



**UNIVERSITÀ DI PARMA**

Progetto del corso Machine Learning:

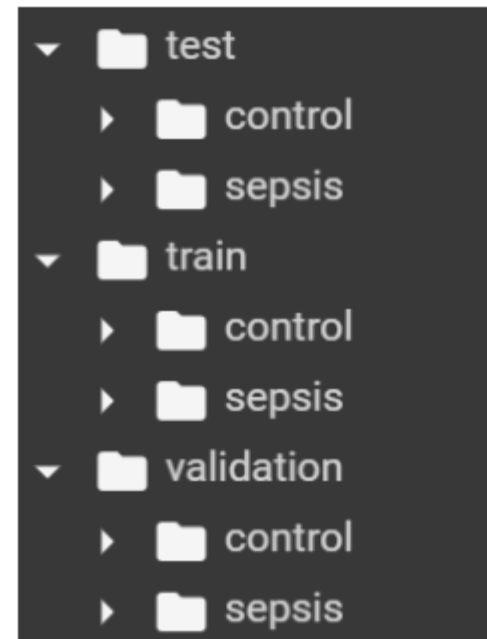
# **Realizzazione di un Autoencoder LSTM per Anomaly Detection su segnali PPG**

**Luca Taverna**

Dicembre 2023

# Introduzione e Dataset utilizzato

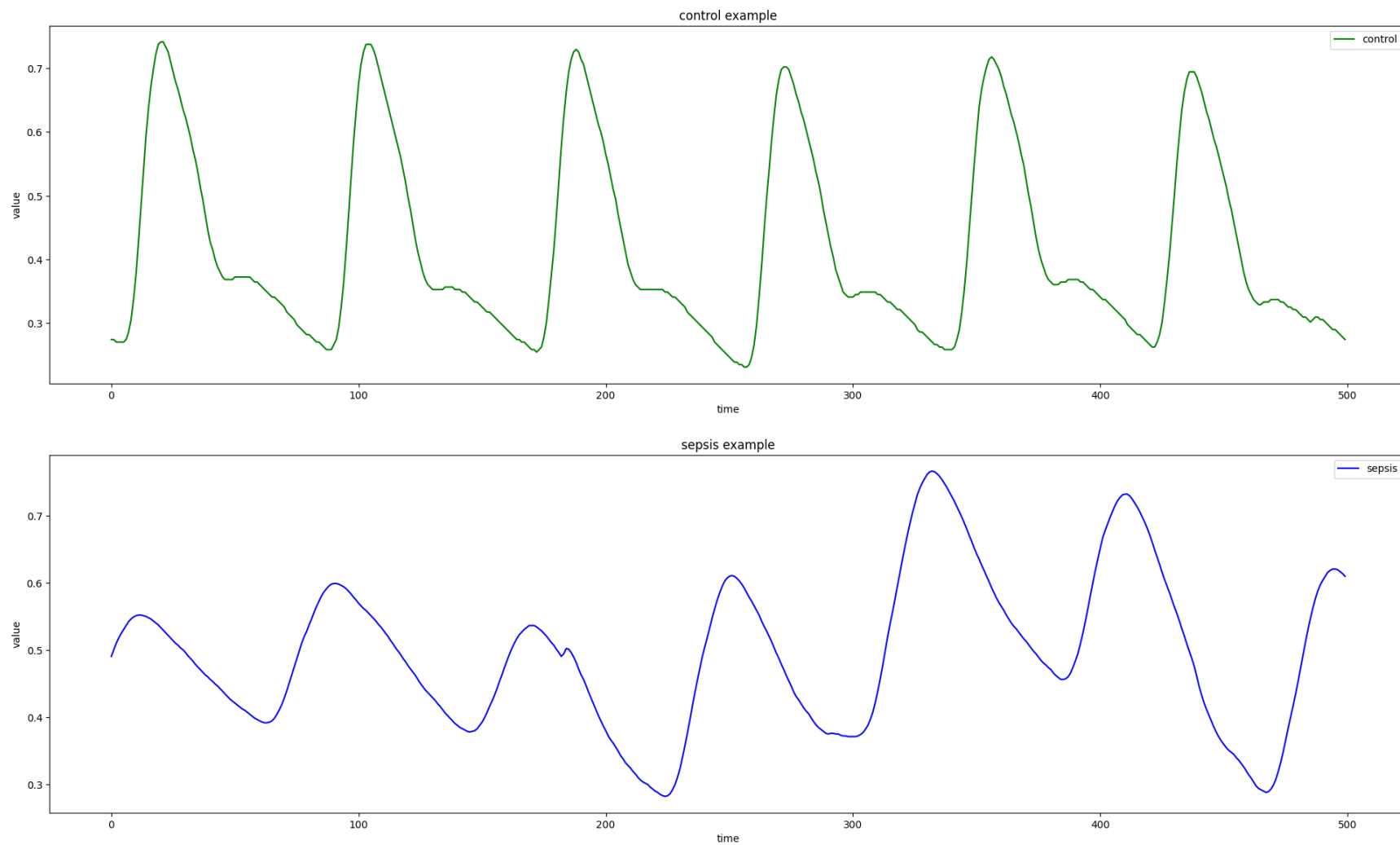
- ✓ **Riconoscere** quanto prima l'avvento della **malattia Seps** ricevendo in input **segnali PPG**
- ✓ 2 tipologie di segnali: *controllo* e *sepsi*
- ✓ Dati preparati precedentemente da altri studenti:
  - ✓ Scaricati dal **MIMIC-III Waveform Database Matched Subset**
  - ✓ Applicate tecniche di **riduzione del rumore** ed **eliminazione** di alcune **istanze**
  - ✓ **Suddivisi** in training (70%), validation (10%) e test set (20%)



# Struttura dei dati

- ✓ **Le finestre temporali** presenti nel dataset sono **raggruppate per paziente**
- ✓ Ogni finestra contiene *15.000 misurazioni* (120 sec x 125 Hz)
- ✓ Ogni cartella contiene file con notazione del tipo:  
“p008281\_#13.npy”
  - “p008281” è l’ID del paziente
  - “#13” identifica la singola finestra temporale nell’insieme delle finestre temporali del paziente

# Esempio segnale di controllo e di sepsi

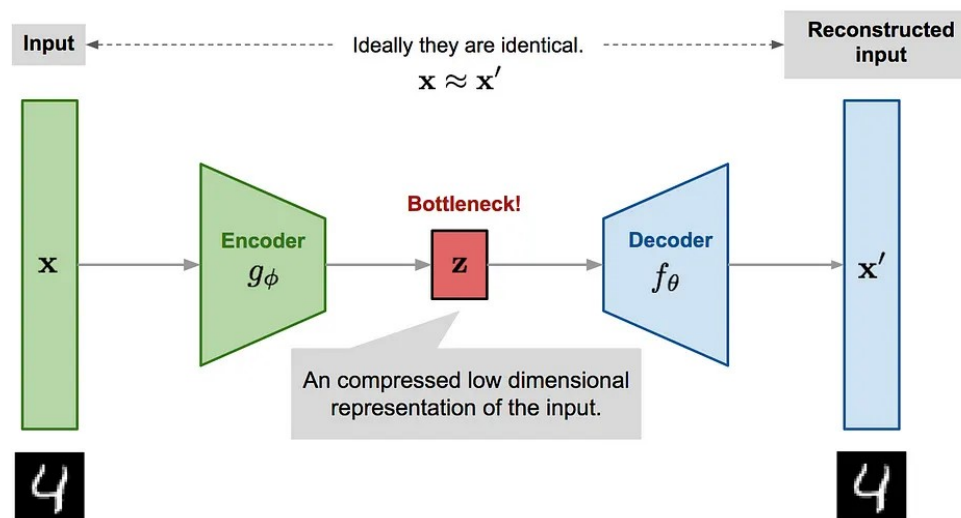


# Tecnologie utilizzate



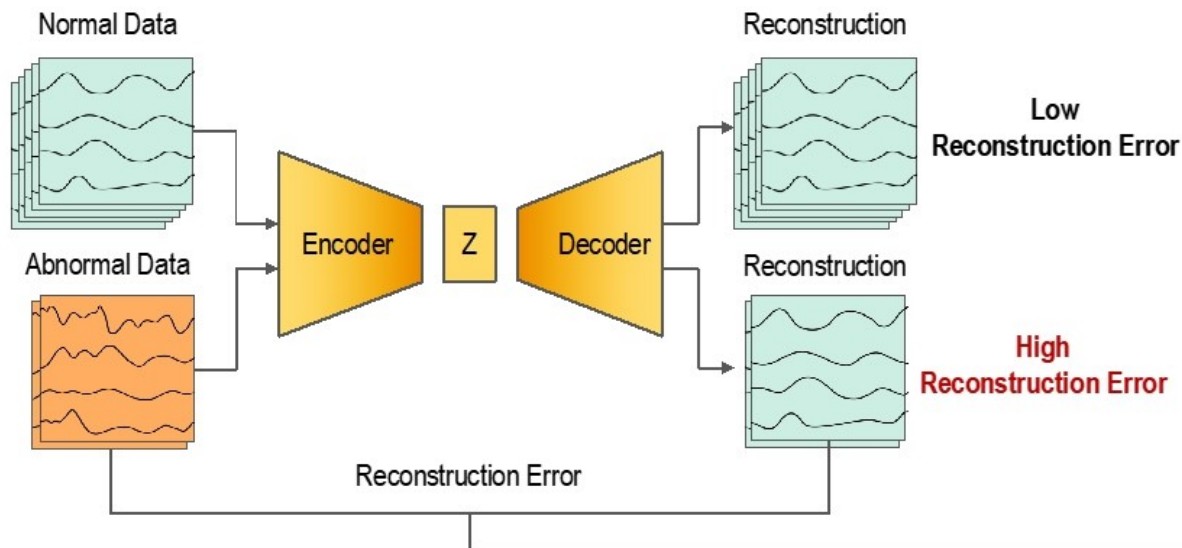
# Introduzione sul modello Autoencoder

- ✓ Un **Autoencoder** è formato da due componenti:
  - ✓ **Encoder**: prende la sequenza di dati in input e ne crea una rappresentazione compressa
  - ✓ **Decoder**: tenta di ricreare il segnale originale partendo dalla rappresentazione compressa



