

POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Biomedica - Biomeccanica

Tesi di Laurea Magistrale

One shot, one SoS: A comparison of speed of sound estimators



**Politecnico
di Torino**

Relatore
Prof.ssa Kristen Mariko Meiburger

Correlatore:
Ing. Silvia Seoni

Laureando
Luca Cervizzi

Dicembre 2025

A me stesso,
Alla mia famiglia,
A chi ha creduto in me.

Sommario

La velocità del suono (SoS) è la velocità di propagazione delle onde acustiche nel mezzo. Nella maggior parte dei sistemi ecografici, la SoS è assunta pari a 1540 m/s; tuttavia, tale approssimazione non considera le differenze tra i tessuti biologici e può introdurre errori durante la formazione dell'immagine. Essendo una proprietà intrinseca dei tessuti umani, la SoS è recentemente emersa come un potenziale biomarcatore nella diagnostica per immagini.

L'obiettivo di questo progetto di tesi è l'implementazione di un algoritmo di regressione capace di stimare la SoS globale e locale del tessuto di interesse da impiegare nel processo di beamforming, tramite analisi radiomica di tessitura. In questo studio, le feature di tessitura vengono calcolate sui segnali pre-beamforming. La radiomica, infatti, consente di estrarre descrittori quantitativi dalle immagini mediche, rivelando pattern strutturali non visibili all'occhio umano e potenzialmente correlati alle proprietà fisiche dei tessuti. Tale approccio è applicato allo scopo di superare le limitazioni dell'ipotesi convenzionale di SoS costante pari a 1540 m/s.

Il lavoro è articolato in tre fasi. Nella prima fase è stata condotta un'analisi esplorativa su un fantoccio mammario 3B Scientific (SoS reale 1540 m/s), valutando tramite coefficiente di Pearson la correlazione lineare e logaritmica tra tutti i parametri di tessitura e il fattore di calibrazione (CF) utilizzato nella formazione dell'immagine. Nella seconda fase l'analisi è stata estesa a un fantoccio a base di PDMS/Agarosio (SoS 1250 m/s), studiando l'andamento delle feature di secondo ordine derivate dalla matrice GLCM in funzione della SoS. Nella terza fase è stato generato un dataset di 140 immagini ecografiche simulate con k-Wave, rappresentanti strutture stratificate a densità costante ma caratterizzate da diverse SoS locali e globali. Su tali immagini è stato addestrato un modello di regressione basato su Extreme Gradient Boosting (XGBoost) per la stima della SoS sia globale sia locale, e sono stati inoltre implementati un regressore secondo il metodo SNAC proposto da *De la Torre et al.* (per la stima locale e globale) e un operatore neurale di Fourier (OFN) per la sola stima locale; le prestazioni sono state confrontate considerando due test set: uno costituito dalle sole immagini simulate e uno dalle immagini acquisite sui due fantocchi reali.

Dall'analisi della seconda fase, cinque parametri di tessitura – autocorrelazione, IDN, media congiunta, IDMN e somma media – hanno mostrato una significativa dipendenza dalla SoS. Nella terza fase, sul dataset simulato XGBoost ha evidenziato prestazioni complessivamente superiori a SNAC nella stima sia globale sia locale (ad esempio, per la stima globale $MSD \approx -21$ m/s contro -79 m/s e R^2 pari a 0.80 contro

0.42). Tuttavia, per le immagini a singolo strato, quindi caratterizzate da un'unica SoS, SNAC ha fornito errori più contenuti e una migliore descrizione degli scenari acusticamente omogenei. Per la stima locale, l'OFN ha mostrato accuratezza intermedia tra XGBoost e SNAC nelle diverse configurazioni. Nel dominio sperimentale, il metodo SNAC ha ulteriormente mostrato prestazioni più accurate: le SoS stimate, sia globali sia locali, risultano più vicine ai valori reali rispetto a quelle ottenute con XGBoost e OFN, evidenziando una maggiore robustezza di SNAC nella caratterizzazione di fantocci con distribuzione di SoS non ideale.

In conclusione, questo studio dimostra la potenziale applicazione della radiomica di tessitura per la stima della SoS nei tessuti e pone le basi per metodologie di imaging ecografico adattativo mirate al miglioramento della qualità d'immagine e dell'accuratezza diagnostica.

Indice

Elenco delle tabelle	8
Elenco delle figure	10
Abbreviazioni	13
1 Ultrasonografia	14
1.1 Principi di funzionamento	14
1.1.1 Propagazione e attenuazione delle onde ultrasonore	15
1.1.2 Piezoelettricità e sonde	16
1.1.3 Meccanismi di formazione dell'immagine: riflessione e trasmissione	18
1.1.4 Simulazione dell'equazione d'onda	20
1.2 Beamforming	21
1.2.1 DAS beamforming	21
1.2.2 Metodi di beamforming alternativi	24
1.2.3 Limitazioni del modello a velocità costante (1540 m/s)	26
1.3 Post processing	27
1.3.1 Time Gain Compensation e Demodulazione RF	27
1.3.2 Back-end	28
2 Machine Learning	30
2.1 Introduzione al Machine Learning	30
2.1.1 Metodi di apprendimento	30
2.1.2 Compiti di applicazione	31
2.1.3 Valutazione delle prestazioni	32
2.2 Modelli di apprendimento supervisionato	32
2.2.1 Algoritmi di apprendimento parametrici	33
2.2.2 Algoritmi di apprendimento non parametrici	35
2.3 Bagging: Random forest	39
2.4 Boosting	40
2.4.1 Boosting Adattivo: ADABOOST	40
2.4.2 Gradient boosting	41
2.4.3 XGBoost: Extreme gradient boost	42

3	Materiali e Metodi	45
3.1	Dataset	45
3.1.1	Immagini sperimentali	45
3.1.2	Simulazione delle immagini	47
3.2	Strumentazione	50
3.2.1	Ecografo	50
3.2.2	Simulatore k-Wave	51
3.3	Ricostruzione dell'immagine	53
3.3.1	Beamforming	53
3.3.2	Post processing	54
3.4	Metodi di analisi	55
3.4.1	Analisi di regressione lineare e logaritmica	55
3.4.2	Analisi radiomica di tessitura	57
3.4.3	Feature di tessitura del primo ordine	59
3.4.4	Feature di tessitura di ordine superiore	59
3.5	Stima della velocità del suono globale mediante regressore XGBoost . .	61
3.5.1	Struttura del modello	62
3.5.2	Metriche di valutazione	65
3.6	Metodo SNAC per la stima della velocità del suono globale	65
3.6.1	Principi teorici	65
3.6.2	Implementazione e parametri	66
3.7	Confronto tra gli stimatori della velocità del suono globale	67
3.8	Stima della velocità del suono locale mediante regressore XGBoost . . .	68
3.8.1	Struttura del modello	68
3.9	Adattamento del metodo SNAC alla stima locale	72
3.10	Fourier Neural Operators per la stima locale della SoS	72
3.11	Confronto tra gli stimatori della velocità del suono locale	75
3.12	Test di normalità e analisi statistica	75
4	Risultati e Discussioni	77
4.1	Coefficienti di calibrazione e tessitura dell'immagine	77
4.1.1	Variazione del coefficiente di calibrazione lungo gli assi x e z . .	78
4.1.2	Variazione del coefficiente di calibrazione lungo l'asse z	80
4.1.3	Discussione dei risultati	82
4.2	Relazione tra la velocità del suono e le feature GLCM	82
4.2.1	Analisi su regioni di interesse	82
4.2.2	Estensione dell'analisi a livello globale di immagine	85
4.2.3	Mappe parametriche delle feature: confronto pre e post beam- forming	86
4.2.4	Discussione dei risultati	88
4.3	Metodi di stima della velocità del suono	88
4.3.1	Modello XGBoost per la stima globale della velocità del suono .	89
4.3.2	Confronto delle prestazioni nella stima globale della SoS	93
4.3.3	Modello XGBoost per la stima locale della velocità del suono . .	96

4.3.4	Confronto delle prestazioni nella stima locale della SoS	101
	Conclusioni	105
	Limitazioni	107
	Approcci futuri	107
	Bibliografia	108

Elenco delle tabelle

1.1	Valori di α per alcuni tessuti biologici alla frequenza di 1 MHz.	16
1.2	Valori di impedenza acustica (Z) di diversi materiali. La tabella elenca la densità (ρ), la velocità del suono (c) e la conseguente impedenza acustica.	19
3.1	Parametri di ricostruzione e caratteristiche delle immagini acquisite in-vitro.	47
3.2	Parametri di ricostruzione e caratteristiche delle immagini simulate.	48
3.3	Caratteristiche tecniche della sonda ecografica.	50
3.4	Parametri di tessitura del primo ordine.	60
3.5	Sottoinsieme di parametri di tessitura derivati dalla matrice GLCM selezionati nell'analisi esplorativa.	62
3.6	Iperparametri e intervalli di valori considerati durante la fase di tuning del modello XGBoost per la stima della SoS globale.	64
3.7	Iperparametri e intervalli di valori considerati durante la fase di tuning del modello XGBoost per la stima patch-wise della velocità del suono.	71
4.1	Comportamento delle feature GLCM in corrispondenza della SoS ottimale.	86
4.2	Iperparametri ottimali del modello XGBoost ottenuti tramite nested cross-validation.	90
4.3	Importanza delle 19 feature radiomiche di tessitura selezionate dopo scrematura per varianza e correlazione.	91
4.4	Confronto delle prestazioni del modello tra i diversi set: training, validation e test simulato.	91
4.5	Prestazioni del modello sul test set sperimentale. Confronto tra la velocità del suono reale e quella stimata dal modello XGBoost per i due fantocci sperimentali.	92
4.6	Confronto delle prestazioni dei metodi XGBoost e SNAC sul test set di immagini simulate. *Differenza statisticamente significativa tra XGBoost e SNAC secondo test statistico.	94
4.7	Confronto tra la velocità del suono reale e quella stimata dai metodi XGBoost e SNAC sul test set sperimentale.	94
4.8	Iperparametri ottimali del modello XGBoost per la stima locale della SoS ottenuti tramite 5-fold cross validation.	97
4.9	Importanza delle 20 feature radiomiche di tessitura più rilevanti per il modello XGBoost nella stima locale della velocità del suono.	99

4.10	Confronto delle prestazioni del modello XGBoost nella stima locale della SoS tra i diversi set: training-validation e test simulato.	99
4.11	Confronto tra la velocità del suono reale e quella stimata dal modello XGBoost nella stima locale della SoS sul test set sperimentale. Per ciascun fantoccio, la SoS predetta locale è riportata come media \pm deviazione standard delle stime ottenute sulle 8 patch che compongono l'immagine.	100
4.12	Confronto delle prestazioni dei metodi XGBoost, SNAC e OFN sul test set di immagini simulate. <i>*Differenza statisticamente significativa tra XGBoost e SNAC secondo test statistico.</i> <i>†Differenza statisticamente significativa tra XGBoost e OFN secondo test statistico.</i>	102
4.13	Confronto tra la velocità del suono reale e quella stimata dai metodi XGBoost, SNAC e OFN sul test set sperimentale.	102

Elenco delle figure

1.1	Rappresentazione dei principali tipi di scansione ecografica. Da sinistra a destra: A-mode, B-mode, M-mode.	15
1.2	Tipologie di sonde ecografiche. Da sinistra verso destra: lineare, phased array, convex, microconvex.	17
1.3	Riflessione e rifrazione di un'onda acustica. Il diagramma illustra un'onda acustica che incide sull'interfaccia tra due mezzi con indici di rifrazione differenti.	19
1.4	Schema a blocchi che illustra il principio del beamforming DAS. Un segnale acustico $S(t)$ viene acquisito da un array di trasduttori (raffigurato in grigio). Il segnale ricevuto da ciascun elemento dell'array viene ritardato in modo differenziato ($\tau_1, \tau_2 \dots \tau_7$, rappresentati dal modulo verde) per compensare le diverse distanze e angolazioni rispetto al punto di interesse. Successivamente, tutti i segnali temporizzati vengono sommati per ottenere un segnale finale coerente, migliorando la qualità e la risoluzione dell'immagine ecografica.	22
1.5	Principio del calcolo del ritardo temporale.	23
1.6	Confronto tra le funzioni di apodizzazione della finestra di Tukey per diversi valore di α . Per $\alpha = 1$ si ottiene la finestra di Hanning, mentre al diminuire di α la funzione assume un andamento simile a quella rettangolare.	24
1.7	Schema a blocchi che illustra il principio del beamforming DMAS. . . .	25
1.8	Rappresentazione TGC. La curva mostra il guadagno relativo applicato a ogni specifica profondità. L'aumento del guadagno (amplificazione degli echi di ritorno) all'aumentare della profondità viene rappresentato con la visualizzazione della curva sempre più a destra procedendo verso il basso.	28
1.9	Pipeline di elaborazione per modalità Brightness.	29
2.1	Schema illustrativo di un albero decisionale con profondità pari a 2. Il nodo in azzurro rappresenta la radice da cui parte il processo, i nodi in verde indicano le decisioni intermedie, mentre i nodi in rosso rappresentano le foglie contenenti i valori di output.	36
2.2	Rappresentazione metodo di insieme. L'insieme dei dati contiene due attributi differenti e il sistema utilizza tre sistemi di classificazione per modellare l'informazione.	38

2.3	Rappresentazione del metodo di bagging.	39
2.4	Evoluzione degli alberi decisionali.	42
3.1	Rappresentazione dei fantocci utilizzati. A sinistra è mostrato il fantoccio 3B Scientific Breast, mentre a destra è visibile il fantoccio in PDMS con emulsione.	46
3.2	Rappresentazione Immagini Ecografiche Fantocci. A sinistra fantoccio 3B Scientific Breast Phantom, a destra il fantoccio in PDMS/Agarosio.	47
3.3	Rappresentazione di immagini ecografiche simulate con <i>k-Wave</i> e delle corrispondenti mappe della velocità del suono. In alto è mostrata un'immagine simulata con due strati orizzontali aventi SoS pari a 1340 e 1440 m/s, mentre in basso è riportata un'immagine simulata con due strati verticali caratterizzati da SoS pari a 1200 e 1380 m/s.	49
3.4	Adattatore <i>UTA 260-D</i>	50
3.5	A destra la sonda Verasonics L11-5v. Al centro la risposta in frequenza (spettro) della sonda. A sinistra la risposta in tensione nel dominio del tempo (Output Voltage vs. Time) del trasduttore, che caratterizza l'impulso acustico generato.	51
3.6	Esempio di costruzione della matrice di co-occorrenza (GLCM). a) Valori numerici corrispondenti ai livelli di intensità dei pixel dell'immagine. b) GLCM corrispondente. c) Rappresentazione in scala di grigi dell'immagine mostrata in (a). d) Visualizzazione in scala di grigi della matrice GLCM. I cerchi rossi in (a) indicano le adiacenze (1,6), corrispondenti al valore evidenziato in rosso nel pannello (b).	61
3.7	Framework del metodo SNAC. I dati vengono acquisiti tramite una singola trasmissione a onda piana (passo 1). Successivamente, i campioni di ciascun canale vengono opportunamente ritardati e corretti per la rifrazione in corrispondenza di ciascun candidato SoS (passo 2). Al passo 3, viene calcolata la funzione di perdita per ogni valore di SoS candidato. Infine, la stima della SoS è ottenuta selezionando il minimo della funzione di perdita opportunamente interpolata (passo 4).	67
4.1	Coefficienti di determinazione R^2 delle famiglie di feature di tessitura. Sono riportati i coefficienti di determinazione R^2 lineari e logaritmici calcolati tra i coefficienti di calibrazione lungo gli assi x e z e i parametri di tessitura (per le famiglie di feature di Primo ordine, GLCM, GLDM, GLRLM, GLSZM e NGDTM).	79
4.2	Coefficienti di determinazione R^2 delle famiglie di feature di tessitura. Sono riportati i coefficienti di determinazione R^2 lineari e logaritmici calcolati tra i coefficienti di calibrazione lungo l'asse z e i parametri di tessitura (per le famiglie di feature di Primo ordine, GLCM, GLDM, GLRLM, GLSZM e NGDTM).	81

4.3	Posizionamento delle regioni di interesse nelle immagini ecografiche acquisite sui due fantocci. A sinistra è riportata l'immagine relativa al fantoccio in PDMS/Agarosio, con una ROI superficiale evidenziata in blu e una ROI profonda evidenziata in rosso. A destra è mostrata l'immagine del fantoccio di seno mammario, in cui la ROI in rosso è collocata all'interno della lesione e la ROI in blu è posizionata nel tessuto superficiale di riferimento.	83
4.4	Relazione tra SoS e feature GLCM nel fantoccio in PDMS/Agarosio. Le curve mostrano l'andamento delle feature di tessitura derivate dalla matrice GLCM, calcolate sulle due ROI (superficiale e profonda) del fantoccio al variare della velocità del suono assunta nel beamforming. Nell'immagine di sinistra le feature sono calcolate sui dati immediatamente dopo il beamforming, mentre nell'immagine di destra sono calcolate sull'immagine ecografica finale. La linea verticale nera rappresenta la velocità di propagazione reale del suono nel mezzo (SoS = 1250 m/s).	84
4.5	Relazione tra SoS e feature GLCM nel fantoccio mammario. Le curve mostrano l'andamento delle feature di tessitura derivate dalla matrice GLCM, calcolate sulle due ROI (all'interno del nodulo benigno e nel tessuto circostante) del <i>3B Scientific Breast Phantom</i> al variare della velocità del suono assunta nel beamforming. Nell'immagine di sinistra le feature sono calcolate immediatamente dopo il beamforming, mentre nell'immagine di destra sono calcolate sull'immagine ecografica finale. La linea verticale nera rappresenta la velocità di propagazione reale del suono nel mezzo (SoS = 1540 m/s).	85
4.6	Andamento globale dell'autocorrelazione in funzione della velocità del suono. Le due curve mostrano l'autocorrelazione normalizzata calcolata sull'intera immagine ecografica, in funzione della velocità del suono assunta nel beamforming, per il fantoccio in PDMS/Agarosio e per il fantoccio di seno mammario. Le linee tratteggiate indicano le rispettive velocità di propagazione reali del suono nei due mezzi.	86
4.7	Mappe spaziali della feature IDN pre e post-beamforming. Confronto tra le mappe IDN ottenute prima e dopo il beamforming per i due fantocci, che mostrano una distribuzione spaziale complessivamente simile in entrambe le condizioni.	87
4.8	Confronto dei valori delle feature nei dati simulati e sperimentali. La figura mostra il confronto tra i valori delle feature <i>Root Mean Squared</i> (sinistra) e <i>Skewness</i> (destra) calcolati sulle immagini simulate a singolo strato (tutte le barre a sinistra) e sulle due immagini acquisite sui fantocci in vitro (ultime due barre).	93
4.9	Errori di stima della velocità del suono sui fantocci sperimentali. L'errore è definito come differenza tra la SoS predetta e quella reale per ciascun metodo (XGBoost e SNAC) e per ciascun fantoccio.	95

Abbreviazioni

BF Beamforming

DSC Digital Scan Conversion

GLCM Gray Level Co-occurrence Matrix

GLDM Gray Level Dependence Matrix

GLRLM Gray Level Run Length Matrix

GLSZM Gray Level Size Zone Matrix

ML Machine Learning

NGTDM Neighborhood Gray Tone Difference Matrix

RF Radiofrequenza

ROI Regione di interesse

SoS Velocità del suono

US Ultrasound

Capitolo 1

Ultrasonografia

1.1 Principi di funzionamento

Gli ultrasuoni (US) risalgono a oltre 50 anni fa e sono una delle modalità di imaging medico più comuni [1]. Il loro basso costo, l'elevata sicurezza, la natura non ionizzante, la facilità d'uso e l'acquisizione in tempo reale li rendono il *gold standard* per molti esami diagnostici, ad esempio in ostetricia e cardiologia [2].

A differenza della maggior parte delle altre tecniche di imaging, come la radiografia e la tomografia computerizzata (TC), gli ultrasuoni non utilizzano radiazioni ionizzanti, le quali hanno effetti biologici significativi sul corpo umano. Pertanto, possono essere utilizzati per lunghi periodi di tempo e applicati a pazienti sensibili come donne in gravidanza e bambini [3].

I primi sistemi a ultrasuoni utilizzavano la modalità A (*Amplitude Mode*), che prevede l'impiego di una singola linea di scansione. In questa modalità, viene inviata un'onda ultrasonora e, a ogni discontinuità di impedenza acustica tra i tessuti, il trasduttore riceve un eco. Il segnale viene visualizzato su un display in funzione della profondità, proporzionale al tempo di volo, e dell'ampiezza, proporzionale all'intensità del segnale riflesso. Questa modalità A era principalmente di interesse accademico fino a quando fu sviluppata la modalità M (*Motion mode*), o modalità di movimento, in cui l'ampiezza del segnale era visualizzata in funzione del tempo. Da diversi decenni, la modalità B (*Brightness mode*) è la tecnica ecografica più utilizzata, in cui i fasci ultrasonori focalizzati vengono inviati in modo sequenziale nel corpo e le onde ricevute vengono utilizzate per formare una linea di scansione dell'immagine, coprendo progressivamente, linea per linea, la regione di interesse. In particolare, la sonda genera numerose linee di scansione adiacenti, che insieme formano l'immagine bidimensionale. In questa strategia, la luminosità dell'immagine visualizzata riflette l'ampiezza dell'eco (maggiore è l'ampiezza del segnale riflesso, maggiore è la luminosità), mentre il tempo di volo determina la profondità della discontinuità nei tessuti [4].

Per “formazione dell'immagine” si intende l'intero procedimento di ricostruzione e generazione dell'immagine. Questo processo comprende diverse fasi successive: dalla definizione della strategia di trasmissione, alla ricezione dei segnali, alla formazione del

fascio (beamforming) fino all'elaborazione e rappresentazione dell'immagine finale [5]. Questo processo si basa sui principi della riflessione del suono, descritti in precedenza: il trasduttore emette impulsi ultrasonori lungo linee di propagazione attraverso il tessuto e riceve gli echi riflessi generati dalle interfacce tra mezzi con diversa impedenza acustica. I segnali vengono quindi tracciati come ampiezza in funzione del tempo (successivamente convertito in profondità, assumendo una velocità costante del suono nel tessuto) [7].

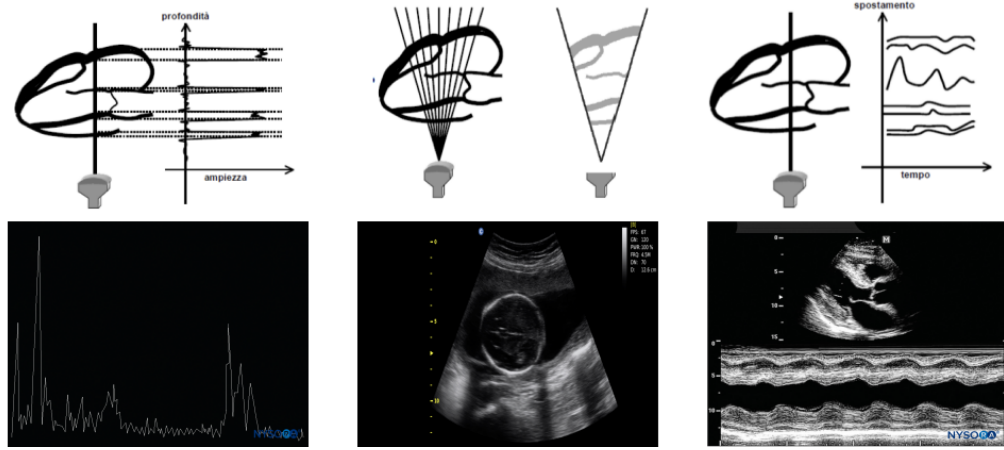


Figura 1.1: Rappresentazione dei principali tipi di scansione ecografica. Da sinistra a destra: A-mode, B-mode, M-mode.

1.1.1 Propagazione e attenuazione delle onde ultrasonore

Gli ultrasuoni sono onde meccaniche longitudinali, il cui spostamento delle particelle del mezzo in cui si propagano è parallelo alla direzione del moto dell'onda. Le grandezze che descrivono un ultrasuono, e le onde acustiche in generale, sono la frequenza, il periodo, la lunghezza d'onda, la velocità di propagazione, l'ampiezza e l'intensità. La frequenza f indica le oscillazioni complete (cicli) compiute in un secondo, mentre il periodo T è il tempo necessario per eseguire un ciclo completo. Le due grandezze sono legate dalla seguente relazione:

$$f = \frac{1}{T} \quad (1.1)$$

La lunghezza d'onda rappresenta la distanza percorsa dall'onda durante un ciclo completo. La velocità di propagazione è la velocità con cui l'onda sonora attraversa un particolare mezzo e aumenta all'aumentare della densità e della rigidità dello stesso [6]. Le onde sonore sono generate da trasduttori stimolati da impulsi elettrici a vibrare alle frequenze desiderate f_0 , solitamente comprese tra 2 e 18 MHz [7]. Nel corpo umano, la velocità del suono c a queste frequenze è circa 1540 m/s, e la lunghezza d'onda λ è data da:

$$\lambda = \frac{c}{f_0} \quad (1.2)$$

che produce valori compresi nell'intervallo tra 0,10 e 0,77 mm [1]. La lunghezza d'onda fornisce una misura della risoluzione spaziale ottenibile nell'immagine ecografica. Dalla Equazione 1.2, è chiaro che λ è inversamente proporzionale alla frequenza dell'onda. Pertanto, a frequenze elevate corrispondono lunghezze d'onda minori e, di conseguenza, una risoluzione spaziale più alta.

Durante la propagazione attraverso i tessuti, le onde ultrasonore subiscono una progressiva attenuazione causata dall'assorbimento da parte dei tessuti stessi. Ciò limita la profondità di penetrazione dell'onda e, di conseguenza, la profondità dell'esame medico. L'ampiezza dell'onda decresce in modo esponenziale:

$$A(\zeta) = A_0 e^{-\alpha(f)\zeta} \quad (1.3)$$

dove le varie quantità hanno il seguente significato:

1. $\alpha(f)$ è il coefficiente di attenuazione, che dipende sia dalla frequenza f sia dal tipo di tessuto (Tabella 1.1);
2. A_0 è l'ampiezza dell'onda iniziale;
3. $A(\zeta)$ è l'ampiezza dell'onda alla profondità ζ .

In particolare, α aumenta all'aumentare della frequenza, rendendo preferibili le basse frequenze. Al contrario, la risoluzione spaziale migliora all'aumentare della frequenza [8]. Pertanto, è necessario trovare un compromesso tra profondità di penetrazione e risoluzione spaziale: gli organi superficiali vengono esaminati a frequenze più elevate per favorire la risoluzione spaziale, al contrario, per gli organi più profondi vengono impiegate frequenze più basse per limitare l'attenuazione.

Materiali	α [cm ⁻¹]
Acqua	0,0006
Fegato	0,34
Muscolo	0,4
Plasma	0,014
Rene	0,44
Sangue intero	0,04
Tessuto adiposo	0,26

Tabella 1.1: Valori di α per alcuni tessuti biologici alla frequenza di 1 MHz.

