

Progetto per l'insegnamento di Applicazioni dell'Intelligenza Artificiale

Corso di Laurea Magistrale in Informatica Applicata

Università degli studi di Urbino "Carlo Bo"

Sessione autunnale a.a. 2023/2024

Relatore:

Matteo Marco Montanari
(Matricola 323293)

Docenti:

Prof. Stefano Ferretti
Prof.ssa. Sara Montagna

Obiettivo del progetto

Si vuole progettare e addestrare una *rete neurale profonda* basata su Convolutional Neural Networks e Long Short Term Memory Networks per la *valutazione di qualità* di segnali ECG dinamici e di lunga durata. Consultando la letteratura di riferimento, l'obiettivo del progetto è replicare in buona parte il lavoro svolto da He, C. et Al. intitolato “*Dynamic Electrocardiogram Signal Quality Assessment Method Based on Convolutional Neural Network and Long Short-Term Memory Network*” [1] su un diverso dataset e utilizzando il framework *PyTorch* al posto di TensorFlow. Il dataset scelto per il progetto è “*Brno University of Technology ECG Quality Database (BUT QDB)*” [2] raccolto tramite la banca dati PhysioNet [3].

Dataset utilizzato

“**Brno University of Technology ECG Quality Database (BUT QDB)**” [2] è un database creato dal team di cardiologia del Dipartimento di Ingegneria Biomedica della Brno University of Technology, allo scopo di valutare la qualità di segnali ECG. I dati comprendono *18 registrazioni a lungo termine di segnali ECG a derivazione singola* e i dati associati di un accelerometro a 3 assi, raccolti da 15 soggetti mentre svolgevano normali attività quotidiane («*free living conditions*»). I dati sono stati raccolti utilizzando un ECG mobile e un accelerometro (Bittium Faros 180) con una frequenza di campionamento di 1.000 Hz per i segnali ECG e 100 Hz per i segnali dell'accelerometro.

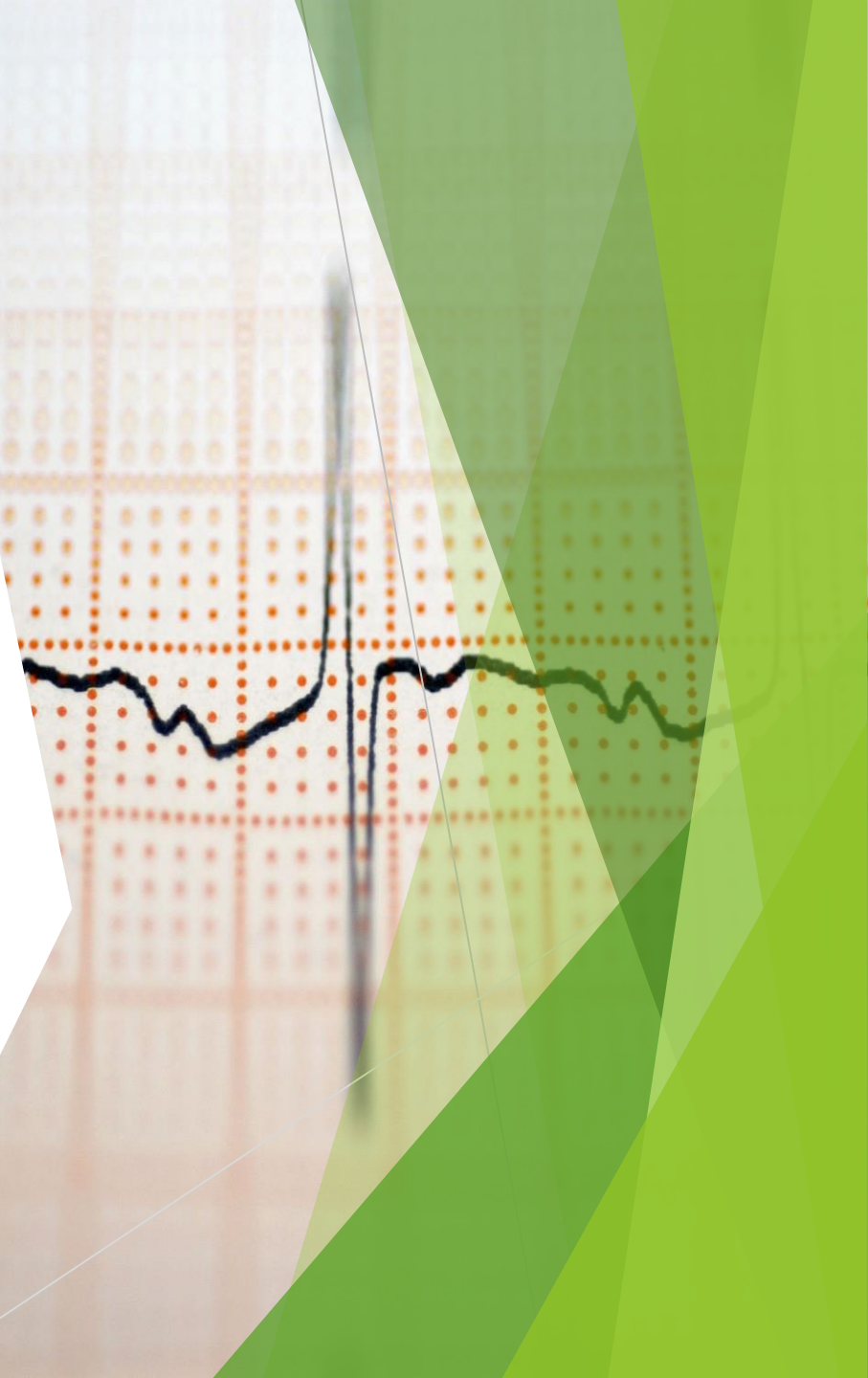
Per questo progetto, i dati degli **accelerometri** non sono stati utilizzati perché non rilevanti agli scopi previsti.