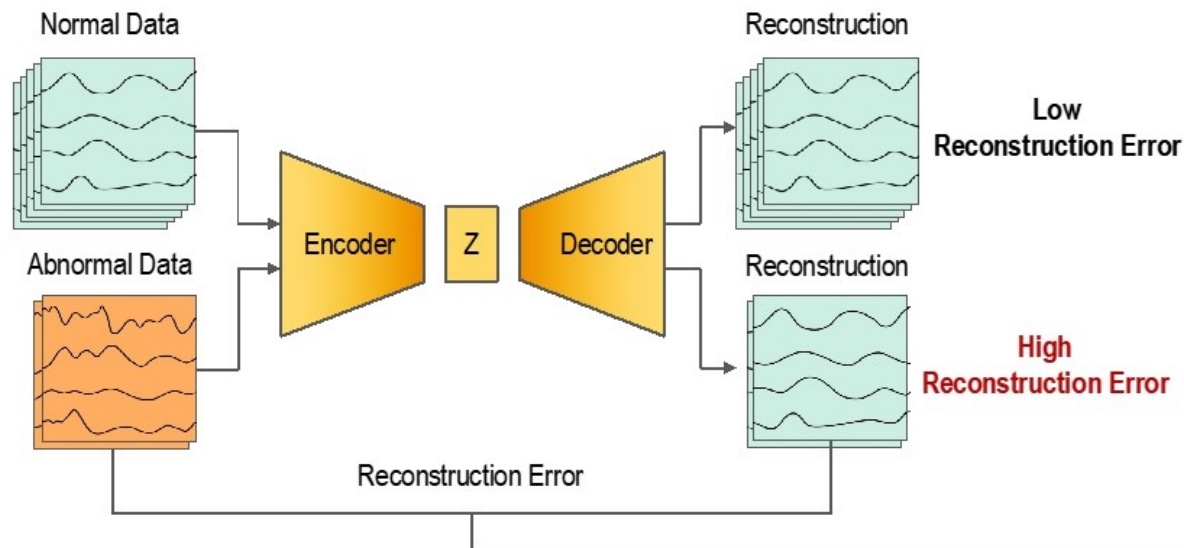
# Task di Anomaly Detection (1/2)

- ✓ **Modello Autoencoder formato da layer LSTM** addestrato sui dati di **controllo** per far sì che li riproduca fedelmente (*differenza minima in termini di funzione di loss utilizzata*)
- ✓ Forniti successivamente in **input segnali di sepsi al modello** la **differenza in termini di loss dovrebbe essere maggiore** rispetto a quella delle istanze di controllo



# Task di Anomaly Detection (2/2)

- ✓ Identificazione di un **valore di loss** da poter utilizzare come **soglia** per poter effettuare la **classificazione**
- ✓ **Ulteriore soglia** per stabilire dopo quanti segnali classificati come sepsi considerare il rispettivo **paziente come «malato di sepsi»**





# Preprocessing del dataset (1/2)

- ✓ **Rimozione** della colonna contenente la **label**
- ✓ **Raggruppamento e ordinamento delle finestre temporali dei pazienti**
- ✓ **Normalizzazione** dei valori tra (0, 1) tramite **MinMaxScaler**
- ✓ Possibilità di aggiungere pazienti del dataset training-sepsis ai dataset di validation e test dato che erano sbilanciati verso i dati di controllo

# Preprocessing del dataset (2/2)

✓ I dataset utilizzati al fine del progetto sono quindi così composti:

**training (control)** dataset shape: (2772, 15000, 1)  
consisting of **95 patients**

**validation (control)** dataset shape: (390, 15000, 1)  
consisting of **13 patients**

**validation (sepsis)** dataset shape: (420, 15000, 1)  
consisting of **14 patients**

**test (control)** dataset shape: (810, 15000, 1)  
consisting of **27 patients**

**test (sepsis)** dataset shape: (873, 15000, 1)  
consisting of **30 patients**

# Modello utilizzato – Autoencoder

```
model = Sequential(  
    encoder,  
    decoder  
], name="LSTMAutoencoder")
```

# Modello utilizzato - Encoder

```
encoder = Sequential([  
    Input(shape=(N_SAMPLES, 1)),  
    LSTM(units=outer_lstm_units, return_sequences=True),  
    Dropout(rate=0.2),  
    LSTM(units=inner_lstm_units, return_sequences=True),  
    Dropout(rate=0.2)  
], name="encoder")
```

- ✓ ***N\_SAMPLES = 15.000 (120 sec x 125 Hz)***
- ✓ ***outer\_lstm\_units > inner\_lstm\_units***

# Modello utilizzato - Decoder

```
decoder = Sequential([  
  
    LSTM(units=inner_lstm_units, return_sequences=True),  
  
    Dropout(rate=0.2),  
  
    LSTM(units=outer_lstm_units, return_sequences=True),  
  
    Dropout(rate=0.2),  
  
    TimeDistributed(Dense(units=1))  
  
], name="decoder")
```

- ✓ `inner_lstm_units` ed `outer_lstm_units` sono invertiti
- ✓ `TimeDistributed` layer permette di applicare il `Dense` layer ad ogni time-step della sequenza di input

# Addestramento del modello (1/2)

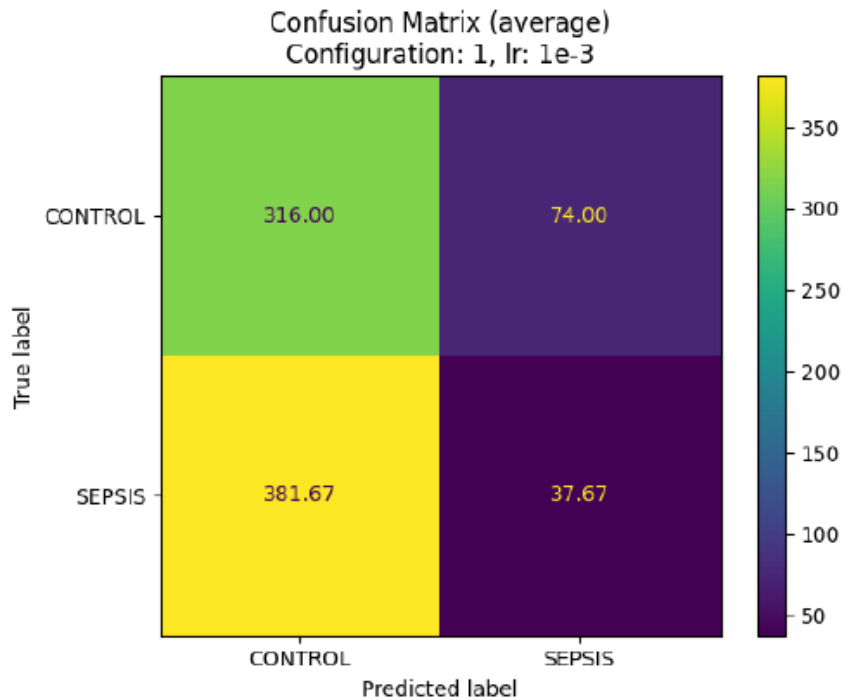
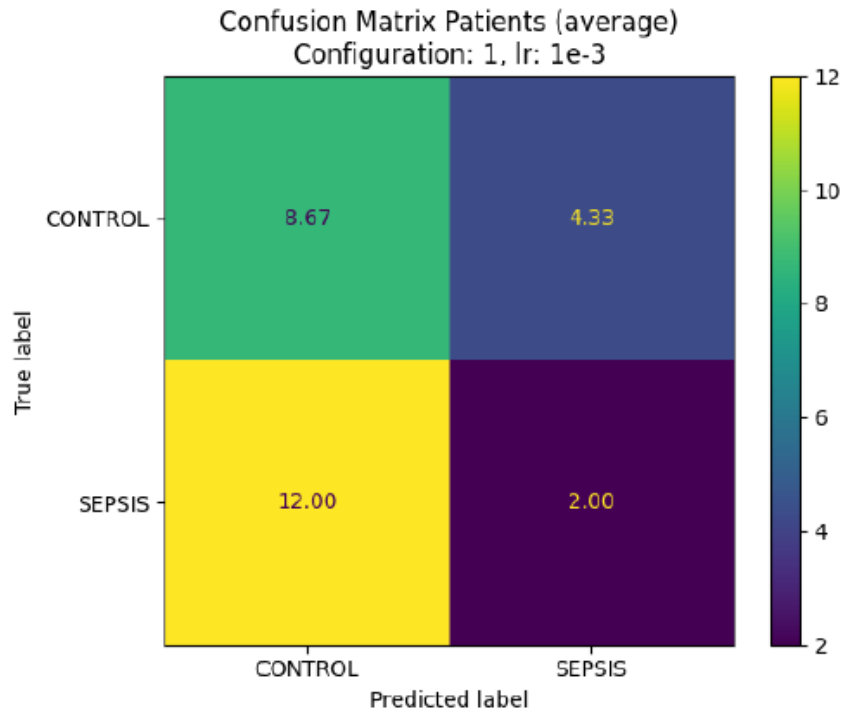
- ✓ Alcuni **parametri** sono stati mantenuti a valori prefissati durante l'addestramento:
  - ✓ `BATCH_SIZE = 64`
  - ✓ `EPOCHS = 50`
  - ✓ `VALIDATION_SPLIT = 0.2`
  - ✓ `LOSS_FUNCTION = "mse"`
  - ✓ `EarlyStopping(monitor="val_loss", patience=5, mode="min")`
  - ✓ `THRESHOLD_FOR_LABEL_CLASSIFICATION = 85%` of the training set loss distribution
  - ✓ `THRESHOLD_PATIENT_HAVE_SEPSIS = after 1 time windows` classified as sepsis

# Addestramento del modello (2/2)

- ✓ Alcuni **parametri** sono stati selezionati tramite una grid search:
  - Valutate 2 configurazioni di unità dei layer LSTM:
    - **Config\_1**: `outer_lstm_units = 64` e `inner_lstm_units = 32`
    - **Config\_2**: `outer_lstm_units = 128` e `inner_lstm_units = 64`
  - E 4 valori di learning rate: `1e-3`, `1e-4`, `1e-5`, `1e-6`
- ✓ Sono state effettuate **5 esecuzioni** per ogni combinazione di `<numero_di_unità, learning_rate>` per trovare la migliore
- ✓ Migliore scelta sulla base del valore di Recall (sepsi) sul validation set:
  1. Valore dei **True positive nella matrice di confusione dei pazienti**
  2. Valori dei **True positive nella matrice di confusione considerando le finestre come indipendenti**