1.1.2 Piezoelettricità e sonde

Le sonde ecografiche sono trasduttori, ovvero dispositivi utilizzati per convertire una forma di energia in ingresso in una diversa forma di energia in uscita. Il loro funzionamento è reso possibile dalla presenza, al loro interno, di materiali ceramici o di cristalli piezoelettrici. I sistemi recenti utilizzano microchip al silicio contenenti, ad esempio,

trasduttori ultrasonici micromeccanici capacitivi (CMUT) [7]. Il funzionamento delle sonde si basa sull'effetto piezoelettrico generato da questi materiali. La piezoelettricità è la proprietà di alcuni materiali di generare una differenza di potenziale quando sottoposti a deformazione meccanica (effetto piezoelettrico diretto) e, allo stesso tempo, di deformarsi elasticamente quando sottoposti a tensione elettrica (effetto piezoelettrico inverso). Questo fenomeno si verifica solo lungo una direzione specifica (anisotropia). Nei dispositivi ultrasonografici, l'effetto piezoelettrico viene spesso sfruttato in entrambe le direzioni: inverso per generare impulsi e diretto per misurare gli echi di ritorno. Le principali tipologie di sonde utilizzate sono:

- *Sonda lineare*: presenta cristalli disposti in modo lineare. Sfrutta alte frequenze, comprese tra 5 e 15 MHz, ed è impiegata per ecografie superficiali o pediatriche.
- *Sonda phased array*: utilizza frequenze tra 2 e 10 MHz e presenta un fascio che può essere focalizzato nella direzione desiderata, permettendo di evitare ostacoli come le ossa. Questa sonda è comunemente impiegata per ecografie transcraniche e transtoraciche.
- *Sonda convex*: utilizza frequenze leggermente più basse rispetto alle sonde precedenti (2-7 MHz); l'onda sonora subisce una minore attenuazione, consentendo la scansione di tessuti più profondi. Presenta un raggio di curvatura di circa 60 mm, che permette di ottenere immagini divergenti: il campo di vista si allarga allontanandosi dall'apparecchio, causando leggere distorsioni soprattutto nei campi prossimali.
- *Sonda microconvex*: ha una frequenza di utilizzo compresa tra 4 e 10 MHz. Questa sonda appartiene alla famiglia delle sonde Convex ed è impiegata per esami endocavitari.



Figura 1.2: Tipologie di sonde ecografiche. Da sinistra verso destra: lineare, phased array, convex, microconvex.

La porzione d'onda riflessa verso il trasduttore è denominata eco di ritorno. Tale eco è generato dall'interazione tra l'impulso emesso dalla sonda e un'interfaccia tissutale. In risposta all'eco di ritorno, ciascun elemento del trasduttore genera un segnale a radiofrequenza (RF). Durante questa fase di acquisizione, il segnale RF rappresenta un

dato grezzo (*Raw RF data*), raccolto individualmente da ciascun elemento (o canale della sonda); la matrice risultante è definita *segnale RF a livello di canale*. Questo segnale è bidimensionale e cattura la conversione dell'energia acustica di ritorno in energia elettrica in funzione del tempo:

- Asse orizzontale: rappresenta il tempo t , definito tempo di volo (TOF - *time of flight*) dell'onda, ovvero l'intervallo di tempo che intercorre dall'istante in cui l'impulso viene emesso all'istante in cui l'eco di ritorno viene registrato. Il TOF consente di determinare la profondità d della regione del tessuto che ha generato l'eco e, di conseguenza, anche la posizione spaziale e i confini del tessuto stesso. La distanza d può essere calcolata utilizzando la seguente formula matematica:

$$d = \frac{ct}{2} \quad (1.4)$$

dove c è la velocità del suono nel mezzo.

- Asse verticale: indica l'ampiezza del segnale elettrico, misurata in Volt (V), direttamente proporzionale all'intensità dell'eco di ritorno.

La conversione della grandezza acustica in elettrica avviene grazie all'effetto piezoelettrico diretto, come descritto in precedenza. L'eco acustico di ritorno, essendo una fluttuazione di pressione, colpisce il cristallo piezoelettrico all'interno della sonda ed esercita una forza meccanica su di esso. Il cristallo genera, quindi, una carica elettrica (voltage) proporzionale all'ampiezza dell'onda acustica.

Nel presente progetto di tesi è stata utilizzata una sonda lineare con 128 elementi piezoelettrici; ciascun elemento riceve un segnale distinto. L'acquisizione simultanea dei segnali provenienti da tutti gli elementi della sonda genera una matrice bidimensionale, in cui ciascuna entrata $RF(t, n)$ al tempo t e per l'elemento n rappresenta la tensione elettrica in Volt.

Tali dati costituiscono il punto di partenza per l'elaborazione e la visualizzazione dell'immagine.

1.1.3 Meccanismi di formazione dell'immagine: riflessione e trasmissione

Gli ecografi moderni si basano sul rilevamento e sulla visualizzazione degli echi di ritorno. La loro formazione richiede un'interfaccia riflettente, costituita da una giunzione tra tessuti o materiali con proprietà fisiche differenti. La quantità di energia riflessa dipende dalla differenza delle impedenze acustiche tra i materiali che costituiscono l'interfaccia. L'impedenza acustica (Z) è una grandezza fondamentale nel campo degli ultrasuoni, poiché caratterizza intrinsecamente il mezzo di propagazione e costituisce un valore chiave nelle applicazioni biomediche. Per questo motivo, è di interesse pratico conoscerne il valore nei diversi mezzi biologici (Tabella 1.2) [9]. In particolare, Z dipende dalla densità del mezzo in cui si propaga il suono ρ e dalla velocità di propagazione del suono nel mezzo c , secondo la relazione:

$$Z = \rho c \quad (1.5)$$

I tessuti con un elevato contenuto d'aria presentano un'impedenza acustica molto bassa, mentre la maggior parte dei tessuti biologici possiede un intervallo ristretto di valori di impedenza, ad eccezione dell'osso che mostra un'impedenza acustica molto più elevata. L'impedenza acustica dell'osso è circa cento volte superiore a quella del tessuto muscolare.

Materiale	ρ (25°C) [Kg/m ³]	c [m/s]	Z [10 ⁶ Kg s/m ²]
Aria	1.2	331	0,0004
Acqua	997	1497	1,493
Sangue	1050	1560	1,638
Muscolo	1060	1570	1,642
Osso	1850	3360	6,216
Fegato	1060	1595	1,64
Grasso	920	1478	1,36
Pelle	1090	1519	1,58
Vaso sanguigno	1080	1560	1,65

Tabella 1.2: Valori di impedenza acustica (Z) di diversi materiali. La tabella elenca la densità (ρ), la velocità del suono (c) e la conseguente impedenza acustica.

Quando le onde ultrasonore incontrano un'interfaccia riflettente, una parte dell'onda viene riflessa e una parte trasmessa. La riflessione e la rifrazione sono descritte mediante la legge di Snell.

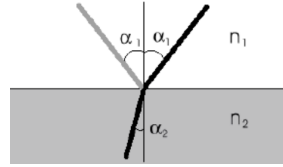


Figura 1.3: Riflessione e rifrazione di un'onda acustica. Il diagramma illustra un'onda acustica che incide sull'interfaccia tra due mezzi con indici di rifrazione differenti.

Se un'onda ha origine in un mezzo con indice di rifrazione n_1 ed entra in una regione con indice di rifrazione n_2 (come nella Figura 1.3), gli angoli di incidenza (α_1) e di rifrazione (α_2) sono legati dalla seguente equazione:

$$\frac{\sin(\alpha_1)}{\sin(\alpha_2)} = \frac{c_1}{c_2} = \frac{n_2}{n_1} \quad (1.6)$$

Qui c_1 e c_2 rappresentano le velocità di propagazione dell'onda nei due mezzi [10]. È possibile conoscere la frazione di energia riflessa e trasmessa rispettivamente attraverso il coefficiente di riflessione R e il coefficiente di trasmissione T [11].

$$R = \left(\frac{z_1 \cos \alpha_2 - z_2 \cos \alpha_1}{z_1 \cos \alpha_2 + z_2 \cos \alpha_1} \right)^2 \quad T = 1 - R \quad (1.7)$$

Nel caso in cui il fascio ultrasonoro incida perpendicolarmente sull'interfaccia dei tessuti, i coefficienti R e T sono determinati dai seguenti rapporti:

$$R = \frac{(Z_2 - Z_1)^2}{(Z_2 + Z_1)^2} \quad T = \frac{4Z_1Z_2}{(Z_1 + Z_2)^2} \quad (1.8)$$

1.1.4 Simulazione dell'equazione d'onda

Le acquisizioni in-vivo e in-vitro sono fondamentali per la ricerca ecografica; tuttavia, il loro impiego nella creazione di dataset per l'allenamento di algoritmi di *Machine Learning* presenta alcune limitazioni. In particolare, la mancata disponibilità di dati e la difficoltà nella loro generazione, unite alla mancanza della verità di riferimento (*ground truth*) e all'impossibilità di isolare l'effetto di singoli parametri fisici (come la velocità del suono e la densità), possono essere superate attraverso simulazioni di immagini ecografiche [8].

Modello Matematico

Il moto di un'onda acustica che si propaga in un mezzo comprimibile può essere descritto mediante le equazioni linearizzate del flusso acustico. Durante la propagazione di un'onda acustica in un mezzo, si verificano fluttuazioni dinamiche nella pressione, nella densità, nella temperatura e nella velocità delle particelle. Tali variazioni sono descritte da un sistema di equazioni differenziali parziali del primo ordine accoppiate, che esprimono la conservazione della massa, della quantità di moto e dell'energia nel mezzo. Nel caso ideale di un'onda acustica a piccola ampiezza che si propaga in un fluido omogeneo e non dissipativo, le equazioni di primo ordine sono date da:

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho_0} \nabla p, \quad (\text{conservazione quantità di moto}) \quad (1.9)$$

$$\frac{\partial p}{\partial t} = -\rho_0 \nabla \cdot u, \quad (\text{conservazione massa}) \quad (1.10)$$

$$p = c^2 \rho, \quad (\text{relazione pressione-densità}) \quad (1.11)$$

dove u è la velocità delle particelle acustiche, p rappresenta la pressione acustica, ρ_0 indica la densità dell'ambiente e c è la velocità del suono. Quando si considerano l'assorbimento acustico e le eterogeneità nei parametri del mezzo, il sistema di equazioni differenziali parziali del primo ordine accoppiate si modifica come segue:

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho_0} \nabla p, \quad (\text{conservazione quantità di moto}) \quad (1.12)$$

$$\frac{\partial p}{\partial t} = -\rho_0 \nabla \cdot u - u \cdot \nabla \rho_0, \quad (\text{conservazione massa}) \quad (1.13)$$

$$p = c^2(\rho + d \cdot \nabla \rho_0 - L\rho), \quad (\text{relazione pressione-densità}) \quad (1.14)$$

dove d rappresenta lo spostamento acustico delle particelle e L è un operatore che tiene conto dell'assorbimento e della dispersione acustica nel mezzo.

Simulazione mediante k-Wave

La simulazione numerica di questo sistema di equazioni differenziali può essere realizzata mediante il toolbox *k-Wave*. *k-Wave* è un toolbox open source in MATLAB, sviluppato per la simulazione nel dominio del tempo della propagazione delle onde acustiche in una, due o tre dimensioni. Il toolbox offre un'ampia gamma di funzionalità, il cui nucleo è rappresentato da un modello numerico avanzato in grado di tenere conto sia della propagazione lineare sia di quella non lineare delle onde, sia della distribuzione arbitraria dei parametri dei materiali eterogenei e dell'assorbimento acustico secondo la legge di potenza (*Power Law*). L'interfaccia di simulazione è stata progettata per essere intuitiva e flessibile, mentre il motore di calcolo è ottimizzato per garantire elevata efficienza e precisione. Le funzioni vengono eseguite mediante script MATLAB, nei quali l'utente definisce i parametri. *k-Wave* è tuttora in fase di sviluppo attivo e le sue funzionalità sono in continua evoluzione.[12] Il simulatore viene quindi utilizzato per la risoluzione delle equazioni differenziali al fine di ottenere come output il segnale grezzo RF a livello di canale (*Raw RF Data*). Questo segnale rappresenta la pressione acustica (o, in alternativa, la velocità acustica) in funzione del tempo. Per ogni elemento si ottiene la traccia temporale contenente le informazioni relative alle onde riflesse dalle diverse interfacce presenti, dalle quali è possibile ricavare la distanza mediante l'Equazione 1.4.

1.2 Beamforming

Nell'imaging ecografico in modalità luminosità (*B-mode*), gli echi generati da un impulso ultrasonico vengono impiegati per ricostruire le immagini in base alla loro intensità. Gli array di trasduttori generano un'interferenza costruttiva mediante l'emissione di segnali acustici opportunamente sfasati, concentrando la quasi totalità della potenza del segnale trasmesso lungo la direzione d'interesse. Questa tecnica, nota come *beamforming* (BF), consente di ottimizzare la ricezione del segnale. Il *beamformer* risolve il problema derivante dalle differenti distanze tra i trasduttori e il punto di focalizzazione, che altrimenti causerebbe la ricezione di segnali fuori fase. A tale scopo, viene introdotto un ritardo che compensa le diverse lunghezze di percorso: al trasduttore più vicino viene assegnato il ritardo maggiore e viceversa, in modo da allineare i segnali. La somma dei segnali rifasati consente di massimizzare l'informazione contenuta e di ricostruire correttamente l'immagine [13].

1.2.1 DAS beamforming

Il *Delay-And-Sum* (DAS) rappresenta il *beamformer* digitale più semplice e maggiormente impiegato per l'imaging ecografico medico. Grazie alla sua semplicità ed efficienza computazionale, è largamente impiegato nelle applicazioni ecografiche ad alta frequenza di fotogrammi (ultraveloce) [14]. Agendo opportunamente sui parametri di ritardo dei singoli componenti dell'array, è possibile controllare la direzione e la focalizzazione del fascio ultrasonoro.

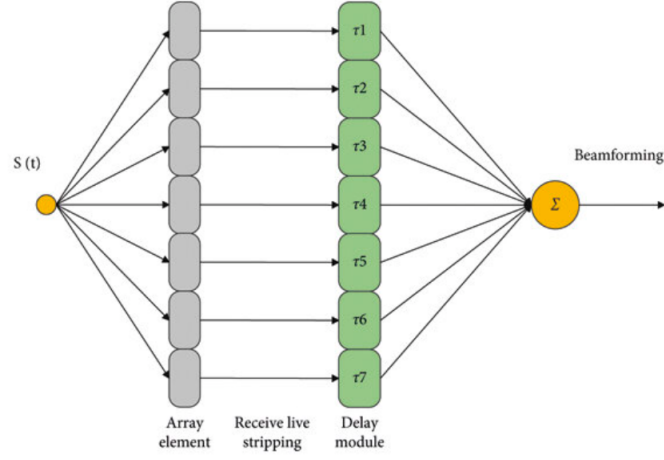


Figura 1.4: Schema a blocchi che illustra il principio del beamforming DAS. Un segnale acustico $S(t)$ viene acquisito da un array di trasduttori (raffigurato in grigio). Il segnale ricevuto da ciascun elemento dell'array viene ritardato in modo differenziato ($\tau_1, \tau_2 \dots \tau_7$, rappresentati dal modulo verde) per compensare le diverse distanze e angolazioni rispetto al punto di interesse. Successivamente, tutti i segnali temporizzati vengono sommati per ottenere un segnale finale coerente, migliorando la qualità e la risoluzione dell'immagine ecografica.

Il meccanismo del beamforming DAS è illustrato in Figura 1.4, prendendo come esempio un array di trasduttori con sette elementi di ricezione. Si indica con $S(t)$ il segnale d'eco riflesso dal punto focale. Per un array lineare, il principio DAS consiste nel calcolare, in funzione del punto di focalizzazione e della posizione geometrica di ciascun elemento, le differenze nei tempi; tali differenze si traducono in specifici ritardi applicati ai segnali d'eco ricevuti da ogni elemento dell'array. I segnali così ritardati vengono poi sommati coerentemente, generando il segnale finale di beamforming DAS, successivamente impiegato per la ricostruzione dell'immagine.

Il principio di calcolo del tempo di ritardo per il singolo elemento è illustrato nella Figura 1.5. Sia i il generico elemento dell'array. Si indica con r_1 la distanza tra il punto target e l'elemento i -esimo, con r_2 la distanza tra il punto target e il centro dell'array, e con x_i l'offset dell'elemento i rispetto al centro dell'array. Ne consegue che la differenza relativa del percorso del suono tra l' i -esimo elemento dell'array e il centro dell'array è $\delta S = r_1 - r_2$. Pertanto, il tempo di ritardo che dovrebbe essere impostato per l' i -esimo elemento dell'array è:

$$\tau_i = \frac{\delta S}{c} \quad (1.15)$$

Il segnale in uscita dal *beamformer* S_{DAS} può essere espresso come:

$$S_{DAS}(t) = \sum_{i=1}^N S_i(t - \tau_i) \quad (1.16)$$

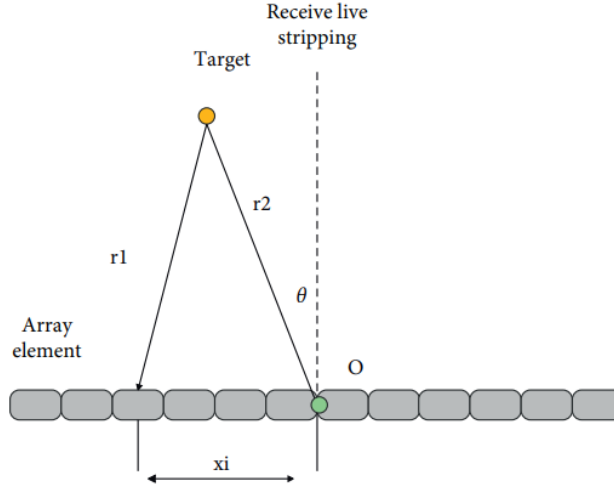


Figura 1.5: Principio del calcolo del ritardo temporale.

dove N rappresenta il numero di elementi dell'array, τ_i è il ritardo associato all' i -esimo elemento e c è la velocità del suono impostata durante il BF [13].

L'algoritmo standard DAS presenta alcune limitazioni nella formazione dell'immagine B-mode: ridotta risoluzione spaziale, elevata presenza di lobi laterali [15] e limitata capacità di sopprimere le interferenze provenienti dalle direzioni indesiderate [16]. I lobi laterali (*side lobes*) rappresentano echi indesiderati dovuti alla diffusione dell'energia acustica al di fuori del fascio principale. Questi degradano la qualità dell'immagine, riducendo il rapporto contrasto-rumore e la capacità di rilevare piccoli target, compromettendo la risoluzione dell'immagine. Una possibile soluzione consiste nell'applicazione dell'*apodizzazione* durante il beamforming. Questo processo prevede l'applicazione di una funzione di peso (o finestra di apodizzazione) ai segnali trasmessi e ricevuti lungo l'apertura del trasduttore, attenuando il contributo degli elementi laterali e privilegiando quello proveniente dagli elementi centrali. Il risultato è una significativa riduzione dei lobi laterali e un miglioramento della focalizzazione dell'energia nel punto di interesse [17].

La scelta della funzione di apodizzazione riveste un ruolo cruciale nello sviluppo dei sistemi ad US, poiché le caratteristiche del fascio influenzano direttamente il contrasto e la risoluzione spaziale dell'immagine ottenuta [18]. Tra le diverse funzioni di apodizzazione, le più comunemente implementate per la loro semplicità computazionale sono le finestre *Rettangolari*, *Triangolari* (o *Bartlett*), *Hamming*, *Hanning* (*Hann*), *Chebyshev* e *Gaussiane*.

Queste funzioni consentono di regolare il *trade-off* (compromesso) tra la larghezza del lobo principale e l'altezza dei lobi laterali. La risoluzione spaziale aumenta quanto più il lobo principale è stretto, mentre il contrasto migliora quanto più è ridotta l'ampiezza dei lobi laterali (minore dispersione di energia). In questo progetto di tesi è stata impiegata la finestra di Tukey, nota anche come finestra del coseno rastremato (*tapered*

cosine window). Tale finestra fornisce un ottimo equilibrio del trade-off, combinando i vantaggi di due funzioni base: la rettangolare e la Hanning. La sua forma è controllata dal parametro α (coefficiente di *tapering*):

- $\alpha = 0$: la finestra di Tukey equivale alla finestra rettangolare (massima risoluzione, minimo contrasto)
- $\alpha = 1$: la finestra di Tukey equivale alla finestra di Hanning (buon contrasto, minore risoluzione).

In Figura 1.6 è riportata la finestra a coseno rastremato: la porzione centrale è rettangolare (per mantenere il lobo principale stretto), mentre i bordi esterni sono rastremati fino a 0, con conseguente riduzione dei lobi laterali.

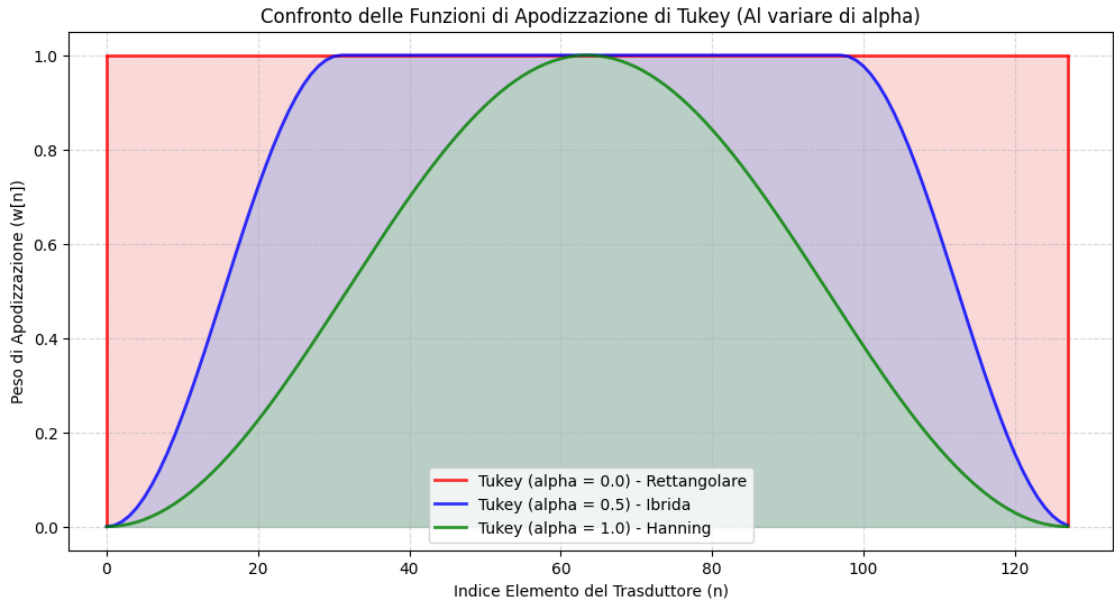


Figura 1.6: Confronto tra le funzioni di apodizzazione della finestra di Tukey per diversi valore di α . Per $\alpha = 1$ si ottiene la finestra di Hanning, mentre al diminuire di α la funzione assume un andamento simile a quella rettangolare.

1.2.2 Metodi di beamforming alternativi

Nel tempo sono stati sviluppati diversi metodi e algoritmi per migliorare le prestazioni del beamforming nell'imaging ecografico. Tra le principali alternative al DAS vi è il beamforming a varianza minima (MV), che offre una risoluzione più elevata rispetto al DAS ma a fronte di un costo computazionale maggiore. Negli ultimi anni, il beamformer adattivo MV è stato oggetto di numerose varianti al fine di ridurre la complessità computazionale [19]. Nel 2015, è stato introdotto il *delay multiply and sum* (DMAS) per l'imaging ad US [20]. Analogamente al DAS, il DMAS applica ritardi ai segnali

ricevuti dagli elementi dell'array per ottenere la focalizzazione degli echi. Tuttavia, la differenza principale è che, invece della somma diretta, i segnali ritardati vengono accoppiati e moltiplicati tra loro. Questa operazione rende l'algoritmo non lineare. Il diagramma a blocchi dell'algoritmo è riportato in Figura 1.7. Il segnale $y_{\text{DMAS}}^*(t)$ in

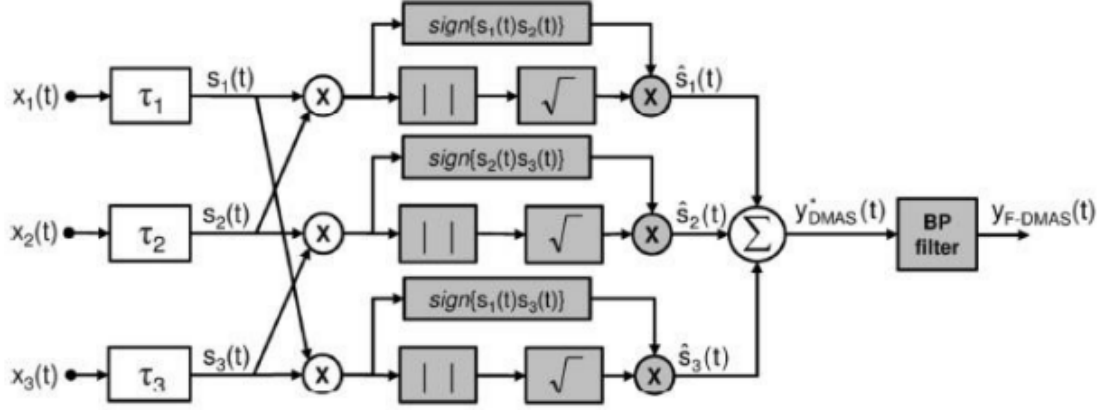


Figura 1.7: Schema a blocchi che illustra il principio del beamforming DMAS.

uscita dal beamformer è definito come:

$$y_{\text{DMAS}}^*(t) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \text{sign}(s_i(t)s_j(t)) \sqrt{|s_i(t)s_j(t)|} \quad (1.17)$$

dove $s_i(t)$ indica il segnale RF ritardato ricevuto dall'elemento i . L'operazione può essere interpretata come una correlazione tra i segnali: a ogni istante, il prodotto tra coppie di canali realizza una forma di cross-correlazione spaziale tra i segnali ricevuti dagli elementi attivi.

Un altro algoritmo di beamforming proposto in letteratura per mitigare alcune limitazioni del DAS è il *Coherent Factor (CF)* beamforming. L'idea consiste nel pesare il segnale ottenuto dal DAS mediante un fattore di coerenza. Il fattore di coerenza è una metrica che quantifica la qualità della focalizzazione dell'onda ricevuta: misura quanto l'energia dell'ultrasuono sia concentrata nel lobo primario rispetto ai lobi secondari [21]. Il CF è una metrica di coerenza spaziale definita come il rapporto tra la somma coerente e quella incoerente dei segnali RF ricevuti dagli elementi dell'array, dopo l'applicazione dei ritardi di focalizzazione. Il valore del CF in funzione del tempo t è espresso dalla seguente formula matematica:

$$CF(t) = \frac{\left| \sum_{i=1}^N s_i(t) \right|^2}{N \sum_{i=1}^N s_i^2(t)} \quad (1.18)$$

Il fattore di coerenza assume valori prossimi a 1 quando la coerenza del segnale è alta, ossia quando la qualità della focalizzazione dell'onda è buona; viceversa, tende a valori

prossimi allo zero quando la qualità della focalizzazione è ridotta. Una volta calcolato, il fattore può essere utilizzato per ottenere il segnale pesato:

$$y_{CF}(t) = CF(t)y_{DAS}(t) \quad (1.19)$$

1.2.3 Limitazioni del modello a velocità costante (1540 m/s)

Un aspetto dell'imaging a ultrasuoni che lo distingue da altre tecniche di imaging, come la risonanza magnetica e la tomografia computerizzata, è la presenza di distorsioni e aberrazioni del fascio ultrasonoro dovute alle caratteristiche dei tessuti attraversati. Una delle forme più semplici di aberrazione è dovuta alla variazione della velocità del suono (SoS) nei diversi tessuti, in particolare alla discrepanza tra la SoS assunta nel processo di beamforming e quella effettiva del tessuto [22]. I sistemi ecografici assumono una velocità di propagazione del suono costante pari a 1540 m/s. In realtà, la velocità del suono è influenzata dalla densità e dal modulo di elasticità del mezzo attraversato, parametri che variano da tessuto a tessuto. La velocità di propagazione aumenta nei tessuti caratterizzati da maggiore rigidità e minore densità [23]. Nella Tabella 1.2 sono riportati alcuni esempi di SoS nei principali tessuti biologici. In particolare, nei tessuti molli, la SoS è inferiore rispetto alla maggior parte degli altri tessuti (1440 m/s contro 1540 m/s). Pertanto, l'assunzione di una SoS costante pari a 1540 m/s nel BF può introdurre errori nella messa a fuoco, poichè comporta una discrepanza tra i ritardi temporali calcolati e quelli effettivi di propagazione nei tessuti [4].