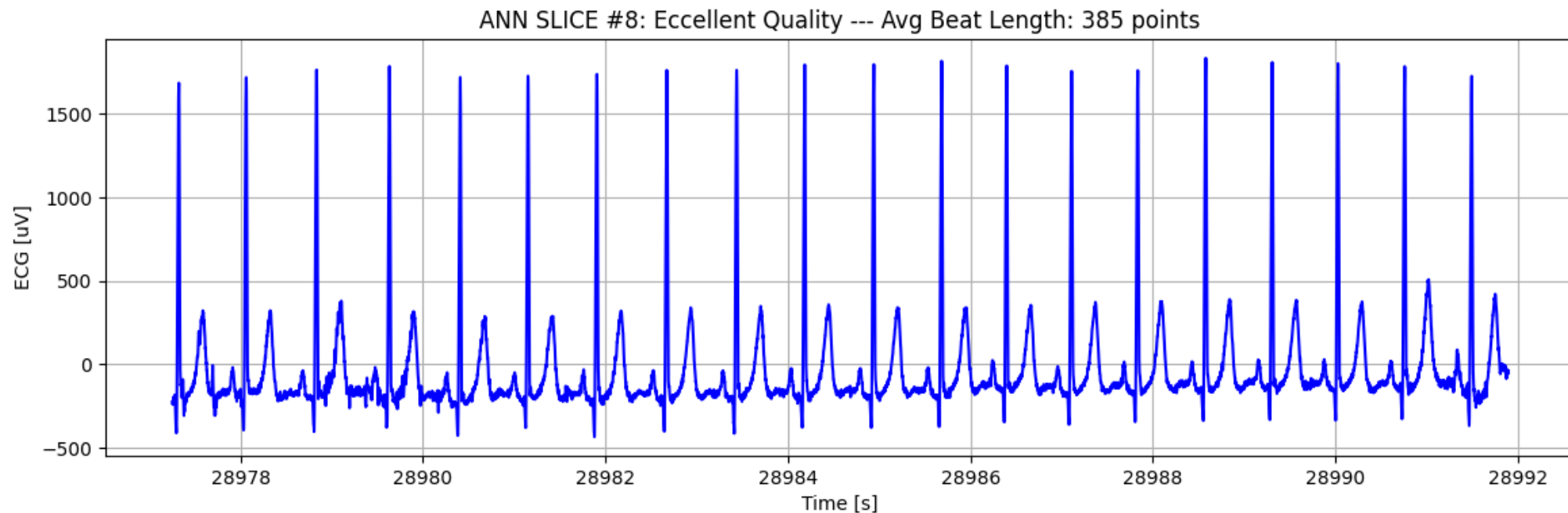
Tutti i dati sono forniti nel formato **WaveForm Database (WFDB)** e sono distribuiti liberamente sotto licenza Creative Commons Attribution (CC-BY-4.0) attraverso la banca dati **PhysioNet** [3].

Classi di qualità

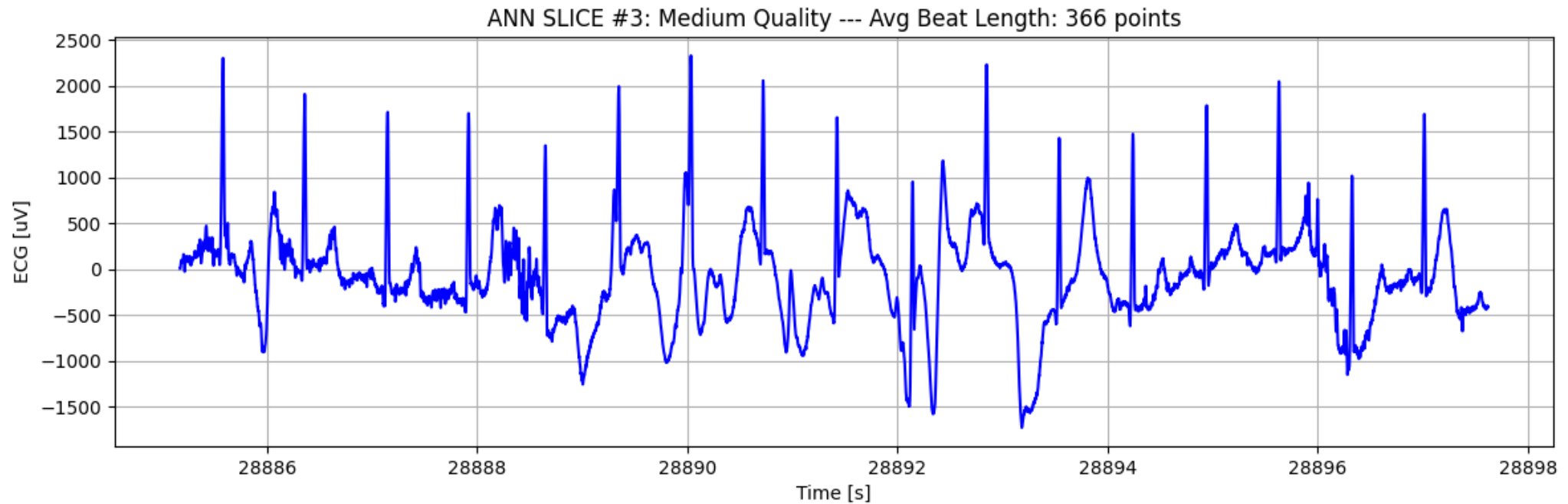
Tre segnali, dalla durata minima di 24 ore, sono stati **completamente annotati** in termini di qualità del segnale ECG. I restanti 15 segnali sono stati annotati in due segmenti selezionati, ciascuno della durata di 20 minuti. Inoltre, sono stati annotati anche cinque segmenti aggiuntivi di scarsa qualità del segnale. La **qualità del segnale ECG** è stata classificata come segue:

1. **Classe 1 (Excellent Quality)**: indica che tutte le forme d'onda significative dell'ECG (onda P, onda T e complesso QRS) sono chiaramente visibili e i loro inizi e offset possono essere rilevati in modo affidabile;
2. **Classe 2 (Medium Quality)**: indica che vi è un maggiore livello di rumore rispetto alla classe precedente e i punti significativi dell'ECG non possono essere individuati in modo affidabile, ma il segnale consente comunque un rilevamento QRS affidabile;
3. **Classe 3 (Bad Quality)**: indica che i complessi QRS non possono essere rilevati in modo affidabile e il segnale non è adatto per ulteriori analisi.