# Performance sul validation set (1/8)

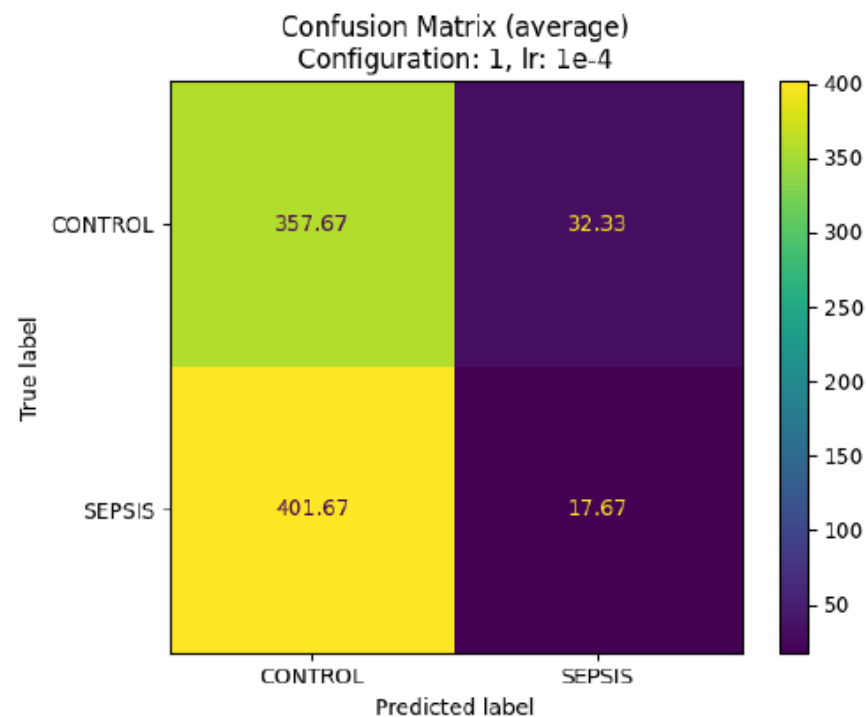
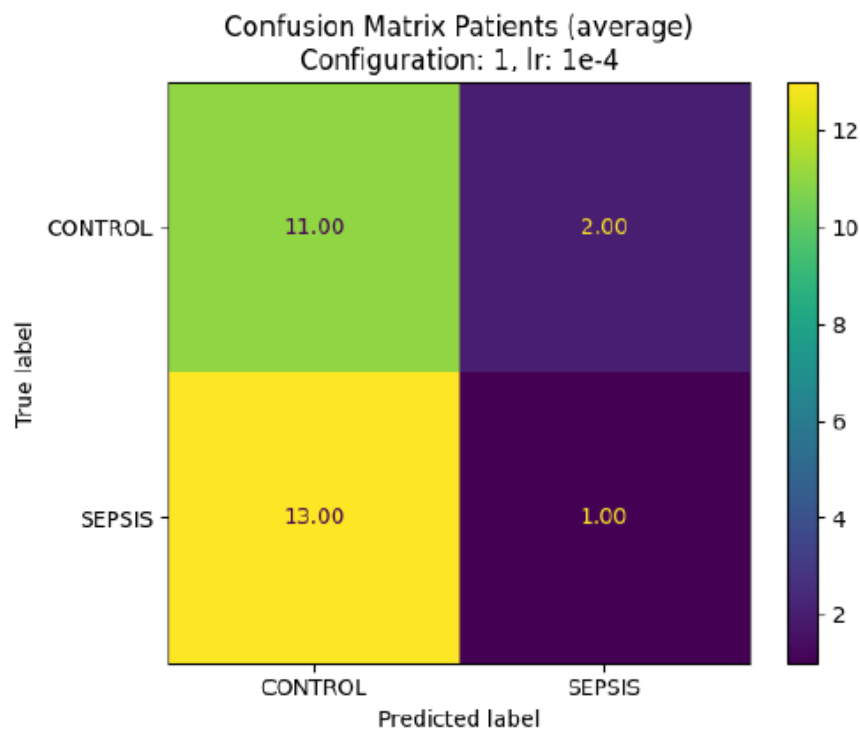
- `outer_lstm_units = 64, inner_lstm_units = 32, learning_rate = 1e-3`:





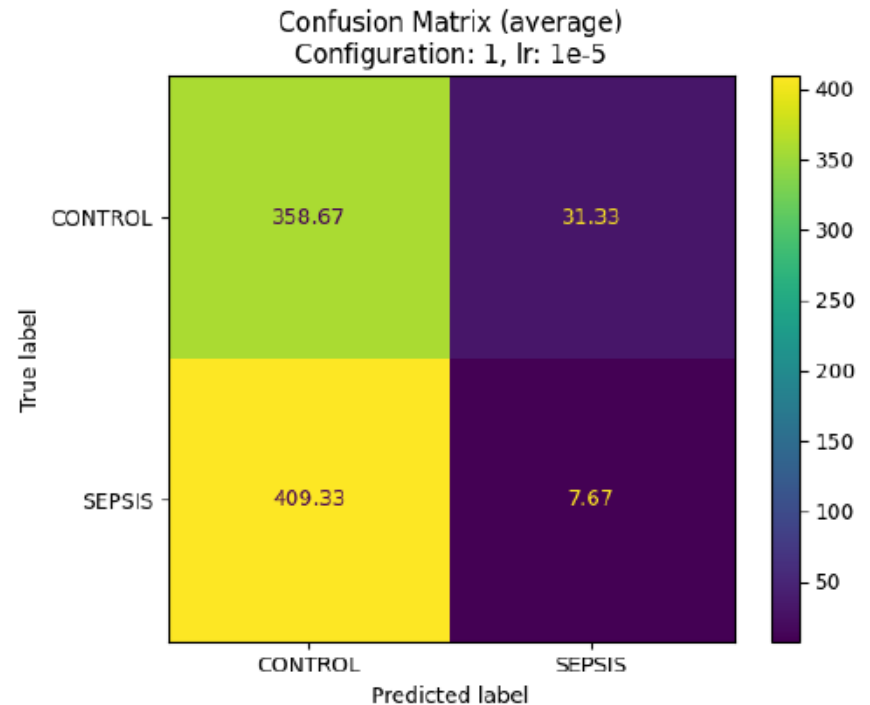
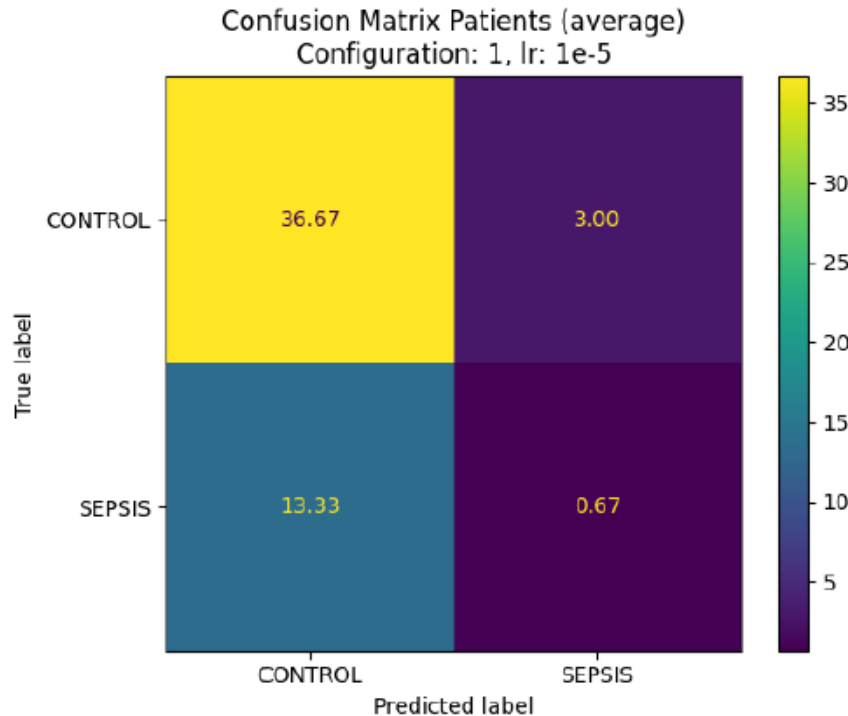
# Performance sul validation set (2/8)

- `outer_lstm_units = 64, inner_lstm_units = 32, learning_rate = 1e-4:`



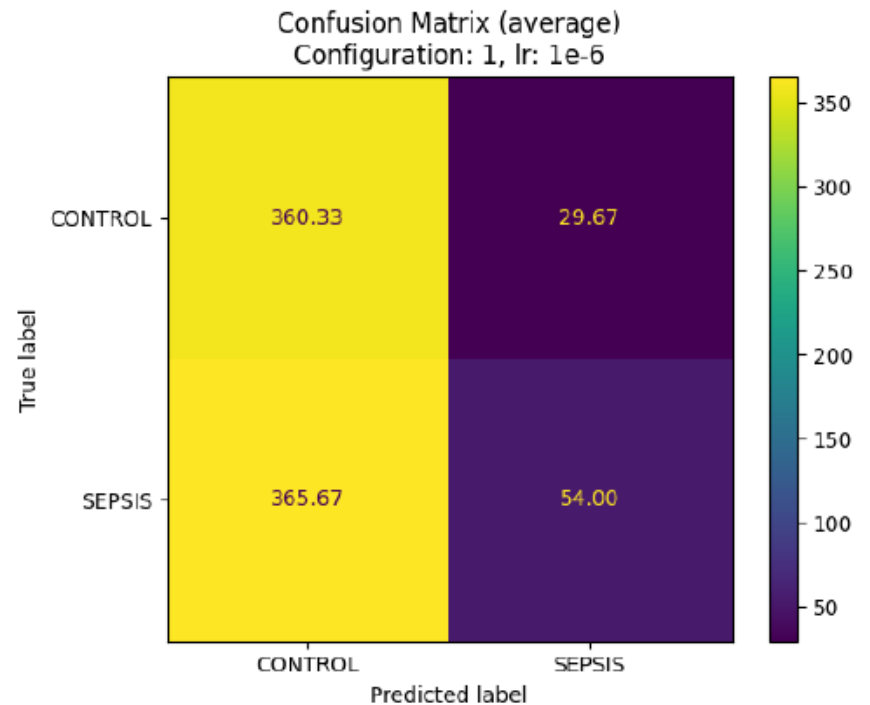
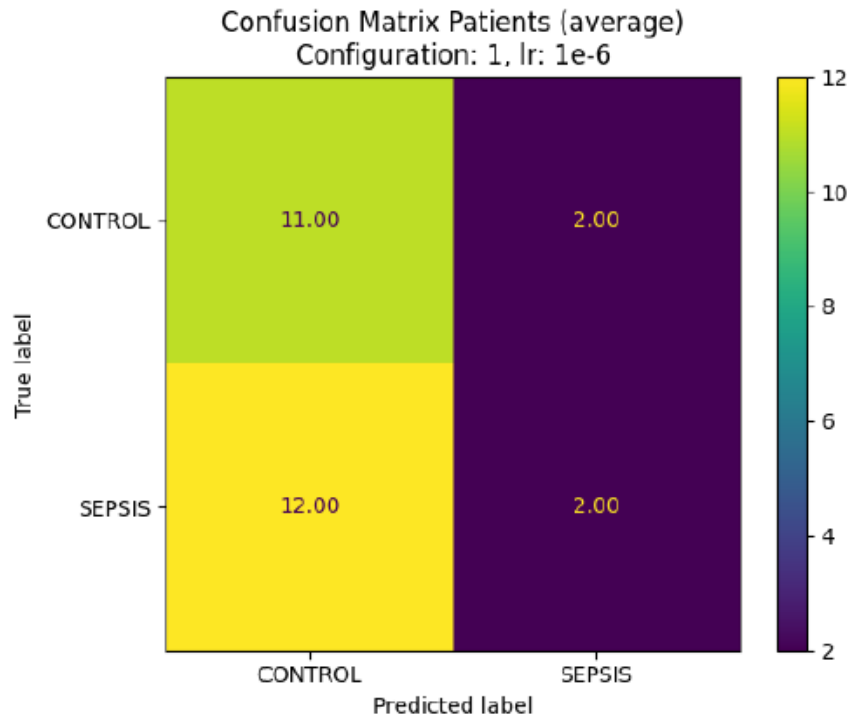
# Performance sul validation set (3/8)