La correzione dell'aberrazione è oggetto di studio a livello accademico da molti anni. L'approccio basato sulla correlazione incrociata ha prodotto risultati convincenti in contesti sperimentali controllati, ma non è stato sufficientemente robusto da consentirne l'impiego in prodotti commerciali, poichè la distorsione è un fenomeno bidimensionale che non può essere risolto con un array monodimensionale. Altri approcci di correzione dell'aberrazione si basano sull'analisi della luminosità dei punti focali [24]. Più recentemente, sono stati proposti metodi combinati di beamforming adattivo, in grado di migliorare la qualità complessiva dell'immagine. Tuttavia, l'implementazione di questi metodi risulta complessa a causa dell'elevato costo computazionale [25]. Un approccio alternativo per ridurre le distorsioni e le aberrazioni di fase consiste nell'impiegare una stima accurata della SoS. A tal fine, sono stati proposti diversi algoritmi per la stima della SoS [26, 27]. Più recentemente, è stato introdotto un metodo di stima diretta, basato sul miglior adattamento dei modelli di ritardo geometrico unidirezionale ai dati del canale a radiofrequenza (RF) pre-beamforming [28]. In aggiunta, è stato introdotto il fattore di coerenza (CF) nell'imaging ecografico, utilizzato per stimare la SoS nei tessuti, poichè consente di individuare eventuali errori di focalizzazione nel beamforming [29]. Inoltre, è stato proposto un metodo per la stima della velocità media del suono ottimale, caratterizzato da un'elevata efficienza hardware. In tale approccio, il fattore di qualità della focalizzazione viene valutato attraverso il calcolo della *Minimum Average Sum of Absolute Differences* (MASAD) sui dati RF pre-beamforming, migliorando così la risoluzione spaziale nell'imaging ecografico medico [30].

Tra i metodi più recenti, merita particolare attenzione la tecnica che consente di stimare in modo robusto la SoS nei tessuti mediante una singola trasmissione pulse-echo,

utilizzando un trasduttore. Questo approccio ha reso possibile l'integrazione della stima della SoS all'interno della catena di elaborazione delle immagini ecografiche. La metodologia introduce un nuovo algoritmo di stima della SoS globale che, a differenza dei metodi precedenti, riduce significativamente la complessità computazionale. Tale metodo, definito *single-shot*, stima la velocità del suono calcolando i coefficienti di autocorrelazione dei segnali pre-beamforming, definendo una funzione di perdita valutata su diversi valori candidati di SoS e selezionando quello che la minimizza [31].

In continuità con questo approccio di stima della SoS globale, il presente progetto di tesi propone un nuovo metodo basato sull'analisi di tessitura per determinare la SoS ottimale mediante l'implementazione di un modello di regressione basato su un algoritmo di *Machine Learning*, descritto successivamente in dettaglio.

1.3 Post processing

Dopo il beamforming, si ottengono tipicamente tra 64 e 256 (o più) linee di dati in radiofrequenza beamformati (*Beamformed RF signals*), che rappresentano il segnale grezzo, ossia il segnale acustico retrodiffuso proveniente dalle diverse direzioni di scansione. Queste fasi iniziali di elaborazione, che si estendono fino al trasferimento dei segnali al back-end, costituiscono il front-end del sistema. Successivamente, viene applicata la *Time Gain Compensation* (TGC), una procedura che consente di compensare l'attenuazione del segnale dovuta alla propagazione nel mezzo.

1.3.1 Time Gain Compensation e Demodulazione RF

La TGC regola il guadagno del segnale in funzione del tempo o della profondità, al fine di ottenere un'immagine con luminosità uniforme, sia nelle regioni superficiali sia in quelle più profonde. Gli echi provenienti dalle strutture più profonde risultano attenuati rispetto a quelli generati dalle strutture superficiali, a causa della perdita di intensità del suono durante la propagazione, come descritto in precedenza. Per compensare selettivamente gli echi più deboli, il sistema incrementa il guadagno in funzione del tempo di volo dell'eco. Le impostazioni della TGC vengono rappresentate mediante una curva caratteristica (come mostrato in Figura 1.8).

Successivamente, si applica la demodulazione in radiofrequenza, che converte il segnale in un segnale analitico $I(t) + jQ(t)$, costituito da due componenti che variano nel tempo t :

- *In-phase* (I): componente del segnale in fase con il segnale di riferimento.
- *Quadratura* (Q): componente del segnale sfasata di 90° rispetto alla componente in-phase.

La demodulazione viene effettuata moltiplicando il segnale RF (ad alta frequenza) per un segnale di riferimento, al fine di ridurre la larghezza di banda.

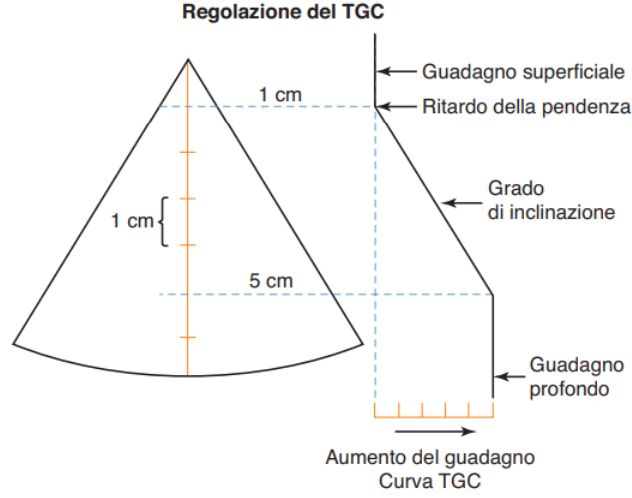


Figura 1.8: Rappresentazione TGC. La curva mostra il guadagno relativo applicato a ogni specifica profondità. L'aumento del guadagno (amplificazione degli echi di ritorno) all'aumentare della profondità viene rappresentato con la visualizzazione della curva sempre più a destra procedendo verso il basso.

1.3.2 Back-end

Una volta completata questa prima fase di elaborazione, i segnali vengono trasferiti al back-end del sistema. La prima operazione è *l'envelope detection*, durante la quale si calcola l'ampiezza dell'involuppo per estrarre l'intensità degli echi:

$$B_a = \sqrt{I^2(t) + Q^2(t)} \quad (1.20)$$

In questa fase, il segnale presenta un'ampia gamma dinamica (pari o superiore a 80 dB). Poichè i monitor di visualizzazione e il sistema visivo umano dispongono di un intervallo dinamico compreso tra 30 e 40 dB, il segnale viene sottoposto a compressione logaritmica:

$$B_b = 20 \log_{10}[B_a(t)] \quad (1.21)$$

Il segnale viene ulteriormente sottoposto a filtraggio per ridurre il rumore. Il filtraggio viene eseguito sia in direzione assiale sia laterale, con l'obiettivo di migliorare il rapporto segnale-rumore. Successivamente, un'elaborazione di persistenza applica un filtraggio temporale adattivo per gestire gli artefatti di movimento (nel caso di sequenze di fotogrammi). In seguito, è necessario eseguire la *digital scan conversion* (DSC), che consente di convertire i dati ecografici dalle coordinate polari a quelle cartesiane per la visualizzazione. Il convertitore di scansione esegue operazioni di interpolazione tra valori di pixel adiacenti, attenuando così gli effetti del ricampionamento delle coordinate. Prima della visualizzazione delle immagini, viene applicata una correzione gamma per compensare la sensibilità non lineare del monitor e del sistema visivo umano ai diversi livelli di luminosità. Sebbene l'elaborazione in modalità B sia simile nella maggior

parte dei sistemi ecografici, l'ordine delle fasi di elaborazione può variare da un sistema all'altro [32].

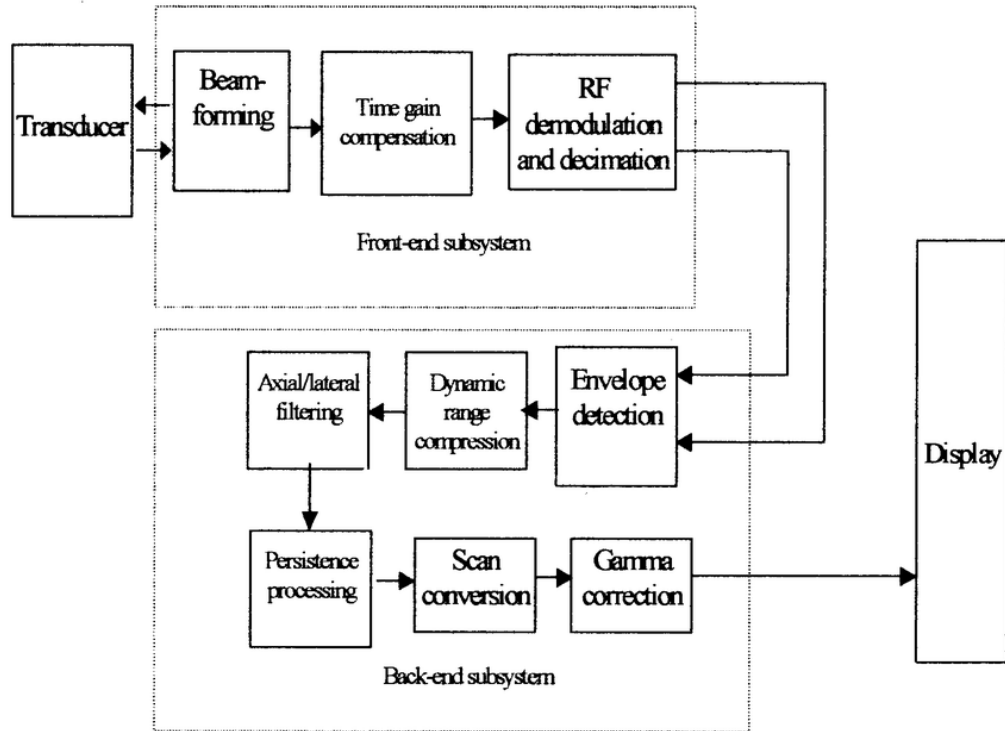


Figura 1.9: Pipeline di elaborazione per modalità Brightness.

Capitolo 2

Machine Learning

2.1 Introduzione al Machine Learning

I computer sono in grado di risolvere, in tempi estremamente ridotti, problemi di natura matematica che per un essere umano risulterebbero particolarmente complessi. Al contrario, esistono operazioni che per un essere umano risultano naturali, ma che per una macchina si rivelano complesse, come ad esempio il riconoscimento dei volti, degli oggetti o della voce. In tali casi, la programmazione tradizionale a codifica fissa risulta poco efficace [33]. Per questo motivo, si ricorre a approcci basati su algoritmi in grado di simulare il processo cognitivo umano, ovvero l'apprendimento dai dati. Tali metodi vengono definiti di *Machine Learning* (ML) o di apprendimento automatico.

Oggi, con il termine *Machine Learning* si intende la capacità di una macchina di apprendere e imitare il comportamento e l'intelligenza umana [34]. Il Machine Learning costituisce un campo dell'Intelligenza Artificiale (IA) che si basa sulla capacità delle macchine di apprendere autonomamente attraverso lo sviluppo di algoritmi. A differenza della programmazione classica, in cui le regole sono definite esplicitamente dall'essere umano e l'algoritmo viene codificato manualmente, nel ML è il sistema stesso a generare un modello sulla base dei dati a disposizione. Negli ultimi anni, le macchine hanno dimostrato la capacità di apprendere e padroneggiare compiti che in passato erano considerati troppo complessi, evidenziando il loro potenziale nell'ambito della diagnosi assistita e del supporto decisionale. Ancora più sorprendente è la constatazione che alcuni algoritmi di apprendimento automatico sono in grado di individuare pattern e relazioni non immediatamente percepibili dall'uomo. Questo risultato ha alimentato l'interesse crescente verso il ML, soprattutto per le sue potenziali applicazioni alle immagini mediche [33].

2.1.1 Metodi di apprendimento

La maggior parte degli algoritmi di ML riceve in input un insieme di esempi, denominato *Dataset*, che può essere costituito da immagini, vettori contenenti numeri o altre rappresentazioni di dati, dai quali il modello apprende per eseguire un determinato

compito [35]. A seconda del tipo di esperienza, ovvero della natura e tipologia dei dati forniti al modello, si possono definire quattro categorie di apprendimento:

- *Supervisionato*: i dati di input sono accompagnati da etichette che indicano l'output atteso. L'obiettivo del modello è apprendere una funzione in grado di mappare correttamente gli input sugli output corrispondenti. Questo approccio è comunemente utilizzato nei problemi di regressione e classificazione.
- *Non Supervisionato*: i dati di input non sono etichettati. Questi algoritmi mirano a rilevare pattern, correlazioni o strutture nei dati; e sono comunemente utilizzati per tecniche di *clustering* [35].
- *Semi-supervisionato*: rappresenta un approccio intermedio tra l'apprendimento supervisionato e quello non supervisionato, utile quando si dispone di un dataset non completamente etichettato. Questo approccio è tipicamente impiegato nell'imaging medicale. Durante la fase di addestramento, viene fornita solo la parte del dataset etichettata, mentre la restante parte viene etichettata dal modello. Successivamente, il modello finale viene addestrato sull'intero dataset.
- *Con rinforzo*: si costruisce un sistema che migliora le prestazioni interagendo con l'ambiente [33].

2.1.2 Compiti di applicazione

Gli algoritmi di Machine Learning, come precedentemente accennato, hanno il compito di risolvere problemi semplici per gli esseri umani, che per i programmi tradizionali risultano complessi. Le prestazioni del modello vengono valutate in termini di performance o accuratezza, utilizzando parametri opportunamente selezionati in funzione del tipo di modello impiegato. I principali compiti svolti dagli algoritmi di ML possono essere ricondotti a tre categorie principali:

- *Classificazione*: l'algoritmo assegna ciascun dato di input a una specifica categoria, selezionandola all'interno di un insieme predefinito di classi. L'algoritmo può inoltre stimare la probabilità che un dato appartenga a una determinata classe piuttosto che a un'altra. In questo tipo di problema, l'output assume un numero finito di valori discreti.
- *Regressione*: a partire da uno o più dati di input, l'algoritmo prevede il valore di una o più variabili di output continue.
- *Clustering* (raggruppamento): l'algoritmo suddivide i dati in gruppi omogenei, in modo che gli elementi appartenenti allo stesso gruppo presentino caratteristiche o comportamenti simili [33, 36].

I compiti di classificazione e regressione vengono generalmente affrontati mediante tecniche di apprendimento supervisionato, mentre il clustering si basa su metodi di apprendimento non supervisionato [35].

2.1.3 Valutazione delle prestazioni

Le metriche impiegate per valutare le prestazioni dipendono strettamente dal tipo di modello adottato e dal compito di apprendimento considerato. L'obiettivo principale di un algoritmo di Machine Learning (ML) è massimizzare la capacità di generalizzare le proprie prestazioni su dati mai visti durante l'addestramento. Per questo motivo, il dataset iniziale viene comunemente suddiviso in tre sottoinsiemi distinti: *training set*, *validation set* e *test set*. Il *training set*, che generalmente rappresenta il 60–80% del dataset totale, viene utilizzato per addestrare il modello, ovvero per l'ottimizzazione dei pesi e la definizione delle regole decisionali. Il *validation set* viene impiegato per ottimizzare gli iperparametri del modello e per monitorare l'eventuale presenza di fenomeni di *overfitting* e *underfitting*. Un modello è affetto da *overfitting* (sovradattamento) quando apprende eccessivamente i dettagli e il rumore presenti nei dati di addestramento, ottenendo così prestazioni elevate sul *training set* ma scarse sui nuovi dati. Al contrario, si parla di *underfitting* (sottodattamento) quando il modello non riesce a rappresentare adeguatamente la complessità del problema, mostrando prestazioni insoddisfacenti anche sui dati di addestramento. Infine, il *test set* viene utilizzato esclusivamente per valutare le prestazioni finali del modello e stimare la capacità di generalizzazione su dati non visti [33].

2.2 Modelli di apprendimento supervisionato

In questo lavoro viene posta particolare attenzione agli algoritmi di apprendimento supervisionato applicati ai problemi di regressione. La regressione rappresenta un problema in cui si dispone di un insieme di variabili indipendenti $X = (x_1, x_2, \dots, x_n)$ e una variabile target continua y . L'obiettivo dell'algoritmo è individuare una funzione f in grado di approssimare nel modo più accurato possibile la relazione esistente tra le variabili indipendenti e il target:

$$y = f(X, \beta) + \epsilon \quad (2.1)$$

dove β rappresenta un vettore di n parametri, mentre ϵ è un termine di errore aleatorio (ovvero la componente non spiegata dal modello) la cui distribuzione può essere assunta come normale o non normale, a seconda dei casi. A seconda della forma assunta dalla funzione f , si distinguono differenti categorie di algoritmi supervisionati [37].

La prima distinzione consente di suddividerli in due categorie:

- *Algoritmo parametrico*: la funzione f da apprendere è caratterizzata da un numero fisso e ben definito di parametri. L'apprendimento consiste nella stima dei valori ottimali di tali parametri al fine di minimizzare una funzione di errore. A questa categoria appartengono, ad esempio, la regressione lineare, la regressione logistica, la backpropagation e il Naïve bayes.
- *Algoritmo non parametrico*: si adatta ai dati a disposizione attraverso una distribuzione o un modello privo di ipotesi, mantenendo comunque una buona generalizzazione sui dati mai visti. In questo caso, la funzione f non ha una forma

predefinita. Questa categoria include i seguenti algoritmi: K-Nearest Neighbors (KNNs), Support Vector Machines (SVMs), Decision trees (Alberi decisionali) e Metodi di Ensemble (Random Forest e Metodi di Boosting) [38].

2.2.1 Algoritmi di apprendimento parametrici

I modelli parametrici risultano più facili da utilizzare, stimare e interpretare. In questi modelli, il vettore dei parametri β (definito nell'Equazione 2.1) ha una dimensionalità pari a p , che può essere maggiore o minore rispetto al numero di variabili indipendenti n . Il ricercatore assume a priori la forma della funzione f e la distribuzione dell'errore ϵ . Dopo aver stimato i parametri del modello, è necessario verificare che le ipotesi assunte siano valide. Questo avviene analizzando i residui (e_i), definiti come le differenze tra i valori stimati (\hat{y}_i) e quelli reali (y_i):

$$e_i = \hat{y}_i - y_i \quad (2.2)$$

I residui rappresentano quanto il modello si discosta dai dati reali e costituiscono una stima empirica degli errori teorici ϵ_i , i quali, definiti a priori, descrivono la componente casuale del modello. Si procede quindi con un'analisi di validità delle ipotesi effettuate, confrontando la distribuzione dei residui ottenuta empiricamente con la distribuzione ϵ assunta.

Successivamente, per valutare la qualità complessiva dell'approssimazione e individuare i parametri ottimali β , si impiegano i metodi di ottimizzazione. Essi permettono di minimizzare una *funzione di costo* che quantifica quanto le previsioni dell'algoritmo si discostino dai risultati attesi, tenendo conto di tutti i residui e_i ottenuti sul training set [37]. La discesa del gradiente (*Gradient Descent*) rappresenta uno dei metodi di ottimizzazione più utilizzati [35]. Questo metodo si basa sul concetto matematico di derivata di una funzione $f(x)$, calcolata in un punto x_0 , che rappresenta la pendenza della funzione in quel punto. Generalizzando il concetto di derivata a funzioni definite su uno spazio multidimensionale, si definisce il gradiente. Il gradiente $\nabla f(x)$ di una funzione, definito in un punto x_0 qualsiasi, è un vettore che indica la direzione di massima crescita della funzione in x_0 . Pertanto, l'opposto del gradiente fornisce la direzione di massima discesa:

$$d = -\nabla f(x)$$

Variando in modo iterativo il valore della variabile x nella direzione di d , il modello converge verso il punto di minimo della funzione.

$$x' = x + \epsilon d$$

dove ϵ rappresenta la lunghezza dello spostamento (o *learning rate*). Tale procedimento consente di aggiornare progressivamente i parametri del modello supervisionato, al fine di minimizzare la funzione di errore e migliorare la capacità predittiva dell'algoritmo [36].

Modelli lineari

Tra gli algoritmi di apprendimento supervisionato di tipo parametrico, i più semplici sono quelli lineari. Un esempio è rappresentato dall'utilizzo di una funzione polinomiale f lineare nei parametri β . Esistono diversi modelli lineari, tra i quali si possono menzionare i seguenti:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon \quad (2.3)$$

(Modello di regressione lineare multipla.)

$$Y = \beta_0 + \beta_{10} x_1 + \beta_{11} x_1^2 + \beta_{20} x_2 + \beta_{21} x_2^2 + \varepsilon \quad (2.4)$$

(Modello di regressione polinomiale di secondo ordine)

In questa tipologia di modelli, è possibile utilizzare diversi metodi per la stima dei parametri ottimali. Tra questi, sicuramente il metodo dei minimi quadrati (OLS) è quello più diffuso. Questo metodo di ottimizzazione consiste nel minimizzare lo scarto quadratico medio (MSE), definito come:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.5)$$

Solitamente, poichè la funzione obiettivo è nota, la soluzione può essere espressa in forma analitica chiusa. Tuttavia, a volte, essa può assumere una forma complessa e richiedere, quindi, l'utilizzo di metodi di risoluzione iterativi, come la discesa del gradiente [36, 37].

Modelli non lineari

I modelli di regressione non lineare sono modelli parametrici e la funzione f è nota, ma non lineare rispetto ai parametri. Alcuni esempi possono essere:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 e^{(\beta_3 x_2)} + \varepsilon \quad (2.6)$$

$$Y = \frac{e^{\beta_0 + \beta_1 x_1}}{1 + e^{\beta_0 + \beta_1 x_1}} + \varepsilon \quad (2.7)$$

$$Y = \beta_0 + (0.4 - \beta_0) e^{-\beta_1 (x_1 - 5)} + \beta_2 x_2 + \varepsilon \quad (2.8)$$

Nel caso dei modelli non lineari, la stima dei parametri mediante il metodo dei minimi quadrati risulta più complessa e richiede l'impiego di algoritmi iterativi. Tra i più utilizzati si menzionano il metodo di Newton, l'algoritmo di Gauss-Newton e il metodo di Levenberg-Marquardt. Inoltre, alcuni modelli non lineari possono essere resi lineari nei parametri attraverso semplici trasformazioni.

Tuttavia, in molte applicazioni, i modelli lineari non si adattano correttamente ai dati [37].

2.2.2 Algoritmi di apprendimento non parametrici

Il termine 'non parametrico' non implica che il modello sia privo di parametri β , ma che il loro numero possa variare in funzione dei dati impiegati [37, 38]. In questi modelli, i parametri appartengono a sottoinsiemi di spazi vettoriali di dimensione infinita e le funzioni obiettivo f non sono note a priori. Di conseguenza, tali funzioni potrebbero essere sia lineari sia non lineari [37]. Tutto ciò permette una maggiore flessibilità nella modellazione di relazioni complesse tra variabili di input e di output, grazie alla capacità del modello di adattarsi ai dati senza imporre assunzioni rigide [39]. Questo approccio è consigliato quando la relazione non è facilmente rappresentabile tramite un'equazione. Tra i principali metodi di apprendimento supervisionato non parametrico si menzionano gli Alberi decisionali (*Decision Trees*) e i loro derivati, i *Metodi di Ensemble*, come le Foreste casuali (*Random Forests*) e il Potenziamento del gradiente (*Gradient Boosting*) [38].

Alberi Decisionali

Gli alberi decisionali offrono il vantaggio di generare regole comprensibili dagli esseri umani, utili per classificare un dato esempio [40]. Sono modelli predittivi utilizzati sia per compiti di regressione sia per compiti di classificazione [41]. Il loro nome deriva dalla struttura ad albero, simile a un diagramma di flusso. Il presente metodo adotta un approccio *greedy* (avido), in cui gli alberi decisionali vengono costruiti in modo ricorsivo dall'alto verso il basso, secondo il principio *divide et impera*. La maggior parte degli algoritmi per l'induzione degli alberi decisionali utilizza un approccio *top-down*, che parte da un insieme di addestramento composto da tuple di attributi (cioè le caratteristiche utilizzate per prendere decisioni) e dalle relative etichette di classe associate. Tale insieme viene suddiviso ricorsivamente in sottoinsiemi sempre più piccoli, man mano che la costruzione dell'albero procede [35]. Un singolo albero di decisione è composto da:

- *Nodi*: rappresentano i punti in cui avviene una decisione o una verifica relativa a una caratteristica del dataset.
- *Rami*: collegano i nodi e rappresentano i risultati del test.
- *Foglie*: contengono le etichette delle classi.

Il processo di stima dell'output in un albero di regressione può essere descritto mediante il seguente flusso:

1. Si parte dal nodo radice, ossia il nodo situato all'inizio dell'albero, al quale vengono assegnati tutti gli esempi di addestramento.
2. Si seleziona un attributo in base a un criterio di suddivisione definito a priori.
3. Si generano un certo numero di nodi figli, pari al numero di intervalli dei valori dell'attributo scelto; ciascun esempio di addestramento viene assegnato al nodo figlio corrispondente.

4. Si procede in modo ricorsivo, utilizzando come nuove radici i nodi generati;
5. La costruzione dell'albero prosegue fino al raggiungimento di una o più condizioni di arresto, tra le quali:
 - la varianza dei valori della variabile target è inferiore a una soglia minima stabilita a priori;
 - non sono più disponibili attributi sui quali suddividere l'albero;
 - è stata raggiunta la profondità massima dell'albero, definita a priori.

Durante l'addestramento, l'albero adatta le regole decisionali dei propri nodi al fine di separare nel modo più efficace i dati di addestramento. Una volta raggiunto un nodo foglia, l'algoritmo restituisce come valore di output la media (o, in alternativa, la mediana) dei valori della variabile target associati ai dati di addestramento presenti nel nodo.

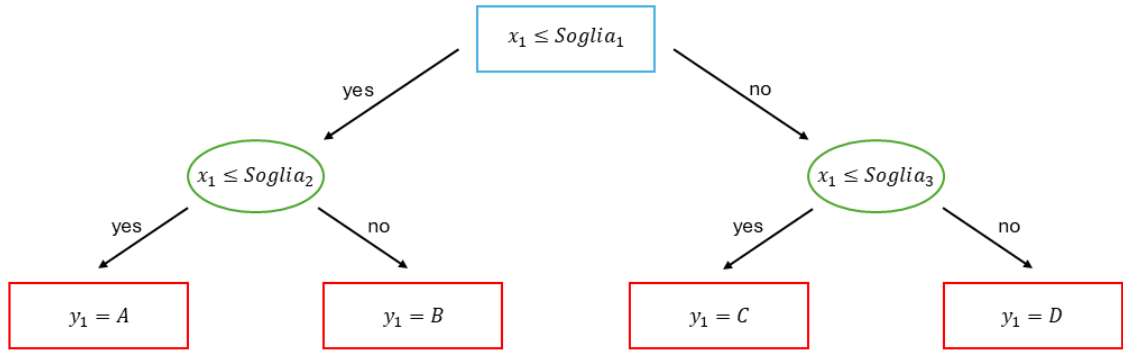


Figura 2.1: Schema illustrativo di un albero decisionale con profondità pari a 2. Il nodo in azzurro rappresenta la radice da cui parte il processo, i nodi in verde indicano le decisioni intermedie, mentre i nodi in rosso rappresentano le foglie contenenti i valori di output.

