vito Simone

July 17, 2020

1 Introduzione

L'invecchiamento della popolazione sta comportando un incremento dei pazienti che soffrono di patologie cardiache e che quindi necessitano di un monitoraggio elettrocardiografico.

Già oggi la refertazione degli elettrocardiogrammi normali comporta un notevole impiego di tempo da parte degli specialisti cardiologi, la cui preparazione è oltremodo lunga e costosa. Sarebbe quindi importante poter escludere il maggior numero possibile di ecg che sicuramente non presentano anomalie dalla necessità che siano refertati dallo specialista evitando però nel modo più assoluto che la presenza di una alterazione elettrocardiografica, che potrebbe essere la spia di una patologia cardiologica, non sia riconosciuta e sfugga all'osservazione del cardiologo.

Allo stesso tempo si vanno diffondendo numerosi device come Kardia(www.quiver.store) e dheart(<https://www.d-heartcare.com/>) che permettono ai cittadini di registrare in modo autonomo il proprio ecg. E' prevedibile che il numero di tracciati registrati nei prossimi anni cresca in modo esponenziale e non è assolutamente pensabile che tutte le registrazioni prodotte potranno essere valutate da un cardiologo.

Devono quindi essere sviluppati dei sistemi che possano in modo automatico selezionare quali sono i tracciati che necessitano di valutazione specialistica e possibilmente anche l'urgenza di tale valutazione.

Per questi motivi è stata proposta un'architettura di rete neurale convoluzionata (CNN) per lo sviluppo di un sistema automatico in grado di riconoscere la presenza o meno di un'anomalia all'interno di un elettrocardiogramma (ECG). La CNN quindi effettuerà una classificazione binaria degli ECG, Normale (N) o con Anomalie (A).

2 Dati utilizzati

E' stato utilizzato il dataset MIT-BIH Arrhythmia Database da PhysioNet e da Kaggle. Il database delle aritmie del MIT-BIH contiene 48 estratti di mezz'ora

di registrazioni di ECG ambulatoriali a due canali, ottenuti da 47 soggetti studiati dal Laboratorio di Aritmia della BIH tra il 1975 e il 1979. 23 registrazioni sono state scelte a caso da un set di 4000 registrazioni di ECG ambulatoriali a 24 ore raccolte da una popolazione mista di pazienti ricoverati (circa il 60 Boston; le restanti 25 registrazioni sono state selezionate dallo stesso set per includere aritmie meno comuni ma clinicamente significative che non sarebbero ben rappresentate in un piccolo campione casuale. Le registrazioni sono state digitalizzate a 360 campioni al secondo per canale con risoluzione a 11 bit su una gamma di 10 mV. Due o più cardiologi hanno annotato indipendentemente ogni registrazione; i disaccordi sono stati risolti per ottenere le annotazioni di riferimento leggibili dal computer per ogni battito (circa 110.000 annotazioni in tutto) incluse nel database. Kaggle contiene i vari campioni dei tracciati ECG del database MIT-BIH Arrhythmia Database di PhysioNet, strutturati in csv e le annotazioni dei picchi registrati nei tracciati in file di testo. Le annotazioni relative ai picchi sono le seguenti:

Symbol	
· or N	Normal beat
L	Left bundle branch block beat
R	Right bundle branch block beat
A	Atrial premature beat
a	Aberrated atrial premature beat
J	Nodal (junctional) premature beat
S	Supraventricular premature beat
V	Premature ventricular contraction
F	Fusion of ventricular and normal beat
[Start of ventricular flutter/fibrillation
!	Ventricular flutter wave
]	End of ventricular flutter/fibrillation
e	Atrial escape beat
j	Nodal (junctional) escape beat
E	Ventricular escape beat
/	Paced beat
f	Fusion of paced and normal beat
x	Non-conducted P-wave (blocked APB)
Q	Unclassifiable beat
	Isolated QRS-like artifact

Figure 1: Tabella dei simboli

Per ulteriori informazioni sul dataset utilizzato si possono consultare i seguenti link: <https://archive.physionet.org/physiobank/database/html/mitdbdir/intro.htm>, <https://archive.physionet.org/physiobank/annotations.shtml>

2.1 Preprocessing del Dataset

Dati i file presenti in kaggle, sono state effettuate le seguenti procedure: È stato modificato il formato dei files di annotazione eliminando dati superflui in modo da poter essere processati come un file csv.