- `outer_lstm_units = 64, inner_lstm_units = 32, learning_rate = 1e-5:`



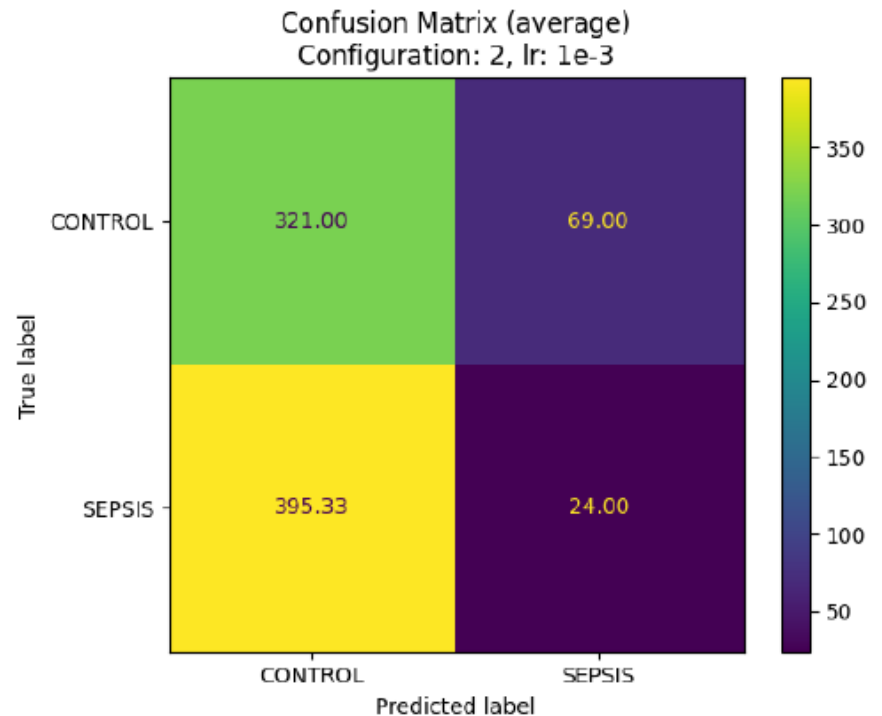
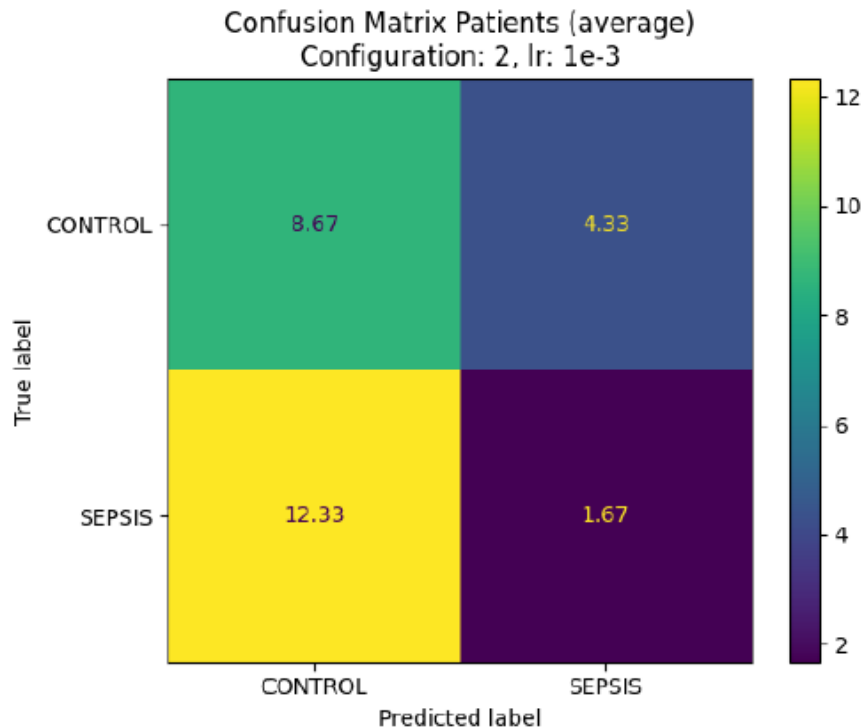
# Performance sul validation set (4/8)

- `outer_lstm_units = 64, inner_lstm_units = 32, learning_rate = 1e-6`:



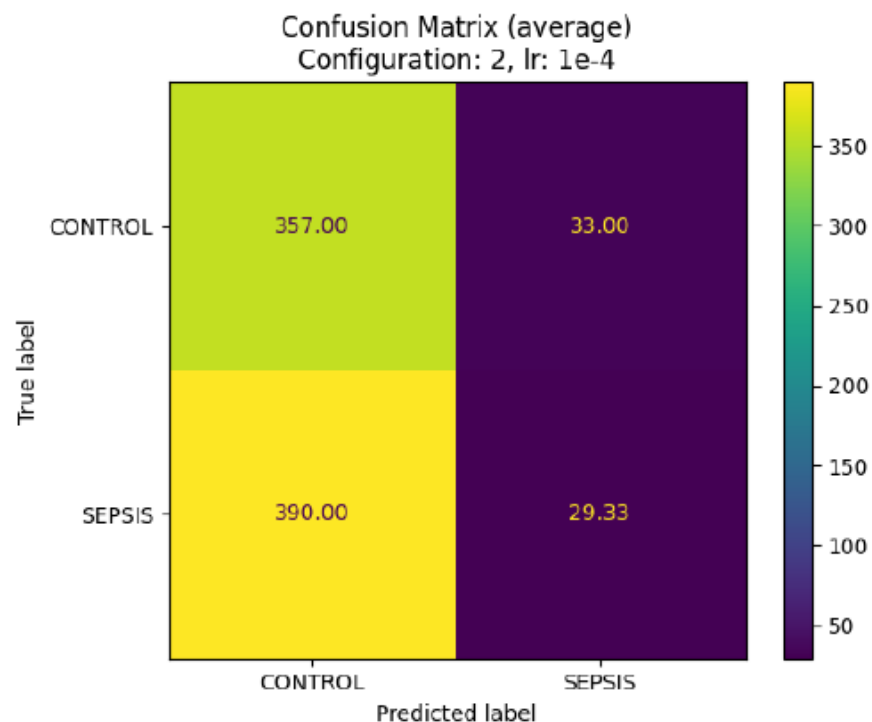
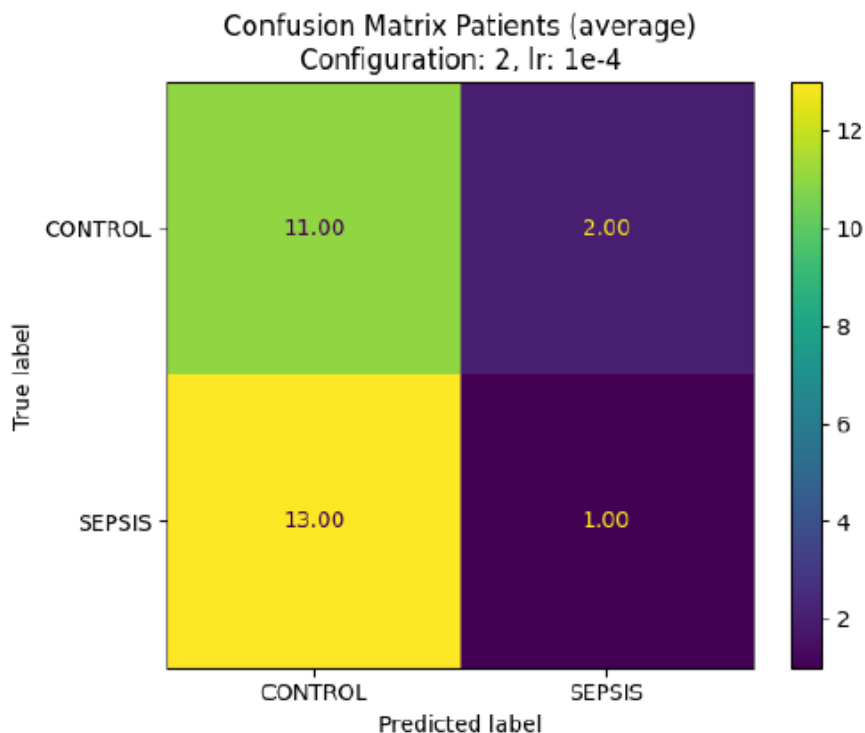
# Performance sul validation set (5/8)