Nel secondo passo, l'algoritmo ha l'obiettivo di selezionare l'attributo che permette di ottenere la migliore separazione tra gli esempi, suddividendo i campioni eterogenei in sottoinsiemi il più possibile omogenei rispetto alla variabile di output. Nel caso degli alberi di regressione, in cui la variabile di output è continua, le metriche comunemente utilizzate vengono definite in termini di dispersione dei valori target. In particolare, il criterio di suddivisione più diffuso si basa sulla riduzione della varianza: viene selezionato l'attributo (e l'eventuale soglia) che minimizza la varianza complessiva della variabile di output nei nodi risultanti. Sia S l'insieme degli esempi associati a un nodo dell'albero; si definisce il parametro denominato impurità della varianza (*Variance Impurity*) come:

$$I_{\text{var}}(S) = \frac{1}{|S|} \sum_{i \in S} (y_i - \bar{y}_S)^2 \quad (2.9)$$

dove y_i rappresenta il valore della variabile target dell'esempio i -esimo, \hat{y}_S indica la media dei valori target delle istanze contenute nel nodo S e $|S|$ è il numero di esempi. Questa misura quantifica il grado di dispersione dei valori di output all'interno del nodo: quanto più è bassa la varianza, tanto più la scelta dello split ottimale avviene massimizzando la riduzione dell'impurità di varianza tra il nodo padre e i nodi figli. Essa può essere espressa in generale dalla relazione:

$$\Delta I_{\text{var}}(S, A) = I_{\text{var}}(S) - \left(\frac{|S_L|}{|S|} I_{\text{var}}(S_L) + \frac{|S_R|}{|S|} I_{\text{var}}(S_R) \right) \quad (2.10)$$

dove:

- S_L e S_R rappresentano rispettivamente i sottoinsiemi del nodo S ottenuti a seguito della suddivisione in base all'attributo A ;
- $|S_L|$ e $|S_R|$ sono i numeri di esempi dei due sottoinsiemi;
- $I_{\text{var}}(S)$ indica l'impurità di varianza del nodo S .

L'algoritmo seleziona la suddivisione che massimizza $\Delta I_{\text{var}}(S, A)$, poiché corrisponde alla suddivisione che produce nodi figli più omogenei e, di conseguenza, una minore dispersione dei valori della variabile target [42].

La diffusione degli alberi decisionali è dovuta alla loro semplicità di implementazione e di interpretazione, nonché alla capacità di adattarsi ai dati di addestramento. Tuttavia, questa caratteristica espone il modello al problema dell'*overfitting*, ossia alla tendenza ad adattarsi eccessivamente ai dati di addestramento, perdendo la capacità di generalizzazione sui dati non visti. Per ridurre la probabilità di *overfitting*, è comune applicare un algoritmo di affinamento denominato *pruning* (potatura). Esistono due approcci di potatura: il *pre-pruning* e il *post-pruning*. Nel primo caso, la costruzione dell'albero viene interrotta anticipatamente quando vengono soddisfatte determinate condizioni di arresto (ad esempio, quando si decide di non suddividere ulteriormente i nodi). Nel secondo approccio, più comune, il *post-pruning* consiste nella rimozione di sottoalberi da un albero già completamente costruito. In questo caso, un sottoalbero associato a un nodo interno viene sostituito da una foglia, la quale rappresenta il valore medio delle istanze presenti in quel nodo [35, 41].

Metodi di Ensemble

I modelli basati su un singolo albero decisionale presentano alcuni limiti intrinseci, tra cui l'elevata varianza e la tendenza all'*overfitting*. Piccole variazioni nei dati di addestramento possono generare alberi profondamente diversi e con prestazioni non sempre stabili. Per superare tali limitazioni, sono stati sviluppati i metodi d'insieme (o di *ensemble*), che combinano più modelli deboli per ottenere un modello complessivo più accurato e robusto. I modelli d'insieme più comuni sono il *Bagging* e il *Boosting*, che verranno trattati nella sezione successiva. Questa tipologia di modelli combina semplici modelli di regressione, detti stimatori di base. Durante la fase di predizione, ogni

stimatore di base restituisce una stima numerica del valore di output e l'*ensemble* aggrega queste stime, solitamente tramite la media aritmetica, per produrre la previsione finale. Anche se i singoli stimatori possono commettere errori, la combinazione delle loro predizioni tende a compensare le imprecisioni individuali, generando un risultato complessivo più stabile e accurato (come mostrato in Figura 2.2) [43]. L'efficacia di un ensemble dipende dalla diversità dei modelli base: la combinazione risulta tanto più vantaggiosa quanto minore è la correlazione tra gli stimatori. Inoltre, poiché ogni modello può essere addestrato in modo indipendente, questi metodi risultano naturalmente parallelizzabili ed efficienti anche su dataset di grandi dimensioni [35].

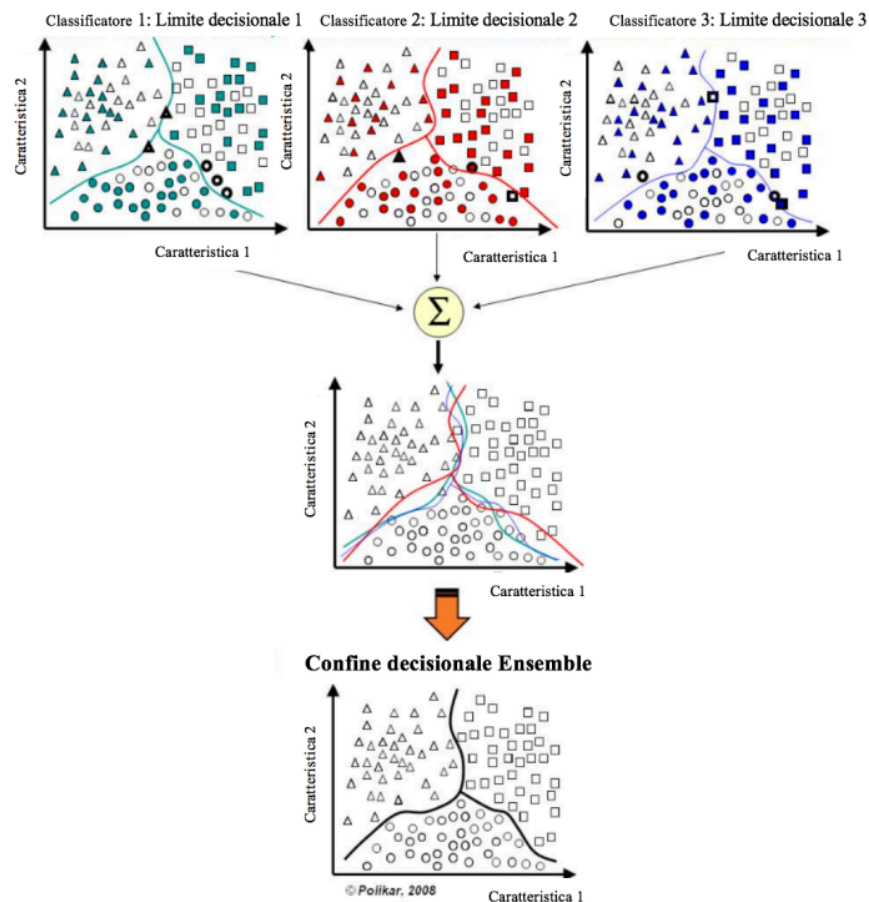


Figura 2.2: Rappresentazione metodo di insieme. L'insieme dei dati contiene due attributi differenti e il sistema utilizza tre sistemi di classificazione per modellare l'informazione.

2.3 Bagging: Random forest

La tecnica del *Bagging* è un metodo d'insieme che può essere utilizzato per compiti di regressione e classificazione. È stata progettata con l'obiettivo di superare il limite dell'*overfitting* presente nel singolo albero decisionale. La Figura 2.3 illustra il funzionamento di questo metodo. A partire da un dataset contenente K esempi di addestramento, vengono generati N sottoinsiemi della stessa dimensione tramite campionamento casuale con reinserimento; ciò implica che alcuni esempi possono comparire più volte all'interno dello stesso insieme di dati, mentre altri potrebbero non essere selezionati. Ciascuno di questi sottoinsiemi viene utilizzato per addestrare un modello di regressione, e le singole previsioni vengono combinate per produrre il risultato finale. L'algoritmo più noto che utilizza il Bagging è il *Random Forest*. Questo algoritmo si

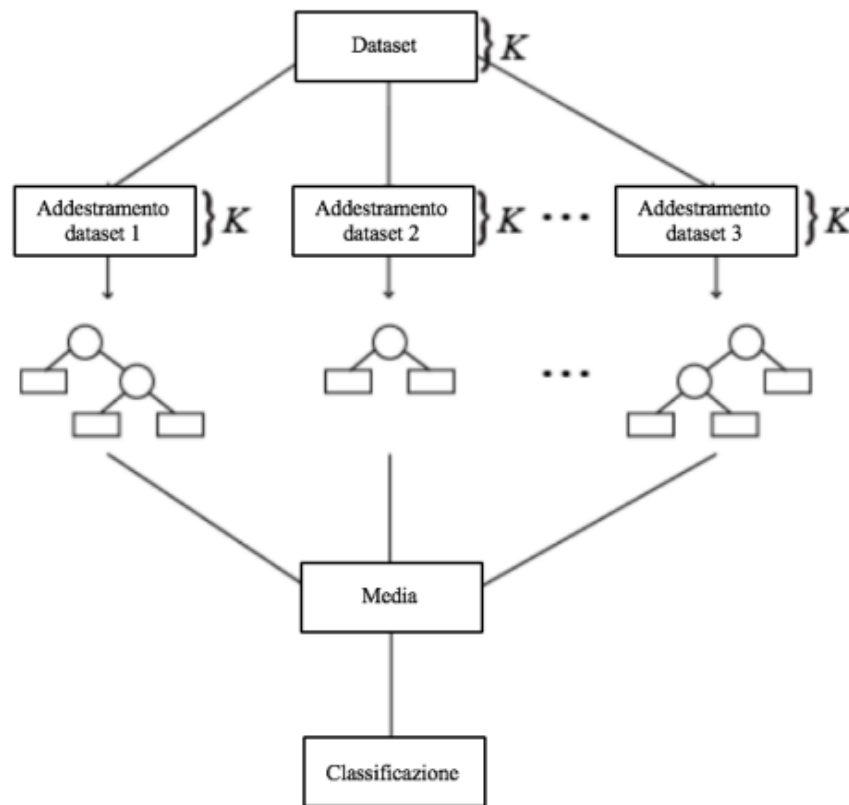


Figura 2.3: Rappresentazione del metodo di bagging.

basa sull'utilizzo di un insieme di alberi decisionali e sulla loro combinazione per ottenere una previsione finale. Gli alberi decisionali ottenuti sono casuali poiché costruiti attraverso due processi principali:

1. *Campionamento casuale dei dati di addestramento*: ciascun albero nella *Random Forest* viene costruito utilizzando un campione casuale di dati. I campioni sono

estratti tramite la tecnica del campionamento con reinserimento (*Bootstrapping*). Tale approccio tende ad aumentare la varianza dei singoli modelli rispetto a un addestramento effettuato su un unico campione di dati. Tuttavia, la varianza complessiva dell'intera foresta risulta ridotta, senza incorrere in un incremento del bias.

2. *Insiemi casuali di caratteristiche usati nella scelta della partizione*: in ogni nodo di un albero decisionale viene considerato un sottoinsieme casuale di tutte le caratteristiche disponibili per la suddivisione dei dati. Generalmente, la dimensione di tale sottoinsieme è pari alla radice quadrata del numero di caratteristiche.

In sintesi, la foresta casuale è costituita da centinaia o migliaia di alberi decisionali, ciascuno addestrato su un sottoinsieme di dati e, in ogni nodo, su una parte delle caratteristiche. Le predizioni finali nei compiti di regressione vengono ottenute mediando le previsioni dei singoli alberi decisionali, in modo che ciascuno contribuisca in egual misura alla decisione finale [43].

2.4 Boosting

Il *Boosting* è una tecnica introdotta con l'obiettivo di ridurre il bias e la varianza, trasformando una serie di regressori "deboli" in un regressore "forte" [44]. Essa rientra nella categoria di *ensemble* sequenziale, poiché, a differenza della *random forest*, costruisce i singoli alberi decisionali in sequenza anziché in parallelo. In questo approccio, ciascun regressore debole viene generato in modo da correggere gli errori commessi dai modelli precedenti, al fine di minimizzare l'errore di addestramento. Tale procedura è definita *apprendimento lento*, poiché albero dopo albero si cerca di migliorare il modello finale che non performa al meglio [43, 45].

L'algoritmo dipende da tre iperparametri principali: il numero di regressori deboli, che bilancia l'accuratezza e l'*overfitting*; il tasso di apprendimento, coefficiente moltiplicativo che controlla il contributo di ciascun albero; e la profondità dei singoli alberi, che determina la complessità [45].

Gli algoritmi di *Boosting* differiscono principalmente per il modo in cui generano e combinano i regressori deboli durante il processo di apprendimento; tra i più noti si distinguono il *Boosting adattivo* e il *Gradient boosting*.

2.4.1 Boosting Adattivo: ADABOOST

La prima implementazione dell'algoritmo di *AdaBoost* è stata attribuita a Yoav Freund e Robert Schapire nel 1997. Questo metodo opera in modo iterativo, individuando i dati classificati erroneamente e modificando i loro pesi per minimizzare gli errori commessi [44]. L'obiettivo principale è la creazione di un regressore forte $H(x)$ come combinazione lineare di regressori semplici $h_t(x)$:

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x) \quad (2.11)$$

dove T rappresenta il numero di classificatori deboli e α_t il coefficiente di importanza associato a h_t , proporzionale alla sua accuratezza [43]. Il boosting adattivo segue i seguenti passaggi:

1. *Inizializzazione dei pesi*: a tutti i campioni (dati di addestramento) viene assegnato lo stesso peso w_i .
2. *Addestramento del regressore debole*: viene addestrato un classificatore debole h_t e vengono identificate le predizioni errate attraverso una metrica di errore.
3. *Aggiornamento dei pesi dei campioni*: i pesi vengono aggiornati in modo da aumentare l'importanza dei dati classificati erroneamente.
4. *Iterazione*: i passaggi 2 e 3 vengono ripetuti fino a quando la metrica utilizzata per l'addestramento non converge, con l'obiettivo di ridurre gli errori di regressione.

In sintesi, l'*ADABOOST* rappresenta la prima implementazione della tecnica di *boosting*. Tuttavia, negli anni si è dimostrata suscettibile al rumore e facilmente soggetta al sovradattamento. [43]

2.4.2 Gradient boosting

Un'evoluzione del *boosting* adattivo è il *Gradient boosting*. A differenza dell'*AdaBoost*, che aggiorna i pesi in base agli errori di classificazione, tale approccio prevede la stima iterativa di nuovi alberi decisionali sulla base dei residui, definiti come la differenza tra il valore reale y dell'output e il valore approssimato \hat{y} . Questo modello è costruito inserendo, a ogni iterazione, un albero decisionale che, unito ai precedenti, definisce un modello più robusto.

Il principio di funzionamento si basa sulla definizione e minimizzazione di una funzione di perdita (*Loss function*) $L(y, F(x))$, attraverso l'applicazione dell'algoritmo di *Gradient descent* (definito in precedenza nel Paragrafo 2.2.1). A ogni iterazione, infatti, il modello aggiorna le proprie previsioni muovendosi nella direzione di discesa del gradiente della funzione di perdita, riducendo progressivamente l'errore residuo. Nel contesto del Gradient Boosting, a ogni passo viene costruito un nuovo albero decisionale che approssima la direzione del gradiente.

Successivamente, si minimizzano i residui mediante l'utilizzo il gradiente della funzione di perdita, come mostrato nella seguente relazione:

$$r_{im} = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \quad (2.12)$$

dove r_{im} rappresenta il residuo per il dato i all'iterazione m . Per problemi di regressione, come quello affrontato in questo progetto di tesi, una delle funzioni di costo più utilizzate è il *Mean Squared Error* (MSE), che rappresenta la differenza quadratica tra il valore predetto e quello osservato:

$$L(y, F(x)) = \frac{1}{2} \sum_{k=1}^n (y_k - \hat{y}_k)^2 \quad (2.13)$$

dove n rappresenta il numero di esempi considerati e $F(x)$ indica i parametri del modello, come le caratteristiche degli alberi decisionali. In questo caso specifico, utilizzando l'Equazione 2.12, il gradiente risulta:

$$r_{im} = y_i - F_{m-1}(x_i) \quad (2.14)$$

2.4.3 XGBoost: Extreme gradient boost

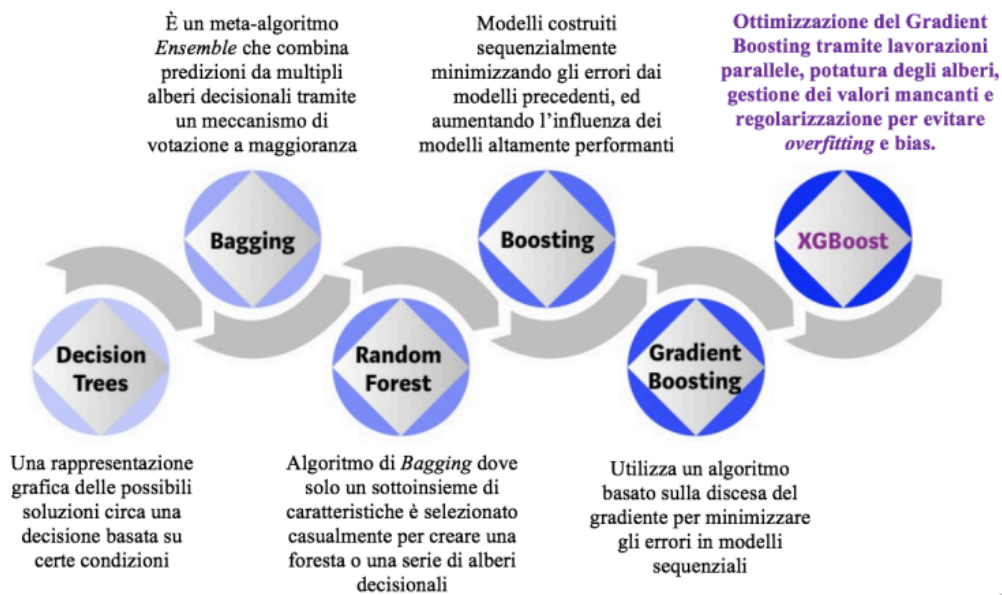


Figura 2.4: Evoluzione degli alberi decisionali.

Come si può notare nella Figura 2.4, l'*extreme gradient boosting* (XGBoost) è l'ultima tecnologia sviluppata. Si tratta di un algoritmo di Machine learning basato su alberi decisionali che può essere considerato un'evoluzione del metodo di gradient boosting. Proposto da Tianqi Chen e Carlos Guestrin nel 2016, consente di ottenere migliori prestazioni in termini di risorse computazionali e di performance [43]. Il principio di base rimane invariato rispetto al gradient Boosting tradizionale, ossia il miglioramento progressivo delle prestazioni del modello mediante l'aggiunta sequenziale di modelli deboli (*weak learners*) [46]. Questo approccio introduce diverse ottimizzazioni a livello di sistema e potenziamenti di natura algoritmica.

Ottimizzazioni del sistema

- *Parallelizzazione:* l'algoritmo supporta la costruzione sequenziale degli alberi mediante un'implementazione parallela. Ciò è reso possibile dalla natura intercambiabile dei due cicli (*loop*) utilizzati nella generazione degli alberi decisionali: un ciclo esterno, che enumera i nodi foglia, e un ciclo interno, che elabora le caratteristiche. Nel metodo tradizionale, questa struttura limita la parallelizzazione,

in quanto il ciclo interno deve essere completato prima di poter eseguire quello esterno. *XGBoost* ottimizza il processo invertendo l'ordine dei cicli: effettua una scansione globale di tutti i dati di addestramento e li ordina utilizzando *thread* paralleli, riducendo i tempi di esecuzione.

- *Potatura (pruning)*: *XGBoost* utilizza l'approccio definito *depth-first pruning*, ossia gli alberi vengono inizialmente costruiti fino alla profondità massima (*max-depth*) e successivamente potati a ritroso. Nell'algoritmo tradizionale, la potatura viene effettuata durante il processo di generazione dell'albero, aumentando notevolmente il costo computazionale.
- *Ottimizzazione Hardware*: sfrutta in modo efficiente le risorse hardware grazie alla consapevolezza della cache. Ulteriori ottimizzazioni, come il calcolo *out-of-core*, migliorano la gestione dello spazio su disco, consentendo l'elaborazione di grandi insiemi di dati.

Potenziamenti algoritmici

- *Regolarizzazione*: l'algoritmo penalizza la complessità dei modelli per ridurre il rischio di *overfitting* mediante tecniche di regolarizzazione *Lasso* (L1) e *Ridge* (L2). La regolarizzazione L1 introduce un termine di penalità nella funzione di perdita con l'obiettivo di eliminare le caratteristiche multicollineari (fortemente correlate tra loro) e di favorire la selezione delle variabili rilevanti all'interno del modello. La regolarizzazione L2, invece, aggiunge anch'essa un termine di penalità, ma non comporta la selezione delle caratteristiche: riduce progressivamente i pesi associati alle variabili, senza annullarli completamente [47].
- *Gestione dei valori mancanti*: il modello è in grado di apprendere la presenza di dati mancanti e di gestirla in modo efficace.
- *Weighted Quantile Sketch*: si utilizza questo algoritmo per identificare il punto migliore di *split* durante la costruzione dei dati.

Esistono diversi iperparametri che controllano il modello *XGBoost*. Per massimizzare l'accuratezza di *XGBoost*, questi iperparametri devono essere ottimizzati attraverso un processo di ottimizzazione dedicato. Questa attività comprende generalmente un processo noto come *cross-validation*. La *cross-validation* suddivide casualmente il set di addestramento per produrre un piccolo set di convalida, utilizzato per misurare le prestazioni. In questo modo, è possibile quantificare l'effetto della modifica di un iperparametro e selezionare il valore ottimale [48].

I principali iperparametri comunemente utilizzati sono:

- *Learning rate* (tasso di apprendimento): determina la velocità con cui l'algoritmo di boosting apprende ad ogni iterazione. Un valore basso corrisponde a un apprendimento più lento e un rischio di sovradattamento minore. Viceversa, un valore più elevato accelera l'apprendimento, ma può portare a *overfitting*.

- *Numero di stimatori*: specifica il numero di alberi decisionali da costruire nel modello. Questo iperparametro influenza la complessità del modello, il tempo di addestramento e la capacità di generalizzazione.
- *Gamma*: rappresenta la soglia minima di riduzione della funzione di perdita per effettuare un'ulteriore suddivisione su un nodo foglia. Un valore più basso comporta l'arresto anticipato dell'addestramento, con il rischio che l'algoritmo non individui la soluzione ottimale; al contrario, un valore più elevato consente di proseguire l'apprendimento più a lungo, aumentando la probabilità di sovradattamento.
- *Max depth*: indica la profondità massima raggiungibile da un albero decisionale. La profondità di un albero corrisponde al numero di suddivisioni comprese tra il nodo radice e il nodo foglia [49].

In conclusione, nel presente progetto di tesi è stato applicato il metodo *XGBoost*. Tale algoritmo è stato selezionato per la sua capacità di garantire un'elevata efficienza computazionale e per l'inclusione di tecniche di regolarizzazione, al fine di ridurre la probabilità di *overfitting*.

Capitolo 3

Materiali e Metodi

I metodi descritti di seguito sono stati implementati mediante i linguaggi di programmazione Matlab e Python, quest'ultimo in ambiente Visual Studio Code (VSC) e Google Colaboratory (Colab). Per Matlab è stato utilizzato il sistema operativo Windows, mentre per Python si è lavorato esclusivamente con il sistema operativo Linux.

3.1 Dataset

3.1.1 Immagini sperimentali

Per le analisi preliminari, sono state impiegate le immagini sperimentali acquisite su fantocci (*phantoms*), ossia modelli fisici progettati per riprodurre le proprietà acustiche e strutturali dei tessuti biologici. Tali modelli presentano un'elevata reperibilità e riproducibilità, inoltre, essendo privi di variabilità biologica, permettono di operare in un ambiente controllato. Sono stati utilizzati due diversi tipi di fantocci, caratterizzati da due velocità del suono (SoS) differenti:

1. *3B Scientific Breast Phantom*: modello solido commerciale realizzato in silicone 3B SKINlike, con dimensioni pari a 25,4 x 17,8 x 17,8 cm e massa di circa 0,8 Kg. Il fantoccio riproduce realisticamente la morfologia e la superficie del seno umano ed è dotato di una regione benigna (come mostrato nella Figura 3.1) caratterizzata da diverse proprietà acustiche. All'interno del fantoccio, la velocità del suono è costante e pari a 1540 m/s [50, 51].
2. *Fantoccio ad emulsione acqua-in-olio a base di PDMS/Agarosio*: modello solido di laboratorio costituito da un'emulsione acqua-in-olio (W/O) formata da polimetilsilossano (PMDS) come fase continua e agarosio come fase dispersa [52, 53]. Tale miscela è stata formulata per riprodurre le proprietà ottiche e meccaniche dei tessuti biologici molli. Nello specifico, il PDMS è un elastomero trasparente, meccanicamente stabile e facilmente lavorabile, mentre la fase dispersa è costituita da un idrogel di agarosio (concentrazione 1-2% p/p), che permette di modulare la

diffusione e l'assorbimento ottico [54]. Per ottimizzare le proprietà ottiche, sono stati aggiunti i seguenti componenti:

- Nanoparticelle di TiO_2 : utilizzate per controllare lo scattering ottico;
- Assorbitori ottici: come l'inchiostro di seppia o di china, utilizzati per simulare l'assorbimento della melanina e dell'emoglobina presenti nei tessuti biologici.
- *Nanorods* d'oro: impiegati per ottenere diversi livelli di ossigenazione [54].

Il fantoccio è stato fabbricato con la tecnica dell'*injection molding*. Inoltre, la geometria interna include canali microfluidici (dell'ordine di alcuni millimetri) che simulano la presenza di vasi sanguigni; la struttura finale presenta uno spessore totale di 10 mm con una velocità del suono pari a 1250 m/s [52], compatibile con quella dei tessuti molli.



Figura 3.1: Rappresentazione dei fantocci utilizzati. A sinistra è mostrato il fantoccio 3B Scientific Breast, mentre a destra è visibile il fantoccio in PDMS con emulsione.

In Figura 3.2 sono riportate le immagini ecografiche acquisite sui fantocci precedentemente descritti, che fanno parte del nostro dataset di riferimento. Le due immagini mostrano la ricostruzione in dB dopo aver applicato tutta la pipeline di elaborazione, descritta nel Paragrafo 3.3.2, senza eseguire il processo finale di DSC. Le due immagini costituiscono mappe di intensità acustica espresse in dB: l'asse orizzontale (x) indica la posizione laterale in millimetri, mentre l'asse verticale (z) rappresenta la profondità in millimetri. Le ricostruzioni sono state ottenute mediante il processo di *beamforming*, utilizzando due velocità del suono diverse: 1540 m/s per il primo fantoccio (immagine a sinistra) e 1250 m/s per il secondo fantoccio (immagine a destra). Nella Tabella 3.1 sono riportati i principali parametri geometrici e di visualizzazione. In particolare, la dimensione laterale dei pixel risulta costante per entrambe le immagini. La dimensione dei pixel lungo la profondità dipende dalla c e dalla frequenza di campionamento f_s della sonda:

$$\Delta_z = \frac{c}{2f_s} \quad (3.1)$$

	Parametro	Fantoccio 1	Fantoccio 2	Unità
Dimensioni	Numero di colonne (laterale)	128	128	pixel
	Numero di righe (assiale)	1920	2304	pixel
	Dimensione pixel Δ_x (laterale)	0,3	0,3	mm
	Dimensione pixel Δ_z (assiale)	Dipendente da c, f_s	Dipendente da c, f_s	mm
Parametri di BF	Velocità del suono (c)	1540	1250	m/s
	Frequenza di campionamento (f_s)	31,25	31,25	MHz
Visualizzazione	Dinamica ($range$)	-50 a 0	-50 a 0	dB
	Mappatura colori	Nero (-50 dB) a Bianco (0 dB)		—

Tabella 3.1: Parametri di ricostruzione e caratteristiche delle immagini acquisite in-vitro.