I tracciati ECG da 30 minuti sono stati frammentati in segmenti da 30 secondi ciascuno, quindi per ogni segmento si hanno 10800 campioni, si sono memorizzati solo i valori della derivazione MLII per ogni campione del segmento, per questo motivo sono stati scartati i file 102 e 104 in quanto presentavano

Time	Sample #	Type	Sub Chan	Num	Aux	Time,Sample,Y
0:00.019	7	+	0	0	0	(N
0:00.231	83	N	0	0	0	0:00.050,18,N
0:01.100	396	N	0	0	0	0:00.214,77,N
0:01.975	711	N	0	0	0	0:01.028,370,N
0:02.867	1032	N	0	0	0	0:01.839,662,N
0:03.800	1368	N	0	0	0	0:02.628,946,N
0:04.756	1712	N	0	0	0	0:03.419,1231,N
0:05.656	2036	N	0	0	0	0:04.208,1515,N
0:06.525	2349	N	0	0	0	0:05.025,1809,N

(a) prima

(b) dopo

Figure 2: files di annotazione

57 solamente le derivazioni V1 e V5.

58 In base alle annotazioni assegnate ai vari picchi presenti, ogni segmento è stato
 59 etichettato nel modo seguente:

- 60 • Se nel segmento tutti i picchi sono annotati con N (normale) l'intero seg-
 61 mento verrà etichettato con la label N
- 62 • Se invece ci sono dei picchi etichettati con un simboli diversi da N che
 63 rappresentano vari tipi di anomalie, il segmento verrà etichettato con A
 64 (Anomalia)

65 2.2 Proporzione tra le due classi N ed A

66 Per ogni tracciato di 30 minuti è stato creato un file csv dove ogni riga rappre-
 67 senta un segmento etichettato con N o con A. Per avere un unico csv rappresen-
 68 tate tutto il dataset, tutti i csv dei tracciati prima descritti sono stati uniti in
 69 un unico csv. Il dataset risultante presenta 871 esempi di segmenti etichettati
 70 come Normali (N) e 1769 etichettati come Anomali (A).

71 3 Modello

72 3.1 Struttura di base della CNN

73 Il modello utilizzato per la classificazione binaria (tracciato Normale o tracciato
 74 con Anomalie) di ECG è una rete convoluzionata sequenziale con 5 layer , con
 75 la seguente struttura: I primi 4 layer sono composti da:

- 76 • Conv1D
- 77 • BathcNorm1D
- 78 • MaxPool1D
- 79 • Funzione di attivazione RELU

80 L'ultimo layer è composto:

- 81 • AveragePooling1D
- 82 • Layer Flatten
- 83 • Layer Dense con funzione di attivazione Softmax

84 La figura seguente mostra un estratto della struttura della rete neurale im-
85 plementata in Python utilizzando la libreria Keras.

```
'''layer 1'''
cnn = Sequential()
cnn.add(Conv1D(2**num_unit, kernel_size=initial_kernel_size, strides=strides, input_shape=(10800,1)))
cnn.add(BatchNormalization())
cnn.add(MaxPooling1D(pool_size=maxpooling_poolsize))
cnn.add(Activation(activation))

'''layer 2'''
cnn.add(Conv1D(2**num_unit, kernel_size=kernel_size))
cnn.add(BatchNormalization())
cnn.add(MaxPooling1D(pool_size=maxpooling_poolsize))
cnn.add(Activation(activation))

'''layer 3'''
cnn.add(Conv1D(2**(num_unit+1), kernel_size=kernel_size))
cnn.add(BatchNormalization())
cnn.add(MaxPooling1D(pool_size=maxpooling_poolsize))
cnn.add(Activation(activation))

'''layer 4'''
cnn.add(Conv1D(2**(num_unit+2), kernel_size=kernel_size))
cnn.add(BatchNormalization())
cnn.add(MaxPooling1D(pool_size=maxpooling_poolsize))
cnn.add(Activation(activation))

'''layer 5'''
cnn.add(AveragePooling1D(pool_size = avg_poolsize))
cnn.add(Flatten())
cnn.add(Dense(n_classes, kernel_initializer=kernel_initializer, activation='softmax'))
return cnn
```

