Politecnico di Torino

Facoltà Di Ingegneria

Corso Di Laurea Magistrale In Ingegneria Biomedica

A. A. 2020/2021
Ottobre 2021

Costruzione di uno strumento di aiuto all'individuazione delle condizioni ottimali di svezzamento dalla ventilazione meccanica tramite tecniche di Machine Learning e Data Science

Relatrice:
Prof. Gabriella Balestra
Correlatrice:
Prof. Samanta Rosati

Candidata:
Ilaria Lazzarini

ABSTRACT

La ventilazione meccanica forzata è una tecnica clinica utilizzata per diversi scopi, è fondamentale infatti per ridurre o sostituire lo sforzo respiratorio e mantenere lo scambio gassoso ottimale. Le cause che portano all'utilizzo di tale tecnica sono molteplici, infarti, asma, condizione di ostruzione delle vie aree, ma anche certi tipi di operazioni chirurgiche effettuate in anestesia generale.

La tecnica di ventilazione meccanica non è una terapia, ma un aiuto al soggetto. Non ha lo scopo di curare il paziente, ma di supportarlo fino al momento in cui non sia in grado di fare a meno della ventilazione forzata.

Si tratta di una tecnica che tuttavia non è priva di controindicazioni; i soggetti sottoposti a ventilazione forzata sono esposti a rischi di vario tipo: collasso, danneggiamento polmonare, infezioni, i rischi associati alla sedazione e in casi estremi alla condizione di dipendenza dal supporto ventilatorio.

È fondamentale, dunque, non prolungare il trattamento oltre il tempo necessario al fine di ridurre l'impatto degli eventi avversi, al tempo stesso è importante non estubare precocemente il paziente per evitare il fallimento del tentativo di estubazione, condizione che può portare la probabilità di morte associata alla ventilazione forzata a crescere significativamente.

L'interesse nei confronti di questa tematica ha subito un incremento importante negli ultimi anni e in particolare a partire dal 2020 si è assistito ad un ulteriore picco che può essere con buona sicurezza associato alla situazione di pandemia legata al virus Sars-Cov-2, in cui il ricorso alle terapie intensive è un fatto drammaticamente noto.

Con il presente progetto di tesi si vuole cercare di proporre un sistema di auto alla decisione con l'obbiettivo di assistere il personale medico nell'individuazione del momento ottimale di svezzamento dal ventilatore meccanico.

Il presente progetto è una continuazione del lavoro effettuato in precedenza con i medesimi obiettivi e da cui si è ottenuto il dataset e le prime valutazioni sui classificatori.

I dati utilizzati nel presente progetto sono stati estratti dal database MIMIC-III (Medical Information Mart for Intensive Care).

Il primo obiettivo è stata la valutazione della stabilità del classificatore LightGBM testato nel lavoro precedente, al fine di effettuare tale valutazione sono state estratte quattro ulteriori coppie di Training Set e Test Set a partire dal medesimo dataset; ciò ha consentito di valutare il medesimo modello di classificatore allenato su dati differenti e di asserirne la stabilità. Tuttavia, il classificatore ha mostrato la medesima criticità già individuata in precedenza: il LightGBM soffre di una forte tendenza all'overfitting, problema che non è risultato superabile modificando i parametri e che può essere ricondotto alla scarsa dimensionalità dei dati.

Si è quindi proseguito con il test di nuovi classificatori, rivolgendosi al mondo del Deep Learning. Sono state costruite e valutate quattro strutture di Deep Neural Network che differiscono per il numero di hidden layer. Visti i risultati incoraggianti si è scelto di proseguire con una fase di ottimizzazione su ciascuna delle quattro tipologie di strutture cercando di superare le problematiche di instabilità con una funzione di fitness robusta; Tuttavia ciò non è stato reso possibile a causa delle risorse computazionali non adeguate e pertanto si è proseguito con l'ottimizzazione della struttura che nella prima fase di valutazione aveva ottenuto prestazioni migliori e maggiormente equilibrate e con una semplificazione della funzione di fitness che risulta così meno robusta.

I risultati della fase di ottimizzazione hanno permesso di effettuare valutazioni sia in termini di prestazioni sia per quanto riguarda le variabili selezionate e hanno portato ad ipotizzare nuove strade di combinazioni di reti per ottenere miglioramenti.

Sommario

ABSTRACT 1
1. INTRODUZIONE 4
1.0 Introduzione allo svezzamento dal ventilatore polmonare 4
1.1 Stato dell'arte: protocolli di svezzamento 6
1.2 Scopo del progetto di tesi 7
1.3 Panoramica del progetto di tesi: passi fondamentali 7
2. DATASET E SUDDIVISIONE DEL CONSTRUCTION SET 9
2.0 Database MIMIC III 9
2.1 Costruzione di quattro coppie di Training Set – Test Set 85 features 11
2.2 Costruzione di quattro coppie di Training Set – Test Set 58 features 12
3. LIGHTGBM (Performance, features e stabilità) 13
3.0 Light Gradient Boosting Machine 13
3.1 Apprendimento, Validazione e Stabilità del classificatore 15
3.3 Features importance 21
4. DEEP NEURAL NETWORKS 23
4.0 Dal Perceptron alle Deep Neural Networks 23
4.1 Implementazione delle DNN 25
4.2 DNN: Features and Structure Selection 27
4.2.0 Parametri dell'algoritmo genetico nel presente progetto e implementazione definitiva *29*
4.2.1 Run dell'Algoritmo Genetico *35*
4.2.2 Prestazioni delle reti costruite a partire dalle soluzioni degli Algoritmi Genetici *38*
4.2.3 Analisi degli errori *39*
4.2.4 Decisioni in merito al majority voting *43*
4.3 DNN: valutazioni di strutture di maggior complessità 47
5. DISCUSSIONE E CONCLUSIONI 49
APPENDICE A 54
Composizioni dei Majority voting 54
Majority Voting sulle reti ottenute dal GA con comorbidità 54
Majority Voting sulle reti ottenute dal GA con comorbidità sostituite 55
APPENDICE B 56
Variabili presenti nel Dataset 56
BIBLIOGRAFIA 59

1. INTRODUZIONE

1.0 Introduzione allo svezzamento dal ventilatore polmonare

La ventilazione meccanica è una tecnica clinica che consente lo spostamento di gas da e verso i polmoni tramite l'utilizzo di un dispositivo esterno collegato al paziente. Diversi sono gli obbiettivi per cui viene utilizzata questa tecnica, può essere infatti impiegata per ridurre o sostituire lo sforzo respiratorio, mantenere lo scambio gassoso, diminuire il consumo di ossigeno ottenere espansione polmonare e in numerosi altri casi. [1]
La necessità di utilizzare un sistema di ventilazione meccanica forzata è legata a condizioni di insufficienza respiratoria; condizione che secondo la NHLB (*National Heart, Lung, and Blood Institute)* si sviluppa quando i polmoni non sono in grado di garantire il corretto apporto di ossigeno al sangue, producendo un aumento di CO_2 che porta a danneggiamento di tessuti e organi. [2]
La NHLB riporta tra le cause di insufficienza respiratoria la sindrome da stress respiratorio acuto, infarto, asma, condizione di ostruzione cronica delle vie aree e altre malattie polmonari, arresti cardiaci e condizione di overdose. [3]
Anche possibili operazioni chirurgiche che richiedano l'uso di anestesia generale possono necessitare di supporto postoperatorio da parte del ventilatore meccanico.
La ventilazione meccanica forzata non è priva di rischi, per tanto è estremamente importante non sottoporre i pazienti a trattamenti prolungati oltre il necessario; hanno particolare importanza nell'utilizzo del ventilatore meccanico le infezioni, i collassi polmonari, il danneggiamento polmonare a causa della pressione insufflata, motivo per il quale occorre mantenere la pressione al minimo necessario; ulteriori fattori di rischio sono gli effetti collaterali della sedazione, che possono portare a condizioni di delirio anche una volta cessato l'utilizzo di farmaci sedanti e infine talvolta l'utilizzo del ventilatore meccanico può portare all'impossibilità di interruzione del supporto ventilatorio. [4]
La ventilazione meccanica non risolve i problemi che ne causano l'utilizzo, ma aiuta il soggetto fino al miglioramento delle sue condizioni.
Il processo di svezzamento dal ventilatore meccanico risulta, per quando detto sopra, di fondamentale importanza; riguarda l'intero processo di liberazione del paziente dalla ventilazione forzata inclusi anche tutti gli aspetti della terapia intensiva. Esso viene effettuato in 6 step fondamentali (*Figura 1*). [5]

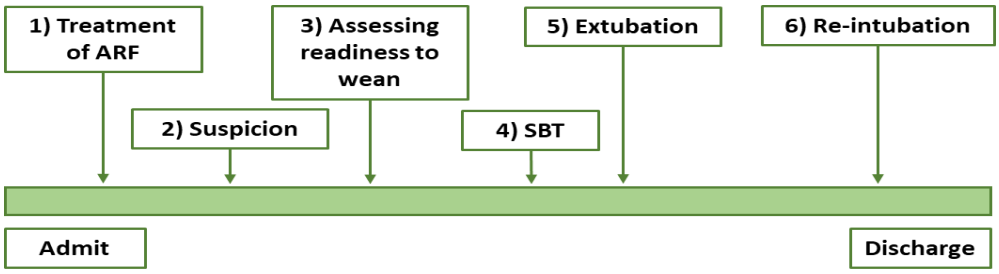


Figura 1 ***Fasi dello svezzamento da ventilazione meccanica forzata***
Fase 1 *Trattamento dell'insufficienza respiratoria acuta.* **Fase 2** *Sospetto di possibile svezzamento.* **Fase 3** *Valutazione della condizione favorevole di svezzamento.* **Fase 4** *Prova di Spontaneous Breathing Trial.* **Fase 5** *Estubazione.* **Fase 6** *Re-intubazione se necessaria. [5]*

Lo Step 1 prevede il trattamento dell'insufficienza respiratoria.
Lo Step 2 inizia quando il clinico valuta una ragionevole probabilità di estubazione con successo.
Lo Step 3 ha inizio nel momento in cui il clinico inizia a valutare tramite test giornalieri l'effettiva prontezza all'estubazione. Lo Step 3 termina nel momento in cui i risultati dei test giornalieri

indicano una probabilità abbastanza elevata da poter procedere con la prova di Spontaneous Breathing Trial (SBT).
Nella maggior parte degli studi il fallimento dell'estubazione è individuato come il fallimento della procedura di SBT o la necessità di re-intubazione entro le 48h successive al tentativo dell'estubazione. Spesso la valutazione iniziale di prontezza all'estubazione viene effettuata tramite la valutazione del parametro di Rapid Shallow Breathing Index (RSBI) calcolato come frequenza respiratioria/ Volume (Tidal). Questo parametro secondo quanto riportato nello studio di '*Weaning from mechanical ventilation'* (Boles et all.) [5] è in grado di predire un SBT eseguito con successo con una sensibilità del 97% e specificità del 65%. Se quindi il valore RSBI risulta essere favorevolmente predittivo all'SBT allora viene effettuato quest'ultimo. Gli studi dimostrano che il fallimento della prova SBT avviene nei primi 20 minuti; quindi, al fine di valutare un possibile esito positivo risulta essere indifferente effettuare la prova per 30 o 120 minuti. [5]
La review di Boles et al. '*Weaning from mechanical ventilation'* [5] mette in evidenza che il 13% dei soggetti che hanno ottenuto esito positivo al test SBT necessita di successiva re-intubazione, contro il 40% laddove il test non viene effettuato.
Da quanto detto finora la questione dello svezzamento da ventilatore risulta quindi essere critica; il 40-50% del tempo trascorso sotto ventilazione forzata riguarda il processo di svezzamento. Ogni giorno di terapia intensiva trascorso porta ad un aumento della mortalità, oltre ad avere un peso ingente in termini di costi sul sistema sanitario. [6]
La probabilità di morte associata a una ventilazione forzata a seguito di un'estubazione fallita riportano alcuni studi essere compresa tra il 25% e il 50%. [7]
Inoltre, non solo il fallimento dell'estubazione induce a gravi rischi, ma al tempo stesso anche un'estubazione eseguita precocemente può portare a conseguenze gravi come l'insufficienza cardio respiratoria, dovuta ad un aumento eccessivo richiesto alle vie aeree in assenza del supporto da parte del ventilatore. [8]
Riuscire dunque a predire il momento ottimale in cui svezzare il paziente risulta essere quindi di importanza cruciale.

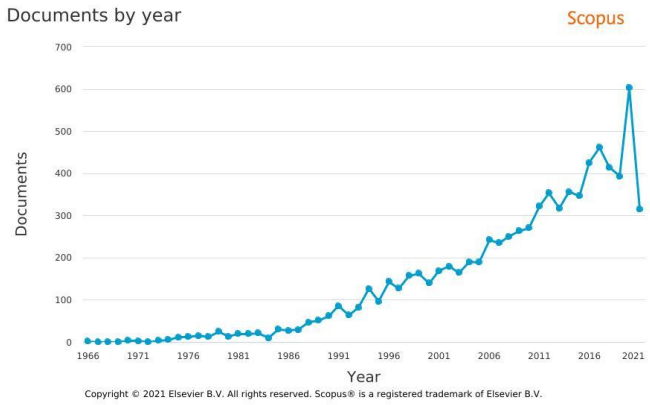


Figura 2 ***Documenti sul problema dell'estubazione negli anni***
In figura vengono riportati il numero di documenti sul tema dell'estubazione per ogni anno. Query utilizzata ai fini della ricerca (TITLE-ABS-KEY (mechanical AND ventilation) AND TITLE-ABS-KEY (extubation OR weaning))

L'interesse nei confronti della tematica dello svezzamento da ventilatore forzato ha visto una crescita importante negli ultimi anni e in particolare da inizio 2020 con l'aumento di ingressi in terapia intensiva legato al virus Sars-Cov-2.
Ricercando sul database Scopus i documenti riguardanti il problema dell'estubazione e dello svezzamento da ventilatore polmonare si nota come l'interesse a partire dagli anni 60' abbia mantenuto un andamento di crescita generale con un aumento significativo nell'ultimo anno. (*Figura2*)

1.1 Stato dell'arte: protocolli di svezzamento

Il processo di svezzamento risulta tutt'ora una questione critica, le incertezze riguardano la diversa sensibilità da parte dei clinici nella valutazione dell'effettiva prontezza del paziente allo svezzamento. Esistono linee guida, checklist e protocolli che forniscono un supporto alla decisione al fine di ridurre quella che è la possibile variabilità tra diversi operatori; le linee guida sono di carattere generale, permettono ai clinici di basarsi essenzialmente sulle proprie capacità di giudizio. Le checklist hanno un'organizzazione sistematica di quelle che sono le azioni o i criteri da utilizzare. [9] I protocolli sono invece una serie di 'regole' che il clinico deve seguire; diversi studi hanno dimostrato come l'utilizzo di protocolli in questa fase consenta di uniformare i risultati.
Occorre in primis dare una definizione di protocollo, quest'ultima è descritta in maniera efficace dall'articolo di Borges et.all *'Mechanical ventilation weaning protocol improves medical adherenceand results'* dove il protocollo viene definito come:

"a detailed plan of a scientific or medical experiment, treatment, or procedure"

Si tratta dunque di un piano dettagliato e specifico in grado di fornire una linea guida per trattamenti e procedure. [10]
Il fine di un protocollo di estubazione è quindi quello di garantire uno svezzamento efficace e sicuro per il paziente, mirando a ridurre interventi non necessari o che possano mettere a rischio la salute già di per sé precaria di quest'ultimo.
Fino ai primi anni del XXI secolo poche strutture ospedaliere includevano nella propria pratica clinica l'uso di protocolli; diversi studi hanno evidenziato come l'utilizzo di protocolli consenta di avere un maggior numero di successi nell'estubazione, riducendo i giorni di ventilazione e i tempi di svezzamento. Inoltre, l'adozione di procedure standardizzate sembrerebbe consentire anche una riduzione dei costi, permettendo a figure professionali non 'medici' di eseguire l'estubazione. È stato infatti dimostrato che i risultati ottenuti da personale sanitario non medico tramite il supporto di un protocollo siano migliori di quelli ottenuti da personale medico senza l'impiego di quest'ultimo. [11]
Il processo di svezzamento, come già descritto nei paragrafi precedenti, si sviluppa in sei fasi, uno degli step importanti che vanno a definire la possibilità di eseguire l'estubazione riguarda il sospetto da parte del medico di un miglioramento delle condizioni in grado di ridurre e successivamente rimuovere il supporto polmonare. Al fine di aiutare nella decisione sono state stilate dall'American Association for Respiratory Care una serie di raccomandazioni che permettono di valutare una potenziale condizione favorevole alla cessazione del supporto meccanico alla ventilazione. [11]
Numerosi sono gli studi in letteratura che hanno valutato l'impatto di indici e score sull'esito del processo di svezzamento, ma non hanno condotto a risultati in grado di garantirne l'ingresso nella comune pratica clinica.
Esistono inoltre anche sistemi automatici di svezzamento, che in linea teorica dovrebbero garantire una miglior efficacia, ma di fatto sono una versione computerizzata dei classici protocolli di estubazione. Questi tipi di sistemi sembrerebbero garantire risultati incoraggianti, ma il loro uso effettivo è ancora modesto. [8]

1.2 Scopo del progetto di tesi

Il presente progetto di tesi mira a indagare le possibili soluzioni, a partire da quanto già individuato in letteratura, al fine di fornire aiuto nella decisione da parte del clinico nell'individuazione della condizione ottimale per procedere con lo svezzamento dal ventilatore polmonare.
Il lavoro effettuato vuole proporre un sistema di aiuto alla decisione, il cui obiettivo è quello di fornire supporto al clinico nella verifica della condizione di possibilità di estubazione e che possa inserirsi nel protocollo di valutazione di predisposizione allo svezzamento da parte del paziente, andando quindi ad affiancare i test abitualmente in uso nella pratica clinica.
Occorre sempre ricordare che i sistemi di questo tipo non mirano alla sostituzione del processo decisionale da parte del clinico, ma bensì hanno lo scopo di supportare la decisione, risultando di particolare aiuto laddove le condizioni non siano sufficientemente chiare e tali da garantire una ragionevole certezza sul possibile esito del processo in questione.
Il data science framework si inserisce in questo proposito fornendo numerosi strumenti in grado di valutare le condizioni dei pazienti e la loro predisposizione allo svezzamento, per tanto questo progetto di tesi mira ad indagare metodi di classificazione e di features selection che permettano l'individuazione di quelle condizioni tali per cui l'esito dell'estubazione possa essere precocemente riconosciuto essere positivo e/o negativo.

1.3 Panoramica del progetto di tesi: passi fondamentali

Il presente progetto di tesi nasce in continuità al lavoro precedente, che ha portato alla definizione di un Dataset costruito a partire dai dati messi a disposizione dal database MIMIC-III (Medical Information Mart for Intensive Care) e dai classificatori valutati su questi dati: LigthGBM e reti neurali.
Il primo passo fondamentale è stato la presa di consapevolezza sulla tematica dell'estubazione, valutando gli studi già individuati nel lavoro precedente ed effettuando un aggiornamento della bibliografia, andando quindi a indagare i risultati ottenuti nell'ambito della Data Science e del Machine Learning relativi al problema dello svezzamento dal 2019 al 2021.
Ottenuta consapevolezza della tematica, si è proseguito con la valutazione della stabilità e delle prestazioni ottenute con i classificatori utilizzati in precedenza; come primo passo si sono quindi estratti quattro ulteriori Training e Test Set, con il fine di valutare se le prestazioni ottenute con i classificatori fossero legate alla composizione del set di allenamento e se quindi quest'ultimo fosse sufficientemente rappresentativo dei dati a disposizione.
Per ciascuna delle 4 coppie di set si è andati a effettuare nuovamente l'ottimizzazione e l'allenamento di un classificatore LightGBM: i risultati sui 4 set vanno a confermare quelle che sono le problematiche già individuate nel lavoro precedente per questo modello di classificatore. Nonostante l'utilizzo di Training Set e Test Set diversi e la modifica dei range in cui effettuare l'ottimizzazione, tutti i classificatori LightGBM sono affetti da overfitting che porta ad una classificazione perfetta sul Training Set, ma ad un deterioramento delle prestazioni sul Test e Validation Set.
Data l'impossibilità di migliorare ulteriormente le prestazioni del classificatore LightGBM, il passo successivo è stato quello di andare a lavorare sulle reti neurali. Prendendo spunto, dal lavoro di tesi precedente si è scelto di implementare delle Deep Neural Network (DNN), rivolgendosi dunque

all'ambito del Deep Learning. Si sono impostate quattro tipi di DNN, che differiscono tra loro per il numero di neuroni in hidden layer successivi. A seguito di una serie di prove effettuate sul totale delle features degli elementi dei tre set (Training, Test e Validation Set) e avendo ottenuto risultati promettenti, si è quindi proseguito andando a impostare un algoritmo di features selection (GA). Si è quindi effettuato il run di due algoritmi genetici, il primo consente di ottimizzare il numero e il tipo di features tra quelle a disposizione e conseguentemente anche la rete in quanto la struttura è costruita da un numero variabile di hidden layer in cui ciascuno presenta una differenza di quattro neuroni rispetto al precedente fino all'ultimo layer di output e dove il layer di input viene codificato direttamente dalla soluzione del metodo di ottimizzazione.
Il secondo algoritmo è stato invece impostato in modo da lavorare nelle medesime condizioni del precedente, ma su una coppia Training e Test set in cui le comorbidità sono state sostituite dalla codifica decimale associata alla stringa di 0 e 1 che individua le comorbidità e da una seconda variabile contente il numero totale delle comorbidità per ogni elemento.
I due algoritmi di ottimizzazione hanno fornito rispettivamente 60 e 50 soluzioni che hanno permesso di valutare le prestazioni delle reti e le features selezionate.
Dal momento che le prestazioni non sono risultate, a seguito dell'ottimizzazione, soddisfacenti si è considerata la possibilità di combinare gli output di queste reti andando ad effettuare una majortiy voting su un numero dispari di reti. I Majority sono stati impostati secondo due criteri, in un primo caso si è sfruttato un indice di similarità calcolato sui vettori degli errori per individuare le reti con minor similarità per poi includere anche quelle più simili, in un secondo caso invece si sono valutate le intersezioni dei vettori di errori.
I Majority voting hanno permesso di avvicinare i valori dei parametri di sensibilità e specificità che nel caso delle singole reti mostravano andamenti speculari, senza però portare ad un incremento effettivo di questi.
In ultimo, si sono valutate delle ulteriori strutture di DNN, con un maggior numero di neuroni e hidden layer, che tuttavia hanno mostrato l'inadeguatezza della piattaforma Matlab nella gestione di reti di tale complessità, non garantendo stabilità nelle prestazioni e pertanto queste ultime reti sono state scartate.

2. DATASET E SUDDIVISIONE DEL CONSTRUCTION SET

Il presente progetto di tesi è stato svolto tramite Matlab 2020b, piattaforma di programmazione e calcolo numerico di Mathworks e Google Colab strumento gratuito della suite Google che consente di sviluppare progetti in linguaggio Python direttamente sul browser.