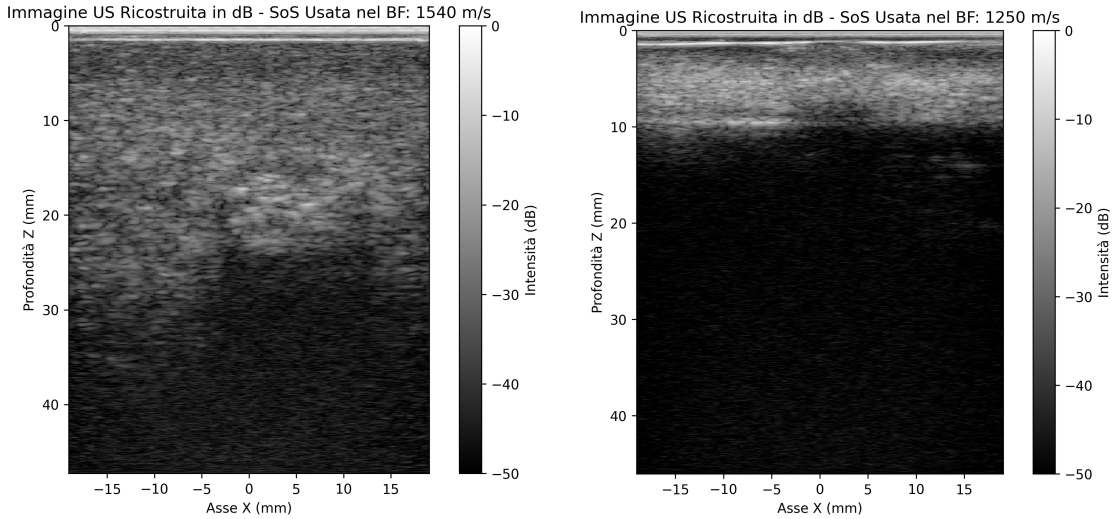


Figura 3.2: Rappresentazione Immagini Ecografiche Fantocci. A sinistra fantoccio 3B Scientific Breast Phantom, a destra il fantoccio in PDMS/Agarosio.

3.1.2 Simulazione delle immagini

Per l'addestramento del regressore *XGBoost*, è stato creato un dataset di immagini simulate. La simulazione è stata realizzata mediante il toolbox *k-wave* in *Matlab R2023a*, come descritto nel Paragrafo 3.2.2. Il dataset è composto da 140 immagini, ciascuna caratterizzata da una diversa distribuzione della velocità del suono (SoS), in modo da rappresentare differenti condizioni fisiologiche. In particolare, il dataset è suddiviso come segue:

- *51 immagini a velocità costante*: ogni immagine presenta un valore uniforme di SoS su tutta l'area; i 51 valori sono compresi tra 1200 e 1700 m/s, con un incremento di 10 m/s.
- *45 immagini con distribuzione orizzontale della velocità*: considerando l'intervallo 1200-1700 m/s (passo 10), sono state estratte 45 coppie casuali di valori di SoS. Ogni immagine è formata da due strati orizzontali, ai quali sono assegnati i valori di SoS corrispondenti alla coppia selezionata.
- *44 immagini con distribuzione verticale della velocità*: analogamente al caso precedente, sono state considerate 44 coppie casuali di valori di SoS nello stesso intervallo, generando immagini costituite da due strati verticali.

In Figura 3.3 sono mostrati due esempi di immagini ecografiche simulate, insieme alle corrispondenti mappe di velocità del suono (SoS), facenti parte del dataset di addestramento. Le immagini sono state acquisite mediante la trasmissione a onda piana e rappresentano la ricostruzione in dB ottenuta al termine dell'intera pipeline di elaborazione descritta nel Paragrafo 3.3.2, ad eccezione della fase finale di DSC. L'asse orizzontale (x) indica la posizione laterale in millimetri, mentre l'asse verticale (z) rappresenta la profondità in millimetri. Le ricostruzioni sono state ottenute mediante il processo di *DAS beamforming*, implementato per la modalità a onda piana e basato sull'utilizzo di mappe di velocità del suono eterogenee, che consentono di modellare in modo più realistico la propagazione acustica all'interno del mezzo. È importante sottolineare che queste ricostruzioni e visualizzazioni sono state generate esclusivamente a scopo di visualizzazione e non sono state impiegate nelle analisi quantitative o nei processi di addestramento sviluppati nel corso del progetto di tesi. I principali parametri geometrici e di visualizzazione sono riportati nella Tabella 3.2.

	Parametro	Fantoccio 1	Fantoccio 2	Unità
Dimensioni	Numero di colonne (laterale)	128	128	pixel
	Numero di righe (assiale)	1166	1302	pixel
	Dimensione pixel Δ_x (laterale)	0,3	0,3	mm
	Dimensione pixel Δ_z (assiale)	Dipendente da c, f_s	Dipendente da c, f_s	mm
Parametri di BF	Velocità del suono (c)	1540	1250	m/s
	Frequenza di campionamento (f_s)	31,25	31,25	MHz
Visualizzazione	Dinamica (<i>range</i>)	-25 a 0	-25 a 0	dB
	Mappatura colori	Nero (-25 dB) a Bianco (0 dB)		—

Tabella 3.2: Parametri di ricostruzione e caratteristiche delle immagini simulate.

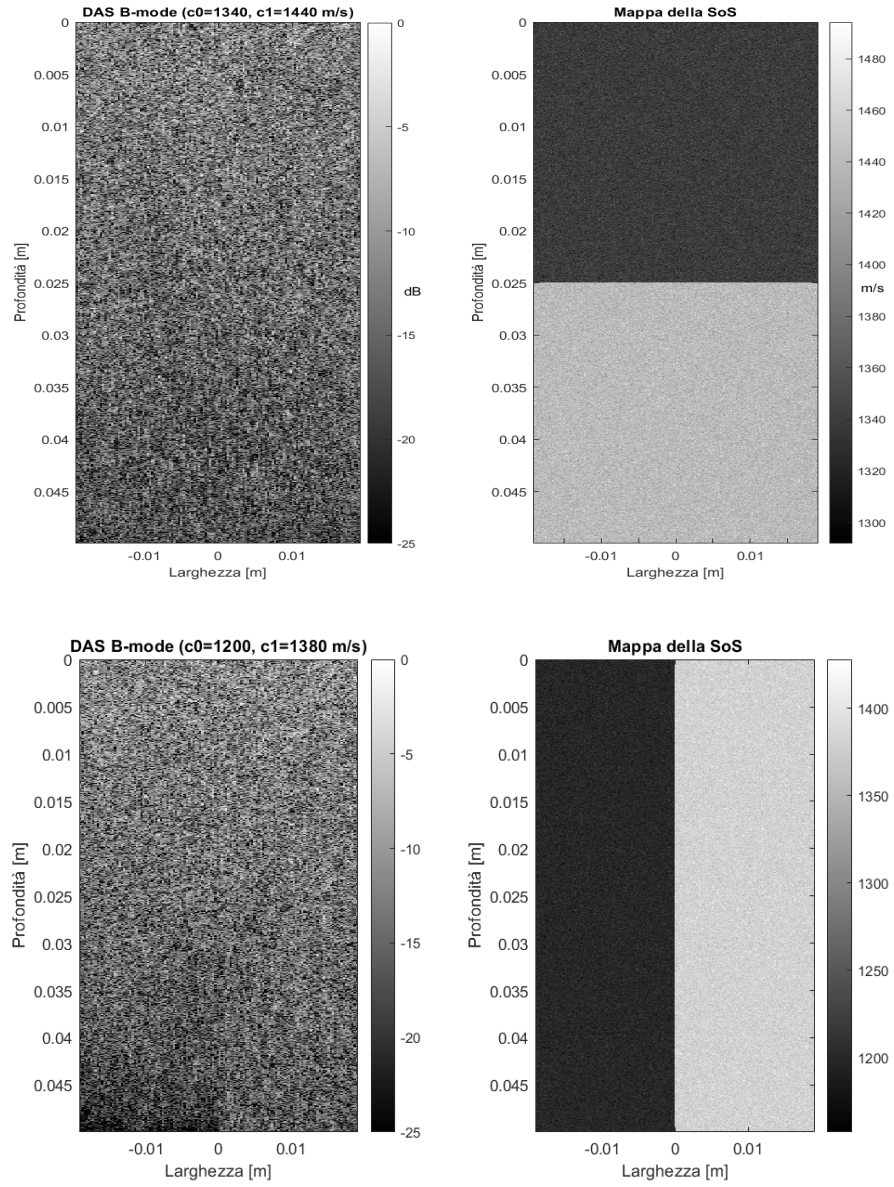


Figura 3.3: Rappresentazione di immagini ecografiche simulate con *k-Wave* e delle corrispondenti mappe della velocità del suono. In alto è mostrata un'immagine simulata con due strati orizzontali aventi SoS pari a 1340 e 1440 m/s, mentre in basso è riportata un'immagine simulata con due strati verticali caratterizzati da SoS pari a 1200 e 1380 m/s.

3.2 Strumentazione

3.2.1 Ecografo

Nel contesto del presente progetto di tesi, i dati in radiofrequenza (RF) sono stati acquisiti mediante il sistema *Verasonics Vantage 256 UTA High Frequency* (*Verasonics Inc., Kirkland, WA, USA*) e un trasduttore ad array lineare *Verasonics L11-5v*, caratterizzato da una frequenza centrale di $7,6\text{ MHz}$ e da una banda operativa compresa tra $4,68$ e $10,52\text{ MHz}$. La frequenza di campionamento del trasduttore è pari a $31,25\text{ MHz}$, garantendo il rispetto del teorema di Nyquist. Le caratteristiche tecniche della sonda sono riportate nella Tabella 3.3. La sonda è costituita da una matrice di 128 elementi piezoelettrici, disposti con un *pitch* di $0,3\text{ mm}$ su una superficie piana. Un adattatore *UTA 260-D*, che collega la sonda al sistema *Vantage 256*, è dotato di 260 pin; ciascun connettore fornisce segnali per 128 canali (come illustrato in Figura 3.4) [55]. La sonda impiegata, mostrata in Figura 3.5, presenta un'impugnatura ergonomi-



Figura 3.4: Adattatore *UTA 260-D*.

ca bombata e dimensioni pari a 12 cm in altezza, 1 cm in profondità e circa 5 cm in larghezza. Per ciascuna acquisizione, la sonda è stata posizionata perpendicolarmente all'oggetto in esame.

Caratteristica tecnica	Valore
Numero di elementi	128
Larghezza elemento (mm)	0,27
Passo (mm)	0,3
Messa a fuoco in elevazione (mm)	18
Apertura in elevazione (mm)	5
Sensibilità (dB)	-52 ± 3
Compatibilità con sistemi	Tutte le configurazioni dei sistemi <i>Vantage</i>

Tabella 3.3: Caratteristiche tecniche della sonda ecografica.

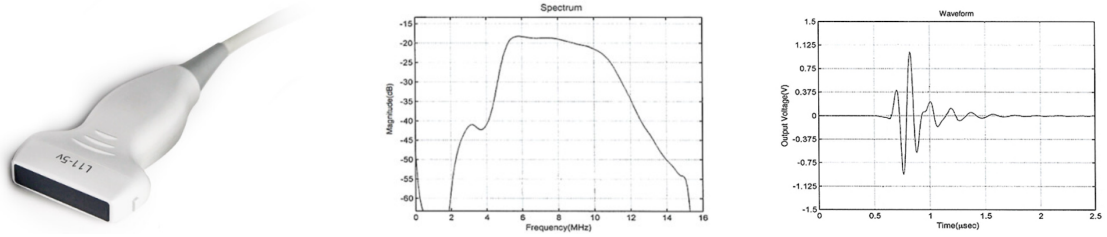


Figura 3.5: A destra la sonda Verasonics L11-5v. Al centro la risposta in frequenza (spettro) della sonda. A sinistra la risposta in tensione nel dominio del tempo (Output Voltage vs. Time) del trasduttore, che caratterizza l'impulso acustico generato.

3.2.2 Simulatore k-Wave

Come precedentemente affermato, per l'implementazione e l'addestramento del classificatore sono state simulate immagini ecografiche mediante il toolbox *k-wave* in *Matlab R2023a*. Il pacchetto include tre funzioni di simulazione che implementano il modello alle derivate del primo ordine: *kspaceFirstOrder1D*, *kspaceFirstOrder2D*, *kspaceFirstOrder3D*, corrispondenti rispettivamente alla propagazione delle onde in una, due e tre dimensioni. Le funzioni di simulazione vengono richiamate mediante quattro strutture di input: *kgrid*, *medium*, *source* e *sensor*. Le quattro strutture definiscono, rispettivamente, le proprietà della griglia di calcolo, le caratteristiche fisiche del mezzo, le proprietà e la posizione delle sorgenti acustiche, nonché le proprietà e le coordinate dei punti sensori impiegati per registrare l'evoluzione dei campi di pressione e della velocità delle particelle nel tempo. Nello specifico:

- *kgrid*: definisce le proprietà della griglia computazionale. Essa determina la discretizzazione del mezzo continuo in una *mesh* uniformemente campionata, costituita da punti di griglia (*grid points*). Tali punti rappresentano le posizioni discrete nello spazio in cui vengono risolte le equazioni di propagazione che governano il sistema.
- *Medium*: specifica le proprietà materiali del mezzo in ciascun punto della griglia. Tipicamente, vengono definiti la velocità del suono, la densità e, se presenti, i parametri di non linearità.
- *Source*: definisce le proprietà e la posizione di qualunque sorgente acustica. Questa struttura specifica la posizione spaziale degli elementi attivi, corrispondenti ai punti della griglia da cui viene emesso il segnale, nonché il profilo temporale e l'ampiezza dell'impulso acustico. Nel simulatore è possibile definire tre differenti tipologie di sorgenti. La prima corrisponde a una distribuzione di pressione iniziale. Il secondo tipo di sorgente implementabile in *k-Wave* è una sorgente di pressione variabile nel tempo. La terza tipologia definibile è una sorgente di velocità delle particelle variabile nel tempo.

- *Sensor*: definisce le proprietà e la posizione dei punti di rilevamento impiegati per registrare il campo acustico a ogni passo temporale della simulazione. Specifica inoltre la disposizione spaziale degli elementi del trasduttore che agiscono come ricevitori attivi, determinando le grandezze fisiche da registrare.

Le simulazioni con trasduttori ecografici possono essere eseguite definendo i parametri *source* e *sensor*. Tuttavia, l'assegnazione manuale dei punti della griglia corrispondenti a ciascun elemento fisico può risultare complessa. Il pacchetto *k-Wave* fornisce la classe *kWaveTransducer*, la quale, mediante la funzione *makeTransducer*, gestisce automaticamente i punti della griglia associati a ciascun elemento del trasduttore, consentendo di sostituire i due parametri in presenza di trasduttori più complessi. In particolare, la funzione *kWaveTransducer* accetta in ingresso diversi parametri, tra cui: la posizione del trasduttore nella griglia, il numero di elementi fisici, la dimensione e la distanza tra gli elementi (*pitch*), la distanza di focalizzazione, la velocità del suono, la focalizzazione in elevazione, il segnale di ingresso e i parametri di apodizzazione in ricezione e trasmissione [12].

Parametri impiegati

Nel presente progetto di tesi, le simulazioni ecografiche sono state realizzate mediante la funzione di simulazione *kspaceFirstOrder2D*, che consente di modellare numericamente la propagazione dell'onda acustica in due dimensioni. Sono stati impostati i seguenti parametri fisici e numerici, con l'obiettivo di riprodurre diverse condizioni acustiche del mezzo e generare dati realistici per l'addestramento del classificatore:

- *kgrid*: la griglia computazionale è stata definita con un passo spaziale, ossia la distanza fisica tra due punti adiacenti della griglia, determinato in funzione della frequenza di campionamento $f_s = 31,25 \text{ MHz}$ e della velocità del suono nel mezzo c_0 . In particolare, è stata impostata la profondità massima $z_{max} = 50 \text{ mm}$, utilizzata per determinare il numero di punti della griglia lungo la direzione della profondità $N_z = z_{max}/dz$, dove dz è il corrispondente passo spaziale. Lungo la direzione laterale, il numero di punti della griglia N_y è stato fissato al doppio del numero di elementi del traduttore (128), con un passo spaziale dy pari al *pitch* degli elementi. Il numero di passi temporali è stato fissato a $N_t = 2N_z$, con un intervallo temporale $dt = 1/f_s$.
- *Medium*: il mezzo di propagazione è stato rappresentato come un dominio bidimensionale costituito da una o due regioni sovrapposte, caratterizzate da differenti valori della SoS, a seconda della configurazione di simulazione: singola o a doppio strato, con discontinuità orizzontali o verticali. La densità del mezzo è stata mantenuta costante a $\rho_0 = 1000 \text{ kg/m}^3$, mentre sono state introdotte variazioni casuali con deviazione standard $\sigma = 0.008$ nella mappa della velocità del suono, con l'obiettivo di simulare l'eterogeneità acustica del mezzo.
- *Source*: la sorgente è stata modellata in modo analogo a quella sperimentale, come un array lineare composto da 128 elementi, con *pitch* = 0,3 mm. Il segnale

emesso da ciascun elemento è stato definito come un *tone burst* con 4 cicli, con una frequenza centrale $f_0 = 7,5 \text{ MHz}$, ampiezza 10^6 e condizioni di pressione al contorno di tipo *Dirichlet*. L'angolo di steering è stato impostato a 0° per simulare un fascio perpendicolare al mezzo di propagazione.

- *Sensor*: la maschera dei sensori è stata definita coincidente con quella della sorgente. Durante ogni simulazione, sono stati memorizzati i campi di pressione istantanea, massima e finale, utili per le fasi successive.
- *Elaborazione e salvataggio dei risultati*: i segnali RF acquisiti sono stati convoluti con il segnale di trasmissione e successivamente normalizzati rispetto all'ampiezza massima per ciascuna riga.

3.3 Ricostruzione dell'immagine

Una volta acquisiti i 128 segnali relativi a ciascuna immagine del dataset (sia per le immagini simulate sia per quelle acquisite su fantocci in vitro), si è proceduto alla ricostruzione di immagini bidimensionali (B-mode), basata sulla teoria di formazione delle immagini ecografiche cliniche.

3.3.1 Beamforming

Le ricostruzioni sono state ottenute mediante il processo di *DAS beamforming*, implementato per la modalità a onda piana. In particolare, per ogni pixel dell'immagine, la velocità utilizzata per calcolare il ritardo dipende dalla posizione e viene quindi adattata localmente nel mezzo. Nel seguente progetto di tesi è stato implementato il *DAS beamforming* (in ambiente *Matlab 2023a*), il quale, pur presentando come limiti la bassa risoluzione dell'immagine e una ridotta capacità di reiezione delle interferenze provenienti da direzioni non scansionate, è stato scelto poiché rappresenta un metodo lineare e computazionalmente efficiente.

In primo luogo, l'algoritmo implementato calcola separatamente i ritardi di trasmissione e di ricezione; la loro somma determina il tempo complessivo tale per cui ciascun segnale ricevuto da un elemento contribuisca correttamente al pixel finale. Successivamente, è stata applicata l'apodizzazione, finalizzata alla riduzione degli artefatti (lobi) laterali e al miglioramento della qualità dell'immagine. Nel presente studio è stata utilizzata la *finestra di Tukey* con un coefficiente di rastrematura (*tapering coefficient*) $\alpha = 0,75$. In seguito, mediante la funzione di interpolazione *spline*, i segnali RF dei singoli elementi sono stati elaborati, allineando ciascun campione in base ai ritardi calcolati. Infine, la moltiplicazione tra la somma dei segnali interpolati e la funzione di apodizzazione genera la matrice beamformata finale, nella quale ciascun pixel rappresenta l'ampiezza del segnale focalizzato nella corrispondente posizione spaziale.

Il risultato finale è pertanto una matrice bidimensionale (2D) di dati RF, pronta per le successive fasi di elaborazione e visualizzazione.

3.3.2 Post processing

Come indicato nel Paragrafo 1.3, è stata sviluppata una *pipeline* di elaborazione delle immagini utilizzando il linguaggio di programmazione *Python* nell'ambiente *Google Colaboratory*, con l'obiettivo di ottenere immagini visualizzabili con valori dei pixel compresi tra 0 e 255.

Rispetto alla *pipeline* tradizionale indicata in Figura 1.9, nel presente studio è stata implementata una pipeline di elaborazione del segnale semplificata, finalizzata alla conversione dei dati ecografici grezzi post-beamforming in un'immagine visualizzabile. Il processo inizia con il caricamento dei dati grezzi, sui quali viene calcolata la funzione di involuppo (*Envelope*) mediante la trasformata di *Hilbert*. L'output di questa fase consiste in una matrice che rappresenta l'ampiezza del segnale retrodiffuso. Successivamente, è stata applicata la compressione logaritmica (o *Log Compression*), secondo l'Equazione 1.21. In seguito, è stata applicata la *digital scan conversion* (DSC), che comprende quattro passaggi per la generazione dell'immagine finale in scala di grigio. Prima dell'interpolazione necessaria a colmare i pixel mancanti, i dati di ampiezza vengono normalizzati in modo da essere rappresentati in un intervallo di valori compresi tra 0 e 1. Questa operazione viene effettuata mediante le seguenti fasi:

1. *Offset*: si esegue una traslazione matematica, pari a *vrangle* (-50 dB), al fine di rendere positivi i valori negativi ottenuti dalla *log-compression*.

$$B_b = B_b + |vrangle| \quad (3.2)$$

2. *Normalizzazione*: i valori negativi rimanenti vengono impostati pari a 0. Successivamente, i dati vengono normalizzati rispetto al valore massimo, così che tutti i valori risultino scalati nell'intervallo [0,1].
3. *Interpolazione*: rappresenta la fase centrale della DSC. Tale processo è fondamentale per stimare i valori dei nuovi pixel e per colmare i punti mancanti tra i dati acquisiti. In particolare, la funzione dell'interpolazione è quella di ottenere una griglia uniforme di pixel. Per definire la risoluzione spaziale dell'immagine, vengono introdotti i parametri di calibrazione CF (*Calibration Factor*), che specificano la dimensione del singolo pixel lungo gli assi *x* e *z*. Tali parametri CF definiscono la risoluzione spaziale desiderata dei pixel finali (in *mm*), consentendo di calcolare il numero di pixel lungo ciascuna direzione e di costruire una griglia uniforme per l'interpolazione. L'interpolazione è realizzata mediante la funzione *RectBivariateSpline* della libreria *Scipy*, che implementa un'interpolazione *spline* bicubica. Questo approccio, a differenza dell'interpolazione bilineare basata su polinomi di primo grado, impiega polinomi di terzo grado, garantendo la continuità della superficie di luminosità generata non solo nel valore del segnale, ma anche nelle prime e seconde derivate.
4. *Normalizzazione 0-255*: l'immagine interpolata viene normalizzata tramite la funzione *cv2.normalize* della libreria *OpenCV*, in modo da essere convertita nel formato visualizzabile *uint8* (8 bit), con valori compresi nell'intervallo tra 0 e 255.

Completato il processo DSC, l'immagine di intensità viene visualizzata nella fase di *display* mediante la libreria *Matplotlib* e costituisce il dato di ingresso per le successive fasi di analisi quantitativa descritte nel paragrafo seguente.

3.4 Metodi di analisi

L'analisi delle immagini mediche digitali si basa sull'applicazione di metodologie specifiche volte all'estrazione di informazioni significative dai dati medici. Tali analisi risultano essenziali per la quantificazione delle proprietà dell'immagine non identificabili direttamente mediante la sola visualizzazione. La scelta del metodo più appropriato dipende dalla tipologia dei dati a disposizione e dalle proprietà che si intendono analizzare. Nel seguente paragrafo vengono trattate due metodologie distinte: la correlazione lineare e logaritmica, impiegate per valutare la relazione tra le variabili, e l'analisi di tessitura (*Texture analysis*), che permette di ottenere informazioni riguardanti la posizione e l'intensità dei livelli di grigio dei pixel [57].

3.4.1 Analisi di regressione lineare e logaritmica

L'analisi di regressione rappresenta un importante metodo statistico per l'analisi dei dati medici, in quanto consente l'identificazione e la caratterizzazione delle relazioni tra due o più variabili [58]. Questo tipo di analisi consente di svolgere le seguenti operazioni:

- *Descrizione.* Le relazioni tra le variabili dipendenti e indipendenti possono essere descritte mediante modelli statistici.
- *Stima.* I valori delle variabili dipendenti possono essere stimati sulla base dei valori osservati delle variabili indipendenti.

Esistono diversi tipi di regressione, tra le più comuni si menzionano la regressione lineare, la regressione logistica e la regressione di Poisson. Nel presente progetto sono state implementate due tipologie di regressione: la regressione lineare e la regressione lineare-logaritmica, entrambe realizzate mediante l'utilizzo di *Python*, del suo pacchetto *Scikit-Learn* e della libreria *sklearn.linear.model.LinearRegression*. Nel contesto del presente studio, la regressione lineare è stata implementata per analizzare la relazione tra una variabile dipendente (Y) e una indipendente (X). Nonostante sia possibile includere più variabili indipendenti, è stata adottata la formulazione con una sola variabile (Regressione lineare univariata). La variabile Y deve essere continua, mentre la variabile X può essere continua, binaria o categorica. Il modello di regressione lineare descrive la relazione tra le due variabili tramite una retta definita dall'equazione:

$$Y = a + b * X \quad (3.3)$$

Dove *a* rappresenta l'intercetta della retta con l'asse y mentre *b* il suo coefficiente angolare. Entrambi i parametri (*a* e *b*) vengono stimati a partire dai valori di X e Y

mediante metodi statistici. La pendenza b , detta coefficiente di correlazione, rappresenta la variazione della variabile dipendente per unità di variazione della variabile indipendente nel caso di variabili continue. A complemento di b , per quantificare la forza e la direzione della relazione tra due variabili continue si utilizza il *Coefficiente di correlazione di Pearson* (R) oppure il suo quadrato R^2 definito coefficiente di determinazione, che misura quanto accuratamente il modello di regressione descrive i dati osservati [58].

$$R = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (3.4)$$

Il coefficiente R si calcola mediante l'Equazione 3.4, dove:

- n : numero totale di osservazioni considerate
- X_i e Y_i : singoli valori osservati delle variabili X e Y
- \bar{X} e \bar{Y} : medie aritmetiche delle variabili X e Y
- $\text{COV}(X, Y)$: covarianza, misura il grado di variazione congiunta tra X e Y .
- σ_X e σ_Y : deviazioni standard delle variabili X e Y .

Il valore numerico del coefficiente R è compreso tra -1 e $+1$:

- $R = \pm 1$: indica una relazione perfettamente lineare. Più R si avvicina ai valori estremi dell'intervallo, tanto più la relazione tra le variabili risulta forte.
- $R = 0$: assenza di relazione lineare tra le variabili.
- $R < 0$: indica una correlazione negativa; i valori elevati di una variabile si verificano con i valori bassi dell'altra variabile.
- $R > 0$: rappresenta una correlazione positiva; i valori elevati di una variabile tendono a verificarsi insieme ai valori elevati dell'altra variabile.

C'è una relazione diretta tra il coefficiente di correlazione (R) e il coefficiente angolare (b) della retta:

$$b = R \left(\frac{\sigma_Y}{\sigma_X} \right) \quad (3.5)$$

Oltre alla correlazione lineare, è possibile considerare la correlazione lineare-logaritmica tra le variabili X e Y , rappresentata dalla seguente relazione matematica:

$$Y = a + b * \ln(X) \quad (3.6)$$

Le considerazioni precedenti relative ai coefficienti R e b rimangono le medesime anche per il seguente modello, nel quale la variabile indipendente X viene sostituita con il suo logaritmo naturale $\ln(X)$.