Figure 3: Implementazione della CNN in Python

86 Tale modello è stato realizzato ispirandosi alla rete neurale proposta dall'articolo
87 Automatic ECG Diagnosis Using Convolutional Neural Network

88
89

90 3.2 Descrizione della struttura

91 I layer che effettuano la Conv1D (convoluzione ad una dimensione) creano un
92 kernel di convoluzione che viene convoluto con l'input del layer su una singola
93 dimensione per produrre un tensore di output.

94 Al layer iniziale di Conv1D viene fornito un argomento inputshape relativo alla
95 dimensione del segmento di input.

96 La Batch Norm 1D (cioè la batch normalisation ad 1 dimensione) riduce la
97 covarianza dei valori, normalizza i valori di output del livello precedente sot-
98 traendo la media e dividendo per la deviazione standard.

99 Il MaxPool 1D (Max Pooling) ridimensiona la rappresentazione degli input pren-
100 dendo il valore massimo sulla finestra definita da poolsize. La finestra viene
101 spostata di tanti passi quanti definiti da strides. AvgPooling effettua la stessa
102 operazione di MaxPool però prende il valore medio della finestra. Il layer Flatten
103 appiattisce l' input per poi passarlo ad un layer Dense. Il layer Dense imple-
104 menta l'operazione:

105 $\text{output} = \text{activation}(\text{dot}(\text{input}, \text{kernel}) + \text{bias})$ dove l'attivazione è la funzione
106 di attivazione dell'elemento passata come argomento di attivazione, il kernel è
107 una matrice di pesi creata dal layer, e il bias è un vettore di bias creato dal
108 layer.

109 4 Validazione e Testing

110 4.1 Validazione con LearningSet

111 L' intero dataset è stato diviso nel seguente modo: 70% di Learning Set e 30%
112 di Test Set



Figure 4: Divisione del Dataset

113 Le proporzioni delle A e delle N sono state mantenute sia nel LearningSet
114 sia nel TestingSet, così come erano del Dataset completo, quindi con numero
115 delle A doppio rispetto alle N.

116 Per la validazione degli iperparametri del modello è stata utilizzata la con-
117 validazione incrociata K-Fold con $k = 10$ sul LearningSet. A ogni iterazione del k-Fold
118 si sono effettuate le seguenti operazioni:

- 119 • Creazione della CNN
- 120 • Suddivisione del LearningSet in k parti dove k-1 parti sono state utilizzate
121 per l' addestramento (TrainingSet) e la restante parte come ValidationSet

122

- Calcolo dell'accuratezza e il recall

123

Alla fine delle varie iterazioni si è calcolata la media e la deviazione standard

124

delle accuratze e del recall.