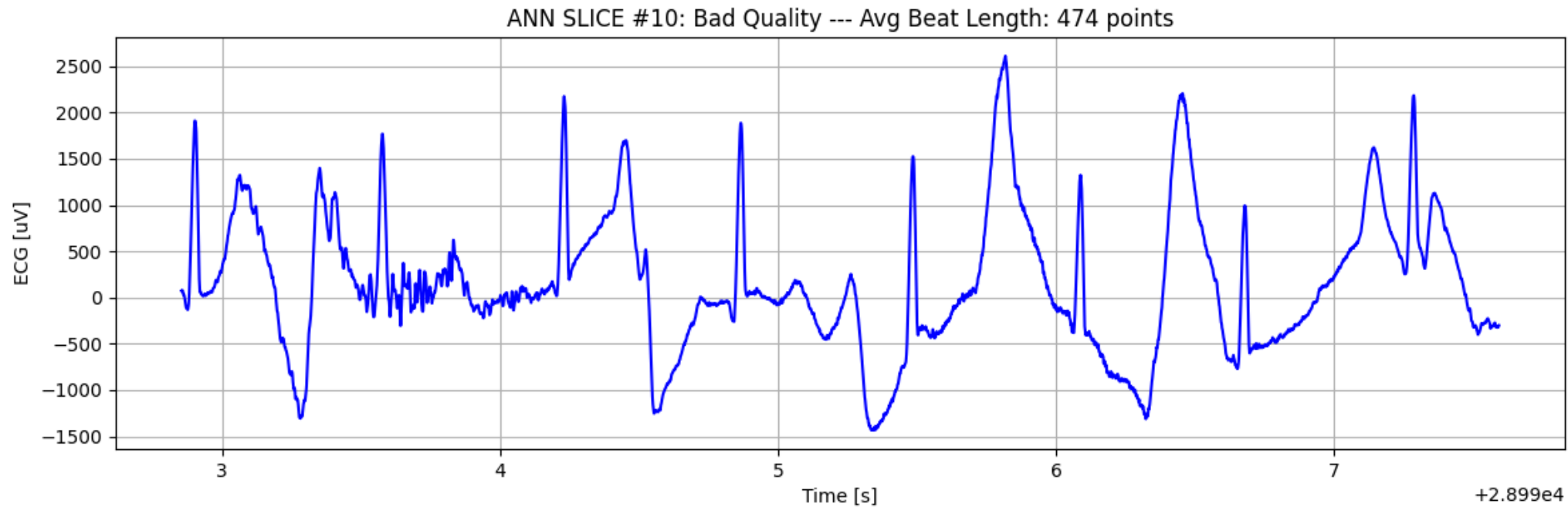




Classe 1 (Excellent Quality): indica che tutte le forme d'onda significative dell'ECG (onda P, onda T e complesso QRS) sono chiaramente visibili e i loro inizi e offset possono essere rilevati in modo affidabile;



Classe 2 (Medium Quality): indica che vi è un maggiore livello di rumore rispetto alla classe precedente e i punti significativi dell'ECG non possono essere individuati in modo affidabile, ma il segnale consente comunque un rilevamento QRS affidabile;



Classe 3 (Bad Quality): indica che i complessi QRS non possono essere rilevati in modo affidabile e il segnale non è adatto per ulteriori analisi.

An ECG signal waveform is displayed on the left side of the slide, showing multiple heartbeats. The waveform is black on a light blue background, with a vertical blue line indicating a specific point in time. The signal is periodic, showing the characteristic P, Q, R, and S waves of an ECG trace.

Annotazioni di qualità

Le annotazioni di qualità del segnale ECG sono state registrate in un file **CSV con 12 colonne**. Per ogni annotatore sono state assegnate tre colonne mentre le ultime tre colonne sono per il consenso (**3 colonne x 3 annotatori + 3 colonne x consenso**). La prima colonna per ogni annotatore contiene l'indice che identifica il primo campione del segmento annotato; la seconda colonna contiene l'indice che identifica il campione finale del segmento annotato; e la terza colonna contiene la **classe di qualità** assegnata (1, 2 o 3; 0 significa che la qualità non è stata annotata in questo segmento).

Per il progetto si è deciso di considerare solamente le annotazioni relative alle colonne di **consenso** (ignorando quelle dei singoli annotatori).

File: <base> / 104001 / 104001_ANN.csv (7,287 bytes)

| | | | | | | | | | | | |
|----------|----------|---|----------|----------|---|----------|----------|---|----------|----------|---|
| 1 | 28800000 | 0 | 1 | 28800000 | 0 | 1 | 28800000 | 0 | 1 | 28800000 | 0 |
| 28800001 | 29028285 | 1 | 28800001 | 29029015 | 1 | 28800001 | 29028996 | 1 | 28800001 | 29028996 | 1 |
| 29028286 | 29041643 | 2 | 29029016 | 29038208 | 2 | 29028997 | 29036404 | 2 | 29028997 | 29038208 | 2 |
| 29041644 | 29830120 | 1 | 29038209 | 29833496 | 1 | 29036405 | 29041070 | 1 | 29038209 | 29041070 | 1 |
| 29830121 | 29843829 | 2 | 29833497 | 29842799 | 2 | 29041071 | 29041373 | 2 | 29041071 | 29041373 | 2 |
| 29843830 | 30000000 | 1 | 29842800 | 30000000 | 1 | 29041374 | 29833350 | 1 | 29041374 | 29833350 | 1 |
| 30000001 | 57600000 | 0 | 30000001 | 57600000 | 0 | 29833351 | 29842799 | 2 | 29833351 | 29842799 | 2 |
| 57600001 | 57702386 | 2 | 57600001 | 57614269 | 1 | 29842800 | 29847171 | 1 | 29842800 | 30000000 | 1 |
| 57702387 | 57704739 | 3 | 57614270 | 57614654 | 2 | 29847172 | 29847839 | 2 | 30000001 | 57600000 | 0 |
| 57704740 | 58265306 | 2 | 57614655 | 57617910 | 1 | 29847840 | 30000000 | 1 | 57600001 | 57614172 | 1 |
| 58265307 | 58268218 | 3 | 57617911 | 57618319 | 2 | 30000001 | 57600000 | 0 | 57614173 | 57614654 | 2 |
| 58268219 | 58800000 | 2 | 57618320 | 57620535 | 1 | 57600001 | 57614172 | 1 | 57614655 | 57617910 | 1 |
| 58800001 | 87178000 | 0 | 57620536 | 57621106 | 2 | 57614173 | 57614637 | 2 | 57617911 | 57618319 | 2 |

Stato dell'arte

Open Access

Article

Dynamic Electrocardiogram Signal Quality Assessment Method Based on Convolutional Neural Network and Long Short-Term Memory Network

by **Chen He**, **Yuxuan Wei**, **Yeru Wei**, **Qiang Liu**  and **Xiang An** ^{*} 

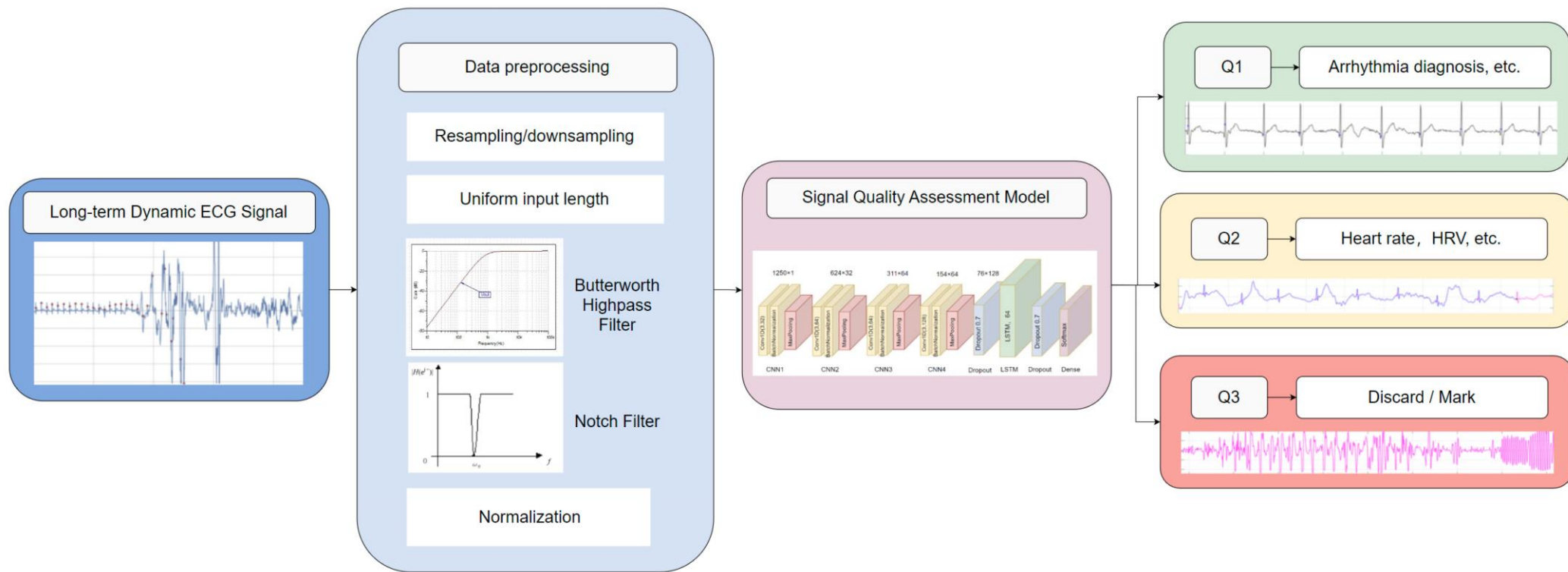
Academy of Artificial Intelligence, Beijing Institute of Petrochemical Technology, Beijing 102617, China