2.0 Database MIMIC III

I dati su cui si è lavorato nel presente progetto di tesi sono stati estratti dal database MIMIC-III (Medical Information Mart for Intensive Care').
MIMIC-III è un vasto database costruito su dati, ottenuti da un singolo centro, che includono segni vitali, trattamenti, osservazioni, misure, note, codici diagnostici, report di immagini mediche e molti altri. [12]I dati sono relativi a pazienti ammessi al Beth Israel Deaconess Medical Center in Boston, Massachusetts tra il 2001 e il 2012. [13]
Questi dati sono stati analizzati e organizzati nel progetto di tesi precedente, il cui primo step è stata l'individuazione dei soggetti che risultavano sottoposti per un periodo a ventilazione forzata. Sono stati poi applicati criteri di inclusione ed esclusione che hanno ridotto ulteriormente la dimensionalità dei dati. Sono stati quindi inclusi soggetti di età compresa tra i 18 e gli 89 anni, sono stati esclusi i pazienti deceduti prima del tentativo di estubazione, i soggetti che hanno subito un'estubazione non pianificata e che hanno subito in un'intubazione con durata inferiore alle 24 h. Per ogni paziente è stato considerato solo il primo evento di ventilazione forzata e il primo accesso in terapia intensiva. Una volta applicati i criteri di inclusione ed esclusione ogni elemento è stato associato alla corrispondente classe EF (Extubation Failure) o ES (Extubation Success); la discriminazione tra le classi è valutata sull'evento di re-intubazione entro le 48h dal tentativo di estubazione.
Sono quindi state estratte inizialmente 91 variabili che spaziavano da dati demografici a scale di gravità e ancora parametri vitali e indicatori tipici delle analisi ematiche ed emogasanalisi.
Il limite principale era la forte presenza di missing values su alcune di queste, per tanto il database è stato ripulito andando a eliminare quelle variabili che presentavano un elevato numero elementi con missing values ed è stata eseguita un'imputazione dei valori mancanti laddove la percentuale di elementi privi della variabile è stato considerato accettabile.
Al termine di queste operazioni di cleaning ciascun elemento del dataset così ottenuto risulta essere costituito da 85 variabili.
L'intero set di dati è quindi stato suddiviso: 150 elementi sono stati inseriti nel Validation set di cui 50 appartenenti alla classe ES imputata, 50 alla classe ES non imputata e 25 soggetti da ciascuna delle due sottoclassi EF (imputata/non imputata). Il Validation set è stato utilizzato per la validazione anche nel presente progetto di tesi, 13673 elementi hanno invece costituito il Construction Set. Su quest'ultimo è stata effettuata una clusterizzazione sulle quattro sottoclassi (EF/ES imputati/non imputati), in grado di garantire variabilità simile all'interno dei singoli cluster per una stessa classe.
Tramite estrazione proporzionale dai cluster è stato quindi creato la prima coppia Training Set/Test set, il primo set risulta costituito da 1000 elementi per ciascuna delle classi EF/ES imputate e 150 dagli elementi appartenenti alle classi non imputate, mentre il secondo è specularmente costituito da tutti gli elementi non estratti e inseriti nel primo set.
Il presente progetto trova il suo incipit a partire da questi dati, si è lavorato quindi su un dataset costituito da un Construction set (unione di Training Set e Test Set) di 13673 elementi, ciascuno con

le 85 variabili e le informazioni relative alla classe, all'avvenuta imputazione e al cluster di appartenenza e dal Validation set costituito da 150 elementi anch'essi con 85 features e le informazioni relative alla classe e all'imputazione.
Entrando quindi nel dettaglio della costituzione del dataset, le variabili relative a ciascun elemento sono così organizzate:

- 1-3 **Variabili Demografiche**: *'Age', 'Gender', 'Weight_loss'*
- 4-5 **Punteggi di gravità**: *SOFA score*, e *Glasgow Coma Scale* (GCS)
- 6-7 **Somministrazione di terapia**: *'Sedation_days'* e *'Vasopressor'*
- 8-36 **Comorbidità** *'Chronic_cardiac_insufficiency', 'Arrhytmia', 'Valvular_disease', 'Pulmonary_heart_disease', 'Peripheral_vascular', 'Hypertension', 'Stroke', 'Other_neurological', 'Chronic_Obstructive_Pulmonary_Disease', 'Diabetes_uncomplicated', 'Diabetes_complicated', 'Hypothyroidism', 'Chronic_kidney_disease', 'Liver_disease', 'Peptic_ulcer', 'AIDS', 'Lymphoma', 'Metastatic_tumors', 'Solid_tumor', 'Rheumatoid_arthritis', 'Coagulopathy', 'Obesity', 'Blood_loss_anemia', 'Deficiency_anemia', 'Alcohol_abuse', 'Drug_abuse_psychoses', 'Depression', 'Electrolyte_and_fluid_disorder','Psychoses'*
- 37-42 **Parametri ventilatori**: *'MV_duration_hours', 'Tidal_volume', 'Minute_ventilation', 'Mean_Airway_Pressure', 'Peak_inspiratory_pressure', 'mean_Heart_rate'*
- 43-62 **Parametri vitali**: *'max_Heart_rate' 'min_Heart_rate' 'mean_Respiratory_rate', 'max_Respiratory_rate','min_Respiratory_rate','mean_Temperature', 'max_Temperature', 'min_Temperature', 'mean_systolic_blood_pressure', 'max_systolic_blood_pressure', 'min_systolic_blood_pressure', 'mean_diastolic_blood_pressure', 'max_diastolic_blood_pressure', 'min_diastolic_blood_pressure', 'mean_mean_arterial_pressure', 'max_mean_arterial_pressure', 'min_mean_arterial_pressure', 'mean_SpO2', 'max_SpO2' , 'min_SpO2'*
- 63-85 **Analisi ematiche ed emogasanalisi**: *'White_blood_cells', 'Hemoglobin', 'Platelet', 'Sodium'. 'Potassium', 'Calcium', 'Chloride', 'Lactate', 'Glucose', 'Creatinine', 'Blood_Urea_Nitrogen', 'Bilirubin', 'Albumin', 'Prothrombin_Time', 'activated_Partial_Thromboplastin_Time', 'International_Normalized_Ratio', 'GPT', 'GOT', 'Central_vein_oxygenated', 'arterial_pH', 'PaCO2', 'PaO2', 'Base_excess'*
- 86 **Classe**: Extubation Failure *EF* (0), Estubation Success *ES* (1)
- 87 **Imputazione**: 1 pazienti ES con valori sostituiti, 2 pazienti EF con valori sostituiti, 3 pazienti ES senza missing values, 4 pazienti EF senza missing values
- 88 **Cluster** (assente nel caso del Validation Set)

La natura delle variabili è varia, il dataset è composto da variabili di tipo discreto, ne sono d'esempio la variabile relativa all'età del paziente ammesso all'ICU o gli score relativi alle scale di gravità (SOFA,GCS), sono presenti variabili di tipo continuo come ad esempio i diversi valori dei parametri ventilatori e i parametri relativi ad analisi ematiche e emogasanalisi, infine uno spazio importante è occupato dalle variabili di tipo binario; queste ultime riguardano le comorbidità di cui viene indicata attraverso la codifica 0/1 la presenza o assenza della malattia a cui si aggiunge la variabile relativa al genere e all'utilizzo di Vasopressori.

In un secondo momento sono state ottenute le coppie di Training e Test Set e la versione del Validation Set in cui le variabili di comorbidità sono state sostituite da due variabili:

- **Comorbidità_decimale**: variabile che corrisponde alla codifica binaria della sequenza di 0 e 1 indicante la presenza o non presenza della comorbidità
- **Totale_comorbidità**: variabile indicante il numero totale delle comorbidità associate a ciascun elemento

2.1 Costruzione di quattro coppie di Training Set – Test Set 85 features

A partire dalle clusterizzazioni presenti nel primo Construction set, costruito secondo quanto indicato nel paragrafo precedente, si è sfruttata l'informazione relativa ai cluster al fine di ottenere quattro coppie di Training set e Test set.
Si è andati quindi a ricostruire il Construction set, dal Training e Test set a disposizione, il quale è stato poi suddiviso nelle classi EF imputato/non imputato ed ES imputato/non imputato.
Sfruttando l'informazione relativa alla clusterizzazione effettuata su queste quattro sottoclassi si è andati ad estrarre proporzionalmente da ciascuna gli elementi da inserire nel Training set (*Figura 3*).
Circa 1000 elementi sono stati estratti dalle sottoclassi imputate (EF ed ES), mentre da ciascuna sottoclasse non imputata (EF ed ES) si è applicata un'estrazione di circa 150 elementi ciascuna. Gli elementi che non fanno parte del Training set hanno costituito il Test set.
Nel presente lavoro si è scelto di estrarre almeno un elemento da ciascuno dei cluster, al fine di garantire la rappresentatività a livello del Training set anche degli elementi appartenenti ai cluster meno numerosi. Tale scelta ha determinato un eccesso di due elementi rispetto al previsto, estratti dalle sottoclassi ES imputata ed ES non imputata; i Training set sono quindi costituiti da 2302 pazienti.
Il procedimento di costruzione del Training set così come descritto è stato eseguito per un totale di quattro volte, ottenendo così quattro nuove coppie di Training set e Test set in aggiunta a quella già precedentemente estratta nel lavoro di tesi precedente.

Figura 3 ***Schema di estrazione del Training e Test Set***
In figura sono schematizzati i passaggi che hanno consentito a partire dal dataset costituito da Training Set, Test Set e Validation Set.

I Training set così estratti risultano quindi avere un equilibrio in termini di numerosità tra le due classi.
La scelta di effettuare quattro ulteriori estrazioni di Training set e Test set in aggiunta a quello già in possesso è giustificata dalla necessità di valutare la stabilità nelle performance dei classificatori; di garantire quindi che le performance non siano vincolate ad un determinato Training set e che quindi determinate performance positive o negative non siano legate ad una particolare composizione dei dati di allenamento.
Le coppie di Training e Test set a disposizioni sono in totale cinque, il Validation set è invece comune a tutte.

2.2 Costruzione di quattro coppie di Training Set – Test Set 58 features

Le cinque coppie Training e Test set a disposizione e il Validation set sono state sottoposte ad una codifica delle variabili relative alle comorbidità che ne prevede la sostituzione con due nuove variabili; La prima variabile 'Comorbidità_decimale' è di fatto la conversione in numeri decimali della stringa relativa alle comorbidità (variabili 7-36), la seconda invece 'Totale_comorbidità' riporta il numero di comorbidità tototali a cui ogni elemento o paziente è soggetto.

3. LIGHTGBM (Performance, features e stabilità)

3.0 Light Gradient Boosting Machine

Il primo classificatore impiegato in questo lavoro è il LightGBM, si tratta di una tecnica di apprendimento supervisionato appartenente al panorama del Machine Learning.
Il classificatore LightGBM è un nuovo modello di gradient boosting, rilasciato da Microsoft il 1° ottobre 2016. Spicca per l'elevata efficienza e la scalabilità ed è inoltre in grado di supportare diversi algoritmi inclusi GBDT (Gradient Boosting Decision Tree), GBRT (Gradient Boosting Regression Tree), GBM (Gradient Boosting Machine) e MART (Multiple Additive Regression Tree). [14]
Le evidenze indicano che il LightGBM sia più veloce delle implementazioni classiche di Gradient Boosting fino a 20 volte ottenendo la stessa accuratezza e secondo quanto riportato da Microsoft permette inoltre la gestione di dati su larga scala.
Si tratta di un modello costituito da un ensemble di decision tree allenati in sequenza. Ad ogni iterazione i decision tree apprendono andando a adattarsi a un gradiente negativo anche chiamato errore residuo. [15]
Questo tipo di algoritmo risulta diverso rispetto ai modelli di Decision Tree più popolari nel modo in cui identifica il miglior split-point, collo di bottiglia che determina il tempo consumato nel processo di apprendimento.
L'algoritmo pre-sorted è il metodo più popolare per l'individuazione dei punti di split il quale va a indicare tutti i possibili punti di split sui valori di features ordinate in precedente. Sebbene questo algoritmo sia efficace nell'individuazione dei punti ottimali di split non lo è altrettanto dal punto di vista della velocità di training e del consumo di memoria.
Un secondo algoritmo proposto, utilizzato nel LightGBM, è quello basato sugli istogrammi, in cui anziché andare a individuare i punti di split sui valori di features ordinati, queste ultime sono raggruppate in bins utilizzati per la costruzione di istogrammi sulle features in fase di training su cui va a identificare il miglior split. Il costo computazionale è definito come ***#data x #features*** per la costruzione dell'istogramma e ***#bin x #features*** per l'individuazione del punto di split ed essendo il numero di bin generalmente inferiore al numero di dati questo consente di velocizzare la procedura. [15]
La maggior velocità è inoltre garantita da due tecniche, la prima chiamata Gradient-Based One-Side Sampling (GOSS) e la seconda nota come Exclusive Feature Bundling (EFB).
La prima tecnica gestisce il problema di non avere pesi nativi dei campioni di dati, questo problema viene superato grazie al fatto che le tecniche di Gradient Boosting forniscono un gradiente per ogni istanza di dati; quindi, grazie a questo è possibile avere informazioni utili sui dati; se un'istanza ha un gradiente piccolo significa che l'errore di addestramento lo è altrettanto e quindi il classificatore è già ben allenato. L'idea è quindi quella di andare a scartare quelle istanze con gradienti piccoli, ma questo porterebbe a una ri-organizzaizione nella distribuzione dei dati. La tecnica GOOS si inserisce in questo problema mantenendo le istanze con grandi gradienti ed eseguendo il campionamento casuale sulle istanze con gradienti piccoli. [15]
La seconda tecnica Exclusive Feature Bundling consente invece di ridurre il numero di features, sfruttando la scarsa dimensionalità dello spazio delle features; viene consentita la riduzione di queste ultime con perdite minime, in uno spazio di questo tipo spesso le features sono mutuamente esclusive. [15]

Ulteriore differenza che caratterizza il LightGBM dal più famoso XGBoost è la strategia implementata per l'accrescimento degli alberi decisionali, nel primo caso viene utilizzata una tecnica definita leaf-wise in cui l'albero viene accresciuto dalla foglia che porta il maggior grado di informazione, tecnica che consente di ridurre ulteriormente la perdita ma che può condurre ad overfitting, nel secondo caso viene invece utilizzata una tecnica level-wise. In figura (*Figura 4*) viene riportata un'efficace sintesi visiva delle differenze tra le due tecniche a livello di alberi decisionali. [16] [17]

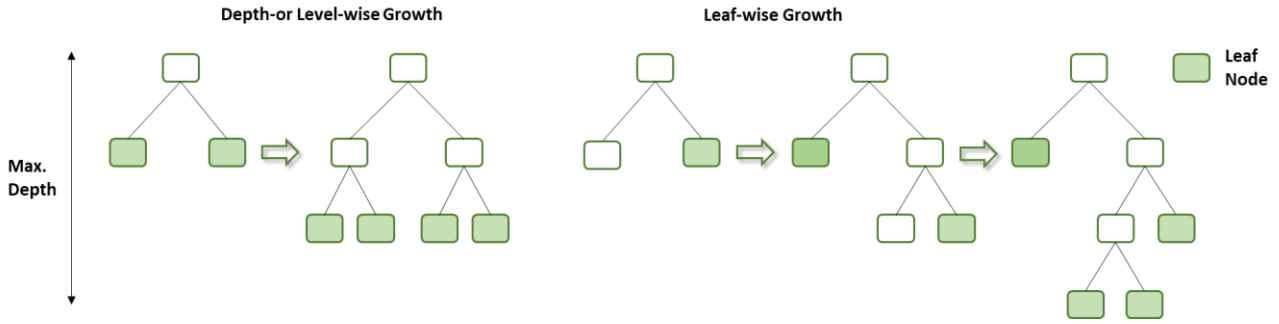


Figura 4 ***Strategie Level-wise e Leaf-wise a confronto***
Nella rappresentazione a sinistra si può comprendere come avviene la crescita in un modello level-wise, modello in cui le foglie vengono aggiunte ai nuovi livelli a differenza del caso Leaf-wise (alberi a destra) dove invece la crescita avviene foglia per foglia portando ad avere alberi decisamente più profondi rispetto al primo caso, seppur a parità di foglie. [18]

Il training tramite strategia level-wise può essere considerato un modo di regolarizzare il processo di training. Il training effettuato tramite strategia leaf-wise è certamente più soggetto a overfitting, con una strategia di questo tipo porta alla costruzione di alberi più profondi a parità del numero di foglie. [18]

Il LigthGBM viene utilizzato per diverse applicazioni può infatti essere utilizzato per affrontare problemi di regressione, per effettuare ranking per problemi di classificazione multi-classe o binaria, quest'ultimo caso è quello di interesse per questo progetto. L'implementazione di un algoritmo di questo tipo richiede che vengano specificati determinati parametri in fase di allenamento; come detto in precedenza i classificatori LightGBM adottano un approccio definito leaf-wise, al fine di ottenere risultati ottimali è necessario impostare determinati parametri che secondo quando riporta la documentazione Microsoft sono:

- *num_leaves* parametro che controlla la complessità dell'albero, teoricamente può essere settato come *num_leaves = 2^(max_depth)*, ma questa conversione può non risultare ottimale; pertanto, è indicato effettuare il tuning di questo parametro.
- *min_data_in_leaf* è un parametro importante per prevenire overfitting. Il suo valore ottimale dipende dalla dimensione del training set e dal num_leaves. Importare questo parametro troppo grande può però portare a underfitting.
- *max_depth* parametro che limita la profondità dell'albero.

Ulteriori parametri riguardano l'utilizzo dei thread per il running, determinando quindi le risorse computazionali allocate. Parametri importanti riguardano poi il numero di alberi da costruire, controllati dal parametro *num_iteration* e la velocità di allenamento che può essere monitorata tramite *l'early_stopping_rounds*, parametro che indica dopo quante iterazioni senza miglioramento andare a fermare l'allenamento del modello. Per ottenere una buona accuratezza viene inoltre

consigliato di utilizzare un basso *learning_rate* con un *num_iteration* elevato. Numerose sono le metriche disponibili per la valutazione del modello tra L1 e L2 loss, Classification error rate, AUC, Multi-class error rate, Average precision.

Il dataset con 85 features, così come descritto nel capitolo precedente è composto da variabili di diversa natura, sono presenti variabili discrete, continue e binarie o categoriche. Queste ultime vengono abitualmente rappresentate tramite codifica one-hot, ma questo approccio non risulta ottimale per classificatori di questa natura; pertanto, il LigthGBM secondo quanto riportato nella documentazione divide le variabili categoriche suddividento le categorie in due sottoinsiemi. Se una feature ha k categorie allora saranno possibili 2^(k-1) -1 partizioni possibili, occorre quindi indicare quali variabili categoriche sono presenti. [14]

3.1 Apprendimento, Validazione e Stabilità del classificatore

L'allenamento e la validazione del classificatore sono stati effettuati sfruttando la libreria lightgbm disponibile per il linguaggio Python tramite la piattaforma Google Colab.

Numerosi sono i parametri che permettono l'implementazione di un modello LightGBM ed è di fondamentale importanza effettuare il tuning di questi ultimi in modo da adattare il modello ai dati su cui effettuare la classificazione, solo in questo modo sarà infatti possibile ottenere prestazioni migliori. L'impostazione dei parametri è stata ripresa dal lavoro di tesi precedente in quanto uno degli obiettivi del presente progetto è quello di valutare la stabilità delle soluzioni già individuate; pertanto, riprendendo il lavoro precedente i parametri del classificatore sono stati definiti come descritto a seguire (*Tabella 1*). Per alcuni è possibile la definizione a priori sulla base del problema da affrontare, ad esempio volendo classificare gli elementi come ES (1) o EF (0) il parametro relativo all'obiettivo di classificazione verrà impostato come 'binary'. Non sono stati sottoposti a tuning i parametri relativi alla metrica per la valutazione dell'allenamento, tra quelle disponibili descritte precedentemente si è scelto di utilizzare la 'AUC' (Area Under Curve). Il metodo di boosting è stato imposto come 'goss' che per quanto descritto precedentemente differenzia il LightGBM dagli altri classificatori di tipo 'Decision Tree' e conseguentemente al metodo di Boosting anche i parametri '*top_rate*' e '*other_rate*' sono stati scelti; si noti che la somma di questi ultimi deve essere pari ad 1.

PARAMETRI NON SOTTOPOSTI A TUNING	
PARAMETRO	***VALORE***
objective	binary
metric	auc
is_unbalance	True
boosting	goss
num_trees	500
early_stopping_rounds	50
subsample	1
top_rate	0.6
other_rate	0.3

Tabella 1 ***Parametri non sottoposti a tuning***

Il parametro *subsample* è stato posto pari ad 1 in modo da fornire tutto il Training Set in fase di addestramento, così da cercare di controllare il fenomeno dell'overfitting, tanto più importante quanto minore è la dimensione del set di allenamento.

Il parametro *is_unbalance* è impostato a True, nel set si ha uno sbilanciamento tra le classi ES ed EF nel Test Set in quanto l'incidenza di estubazioni fallimentari è minore di quella delle estubazioni avvenute con successo.

Il Tuning effettivo riguarda 5 parametri i cui rispettivi range di esplorazione sono riportati in tabella (*Tabella 2*). I range di esplorazione consentono di individuare il modello più adatto al problema di

classificazione proposto. In un secondo momento è stato effettuato il tuning dei parametri andando ad agire sul range di esplorazione relativo al numero di foglie *num_leaves* e al minimo numero di dati nella foglia *min_data_in_leaf* così come riportato in *Tabella 3*.

PARAMETRI SOTTOPOSTI A TUNING	
PARAMETRO	***RANGE DI ESPLORAZIONE***
Learning rate	[0.01 , 0.1]
max_depth	[1 , 30]
num_leaves	[10 , 200]
min_data_in_leaf	[10 , 50]
feature fraction	[0.1 , 1]

Tabella 2 ***Parametri sottoposti a tuning***
In Tabella sono riportati i parametri sottoposti a tuning e i rispettivi range di esplorazione

ULTERIORI PROVE DI TUNING		
	PARAMETRO	***RANGE DI ESPLORAZIONE***
Tuning 2	*num_leaves*	[10 ,100]
Tuning 3	*min_data_in_leaf*	[60 , 100]
Tuning 4	*min_data_in_leaf*	[10 , 100]

Tabella 3 ***Ulteriori prove di tuning***
Sono stati effettuati tre ulteriori tentativi di tuning modificando per ciascuno il range del parametro indicato in tabella e lasciando i restanti come indicato in precedenza (Tabella 2)

Al fine di individuare il modello di classificatore migliore sono state valutate cento configurazioni dei parametri selezionati in maniera casuale. Ogni valutazione viene effettuata allenando il classificatore sul Training set e valutando le prestazioni sul Test set a partire dall'insieme di parametri selezionato. La funzione obiettivo che viene minimizzata tramite le operazioni di tuning è stata definita come f(x) = -1 × AUC sul Test set, tale ottimizzazione viene effettuata tramite la libreria Scikit-Optimize.

Uno dei primi obiettivi del presente progetto è stato la valutazione della stabilità delle soluzioni ottenute, pertanto si è costruito un modello di LightGBM su ciascuna delle quattro nuove coppie di Training set e Test set ottenute dal dataset con 85 features (2.1 Costruzione di quattro coppie di Training Set – Test Set) così che i risultati di classificazione, importanza delle variabili e parametri selezionati possano essere confrontati tra i nuovi modelli e rispetto a quello ottenuto in precedenza.

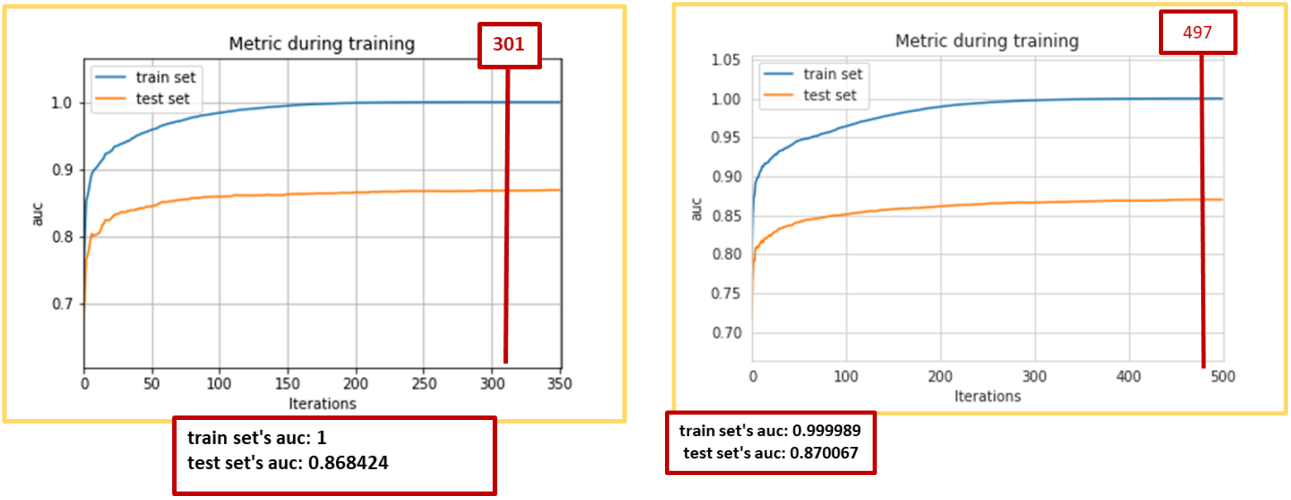
La fase di tuning è stata quindi eseguita sui modelli costruiti sulle quattro nuove coppie di Training set e Test set a cui è seguita una successiva fase di allenamento e valutazione a partire dai parametri selezionati, mentre il modello ottenuto sulla coppia Training set e Test set del lavoro di tesi precedente è rimasto invariato. Al termine della fase di tuning per ogni modello sono quindi stati identificati i parametri che consentono l'ottimizzazione, in tabella (*Tabella 4*) si riportano i valori dei parametri selezionati identificati dai modelli nel presente progetto e quelli ottenuti nel lavoro precedente.

Best parameters	*Mod.1*	*Mod.1*	*Mod.3*	*Mod.4*	*Mod.tesi prec.*
learning_rate	0,052	0,031	0,021	0,020	0,032
max_depth	7	9	8	9	14
num_leaves	161	191	99	188	131
min_data_in_leaf	50	43	28	50	48
feature_fraction	0,183	0,464	0,271	0,419	0,39

Tabella 4 ***Parametri selezionati a seguito della fase di tuning dai quattro nuovi classificatori e dal classificatore ottenuto nel lavoro precedente***
I classificatori hanno mostrato notevole variabilità nella selezione dei parametri.

A seguito dell'individuazione dei parametri al termine della fase di tuning viene quindi effettuato l'allenamento dei modelli.

Lo stato dell'allenamento viene controllato attraverso la valutazione della metrica AUC sia per il Training set che per il Test set. Sono possibili fino a 500 iterazioni, ma qualora la metrica presenti valori stabili e che quindi si sia raggiunto il plateau l'allenamento viene fermato prima del totale delle iterazioni possibili (*Figura 5*).



a) b)

Figura 5 ***Valutazione AUC in fase di allenamento***
Figura 5 ***a****) Allenamento sulla coppia 1 Training Set/Test Set, l'allenamento termina dopo 301 iterazioni in quanto non si hanno ulteriori miglioramenti*
Figura 5 ***b****) Allenamento sulla coppia 4 Training Set/Test Set, l'allenamento termina dopo 497 iterazioni in quanto non si hanno ulteriori miglioramenti.*
Le curve sono calcolate sia sul Training Set che sul Test set

La funzione predict del lightgbm non fornisce direttamente le classi, ma uno score ed è quindi necessaria l'imposizione di una soglia per ottenere la binarizzazione nelle due classi. Tale soglia è stata ricavata dalla valutazione della curva ROC ottenuta a partire dal vettore di predizione sul test e dal vettore contenente le classi reali, ed è stata selezionata in modo da garantire il maggior scarto tra veri positivi e falsi positivi.

Le curve ROC sono indicatrici importanti dell'andamento della predizione, forniscono informazioni rilevanti sulla sensibilità e specificità del classificatore.