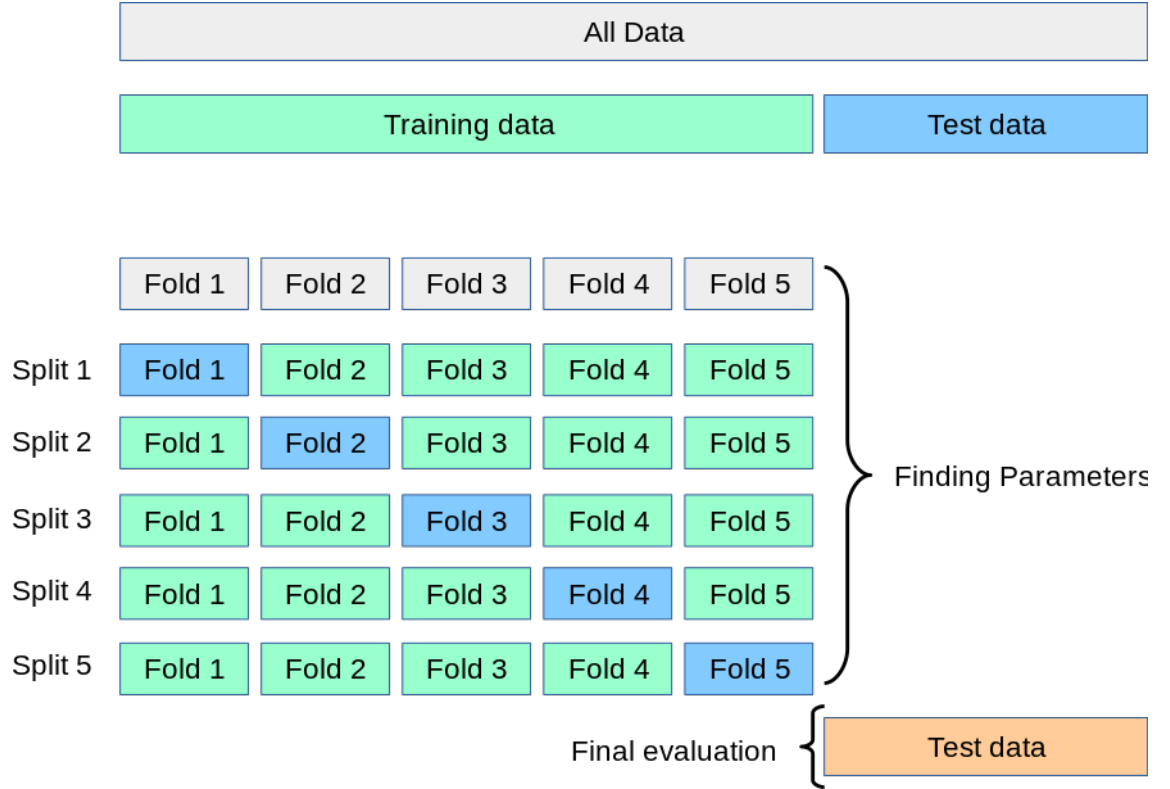


Figure 5: Esempio di k fold con k = 5

125

4.2 Predizione con TestSet

126

Dopo aver effettuato la validazione è stata effettuata la predizione. Per la predizione la CNN è stata addestrata sull' intero LearningSet e successivamente si è effettuata la predizione per i dati del TestSet utilizzando il modello appena creato. Infine si è visualizzata la matrice di confusione e le seguenti metriche:

129

130

131

$$Precision = \frac{TP}{(TP + FP)} \quad (1)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (2)$$

$$F1 = \frac{2TP}{(2TP + FP + FN)} \quad (3)$$

$$Accuracy = \frac{TP + TN}{(TP + TN + FN + FP)} \quad (4)$$

132 5 Iperparametri

133 5.1 Iperparametri iniziali

134 Per la scelta degli iperparametri iniziali si è fatto riferimento all' articolo "Au-
 135 tomatic ECG Diagnosis Using Convolutional Neural Network di Roberta Avan-
 136 zato and Francesco Beritelli ",gli iperparametri scelti inizialmente sono stati i
 137 seguenti:

```
#Parametres
#-----
EPOCHS = 100
BATCH_SIZE = 24
MAXPOOL_POOL_SIZE = 4
AVGPOOL_POOL_SIZE = 2
KERNEL_SIZE = 4
OPTIMIZER = tensorflow.keras.optimizers.Adam
ACTIVATION = 'relu'
KERNEL_INITIALIZER = 'normal'
STRIDES = 4
NUM_UNIT = 7 # 2^NUM_UNIT 128 unit
LOSS_FUNCTION = 'binary_crossentropy'
INITIAL_KERNEL_SIZE = 80
LEARNING_RATE = 0.1

#-----

def CNN_LOAD():
    model = createCNN(num_unit = NUM_UNIT,
                      initial_kernel_size = INITIAL_KERNEL_SIZE,
                      strides = STRIDES ,
                      kernel_initializer = KERNEL_INITIALIZER,
                      maxpooling_poolsize = MAXPOOL_POOL_SIZE ,
                      activation= ACTIVATION,
                      avg_poolsize = AVGPOOL_POOL_SIZE,
                      kernel_size = KERNEL_SIZE)
    model.compile(optimizer=OPTIMIZER(lr = LEARNING_RATE), loss=LOSS_FUNCTION, m
    return model
```

Figure 6: Iperparametri iniziali

Con questi parametri si sono ottenuti i seguenti risultati
 Nella validazione:

value	accuracy	loss	recall
mean	0.869	0.732	0.971
variance	0.02	0.124	0.002

Table 1: Validation Results

Nel Test:

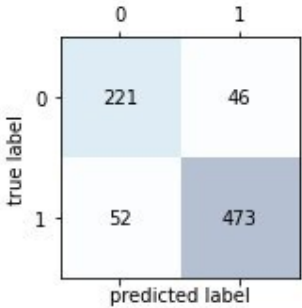


Figure 7: Confusion Matrix

	precision	recall	f1-score	support
N	0.81	0.83	0.82	267
A	0.91	0.90	0.91	525
accuracy			0.88	792
macro avg	0.86	0.86	0.86	792
weighted avg	0.88	0.88	0.88	792

Figure 8: Test Report

5.2 Test con altri iperparametri

Per migliorare i risultati ottenuti sono stati effettuati una serie di test cambiando gli iperparametri della CNN, si è scelto di modificare i parametri riguardanti gli argomenti presi in input da ogni layer senza cambiare la struttura di base della rete neurale, i parametri modificati nei vari test includono la funzione di attivazione, la funzione di ottimizzazione, le dimensioni delle finestre di pooling (poolsize), le dimensioni dei kernel di convoluzione, il numero di epoche e la dimensione di ogni batch, le dimensioni degli output dei layer di convoluzione,

l'inizializzazione dei pesi.
I risultati considerati più rilevanti sono riassunti nella seguente tabella.

Testing Result				Validation Result	
Test	Recall A	Recall N	Accuracy	avg_accuracy	avg_recall
1	0,99	0,40	0,80	0,834 +/- 0,029	0,956 +/- 0,002
2	0,90	0,83	0,88	0,889 +/- 0,023	0,968 +/- 0,002
3	0,91	0,72	0,85	0,841 +/- 0,036	0,961 +/- 0,002
4	0,94	0,77	0,89	0,861 +/- 0,045	0,964 +/- 0,002
5	0,95	0,77	0,88	0,754 +/- 0,096	0,901 +/- 0,003
6	0,95	0,85	0,92	0,765 +/- 0,072	0,891 +/- 0,003
7	0,98	0,51	0,83	0,845 +/- 0,038	0,925 +/- 0,003
8	0,91	0,78	0,87	0,856 +/- 0,026	0,967 +/- 0,003

Figure 9: Test con altri parametri

I test evidenziati sono quelli considerati migliori poiché hanno un recall delle A più alto. Avere un recall delle A alto significa che la probabilità che un ECG che presenta anomalie venga classificato come normale è molto bassa. Si è data meno importanza al recall delle N poiché se un ECG normale viene classificato come anomalo è un errore meno grave, poiché si presuppone che gli ECG classificati come anomali vengano successivamente analizzati da un cardiologo.

6 Esperimenti con segmenti di 15 secondi

Dividendo i vari tracciati in segmenti di 30 secondi si aveva un eccessivo sbilanciamento del dataset poiché il numero dei segmenti etichettati come anormali era circa il doppio di quelli etichettati come normali, di conseguenza si è deciso di suddividere i tracciati non più in segmenti di 30 secondi bensì in segmenti di 15 secondi.

In tutto si sono ottenuti i 5280 segmenti di cui:

- 2105 segmenti etichettati come N (40% del dataset)
- 3175 segmenti etichettati come A (60% del dataset)

Anche in questo caso l'intero dataset è stato suddiviso in una parte di learning e una parte di test con una percentuale di 70 % per il learning e 30% per il test. Sia nel learning set che nel test set sono state mantenute le proporzioni iniziali tra segmenti etichettati come A e segmenti etichettati come N.

Eseguito il test con i gli iperparametri di partenza (seguendo l'articolo precedentemente citato) si sono ottenuti i seguenti risultati:

Testing Result			Validation Result	
Recall A	Recall N	Accuracy	avg_accuracy	avg_recall
0.91	0.89	0.90	0.880 +/- 0.063	0.970 +/- 0.002

Figure 10: Test con parametri iniziali

175 Sono stati effettuati altri test cambiando gli iperparametri e i risultati sono
176 presentati nella seguente tabella.