^{*} Author to whom correspondence should be addressed.

Big Data Cogn. Comput. **2024**, *8*(6), 57; <https://doi.org/10.3390/bdcc8060057>

Submission received: 18 March 2024 / Revised: 24 May 2024 / Accepted: 24 May 2024 / Published: 31 May 2024

Data Pipeline

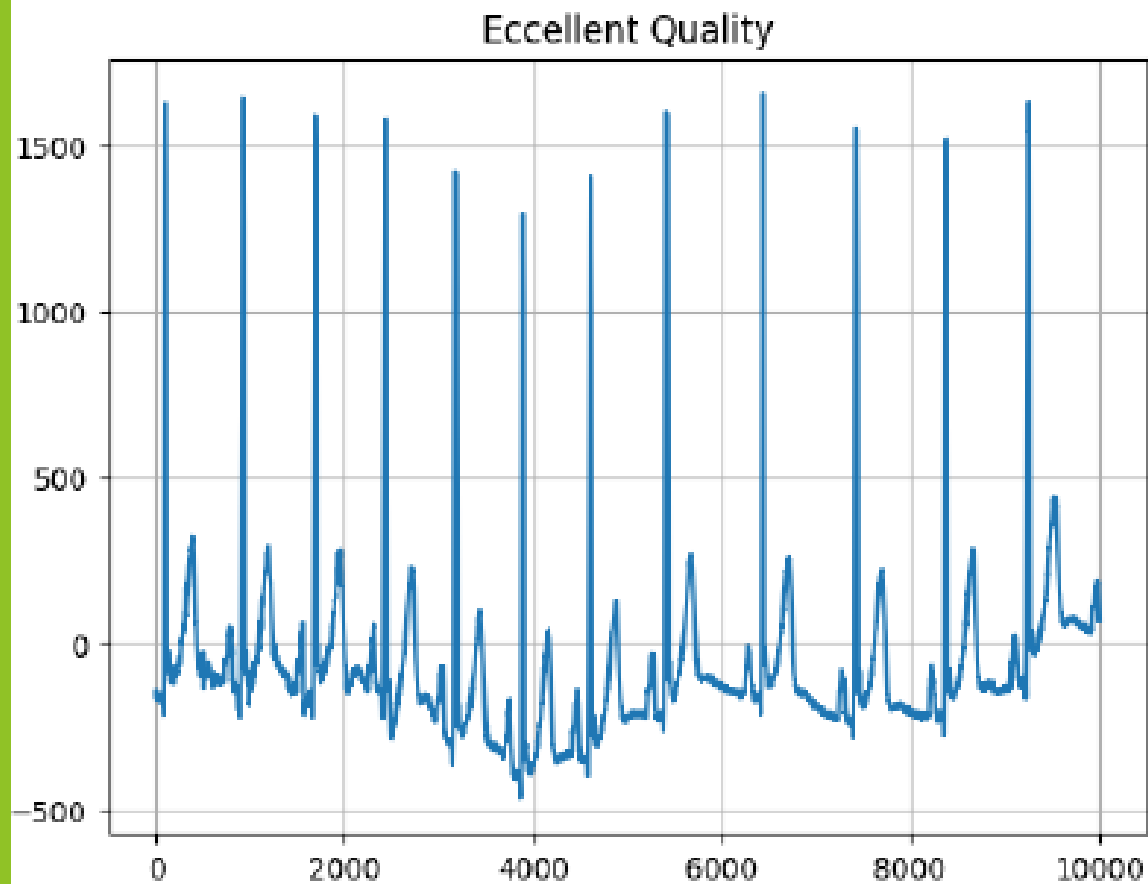


Preprocessing

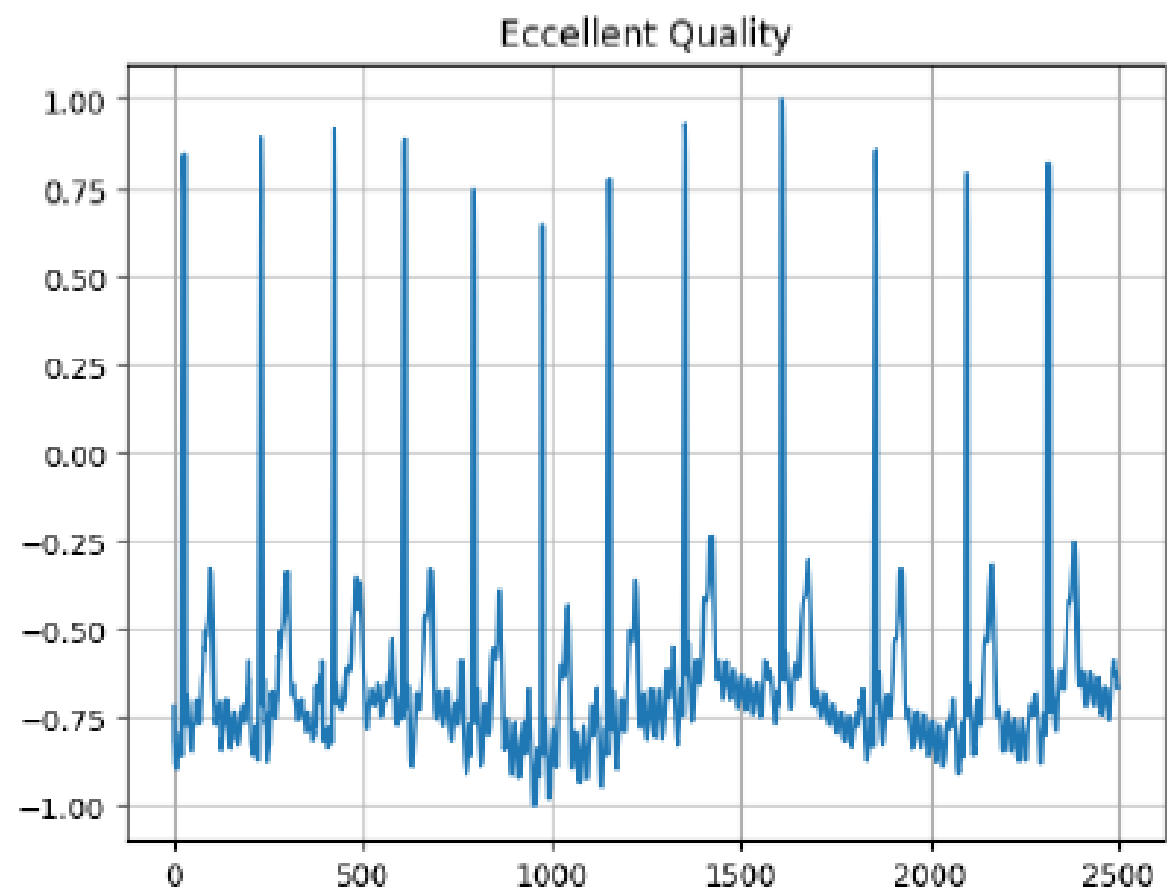
1. Un segnale ECG dinamico di lunga durata viene prima suddiviso in segmenti utilizzando una *finestra scorrevole di 10 secondi (senza sovrapposizioni)* per garantire che i dati in ingresso al modello siano di dimensione costante.
2. I segmenti così ottenuti vengono poi *sottocampionati* per ridurre la frequenza di campionamento di tutti i segnali a un valore costante di 250 Hz (rispetto all'originale di 1000 Hz). In questo modo i dati potranno essere elaborati in maniera più efficiente.