- `outer_lstm_units = 128, inner_lstm_units = 64, learning_rate = 1e-3:`



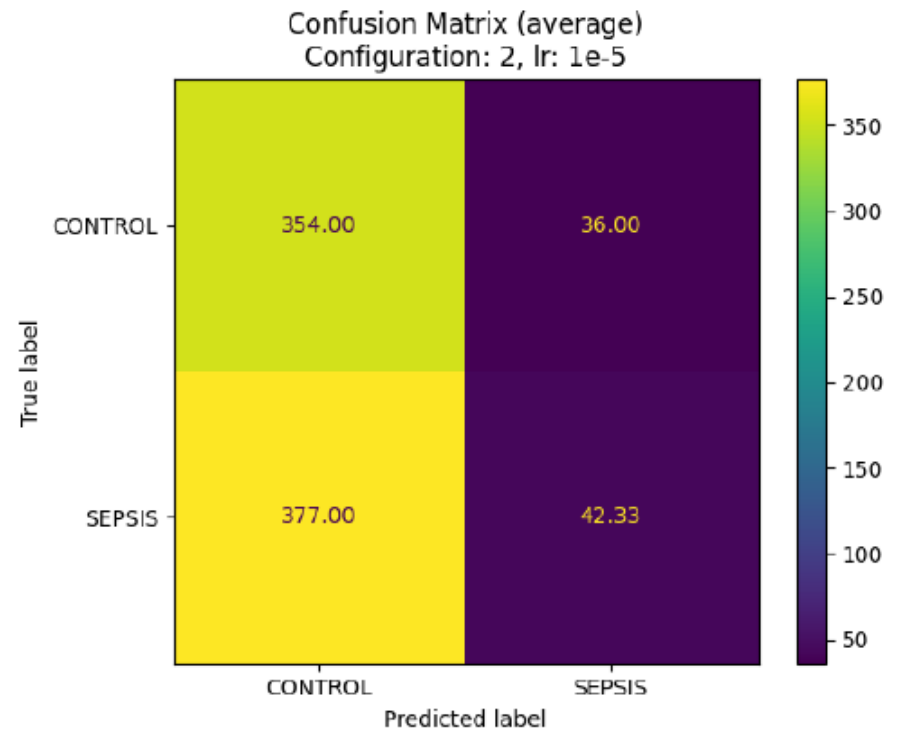
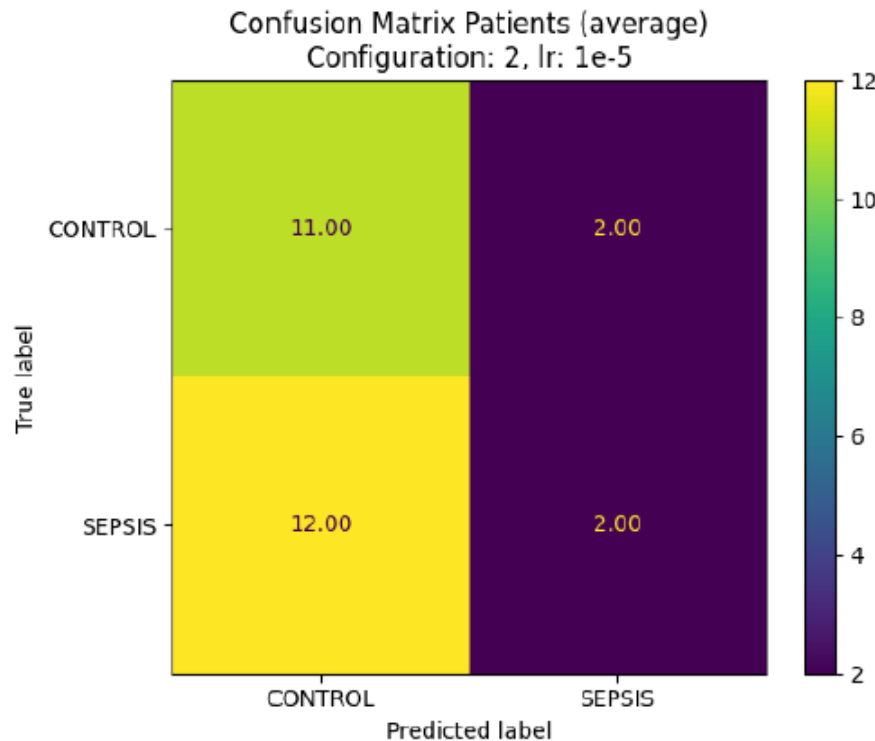
# Performance sul validation set (6/8)

- `outer_lstm_units = 128, inner_lstm_units = 64, learning_rate = 1e-4:`



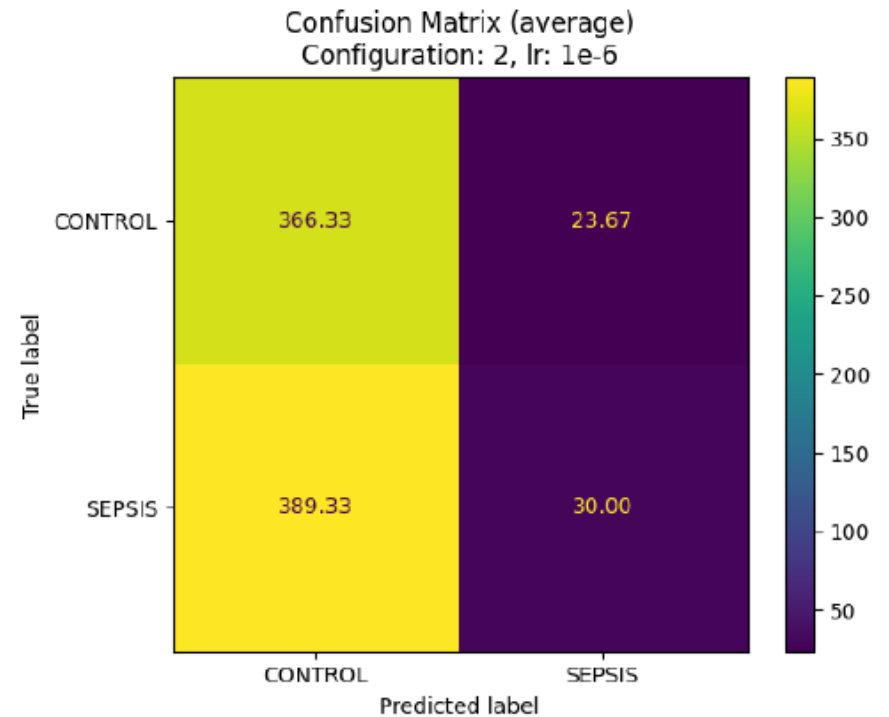
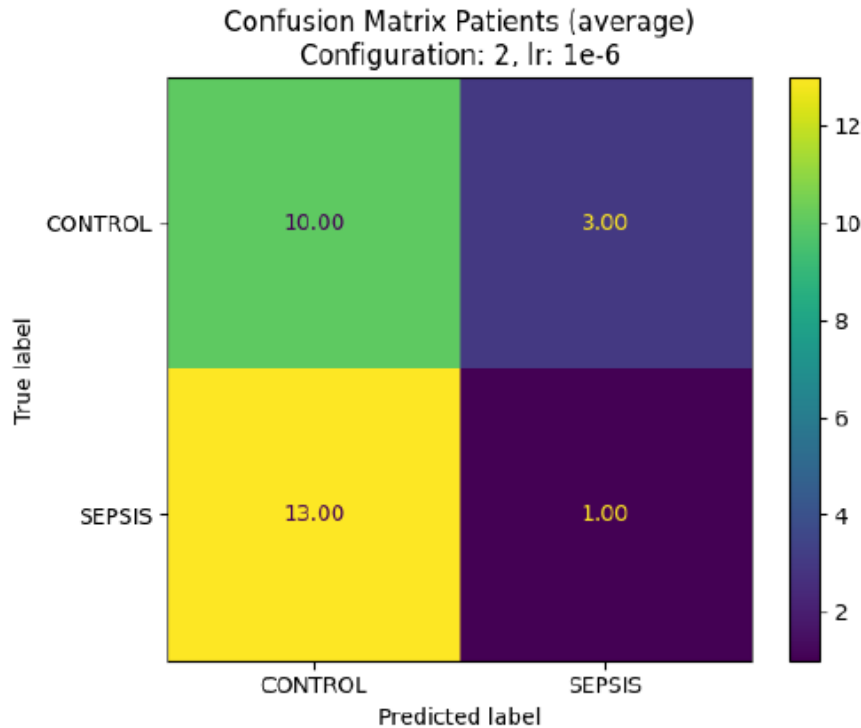
# Performance sul validation set (7/8)

- `outer_lstm_units = 128, inner_lstm_units = 64, learning_rate = 1e-5:`



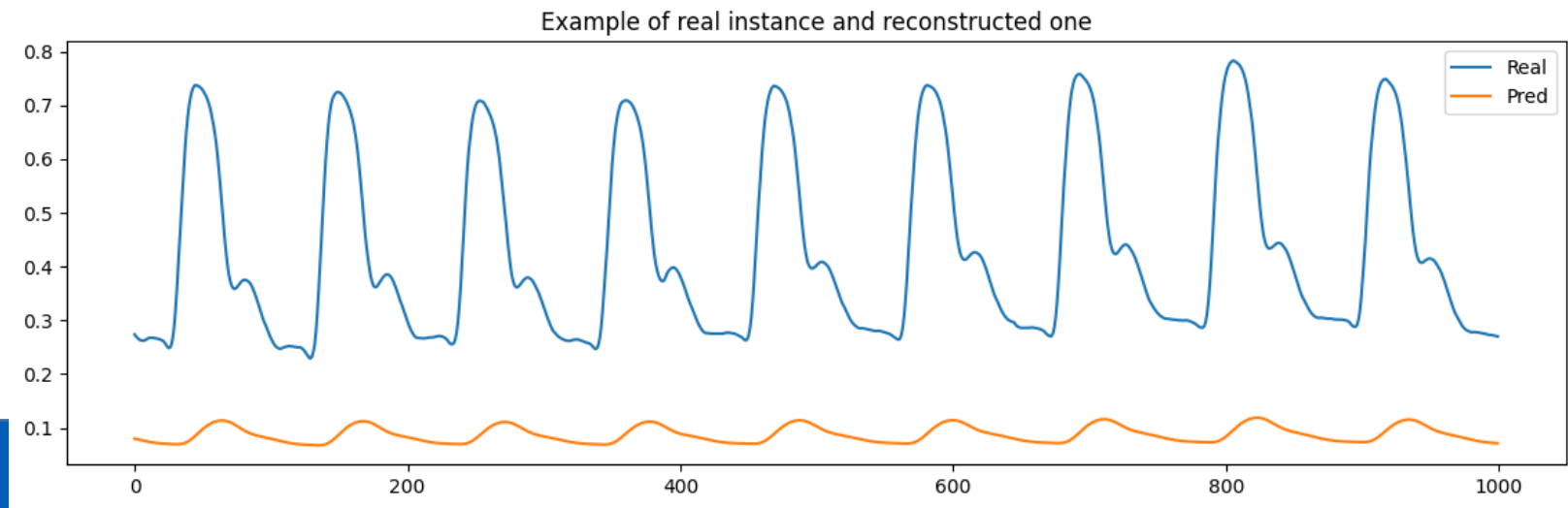
# Performance sul validation set (8/8)