Nate per valutare la qualità dei radar le curve ROC sono utilizzate in medicina per asserire l'accuratezza di un test diagnostico e per ottenere la miglior soglia in grado di distinguere tra test positivi e negativi che nel presente lavoro di tesi coincidono con le classi ES ed EF. [19]

Le curve ROC forniscono la rappresentazione grafica del compromesso tra sensibilità e specificità. [19]

Il miglior cutoff è quello che garantisce la più elevata sensibilità e contemporaneamente la maggior specificità ed è situato nell'angolo in alto a sinistra della curva (*Figura 6*)

Una curva ROC che viene a coincidere con la diagonale produce lo stesso numero di falsi positivi e veri positivi. L'area sottesa alla curva ROC è una misura di quanto il test sia discriminativo, tanto più il valore dell'area è circa 1 maggiore è la capacità del test di distinguere tra le classi, mentre un valore di 0.5 è indice dell'incapacità del test di discriminare tra le classi. [20]

Curva ROC sul Training Set 1

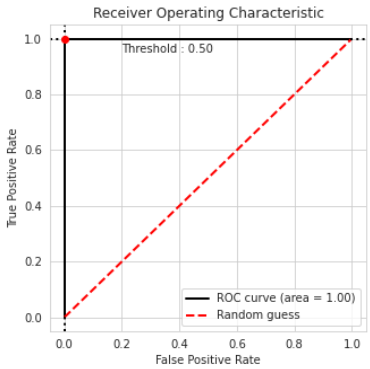


Figura 6 ***curva ROC ottenuta sul Training Set***
In figura è mostrata la curva ROC ottenuta sul primo Training Set testato; in questo caso l'andamento è quello ideale, un valore di AUC pari a 1, e un cutoff nell'angolo in alto a sinistra. In rosso invece è evidenziata la condizione random.
Una curva di questo genere suggerisce condizioni di perfetta classificazione sul set su cui è costruita.

Nel presente progetto si è effettuato il tuning, l'allenamento, il test e la successiva validazione per ciascuna delle coppie di Training set e Test set costruito (il Validation set è lo stesso per tutte) e per ciascuna si sono ottenute le curve ROC sui tre dataset; in *Figura 7* si riportano le curve ottenute sulla prima e ultima coppia Training Set e Test Set.

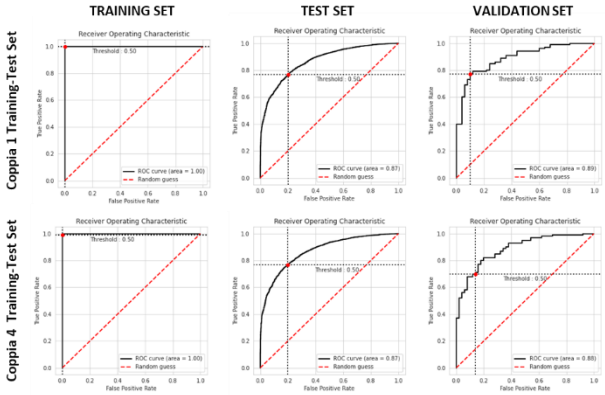


Figura 7 ***Curve ROC su Training, Test e Validation Set***
In figura si possono osservare sulla riga in alto le curve ROC ottenute per la prima coppia Training Set /Test Set, mentre nella riga inferiore sono riportate le curve relative alla quarta coppia.
Entrambe mostrano un comportamento ideale sul Training Set che viene a peggiorare sui restanti set.

Le curve relative alle restanti coppie di Training set e Test set presentano lo stesso comportamento. La curva ROC sui Training set mostra un andamento ottimale con un valore di AUC pari ad 1. Questi modelli sono infatti in grado di distinguere perfettamente gli elementi appartenenti alle classi ES ed EF sui dati di allenamento, ma questa capacità diminuisce notevolmente sul Test set dove la curva si sposta verso la condizione randomica e questo comportamento si conserva anche sul Validation set.

L'output del classificatore costruito è per ogni paziente valutato compreso tra 0 e 1, quindi fornisce la probabilità che il paziente sia appartenente alla classe positiva, dunque potrà tollerare l'estubazione.

Le classificazioni binarie sono quindi state ottenute valutando una soglia per ciascuna delle coppia di dati, ottenuta sulle predizioni del Test set; le soglie coincidono con il valore che permette di ottenere la maggiore differenza tra il tasso di veri positivi e quello di falsi positivi e grazie a queste ultime è stata dunque possibile la valutazione delle prestazioni dei modelli tramite le confusion matrices espresse in percentuali.

Le confusion matrices consentono di comprende la principale problematica di questo classificatore e al tempo stesso il suo punto di forza (*Figura 8*).
Nota positiva è la sostanziale stabilità del classificatore, che non risente nei risultati di una diversa costruzione del set di addestramento e di test; infatti, pur utilizzando coppie di Training set e Test set differenti i risultati sui diversi modelli sono simili. Di fatto però ciascuno dei modelli presentati riscontra la medesima problematica: una classificazione sostanzialmente perfetta sul Training set e prestazioni decisamente più scarse sul Test set.
In tutti i casi le prestazioni sul Test set e sul Validation set sono significativamente inferiori a quelle ottenute sui dati di allenamento, mostrando una tendenza a una maggior correttezza nella classificazione degli elementi classificati come EF; i modelli hanno quindi la tendenza ad essere più specifici che sensibili.

TRS

Mod.1 (Thr=0,491)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	100%	0%
CLASSE PREDETTA: ES	0%	100%

Mod.2 (Thr=0,575)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	100%	0%
CLASSE PREDETTA: ES	0%	100%

Mod.3 (Thr=0,514)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	100%	0%
CLASSE PREDETTA: ES	0%	100%

Mod.4 (Thr=0,575)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	100%	1%
CLASSE PREDETTA: ES	0%	99%

Mod.tesi prec. (Thr=0,506)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	100%	0%
CLASSE PREDETTA: ES	0%	100%

TSS

Mod.1 (Thr=0,491)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	79%	23%
CLASSE PREDETTA: ES	21%	77%

Mod.2 (Thr=0,575)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	85%	29%
CLASSE PREDETTA: ES	15%	29%

Mod.3 (Thr=0,514)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	81%	24%
CLASSE PREDETTA: ES	19%	76%

Mod.4 (Thr=0,575)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	83%	25%
CLASSE PREDETTA: ES	17%	75%

Mod.tesi prec. (Thr=0,506)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	78%	22%
CLASSE PREDETTA: ES	22%	78%

VALID

Mod.1 (Thr=0,491)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	90%	25%
CLASSE PREDETTA: ES	10%	75%

Mod.2 (Thr=0,575)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	94%	36%
CLASSE PREDETTA: ES	4%	64%

Mod.3 (Thr=0,514)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	100%	31%
CLASSE PREDETTA: ES	0%	69%

Mod.4 (Thr=0,575)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	100%	32%
CLASSE PREDETTA: ES	1%	68%

Mod.tesi prec. (Thr=0,506)	CLASSE REALE: EF	CLASSE REALE: ES
CLASSE PREDETTA: EF	88%	28%
CLASSE PREDETTA: ES	12%	72%

	TRS/TS1	TRS/TS2	TRS/TS3	TRS/TS4	TRS/TS prec.
Threshold	0.492	0.575	0.514	0.575	0.506

*Figura 8 Soglie e **Confusion Matrices su Training, Test e Validation Set***
In figura in basso è riportata la tabella con le soglie individuate da ciascun classificatore grazie alle quali gli elementi sono stati classificati nelle classi ES ed EF. In alto invece sono riportate le Confusion Matrices relative alle quattro nuove coppie Training/Test Set e quelle relative ai risultati ottenuti sul dataset della tesi precedente.
Il Validation Set è comune a tutte le coppie.

I classificatori mostrano dunque una tendenza all'overfitting che si è cercato di evitare modificando i range di esplorazione in fase di tuning, agendo separatamente sui range relativi al numero di foglie e al minimo numero di dati per ciascuna foglia e costruendo questi ulteriori modelli sulla prima

coppia di dataset. I risultati non hanno mostrano tuttavia miglioramenti, la tendenza all'overfitting rimane pressoché invariata e rimarcano il fatto che seppur vengono selezionati in fase di tuning parametri diversi questi ultimi non siano incisivi sulla classificazione che di fatto rimane simile tra diversi modelli.

Infine, come ultima valutazione si è effettuata l'ottimizzazione di un'ulteriore modello di LightGBM sulla codifica decimale della quarta coppia Training e Test Set estratta. I range di ottimizzazione dei parametri sono i medesimi utilizzati nella prima fase di tuning e valutazione dei cinque modelli costruiti. I risultati come si può osservare nelle immagini sottostanti (*Figura 9*) non mostrano sostanziali differenze dai casi precedenti, le percentuali di classificazione sul Test e sul Validation set sono completamente in linea con quelli osservati nella figura precedente (*Figura 8*), mentre il Training set sembra avere prestazioni leggermente inferiori rispetto al caso precedente, ma senza differenze in grado di condurre verso considerazioni differenti rispetto alla problematica dell'overfitting.

Figura 9 ***Prestazioni del classificatore costruito sulla quarta coppia Training Set e Test Set soggetta a codifica decimale***
Sotto la dicitura 'Parametri' la figura riporta la tabella dei parametri selezionati a seguito della fase di Tuning del classificatore. Sotto la dicitura 'Allenamento' si può osservare l'andamento del valore di AUC su Training e Test set durante il perdurare dell'allenamento che in questo specifico caso si è concluso dopo 304 iterazioni, delle 500 possibili in quanto si è verificata la condizione che porta all'intervento del parametro di early_stopping. A destra si possono invece osservare i risultati in termini di curve ROC e mediante le Confusion Matrix.

3.2 Majority Voting

Un'ultima valutazione sui classificatori LightGBM ha portato ad effettuare un tentativo di Majority Voting sul Validation set con lo scopo, qualora gli elementi classificati erroneamente fossero stati diversi tra i modelli, di ottenere un significativo miglioramento in termini di prestazioni. Il Majority Voting prevede l'assegnazione della classe più presente delle classificazione di ciascun

elemento, ma come si può notare dalle Confusion Matrices riportate nell'immagine (*Figura 10*) non si sono ottenuti sostanziali miglioramenti; si può conseguentemente dedurre che la gran parte degli elementi classificati erroneamente sono tali per la maggioranza dei modelli.

VALIDATION SET

Mod.1			Mod.2			Mod.3			Mod.4			Mod.tesi prec.		
Thr=0,491	CLASSE REALE		Thr=0,575	CLASSE REALE		Thr=0,514	CLASSE REALE		Thr=0,575	CLASSE REALE		Thr=0,506	CLASSE REALE	
CLASSE PREDETTA	EF	ES	CLASSE PREDETTA	EF	ES	CLASSE PREDETTA	EF	ES	CLASSE PREDETTA	EF	ES	CLASSE PREDETTA	EF	ES
EF	90%	25%	EF	94%	36%	EF	100%	31%	EF	100%	32%	EF	88%	28%
ES	10%	75%	ES	6%	64%	ES	0%	69%	ES	0%	68%	ES	12%	72%

VALIDATION SET MAJORITY VOTING SUI CINQUE MODELLI

Combinazione degli output.

	CLASSE REALE			CLASSE REALE	
CLASSE PREDETTA	EF	ES	CLASSE PREDETTA	EF	ES
EF	45	30	EF	90%	30%
ES	5	70	ES	10%	70%

EF= se l'ouput è 0 per almeno 3/5 dei classificatori
ES= se l'ouput è 1 per almeno 3/5 dei classificatori

Figura 10 ***Prestazioni dei singoli modelli sul Validation set e confronto con il Majority Voting***
Come si può notare dalle confusion matrices riportate in figura non si nota un sostanziale miglioramento nell'implementazione del majortiy voting sui cinque modelli rispetto ai singoli casi, sintomo che gli errori di classificazione commessi siano comuni alle diverse implementazioni

3.3 Features importance

I classificatori LigthGBM consentono di comprendere il peso delle features ai fini della classificazione. È infatti possibile valutare l'importanza delle variabili sia in termini di quanti split la specifica variabile ha prodotto, sia in termini di valore di Shapley che indica la forza della variabile nel guidare la classificazione verso una delle classi.

Lo sviluppo di cinque modelli consente di valutare quanto l'importanza delle variabili sia stabile tra questi. È quindi possibile rilevare se le variabili che hanno prodotto un maggior numero di split in uno dei modelli abbiano la stessa capacità anche sui modelli costruiti su coppie di Training set e Test set differenti.

Il classificatore LightGBM permette di ottenere l'ordinamento delle variabili per importanza in termini di split, si è quindi riusciti a costruire una tabella che

	Variabili	Mod.1	Mod.2	Mod.3	Mod.4	Mod. Tesi prec.
Dati demografici	'Age'	37	26	27	29	35
	'Gender'	55	57	58	54	56
	'Weight_loss'	6	5	3	4	2
Punteggi di gravità	'Glasgow_Coma_Scale'	32	36	41	44	42
	'SOFA_score'	51	52	53	52	51
Somministrazione terapia	'Sedation_days'	1	4	1	1	1
	'Vasopressor'	59	54	59	61	60
Comorbidità	'Chronic_cardiac_insufficiency'	67	62	60	63	62
	'Arrhytmia'	62	58	56	57	55
	'Valvular_disease'	61	64	62	64	71
	'Pulmonary_heart_disease'	71	-1	73	69	-1
	'Peripheral_vascular'	77	73	67	68	70
	'Hypertension'	56	55	55	56	54
	'Stroke'	-1	59	75	-1	-1
	'Other_neurological'	54	53	57	53	56
	'Chronic_Obstructive_Pulmonary_Disease'	69	61	65	72	74
	'Diabetes_uncomplicated'	63	68	70	66	59
	'Diabetes_complicated'	68	65	69	-1	-1
	'Hypothyroidism'	76	72	66	70	67
	'Chronic_kidney_disease'	70	67	74	74	69
	'Liver_disease'	66	63	64	65	63
	'Peptic_ulcer'	-1	-1	-1	-1	66
	'Psychoses'	65	66	61	67	65

Tabella 5 ***Confronto importanza variabili***
In tabella è riportato un sottoinsieme delle variabili presenti nei dataset.
Le variabili sono associate alla posizione che ricoprono in termini di importanza secondo l'ordinamento effettuato dal modello LightGBM.
I modelli presentano posizioni differenti, ma le posizioni appartengono a range di valori contenuto.

permettesse di confrontare diversi modelli a partire dalla posizione che ciascuna features occupa nella scala di importanza.

La posizione, e conseguentemente l'importanza delle features, non è sempre la stessa, seppur si mantenga in generale in un range di valori simili (*Tabella 5*); questa differenza però non si traduce in un'effettiva differenza nelle prestazioni dei cinque modelli.

Osservando la tabella su tutte le variabili è possibile notare che le variabili di comorbidità sono spesso associate ai valori più elevati in termini di posizioni, e conseguentemente sembrano avere una minor importanza nella generazione degli split, mentre le prime posizioni sembrano essere distribuite sulle altre categorie di parametri e non è possibile asserire una maggior importanza di una categoria rispetto alle restanti.

Medesime considerazioni sono state fatte per il valore di Shapley, anche in questo caso si notano differenze tra i modelli che però non conducono ad una differenza di prestazioni.

4. DEEP NEURAL NETWORKS

4.0 Dal Perceptron alle Deep Neural Networks

Le Artificial Neural Networks stanno sempre più emergendo per la loro applicabilità a numerose discipline e grazie alla loro capacità di adeguarsi al modellamento di problemi reali anche complessi. I modelli di ANN (Artificial Neural Network) sono costituiti da unità semplici chiamate neuroni e da ciò si nota il parallelismo con la biologia umana, sebbene lo scopo delle ANN non sia quello di replicare il funzionamento del sistema biologico quanto mimare i meccanismi attraverso cui le reti di neuroni gestiscono problemi complessi.
I neuroni sono costituiti da tre costituenti fondamentali: il corpo cellulare, le estensioni chiamate dendriti e l'assone. Il primo contiene il nucleo della cellula e quindi il materiale genetico, l'assone si presenta come una lunga corda in grado di trasmettere i messaggi dalla cellula, mentre i dendriti sono più corti e consentono la ricezione dei messaggi. La comunicazione avviene tramite neurotrasmettitori, segnali chimici rilasciati a livello delle sinapsi, punto in cui assone e dendriti di neuroni adiacenti vengono in contatto. [21]
Il neurotrasmettitore viene rilasciato nel divario sinaptico e può determinare la generazione da parte dei neuroni vicini di segnale elettrico.
La similitudine tra le ANN e la biologia dei neuroni risiede nelle connessioni tra i nodi, queste ultime infatti rappresentano gli assoni e i dendriti, i pesi delle connessioni rappresentano le sinapsi e la sogliatura l'attività nel nucleo della cellula.
I modelli ANN nascono nel 1958 anno in cui Rosenblatt introdusse il primo modello di neurone artificiale, il Perceptron. Un neurone artificiale riceve gli stimoli dall'ambiente esterno e ne effettua una combinazione così da costituirne una rete che viene passata attraverso una sogliatura lineare che consente di trasmettere l'output al neurone successivo o nel caso del Perceptron all'ambiente, di fatto restituendo l'output. (*Figura 11*)
La rete di input è ottenuta come prodotto dei segnali di input *x* per un vettore di forza *w*.
I pesi delle connessioni tra l'input e l'output di un Perceptron sono modificati proporzionalmente all'errore tra il target e l'output della rete.
Il principale limite del modello Perceptron è l'incapacità di gestire problemi non separabili linearmente, questa difficoltà è stata superata aggiungendo tra il layer di input e quello di output ulteriori layer chiamati hidden layer, in questo caso la rete prende il nome di Multi Layer Perceptron.

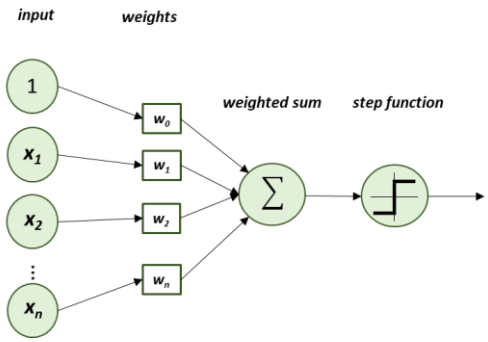


Figura 11 ***Modello di Perceptron con n valori di input***
In figura è rappresentato un modello di Perceptron che valuta la somma pesata degli input moltiplicati per i pesi e la valuta su una funzione 'step' che permette di ottenere l'output

Le MLP agiscono in maniera simile al Perceptron, le informazioni vengono passate dai nodi di input a quelli dei layer nascosti e raggiungono infine il layer di output. Queste reti necessitano di un algoritmo di allenamento più complesso, uno dei più famosi è quello di backpropagation, metodo che si basa sulla retropropagazione degli errori al fine di ottenere i pesi corretti delle connessioni sinaptiche tra i neuroni.
Il learning è uno degli elementi caratteristici dei sistemi di intelligenza artificiale e può essere visto come il processo attraverso cui una rete mutua le sue caratteristiche in risposta ad uno stimolo esterno.

Il mondo del Machine Learning e dell'Intelligenza Artificiale si è continuamente evoluto negli anni e in tale framework il Deep Learning sta prendendo sempre più piede grazie alla sua stabilità in fase di training, la capacità di generalizzazione e la scalabilità su grossi dataset.
Gli algoritmi di Deep Learning sono strutture molto profonde costituite da un elevato numero di layer di rappresentazione e astrazione. [22]
Ad ogni livello della rete viene calcolata la somma pesata moltiplicando gli input (o gli output del layer precedente) per il vettore di pesi del livello stesso.
Le reti di Deep Neural Network consistono di un input layer un layer di output e tra questi un set di livelli nascosti (*Figura 12*).

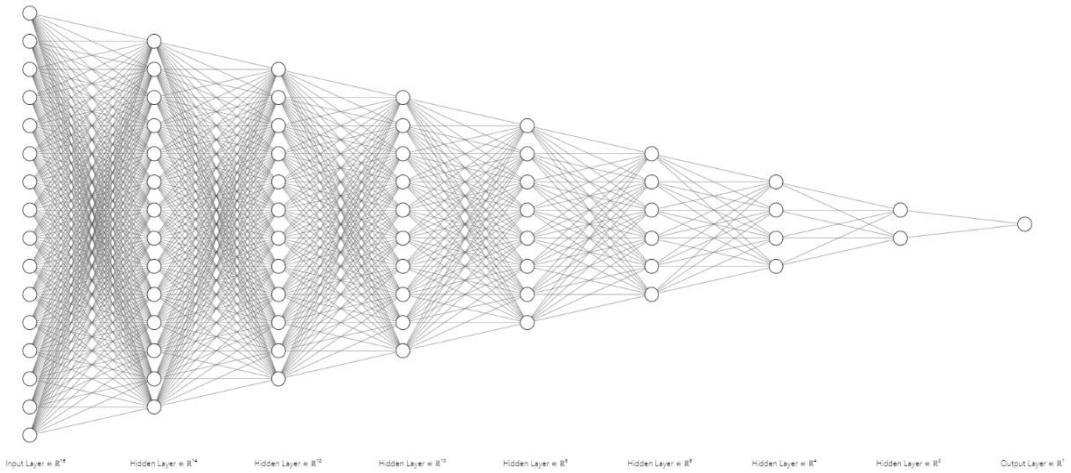


Figura 12 ***DNN con 16 layers di input***
La rete scala di due neuroni in hidden layers successivi fino al singolo neurone di output.
In figura si può osservare un esempio di struttura di una Deep Neural Network; come si può notare la rete risulta altamente complessa sia per il numero di neuroni e layer nascosti, sia per le connessioni tra neuroni di layer successivi. Si tratta infatti di reti dense e completamente connesse e questa caratteristica spiega i tempi di allenamento piuttosto lunghi.

Le reti neurali dense (DNN) sono reti MLP completamente connesse e presentano in generale un numero maggiore di livelli nascosti. Per alcuni anni lo sviluppo delle ANN ha subito una battuta di arresto nella sua evoluzione a causa di un addestramento delle reti che richiedeva un tempo troppo elevato. Le reti MLP e conseguentemente DNN sono state riprese a seguito dell'evoluzione di algoritmi di ottimizzazione rapida e grazie allo sviluppo di unità di elaborazione più avanzate in grado di supportare l'implementazione degli algoritmi di deep learning. [23]
Le tecniche di Deep Learning stanno prendendo nel corso degli ultimi anni sempre più piede; come mostrato in *Figura 13* dall'inizio del secondo decennio del XXI secolo l'utilizzo di tecniche di Deep Learning in ambito medicale è cresciuto in esponenzialmente, passando da un numero limitato di documenti sulla tematica nel 2010 ai quasi mille documenti nel 2020.
Le principali aree di interesse sono la medicina, la Computer Science e l'ingegneria che da sole costituiscono più del 50% di tutti i documenti prodotti fino ad ora. Con ottimismo si può immaginare che questa tendenza si manterrà in crescita, grazie ad un continuo miglioramento delle tecnologie a disposizione che permettono di utilizzare sistemi sempre più complessi e computazionalmente pesanti. A suo favore il framework del Deep Learning mostra la capacità di adattarsi a innumerevoli situazioni.

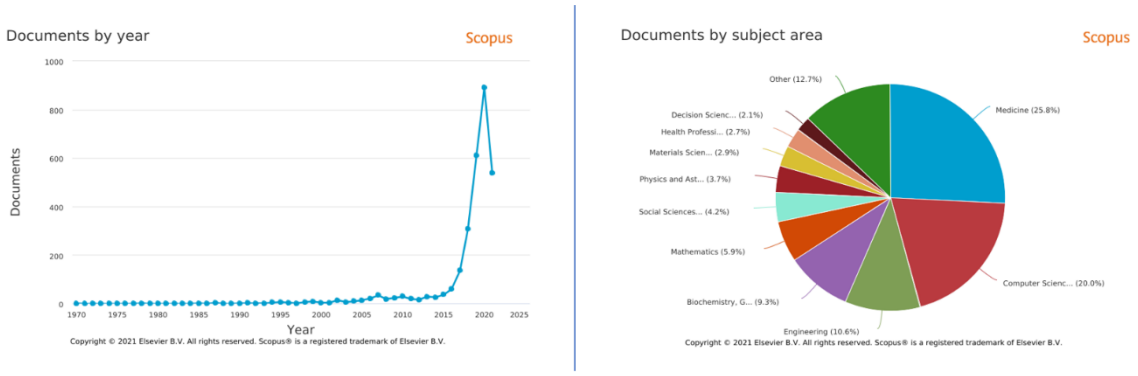


Figura 13 ***Grafici di interessamento del Deep Learning nell'ambito della medicina***
Query utilizzata per effettuare la ricerca (Deep Learning) AND (medicine).
Dal grafico a sinistra che riporta sull'asse delle ordinate l'anno di pubblicazione e sull'asse delle ascisse il numero di documenti si può notare come l'utilizzo di tecniche di Deep Learning abbia subito una crescita notevole soprattutto nell'ultimo decennio.