3. In seguito, tali dati vengono filtrati mediante un *filtro passa-alto di tipo Butterworth da 0,67 Hz* e successivamente passano attraverso un ulteriore *filtro Notch da 50 Hz* per ridurre le interferenze più facilmente rimovibili generate durante la misurazione.
4. I segnali filtrati vengono quindi normalizzati nell'intervallo $[-1,1]$ attraverso una *normalizzazione min-max*:

$$x = \frac{x - \min}{\max - \min}$$

5. Infine, i dati vengono suddivisi randomicamente in *dati di addestramento, test e validazione*, rispettivamente nel rapporto: 70% - 20% - 10%.

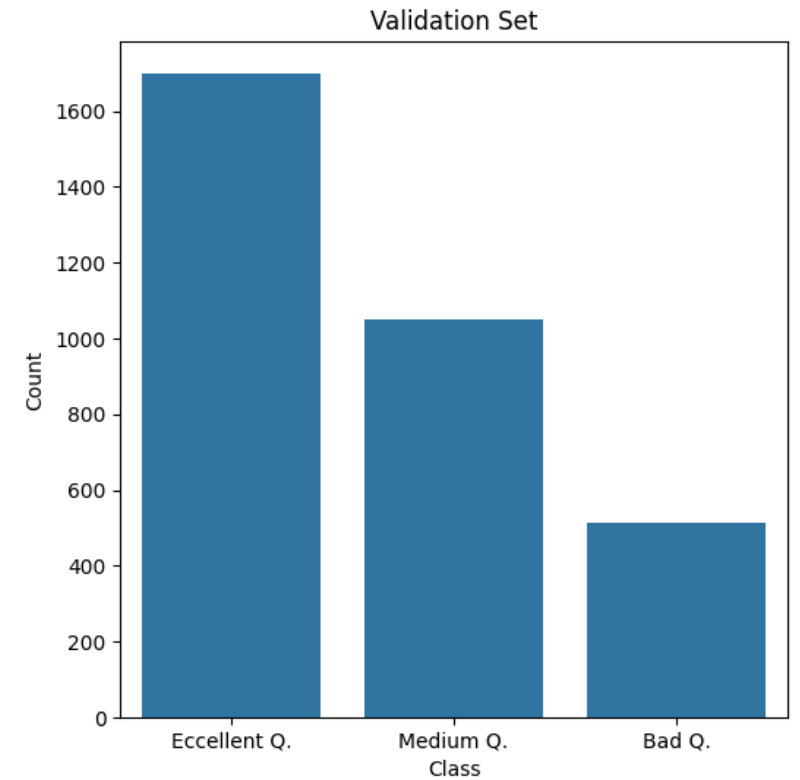
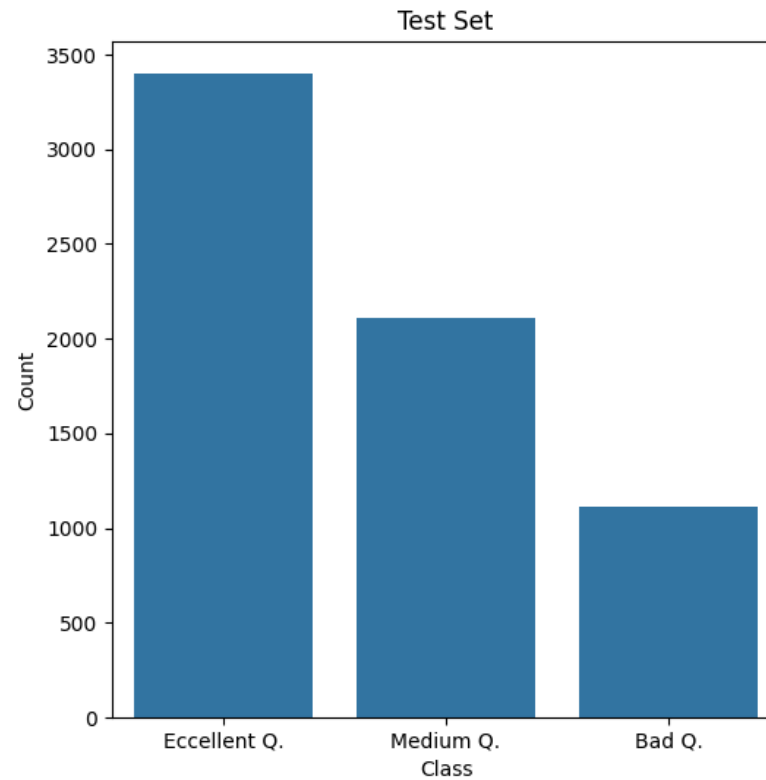
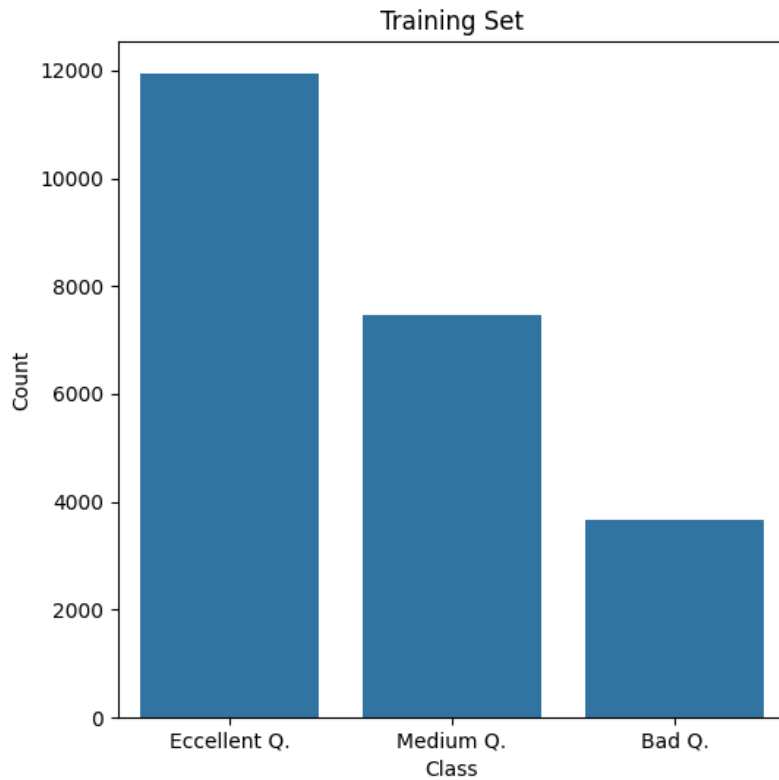