- `outer_lstm_units = 128, inner_lstm_units = 64, learning_rate = 1e-6:`



# Miglior combinazione di parametri (1/4)

- ✓ La configurazione che fornisce le migliori performance sulla metrica selezionata è:
  - `outer_lstm_units = 64`, `inner_lstm_units = 32` e `learning_rate = 1e-6`
- ✓ La scelta di valutare i modelli tramite le performance sul validation set **non tiene però in considerazione il livello di fedeltà di riproduzione del segnale**
- ✓ Esempio di segnale riprodotto dal modello addestrato con questa configurazione:



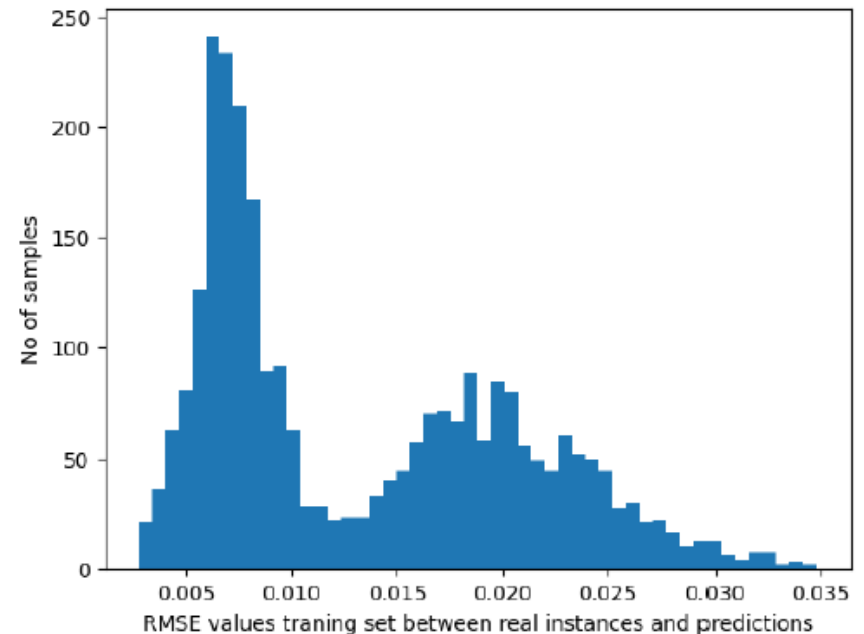
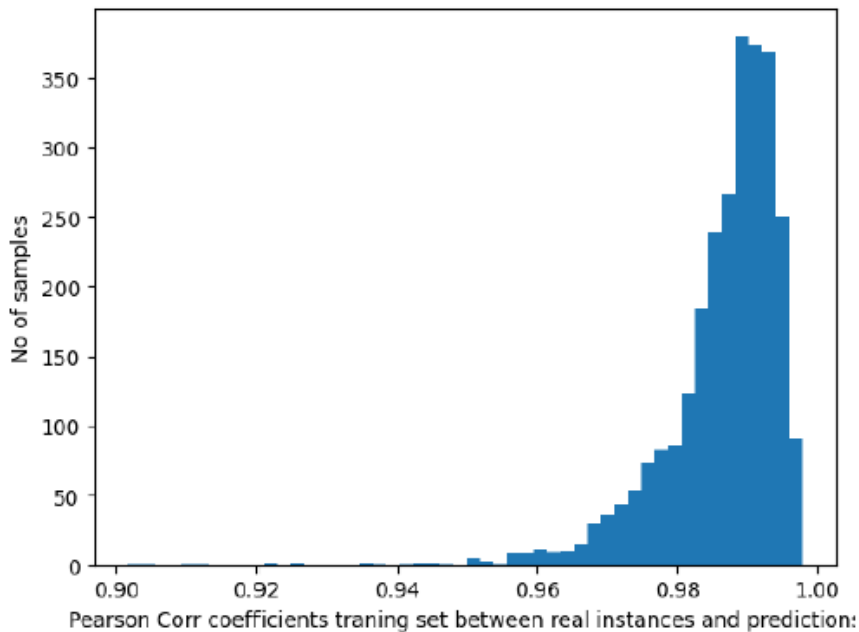


# Miglior combinazione di parametri (2/4)

- ✓ Sono quindi state introdotte ulteriori due metriche per scegliere la miglior combinazione di parametri:
  - **Coefficiente di correlazione di Pearson**
  - **Root Mean Squared Error**
- ✓ Tenendo in considerazione queste metriche il modello che si comporta meglio è quello avente questi parametri:
  - `outer_lstm_units = 128,`
  - `inner_lstm_units = 64,`
  - `learning_rate = 1e-4.`

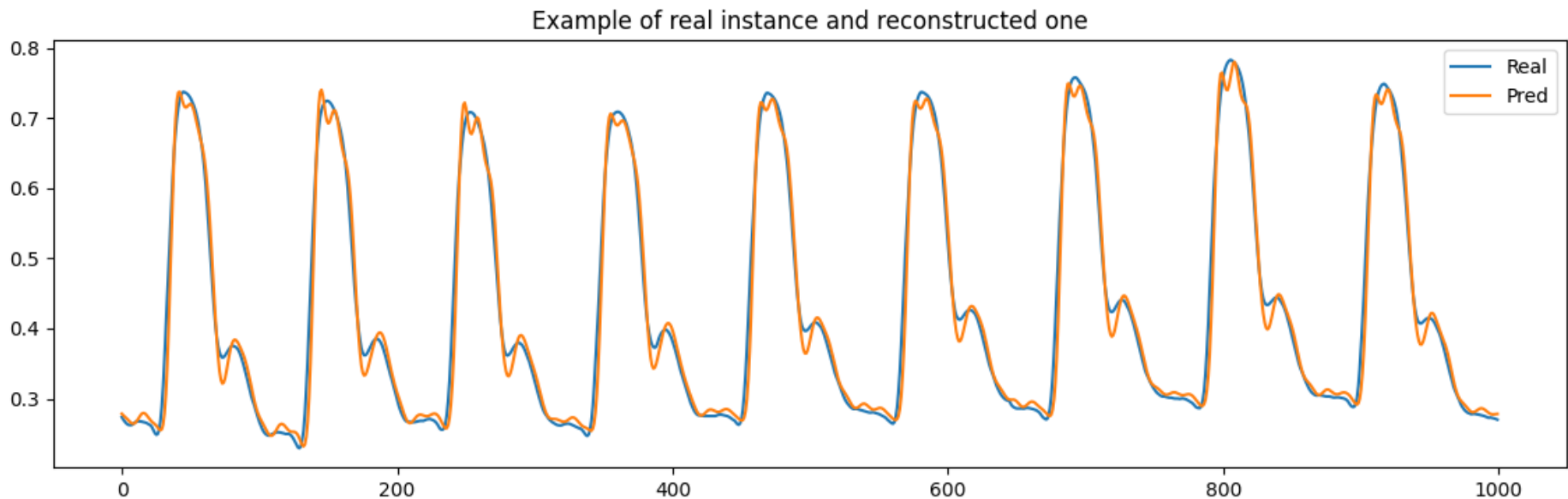
# Miglior combinazione di parametri (2/4)

- Che fornisce i seguenti valori di correlazione di Pearson e RMSE (calcolati sul training set):



# Miglior combinazione di parametri (4/4)

- ✓ La distribuzione dei valori mostrata nella slide precedente dimostra una **ricostruzione ottimale dei segnali in ingresso**
  - ✓ Valore di correlazione di *Pearson* molto vicino a 1
  - ✓ Valore di *RMSE* molto vicino a 0
- ✓ Esempio di ricostruzione di segnale:



# Sommario del modello (1/3)

Model: "LSTMAutoencoder"

Layer (type)	Output Shape	Param #
encoder (Sequential)	(None, 15000, 64)	115968
decoder (Sequential)	(None, 15000, 1)	131969

Total params: 247937 (968.50 KB)

**Trainable params: 247937** (968.50 KB)

Non-trainable params: 0 (0.00 Byte)

# Sommario del modello (2/3)

Model: "encoder"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 15000, 128)	66560
dropout (Dropout)	(None, 15000, 128)	0
lstm_1 (LSTM)	(None, 15000, 64)	49408
dropout_1 (Dropout)	(None, 15000, 64)	0

Total params: 115968 (453.00 KB)

**Trainable params: 115968** (453.00 KB)

Non-trainable params: 0 (0.00 Byte)



# Sommario del modello (3/3)

Model: "decoder"

Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 15000, 64)	33024
dropout_2 (Dropout)	(None, 15000, 64)	0
lstm_3 (LSTM)	(None, 15000, 128)	98816
dropout_3 (Dropout)	(None, 15000, 128)	0
time_distributed (TimeDistributed)	(None, 15000, 1)	129

Total params: 131969 (515.50 KB)

**Trainable params: 131969** (515.50 KB)

Non-trainable params: 0 (0.00 Byte)