4.1 Implementazione delle DNN

Una prima serie di prove è stata effettuata al fine di valutare l'idoneità delle reti di Deep Learning all'affrontare la problematica dell'individuazione dell'esito di estubazione. Si sono quindi analizzate le prestazioni dei classificatori sulle coppie Training e Test set con tutte le features per poi successivamente proseguire con una fase di features selection.
Si è dunque effettuata una prima verifica, implementando, tramite la funzione di Matlab 'feedforward', quattro tipi di strutture con funzione di attivazione differenti, in un primo caso si è utilizzata la funzione sigmoidea tangente iperbolica (tansig) agli hidden layer e una funzione lineare (purelin) al layer di output, mentre in un secondo momento si sono effettuate le valutazioni impostando una funzione di attivazione ReLu agli hidden layer.
La classe di ogni elemento viene ottenuta dalla binarizzazione del valore fornito dall'output del classificatore; Ogni volta che viene creata e allenata una rete viene ottenuta anche la curva ROC relativa al Training set; la curva consente l'individuazione della soglia ottenuta come il miglior compromesso tra sensibilità e specificità del modello. La binarizzazione avviene confrontando il valore di output con il valore soglia e conseguentemente l'elemento verrà associato ad una delle classi.
Le strutture testate differiscono per il numero di neuroni in hidden layer successivi, la prima struttura presenta un decremento in numero di neuroni tra hidden layer successivi pari ad 1, mentre le successive pari a 2, 3 e 4. Le reti presentano quindi una struttura triangolare con un numero di hidden layer differenti. Il layer di input è costituito da 85 neuroni in tutte le strutture, pari quindi al numero di features dell'elemento di input (dataset con tutte le variabili 2.0 Database MIMIC III), il primo hidden layer presenta un numero di neuroni pari alla differenza tra i neuroni del layer di input e il decremento applicato (1,2,3,o 4); il medesimo decremento viene applicato agli hidden layer successivo fino al raggiungimento del numero minimo di neuroni nell'ultimo hidden layer che sarà quindi seguito dal neurone di output. All'aumentare del decremento, ovvero della differenza di neuroni tra hidden layer successivi, il numero di quest'ultimi diminuisce e conseguentemente anche la complessità della rete e le tempistiche di training (*Figura 14*).

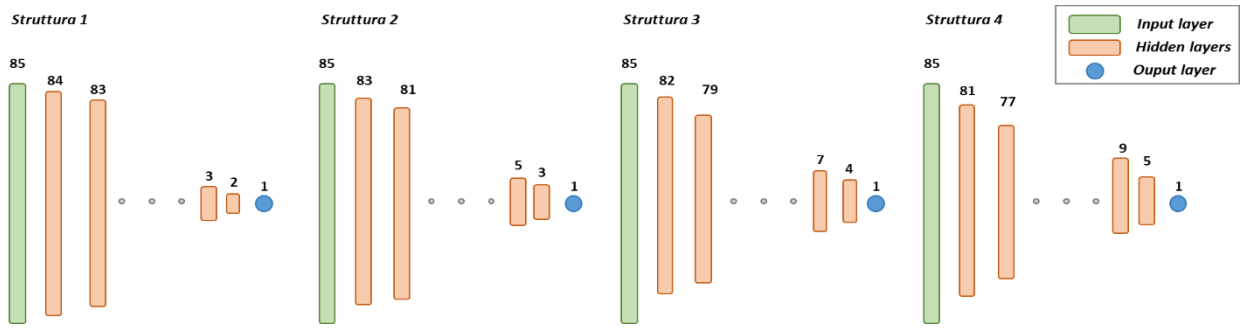


Figura 14 ***Strutture delle reti DNN****.*
In figura sono riportate le schematizzazioni delle strutture di reti di Deep Learning implementate; presentano il layer di input con dimensione di 85 neuroni per poi proseguire con un numero di hidden layer con un numero di neuroni che decresce dagli 85 del layer di input a passi di 1,2,3 e 4 (rappresentati nelle 4 strutture); infine le reti presentano un layer di output costituito da un neurone.

Le prime prove hanno evidenziato l'adeguatezza delle strutture al problema, mostrano infatti, prestazioni soddisfacenti sulle reti che presentano funzione di attivazione sigmoidea tangente iperbolica agli hidden layer; al contrario le reti che presentavano funzione di attivazione ReLu agli hidden layer hanno manifestato invece prestazioni estremamente sbilanciate nei confronti di una classe.
I diversi allenamenti effettuati sulle reti risultavano affetti una certa instabilità di classificazione, soprattutto per le reti più complesse; in alcuni allenamenti la classificazione risultava essere fortemente sbilanciata nei confronti di una delle classi; pertanto, si è reso necessario effettuare un Majority Voting su cinque allenamenti della medesima struttura, imponendo una soglia di allenamento di almeno il 10% di classificati corretti su entrambe le classi, così da effettuare un nuovo training della rete se questa soglia non viene raggiunta. Sulle prestazioni ottenute tramite Majority Voting si sono costruite le Confusion Matrices delle quattro strutture testate.
Dalla *Figura 15* si può notare che al crescere della differenza in numero di neuroni tra hidden layer successivi si ottiene un miglioramento delle prestazioni e un maggior equilibrio tra le due classi.

TRAINING SET

DEC 1 NEURONE			DEC 2 NEURONI			DEC 3 NEURONI			DEC 4 NEURONI		
CLASSE PREDETTA \ CLASSE REALE	EF	ES	CLASSE PREDETTA \ CLASSE REALE	EF	ES	CLASSE PREDETTA \ CLASSE REALE	EF	ES	CLASSE PREDETTA \ CLASSE REALE	EF	ES
EF	32,78%	24,57%	EF	61,57%	35,85%	EF	74,52%	36,11%	EF	69,39%	28,13%
ES	67,22%	75,43%	ES	38,43%	64,15%	ES	25,48%	63,89%	ES	30,61%	71,88%

TEST SET

DEC 1 NEURONE			DEC 2 NEURONI			DEC 3 NEURONI			DEC 4 NEURONI		
CLASSE PREDETTA \ CLASSE REALE	EF	ES	CLASSE PREDETTA \ CLASSE REALE	EF	ES	CLASSE PREDETTA \ CLASSE REALE	EF	ES	CLASSE PREDETTA \ CLASSE REALE	EF	ES
EF	31,60%	25,64%	EF	56,50%	38,37%	EF	72,93%	37,37%	EF	67,25%	30,17%
ES	68,40%	74,36%	ES	43,50%	61,63%	ES	27,07%	62,63%	ES	32,75%	69,83%

Figura 15 ***Confusion Matrices del Majority voting su 5 allenamenti per ciascuna delle strutture testate.***
In figura sono riportate le Confusion Matrices rispettivamente su una coppia Training Set e Test Set di ciascuna delle strutture testate. Per ciascuna struttura si è effettuato un Majority Voting allenando 5 volte la medesima rete e imponendo un numero di corretti classificati pari ad almeno il 10% su entrambe le classi.
Le medesime valutazioni sono state ottenute per tutte le coppie Training set e Test set.

Infine, un'importante elemento a favore dei classificatori testati è il sostanziale mantenimento delle prestazioni ottenute sul Training Set anche sul Test Set.
Le prestazioni sono state valutate sulle cinque coppie di Training set e Test set con 85 variabili create precedentemente e si sono rivelate pressoché comparabili seppur testate e allenate su set di dati differenti ad ulteriore riprova della soddisfacente rappresentatività del Training set.
Le quattro strutture finali hanno quindi in comune la funzione di attivazione dei neuroni dei layer nascosti (sigmoidea tangente iperbolica "tansig") e sul il neurone di output una funzione lineare (purelin), mentre differiscono per il decremento in numero di neuroni tra hidden layer successivi (1,2,3 e 4 neuroni/layer).
Si è quindi deciso di sottoporre le strutture testate alla fase di ottimizzazione, tramite features selection che comporta non solo la selezione delle variabili, ma anche l'identificazione della struttura ottimale.

4.2 DNN: Features and Structure Selection

I dataset sono costituiti rispettivamente da 85 e 58 variabili (2.0 Database MIMIC III), un numero così elevato di variabili può risultare difficile da gestire nella pratica quotidiana; al tempo stesso non tutte le variabili potrebbero risultare importanti per la classificazione e alcune potrebbero essere anche causa di rumore.
Al fine di effettuare la selezione delle variabili e con lo scopo di ottimizzare le prestazioni dei classificatori si è resa necessaria una fase di features selection; la riduzione del numero delle variabili permette di ridurre la complessità dei modelli, la cui struttura, come descritto in precedenza, dipende fortemente dal numero di variabili presenti, traducendosi in una diminuzione delle tempistiche di allenamento e classificazione. Le reti neurali costruite presentano un elevato numero di hidden layer e sono in grado di estrarre conoscenza dai dati anche riconoscendo legami non lineari tra le variabili; pertanto, l'utilizzo di metodi di selezione lineari risulta essere un approccio non indicato. Nel presente progetto si è scelto di sfruttare in fase di Features Selection un algoritmo genetico che consenta di ottimizzare il numero di variabili e conseguentemente anche la struttura della rete di Deep Neural Network.
Molti dei metodi utilizzati nella Computer Science si ispirano al mondo reale umano e/o animale; un primo esempio è già stato affrontano nella descrizione delle reti neurali che tentano l'imitazione del complesso sistema nervoso umano per proporre classificazioni adattandosi al contesto. Un secondo metodo che trae spunto dalla biologia sono gli Algoritmi Genetici (GA) i quali, ispirandosi ai meccanismi di riproduzione delle specie e alle mutazioni e caratteristiche che ne consentono una più probabile sopravvivenza, permettono di effettuare ottimizzazioni di diverso tipo. Gli Algoritmi Genetici ricercano l'ottimo simulando l'evoluzione, partendo da un insieme iniziale di soluzioni (cromosomi) e generando soluzioni successive. Tramite un meccanismo di selezione gli individui più adatti tendono a sopravvivere e a trasmettere le proprie caratteristiche alle generazioni successive. [24]
Gli algoritmi genetici sono un metodo di ricerca euristico introdotto per la prima volta da Jhon Holland nei primi anni '70. Questo tipo di algoritmo parte dalla definizione di un set di soluzioni chiamato popolazione di cui ogni soluzione rappresenta un cromosoma. Ad ogni generazione viene effettuato il calcolo della fitness che consente di assegnare una probabilità di selezione per la generazione successiva in base al valore calcolato. Alcuni dei cromosomi vengono selezionati per produrre una nuova generazione (popolazione figlia), ottenuta tramite crossover e mutazioni sui cromosomi genitori.

In accordo con quanto annunciato in precedenza i cromosomi con la maggior capacità di sopravvivenza (fitness) presentano una maggior probabilità di selezione e conseguentemente i cromosomi della nuova generazione tenderanno ad avere una fitness maggiore di quelli precedenti.

Questo procedimento viene ripetuto fino a che non viene raggiunta la condizione di stop [25].

L'algoritmo genetico si sviluppa secondo quattro passaggi fondamentali:

1. **Inizializzazione della popolazione**: viene creata una popolazione di soluzioni generate randomicamente. La dimensione della popolazione iniziale dipende dal problema, ma contiene in genere centinaia o migliaia di soluzioni.
2. **Selezione**: Nelle generazioni successive una parte della popolazione esistente viene selezionata secondo un processo basato sul valore di fitness dove le soluzioni più adatte sono selezionate con maggior probabilità. Molte funzioni di selezione sono create in modo da consentire una selezione, seppur minima, di soluzioni con una fitness meno adatta in modo da evitare la convergenza prematura e conseguentemente soluzioni non ottimali.
3. **Riproduzione**: viene generata una nuova popolazione attraverso l'utilizzo di operatori genetici di crossover o mutazione. La nuova soluzione viene generata a partire da una coppia di soluzioni genitori, condividerà con quest'ultima numerose caratteristiche.
4. La generazione di nuove soluzioni viene mantenuta fino al raggiungimento di una **condizione di stop**. Tra le più comuni si possono individuare:
    a. Soddisfacimento di un criterio minimo
    b. Numero fisso di generazioni
    c. Raggiungimento di un plateau nella fitness
    d. Ispezione manuale
    e. Raggiungimento di un budget

A quelle descritte si aggiunge la combinazione di una o più condizioni.

Figura 16 ***Schema generale di funzionamento di un algoritmo genetico***
In figura è rappresentata una schematizzazione del funzionamento di un algoritmo genetico. Le fasi di selezione, generazione, valutazione e sostituzione proseguono fino alla verifica della condizione di stop.

Nella figura a fianco (*Figura 16*) viene rappresentato lo pseudo-codice del funzionamento di un algoritmo genetico grazie a cui è possibile individuare i passaggi fondamentali di questo metodo di ottimizzazione.

Nel presente progetto l'implementazione dell'Algoritmo Genetico mira a consentire sia l'individuazione della struttura in grado di garantire le performance ottimali sia la selezione delle variabili più adatte alla classificazione. Al fine di effettuare ciò si è inizialmente progettato di effettuare l'ottimizzazione tramite GA di ciascuna delle quattro strutture di Deep Learning, procedendo in un primo momento ad un test semplificato dell'architettura che nelle prime prove aveva mostrato maggior stabilità e prestazioni migliori, ovvero sulle strutture con decremento di 4 neuroni ogni hidden layer

e una fitness valutata sul primo allenamento accettabile (allenamento con il 10% di corretti classificati sulle due classi) di cinque possibili.

Selezione della Coppie Training e Test per l'ottimizzazione

L'implementazione dell'algoritmo di ottimizzazione ha reso necessaria la selezione di una coppia di Training set e Test set tra quelle create.
L'analisi delle prestazioni ottenute sulle prime strutture di DNN aveva evidenziato risultati simili tra le reti costruite sulle cinque coppie Training e Test set; la variabilità delle reti è infatti riconducibile alla natura randomica dell'allenamento piuttosto che alla coppia di dataset su cui è stata allenata e valutata. Le strutture mostrano infatti il medesimo comportamento: Le classificazioni ottenute sui Test set rispecchiano di risultati ottenuti sui Training set.

Nonostante le prestazioni delle reti costruite sulle cinque coppie Training set e Test set siano del tutto simili si è comunque effettuata la scelta della coppia più adatta, confrontando i risultati.
Si è quindi cercata un'elevata specificità, privilegiando la classe EF, ma non rinunciando comunque a un discreto equilibrio tra le classi; si è giunti quindi alla selezione della quarta coppia Training set e Test set creata nel presente progetto di tesi (2.1 Costruzione di quattro coppie di Training Set – Test Set 85 features). (*Figura 17)*

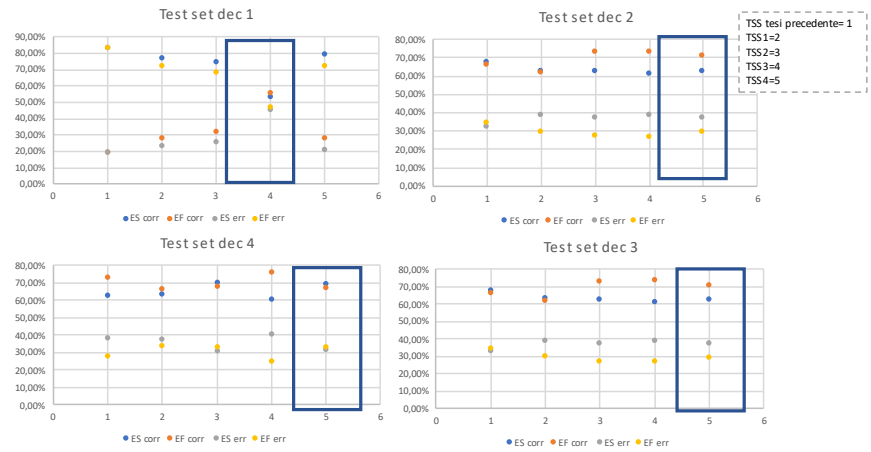


Figura 17 ***Prestazioni percentuali sul Test set delle quattro strutture***
In figura sono rappresentati i risultati sui Test set Majority Voting di cinque allenamenti effettuati sui cinque Training set. Sono riportate le percentuali di corretti classificati e le percentuali di elementi classificati erroneamente. Le prestazioni tra le reti non mostrano differenze significative ed è quindi risultati difficile individuare la coppia più adatta; si è quindi optato per un compromesso tra specificità elevata senza perdere in equilibrio tra le classi

4.2.0 Parametri dell'algoritmo genetico nel presente progetto e implementazione definitiva

Codifica delle soluzioni

Primo passaggio fondamentale nell'impostazione di un algoritmo genetico è la codifica delle soluzioni. Generalmente la codifica utilizzata per questi algoritmi è di tipo binario, vengono solitamente costruite soluzioni costituite di 0 e 1 che permettono la codifica di diversi parametri come il tipo di struttura o il valore di un parametro specifico. Nel presente progetto si è scelta una codifica di tipo yes/no variable, che determina la presenza o assenza nella soluzione di una determinata variabile. La coppia Training e Test set selezionata è composta da 85 variabili, tramite la codifica yes/no variable si costruiscono soluzioni che presentano da 0 a un massimo di 85 variabili. Nel presente progetto non viene dedicata una parte della stringa della soluzione per la definizione della struttura del classificatore dal momento che, secondo quanto già

descritto in precedenza, la struttura risulta condizionata dalla dimensione del layer di input. La soluzione, quindi non determina solamente quali e quante variabili sono state selezionate, ma conseguentemente anche la struttura della rete DNN che verrà testata e sulle cui predizioni avviene il calcolo della fitness.
Il peso computazionale dell'algoritmo implementato risulta piuttosto ingente, pertanto si è scelto di non inserire una condizione di ammissibilità della soluzione individuata, ma piuttosto di assegnare un valore di fitness, a quelle soluzioni con un numero di variabili selezionate inferiore a due, tale da ridurre estremamente la probabilità di una successiva selezione.
In un secondo momento si è implementato un ulteriore GA sulla quarta coppia Training e Test set con 58 features, codifica decimale della medesima coppia selezionata per l'ottimizzazione, ciò ha implicato che la stringa codificante le soluzioni fosse costituita da 58 bit.

Popolazione iniziale

La soluzione così impostata non presenta dunque vincoli di sorta; pertanto, il numero di soluzioni totali del problema non vincolato risulta essere $\mathbf{2^n-1}$ avendo n bit per codificare la soluzione. In un primo momento *n* risulta pari a 85, in una fase successiva verrà effettuata in parallelo l'ottimizzazione sul dataset in cui sono state sostituite le comorbidità, pertanto *n* sarà pari a 58.
La stima del numero di soluzioni esplorate dall'algoritmo tra tutte le possibili soluzioni del problema tiene conto della seguente relazione, ipotizzando che vengano generate soluzioni già esplorate in iterazioni precedenti: #individui nella popolazione + (#individui generati × # iterazioni).
Tendenzialmente è preferibile incrementare la dimensione della popolazione iniziale anziché il numero di iterazioni, così da poter esplorare zone diverse dello spazio; tuttavia, questa scelta può portare a tempistiche di esecuzione importanti.
La popolazione inziale è stata scelta pari a 5000 individui così come il numero di iterazioni possibili per la ricerca dell'ottimo.
Il numero di individui genitori che verranno selezionati al fine di costituire la popolazione figlia è stato imposto pari al 40% degli individui totali, valore piuttosto limitato, ma che viene compensato con la scelta di effettuare il crossover su tutti gli individui della popolazione genitrice.
Al fine di esplorare maggiormente lo spazio delle soluzioni nel presente progetto si è effettuata la sostituzione degli individui generati agli individui genitori e unendo questa nuova popolazione alla precedente non selezionata come genitrice.

Condizioni di ammissibilità

Nel presente progetto non sono state imposte condizioni di ammissibilità della soluzione, in questo modo si alleggerisce il peso dell'algoritmo. Tuttavia, qual ora le soluzioni presentassero un numero di features selezionate inferiore o pari a due allora il valore di fitness relativo alla soluzione forzato a un valore 1000, in tal modo risulterà pressoché impossibile la selezione della soluzione con tali caratteristiche nelle generazioni successive.

Funzione di fitness

La funzione di fitness è di fatto la driving force dell'algoritmo genetico. Può essere adattata alle diverse problematiche affrontate.
Nel presente progetto si è impostata la fitness in modo da effettuare l'ottimizzazione ricercando il minimo valore ottenuto dalle soluzioni.
La funzione di fitness è stata scelta in modo da valutare le prestazioni del classificatore, costruito secondo la soluzione in esame, e in modo da tenere conto del bilanciamento tra sensibilità e specificità, e penalizzando soluzioni che portino a un comportamento sbilanciato del modello.
La funzione di fitness, da minimizzare, è stata definita come:

1 - corretta classificazione totale + 0,3 × | sensibilità – specificità |

Tale funzione è stata valutata sulle classificazioni effettuate sul Test set a seguito dell'allenamento della rete sul Training Set.

Nella prima fase di valutazione delle reti era emerso come certi allenamenti restituissero delle reti polarizzate fortemente nei confronti di una delle classi; pertanto, queste classificazioni non sono state considerate utili. Si è quindi sviluppato un primo metodo di calcolo della fitness che tuttavia si è successivamente modificato per il peso computazionale richiesto. Di seguito, verrà descritta la prima implementazione proposta e a seguire l'implementazione definitiva utilizzata nel presente progetto.

Prima implementazione

La prima implementazione ha previsto l'allenamento di un massimo di cinque reti la cui struttura è determinata dalla soluzione di cui si vuole valutare la fitness; ciascun allenamento è stato considerato valido se garantisce almeno il 10% di corretti classificati su entrambe le classi; in caso contrario la rete viene allenata nuovamente fino ad un massimo di due volte. Se per due tentativi consecutivi la rete non garantisce una percentuale di corretti classificati sopra la soglia, allora tale rete viene scartata. In questo modo, a seconda delle condizioni, si possono avere da una fino a cinque reti allenate, con medesima struttura, che vengono impiegate successivamente nella classificazione del Test Set, ottenendo per ciascun elemento un massimo di cinque classificazioni.
Le classificazioni multiple del Test set sono state successivamente ricondotte al caso singolo tramite Majority Voting, secondo le regole riportate in tabella (*Tabella 6*), su cui è stato effettuato il calcolo del valore della funzione di fitness secondo la formula riportata in precedenza.
L'implementazione descritta risultava essere robusta, ma al tempo stesso computazionalmente pesante, tanto da non poter essere gestita dai sistemi a disposizione e nei tempi di questo progetto; per questo motivo si è proseguito con una seconda implementazione semplificata.

Valutazione delle funzione di Fitness		
Reti allenate	***Majority***	***Fitness Test Set***
0	/	1000
1	/	Valutata sulla classificazione
2 o 4	Classe 1 (ES) se 2/2 o 3/4 sono classificazioni ES	Valutazione sull'esito majority
3 o 5	Classe 1 (ES) se 2/3 o 3/5 sono classificazioni ES	Valutazione sull'esito majority

Tabella 6 ***Casistiche di Majority Voting e valutazione della funzione***

Implementazione definitiva

Nell'implementazione definitiva si è effettuato il ri-allenamento della rete, costruita a partire dalla soluzione in esame, fino ad un massimo di cinque volte nel caso in cui la percentuale di corretti classificati di una delle due classi o di entrambe risultasse inferiore al 10% sul Training set. In questo modo si è ragionevolmente certi che se la prestazione sub-ottimale della rete dovesse essere legata all'allenamento, con un numero maggiore di allenamenti questo problema sarebbe superato; qualora invece la prestazione della rete fosse sub-ottimale per fattori legati alla struttura della rete stessa, inadatta al problema, allora a seguito del quinto allenamento senza ottenere valori di corretti classificati sopra la soglia la fitness verrà forzata ad un valore di 1000; la selezione della soluzione diverrà così estremamente improbabile.
La scelta di effettuare fino a cinque allenamenti è stata presa in seguito alle valutazioni delle prime strutture, dove si è visto che su cinque allenamenti almeno uno non portava a classificazioni estremamente polarizzate nei confronti di una delle classi. In ogni caso se la rete risulta spesso polarizzata verso una delle classi in allenamenti successivi non potrà comunque risultare una rete ottimale per il problema.
A seguito del primo allenamento sopra soglia dei cinque possibili, la fase di training si considera conclusa e si effettua la classificazione del Test set sui cui avviene il calcolo della fitness secondo la formula descritta in precedenza.

Condizioni di stop

Si sono impostate due condizioni di stop.
La prima è definita dal numero massimo di iterazioni in cui cercare l'ottimo, impostato nel presente progetto ad un valore di 5000 iterazioni. Al raggiungimento del numero massimo di iterazioni possibili la ricerca della soluzione migliore si ferma.
La seconda condizione riguarda il valore di fitness; infatti, se per 50 iterazioni il valore di fitness non subisce miglioramenti l'algoritmo raggiunge la condizione di stop e si prosegue con le ripetizioni successive.

Selezione delle soluzioni genitori

I metodi di selezione sono vari, alcuni selezionano le soluzioni con la miglior fitness in altri casi invece la selezione può essere randomica e in altri ancora le soluzioni con fitness migliore hanno maggior probabilità di selezione, ma non viene preclusa la possibilità di selezione di soluzioni con fitness peggiore; queste soluzioni vengono definite stocastiche. [25]
Nel presente progetto si è utilizzato un metodo di selezione noto come Roulette Wheel Selection, si tratta di un metodo stocastico che consente la selezione con probabilità maggiore di soluzioni con fitness migliore seppur non escludendo la selezione anche di soluzioni con fitness peggiore, cercando così di scongiurare la ricaduta in ottimi locali e permettendo una ricerca più vasta nello spazio delle soluzioni.

Questo metodo di selezione prevede che ogni soluzione della popolazione sia associata a una probabilità di selezione ed essendo il numero di soluzioni della popolazione fisso, la somma delle probabilità delle soluzioni presenti risulta essere unitaria. Si può immaginare questa implementazione come una roulette che schematicamente può essere rappresentata come una circonferenza divisa in ***k*** parti (dove ***k*** è pari al numero di soluzioni nella popolazione) e in cui la dimensione di ogni arco di circonferenza è correlata alla probabilità di selezione della soluzioni e quindi al valore di fitness. [24]

Popolazione	Fitness
Soluzione 1	0,2863
Soluzione 2	0,2990
Soluzione 3	0,2892
Soluzione 4	0,2911
Soluzione 5	0,2844

Figura 18 ***Rappresentazione della probabilità di selezione nel metodo Roulette Wheel***
La figura fornisce una chiara rappresentazione del funzionamento del metodo di selezione utilizzato. La tabella a destra riporta per ciascuna delle soluzioni presenti in popolazione un ipotetico valore di fitness (in questo caso la popolazione è composta da cinque elementi per comodità di rappresentazione), a tale valore viene associato una corrispettiva probabilità di selezione rappresentata a sinistra dalla dimensione dei settori circolari.