3.4.2 Analisi radiomica di tessitura

La radiomica si concentra sul miglioramento dell'analisi delle immagini, utilizzando un'estrazione automatizzata ad alto rendimento di grandi quantità (oltre 200) di caratteristiche quantitative derivate da immagini mediche, e rappresenta una delle più recenti innovazioni nell'analisi dell'imaging medico [57]. Tra i principali metodi di estrazione di caratteristiche quantitative, nel presente studio è stata utilizzata l'analisi di tessitura (*texture analysis*). La *texture analysis* è un metodo d'indagine di relativamente recente introduzione, applicato alle immagini digitali per l'estrazione di caratteristiche e proprietà (*features*) della loro *texture*. Non esiste una definizione univoca di *texture* (tessitura), ma può essere considerata come un indicatore che descrive le proprietà della "trama" dell'immagine, fornendo informazioni sulla relazione spaziale tra i pixel dell'immagine piuttosto che sui loro singoli valori di intensità. In altre parole, la *texture* non dipende solamente dal valore del singolo pixel, ma dalla disposizione e dalla ripetizione di schemi di pixel che conferiscono a una regione dell'immagine uno specifico aspetto. L'analisi di tessitura delle immagini mediche fornisce informazioni quantitative utili alla caratterizzazione dei tessuti e alla descrizione della struttura interna degli organi, al fine di supportare la discriminazione di possibili alterazioni patologiche [59]. Esistono quattro approcci di analisi:

- *Statistico*: rappresenta una misura numerica della distribuzione dell'intensità dei pixel dell'immagine, ottenuta tramite il calcolo di parametri statistici chiamati caratteristiche (o feature) di tessitura.
- *Strutturale*: interpreta la texture come una composizione di primitive, dette *texels*, ossia elementi o pattern geometrici ricorrenti. L'analisi si fonda sul riconoscimento di tali primitive e sulle regole che ne governano la disposizione spaziale, responsabili della formazione del pattern complessivo dell'immagine.
- *Spettrale*: si basa sulla trasformata di *Fourier* per analizzare la texture dell'immagine nel dominio della frequenza. Questo approccio è indicato per descrivere la direzionalità e la periodicità di specifici pattern all'interno dell'immagine.
- *Model Based* (basati sui modelli): caratterizza la tessitura dell'immagine mediante modelli stocastici o frattali, dai quali si ricavano parametri utili all'analisi. Tuttavia, tali approcci presentano limiti legati all'elevato costo computazionale e alla ridotta capacità di descrizione locale della struttura dell'immagine.

Nel corso di questo studio, è stato adottato l'approccio statistico, basato sulle *feature* quantitative di tessitura, ovvero parametri matematici estratti dalla distribuzione dei pixel, considerando sia la loro intensità sia la loro posizione spaziale all'interno dell'immagine. Tali caratteristiche si suddividono tipicamente in parametri di primo ordine, che considerano i valori dei singoli pixel, e di ordine superiore, che analizzano le relazioni tra coppie o sequenze di pixel, consentendo di quantificare l'organizzazione spaziale e l'eterogeneità dell'immagine. Entrambe le tipologie di *feature* sono state calcolate mediante la libreria *Pyradiomics* di *Python* nell'ambiente *Visual studio code*. Tale libreria consente il calcolo dei seguenti parametri di tessitura:

- *Primo ordine*: indicatori statistici derivati dall'istogramma dei valori di intensità dell'immagine.
- *Forma (2D)*: descrittori della dimensione e della forma bidimensionale della *Region of Interest* (ROI). Le *feature* sono calcolate a partire dalla forma approssimata definita dalla *mesh* circolare.
- *Forma (3D)*: descrittori delle dimensioni e della forma tridimensionale della ROI. Le *feature* sono derivate dalla forma approssimata definita dalla *mesh* triangolare [60].

Successivamente, è possibile estrarre *feature* di ordine superiore attraverso l'analisi di matrici derivate dall'immagine:

- *Matrice di co-occorrenza dei livelli di grigio (GLCM)*: consente di ottenere le caratteristiche di *Haralick*, analizzando la frequenza con cui specifiche coppie di livelli di grigio (intensità) si verificano in pixel adiacenti.
- *Matrice delle zone dimensionali dei livelli di grigio (GLSZM)*: consente l'analisi delle "zone", ovvero regioni di pixel adiacenti con lo stesso livello di grigio. Tale matrice descrive la distribuzione dei livelli di grigio predominanti e le quantità di zone piccole o grandi all'interno della *ROI* [61].
- *Matrice di lunghezza di esecuzione dei livelli di grigio (GLRLM)*: fornisce una misura della lunghezza delle sequenze (*run*) di pixel consecutivi aventi lo stesso livello di grigio lungo una determinata direzione, evidenziando pattern lineari o striature. Le derivate descrivono la frequenza di sequenze lunghe o corte, utili per quantificare la grossolanità e la frequenza dei *run* a bassa o alta intensità [61].
- *Matrice di differenza dei toni di grigio adiacenti (NGTDM)*: calcola la differenza di intensità tra ciascun pixel e i suoi vicini. Le *feature* associate forniscono una misura diretta dell'omogeneità locale e della granularità o finezza della texture [61].
- *Matrice di dipendenza dei livelli di grigio (GLDM)*: quantifica le dipendenze dei livelli di grigio all'interno di un'immagine. La dipendenza è definita come il numero di pixel connessi entro una determinata distanza. Le *feature* derivate forniscono informazioni sull'omogeneità complessiva dell'immagine [60].

Nelle fasi preliminari di analisi del presente studio, sono state calcolate le *feature* di tessitura di primo ordine e le caratteristiche derivate dalla matrice GLCM. L'impiego congiunto di tali insiemi di parametri consente di catturare sia le variazioni locali di intensità dei livelli di grigio, sia le relazioni strutturali globali. Inoltre, la limitazione del dataset a queste due tipologie di *feature* risulta conforme allo stato dell'arte [22, 62, 63] e consente di evitare dataset eccessivamente ampi, che potrebbero generare *overfitting*. Successivamente, per l'implementazione del regressore basato sull'algoritmo XGBoost, sono stati calcolati tutti i parametri di tessitura mediante la libreria *PyRadiomics*.

3.4.3 Feature di tessitura del primo ordine

Le *feature* di tessitura del primo ordine costituiscono operatori statistici essenziali che dipendono esclusivamente dall'istogramma delle luminosità, poichè possono essere calcolate direttamente a partire da esso [64]. L'istogramma di luminosità di un'immagine rappresenta graficamente la distribuzione dei pixel dell'immagine in funzione dei livelli di grigio (intensità luminosa). In tale rappresentazione, l'asse delle ascisse riporta i livelli di luminosità espressi in valori numerici (da 0 a 255), mentre sull'asse delle ordinate è rappresentato il numero di pixel corrispondenti a ciascun valore. È importante notare che tale rappresentazione non conserva la coerenza spaziale dei pixel, in quanto dipende esclusivamente dal valore di grigio del singolo pixel e non dalle relazioni con i pixel circostanti. Nella Tabella 3.4 sono riportati alcuni parametri di tessitura di primo ordine presi in esame nel presente studio.

3.4.4 Feature di tessitura di ordine superiore

Le caratteristiche di tessitura di ordine superiore si focalizzano sulla descrizione dei pattern spaziali e dell'eterogeneità che emergono tra pixel adiacenti. Tali *feature* derivano da matrici di co-occorrenza o di relazione e consentono di quantificare la distribuzione e la variazione dei livelli di grigio tra coppie, triplette o regioni di pixel, piuttosto che i valori di intensità assoluti [64]. Esistono numerose caratteristiche di texture di ordine superiore impiegate nell'analisi delle immagini ecografiche. In particolare, l'ordine è determinato dal numero di pixel utilizzati nel calcolo dei parametri: per le *feature* di secondo ordine si considerano coppie di pixel, per quelle di terzo ordine triplette, e così via per gli ordini successivi.

Features della matrice GLCM

Le caratteristiche della matrice GLCM appartengono al secondo ordine e dipendono dalla posizione relativa dei pixel; vengono calcolate lungo direzioni specifiche. Si basano sulla matrice di co-occorrenza (*GLCM: Gray Level Co-occurrence Matrix*), che rappresenta l'istogramma bidimensionale dei livelli di grigio dell'immagine [64]. La matrice GLCM, indicata con c , è una matrice quadrata calcolata a partire da un'immagine I di dimensioni $n \times m$, avente dimensioni pari al numero di livelli di grigio nell'immagine. Il valore dell'elemento (i,j) all'interno della matrice c , indicato con $c(i,j)$, rappresenta il numero di volte in cui un pixel di livello di grigio i risulta adiacente a un pixel di livello j , come illustrato nella Figura 3.6. I due pixel possono risultare adiacenti in orizzontale, verticale o lungo le direzioni diagonali; pertanto, possono essere calcolate quattro matrici GLCM distinte, ciascuna corrispondente a una specifica direzione (0° , 45° , 90° , 135°).

Nelle fasi preliminari di analisi del presente studio, il sottoinsieme di caratteristiche GLCM selezionato è riportato nella Tabella 3.5. Queste *feature* di tessitura, scelte sulla base dell'analisi esplorativa, sono state calcolate per tutte e quattro le direzioni.

Tabella 3.4: Parametri di tessitura del primo ordine.

Caratteristica	Formula	Descrizione
Media (\bar{X})	$\frac{1}{N_p} \sum_{i=1}^{N_p} X(i)$	Intensità media dei livelli di grigio all'interno della ROI.
Mediana (M)	P_{50}	Valore di intensità che separa la metà superiore dalla metà inferiore dell'istogramma di luminosità (50° percentile).
Varianza (σ^2)	$\frac{1}{N_p} \sum_{i=1}^{N_p} (X(i) - \bar{X})^2$	Misura della dispersione dei valori di intensità rispetto alla media.
Range (R)	$\max(X) - \min(X)$	Differenza tra il valore massimo e quello minimo dei livelli di intensità osservati nella ROI.
Asimmetria (S_k)	$\frac{1}{N_p} \frac{\sum_{i=1}^{N_p} (X(i) - \bar{X})^3}{\sigma^3}$	Indice dell'asimmetria rispetto alla media dell'istogramma dei livelli di grigio (<i>skewness</i>).
Curtosi (K_t)	$\frac{1}{N_p} \frac{\sum_{i=1}^{N_p} (X - \bar{X})^4}{\sigma^4}$	Misura della "larghezza" delle code della distribuzione. Valori elevati indicano picchi elevati.
Entropia (E)	$-\sum_{i=1}^{N_g} p_i \log_2(p_i)$	Misura del disordine o della casualità (<i>randomness</i>) dell'immagine; valori più elevati indicano maggiore eterogeneità.
Energia (e)	$-\sum_{i=1}^{N_p} X^2(i)$	Misura del livello di uniformità dell'immagine; valori più elevati corrispondono a minore complessità strutturale.
Deviazione Assoluta Media (MAD)	$\frac{1}{N_p} \sum_{i=1}^{N_p} X(i) - \bar{X} $	Distanza media di tutti i valori di intensità dal valore medio dell'immagine.
Valore quadratico medio (RMS)	$\sqrt{\frac{1}{N_p} \sum_{i=1}^{N_p} X^2(i)}$	Rappresenta l'ampiezza media dei valori di intensità all'interno della ROI.
Uniformità (U)	$\sum_{i=1}^{N_g} p_i^2$	Misura la somma dei quadrati delle probabilità di ciascun livello di intensità. Valori più alti indicano maggiore omogeneità.

Legenda dei parametri:

N_g : numero di livelli di grigio discreti.

N_p : numero totale di pixel nella ROI.

$X(i)$: livello di grigio del pixel i .

$\sigma = \sqrt{\sigma^2}$: deviazione standard dei livelli di grigio.

p_i : probabilità che un pixel presenti il livello di grigio i .

P_{50} : 50° percentile dell'istogramma di luminosità.

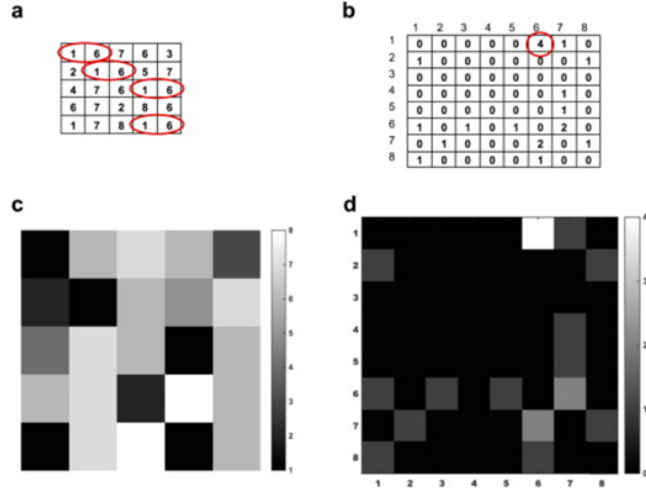


Figura 3.6: Esempio di costruzione della matrice di co-occorrenza (GLCM). a) Valori numerici corrispondenti ai livelli di intensità dei pixel dell'immagine. b) GLCM corrispondente. c) Rappresentazione in scala di grigi dell'immagine mostrata in (a). d) Visualizzazione in scala di grigi della matrice GLCM. I cerchi rossi in (a) indicano le adiacenze (1,6), corrispondenti al valore evidenziato in rosso nel pannello (b).

3.5 Stima della velocità del suono globale mediante regressore XGBoost

Nel capitolo precedente è stato sottolineato come il *Gradient Boosting* rappresenti una delle tecniche di apprendimento più accurate attualmente disponibili in letteratura. A partire dalle caratteristiche di tessitura descritte nel paragrafo precedente, la scelta del modello per la stima della SoS è stata guidata da un'analisi della letteratura, con l'obiettivo di individuare un metodo in grado di garantire robustezza, accuratezza e una buona capacità di generalizzazione su dataset numerosi e complessi, come quelli ecografici. Tra i diversi algoritmi disponibili, è stato pertanto selezionato il *Gradient Boosting*, e in particolare il framework XGBoost, grazie alla sua capacità di ridurre l'overfitting tramite meccanismi di regolarizzazione e di gestire efficacemente un numero elevato di *feature* fortemente correlate tra loro, come quelle presenti nel dataset di questo studio. L'impiego dei parametri di tessitura come caratteristiche di input per l'addestramento è coerente con i più recenti studi in letteratura, che hanno evidenziato una buona correlazione tra tali parametri e la velocità di propagazione del suono impiegata durante il processo di beamforming [63, 65, 66]. A scopo di ricerca, sono state impiegate per l'addestramento del modello tutte le *feature* di tessitura calcolabili mediante la libreria *PyRadiomics*, descritte nel Paragrafo 3.4.2, ad eccezione dei parametri di forma 2D e 3D. Il dataset di input è quindi composto da 140 immagini simulate ottenute con il simulatore *k-Wave* e da due immagini acquisite su fantocci in vitro descritti in precedenza.

Tabella 3.5: Sottoinsieme di parametri di tessitura derivati dalla matrice GLCM selezionati nell'analisi esplorativa.

Caratteristica	Formula	Descrizione
Autocorrelazione (AC)	$\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} c(i, j)ij$	Misura il grado di correlazione lineare tra coppie di pixel adiacenti.
Momento di Differenza Inversa Normalizzato (IDMN)	$\sum_{k=0}^{N_g-1} \frac{c_{x-y}(k)}{1+(\frac{k^2}{N_g^2})}$	Indica il grado di omogeneità locale dell'immagine. Valori elevati denotano regioni più uniformi e regolari.
Differenza Inversa Normalizzata (IDN)	$\sum_{k=0}^{N_g-1} \frac{c_{(x-y)}(k)}{1+k}$	Misura dell'omogeneità locale di un'immagine.
Media Congiunta (JA)	$\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} c(i, j)i$	Rappresenta l'intensità media dei livelli di grigio dell'immagine, pesata sulla matrice di co-occorrenza.
Somma Media (SA)	$\sum_{k=2}^{2N_g} c_{(x+y)}(k)k$	Misura la relazione tra le occorrenze di coppie di pixel aventi livelli di intensità simili o prossimi.

Legenda dei parametri:

N_g : numero di livelli di grigio discreti

$c_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} c(i, j)$ dove $|i - j| = k$. Indica la distribuzione della differenza di intensità.

$c_{x+y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} c(i, j)$ dove $i + j = k$. Indica la distribuzione della somma di intensità.

3.5.1 Struttura del modello

L'algoritmo è stato implementato in *Python* nell'ambiente di sviluppo *Visual Studio Code*. La pipeline sviluppata si articola nelle seguenti fasi principali:

1. *Estrazione e organizzazione delle feature*: su tutte le immagini del dataset sono stati calcolati tutti i parametri di tessitura, includendo i gruppi *FirstOrder*, *GLCM*, *GLRLM*, *GLSZM*, *NGTDM* e *GLDM*. I valori estratti sono stati organizzati in un file *Excel* a più fogli, in cui ciascun foglio rappresenta un gruppo di feature radiomiche. Successivamente, è stato creato un file *csv* contenente le etichette corrispondenti alla velocità del suono associata a ciascuna immagine. Le etichette sono state ricavate dal nome del file, definito in precedenza.
2. *Pre-processing dei dati*: i fogli contenenti le feature radiomiche sono stati unificati in un unico *dataframe* mediante la funzione *merge*. Successivamente, sono stati rimossi i dati relativi alle due immagini di fantoccio acquisite in vitro, destinate a costruire un *test set* indipendente.
3. *Scrematura delle feature*: per ridurre la dimensionalità e migliorare la capacità di generalizzazione del modello, sono state adottate due procedure di scrematura

delle feature:

- *Riduzione per varianza*: le feature con varianza inferiore a una soglia prefissata (10^{-2}) sono state rimosse.
- *Riduzione per correlazione e importanza*: tra le feature fortemente correlate ($R^2 > 0,8-0,95$), è stata mantenuta soltanto quella con maggiore importanza, secondo la valutazione preliminare di un modello XGBoost addestrato sul *training set*.

Tale processo è stato ripetuto all'interno di ogni ciclo di *cross-validation*.

4. *Divisione del dataset e validazione annidata*: i dati sono stati suddivisi in un set di *train-validation* (85%) e un set di test (15%). Il training è stato condotto mediante una *nested cross-validation* (5 x 3 *fold*), che prevede:
 - una validazione interna (*inner loop*), dedicata al *tuning* automatico dei parametri.
 - una validazione esterna (*outer loop*), finalizzata alla stima delle prestazioni generali del modello.
5. *Tuning automatico degli iperparametri*: durante la fase interna di cross-validation, è stata effettuata una ricerca su un insieme di iperparametri comprendente:
 - Profondità massima: profondità massima raggiungibile da un albero decisionale.
 - Tasso di apprendimento: determina la velocità con cui l'algoritmo apprende a ogni iterazione.
 - gamma: soglia minima di riduzione della funzione di perdita necessaria per effettuare un'ulteriore suddivisione su un nodo foglia.
 - Lambda: termine di regolarizzazione L2.
 - Alpha: termine di regolarizzazione L1.
 - Soglia di correlazione: valore soglia di R^2 oltre il quale due caratteristiche sono considerate fortemente correlate e , quindi, soggette alla procedura di scrematura.

Per ciascuna combinazione di iperparametri, il modello è stato addestrato e valutato mediante la metrica di errore assoluto medio (MAE, definita nel Paragrafo 3.5.2). Gli iperparametri ottimali individuati in ciascun ciclo interno sono stati successivamente utilizzati per la valutazione esterna, consentendo di identificare la combinazione con le migliori prestazioni complessive.

6. *Addestramento finale e test*: una volta determinati gli iperparametri ottimali, il set di *training-validation* è stato suddiviso in due set: il *training set* e il *validation set*, rispettivamente il 70% e il 15% del dataset completo. Successivamente, il modello XGBoost finale è stato addestrato utilizzando unicamente il *training set*.

Sul *validation set* è stato valutato l'overfitting attraverso un processo di tuning manuale dei seguenti parametri:

- Numero di iterazioni
- Criterio di arresto anticipato (*early stopping*): tecnica di prevenzione dell'overfitting che consiste nell'interrompere l'addestramento del modello prima che le sue prestazioni sul set di validazione inizino a peggiorare. Viene definito un numero massimo di iterazioni consecutive durante le quali la metrica di errore deve mostrare un miglioramento; se tale miglioramento non si verifica entro questo intervallo, il processo di addestramento viene interrotto automaticamente.
- Frazione di campioni: percentuale di campioni utilizzata per la costruzione di ciascun albero.
- Frazione di caratteristiche per albero: numero di feature selezionate casualmente per ogni albero.

Le prestazioni finali sono state poi valutate sui due *test set*.

7. *Salvataggio dei dati*: i risultati delle predizioni e delle metriche di validazione sono stati salvati in formato *Excel* e *JSON* per garantire la riproducibilità dei risultati
8. *Implementazione e ambiente di calcolo*: sono state utilizzate le librerie *NumPy*, *Pandas*, *SimpleITK*, *PyRadiomics*, *scikit-learn* e *XGBoost*. L'esecuzione è avvenuta su sistema operativo *Windows 11* a 64 bit, con processore Intel i7 e 16 GB di RAM.

Iperparametro	Valori testati
Massima profondità	{2, 3, 4}
Learning rate	{0.01, 0.03, 0.05}
Gamma	{0, 10, 20}
Lambda	{10, 20}
Alpha	{0, 1}
Soglia di correlazione	{0.8, 0.9, 0.95}
Numero di iterazioni	{500, 600, 1000}
Soglia di arresto	{50, 100, 150}
Frazione di campioni	{500, 600, 700, 1000}
Frazione di caratteristiche	{0.8, 0.9, 1}

Tabella 3.6: Iperparametri e intervalli di valori considerati durante la fase di tuning del modello XGBoost per la stima della SoS globale.

3.5.2 Metriche di valutazione

Per la valutazione delle prestazioni del modello possono essere impiegate diverse metriche, in grado di quantificare l'accuratezza con cui il modello riesce a effettuare le predizioni. Tra le metriche disponibili, in questo studio è stato utilizzato l'errore assoluto medio (*Mean Absolute Error*, MAE). Per calcolare la MAE si considera, per ciascun campione, la differenza tra la predizione del modello (\hat{y}_i) e il valore reale (y_i); si prende quindi il valore assoluto di tale differenza e se ne calcola la media rispetto al numero di campioni utilizzati (N). Questa funzione di errore può essere descritta dalla seguente relazione matematica:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (3.7)$$

Nel presente studio è stata scelta questa metrica poichè la variabile da stimare, ossia la SoS, è una variabile continua. La MAE rappresenta una misura diretta e interpretabile della deviazione media tra i valori predetti e quelli reali, espressa nella stessa unità di misura della grandezza fisica stimata. A differenza della *Mean Squared Error* (MSE), descritta nell'Equazione 2.5, che eleva al quadrato gli errori penalizzando in maniera più marcata le deviazioni elevate, la MAE tratta tutti gli errori in modo lineare, senza amplificare l'influenza di eventuali *outliers* presenti nel dataset.

3.6 Metodo SNAC per la stima della velocità del suono globale

Il presente lavoro di tesi si basa sul confronto tra i risultati ottenuti con il classificatore XGBoost e altri metodi presenti in letteratura per la stima della velocità del suono. Tra i metodi selezionati per il confronto, è stato considerato *One shot, one SoS*, sviluppato da *De la Torre et al.* [31], che consente di effettuare la stima globale della SoS a partire da una singola trasmissione non focalizzata. Questa metodologia è stata sperimentalmente validata e permette la stima della SoS in tempo reale, rendendola idonea all'integrazione all'interno della pipeline di elaborazione. Il metodo si basa sulla progettazione di una strategia di minimizzazione della perdita, che confronta la somma dei coefficienti di autocorrelazione normalizzati (*SNAC*) per determinare la stima della SoS [31].

3.6.1 Principi teorici

In Figura 3.7 è illustrato il framework del metodo. L'approccio proposto si basa sull'elaborazione dei dati RF grezzi in dominio canale (*pre-beamforming*), acquisiti da una singola trasmissione a onda piana, sia nella parte simulata del dataset sia in quella acquisita su fantocci in-vitro. Successivamente, i dati vengono filtrati tramite un filtro passa-banda centrato sulla frequenza centrale di trasmissione, e il segnale analitico si ottiene mediante la trasformata di Hilbert. In seguito, si applica la sola fase

geometrica del DAS *beamforming*, considerando l'effetto della rifrazione all'interfaccia sonda-mezzo, per un insieme di valori candidati della SoS, per ciascuna posizione del pixel (Figura 3.7, passo 1). In questo modo, per ogni pixel è possibile estrarre l'insieme dei dati di canale a tempo ritardato, contenenti il campione corrispondente di ciascun canale dopo la compensazione temporale. La somma coerente di tali insiemi consente di generare il valore di pixel beamformato dell'immagine a ultrasuoni (Figura 3.7, passo 2). Il processo di *beamforming* si applica alla matrice bidimensionale di dati RF $[N_S, N_E]$ (numero di campioni per elemento, numero di elementi), producendo un output tridimensionale $[N_Z, N_X, N_E]$ (numero pixel assiali, numero di pixel laterali e numero di elementi). Alcune regioni non vengono utilizzate per il beamforming a causa delle considerazioni legate all'*F-number* (indicate in nero in Figura 3.7, passo 2). Successivamente, a partire dai dati di canale a tempo ritardato $[N_E, 1]$ in ciascuna posizione dei pixel, si calcola una funzione di perdita basata sullo SNAC negato. Tale funzione raggiunge il minimo quando la SoS candidata utilizzata nel beamforming coincide con quella effettiva del mezzo. Tutte le funzioni di perdita vengono poi sommate sui pixel per costruire una metrica dipendente dalla SoS (Figura 3.7, passo 3). Infine, si applica una *spline di smoothing* alla metrica di perdita, in modo da interpolare i valori ottenuti per i diversi valori di SoS preselezionati. La stima finale della SoS è determinata dal minimo della curva di perdita adattata (Figura 3.7, passo 4). La funzione di perdita da minimizzare per stimare la SoS è rappresentata dallo SNAC totale calcolato su tutti gli insiemi di dati di canale a tempo ritardato all'interno della regione di interesse. In altri termini, la funzione di perdita corrisponde allo SNAC totale negato, calcolato su tutti i pixel dell'immagine per ciascun valore candidato di SoS (c) nel processo BF. Essa può essere espressa matematicamente come:

$$\mathcal{L}(c, \alpha) = - \sum_{p \in P} \sum_{l=l_{\min}}^{l_{\max}} \left| \frac{R[d_p(c, \alpha), l]}{\max_{l \in L} \{R[d_p(c, \alpha), l]\}} \right| \quad (3.8)$$

dove $R[d, l]$ rappresenta la funzione di autocorrelazione complessa di un insieme di dati d al ritardo l , e la somma interna costituisce lo SNAC del particolare insieme di dati di canale a tempo ritardato. La funzione $R[d, l]$ è definita come:

$$R[d, l] = \sum_{i=-\infty}^{\infty} d_i d_{i-l}^* \quad (3.9)$$

dove d_i rappresenta l'elemento i nell'insieme di dati d e l'asterisco $*$ indica il complesso coniugato [31].

3.6.2 Implementazione e parametri

Nell'ambito del presente lavoro di tesi, il metodo SNAC è stato implementato in *Matlab R2023a*, seguendo l'architettura descritta da *De la Torre et al.* [31]. I dati RF grezzi sono stati filtrati mediante un filtro passa-banda FIR (ordine 80, banda 6-9 MHz) e, successivamente, trasformati nel dominio analitico tramite la trasformata di Hilbert. La

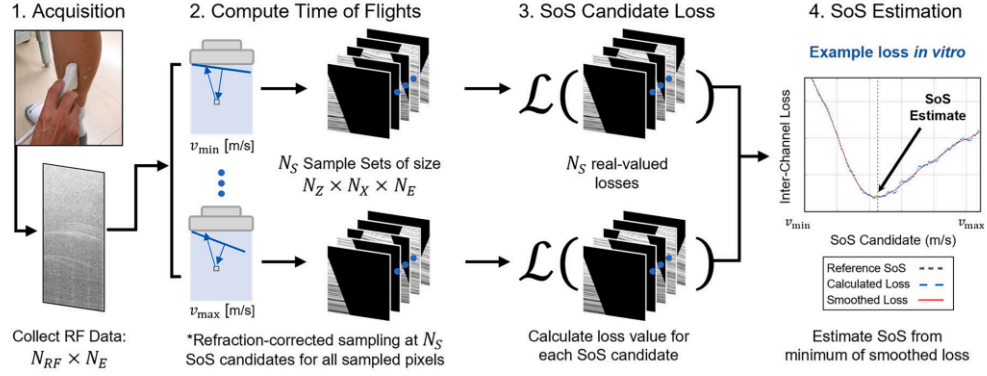


Figura 3.7: Framework del metodo SNAC. I dati vengono acquisiti tramite una singola trasmissione a onda piana (passo 1). Successivamente, i campioni di ciascun canale vengono opportunamente ritardati e corretti per la rifrazione in corrispondenza di ciascun candidato SoS (passo 2). Al passo 3, viene calcolata la funzione di perdita per ogni valore di SoS candidato. Infine, la stima della SoS è ottenuta selezionando il minimo della funzione di perdita opportunamente interpolata (passo 4).

frequenza di campionamento utilizzata è pari a 31,25 MHz, identica a quella impiegata durante la fase di acquisizione. Il processo di stima è stato eseguito considerando valori candidati di velocità del suono compresi tra 1200 e 1700 m/s, con un passo di 10 m/s, e un F -number pari a 1,4. La funzione di perdita finale, calcolata per ciascun valore candidato di SoS, è stata interpolata mediante una *smoothing spline* (parametro di smoothing pari a 0,99), dalla quale è stato ricavato il valore minimo corrispondente alla stima finale della SoS. Per aumentare la robustezza della metrica utilizzata, il calcolo è stato limitato a un sottoinsieme di pixel caratterizzati da un'elevata intensità post-beamforming. In particolare, sono stati considerati solo i pixel appartenenti all'80% più intenso dell'immagine.