Prima del preprocessing



Dopo il preprocessing

Data distribution

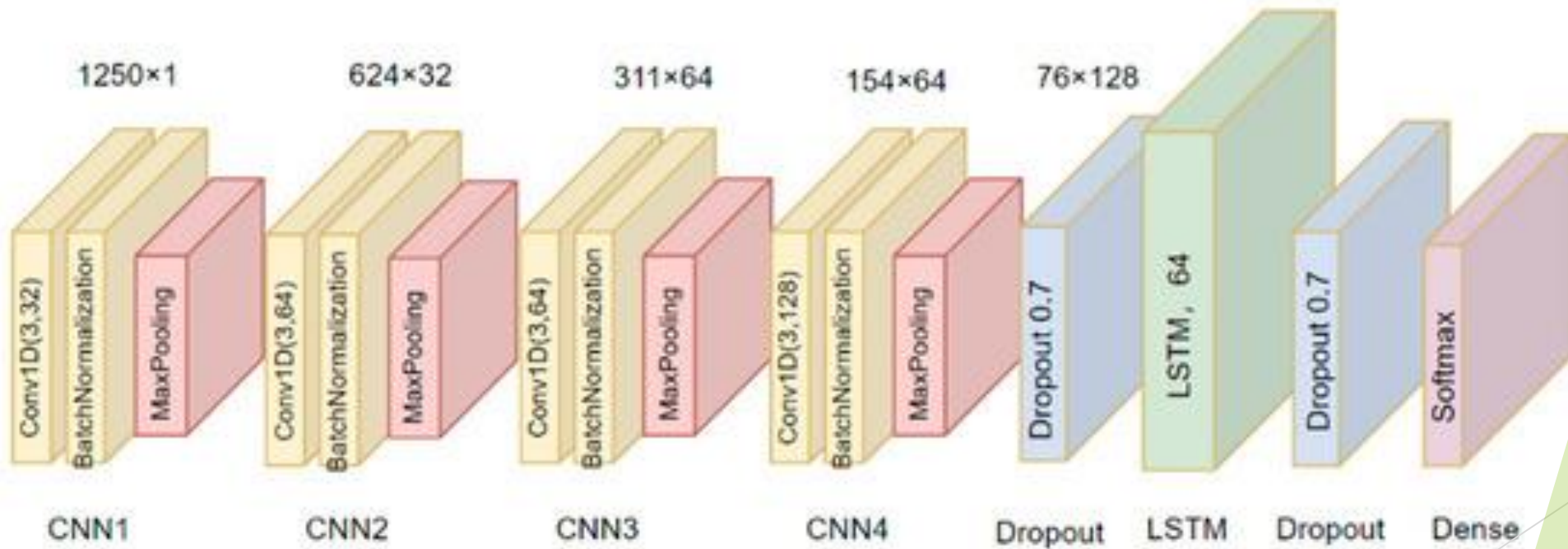


Model

Seguendo il lavoro proposto nell'articolo, si è scelto di addestrare una **rete neurale profonda** composta da strati convoluzionali (CNN) e LSTM. I **layer CNN** sono stati utilizzati per automatizzare il processo di **Feature Extraction** a partire dai dati grezzi, senza doverlo effettuare manualmente ma facendo apprendere alla rete i parametri corretti tramite gli strati convoluzionali. Il **layer LSTM** è stato utilizzato con lo scopo di catturare non solo le variazioni individuabili tra battito e battito, ma anche le **fluttuazioni** della frequenza cardiaca a **lungo termine** rilevate sullo stesso segnale a diversi istanti di tempo.

Per ridurre **l'overfitting** si utilizzano due strati di **dropout** con probabilità 0.7: uno prima e uno dopo il layer LSTM. Per l'addestramento si utilizzano l'ottimizzatore **Adam** e la funzione **Cross Entropy** come funzione di perdita. Infine, per migliorare l'efficienza di training e la capacità di generalizzazione del modello, si è scelto di utilizzare la strategia **Early Exit**: quando le prestazioni sul **validation set** non mostrano alcun miglioramento per più epoche consecutive, il training viene terminato in anticipo per evitare **l'overfitting** sui dati di addestramento.

Model Architecture



```

class CNN_LSTM(nn.Module):
    def __init__(self, lstm_hidden_size, lstm_num_layers, num_classes):
        super(CNN_LSTM, self).__init__()

        self.lstm_hidden_size = lstm_hidden_size
        self.lstm_num_layers = lstm_num_layers
        self.num_classes = num_classes
        # length of a single sample
        self.ecg_sample_len = int(df_train.shape[1] - 1) # 2500

        # Definizione dei layer CNN
        self.conv1 = nn.Conv1d(in_channels=1, out_channels=32, kernel_size=3)
        self.bn1 = nn.BatchNorm1d(32)
        self.pool1 = nn.MaxPool1d(kernel_size=2)

        self.conv2 = nn.Conv1d(in_channels=32, out_channels=64, kernel_size=3)
        self.bn2 = nn.BatchNorm1d(64)
        self.pool2 = nn.MaxPool1d(kernel_size=2)

        self.conv3 = nn.Conv1d(in_channels=64, out_channels=64, kernel_size=3)
        self.bn3 = nn.BatchNorm1d(64)
        self.pool3 = nn.MaxPool1d(kernel_size=2)

        self.conv4 = nn.Conv1d(in_channels=64, out_channels=128, kernel_size=3)
        self.bn4 = nn.BatchNorm1d(128)
        self.pool4 = nn.MaxPool1d(kernel_size=2)

        # Calcolo della dimensione dell'input per LSTM dopo i layer CNN
        self.lstm_input_size = self._get_conv_output_size()

        # Definizione dei layer LSTM e Dropout
        self.dropout1 = nn.Dropout(0.7) # Dropout prima del LSTM
        self.lstm = nn.LSTM(input_size=self.lstm_input_size, hidden_size=self.lstm_hidden_size, num_layers=self.lstm_num_layers)
        self.dropout2 = nn.Dropout(0.7) # Dropout dopo il LSTM

        # Definizione dello strato completamente connesso
        self.fc = nn.Linear(self.lstm_hidden_size, self.num_classes)