Nel presente progetto si è scelto di valutare l'ottimizzazione come un problema di minimizzazione, pertanto, soluzioni con fitness minori sono associate a una probabilità di selezione maggiore.
La *Figura 18* permette di comprendere appieno il funzionamento di questo metodo, le soluzioni con fitness minore, ricoprendo un maggior spazio dell'area sono associate quindi una maggior probabilità che il 'puntatore' si fermi in una zona del cerchio di appartenenza a tali soluzioni. Questo non esclude però che anche le soluzioni meno adatte vengano selezionate seppur meno frequentemente e ciò permette quindi di esplorare maggiormente lo spazio delle soluzioni. Un'ampia esplorazione risulta cruciale laddove lo spazio delle soluzioni è vasto e si deve fare i conti con limiti di tipo computazionale come nel presente progetto.

Parametri di crossover e mutation

A seguito dell'inizializzazione della popolazione iniziale per ciascuna delle soluzioni presenti in essa viene effettuato il calcolo della fitness; tale calcolo consente la selezione della popolazione 'genitore' privilegiando le soluzioni che hanno un valore di fitness minore (trattasi di un problema di minimizzazione). Nel presente progetto si è scelto di estrarre come popolazione genitore il 40% delle soluzioni.
Selezionati i genitori l'algoritmo prevede una fase di generazione di nuove soluzioni tramite meccanismi di crossover e mutation; ottenuta la popolazione figlia, quest'ultima viene sostituita alla popolazione genitore e viene ricostruita la popolazione unendo la popolazione figlia con le soluzioni non selezionate in precedenza.
A seguito della selezione dei genitori viene applicato l'operatore di crossover, si tratta di un operatore che, traendo spunto dalla biologia, scambia sequenze di due cromosomi.

L'operatore di crossover è utilizzato per ottenere una soluzione migliore dalla ricombinazione di due soluzioni genitori. Non vengono create soluzioni nuove, ma soluzioni nate dallo scambio di 'materiale genetico' delle soluzioni genitori. [25]
L'operatore di crossover lavora dunque su coppie di genitori con un numero di tagli definito e che determina quante parti dei cromosomi vengono scambiate. Nel presente progetto si è utilizzato un numero di tagli pari a 4. (*Figura 19*)

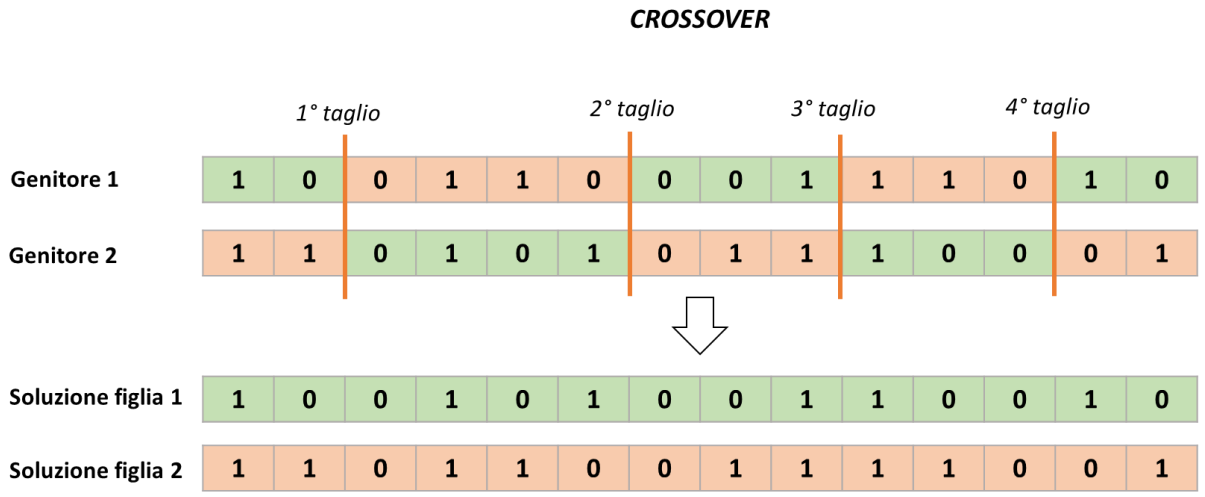


Figura 19 ***Funzionamento dell'operazione di crossover nel presente progetto di tesi***
In figura si può osservare il comportamento dell'operazione di crossover. Le soluzioni genitori, selezionate tra le soluzioni della popolazione, si scambiano porzioni di cromosomi di dimensioni variabili generando due soluzioni figlie con porzioni complementari dei cromosomi di partenza.

La posizione dei punti di taglio è variabile, cambia nelle iterazioni e negli accoppiamenti.
Gli operatori di crossover e di mutazione sono associati a una probabilità, che determina la possibilità di una soluzione genitore di subire o meno le due operazioni; in questo progetto dovendo fare i conti con uno spazio delle soluzioni decisamente vasto (considerando anche la mancanza di vincoli di ammissibilità) e al tempo stesso con un peso computazionale notevole, che ha determinato tempistiche dilatate, si è scelto di imporre una probabilità di crossover p_c pari a 1. In questo modo ognuna delle soluzioni genitore viene sottoposta a crossover, consentendo di aumentare lo spazio di ricerca, pur partendo comunque da una popolazione iniziale limitata.
Il secondo operatore utilizzato nella fase di generazione di nuove soluzioni è quello di mutazione, sugli individui creati in seguito al crossover viene effettuata l'operazione di mutazione, che consiste nella modifica puntuale di un numero variabile di bit scelti in modo random. A differenze dell'operazione precedente, la mutazione riguarda una sola stringa (*Figura 20*).

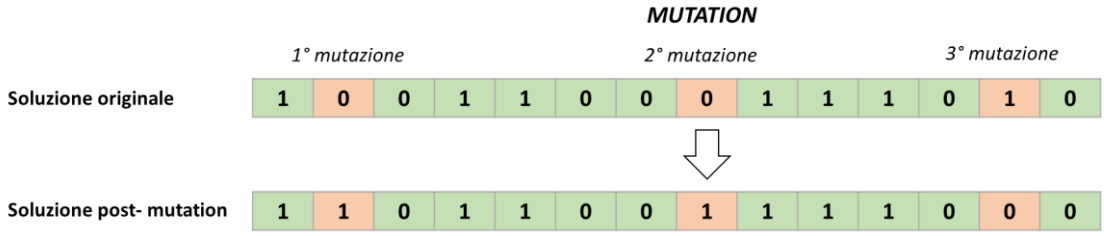


Figura 20 ***Operazione di mutazione dei bit di una soluzione***
In figura è riportata una rappresentazione schematica dell'evento di mutazione; in questo caso la modifica riguarda i singoli bit della soluzione ottenuta a seguito dell'operazione di crossover. La modifica riguarda la singola soluzione.

L'operatore di mutazione è associato ad un valore di probabilità p_m. In questo caso la probabilità è stata inizialmente imposta pari a 0,4 per poi decrescere fino a un minimo di 0,1 con passo 0,001 ogni 25 iterazioni. La mutazione consente di apportare delle modifiche alle soluzioni genitori in modo da poter esplorare maggiormente lo spazio delle soluzioni consentendo di esplorare zone dello spazio che non erano state incluse nella popolazione iniziale. Questa operazione consente inoltre di evitare la stagnazione in ottimi locali.
Al tempo stesso si tratta però di un operatore pericoloso in quanto va ad apportare modifiche a soluzioni già di per sé buone.

4.2.1 Run dell'Algoritmo Genetico

Sono stati costruiti in parallelo due algoritmi di ottimizzazione, il primo è stato applicato al dataset con tutte le variabili mentre per il secondo si è utilizzato il dataset con sostituzione delle comorbidità; In entrambe le implementazioni si è utilizzata in fase di valutazione della fitness la quarta coppia Training/Test set, la cui scelta è stata giustificata in precedenza.
Il primo algoritmo ha lavorato su due popolazioni iniziali differenti che hanno prodotto 30 soluzioni ciascuna. In questo modo si è potuto confrontare 60 soluzioni totali ed è stato possibile notare particolari caratteristiche delle soluzioni legate alla popolazione iniziale differente.
Come facilmente riscontrabile nei grafici riportati in *Figura 21*, le due popolazioni hanno prodotto soluzioni ottimizzate con valori di fitness che variano nel medesimo range tra un massimo di 0,2990 e un minimo di 0,2844. Il grafico relativo al numero di variabili selezionate riporta invece una leggera differenza: le soluzioni della popolazione iniziale 1 hanno mediamente un valore più basso di features rispetto a quelle della popolazione 2 e anche il valore massimo e minimo di features selezionate risulta inferiore a quello della seconda popolazione.
In particolare, la popolazione 1 presenta gli unici casi in cui la fitness e le variabili risultano uguali per più soluzioni; sei soluzioni, delle trenta ottenute, presentano un valore di fitness pari a 0,2935 e 22 variabili selezionate; si tratta del valore minimo di variabili selezionate da una soluzione.
Nei restanti casi anche laddove più soluzioni abbiano ottenuto il medesimo numero di variabili selezionate la tipologia di queste ultime e il valore di fitness associato alla soluzione risultano differenti.

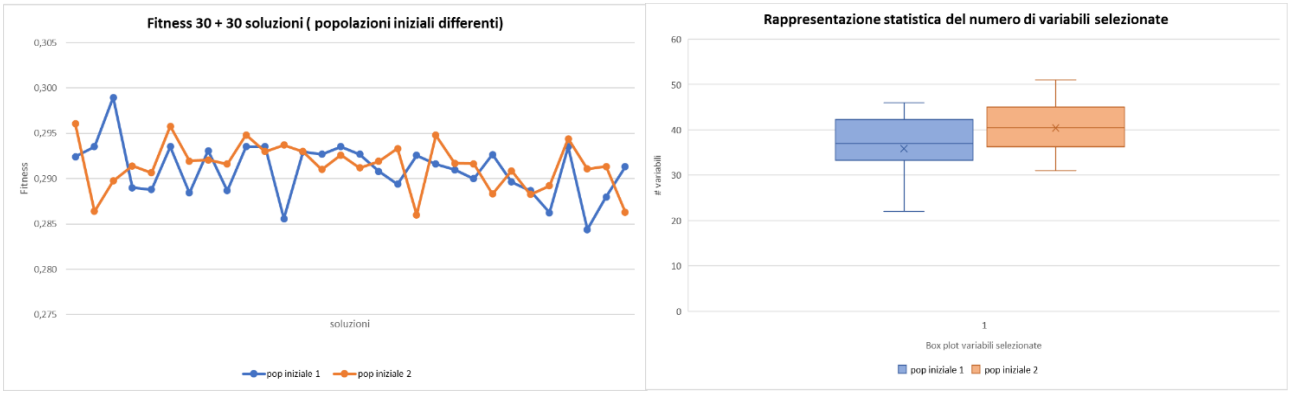


Figura 21 ***Valutazioni e statistiche sulle soluzioni ottenute.***
Nel grafico a sinistra sono rappresentanti i valori di fitness delle 60 soluzioni ottenute a partire da due popolazioni iniziali differenti, a destra invece sono rappresentati i boxplot relativi al numero di variabili selezionate dai due gruppi di soluzioni

Nella figura sottostante (*Figura 22*) è possibile visualizzare tramite grafico, il numero di volte in cui ogni variabile è stata selezionata.
Entrando quindi nel dettaglio di quali sono le variabili comunemente selezionate dalle 60 soluzioni ottenute ciò che risulta evidente a un primo sguardo è che nessuna presenta un rate di selezione nullo. Il minimo rate di selezione è quello della variabile 'Chronic_cardiac_insufficiency' selezionata da nove delle sessanta soluzioni.
Alcune variabili presentano un rate di selezione decisamente alto, tra queste le variabili 'Sedation_days' (indicante il numero di giorni in cui l'individuo è stato sottoposto a sedazione) e 'Mean_Airways_Pressure' (relativa alla pressione a livello delle vie aeree) sono state selezionate da 58 e 60 soluzioni rispettivamente.

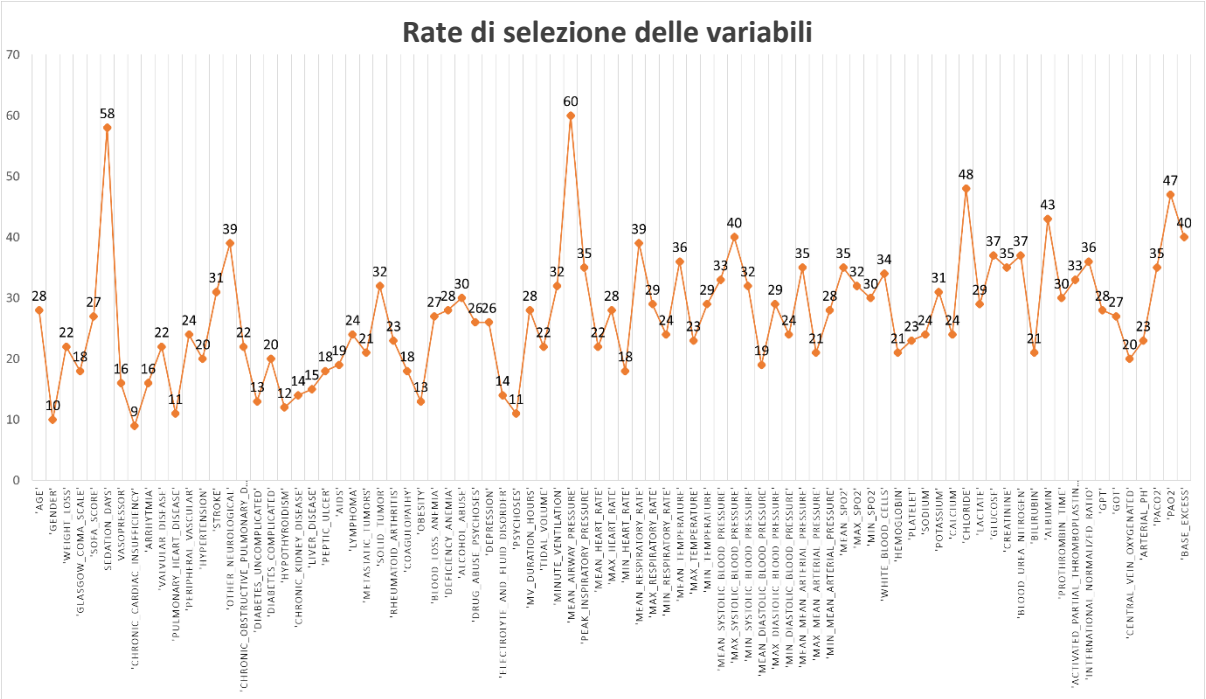


Figura 22 ***Rate di selezione delle 85 variabili***
Il grafico in figura rappresenta per ogni variabile il suo rate di selezione da parte delle 60 soluzioni

A seguire le variabili relative alla cloremia 'Chloride' e alla pressione parziale pressione parziale arteriosa di O2 '$Pa0_2$' selezionate da rispettivamente 48 e 47 soluzioni.
Le variabili 'Sedation_Days' e la variabile 'Chloride' sono risultate avere il rate di selezione maggiore anche nel lavoro di tesi svolto in precedenza.

Le variabili di comorbidità presentano valori simili e piuttosto bassi; le soluzioni presentano in genere elevata variabilità nel tipo di comorbidità selezionate.
Il secondo Algoritmo Genetico è stato quindi implementato: le impostazioni dei parametri sono rimaste invariate, ma in questa seconda implementazione si è utilizzata la quarta coppia Training e Test set con comorbidità sostituite; conseguentemente si è utilizzata una codifica della soluzione a 58 bit. Sono state effettuate un totale di 50 ripetizioni a partire dalla medesima popolazione iniziale che hanno prodotto altrettante soluzioni. Queste ultime presentano un valore di fitness leggermente inferiore al precedente, con un massimo di 0,279 e un minimo di 0,269; anche il numero di variabili selezionate è mediamente inferiore al caso precedente, ma si tenga presente che parte della differenza è anche dovuta al numero inferiore di features a seguito della codifica delle comorbidità. (*Figura 23*)

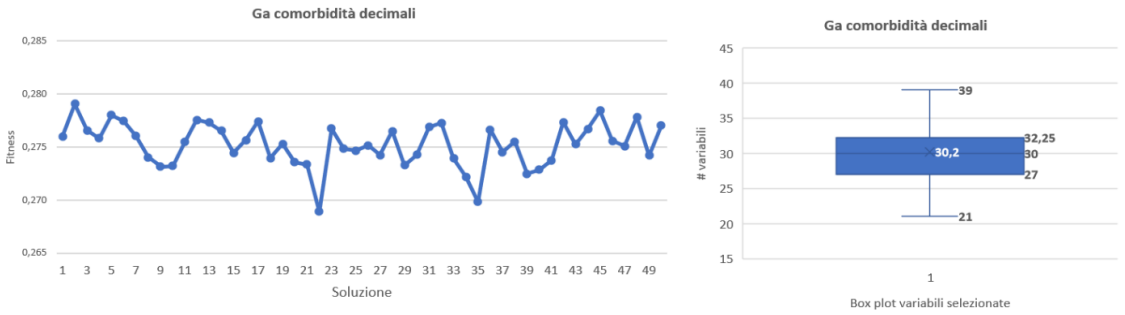


Figura 23 ***Valutazioni e statistiche sulle soluzioni ottenute***
A sinistra la figura riporta i valori delle fitness delle 50 soluzioni ottenute dal GA con comorbidità decimali, a destra invece è possibile osservare una valutazione statistica del numero di variabili selezionate dalle soluzioni ottenute da questo terzo algoritmo genetico.

Questo secondo algoritmo conduce alle medesime conclusioni: le variabili maggiormente selezionate sono ancora 'Sedation_days' e 'Mean_Airway_Pressure' seguite ancora dalla variabile relativa alla cloremia 'Chloride'; il minimo rate di selezione è pari a 3. Non si ottiene quindi un'informazione diversa rispetto a quanto ottenuto con il primo algoritmo implementato.
Si è quindi effettuata la ricerca dei pattern di variabili tra le soluzioni ottenute dai due algoritmi genetici, ma in entrambi i casi non sono venute alla luce particolarità: molteplici combinazioni di variabili consentono di ottenere strutture ottimizzate. In *Figura 24* si riporta la tabella ottenuta sulle soluzioni del GA valutato sulla coppia Training e Test Set con comorbidità, le soluzioni sono state ordinate tramite la funzione 'sortrow' di Matlab, che consente di ordinare le righe di una matrice in base agli elementi della prima colonna.

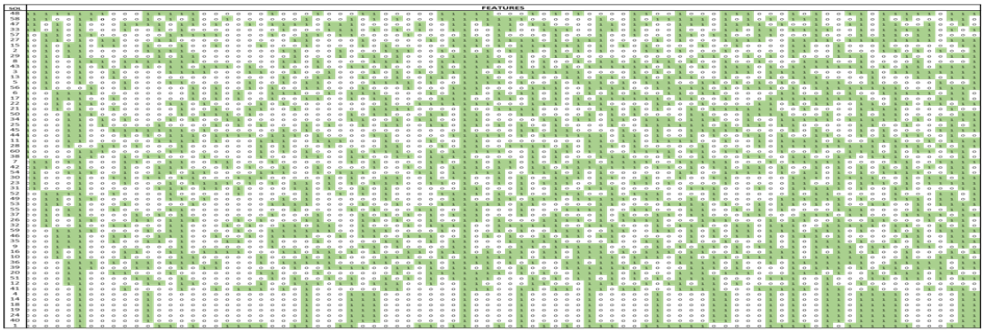


Figura 24 ***Pattern delle variabili selezionate dalle 60 soluzioni dell'algoritmo genetico***
In tabella sono rappresentate le soluzioni ordinate rispetto alla prima riga in ordine decrescente (soluzioni dei GA con le 85 variabili). La matrice in figura è stata ottenuta tramite la funzione 'sortrow' messa a disposizione da Matlab. In verde sono state evidenziate le celle contenenti la selezione (valore '1').

Le soluzioni ottenute dai tre algoritmi di ottimizzazione implementati sono quindi state valutate costruendo per ciascuna la corrispondente rete.

4.2.2 Prestazioni delle reti costruite a partire dalle soluzioni degli Algoritmi Genetici

Ciascuna soluzione dell'algoritmo genetico determina dunque la struttura di una Dense Neural Network (DNN). Nel caso del primo algoritmo genetico, implementato sulla quarta coppia Training e Test set estratta con le 85 variabili, si sono costruite sessanta DNN. Le reti presentano le seguenti caratteristiche di architettura:

- Il layer di input contiene un numero di neuroni definito dalle variabili selezionate dalla soluzione
- un numero variabile di hidden layer che presentano ognuno un decremento di quattro neuroni rispetto al layer precedente
- il layer di output, costituito da un singolo neurone che fornisce la classificazione

Le DNN sono state allenate sul Training Set e valutate sul Test Set già utilizzati in fase di ottimizzazione e imponendo, in fase di allenamento, la soglia del 10% di corretti classificati sulle due classi. Se tale soglia non viene raggiunta la rete viene allenata nuovamente; il mancato raggiungimento della soglia è una condizione rara in quanto la struttura delle reti è già ottimizzata. Infine, si è effettuata la classificazione del Validation set che permette di valutare la capacità di generalizzazione dei classificatori non essendo stato coinvolto nella fase di ottimizzazione.
Il medesimo approccio è stato applicato alle soluzioni del secondo Algoritmo Genetico; dalle cinquanta soluzioni ottenute si sono costruite le rispettive reti in questo caso allenate e testate sulla codifica decimale della quarta coppia Training set e Test set, ancora una volta imponendo la soglia del 10% di corretti classificati, come già avvenuto durante l'ottimizzazione.
Infine, si è testata la capacità di generalizzazione analizzando le prestazioni delle reti sul Validation set ottenuto anch'esso a seguito della codifica delle comorbidità.
Nella figura della pagina precedente (*Figura 25*) sono riportate le prestazioni in termini di specificità

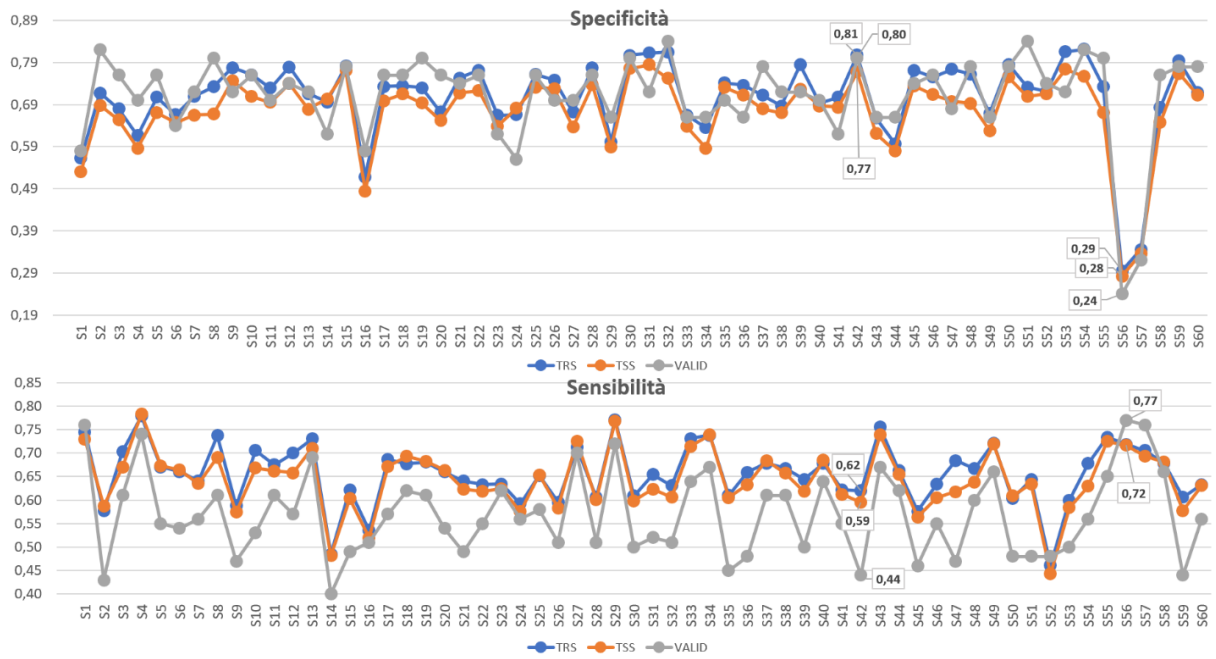


Figura 25 ***Prestazioni delle singole reti***
In figura è riportato il grafico delle prestazioni in termini di specificità e sensibilità delle sessanta reti ottenute a partire dalle soluzioni del GA con comorbidità. In evidenza sono riportati i valori di sensibilità e specificità di alcune reti che meglio consentono di rilevare il comportamento antisimmetrico delle reti rispetto a questi due indicatori di prestazione

e sensibilità delle sessanta reti ottenute dal primo algoritmo genetico; come si può notare le

prestazioni denotano una notevole variabilità tra i classificatori costruiti e in genere il comportamento dei due parametri risulta essere speculare; infatti, laddove uno presenta valori piuttosto elevati, valori discreti si riscontrano sul secondo e viceversa.
Una caratteristica importante è il mantenimento delle prestazioni tra Training e Test set; i due set di dati presentano valori di scarto medi di 3 e 1 centesimi rispettivamente sui parametri di specificità e sensibilità; il Validation set ottiene valori di specificità simili ai set di dati già menzionati, ma sembra avere una sensibilità inferiore.
Prestazioni del tutto simili sono state ottenute sulle cinquanta soluzioni dell'algoritmo genetico valutato sulla quarta coppia Training e Test set con codifica delle comorbidità e sul Validation set ottenuto a seguito della medesima codifica (*Figura 26*).
Nella figura sottostante si possono osservare i valori dei parametri di specificità e sensibilità relativi alle cinquanta reti costruite, ancora una volta si denota una certa variabilità in termini di prestazioni tra le reti. Lo scarto nelle prestazioni di Training e Test set in termini di specificità è in media di 3 centesimi, mentre sul parametro di sensibilità è in media di 0 punti percentuali. Il Validation set presenta, rispetto ai set già citati, valori più elevati in termini di specificità e specularmente inferiori sulla sensibilità.