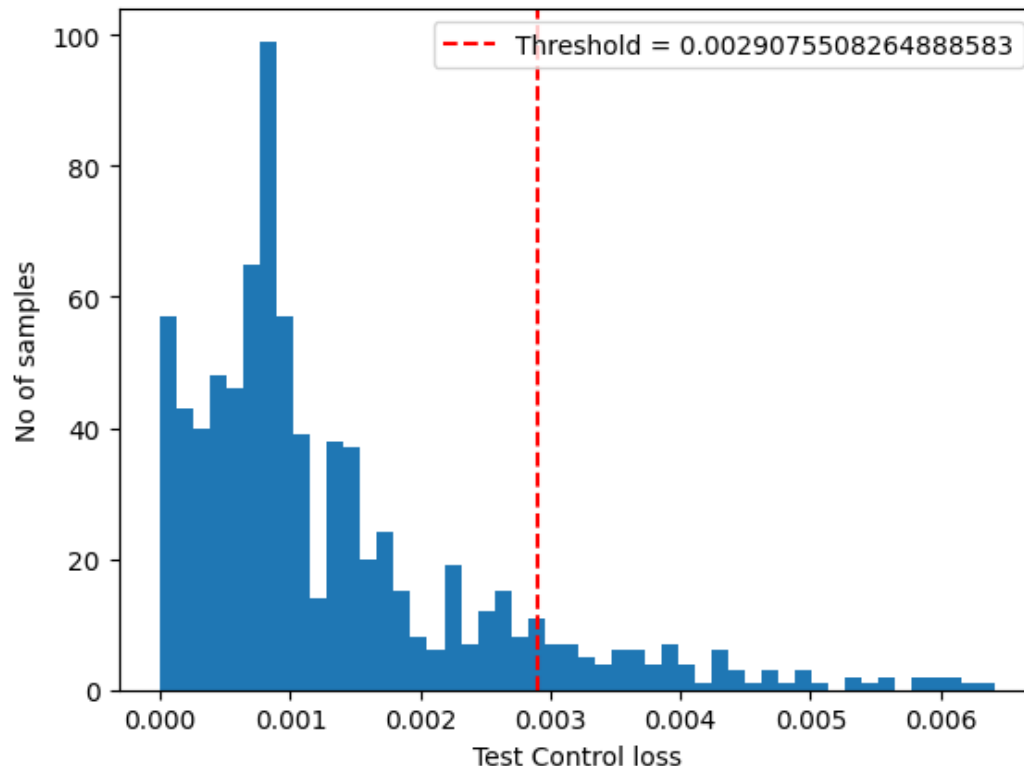


# Performance sul test set (1/5)

- ✓ Predizioni corrette sulle istanze di controllo:

`Correct control predictions on test set: 723/810`

- ✓ di cui si riporta anche la distribuzione dei valori di loss:

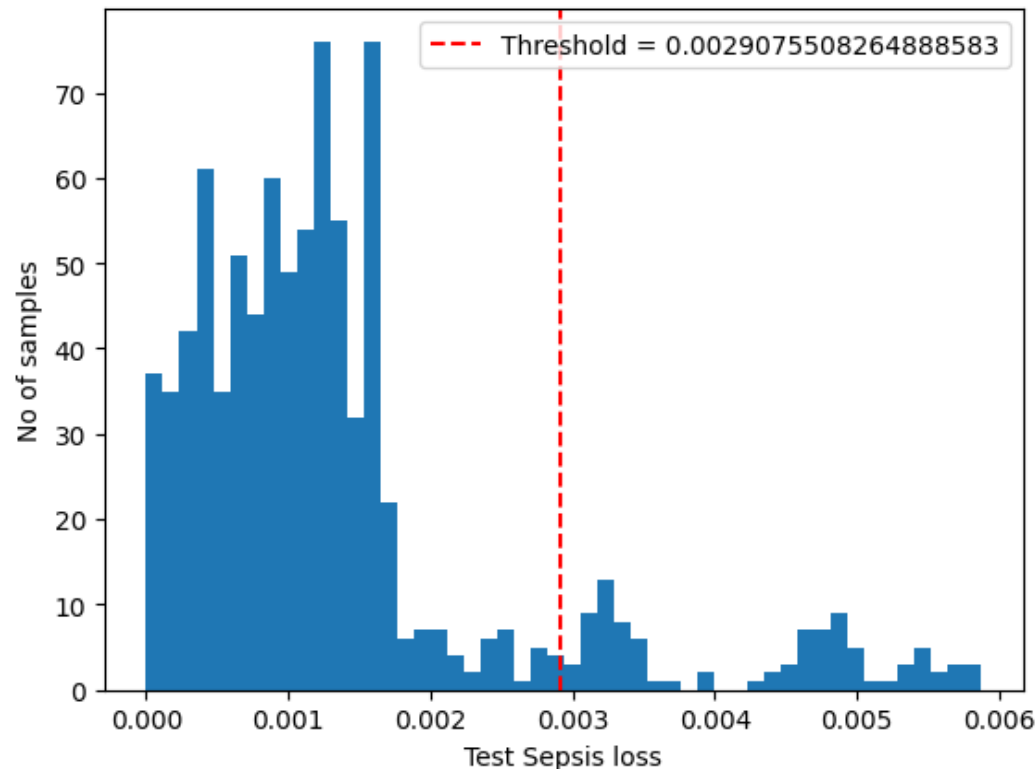


# Performance sul test set (2/5)

- ✓ Predizioni corrette sulle istanze di sepsi:

**Correct sepsis predictions on test set: 95/873**

- ✓ di cui si riporta anche la distribuzione dei valori di loss:

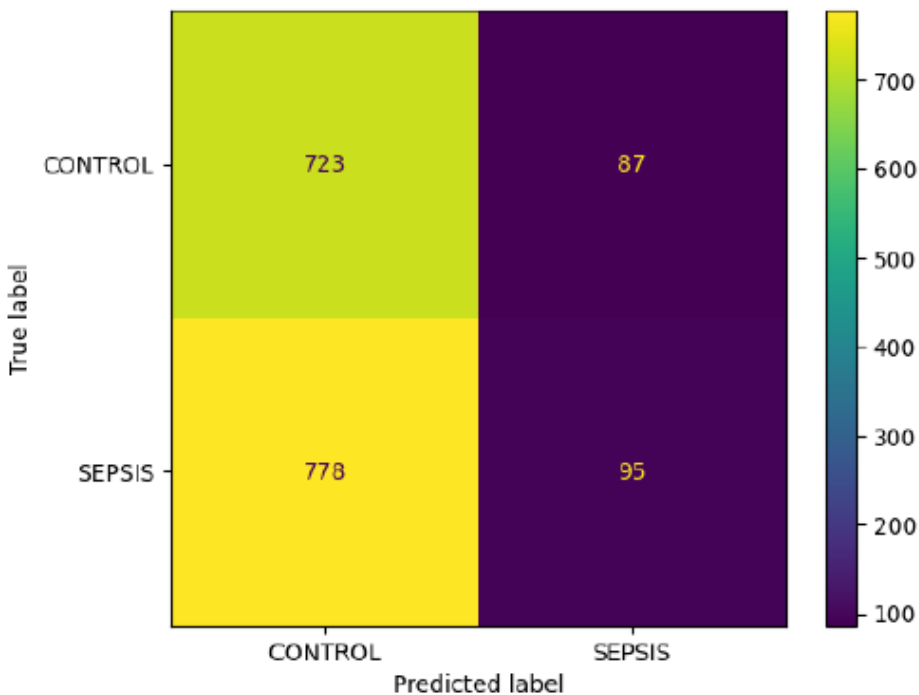




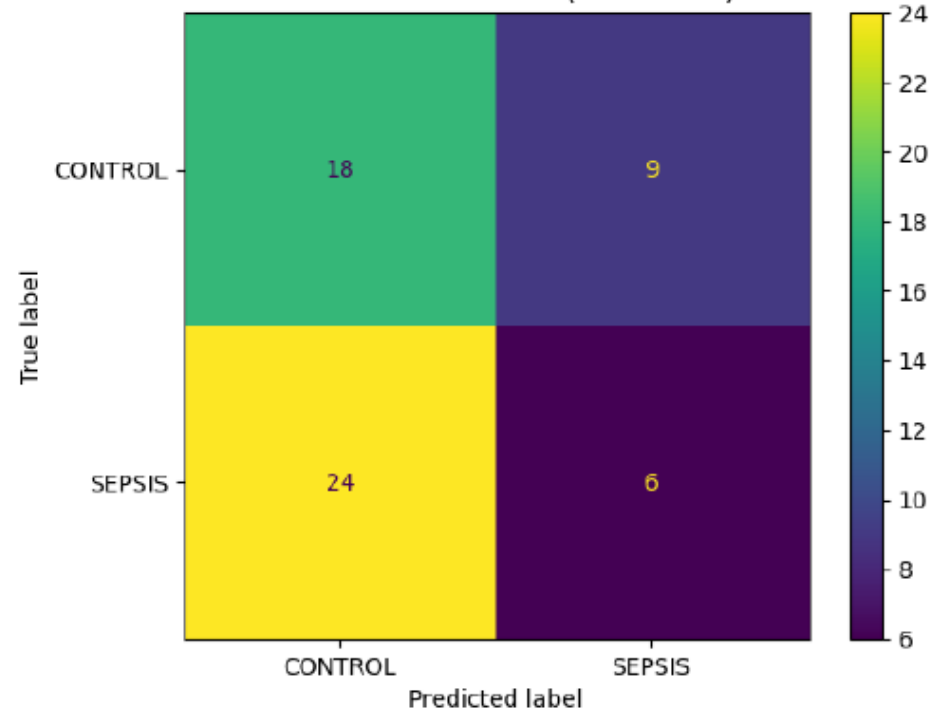
# Performance sul test set (3/5)

## ✓ Matrici di confusione:

Test Confusion Matrix



Test Confusion Matrix (of Patients)



# Performance sul test set (4/5)

- ✓ **Accuracy, precision, recall e score F1** calcolate considerando tutte le finestre indipendenti:

**accuracy** test set: 0.4860368389780155

**precision** test set: 0.521978021978022

**recall** test set: 0.10882016036655212

**F1 score** test set: 0.18009478672985785

- ✓ E quelle calcolate considerando i singoli pazienti malati/sani:

**accuracy** test set (of patients): 0.42105263157894735

**precision** test set (of patients): 0.4

**recall** test set (of patients): 0.2

**F1 score** test set (of patients): 0.266666666666666666

# Performance sul test set (5/5)

- ✓ Si riporta infine **l'anticipazione**, in termini di numero di finestre temporali, **con la quale è stata identificata la sepsi nei pazienti effettivamente malati**:

`23 time windows`

- ✓ Che corrispondono a:

$$23 * 120 \text{ (lunghezza finestre)} / 60 \text{ (secondi di un minuto)} =$$

**46 minuti di anticipo**

- ✓ Nonostante il modello non riesca a riconoscere adeguatamente l'avvento della malattia, **nei pazienti in cui effettivamente ci riesce lo fa con largo anticipo**