3.7 Confronto tra gli stimatori della velocità del suono globale

Per valutare le prestazioni dei due stimatori della velocità del suono (SoS) sono stati adottati criteri di valutazione distinti per il test simulato e per il test sperimentale, in funzione delle caratteristiche dei dati e degli obiettivi dell'analisi.

Nel caso del *test simulato*, in cui la SoS di riferimento è nota con precisione, il confronto quantitativo è stato effettuato tramite la *Mean Signed Deviation (MSD)*, espressa come media dell'errore tra il valore di riferimento y_i e quello stimato \hat{y}_i , accompagnata dalla *deviazione standard* ($\pm\sigma$) degli errori stessi.

$$MSD = \frac{1}{N} \sum_{i=1}^N y_i - \hat{y}_i \quad (3.10)$$

Questa metrica consente di quantificare la tendenza dello stimatore a sovrastimare o sottostimare la SoS reale, fornendo al tempo stesso una misura della variabilità delle stime; a queste grandezze è stato affiancato il coefficiente di determinazione R^2 , utilizzato per valutare la capacità del modello di spiegare la variabilità dei dati simulati. Poiché l'obiettivo dello studio è la stima della SoS globale del mezzo, la valutazione è stata condotta considerando tutte le immagini di ciascun set di test. Tuttavia, per un'analisi più dettagliata, nel caso del test set simulato – che comprende sia immagini caratterizzate da una singola velocità del suono sia immagini bistrato (orizzontali e verticali) con due SoS differenti – il confronto è stato esteso anche ai sottoinsiemi corrispondenti alle diverse configurazioni strutturali del mezzo. In questo modo, è stato possibile analizzare la robustezza e la coerenza di ogni stimatore, sia in condizioni omogenee, sia in presenza di discontinuità acustiche, valutando come la complessità del mezzo influenzi la capacità del modello di fornire una stima accurata. Il modello XGBoost, in particolare, è stato confrontato con il metodo SNAC con l'obiettivo di verificarne l'efficacia nel ridurre l'errore medio (MSD).

Per il *test sperimentale*, invece, in cui la SoS di riferimento è ottenuta da misure effettuate su fantocci reali e risente delle inevitabili incertezze sperimentali, la valutazione delle prestazioni è stata condotta in termini di errore di stima definito come differenza tra la SoS predetta e quella reale per ciascun metodo e per ciascun fantoccio, privilegiando così una metrica direttamente interpretabile dal punto di vista fisico e maggiormente rappresentativa della capacità dei modelli di riprodurre accuratamente le condizioni reali.

3.8 Stima della velocità del suono locale mediante regressore XGBoost

In questa sezione viene presentato un modello di regressione basato sul Gradient Boosting, che utilizza lo stesso framework XGBoost impiegato nel Paragrafo 3.5, ma adattato alla stima locale della velocità del suono (SoS) a livello di patch d'immagine, in linea con il secondo obiettivo di questa tesi. Le immagini considerate corrispondono ai segnali pre-beamforming del dataset, sui quali sono state calcolate le feature radiomiche di tessitura mediante la libreria *PyRadiomics*, escludendo i soli parametri di forma 2D e 3D. Ogni immagine è stata suddivisa in *otto patch* verticali adiacenti, prive di sovrapposizione, ciascuna costituita da due colonne di pixel e da tutte le righe dell'immagine. Il dataset di input coincide con quello descritto in precedenza, ottenuto tramite simulazioni con il software *k-Wave*, ma opportunamente riorganizzato a livello di patch; a esso sono state aggiunte anche le due immagini acquisite in vitro sui fantocci, anch'esse suddivise nelle corrispondenti patch verticali.

3.8.1 Struttura del modello

L'algoritmo è stato implementato in *Python* nell'ambiente di sviluppo *Visual Studio Code*. La pipeline sviluppata si articola nelle seguenti fasi principali:

1. *Estrazione e organizzazione delle patch*: per ciascuna immagine del dataset, contenente i segnali pre-beamforming, è stata eseguita una procedura di suddivisione in patch verticali al fine di ottenere una rappresentazione locale del contenuto informativo. In particolare, dopo il caricamento dei dati in formato MAT e una fase preliminare di ritaglio delle prime e ultime 20 righe per eliminare eventuali componenti non significative, ogni immagine è stata suddivisa in un insieme di patch adiacenti, prive di sovrapposizione, ciascuna costituita da tutte le righe dell'immagine e da due colonne di pixel. La procedura è stata implementata mediante uno script in *Python* che, oltre a generare le patch, associa a ciascuna di esse il valore di velocità del suono corrispondente. Tale valore viene ricavato automaticamente dal nome del file: nel caso di un'immagine omogenea viene assegnata a tutte le patch la medesima *ground truth*, mentre nei file contenenti due valori di SoS il codice distingue tra interfaccia verticale (assegnando alle patch della metà sinistra e destra le due SoS fornite) e interfaccia orizzontale (attribuendo a tutte le patch la media armonica dei due valori). Per ogni immagine sono state inoltre registrate informazioni aggiuntive quali l'indice dell'immagine sorgente, l'indice di riga e colonna della patch e il nome del file originale. Tutti i dati generati sono stati infine salvati in un unico file MAT contenente:
 - la collezione delle patch estratte;
 - i valori di ground truth associati;
 - gli identificativi delle immagini e delle patch;
 - i metadati necessari alla ricostruzione del dataset.
2. *Estrazione e organizzazione delle feature*: a partire dal file MAT contenente le patch verticali e le relative informazioni ausiliarie, per ciascuna patch è stato effettuato il calcolo dei parametri di tessitura mediante la libreria *PyRadiomics*. In particolare, sono stati considerati i gruppi di feature *FirstOrder*, *GLCM*, *GLSZM*, *GLRLM*, *NGTDM* e *GLDM*. Ogni patch, normalizzata e trattata come immagine bidimensionale, è stata analizzata utilizzando una maschera binaria che include tutti i pixel, così da sfruttare l'intera informazione contenuta nel segnale pre-beamforming locale. Le feature ottenute sono state organizzate in una matrice di dimensione $N_{patch} \times N_{feature}$, in cui ogni riga rappresenta una patch e ogni colonna corrisponde a una specifica feature radiomica. In parallelo sono stati mantenuti e riallineati i vettori contenenti la ground truth della velocità del suono, l'identificativo dell'immagine di origine, gli indici di riga e colonna della patch e il nome del file sorgente. L'intero dataset così strutturato è stato infine salvato in un nuovo file MAT, utilizzato nelle successive fasi di addestramento e validazione del modello di regressione.
3. *Scrematura delle feature*: per ridurre la dimensionalità e migliorare la capacità di generalizzazione del modello, sono state adottate due procedure di scrematura delle feature:

- *Riduzione per varianza*: le feature con varianza inferiore a una soglia prefissata (10^{-2}) sono state rimosse.
- *Riduzione per correlazione e importanza*: tra le feature fortemente correlate ($R^2 > 0,8-0,95$), è stata mantenuta soltanto quella con maggiore importanza, secondo la valutazione preliminare di un modello XGBoost addestrato sul *training-validation set*.

Tale processo è stato ripetuto all'interno di ogni ciclo di *cross-validation*.

4. *Divisione del dataset*: a partire dal dataset radiomico descritto in precedenza, costituito dalla matrice delle feature, dal vettore dei valori di ground truth della velocità del suono e dagli identificativi di immagine e di patch, è stata effettuata una suddivisione in due insiemi distinti: *train-validation* e *test simulato*. La divisione è stata eseguita per immagine, utilizzando la funzione *GroupShuffleSplit* di *scikit-learn*, in modo che tutte le patch appartenenti alla stessa immagine venissero assegnate al medesimo insieme ed evitare così ogni forma di *leakage* tra training e test. In particolare, il 15% delle immagini è stato riservato al test simulato, mentre il restante 85% è stato utilizzato per le fasi di addestramento e validazione incrociata del modello. Il test set simulato viene mantenuto completamente separato durante il tuning degli iperparametri ed è impiegato esclusivamente in una fase successiva per la valutazione finale delle prestazioni del regressore XGBoost a livello di patch. Inoltre, è stato definito un ulteriore *test set sperimentale*, costituito dalle feature radiomiche estratte dalle patch delle immagini acquisite in vitro sui fantocci, utilizzato per valutare la capacità di generalizzazione del modello su dati reali non simulati.
5. *5-fold cross-validation e tuning degli iperparametri*: sul solo insieme train-validation è stata condotta una procedura di tuning degli iperparametri del modello XGBoost mediante 5-fold cross-validation a gruppi, utilizzando la classe *GroupKFold* e impiegando come gruppo l'identificativo dell'immagine. In questo modo, in ciascun fold le patch provenienti dalla stessa immagine risultano interamente contenute o nel training set o nel validation set, garantendo una stima più realistica della capacità di generalizzazione del modello su nuove immagini. Per ogni combinazione di iperparametri definita in una griglia di ricerca (comprendente, tra gli altri, profondità massima degli alberi, learning rate, numero di stimatori, termini di regolarizzazione L1 e L2, parametri di campionamento delle istanze e delle feature, soglia di varianza e soglia di correlazione), la pipeline prevede, all'interno di ciascun fold, i seguenti passaggi:
 - (a) Normalizzazione delle feature mediante *MinMaxScaler*, calcolata esclusivamente sui dati di training del fold e applicata poi anche al corrispondente validation set.
 - (b) Scrematura delle feature sul training set, attraverso la rimozione delle feature quasi costanti (varianza inferiore a una soglia prefissata) e una successiva riduzione delle feature altamente correlate, basata su una misura di importanza

calcolata con un modello XGBoost temporaneo. Solo le feature mantenute in questa fase vengono utilizzate anche sul validation set del fold.

- (c) Addestramento del regressore XGBoost con la combinazione corrente di iperparametri, impiegando l'errore assoluto medio (MAE) come metrica di valutazione e una strategia di *early stopping* sul validation set per prevenire l'overfitting.

Per ciascun fold viene calcolato il MAE di validazione, e per ogni combinazione di iperparametri si considera la media dei MAE sui 5 fold come indice di prestazione complessiva. La combinazione che minimizza tale valore medio viene selezionata come configurazione ottimale del modello e i risultati del tuning (inclusi i valori di MAE per tutte le combinazioni testate) vengono salvati per garantire tracciabilità e riproducibilità dell'analisi. Gli iperparametri considerati e i corrispondenti valori testati sono riassunti in Tabella 3.7.

Iperparametro	Valori testati
Massima profondità	{2, 4, 6}
Learning rate	{0.01, 0.03, 0.05}
Gamma	{0, 5, 10}
Lambda	{0, 10, 20}
Alpha	{0, 1, 5}
Soglia di correlazione	{0.85, 0.90, 0.95}
Numero di iterazioni	{800, 1000, 1200}
Soglia di arresto	{100}
Frazione di campioni	{0.8, 1.0}
Frazione di caratteristiche	{0.8, 1.0}

Tabella 3.7: Iperparametri e intervalli di valori considerati durante la fase di tuning del modello XGBoost per la stima patch-wise della velocità del suono.

6. *Addestramento finale e test*: una volta individuata, tramite la procedura di 5-fold cross-validation descritta in precedenza, la combinazione ottimale di iperparametri, il modello XGBoost è stato riaddestrato utilizzando l'intero insieme train-validation, senza effettuare ulteriore tuning. In questa fase gli iperparametri sono stati quindi fissati, mentre la pipeline di normalizzazione e di scrematura delle feature (riduzione per varianza e per correlazione) è stata applicata nuovamente seguendo le stesse modalità adottate durante il tuning. L'addestramento finale è stato eseguito impiegando l'errore assoluto medio (MAE) come metrica di ottimizzazione e una strategia di *early stopping* con numero massimo di iterazioni prefissato, al solo scopo di prevenire l'overfitting, senza ulteriori regolazioni manuali dei parametri. Le prestazioni del modello così addestrato sono state quindi valutate sui due insiemi di test separati.

Questa configurazione consente di stimare sia la capacità di generalizzazione del modello su nuove simulazioni, sia la sua trasferibilità a dati sperimentali reali.

7. *Salvataggio dei dati*: i risultati delle predizioni e delle metriche di validazione sono stati salvati in formato *Excel* e *JSON* per garantire la riproducibilità dei risultati
8. *Implementazione e ambiente di calcolo*: sono state utilizzate le librerie *NumPy*, *Pandas*, *SimpleITK*, *PyRadiomics*, *scikit-learn* e *XGBoost*. L'esecuzione è avvenuta su sistema operativo *Windows 11* a 64 bit, con processore Intel i7 e 16 GB di RAM.

3.9 Adattamento del metodo SNAC alla stima locale

Il metodo SNAC descritto nel Paragrafo 3.6 è stato opportunamente adattato per consentire la stima locale della velocità del suono. In particolare, l'algoritmo è stato applicato non più all'intera immagine beamformata, ma alle singole patch verticali definite nella sezione dedicata al regressore XGBoost: per ciascuna patch sono stati considerati i dati RF pre-beamforming corrispondenti e, per ognuno dei valori candidati di SoS, è stata calcolata la funzione di perdita SNAC limitata alla regione spaziale associata alla patch. La procedura di minimizzazione della metrica, nonché i parametri di implementazione (numero di candidati di SoS, finestra temporale, filtraggio dei segnali), sono stati mantenuti coerenti con quelli impiegati nella versione globale del metodo, così da permettere un confronto diretto con lo stimatore basato su XGBoost in termini di accuratezza locale della stima.

3.10 Fourier Neural Operators per la stima locale della SoS

Oltre al metodo SNAC opportunamente adattato per la stima locale della SoS e al regressore XGBoost basato sulle feature radiomiche, in questo capitolo viene introdotto un approccio di *deep learning* basato sui *Fourier Neural Operators* (FNO) per affrontare il medesimo problema di regressione *patch-wise*. Gli FNO rappresentano una famiglia di modelli progettati per apprendere, risultando particolarmente adatti a problemi di natura fisica o modellistica, nei quali la relazione tra dati di input e variabili può essere interpretata come un operatore integrale o convolutivo.

Nel contesto di questo lavoro, il modello apprende la mappatura

$$\mathcal{F} : \text{patch RF pre-beamforming} \longrightarrow \text{variazione locale della SoS}, \quad (3.11)$$

consentendo di esplorare un paradigma complementare rispetto sia all'approccio fisico-modellistico (SNAC) sia al metodo di *machine learning* tradizionale (XGBoost). L'adozione degli operatori neurali di Fourier permette inoltre di modellare dipendenze non locali, sfruttando una rappresentazione nel dominio delle frequenze.

Architettura del modello FNO

Nel presente lavoro è stato sviluppato un modello FNO che elabora patch di segnale RF pre-beamforming di dimensione $[B, 1, H, W]$, dove B è la dimensione del *batch* (32), H rappresenta l'asse temporale e W l'indice di elemento dell'array (16 per ogni patch). L'uscita della rete è costituita da uno scalare per patch, che esprime la variazione percentuale della velocità del suono rispetto al valore nominale di 1540 m/s.

L'architettura complessiva è stata articolata come segue.

1. *Linear lift (espansione delle feature)*. Il segnale RF in ingresso, caratterizzato da un solo canale, è stato trasformato tramite uno strato lineare completamente connesso, che ha agito come una convoluzione 1×1 sul canale delle feature. L'operazione è stata implementata come una moltiplicazione punto a punto tra la patch e una matrice W di dimensione $[H, W, width]$, così da aumentare la dimensionalità del canale a $width = 48$ e ottenere il tensore

$$[B, 1, H, W] \longrightarrow [B, width, H, W], \quad (3.12)$$

che ha fornito una rappresentazione iniziale più ricca per l'apprendimento spettrale.

2. *Blocchi FNO (SpectralConv2d + ramo residuo)*. La parte centrale del modello è stata costituita da due blocchi che hanno combinato un ramo spettrale e un ramo residuo spaziale.

Nel ramo spettrale è stato impiegato il blocco SpectralConv2d, caratterizzato dalle seguenti operazioni:

- è stato calcolato la trasformata di Fourier 2D (*rfft2*) lungo gli assi temporale e spaziale della patch;
- è stato selezionato un sottoinsieme di componenti a bassa frequenza;
- le componenti selezionate sono state moltiplicate punto a punto per i pesi apprendibili durante l'addestramento, mediante un'operazione di tipo *einsum*;
- è stato applicato uno *spectral dropout*, con annullamento di alcune componenti in frequenza per ridurre l'overfitting;
- tramite la trasformata di Fourier inversa *irfft2* si è fatto ritorno nel dominio spaziale.

In questo modo è stato fornito al modello un campo recettivo globale, in grado di cogliere dipendenze temporali e spaziali a lungo raggio.

In parallelo, nel ramo residuo è stata utilizzata una convoluzione 1×1 , che ha consentito di estrarre l'informazione locale nel dominio spaziale.

L'uscita di blocco, indicata con x_{out} , è stata ottenuta come somma dei contributi spettrale e spaziale:

$$x_{out} = \text{SpectralConv}(x) + \text{Conv}_{1 \times 1}(x). \quad (3.13)$$

3. *Normalizzazione, attivazione e dropout.* A valle della somma dei due rami è stata applicata una sequenza di operazioni composta da:

- *Batch Normalization* 2D per normalizzare le statistiche di batch e favorire una convergenza stabile;
- funzione di attivazione *GELU*;
- *Dropout* 2D, con annullamento casuale di alcune mappe di feature per limitare l'overfitting.

4. *Global average pooling e testa densa.* Successivamente, le dimensioni spaziali sono state compresse mediante una media globale:

$$x_{\text{gap}} = \text{mean}(x, \text{dim} = (2,3)), \quad (3.14)$$

da cui è stato ottenuto un vettore di dimensione $[B, \text{width}]$. Tale vettore è stato quindi fornito a una testa completamente connessa composta da:

- uno strato lineare $\text{width} \rightarrow 64$;
- un'attivazione GELU;
- uno strato di Dropout;
- uno strato lineare finale $64 \rightarrow 1$.

In uscita da questa sequenza è stato ottenuto, per ogni patch, un singolo valore reale y_{raw} che rappresenta la variazione normalizzata della SoS.

5. *Strato finale di calibrazione.* Per correggere l'eventuale bias tra valore predetto e valore reale è stato impiegato uno strato di calibrazione affine:

$$\hat{y} = a \cdot y_{\text{raw}} + b, \quad (3.15)$$

in cui a e b sono stati trattati come parametri apprendibili e sottoposti a termine di costo, con l'obiettivo di mantenerli rispettivamente vicini a 1 e a 0.

Sintesi del flusso operativo

In sintesi, il passaggio *end-to-end* del modello è stato riassunto come:

$$\begin{aligned} \text{RF patch} &\xrightarrow{\text{linear lift}} \left[\text{SpectralConv} + \text{Conv}_{1 \times 1} + \text{BN} + \text{GELU} + \text{Dropout} \right] \\ &\xrightarrow{\text{global avg. pooling}} \text{testa densa} \xrightarrow{\text{calibrazione affine}} \hat{\text{SoS}}_{\text{local}}. \end{aligned} \quad (3.16)$$

3.11 Confronto tra gli stimatori della velocità del suono locale

Per valutare le prestazioni dei tre stimatori locali della velocità del suono (XGBoost, SNAC adattato alle patches e OFN), è stato effettuato un confronto quantitativo sia su dati simulati sia su dati sperimentali acquisiti in vitro. Sul test set simulato, il confronto è stato condotto utilizzando la *Mean Signed Deviation* (MSD) tra il valore di riferimento y_i e quello stimato \hat{y}_i per ciascuna patch, definita come:

$$MSD = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i) \quad (3.17)$$

dove N rappresenta il numero totale di patch nel test set. Per ciascun metodo è stata inoltre calcolata la deviazione standard della distribuzione degli errori $y_i - \hat{y}_i$, al fine di quantificare la variabilità delle stime, e il coefficiente di determinazione R^2 , utilizzato come indice della capacità del modello di spiegare la variabilità della SoS locale. In questo modo, è stato possibile valutare congiuntamente la tendenza sistematica a sovrastimare o sottostimare la SoS (MSD), la dispersione delle stime (deviazione standard) e la qualità complessiva dell'adattamento (R^2) sui dati simulati.

Per quanto riguarda il test set sperimentale, costituito dalle immagini acquisite sui fantocci in vitro, la *ground truth* della velocità del suono è nota ed è identica per tutte le patch appartenenti alla stessa immagine. In questo caso, per ciascun metodo e per ciascuna immagine è stato calcolato il valore medio della SoS predetta sulle patch e la corrispondente deviazione standard tra le patch stesse. Il confronto con il valore di riferimento è stato quindi effettuato considerando la differenza tra la media delle stime locali e la SoS reale dell'immagine, mentre la deviazione standard delle predizioni è stata utilizzata come indicatore della coerenza spaziale delle stime all'interno dello stesso fantoccio. In altre parole, sul test sperimentale non è stato definito un ulteriore indice di errore globale, ma si è analizzato, per ciascuna immagine, l'andamento del valore medio predetto di SoS e della relativa variabilità intra-immagine, confrontando i tre metodi in termini di vicinanza al valore di riferimento e di stabilità delle stime *patch-wise*.

3.12 Test di normalità e analisi statistica

Per valutare in modo oggettivo le differenze tra gli stimatori della velocità del suono, sia nella stima globale sia nella stima locale, è stata adottata una procedura statistica uniforme basata sull'analisi degli errori con segno prodotti dai diversi metodi. Prima di effettuare il confronto statistico vero e proprio, tali errori sono stati sottoposti al test di normalità di *Shapiro-Wilk*, un test utilizzato per verificare se un insieme di dati possa essere considerato come proveniente da una distribuzione gaussiana. Il test valuta la coerenza tra la distribuzione empirica dei dati e quella teorica normale mediante un indice di correlazione: valori di p -value inferiori alla soglia di significatività adottata ($\alpha = 0.05$) indicano una violazione della normalità. Questa fase preliminare è risultata

cruciale poiché ha guidato la selezione del test statistico appropriato per il confronto tra i metodi.

Nei sottoinsiemi in cui la normalità non veniva violata è stato impiegato il *paired Student's t-test*, un test parametrico che valuta se la differenza media tra due insiemi di misure appaiate sia significativamente diversa da zero. Il test presuppone una distribuzione normale delle differenze ed è particolarmente adatto nei contesti in cui le osservazioni derivano dalle stesse entità sperimentali, come avviene nel confronto tra le stime fornite da due metodi sul medesimo insieme di immagini.

Al contrario, nei casi in cui gli errori mostravano deviazioni significative dalla normalità si è fatto ricorso al test non parametrico di *Wilcoxon per dati appaiati*. Questo test valuta la significatività delle differenze tra due metodi basandosi sui ranghi delle differenze anziché sui loro valori assoluti, risultando quindi più robusto rispetto a distribuzioni non gaussiane e alla presenza di valori anomali. Rispetto al t-test, il Wilcoxon non richiede alcuna assunzione sulla forma della distribuzione delle differenze, rappresentando quindi la scelta preferibile quando i dati presentano asimmetrie o variabilità non omogenea.

I test sono stati applicati agli errori con segno calcolati tra le stime fornite dagli stimatori: nel caso della stima globale il confronto è stato eseguito tra XGBoost e SNAC, mentre nel caso della stima locale sono stati considerati i confronti XGBoost–SNAC e XGBoost–OFN. Tutte le verifiche sono state condotte con un livello di significatività pari a $\alpha = 0.05$, e l'intera procedura è stata applicata in modo coerente sia ai confronti globali sia ai confronti locali, garantendo uniformità metodologica e piena comparabilità dei risultati.

Capitolo 4

Risultati e Discussioni

Il presente capitolo ha l'obiettivo di presentare e discutere i risultati ottenuti, facendo uso di grafici, tabelle e immagini. Esso è articolato in tre sezioni principali.

Nella prima, di carattere più esplorativo, viene analizzata la relazione tra i parametri di tessitura e i coefficienti di calibrazione (CF) impiegati durante la fase di *Digital Scan Conversion*, eseguita nel *post-beamforming*.

La seconda parte è dedicata allo studio della relazione tra i parametri di tessitura e la velocità del suono (SoS). In particolare, vengono analizzate le variazioni delle feature derivate dalla matrice GLCM al variare della SoS, al fine di identificare quali descrittori risultino maggiormente sensibili alle diverse condizioni di propagazione acustica.

Infine, l'ultima parte è focalizzata sull'identificazione della SoS ottimale da impiegare nel processo di beamforming, mediante un'analisi comparativa delle prestazioni dei diversi stimatori.

4.1 Coefficienti di calibrazione e tessitura dell'immagine

In questa prima fase è stata condotta un'analisi esplorativa con l'obiettivo di evidenziare l'effetto dei coefficienti di calibrazione introdotti nella fase di *Digital Scan Conversion* sui parametri di tessitura estratti dalle immagini ecografiche acquisite sul fantoccio *3B Scientific Breast Phantom*. Come definito nel Paragrafo 3.3.2, la DSC fa parte della pipeline di elaborazione dell'immagine post-beamforming, la cui fase centrale è rappresentata dall'interpolazione basata sulla definizione dei coefficienti di calibrazione (CF), che determinano la dimensione del pixel lungo l'asse x e l'asse z . I parametri di tessitura sono stati calcolati su regioni di interesse (ROI) e successivamente mediati, in modo da ottenere un parametro globale per ogni coppia di valori dei coefficienti di calibrazione. Sono state condotte due analisi distinte: nella prima sono state variate le dimensioni dei pixel sia in direzione laterale sia in direzione assiale, mentre nella seconda è stato variato il coefficiente di calibrazione esclusivamente in direzione assiale. La relazione tra i descrittori di tessitura e i coefficienti di calibrazione, in entrambe le

analisi, è stata valutata mediante il coefficiente di determinazione (R^2), corrispondente al quadrato del coefficiente di correlazione lineare di Pearson definito nell'Equazione 3.4. In aggiunta, è stata valutata anche la correlazione lineare-logaritmica, ottenuta applicando una trasformazione logaritmica ai coefficienti di calibrazione e calcolando il corrispondente coefficiente di determinazione.

4.1.1 Variazione del coefficiente di calibrazione lungo gli assi x e z

In questa analisi sono state investigate le correlazioni tra tutti i parametri di tessitura calcolabili con la libreria *PyRadiomics*, ad eccezione delle feature di forma 2D e 3D, e i coefficienti di calibrazione, assunti uguali lungo le direzioni x e z . Per i coefficienti di calibrazione sono stati testati i seguenti valori:

$$CF \in \{0,004 - 0,005 - 0,0075 - 0,01 - 0,025 - 0,05 - 0,075 - 0,1 - 0,2\}$$

La relazione è quindi valutata tra una variabile dipendente y (parametro di tessitura) e una variabile indipendente x (coefficiente di calibrazione). Le feature sono state calcolate su (ROI) quadrate di dimensione $5 \times 5 \text{ mm}$ e successivamente mediate. In Figura 4.1 si osserva che, per la maggior parte delle famiglie di feature, il modello di regressione logaritmico fornisce coefficienti di determinazione superiori rispetto al modello lineare. Questo comportamento è evidente soprattutto per le caratteristiche di primo ordine, GLCM, GLDM, GLRLM e GLSZM, dove numerose feature presentano un valore di R^2 logaritmico compreso tra 0,5 e 0,9, mentre il corrispondente modello lineare raramente supera 0,5. Un comportamento opposto si osserva per le feature NGDTM, per le quali il modello lineare fornisce valori R^2 maggiori rispetto a quello logaritmico.