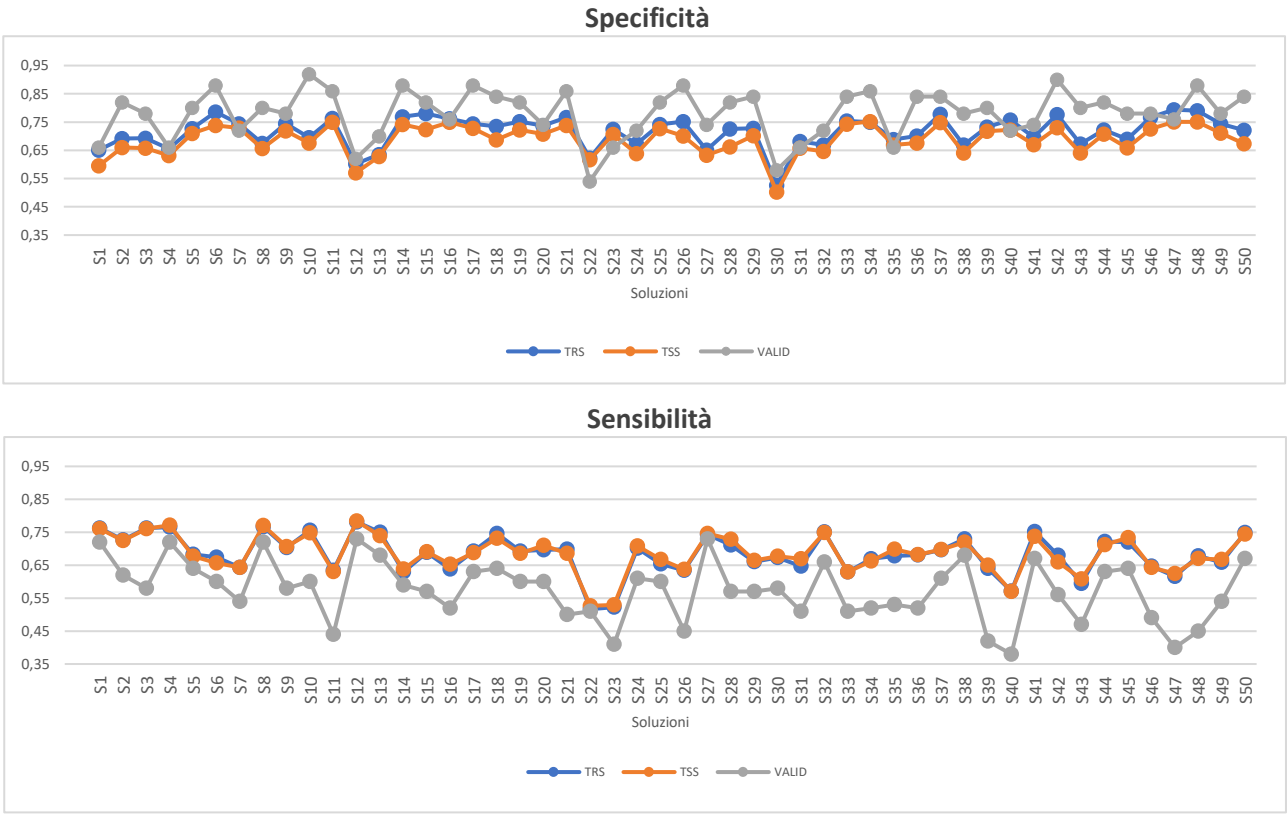


Figura 26 ***Prestazioni delle singole reti***
In figura è riportato il grafico delle prestazioni in termini di specificità e sensibilità delle cinquanta reti ottenute a partire dalle soluzioni del GA costruito su Training e Test Set ottenuti dalla codifica decimale delle comorbidità.

4.2.3 Analisi degli errori

Una delle prime valutazioni effettuate è stata l'analisi degli elementi classificati erroneamente sul Training set e Test set. Le reti producono un numero di errori corrispondente a circa il 30% degli elementi totali presenti nel Training set (2302 elementi).

La prima analisi degli errori è avvenuta tramite i dendrogrammi (*Figura 27*); si tratta di una tecnica di clustering gerarchico che unisce gli elementi sulla base della distanza tra essi.
Ad ogni step della costruzione dei dendrogrammi vengono uniti i due elementi, o i gruppi di elementi che presentano la distanza minore e si procede iterativamente fino a comprendere tutti gli elementi.
Le tecniche di clustering consentono di costruire gruppi di elementi con caratteristiche simili e l'analisi degli errori tramite questa tecnica consente di osservare la presenza di caratteristiche particolari. Se un elemento si trova a distanze importanti rispetto agli altri e questo fa sì che possa costituire un cluster a sé stante, le difficoltà di corretta classificazione di quest'ultimo da parte della rete potrebbero essere spiegate dalla particolarità dell'elemento stesso. Se gli elementi classificati erroneamente possono essere ricondotti al medesimo cluster allora la sua rappresentatività a livello di Training set potrebbe non essere sufficiente a consentire il corretto allenamento del classificatore.
Sono quindi stati individuati gli elementi la cui classificazione errata è comune alle prime otto reti costruite sulle soluzioni ottenute dal primo algoritmo genetico implementato.
Si sono costruiti nuovamente i dendrogrammi relativi alle quattro sottoclassi (EF/ES imputato/non imputato) già utilizzati nella fase di costruzione delle coppie di Training e Test set; il sottoinsieme degli elementi errati è stato anch'esso suddiviso nelle sottoclassi e si è individuato, per ognuno degli elementi errati, il punto di unione di questi ultimi e gli elementi o i gruppi di elementi ad essi più vicini.
I dendrogrammi (*Figura 27*) hanno permesso di valutare le criticità dei classificatori, la presenza di classi e sottoclassi più o meno affette da errori e le distanze a cui questi elementi vengono a unirsi con i più vicini.

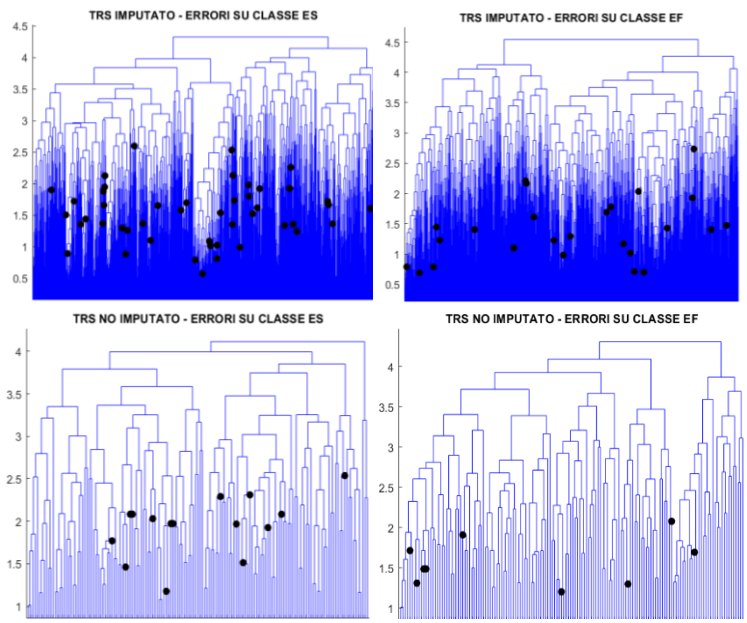


Figura 27 ***Posizionamento degli errori sui rispettivi dendrogrammi***
In figura sono rappresentati i dendrogrammi delle quattro sottoclassi EF/ES imputato e non imputato. Come si può notare dalla figura non esiste una classe più affetta da errori, le classi con un maggior numero di elementi presentano anche un numero maggiore di errori.

In generale non si è notata una maggior incidenza degli errori in una delle sottoclassi.
La posizione dei marcatori, posti sul punto di unione dell'elemento errato a quello ad esso più vicino, assume valori di distanza piuttosto elevata e variabile.
Inoltre, nessun cluster di quelli utilizzati in fase di costruzione del Training Set è risultato particolarmente affetto dalla presenza di errori; considerazione che trova giustificazione nell'attenzione prestata al costruire cluster con variabilità il più simile possibile.
Medesime considerazioni si applicano ai dendrogrammi ottenuti sul Test set.

Un' ulteriore analisi sugli errori è stata effettuata valutando gli errori condivisi da coppie di soluzioni e l'indice di similarità tra i vettori di errori.

L'indice di similarità valuta la somiglianza tra vettori ed è stato sfruttato per valutare la somiglianza tra vettori contenenti gli indici degli elementi classificati erroneamente da ciascuna delle reti DNN ottenute a seguito delle ottimizzazioni. L'indice di similarità calcolato nel presente progetto è noto come indice di Sarnese, quantifica la somiglianza tra comunità effettuando il rapporto tra il numero di elementi condivisi tra le comunità e il numero di elementi presenti in ciascuna.

$$Cs = \left(\frac{2 * c}{S1 + S2}\right)$$

c=numero di specie in comune alle due comunità
S1=numero di specie appartenenti alla comunità1
S2=numero specie appartenenti alla

L'indice di similarità ha valori compresi tra 0 e 1. Valori vicini all'unità indicano una maggior somiglianza tra i vettori di errori.
Il calcolo dell'indice è stato effettuato per ogni possibile coppia di reti. Il risultato è una matrice quadrata, la cui diagonale ha valore unitario in quanto si ha il confronto tra il vettore degli errori di una rete con sé stesso.
In generale la similarità tra le soluzioni è piuttosto elevata. Le valutazioni effettuate sulle reti ottimizzate del primo algoritmo genetico indicano un valore medio dell'indice di similarità tra coppie di classificatori di 0,55 con deviazione standard di 0,07. Circa il 60% dei confronti ha valore maggiore o uguale alla media, ne consegue che la maggior parte delle reti presenta un indice di similarità con le restanti superiore o uguale a circa il 50%.
In *Figura 28* viene raffigurata la matrice di similarità ottenuta selezionando per ogni soluzione il valore minimo. In questo modo si sono individuate le soluzioni che differiscono maggiormente.

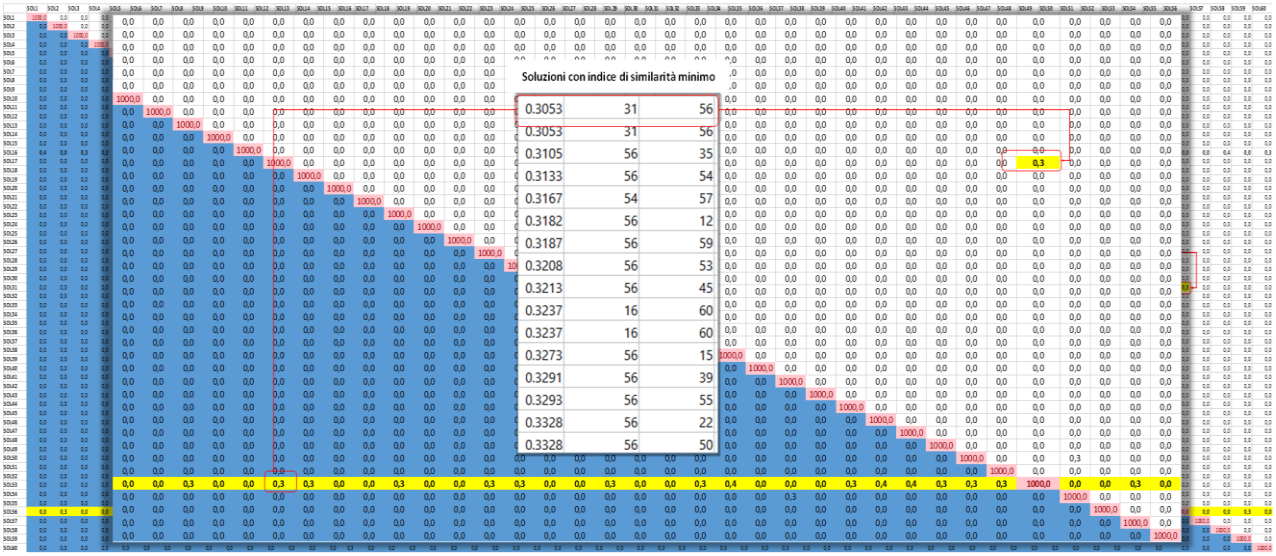


Figura 28 ***Matrice di similarità degli errori delle 60 soluzioni***
In figura viene riportata la tabella contenente per ogni soluzione il valore dell'indice di similarità minore rispetto alle altre soluzioni ottenute. In questo caso viene riportata la tabella relativa alle 60 soluzioni ottenute dai GA con comorbidità. Si tratta di un matrice quadrata che riporta sulla diagonale il confronto della soluzione con sé stessa e quindi indice di similarità unitario.

Si può facilmente osservare che molti dei valori minimi individuati si trovano sulla stessa riga (evidenziata in colore giallo). Si identifica quindi una classificazione che differisce particolarmente dalle restanti: rispetto ai valori di indice similarità ottenuto dal confronto tra tutte le restanti soluzioni, la soluzione 56 ha valore minimo dell'indice di similarità rispetto allo stesso indice

ottenuto in riferimento alle altre soluzioni, e ciò viene confermato dalla tabella riportata sull'immagine, dove si può notare la ricorrenza della classificazione 56 in numerose coppie.
Il valore minimo in assoluto è individuato in corrispondenza delle classificazioni 56 e 31; i vettori degli indici di queste due classificazioni sono risultati dunque essere i meno simili in assoluto.

L'analisi degli elementi in comune alle soluzioni ha portato ad avere una matrice simile, si tratta ancora di una matrice quadrata che tuttavia riporta per ogni confronto, il numero di elementi in comune classificati. I risultati portano a considerazioni paragonabili a quelle già effettuate per la matrice di similarità. In media le reti producono circa 732 errori sul Training set, valore che

corrisponde a circa il 30% degli errori totali (2302 elementi). Le reti condividono in media circa 402 elementi la cui classificazione risulta errata; la metà dei confronti tra le classificazioni condivide un numero di errori uguale o superiore alla media.

Infine, le ultime valutazioni effettuate sugli errori riguardano il loro valore di classificazione.
Il neurone di output delle reti neurali fornisce per ciascun elemento un valore compreso tra 0 e 1 che consente tramite sogliatura di ottenere la classificazione binaria (0 e 1) con la conseguente associazione alla rispettiva classe. La soglia delle DNN è individuata sfruttando la curva ROC sul Training set, come descritto in precedenza. L'analisi del valore di classificazione fornito all'ambiente dal neurone di output consente di riconoscere la natura dell'errata classificazione degli elementi. La classificazione errata può essere dovuta a dei valori di output prossimi al valore della soglia stessa; conseguentemente una minima modifica di quest'ultima potrebbe portare ad una classificazione corretta. In realtà lo studio degli errori ha evidenziato che la maggior parte degli elementi classificati erroneamente non presenta valori prossimi alla soglia, ma altresì piuttosto dispersi. Il valore soglia non è quindi responsabile dell'elevato numero di elementi classificati erroneamente.

In *Figura 29* è possibile osservare i valori di predizione degli elementi classificati errati sul Training set, utilizzato in fase di ottimizzazione, da due DNN ottenute dalle soluzione del GA con comorbidità.

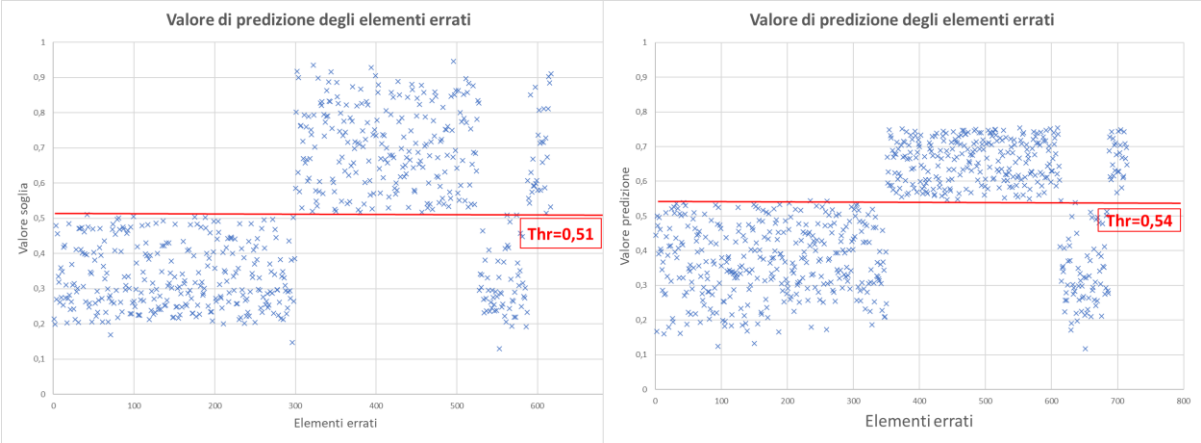


Figura 29 ***Valore assegnato dal valore di output degli elementi classificati erroneamente***
In figura viene rappresentato il valore assegnato ad ogni elemento del Training Set classificato erroneamente da due reti ottenute dalle soluzioni del primo GA. In rosso è evidenziato il valore soglia che consente di discriminare la classe mentre il valore di ogni elemento è indicato tramite una croce. Il numero di elementi con valore prossimo alla soglia è minimo, pur modificando tale valore non si otterrebbero significativi miglioramenti.

4.2.4 Decisioni in merito al majority voting

L'analisi delle prestazioni dei classificatori ottenuti dalle soluzioni dell'algoritmo di ottimizzazione ha indotto all'esplorazione di una nuova strada al fine di ottenerne un miglioramento. Le prestazioni delle singole reti hanno mostrato un comportamento sub-ottimale, con i valori di specificità e sensibilità che in pochi casi hanno raggiunto un valore sopra lo 0.70 sul Test set. In generale le Dense Neural Network hanno mostrato un andamento speculare tra sensibilità e specificità.
Si è quindi deciso di valutare la possibilità di ottenere dei miglioramenti nelle prestazioni effettuando un Majority Voting attraverso la combinazioni di un numero variabile di reti.
La costruzione del Majority Voting è semplice: a ciascun elemento è stata assegnata la classe maggiormente presente nelle classificazioni selezionate tra quelle delle reti ottenute a seguito delle ottimizzazioni. (*Figura 30*)

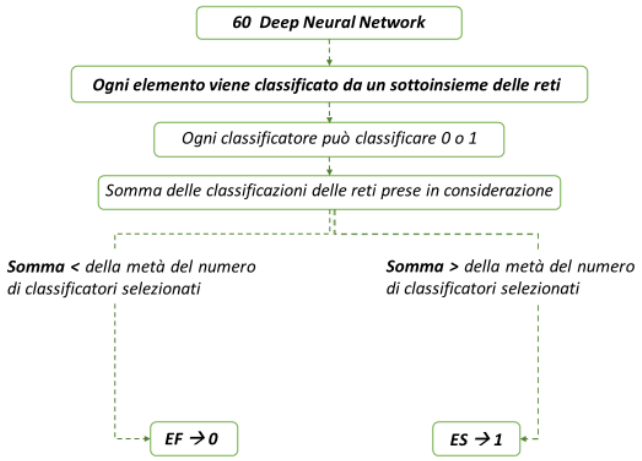


Figura 30 ***Schema di funzionamento del Majority Voting implementato***
Vengono sommate le classificazioni del medesimo elemento da parte di n classificatori, se la maggioranza delle DNN classifica l'elemento con output 1 allora la classe sarà ES altrimenti l'elemento verrà classificato in classe EF.

Le reti che hanno determinato le classificazioni su cui effettuare il Majority Voting sono state ottenute sfruttando i due indicatori definiti nella sezione precedente. In un primo caso l'informazione contenuta nella matrice delle similarità è stata utilizzata per individuare le classificazioni i cui vettori di errori erano più dissimili; in una seconda implementazione si è invece sfruttata la matrice contenente il numero degli errori condivisi dalle reti sul Training set, così da selezionare, almeno inizialmente le reti con il minor numero di errori condiviso.
Il Majority voting per essere significativo necessita di un numero dispari di classificazioni, così da evitare il rischio di un egual numero di classi e conseguentemente una decisione finale che risulta essere randomica o a favore della classe la cui condizione costituisce una maggior sicurezza per il paziente.
Nei brevi paragrafi a seguire si scenderà maggiormente nel dettaglio dell'implementazione del Majority Voting secondo le due vie già anticipate in precedenza.

Majority voting tramite la matrice di similarità

Sfruttando l'informazione relativa all'indice di similarità tra coppie di classificazioni si è inizialmente individuata la coppia di reti il cui indice reciproco è risultato il minimo assoluto.
Si è quindi calcolata la somma degli indici di similarità delle restanti reti rispetto a quelle già selezionate. In tal modo si è riusciti, tramite l'individuazione dell'indice minore, a identificare una terza rete. Si è proceduto iterativamente, secondo la procedura descritta, fino alla selezione di tutte le reti e si sono effettuate le valutazioni in termini di specificità e sensibilità sulle composizioni con un numero dispari di reti.

Majority voting tramite intersezione degli errori

In questo secondo caso, si è valutata l'entità dell'intersezione degli errori effettuati sul Training set (comune a tutti) di tutte le possibili coppie di classificatori. Per ciascuna coppia si è ottenuto il numero di elementi classificati erroneamente da entrambe le reti e i valori sono stati organizzati in una matrice quadrata, come descritto in precedenza.
Si è quindi proceduto individuando le reti che presentavano il numero inferiore in assoluto di elementi in comune.
Come nel caso precedente la classificazione tramite Majority Voting richiede un numero dispari di soluzioni; si è quindi proseguito individuando la terza rete e le successive.
La selezione delle ulteriori reti è stata effettuata secondo il seguente criterio: è stato creato un nuovo vettore di errori, ottenuto dall'unione dei vettori delle prime due reti selezionate, contenente dunque sia gli elementi comuni ad entrambe le soluzioni sia quelli che differiscono; si è quindi intersecato questo nuovo vettore con quelli delle restanti reti e si è individuata la rete che presentava l'intersezione minima rispetto al vettore unione delle prime due.
Si sono effettuati i passaggi descritti fino a includere nella composizione tutte le reti a disposizione e come già descritto in precedenza, si è effettuata la valutazione di specificità e sensibilità sulle composizioni in cui il numero di reti presenti fosse dispari.

Analisi dei risultati del Majority Voting

I risultati ottenuti tramite il Majority Voting sono stati valutati in termini di specificità e sensibilità, rendendo possibile sia il confronto tra le due metodologie applicate per ottenere il Majority Voting, sia il confronto con le prestazioni delle singole reti.

Le due implementazioni di Majortiy descritte in precedenza, nel caso delle sessanta soluzioni ottenute dal GA con comorbidità, hanno dato esito alla costruzione di 29 composizioni totali. Nella figura a seguire (*Figura 31*) sono riportati i risultati ottenuti a partire dalle due reti con intersezione minima.

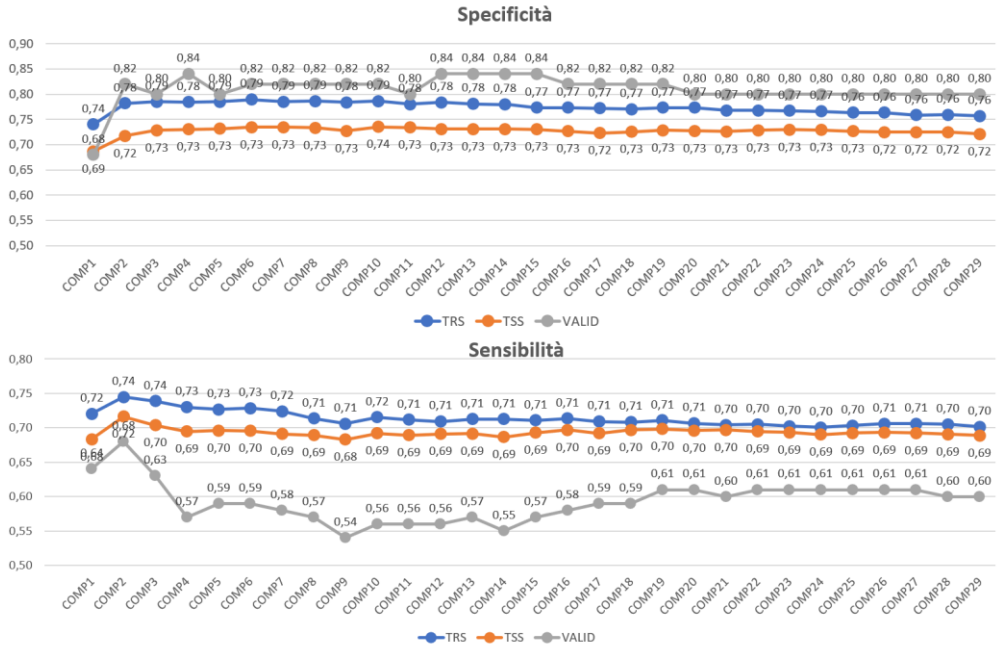


Figura 31 ***Valutazioni su 29 composizioni di Majority Voting ottenute a partire dalle due reti con intersezione minore***
Le prestazioni mostrano valori di specificità e sensibilità simili e uno scarto tra Training Set e Test Set di pochi punti. I risultati sul Validation Set mostrano invece valori di specificità migliori anche di quelli ottenuti sul Training Set, riflettendosi però in prestazioni decisamente scarse in termini di sensibilità.
Prestazioni simili sono state ottenute anche nel caso del majority voting con composizioni ottenute tramite indice di similarità.