# Conclusioni (1/4)

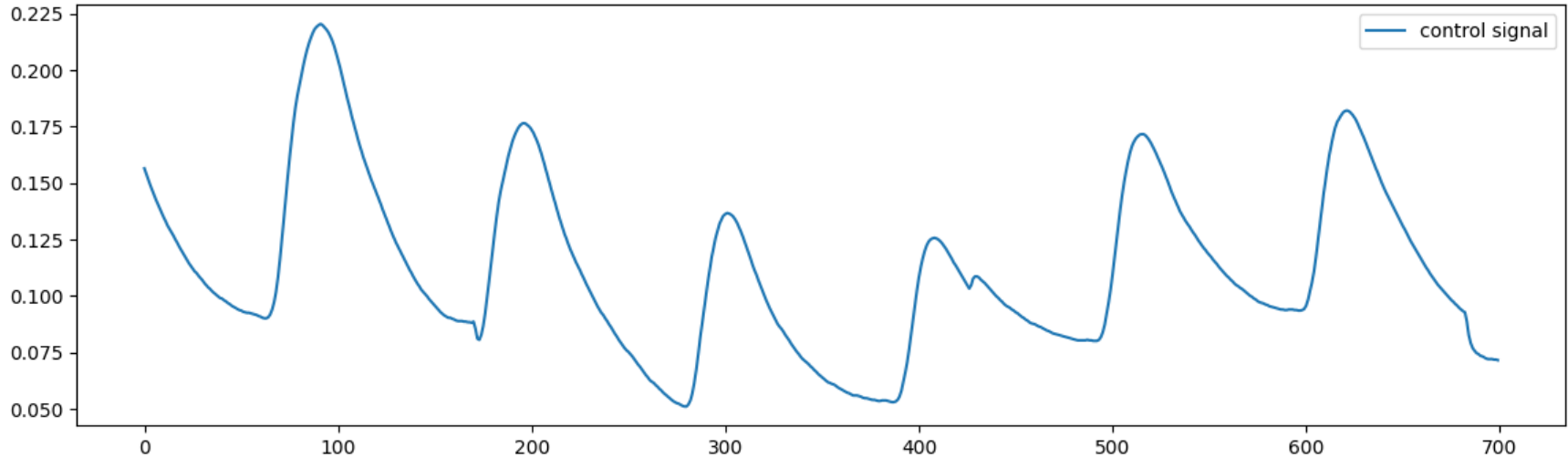
- ✓ **Riassunto del progetto:** verificare le performance di un Autoencoder formato da layer LSTM per riconoscere la sepsi utilizzando segnali PPG di alcuni pazienti tramite la tecnica anomaly detection.
- ✓ I valori di loss ottenuti quando si chiede al modello di predire istanze di sepsi sono pressoché simili a quelli ottenuti sui dati di controllo
- ✓ Da questo si può dedurre che, probabilmente, **il modello utilizzato non è ottimale per il task in analisi**

# Conclusioni (2/4)

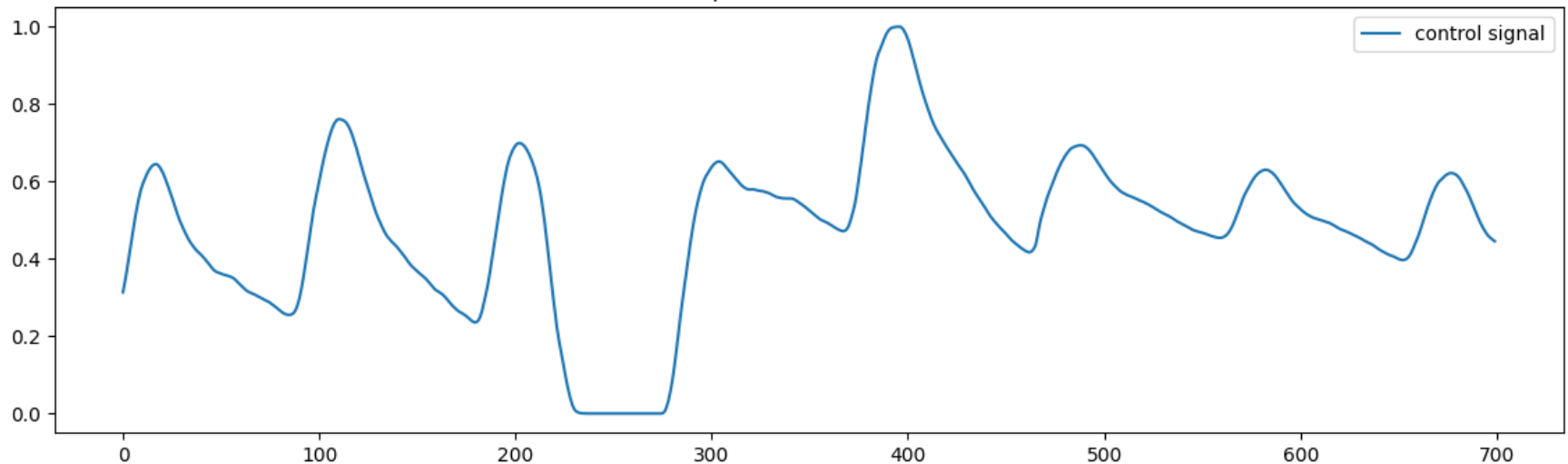
- ✓ Le finestre temporali del dataset non sono certamente ottimali:
  - **Dispositivo utilizzato** per la rilevazione dei segnali PPG infatti non fornisce dati molto buoni per una successiva analisi a causa della **facilità con cui si presenta rumore o errori di misurazione** in essi
  - Inoltre ai dati utilizzati sono state **applicate tecniche per la gestione dei valori mancanti e per la riduzione del rumore**
- ✓ Da questo si ottiene che molti dati di controllo sono tutto fuorché segnali PPG eccellenti di un paziente sano
- ✓ Ed invece sono presenti istanze della classe di sepsi che presentano segnali PPG tipici di un paziente che non presenta la suddetta malattia
- ✓ Questi fattori potrebbero aver **compromesso la fase di addestramento**

# Conclusioni (3/4)

Example of control instance

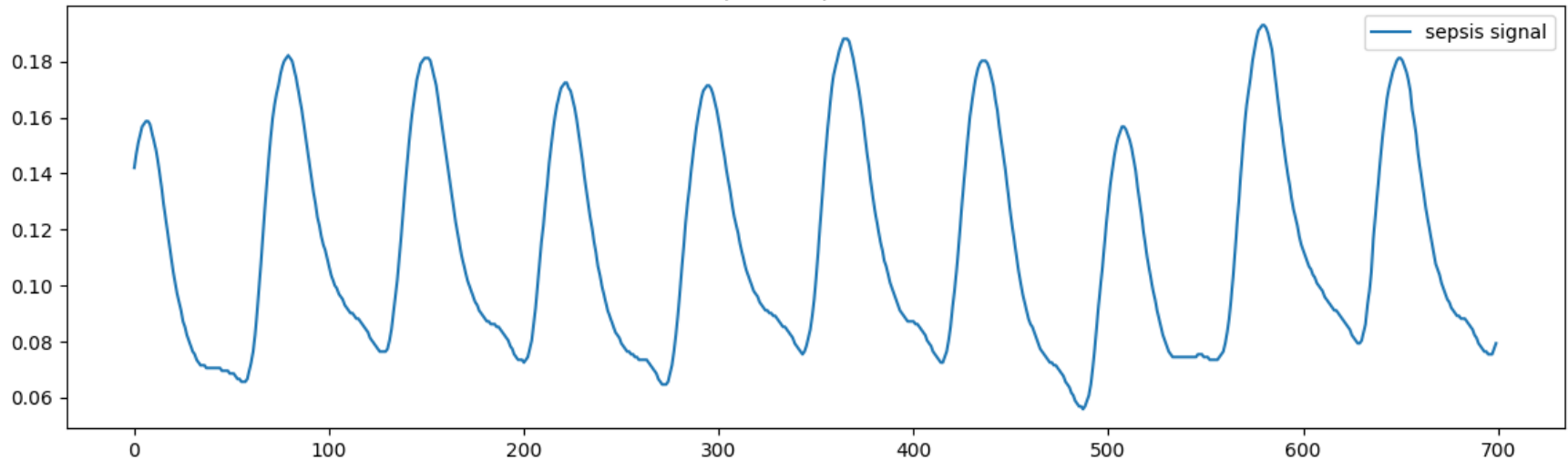


Example of control instance

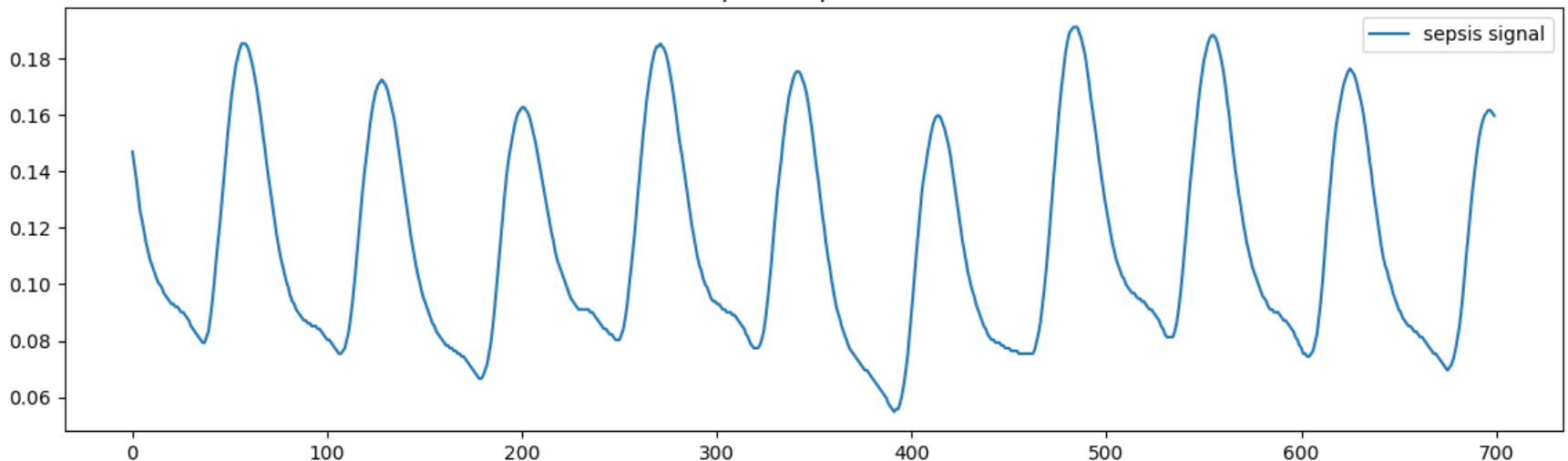


# Conclusioni (4/4)

Example of sepsis instance



Example of sepsis instance



# Sviluppi futuri

- ✓ **Utilizzare un diverso dataset**, o di prendere in analisi la sua versione completa e **prestare** successivamente **particolare attenzione alle righe che vengono eliminate ed alle tecniche di preprocessing utilizzate**
- ✓ **Sottoporre un maggior numero di parametri alla tecnica grid search** per trovare la migliore combinazione degli stessi