	Testing Result			Validation Result	
Test	Recall A	Recall N	Accuracy	avg_accuracy	avg_recall
1	0,88	0,86	0,88	0,784 +/- 0,103	0,927 +/- 0,002
2	0,90	0,88	0,89	0,892 +/- 0,022	0,971 +/- 0,004
6	1.00	0,30	0,72	0,788 +/- 0,076	0,868 +/- 0,002
7	0,99	0,38	0,75	0,728 +/- 0,065	0,901 +/- 0,003

Figure 11: Test con altri parametri

177 7 Ulteriori esperimenti modificando le proporzioni 178 di A ed N nel TestSet

179 Si è provato ad eliminare dal TestSet alcuni segmenti etichettati come A per
180 ottenere una proporzione del 60% di segmenti etichettati come N e il 40% di
181 segmenti etichettati come A, questo per rendere più realistica la fase di Test,
182 poiché precedentemente il test presentava molti più segmenti anomali che nor-
183 mali. Eseguendo nuovamente il test con gli iperparametri di partenza (seguendo
184 l' articolo) si sono ottenuti i seguenti risultati:

Testing Result			Validation Result	
Recall A	Recall N	Accuracy	avg_accuracy	avg_recall
0,90	0,92	0,91	0,909 +/- 0,010	0,969 +/- 0,003

Figure 12: Test con parametri iniziali

185 Si sono effettuati successivamente altri test cambiando gli iperparametri e i
186 risultati ottenuti sono presentati nella seguente tabella:

	Testing Result			Validation Result	
Test	Recall A	Recall N	Accuracy	avg_accuracy	avg_recall
1	0,75	0,94	0,87	0,781 +/- 0,102	0,907 +/- 0,002
4	0,91	0,85	0,88	0,887 +/- 0,021	0,967 +/- 0,001
6	0,80	0,96	0,89	0,821 +/- 0,037	0,866 +/- 0,004
7	0,83	0,88	0,86	0,851 +/- 0,019	0,899 +/- 0,003
9	0,97	0,63	0,76	0,847 +/- 0,045	0,926 +/- 0,005
10	1.00	0,51	0,70	0,858 +/- 0,044	0,926 +/- 0,003
12	0,90	0,91	0,91	0,864 +/- 0,034	0,942 +/- 0,003
13	0,90	0,80	0,94	0,751 +/- 0,083	0,900 +/- 0,004

Figure 13: Test con altri parametri

I test che hanno ottenuto migliori risultati sono il test 9 e il test 10 poiché presentano un recall delle A più alto. Analizzando i tracciati classificati male dai due modelli si è considerato come ottimale il test 10, in quanto i soli due errori sono relativi a due tracciati che contengono un'unica extrasistole che tra l'altro si trova una volta in ultima posizione ed una volta in penultima. Nel test 9, invece oltre ad esserci un maggior numero di tracciati che presentano anomalie, ce ne sono alcuni che ne presentano più di una.

7.1 Iparametri del Test 10 e risultati ottenuti

Per la creazione del modello del test numero 10 sono stati scelti i seguenti iperparametri:

- Numero di epoche 50
- Batch size 16
- Pool size 4 per i layer max pooling 1D
- Pool size di 2 per il Layer average pooling 1D
- Kernel size per il primo layer di convoluzione è 35 con strides di 6, per i successivi layer di convoluzione il kernel size è di 4 con striders di 1
- Funzione di attivazione tanh(tangente iperbolica)
- Funzione di ottimizzazione adamax
- Il layer Dense con funzione di attivazione “softmax” ,kernel initializer “normal” (inizializzazione dei pesi)

- la dimensione dello spazio di output del primo layer di convoluzione è di 64, i successivi tre layer di convoluzione sono rispettivamente 64, 86, 120

Con questi parametri si sono ottenuti i seguenti risultati:

Nella validazione:

Le seguenti immagini mostrano il progresso dei valori di accuratezza, loss e recall.