Dai grafici della *Figura 31* si può notare un iniziale miglioramento delle prestazioni nelle prime composizioni, sia in termini di specificità che di sensibilità, seguito da una successiva stabilizzazione. Nel caso della specificità il valore tende a stabilizzarsi sull'incremento ottenuto nelle prime composizioni, mentre sulla sensibilità si ha un decremento sul Training di qualche centesimo set; il Test set sembra invece mantenere un andamento più costante. Si riscontra quindi sul parametro di sensibilità una riduzione dello scarto tra le prestazioni di questi due set di dati. Il parametro di specificità presenta invece uno scarto costante di 4-5 centesimi.
Le prestazioni del Validation set sono caratterizzate da una specificità con valori al di sopra dello 0,80 per la quasi totalità delle composizioni. Specularmente la sensibilità ha valori significativamente inferiori a quelli ottenuti su Training e Test set.
Ciò che si è potuto inoltre osservare è il maggior equilibrio tra specificità e sensibilità ottenuto tramite il Majority Voting. Su Training e Test set i due parametri, seppur individualmente non raggiungono le prestazioni ottenute dalle singole reti, presentano valori più simili. Le prestazioni singole hanno evidenziato andamenti speculari tra sensibilità e specificità: laddove la specificità raggiunge valori buoni, considerando le prestazioni generali, la sensibilità ha uno scarto di parecchi centesimi e viceversa; un maggior equilibrio si ha invece in quelle classificazioni in cui la prestazione risulta scarsa. L'osservazione dei grafici permette di notare come su Training e Test set i valori di specificità e sensibilità differiscano di qualche centesimo. Ciò, tuttavia, non avviene sul Validation set, dove lo scarto è decisamente maggiore.

Così come già notato nel caso delle reti singole, anche tramite le tecniche di Majority Voting si osserva il mantenimento delle prestazioni sul Test set rispetto al Training set.
Le prestazioni così ottenute non hanno mostrato miglioramenti significativi e il fatto che il Majority Voting non porti a un incremento o decremento delle prestazioni all'aumentare delle composizioni è sintomo del fatto che gli elementi, classificati erroneamente, siano critici per la maggior parte delle reti.
I grafici commentati si riferiscono all'implementazione del Majority Voting tramite valutazione dell'intersezione minima, i risultati ottenuti sfruttando l'indice di similarità sono del tutto paragonabili.

Medesime valutazioni sono state effettuate sulle cinquanta soluzioni ottenute dalla seconda ottimizzazione, effettuata su Training, Test e Validation set ottenuti dalla codifica delle comorbidità. Le composizioni in questo caso sono 24 (*Figura 32*). Gli andamenti in termini di specificità e sensibilità mostrano lo stesso comportamento evidenziato in precedenza: all'aumentare delle composizioni le prestazioni in termini di specificità e sensibilità tendono allo stesso valore, assestandosi in entrambi i casi a circa 0,72. Diverso è invece il comportamento sul Validation Set che presenta in termini di specificità prestazioni positive, che raggiungono un valore di 0,88 sulle composizioni con un maggior numero di reti, ma che al tempo stesso mostra risultati significativamente peggiori in termini di sensibilità, con valori che si aggirano attorno a 0,60-0,65.
Le prestazioni delle implementazioni del Majority Voting sulle reti ottenute dalla seconda ottimizzazione sembrano godere di una minor variabilità e il plateau viene raggiunto già a partire dalle prime composizioni su entrambi i parametri di specificità e sensibilità. Il Validation set è caratterizzato da maggior stabilità soprattutto sul parametro di sensibilità.
I valori riportati si riferiscono alle valutazioni di Majority effettuate tramite l'impiego della matrice di similarità, ma conclusioni del tutto sovrapponibili vengono tratte nel caso del metodo con intersezione minima.

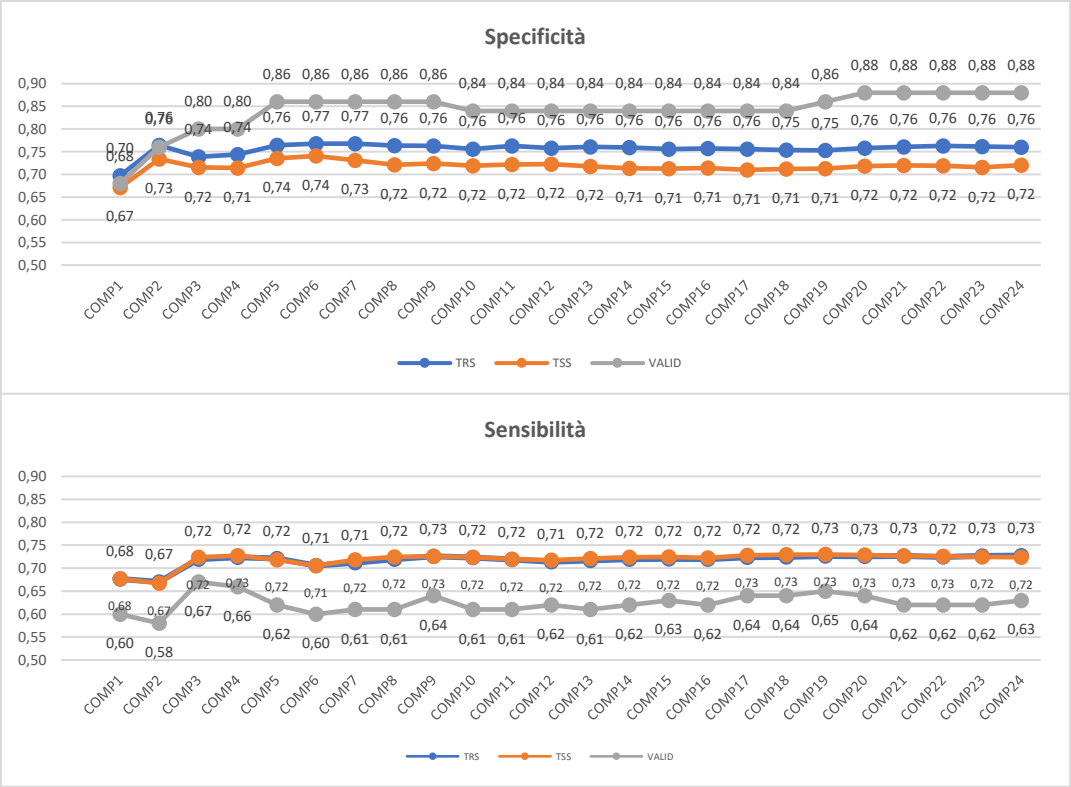


Figura 32 ***Valutazioni su 29 composizioni di Majority Voting ottenute sfruttando l'indice di similarità***
Il miglioramento di prestazioni iniziali è meno evidente rispetto al caso precedente e i valori raggiungono presto una condizione stabile. Anche il Validation set mostra meno variazioni sulle composizioni rispetto al caso precedente.

4.3 DNN: valutazioni di strutture di maggior complessità

Le reti DNN descritte nei paragrafi precedenti e le loro prestazioni hanno indotto ad ipotizzare la necessità di testare reti ancora più complesse. Le prestazioni non eccellenti riscontrate nelle fasi precedenti sono probabilmente legate alla natura dei dati, complessi e difficilmente gestibili da reti con un numero di neuroni e di hidden layer quale quello impiegato in precedenza.

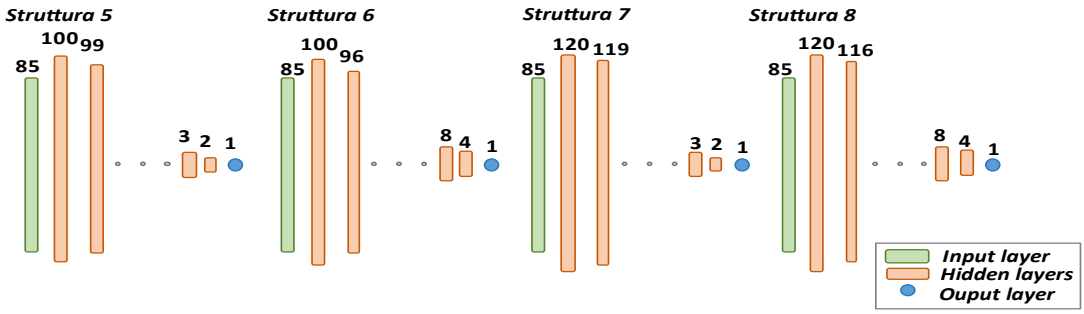


Figura 33 ***Quattro nuove strutture di DNN***
In figura sono riportate le schematizzazioni delle strutture di reti di Deep Learning implementate; presentano il layer di input con dimensione di 85 neuroni per poi proseguire con un primo hidden layer di 100 o 120 neuroni per poi decrescere nei layer successivi a passi di 1 e 4 neuroni; infine, le reti presentano un layer di output costituito da un neurone.

Si è quindi proceduto con la valutazione di quattro ulteriori strutture, che, a differenza delle precedenti, presentavano un primo hidden layer con un numero di neuroni superiore a quello presente nel layer di input.
Le reti così costruite sono state testate sia sulle coppie Training e Test Set con 85 features, sia sulle medesime coppie ottenute a seguito della codifica decimale delle variabili di comorbidità.

Le quattro nuove strutture implementate, mostrate in figura (*Figura 33*), presentano un primo hidden layer di 100 o 120 neuroni, mentre i successivi hidden layer presentano un decremento di 1 o 4 neuroni rispetto al precedente.
Come nelle prime valutazioni delle reti, descritte nel paragrafo 4.1 DNN Implementation, si sono valutate diverse funzioni di allenamento, tra le quali si è scelto di utilizzare la funzione 'scg' con l'aggiunta di una soglia al 10% di corretti classificati sulle classi, scelta necessaria per far fronte all'instabilità che reti così complesse presentano.
Si è effettuata nuovamente la valutazione della rete utilizzando la funzione ReLu agli hidden layer, ma come nelle valutazioni precedenti questa scelta non porta a classificazioni efficaci in quanto sbilanciate nei confronti di una delle classi. In ultimo le strutture sono state valutate mediante un Majority Voting su cinque prove per ciascuna delle coppie Training e Test set, sia su quelle con le comorbidità sia su quelle ottenute a seguito della codifica decimale, se ne riportano in seguito i risultati ottenuti tramite Confusion Matrices (*Figura 34*).
Le prestazioni hanno evidenziato un sostanziale limite nella costruzione delle reti: il software di implementazione Matlab non riesce a gestire l'allenamento tramite la funzione di default 'trainml' per reti così complesse, di fatto obbligando alla scelta di una diversa funzione di allenamento; quindi, nelle presenti prove è stata infatti impostata la funzione 'scg'. Tuttavia, le reti mostrano un comportamento anomalo, portando a classificazioni migliori sulle reti meno complesse, mentre decisamente più sbilanciate verso una delle classi nei restanti casi. Questi risultati sono inaspettati in quanto da reti con un numero così elevato di neuroni e hidden layer è più comune incorrere in fenomeni di overfitting, avendo un eccessivo adattamento nei confronti dei dati di allenamento; al contrario le prestazioni su Training e Test set sono equiparabili.

La piattaforma di Matlab non sembra quindi essere adatta all'implementazione di reti di tale complessità e non si sono quindi effettuate ulteriori valutazioni su queste strutture.

MAJORITY SU 5 PROVE SULLA STESSA STRUTTURA

TRS

DEC1_100	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	46%	34%
ES	54%	66%

DEC1_120	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	67%	57%
ES	33%	43%

DEC4_100	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	78%	37%
ES	22%	63%

DEC4_120	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	74%	32%
ES	26%	68%

TSS

DEC1_100	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	42%	35%
ES	58%	65%

DEC1_120	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	64%	58%
ES	36%	42%

DEC4_100	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	73%	39%
ES	27%	61%

DEC4_120	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	69%	35%
ES	31%	65%

MAJORITY SU 5 PROVE SULLA STESSA STRUTTURA

TRS

DEC1_100	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	41%	31%
ES	59%	69%

DEC1_120	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	41%	33%
ES	59%	67%

DEC4_100	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	73%	29%
ES	27%	71%

DEC4_120	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	79%	39%
ES	21%	61%

TSS

DEC1_100	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	41%	34%
ES	59%	66%

DEC1_120	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	37%	38%
ES	63%	62%

DEC4_100	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	67%	30%
ES	33%	70%

DEC4_120	CLASSE REALE	
CLASSE PREDETTA	EF	ES
EF	76%	40%
ES	24%	60%

Figura 34 ***Confusion matrices del Majority voting su 5 prove Valutazione delle quattro strutture sulla quarta coppia Training e Test*** *set. A sinistra si possono osservare le valutazioni del Majority Voting sulla quarta coppia Training e Test set con 85 features a destra invece le medesime valutazioni ottenute sulla codifica decimale della coppia Training e Test set.*

5. DISCUSSIONE E CONCLUSIONI

Il presente progetto di tesi nasce dalla necessità di costruire un classificatore in grado di aiutare l'operatore sanitario nella decisione di estubare il paziente dalla ventilazione meccanica. Il fine è di evitare il prolungamento del supporto alla ventilazione oltre al tempo necessario e contemporaneamente di evitare l'incorrere in situazioni di fallimento che renderebbero necessaria la re intubazione determinando un aumento notevole della probabilità di morte.
I classificatori utilizzati nell'attuale lavoro appartengono a due categorie distinte. il primo classificatore è il LightGBM, il quale, nello studio effettuato nella tesi precedente, ha garantito le prestazioni migliori. Si tratta di un classificatore appartenente alla famiglia dei Random Forest che nel presente progetto si è distinto per stabilità in termini di risultati, in quanto seppur allenato e testato su coppie di Training e Test set differenti, ha mostrato performances simili; inoltre, la valutazione sul Validation set, unico per tutti i modelli, ha mostrato che anche le classificazioni errate sono comuni: non si sono infatti ottenuti sostanziali miglioramenti nell'utilizzo di un Majority Voting sui cinque modelli ottenuti, condizione indicativa del fatto che gli elementi classificati fossero in maggior parte gli stessi tra i cinque classificatori LightGBM.
Con il secondo modello di classificatore utilizzato ci si è rivolti al ramo del Deep Learning.
Si sono costruite alcune strutture di reti neurali dense (DNN) che presentano un elevato numero di hidden layer con un altrettanto elevato numero di neuroni. Reti di complessità elevata dovrebbero essere in grado di riconoscere relazioni tra le features che reti più semplici non riescono ad individuare. Si sono quindi inizialmente testate diverse strutture con complessità differente; tuttavia, per limiti computazionali dei dispositivi utilizzati, una sola delle quattro strutture valutate in un primo momento è stata sottoposta a features selection.
Lo scopo è stato non solo di individuare la struttura in grado di garantire prestazioni migliori, ma anche di valutare quali features sono effettivamente più adatte alla classificazione della condizione del paziente in una delle due classi ES o EF.
Infine, si sono valutate le reti ottimizzate mediante la tecnica del Majority Voting, che consente di associare ogni paziente alla classe maggiormente presente nelle classificazioni ottenute attraverso un numero di reti variabile.
La valutazione iniziale di idoneità all'estubazione viene effettuata tramite la valutazione del parametro di Rapid Shallow Breathing Index (RSBI) calcolato come frequenza respiratoria/ Volume (Tidal). Il parametro RSBI è in grado di predire un'estubazione eseguita con successo con una sensibilità del 97% e specificità del 65%. Si tratta di un parametro altamente sensibile, ma scarsamente specifico che riscontra maggiori difficoltà nell'individuazione dei possibili fallimenti dei tentativi di estubazione. Risulta quindi di primaria importanza individuare un classificatore con un'elevata specificità che possa essere impiegato come perfezionamento delle prime valutazioni con l'indice RSBI dall'esito positivo.

I classificatori costruiti sono quindi stati valutati in termini di specificità e sensibilità. Il calcolo di questi indicatori richiede la conoscenza di alcuni elementi:

- ***TP*** (veri positivi): il numero di elementi realmente di classe ES che vengono classificati nella classe delle estubazioni riuscite (ES)
- ***TN*** (veri negativi): il numero di elementi realmente di classe EF che vengono classificati nella classe delle estubazioni fallite (EF).

- ***FP*** (falsi positivi): il numero di elementi la cui classe reale di appartenenza è quella delle estubazioni fallite (EF), ma che vengono classificati nella classe ES
- ***FN*** (falsi negativi): il numero di elementi la cui classe reale di appartenenza è quella delle estubazioni riuscite (ES), ma che vengono classificati nella classe EF

Il parametro ***specificità*** indica la capacità del classificatore di identificare correttamente i pazienti che appartengono alla classe negativa; in questo caso la specificità consente di identificare i soggetti che subiranno il fallimento del tentativo di estubazione. Maggiore è il valore di specificità, minore è la probabilità che un paziente venga ritenuto idoneo allo svezzamento per poi incorrere in una successiva re intubazione.
Il parametro di specificità viene calcolato come segue:

$$\boldsymbol{Specificity}: \frac{\boldsymbol{TN}}{\boldsymbol{FP+TN}}$$

Il parametro ***sensibilità*** indica invece la capacità del classificatore di identificare correttamente i pazienti della classe positiva; nel presente progetto un valore elevato di sensibilità significa saper individuare correttamente i pazienti che subiranno un tentativo di estubazione con esito positivo.
Tale parametro viene calcolato come segue:

$$\boldsymbol{Sensitivity}: \frac{\boldsymbol{TP}}{\boldsymbol{FN+TP}}$$

Tabella 7. Prestazioni dei classificatori testati

	PRESTAZIONI DEI MODELLI IMPLEMENTATI			
	CLASSIFICATORI		***SENSITIVITY***	***SPECIFICITY***
LightGBM 85 features	*Modello TRSTSS1*	**TRS**	1,00	1,00
		TSS	0,77	0,79
		VALID	0,75	0,90
	Modello TRSTSS2	**TRS**	1,00	1,00
		TSS	0,71	0,85
		VALID	0,64	0,94
	Modello TRSTSS3	**TRS**	1,00	1,00
		TSS	0,76	0,81
		VALID	0,69	1,00
	Modello TRSTSS4	**TRS**	0,99	1,00
		TSS	0,75	0,83
		VALID	0,68	1,00
	Modello TESIPRECEDENTE	**TRS**	1,00	1,00
		TSS	0,78	0,78
		VALID	0,72	0,88
	Modello num_leaves_10_100	**TRS**	1,00	1,00
		TSS	0,75	0,82
		VALID	0,65	0,92
	Modello min_leaves_60_100	**TRS**	0,97	0,97
		TSS	0,78	0,78
		VALID	0,73	0,88
	Modello min_leaves_10_100	**TRS**	0,97	0,99
		TSS	0,77	0,80
		VALID	0,71	0,88
LightGBM 58 features	*Modello TRSTSS4 58*	**TRS**	0,94	0,97
		TSS	0,68	0,79
		VALID	0,61	0,86

Reti DNN 85 features	*60 reti*		**RANGE**	**RANGE**
		TRS	[0,78 - 0,46]	[0,82 - 0,29]
		TSS	[0,78 - 0,44]	[0,78 - 0,28]
		VALID	[0,77 - 0,40]	[0,84 - 0,24]
	RISULTATO MIGLIORE			
	Rete 32	**TRS**	0,63	0,81
		TSS	0,61	0,75
		VALID	0,51	0,84

Reti DNN 58 features	*50 reti*		**RANGE**	**RANGE**
		TRS	[0,78 - 0,52]	[0,79 - 0,53]
		TSS	[0,78 - 0,53]	[0,75 - 0,50]
		VALID	[0,73 - 0,38]	[0,92 - 0,54]
	RISULTATO MIGLIORE			
	Rete 17	**TRS**	0,69	0,74
		TSS	0,69	0,73
		VALID	0,63	0,88
Majority Reti DNN 85 features	*Intersezione Minima*		**RANGE**	**RANGE**
		TRS	[0,74 - 0,70]	[0,79 - 0,74]
		TSS	[0,72 - 0,69]	[0,74 - 0,69]
		VALID	[0,68 - 0,54]	[0,84 - 0,69]
	RISULTATO MIGLIORE			
	COMPOSIZIONE 6-12-13	**TRS**	0,72- 0,71-0,71	0,79 – 0,78 – 0,78
		TSS	0,69-0,69-0,69	0,73 – 0,73 – 0,73
		VALID	0,58 -0,56- 0,57	0,84 – 0,84- 0,84
	Matrice Similarità		**RANGE**	**RANGE**
		TRS	[0,73 - 0,70]	[0,78 - 0,57]
		TSS	[0,71- 0,67]	[0,74 - 0,53]
		VALID	[0,69 - 0,59]	[0,82 - 0,52]
	RISULTATO MIGLIORE			
	COMPOSIZIONE 29	**TRS**	0,70	0,78
		TSS	0,69	0,74
		VALID	0,61	0,82
Majority Reti DNN 58 features	*Intersezione Minima*		**RANGE**	**RANGE**
		TRS	[0,74 - 0,72]	[0,78 - 0,74]
		TSS	[0,74 - 0,71]	[0,74 - 0,72]
		VALID	[0,65 - 0,57]	[0,90 - 0,86]
	RISULTATO MIGLIORE			
	COMPOSIZIONE 3 -12	**TRS**	0,72-0,78	0,74-0,74
		TSS	0,71-0,74	0,74-0,72
		VALID	0,60-0,90	0,65-0,88
	Matrice Similarità		**RANGE**	**RANGE**
		TRS	[0,73 - 0,67]	[0,77 - 0,70]
		TSS	[0,73 - 0,67]	[0,74 - 0,67]
		VALID	[0,67 - 0,58]	[0,88 - 0,68]
	RISULTATO MIGLIORE			
	COMPOSIZIONE 24	**TRS**	0,73	0,76
		TSS	0,72	0,72
		VALID	0,63	0,88

I risultati riportati in tabella permettono di giungere a conclusioni di diversa natura.

I classificatori LightGBM mostrano risultati incoraggianti, raggiungendo valori di specificità decisamente elevati e in tutti i casi al di sopra dell'80% sul Validation Set. È tuttavia però evidente la problematica che li affligge: i modelli mostrano di essere affetti da overfitting, presentando prestazioni ottime sul Training set, ma decisamente peggiori sui Test e sul Validation Set, sintomo di una capacità di adattamento perfetta ai dati su cui viene allenato, ma associata ad una peggior capacità di generalizzazione. Tale problematica è riconducibile alla scarsa dimensionalità dei dati in

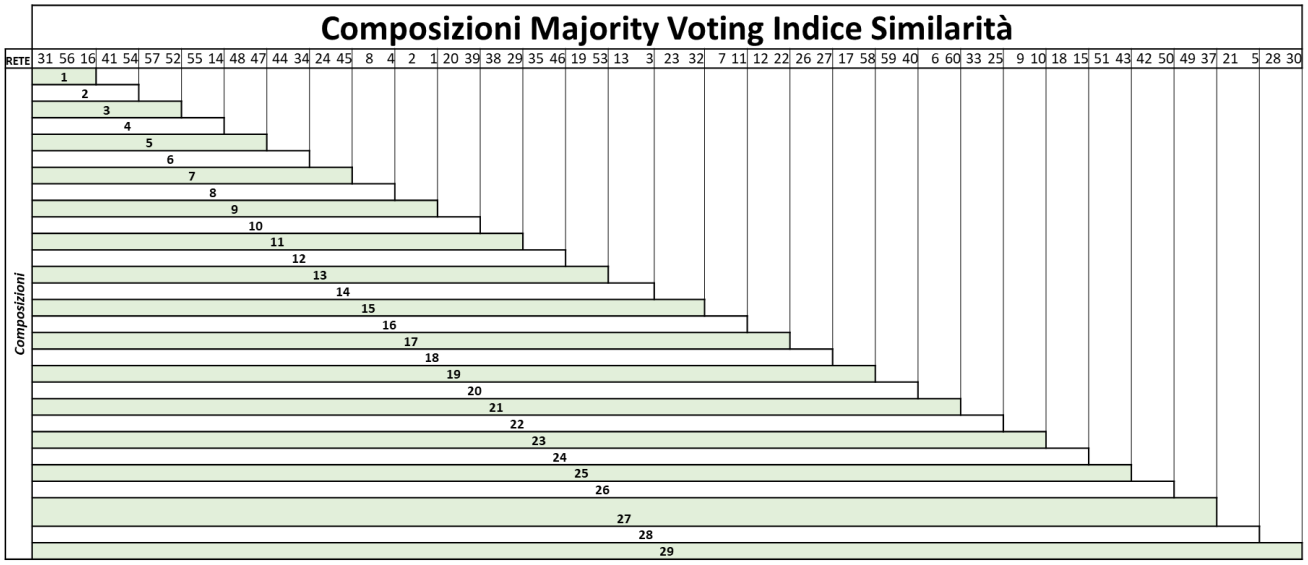
quanto, pur effettuando il tuning dei parametri più sensibili in diversi range di valori, non si è potuto in alcun modo evitare la criticità.
I diversi modelli di classificatore LightGBM non differiscono notevolmente in termini di risultati e anche laddove i modelli hanno subito il tuning dei parametri in range diversi mostrano un comportamento comune.
Le reti DNN soffrono invece di una maggior instabilità. In questo caso in tabella vengono riportati i range di variazione dei parametri. I valori massimi e minimi riportati suggeriscono la presenza di un ampio spettro di valori, di fatto mostrando come l'ottimizzazione abbia portato alla creazione di reti che generano risultati estremamente diversificati.
I risultati della rete migliore, selezionata come buon compromesso tra alti valori di specificità e i valori di sensibilità, mostrano ancora una volta valori di specificità sul Validation set buoni, con valori sul Training e Test set più bassi del caso precedente. Non si osserva però il fenomeno dell'overfitting.
Le reti ottenute dall'ottimizzazione delle strutture sulla coppia Training e Test set su cui è stata effettuata la codifica decimale presentano un range di variazione dei parametri ridotto rispetto al precedente, ma comunque importante, sintomo anche in questo caso di una notevole varietà tra le reti. La rete migliore individuata mostra valori simili a quelli ottenuti con i modelli LightGBM, ma non risulta affetta dalla problematica dell'overfitting.
Le valutazioni effettuate sui Majority Voting non mostrano un significativo miglioramento sui singoli parametri di specificità e sensibilità, ma si possono notare range di variazione più contenuti rispetto alle reti singole. Inoltre, la scelta di effettuare il Majority Voting tra le reti porta ad un avvicinamento su Training e Test set, dei valori di specificità e sensibilità. Al contrario le prestazioni delle singole reti mostrano comportamenti speculari tra questi due parametri. Le due tipologie di Majority sono state implementate sia sulle reti ottenute dal primo GA implementato sulla coppia Training e Test con 85 variabili sia sulle reti ottenute dal secondo GA implementato sulla coppia con 58 variabili, ottenuta a seguito della codifica decimale. I Majority Voting ottenuti sulle reti della seconda ottimizzazione sono caratterizzati da prestazioni più stabili tra le composizioni e garantiscono sul Validation Set prestazioni in termini di specificità migliori, paragonabili a quelle ottenute sul LightGBM, mentre in termini di sensibilità le prestazioni sono simili sulle due ottimizzazioni.
Una seconda valutazione emerge dall'osservazione dei risultati e riguarda le differenze nelle prestazioni dei classificatori costruiti sulle coppie Training e Test set con 85 variabili rispetto a quelle costruite sulle coppie Training e Test Set con 58 variabili. È difficile individuare differenze significative, in quanto il modello di classificatore LightGBM costruito e valutato sui dataset con 58 variabili mostra risultati del tutto in linea con quelli degli altri modelli. Le reti neurali mostrano invece un range più contenuto, ma analizzando i risultati singolarmente è difficile affermare che questa differenza possa essere legata alla codifica e non alla natura randomica dell'allenamento. È tuttavia possibile affermare che la codifica decimale delle comorbidità determini risultati di specificità maggiori, sia sulle reti singole che sui Majority voting.
Di fatto le variabili di comorbidità che vengono sostituite dalla codifica decimale sono di natura binaria, non riportano quindi un valore di gravità e dunque la sostituzione con una variabile codificante in numeri decimali porta ad un'informazione del tutto simile. Come si è potuto osservare nei grafici di selezione delle variabili, le comorbidità risultate maggiormente selezionate sono identiche sia nel caso di features selection delle 85 variabili che nel caso selection delle 58 variabili. La codifica binaria consente però di ridurre il numero di variabili e conseguentemente anche la complessità e i tempi di addestramento dei classificatori.