```

Codice

https://github.com/MatteoMarcoM/Progetto_AIA [4]

Metriche di valutazione

$$\text{Accuracy} = \frac{TP_1 + TP_2 + TP_3}{TP_1 + TP_2 + TP_3 + FN_1 + FN_2 + FN_3 + FP_1 + FP_2 + FP_3}$$

$$\text{MacroPrecision} = \frac{\text{Precision}_1 + \text{Precision}_2 + \text{Precision}_3}{3}$$

$$\text{MacroRecall} = \frac{\text{Recall}_1 + \text{Recall}_2 + \text{Recall}_3}{3}$$

$$\text{MacroF1} = \frac{F1_1 + F1_2 + F1_3}{3}$$

Metriche di valutazione

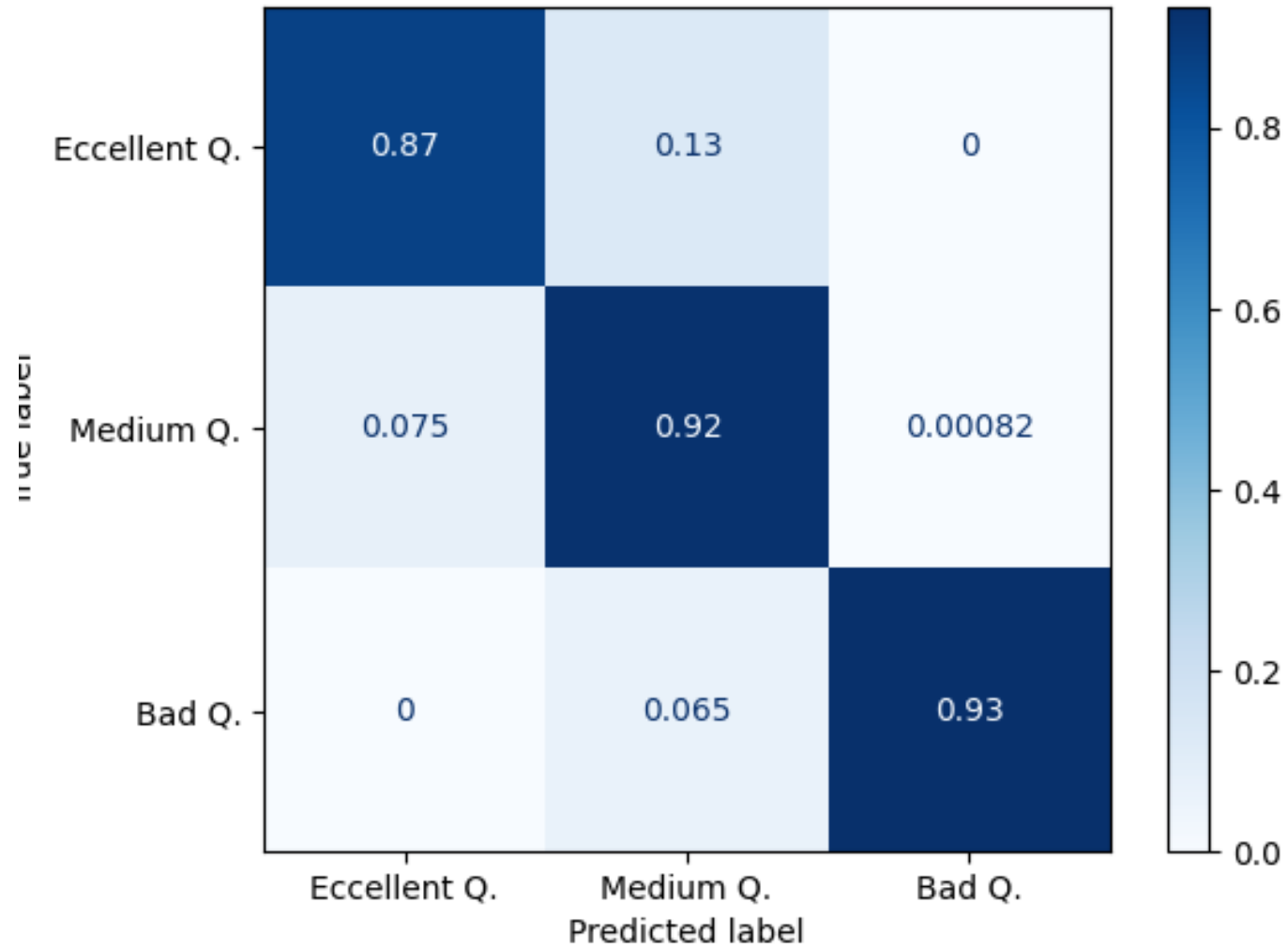
*E inoltre (TP, TN, FP, FN sono calcolati tramite la tecnica **one-versus-all**):*

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} \quad (i = 1,2,3)$$

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \quad (i = 1,2,3)$$

$$F1_i = 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \quad (i = 1,2,3)$$

Matrice di confusione

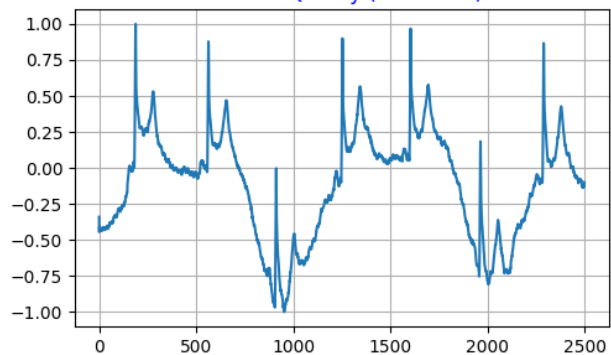


Metriche di valutazione

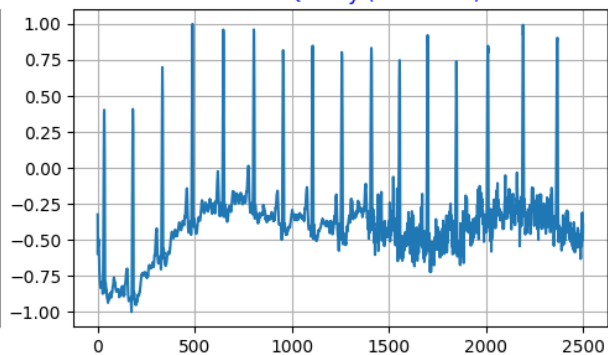
| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| Eccellent Q. | 0.95 | 0.87 | 0.91 | 11821 |
| Medium Q. | 0.79 | 0.92 | 0.85 | 7326 |
| Bad Q. | 1.00 | 0.93 | 0.97 | 3922 |
| accuracy | | | 0.90 | 23069 |
| macro avg | 0.91 | 0.91 | 0.91 | 23069 |
| weighted avg | 0.91 | 0.90 | 0.90 | 23069 |