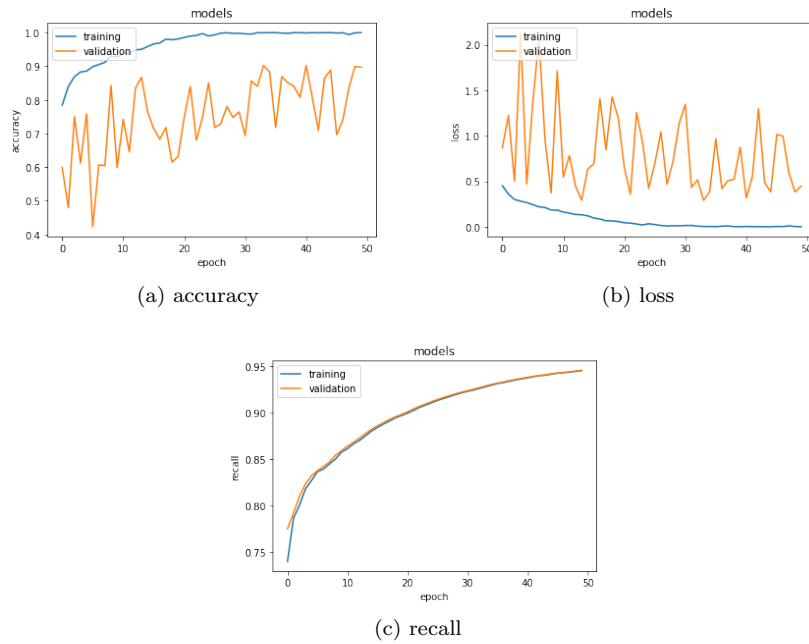


Figure 14: training e validation loss ,accuracy e recall

La tabella riporta i valori medi e la varianza dei valori ottenuti per ogni iterazione del kfold ad ogni iterazione

value	accuracy	loss	recall
mean	0.858	0.519	0.926
variance	0.044	0.192	0.003

Table 2: Validation Results

I valori di accuracy e loss nella validazione non sono molto elevati poiché si è data più importanza a massimizzare il recall, con l'obiettivo di ridurre al minimo il numero di ECG Anomali classificati come Normali.

Risultati ottenuti nella predizione finale con il TestSet:

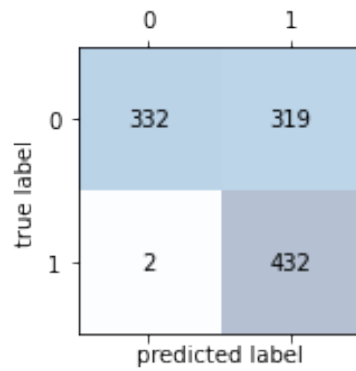


Figure 15: Confusion Matrix del Test 10

	precision	recall	f1-score	support
N	0.99	0.51	0.67	651
A	0.58	1.00	0.73	434
accuracy			0.70	1085
macro avg	0.78	0.75	0.70	1085
weighted avg	0.83	0.70	0.70	1085

Figure 16: Report del Test 10

219 Tracciati classificati in maniera errata dal modello del test 10:

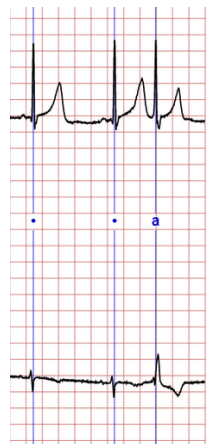


Figure 17: Un'unica exatrasistole atriale presente come ultima del tracciato



Figure 18: Un'unica extrasistole ventricolare presente come penultima del tracciato

8 Conclusioni

I risultati ottenuti sono estremamente soddisfacenti essendo stati classificati come normali soltanto 2 dei 434 ecg che presentavano anomalie e che in entrambi i casi l'anomalia è presente ad un estremo della registrazione. Disponendo di registrazioni della durata di 30 secondi le probabilità che sia presente una unica anomalia solo al termine della registrazione si dimezzerebbero; in ogni caso il mancato riconoscimento delle due anomalie presenti dei due casi non ha rilievo dal punto di vista clinico. L'ottenimento di tale risultato è stato "pagato" con un elevato numero di ecg normali classificati come anomali. Oltre la metà degli ecg normali sono stati classificati correttamente. Se l'algoritmo potesse essere applicato a tutti gli elettrocardiogrammi che ordinariamente vengono valutati presso i diversi ambulatori di cardiologia, già tale risultato permetterebbe un notevole risparmio.