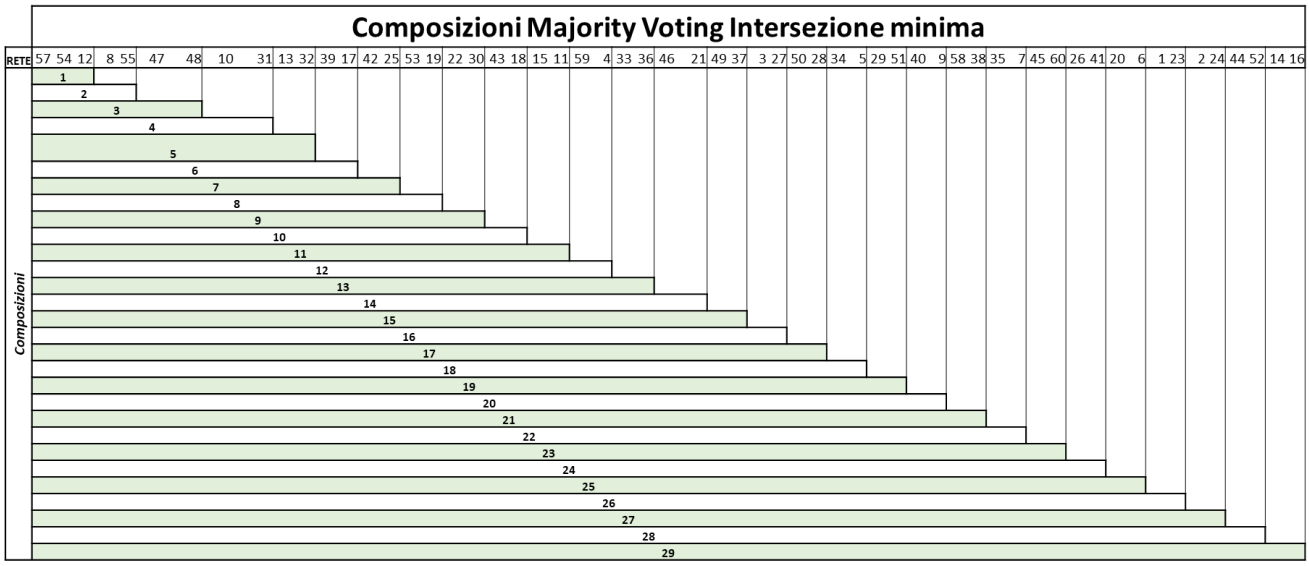
Infine, un'ultima valutazione riguarda l'utilizzo di diverse coppie Training e Test set per la costruzione dei modelli; In tabella si può infatti osservare come i cinque modelli di LigthGBM costruiti su coppie Training e Test differenti portino a risultati del tutto simili, medesima considerazione è stata effettuata in fase di test delle prime reti DNN e pertanto si è scelto di effettuare la successiva ottimizzazione utilizzando una singola coppia Training e Test delle cinque create. Le coppie di Training e Test set sembrano quindi non influire significativamente sull'addestramento dei modelli, di fatto garantendo una buona rappresentazione dei dati.
In conclusione, i classificatori e i metodi testati consentono di ottenere valori di specificità maggiori di quelli ottenuti dall'uso dell'indice RSBI, fattore di primaria importanza vista l'intenzione di utilizzare i modelli in cascata all'indice precedentemente citato, che difetta proprio di questa caratteristica; tuttavia, il classificatore LightGBM sembra assicurare i risultati migliori in quanto non solo consente di ottenere una maggior specificità, ma è dotato anche di una buona stabilità.
Le DNN mostrano in generale un buon potenziale; la loro ottimizzazione è stata effettuata tramite un algoritmo semplificato rispetto a quello inizialmente ipotizzato, che conseguentemente ha portato ad una peggior gestione dell'instabilità dell'allenamento e alla valutazione di un sottogruppo di possibili strutture. In ottica futura la possibilità di ottimizzare più strutture secondo un'implementazione più robusta potrebbe portare all'individuazione di reti più stabili e performanti; inoltre, si potrebbe ricorrere alla codifica decimale delle variabili di comorbidità riducendo la complessità delle strutture e i tempi di training e optando per una combinazione delle reti in grado di avvicinare i parametri di sensibilità e specificità.
Il presente progetto costituisce un riferimento per uno studio prospettico effettuato sui dati della 'Città della Salute e della Scienza di Torino' che mira al medesimo obiettivo di identificazione della condizione ottimale che determina un tentativo di estubazione eseguito con successo.
I risultati presentati consentono quindi di avere una prima valutazione delle strade percorribili e di quelle che possono essere le criticità; da questo punto di vista l'utilizzo di un classificatore come il LigthGBM potrebbe risultare difficoltoso per l'elevata dimensionalità di dati richiesta che consentirebbe di superare il fenomeno dell'overfitting; pertanto, le reti neurali potrebbero essere una strada più facilmente percorribile.
Nelle implementazioni future, risultati positivi si potrebbero riscontrare nell'implementazioni delle strutture più complesse delle Deep Neural Network tramite piattaforme diverse da Matlab, che non ne ha garantito un allenamento ottimale a causa dei limiti computazionali.
Il presente progetto di Tesi rafforza maggiormente l'idea che modelli di machine learning possano essere adatti alla predizione della capacità dei pazienti in condizioni critiche di sopportare la procedura di estubazione.

APPENDICE A

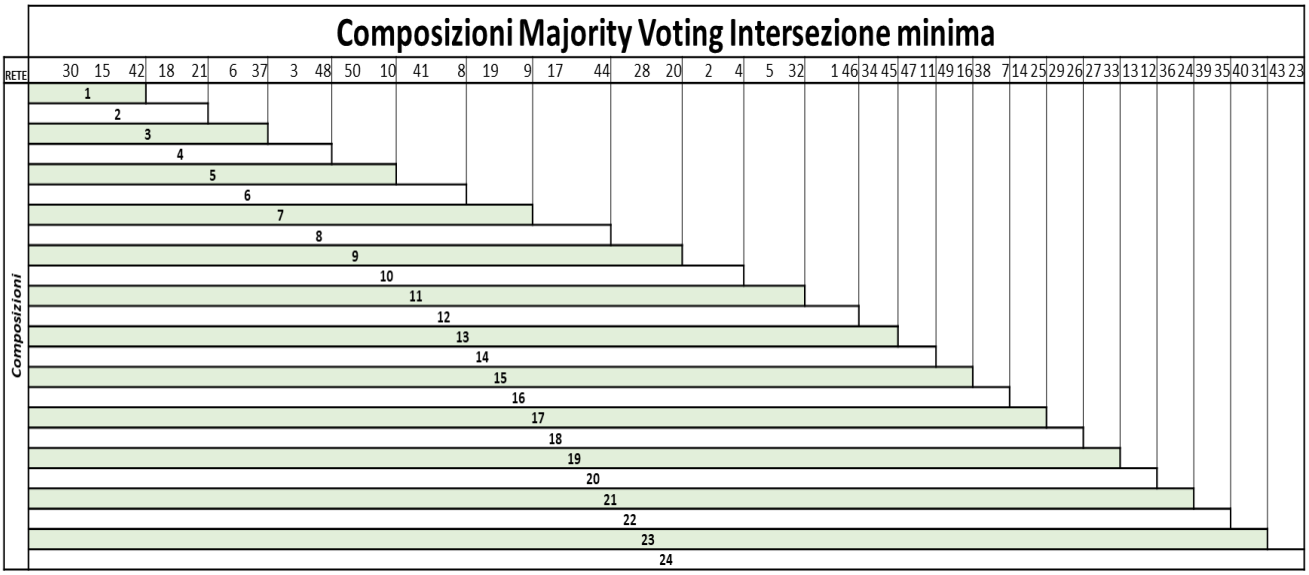
Composizioni dei Majority voting

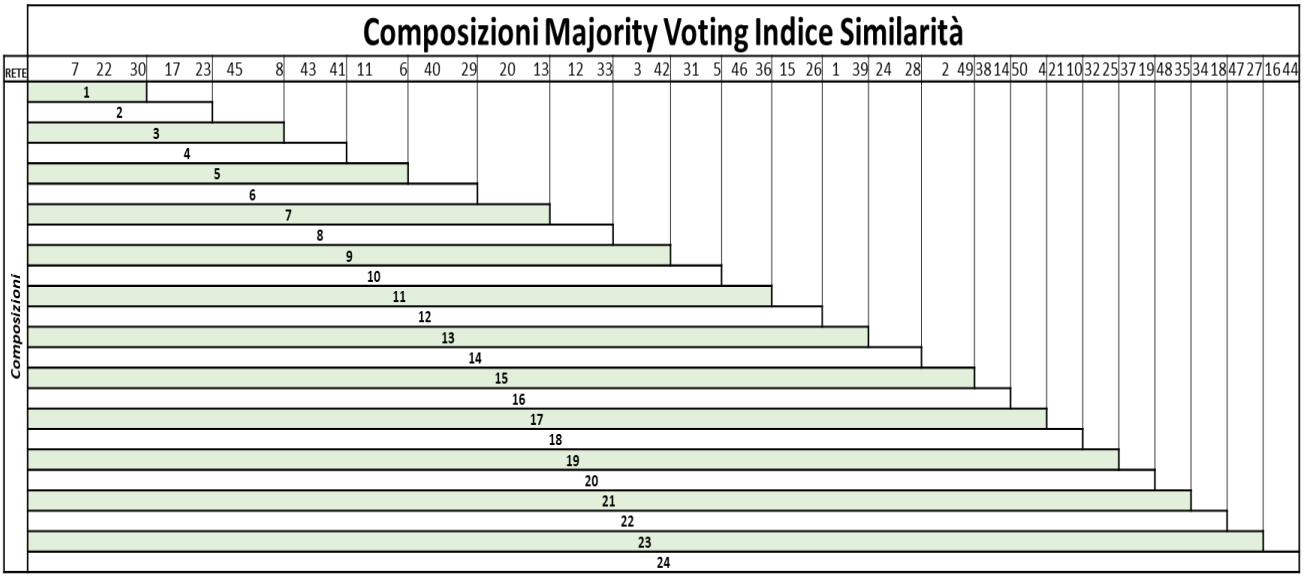
Majority Voting sulle reti ottenute dal GA con comorbidità





Majority Voting sulle reti ottenute dal GA con comorbidità sostituite





APPENDICE B

Variabili presenti nel Dataset

Variabili	*CATEGORIA*	*TIPOLOGIA*	*MOMENTO DELLA MISURAZIONE*	*MISSING DEGREE %*
'Age'	**Dati demografici**	DISCRETA	giorno ammissione	0,0
'Gender'		BINARIA	-	0,0
'Weight_loss'		COONTINUA	differenza giorno intubazione-estubazione	67,9
'Glasgow_Coma_Scale'	**Punteggi di gravità**	DISCRETA	primo giorno ricovero	2,8
'SOFA_score'		DISCRETA	primo giorno ricovero	0,0
'Sedation_days'	**Somministrazione terapia**	CONTINUA	dal primo giorno all'estubazione	0,0
'Vasopressor'		BINARIA	periodo ventilazione meccanica	0,0
'Chronic_cardiac_insufficiency'	**Comorbidità**	BINARIA	ammissione in ospedale	0,0
'Arrhytmia'		BINARIA	ammissione in ospedale	0,0
'Valvular_disease'		BINARIA	ammissione in ospedale	0,0
'Pulmonary_heart_disease'		BINARIA	ammissione in ospedale	0,0
'Peripheral_vascular'		BINARIA	ammissione in ospedale	0,0
'Hypertension'		BINARIA	ammissione in ospedale	0,0
'Stroke'		BINARIA	ammissione in ospedale	0,0
'Other_neurological'		BINARIA	ammissione in ospedale	0,0
'Chronic_Obstructive_Pulmonary_Disease'		BINARIA	ammissione in ospedale	0,0
'Diabetes_uncomplicated'		BINARIA	ammissione in ospedale	0,0
'Diabetes_complicated'		BINARIA	ammissione in ospedale	0,0
'Hypothyroidism'		BINARIA	ammissione in ospedale	0,0
'Chronic_kidney_disease'		BINARIA	ammissione in ospedale	0,0
'Liver_disease'		BINARIA	ammissione in ospedale	0,0
'Peptic_ulcer'		BINARIA	ammissione in ospedale	0,0
'AIDS'		BINARIA	ammissione in ospedale	0,0
'Lymphoma'		BINARIA	ammissione in ospedale	0,0
'Metastatic_tumors'		BINARIA	ammissione in ospedale	0,0
'Solid_tumor'		BINARIA	ammissione in ospedale	0,0
'Rheumatoid_arthritis'		BINARIA	ammissione in ospedale	0,0
'Coagulopathy'		BINARIA	ammissione in ospedale	0,0
'Obesity'		BINARIA	ammissione in ospedale	0,0
'Blood_loss_anemia'		BINARIA	ammissione in ospedale	0,0
'Deficiency_anemia'		BINARIA	ammissione in ospedale	0,0
'Alcohol_abuse'		BINARIA	ammissione in ospedale	0,0
'Drug_abuse_psychoses'		BINARIA	ammissione in ospedale	0,0
'Depression'		BINARIA	ammissione in ospedale	0,0
'Electrolyte_and_fluid_disorder'		BINARIA	ammissione in ospedale	0,0
'Psychoses'		BINARIA	ammissione in ospedale	0,0
'MV_duration_hours'	**Parametri ventilatori**	CONTINUA	ultimo valore registrato prima dell'estubazione	0,0
'Tidal_volume'		CONTINUA	ultimo valore registrato prima dell'estubazione	0,4
'Minute_ventilation'		CONTINUA	ultimo valore registrato prima dell'estubazione	0,6

'Mean_Airway_Pressure'		CONTINUA	ultimo valore registrato prima dell'estubazione	0,9
'Peak_inspiratory_pressure'		CONTINUA	ultimo valore registrato prima dell'estubazione	0,8
'mean_Heart_rate'		CONTINUA	periodo ventilazione meccanica	0,0
'max_Heart_rate'		CONTINUA	periodo ventilazione meccanica	0,0
'min_Heart_rate'		CONTINUA	periodo ventilazione meccanica	0,0
'mean_Respiratory_rate'		CONTINUA	periodo ventilazione meccanica	0,0
'max_Respiratory_rate'		CONTINUA	periodo ventilazione meccanica	0,0
'min_Respiratory_rate'		CONTINUA	periodo ventilazione meccanica	0,0
'mean_Temperature'		CONTINUA	periodo ventilazione meccanica	0,1
'max_Temperature'		CONTINUA	periodo ventilazione meccanica	0,1
'min_Temperature'		CONTINUA	periodo ventilazione meccanica	0,1
'mean_systolic_blood_pressure'		CONTINUA	periodo ventilazione meccanica	0,0
'max_systolic_blood_pressure'	**parametri vitali**	CONTINUA	periodo ventilazione meccanica	0,0
'min_systolic_blood_pressure'		CONTINUA	periodo ventilazione meccanica	0,0
'mean_diastolic_blood_pressure'		CONTINUA	periodo ventilazione meccanica	0,0
'max_diastolic_blood_pressure'		CONTINUA	periodo ventilazione meccanica	0,0
'min_diastolic_blood_pressure'		CONTINUA	periodo ventilazione meccanica	0,0
'mean_mean_arterial_pressure'		CONTINUA	periodo ventilazione meccanica	0,0
'max_mean_arterial_pressure'		CONTINUA	periodo ventilazione meccanica	0,0
'min_mean_arterial_pressure'		CONTINUA	periodo ventilazione meccanica	0,0
'mean_SpO2'		CONTINUA	periodo ventilazione meccanica	0,0
'max_SpO2'		CONTINUA	periodo ventilazione meccanica	0,0
'min_SpO2'		CONTINUA	periodo ventilazione meccanica	0,0
'White_blood_cells'		CONTINUA	ultimo valore registrato prima dell'estubazione	0,3
'Hemoglobin'		CONTINUA	ultimo valore registrato prima dell'estubazione	0,1
'Platelet'	**analisi ematiche ed emogasanalisi**	CONTINUA	ultimo valore registrato prima dell'estubazione	0,2
'Sodium'		CONTINUA	ultimo valore registrato prima dell'estubazione	0,1
'Potassium'		CONTINUA	ultimo valore registrato prima dell'estubazione	0,1

'Calcium'		CONTINUA	ultimo valore registrato prima dell'estubazione	8,9
'Chloride'		CONTINUA	ultimo valore registrato prima dell'estubazione	0,2
'Lactate'		CONTINUA	ultimo valore registrato prima dell'estubazione	20,3
'Glucose'		CONTINUA	ultimo valore registrato prima dell'estubazione	0,1
'Creatinine'		CONTINUA	ultimo valore registrato prima dell'estubazione	0,1
'Blood_Urea_Nitrogen'		CONTINUA	ultimo valore registrato prima dell'estubazione	0,1
'Bilirubin'		CONTINUA	ultimo valore registrato prima dell'estubazione	48,9
'Albumin'		CONTINUA	ultimo valore registrato prima dell'estubazione	52,2
'Prothrombin_Time'		CONTINUA	ultimo valore registrato prima dell'estubazione	3,6
'activated_Partial_Thromboplastin_Time'		CONTINUA	ultimo valore registrato prima dell'estubazione	3,7
'International_Normalized_Ratio'		CONTINUA	ultimo valore registrato prima dell'estubazione	3,6
'GPT'		CONTINUA	ultimo valore registrato prima dell'estubazione	48,8
'GOT'		CONTINUA	ultimo valore registrato prima dell'estubazione	48,9
'Central_vein_oxygenated'		CONTINUA	ultimo valore registrato prima dell'estubazione	28,1
'arterial_pH'		CONTINUA	ultimo valore registrato prima dell'estubazione	2,4
'PaCO2'		CONTINUA	ultimo valore registrato prima dell'estubazione	2,6
'PaO2'		CONTINUA	ultimo valore registrato prima dell'estubazione	2,6
'Base_excess'		CONTINUA	ultimo valore registrato prima dell'estubazione	2,6

BIBLIOGRAFIA

[1] M. B. JI, «Definitions in mechanical ventilation,» Recibido en marzo de 2003..

[2] NHLBI, «National Heart,Lung and Blood Institute,» [Online]. Available: https://www.nhlbi.nih.gov/health-topics/respiratory-failure.

[3] NHLBI, «National Heart, Lung and Blood Institute,» [Online]. Available: https://www.nhlbi.nih.gov/health-topics/ventilatorventilator-support].

[4] M. C. M. M. Martin Tobin, «Mechanical Ventilation,» *American Thoracic Society.*

[5] J. B. A. C. M. H. B. M. C. M. R. P. S. M. S. A. V.-B. T. W. J-M. Boles*, «Weaning from mechanical ventilation,» *EUROPEAN RESPIRATORY JOURNAL,* vol. VOLUME 29, n. NUMBER 5, December 22 2006.

[6] T. P. 3. 4. 5. F. S. 6. L. P. 7. 8. E. Z. 9. 1. M. J. 1. F. G. 1. I. R. 1. N. T. 1. 1. 1. 1. S. G. 1. G. B. 1. P.-G. G. 1. J.-P. Gaëtan Béduneau 1 2, «Epidemiology of Weaning Outcome according to a New Definition,» *American Journal of Respiratory and Critical Care Medicine,* vol. Volume 195, n. Number 6, March 15 2017.

[7] J.-C. M. R. a. L. B. Arnaud W. Thille1, «The Decision to Extubate in the Intensive Care Unit,» *American Thoracic Society,* April 15, 2013).

[8] L. Rose, «Strategies for weaning from mechanical,» *Elsevier Ltd.,* 28 June 2015.

[9] C. F. H. M. R. F. a. P. S. L. RRT, «Ventilator Discontinuation Protocols,» *RESPIRATORY CARE,* vol. VOL 57, n. 10, OCTOBER 2012.

[10] P. a. A. S. R. P. P. C. T. M. P. Luís Guilherme Alegretti Borges, «Mechanical ventilation weaning protocol improves medical adherence,» *Journal of Critical Care,* 2017.

[11] N. R. M. M. FAARC, «Evidence-Based Ventilator Weaning and Discontinuation,» *RESPIRATORY CARE,* vol. VOL 49 , n. NO 7, JULY 2004.

[12] A. E. W. e. a. Johnson, «Data Descriptor: MIMIC-III, a freely accessible critical care database,» *www.nature.com/scientificdata,* 24 May 2016.

[13] A. Long, «Toward Data Science,» [Online]. Available: https://towardsdatascience.com/getting-access-to-mimic-iii-hospital-database-for-data-science-projects-791813feb735.

[14] «Microsoft,» [Online]. Available: https://www.microsoft.com/en-us/research/project/lightgbm/.

[15] Q. M. T. F. T. W. C. W. M. Q. Y. T.-Y. L. Guolin Ke1, «LightGBM: A Highly Efficient Gradient Boosting,» in *31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA., 2017.

[16] A. Ye, «Towards Data Science,» 27 Sept 2020. [Online]. Available: https://towardsdatascience.com/xgboost-lightgbm-and-other-kaggle-competition-favorites-6212e8b0e835.

[17] S. N. Kasturi, «Towards Data Science,» 11 Jul 2019. [Online]. Available: https://towardsdatascience.com/lightgbm-vs-xgboost-which-algorithm-win-the-race-1ff7dd4917d.

[18] A. Y. S. ,. A. O. Haithm Haithm ALSHARI*1, «Comparison of Gradient Boosting Decision Tree Algorithms for CPU Performance,» *Journal of Institue Of Science and Technology,* vol. Volume 37, n. Issue 21, 28.04.2021.

[19] G. Rao, «What is an ROC curve?,» *The Journal of Family Practice,* vol. VOL 52, n. 9, SEPTEMBER 2003.

[20] 1. J. C. D. T. Zhe Hui Hoo, «What is an ROC curve?,» March 2017.

[21] B. B. T. L. a. D. o. a. Neuron, «National Institute of Neurological Disorder and Stroke,» 2019 Mar 16. [Online]. Available: https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Life-and-Death-Neuron. [Consultato il giorno Jul 07 20121].

[22] *. M. H. I.A. Basheera, «Artificial neural networks: fundamentals, computing, design, and application,» *Journal of Microbiological Methods,* 2000.

[23] S. K. D. K. K. S. Manishaa, «ECG Beat Classifiers: A Journey from ANN To DNN,» in *Science, International Conference on Computational Intelligence and Data*, 2019.

[24] T. V. Mathew, *Genetic Algorithm,* Indian Institute of Technology Bombay.

[25] M. H. N. U. &. D. G. Manoj Kumar1, «GENETIC ALGORITHM: REVIEWAND APPLICATION,» *International Journal of Information Technology and Knowledge Management,* vol. 2, n. 2, pp. 451-454, 2010.

[26] G. K. Q. M. F. Wang1, «LightGBM: A Highly Efficient Gradient Boosting,» in *31st Conference on Neural Information Processing Systems (NIPS 2017), L*, ong Beach, CA, USA, 2017.