Sample Predictions from the Model

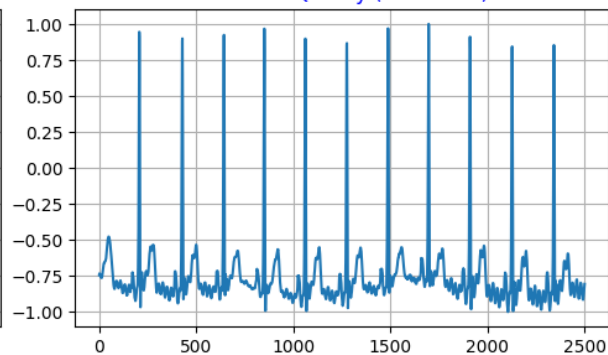
Medium Quality (CORRECT)



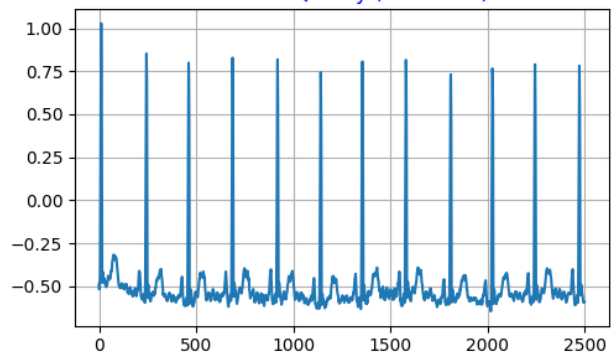
Medium Quality (CORRECT)



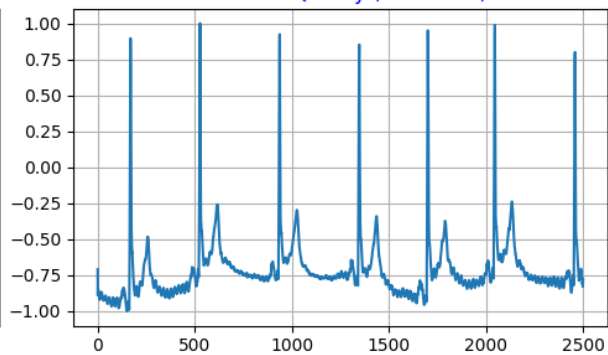
Ecellent Quality (CORRECT)



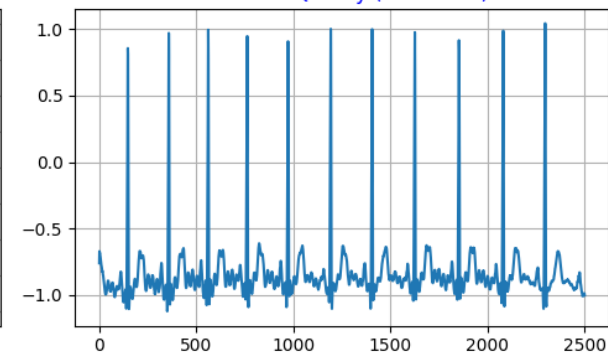
Ecellent Quality (CORRECT)



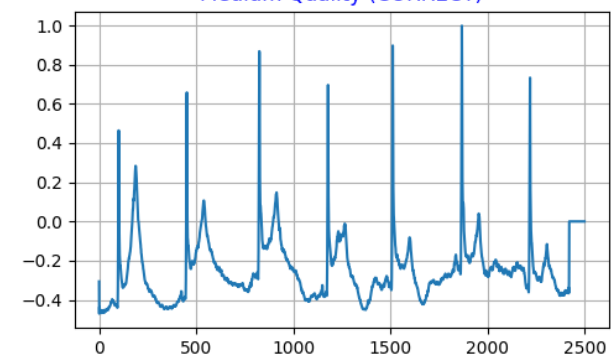
Ecellent Quality (CORRECT)



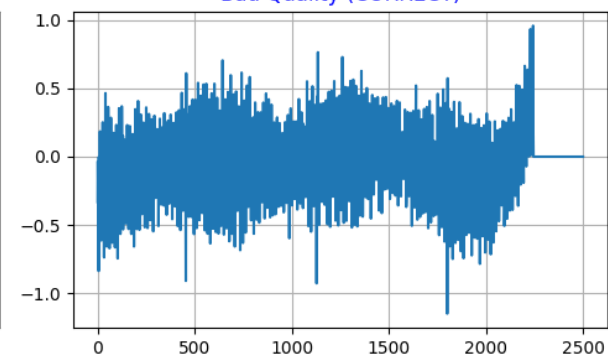
Ecellent Quality (CORRECT)



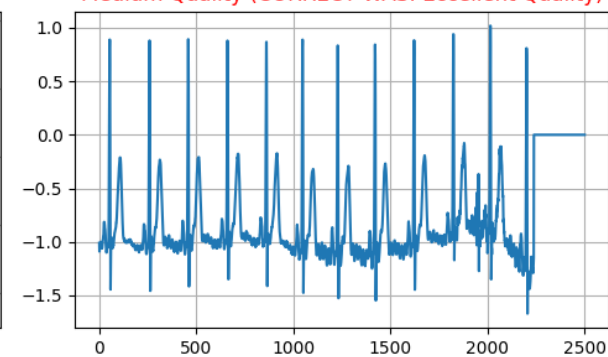
Medium Quality (CORRECT)



Bad Quality (CORRECT)



Medium Quality (CORRECT WAS: Excellent Quality)



Riferimenti Bibliografici

1. He, C.;Wei, Y.;Wei, Y.; Liu, Q.; An, X. **Dynamic Electrocardiogram Signal Quality Assessment Method Based on Convolutional Neural Network and Long Short-Term Memory Network**. *Big Data Cogn. Comput.* 2024, 8, 57.
<https://doi.org/10.3390/bdcc8060057>.
2. Nemcova, A., Smisek, R., Opravilová, K., Vitek, M., Smital, L., & Maršánová, L. (2020). **Brno University of Technology ECG Quality Database (BUT QDB) (version 1.0.0)**. *PhysioNet*. <https://doi.org/10.13026/kah4-0w24>.
3. Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). **PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals**. *Circulation* [Online]. 101 (23), pp. e215–e220.
4. Matteo Marco Montanari, **Project for the Exam of Application of Artificial Intelligence**, a.y. 2023/2024. GitHub Repository.
https://github.com/MatteoMarcoM/Progetto_AIA



1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

Grazie per l'attenzione