Figura 4.1: Coefficienti di determinazione R^2 delle famiglie di feature di tessitura. Sono riportati i coefficienti di determinazione R^2 lineari e logaritmici calcolati tra i coefficienti di calibrazione lungo gli assi x e z e i parametri di tessitura (per le famiglie di feature di Primo ordine, GLCM, GLDM, GLRLM, GLSZM e NGDTM).

4.1.2 Variazione del coefficiente di calibrazione lungo l'asse z

Analogamente a quanto descritto nel paragrafo precedente, è stata analizzata la correlazione tra gli stessi parametri di tessitura e il coefficiente di calibrazione applicato esclusivamente lungo l'asse z, mantenendo quindi invariata la dimensione del pixel lungo l'asse x . Anche in questo caso sono state considerate sia la correlazione lineare sia quella logaritmica tra le variabili dipendenti (le feature di tessitura) e la variabile indipendente (i coefficienti di calibrazione lungo z). Per i coefficienti di calibrazione sono stati utilizzati gli stessi valori dell'analisi precedente, con l'aggiunta del caso senza interpolazione anche lungo l'asse z, che corrisponde a una dimensione del pixel in profondità pari a 0,024625 mm. Le feature sono state calcolate su ROI costituite da due colonne di pixel nel piano laterale, mentre lungo l'asse assiale sono state considerate tutte le righe, coprendo quindi l'intera profondità. Infine, i valori delle caratteristiche ottenuti sono stati mediati al fine di ricavare un parametro globale. Come emerso nell'analisi del paragrafo precedente, anche in questo scenario si osserva che il modello logaritmico fornisce complessivamente valori di R^2 più elevati rispetto al modello lineare: per molte feature il coefficiente di determinazione logaritmico si attesta tra circa 0,6 e 0,9, mentre il corrispondente modello lineare rimane spesso inferiore a 0,5. Le famiglie GLDM, GLRLM e GLSZM mostrano alcune feature con R^2 logaritmico particolarmente alto, indicando una forte sensibilità alla variazione del coefficiente di calibrazione lungo la profondità. Anche le feature NGDTM presentano valori di R^2 complessivamente buoni per il modello logaritmico, pur risultando in media meno correlati rispetto alle famiglie con correlazioni migliori.



Figura 4.2: Coefficienti di determinazione R^2 delle famiglie di feature di tessitura. Sono riportati i coefficienti di determinazione R^2 lineari e logaritmici calcolati tra i coefficienti di calibrazione lungo l'asse z e i parametri di tessitura (per le famiglie di feature di Primo ordine, GLCM, GLDM, GLRLM, GLSZM e NGDTM).

4.1.3 Discussione dei risultati

Nel confronto tra le Figure 4.1 e 4.2 si nota che il comportamento generale dei coefficienti di determinazione R^2 è coerente: le famiglie di feature che in Figura 4.1 mostrano una forte dipendenza dal fattore di calibrazione (soprattutto alcune GLCM; GLDM e GLSZM) mantengono una forte dipendenza anche quando si considera l'interpolazione solo lungo l'asse z , mentre feature meno sensibili restano tali. Tuttavia, passando dalla calibrazione combinata lungo x e z alla sola interpolazione lungo z , si osserva in diversi casi una riduzione dei valori di R^2 , a indicare che anche la risoluzione orizzontale contribuisce in modo non trascurabile alla variabilità delle feature di tessitura. I risultati ottenuti confermano quindi che i parametri di tessitura dipendono dalle scelte sui coefficienti di calibrazione adottate nella fase di Digital Scan Conversion e che il loro valore dipende in modo non trascurabile dalla risoluzione dei pixel. Per questo motivo, invece di selezionare un valore di calibrazione "ottimale" come riferimento, in questo studio si è preferito basare le analisi successive sugli approcci pre-DSC, operanti direttamente sui dati RF prima della ricostruzione dell'immagine. Tale scelta è coerente con l'intera pipeline proposta, in quanto tutti gli stimatori della velocità del suono considerati (SNAC, OFN e il modello XGBoost) si basano su grandezze derivate dal segnale *pre-beamforming* e non sui valori d'intensità dell'immagine post-DSC. In questo modo si evita l'introduzione di una variabilità aggiuntiva nelle feature di tessitura, rendendo più robusto e interpretabile il confronto tra i diversi approcci di stima della SoS.

4.2 Relazione tra la velocità del suono e le feature GLCM

In questa sezione è stata analizzata la relazione tra le caratteristiche di tessitura e la velocità del suono (SoS) assunta nel beamforming. In seguito ai risultati ottenuti nella Sezione 4.1, che hanno messo in evidenza l'effetto dei coefficienti di calibrazione sulle feature di tessitura, in questa sezione è stato studiato il loro andamento in condizioni *pre-DSC*. L'attenzione è stata rivolta in particolare alle feature derivate dalla matrice GLCM, valutandone la variazione in differenti condizioni di elaborazione e in specifiche regioni di interesse (ROI). L'obiettivo è quantificare quanto la scelta della SoS influenzi le caratteristiche di tessitura dell'immagine ecografica, così da fornire informazioni utili per la scelta e l'ottimizzazione della pipeline di stima.

4.2.1 Analisi su regioni di interesse

In questa prima fase dell'analisi, sono state calcolate le feature di tessitura derivate dalla matrice GLCM sulle immagini acquisite per entrambi i fantocci in vitro: il *3B Scientific Breast Phantom* e il fantoccio ad emulsione acqua-in-olio a base di PDMS/Agarosio, caratterizzati da una velocità di propagazione del suono pari rispettivamente a 1250 m/s e 1540 m/s. L'analisi è stata inizialmente condotta considerando, per ciascuna

immagine di ciascun fantoccio, due regioni di interesse (ROI) di dimensioni $5 \times 5 \text{ mm}$. Nel caso del fantoccio mammario, una ROI è stata collocata all'interno del nodulo benigno e la seconda nel tessuto esterno ad esso; per il fantoccio in PDMS, invece, le due ROI sono state posizionate a differenti profondità: una superficiale e una più profonda. Le posizioni delle ROI sono riportate in Figura 4.3 .

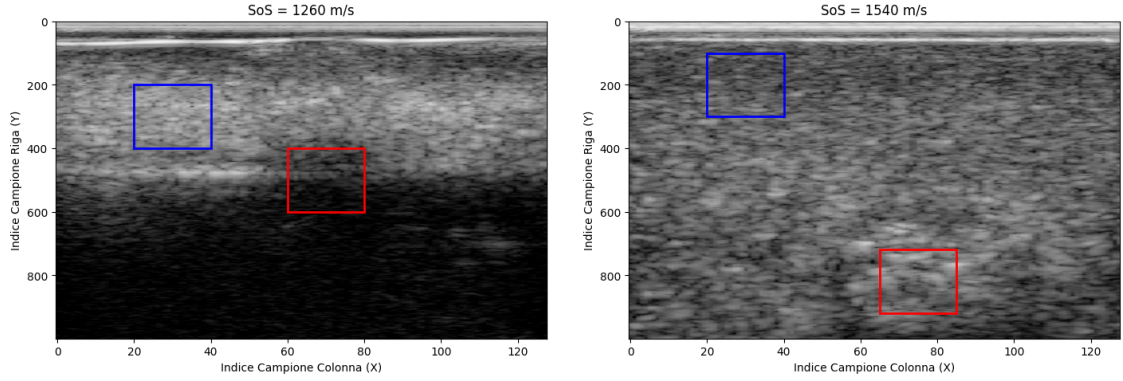


Figura 4.3: Posizionamento delle regioni di interesse nelle immagini ecografiche acquisite sui due fantocci. A sinistra è riportata l'immagine relativa al fantoccio in PDMS/Agaroso, con una ROI superficiale evidenziata in blu e una ROI profonda evidenziata in rosso. A destra è mostrata l'immagine del fantoccio di seno mammario, in cui la ROI in rosso è collocata all'interno della lesione e la ROI in blu è posizionata nel tessuto superficiale di riferimento.

Dopo aver definito le ROI, è stato quindi visualizzato l'andamento di tutte le feature GLCM in funzione della SoS in due fasi distinte della catena di elaborazione. Nella prima fase, l'analisi è stata condotta immediatamente dopo il *beamforming*.

In particolare:

1. è stato eseguito il beamforming ripetutamente, variando la velocità del suono da 1200 a 1700 m/s con passo pari a 20 m/s;
2. i segnali RF complessi ottenuti dal beamforming sono stati utilizzati direttamente per il calcolo delle feature;
3. sono state estratte le due ROI per ciascuna immagine e sono state calcolate le feature normalizzate;
4. infine, è stato rappresentato l'andamento di ogni feature in funzione della SoS in ciascuna ROI dei fantocci.

Successivamente, la stessa analisi è stata ripetuta in una seconda fase, simulando in maniera più realistica la formazione dell'immagine, ma senza applicare l'interpolazione tramite i coefficienti di calibrazione. In questo caso, è stata applicata tutta la pipeline descritta nella Sezione 3.3.2, ad eccezione dell'interpolazione:

1. è stato nuovamente effettuato il beamforming per valori di SoS compresi tra 1200 e 1700 m/s, con passo di 20 m/s;
2. sui segnali RF complessi è stato calcolato l'involuppo, così da ottenere segnali di intensità (ampiezza) reali;
3. è stata applicata una compressione logaritmica al fine di ridurre l'ampia gamma dinamica tipica del segnale ecografico;
4. è stato introdotto un offset con utilizzo di un range dinamico fissato ($dR = 50$) e i valori sono stati normalizzati, senza effettuare alcuna interpolazione;
5. da queste immagini così ottenute, sono stati calcolati i parametri di tessitura GLCM.

In questo modo, è stato possibile confrontare l'evoluzione delle feature GLCM all'interno delle ROI in due diverse fasi di elaborazione dell'immagine, valutando l'impatto che i diversi passaggi della pipeline hanno sulla dipendenza delle feature dalla velocità del suono.

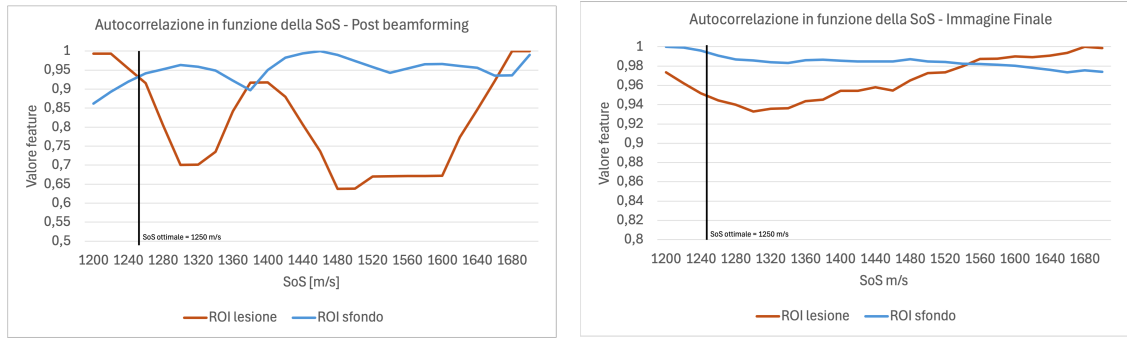


Figura 4.4: Relazione tra SoS e feature GLCM nel fantoccio in PDMS/Agarosio. Le curve mostrano l'andamento delle feature di tessitura derivate dalla matrice GLCM, calcolate sulle due ROI (superficiale e profonda) del fantoccio al variare della velocità del suono assunta nel beamforming. Nell'immagine di sinistra le feature sono calcolate sui dati immediatamente dopo il beamforming, mentre nell'immagine di destra sono calcolate sull'immagine ecografica finale. La linea verticale nera rappresenta la velocità di propagazione reale del suono nel mezzo ($SoS = 1250$ m/s).

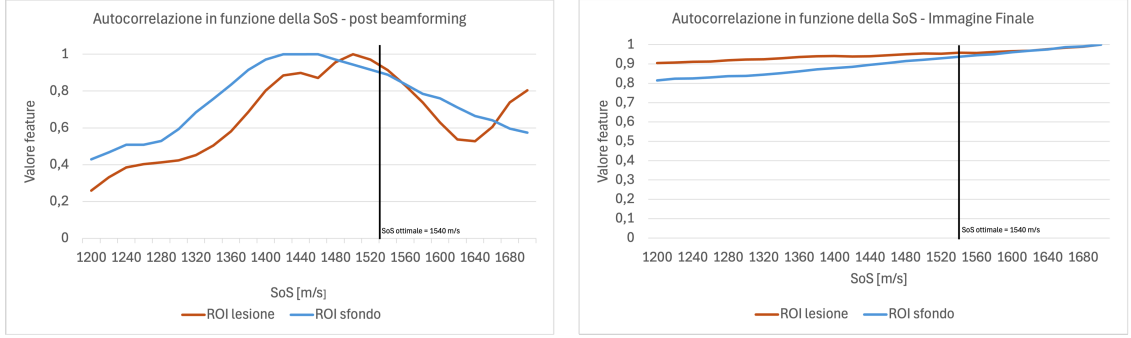


Figura 4.5: Relazione tra SoS e feature GLCM nel fantoccio mammario. Le curve mostrano l'andamento delle feature di tessitura derivate dalla matrice GLCM, calcolate sulle due ROI (all'interno del nodulo benigno e nel tessuto circostante) del *3B Scientific Breast Phantom* al variare della velocità del suono assunta nel beamforming. Nell'immagine di sinistra le feature sono calcolate immediatamente dopo il beamforming, mentre nell'immagine di destra sono calcolate sull'immagine ecografica finale. La linea verticale nera rappresenta la velocità di propagazione reale del suono nel mezzo (SoS = 1540 m/s).

Dall'analisi congiunta degli andamenti delle feature GLCM in funzione della SoS nelle due diverse fasi della catena di elaborazione, si è osservato che, nella fase relativa all'immagine finale (dopo inviluppo, compressione logaritmica e normalizzazione), nessuna feature ha mostrato una dipendenza evidente dalla SoS ottimale. Al contrario, nella fase immediatamente successiva al beamforming, cinque feature GLCM hanno evidenziato la presenza di un massimo in corrispondenza della velocità di propagazione reale del mezzo, in entrambe le ROI dei due fantocci. Nelle Figure 4.4 e 4.5 è riportata, a titolo esemplificativo, la feature di tessitura autocorrelazione (AC), che nella fase subito post-beamforming presenta una variazione evidente in corrispondenza della SoS ottimale, identificata dalla linea verticale nera. Oltre all'autocorrelazione, sono state individuate altre quattro feature: differenza inversa normalizzata (IDN), media congiunta (JA), momento di differenza inversa normalizzato (IDMN) e somma media (SA). Il comportamento di queste cinque feature GLCM è sintetizzato nella Tabella 4.1, nella quale è indicato che per ciascuna di esse si osserva un punto di massimo nelle ROI analizzate.

4.2.2 Estensione dell'analisi a livello globale di immagine

Nella seconda fase dell'analisi, lo studio è stato esteso a livello globale dell'immagine. Le feature GLCM sono state quindi calcolate sull'intera immagine ecografica ricostruita nella fase immediatamente successiva al beamforming, per entrambi i fantocci. In particolare, il beamforming è stato eseguito ripetutamente variando la velocità del suono tra 1200 e 1700 m/s, con passo di 20 m/s; i segnali RF complessi così ottenuti

Feature selezionate	Comportamento
Autocorrelazione (AC)	Massimo
Differenza inversa normalizzata (IDN)	Massimo
Media congiunta (JA),	Massimo
Momento di differenza inversa normalizzato (IDMN)	Massimo
Somma media (SA)	Massimo

Tabella 4.1: Comportamento delle feature GLCM in corrispondenza della SoS ottimale.

sono stati utilizzati direttamente per il calcolo delle caratteristiche su tutta l'immagine e, per ciascun fantoccio, è stato infine rappresentato l'andamento di ogni feature in funzione della SoS. Dall'osservazione di tali curve si è evidenziato come tutte le feature GLCM, incluse quelle che nella precedente analisi condotta sulle ROI mostrano un massimo o un comportamento peculiare in prossimità della SoS ottimale, perdano questa caratteristica quando il calcolo viene esteso all'intera immagine. Un esempio è riportato in Figura 4.6, che mostra l'andamento dell'autocorrelazione; lo stesso fenomeno si riscontra per tutte le altre quattro feature considerate nel capitolo precedente.

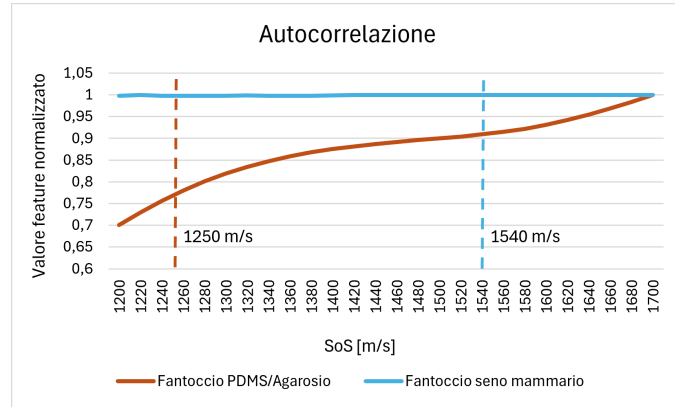


Figura 4.6: Andamento globale dell'autocorrelazione in funzione della velocità del suono. Le due curve mostrano l'autocorrelazione normalizzata calcolata sull'intera immagine ecografica, in funzione della velocità del suono assunta nel beamforming, per il fantoccio in PDMS/Agarosio e per il fantoccio di seno mammario. Le linee tratteggiate indicano le rispettive velocità di propagazione reali del suono nei due mezzi.

4.2.3 Mappe parametriche delle feature: confronto pre e post beamforming

Alla luce dei risultati ottenuti a livello globale di immagine, l'analisi è stata quindi approfondita passando a una descrizione spazialmente risolta delle feature GLCM. In

questa fase, sono state costruite mappe locali delle cinque feature considerate in precedenza, al fine di evidenziare come esse si distribuiscono all'interno dei fantocci. In particolare, ogni immagine è stata suddivisa in regioni di interesse di 5×5 mm, con una sovrapposizione del 50% sia in direzione orizzontale sia verticale. Per ciascuna ROI sono state calcolate le feature GLCM e, nelle zone di sovrapposizione, il valore finale è stato ottenuto come media dei contributi provenienti da tutte le ROI concorrenti. La stessa procedura è stata applicata sia ai dati grezzi pre-beamforming sia ai dati immediatamente successivi al beamforming, senza ulteriori passaggi della pipeline di elaborazione; il beamforming è stato eseguito assumendo, per ogni fantoccio, la velocità del suono reale del mezzo. L'obiettivo era verificare l'esistenza di una relazione tra i valori locali delle feature calcolate sui dati pre-beamforming e quelli ottenuti sulle immagini post-beamforming ricostruite con la SoS ottimale. Dall'osservazione delle mappe parametriche è emerso che, per quattro delle cinque feature analizzate (IDMN, IDN, Joint Average e Sum Average), le strutture spaziali risultano complessivamente simili nei due fantocci, sia prima che dopo il beamforming. Un esempio è riportato in Figura 4.7, che mostra le mappe della feature IDN pre e post beamforming e mette in evidenza la forte corrispondenza tra le due rappresentazioni; un comportamento analogo è stato riscontrato anche per le altre tre feature considerate. Solo l'autocorrelazione mostra un andamento differente, suggerendo una minore coerenza tra le mappe pre e post beamforming per questa specifica feature.

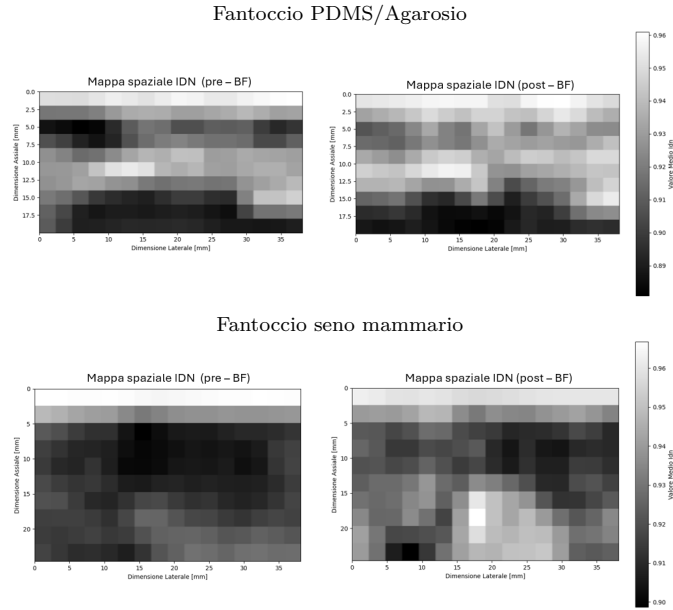


Figura 4.7: Mappe spaziali della feature IDN pre e post-beamforming. Confronto tra le mappe IDN ottenute prima e dopo il beamforming per i due fantocci, che mostrano una distribuzione spaziale complessivamente simile in entrambe le condizioni.

4.2.4 Discussione dei risultati

Dall'insieme dei risultati ottenuti emerge chiaramente che la relazione tra le feature GLCM e la velocità del suono è fortemente dipendente dalla regione analizzata. Le analisi condotte hanno infatti mostrato come, in condizioni locali e controllate, alcune feature presentino una sensibilità marcata alla variazione della SoS, mentre tale comportamento tende a dissolversi quando il calcolo viene esteso all'intera immagine, dove l'eterogeneità spaziale introduce variabilità che riduce la significatività complessiva.

L'analisi comparativa tra le mappe pre e post beamforming ha inoltre evidenziato che alcune feature conservano informazioni coerenti nelle due condizioni, suggerendo che, almeno per una parte delle caratteristiche tessiturali, il contenuto informativo rilevante è già presente nei dati pre-beamforming. Questo risultato apre la possibilità di evitare la generazione di numerose ricostruzioni per la stima della SoS e motiva l'impiego di un regressore capace di operare direttamente sui dati pre-beamforming. Un tale approccio consentirebbe di ridurre significativamente il costo computazionale, mantenendo al contempo la capacità di estrarre le informazioni utili per la stima della velocità del suono.

4.3 Metodi di stima della velocità del suono

In questo capitolo viene descritta la metodologia adottata per la stima quantitativa della velocità del suono (SoS) a partire da feature radiomiche estratte da immagini ecografiche. Dopo l'analisi esplorativa della relazione tra feature GLCM e SoS presentata nel capitolo precedente, l'attenzione viene ora estesa all'intero set di descrittori radiomici, impiegando modelli di regressione avanzati basati su XGBoost e confrontandone i risultati con metodi fisici di riferimento.

L'obiettivo del capitolo è duplice. Da un lato, valutare la capacità del modello XGBoost di stimare la SoS sia in forma globale sia in forma locale, analizzando quali feature risultino effettivamente informative e in quale misura esse contribuiscano alle prestazioni del regressore. Dall'altro lato, confrontare le stime ottenute con quelle fornite dal metodo SNAC, e — nel caso della stima locale — anche con una rete neurale addestrata sul medesimo dataset, al fine di verificare l'accuratezza e la potenziale applicabilità dei diversi approcci in scenari sperimentali e, in prospettiva, clinici.

In una prima parte, l'analisi si concentra sulla stima globale della SoS ottenuta tramite il modello XGBoost, descrivendo il processo di addestramento, la selezione delle feature e le prestazioni conseguite sui diversi insiemi di dati. A questa valutazione segue un confronto diretto tra la stima globale fornita dal modello e quella derivante dal metodo fisico di riferimento, il metodo SNAC. Successivamente, l'attenzione si sposta sulla stima locale della SoS, per la quale XGBoost è stato adattato per operare a livello di singole regioni dell'immagine. Tale approccio viene poi confrontato sia con il metodo SNAC, sia con l'operatore neurale di Fourier sviluppato per lo stesso compito, con l'obiettivo di valutare eventuali differenze, complementarità e potenziali vantaggi.

4.3.1 Modello XGBoost per la stima globale della velocità del suono

Dataset e spazio delle feature

Per la valutazione quantitativa del modello XGBoost basato su feature di tessitura è stato utilizzato l'intero dataset disponibile, costituito da 140 immagini simulate e 2 immagini acquisite in vitro su due fantocci a geometria nota. In tutte le analisi è stata considerata esclusivamente la fase pre-beamforming del segnale ecografico. Il dataset complessivo è stato suddiviso in un *training-validation set* e in due *test set* distinti.

- Il training-validation set comprende 119 immagini, utilizzate sia per l'addestramento sia per la validazione interna del modello nell'ambito della procedura di cross-validation descritta nella Sezione 3.5.1.
- Il primo test set, denominato simulato, è costituito unicamente da immagini simulate (21 immagini) ed è impiegato per la valutazione delle prestazioni del modello in condizioni controllate.
- Il secondo test set, denominato sperimentale, include invece le due immagini acquisite in vitro sui fantocci, ed è utilizzato per verificare la capacità di generalizzazione del modello su dati sperimentali reali.

Per ciascuna immagine sono state estratte 93 feature di tessitura radiomiche, ottenute a partire dalle matrici e dai descrittori menzionati nel capitolo metodologico. Lo spazio delle feature su cui opera il modello XGBoost risulta quindi di dimensione 93, comune a tutte le immagini appartenenti ai tre insiemi (training-validation, test simulato e test in vitro).

Tuning iperparametri

Il tuning degli iperparametri del modello XGBoost è stato effettuato mediante una procedura di *nested cross-validation*. In breve, per ciascun fold di validazione esterna, è stata eseguita una validazione interna a più fold, all'interno della quale sono state esplorate diverse combinazioni di iperparametri. Per ogni combinazione, è stato addestrato un modello sul training set interno e ne è stata valutata la prestazione sul relativo validation set, utilizzando come metrica di ottimizzazione l'errore assoluto medio (MAE). Per ciascun fold esterno è stato quindi selezionato il set di iperparametri che minimizzava il MAE di validazione interna; infine, è stata scelta la combinazione con la migliore prestazione media sui diversi fold esterni. La Tabella 4.2 riporta gli iperparametri ottimali individuati per il modello XGBoost proposto.

Training finale e selezione delle feature

Una volta individuata, mediante procedura di nested cross-validation, la combinazione ottimale degli iperparametri del modello XGBoost, è stato eseguito l'addestramento finale utilizzando esclusivamente il training set.

Iperparametro	Valore ottimale
Profondità massima degli alberi (<code>max_depth</code>)	2
Tasso di apprendimento (<code>learning_rate</code>)	0,05
Gamma (<code>gamma</code>)	10
Regolarizzazione L2 (<code>reg_lambda</code>)	10
Regolarizzazione L1 (<code>reg_alpha</code>)	0
Soglia di correlazione (<code>soglia_corr</code>)	0,9
Numero di alberi (<code>num_estimators</code>)	600
Frazione di campioni per albero (<code>sub_sample</code>)	0,8
Frazione di feature per albero (<code>subcol_tree</code>)	0,8

Tabella 4.2: Iperparametri ottimali del modello XGBoost ottenuti tramite nested cross-validation.

Prima della fase di addestramento è stata effettuata la selezione delle feature sul solo insieme di training, al fine di ridurre la dimensionalità del problema e limitare il rischio di overfitting. In particolare, sono state applicate due procedure sequenziali:

- *Scrematura per bassa varianza*: sono state eliminate le feature con varianza inferiore a 10^{-2} .
- *Scrematura per alta correlazione*: tra le feature rimanenti, sono state rimosse le variabili altamente correlate, utilizzando una soglia di correlazione pari a 0,9; in ciascuna coppia (o gruppo) di feature con correlazione sopra la soglia è stata mantenuta la variabile con maggiore rilevanza secondo la *feature importance* di XGBoost.

Lo spazio delle feature iniziale, costituito da 93 descrittori radiomici di tessitura, è stato così ridotto a un sottoinsieme di 19 feature, che rappresentano le variabili effettivamente utilizzate per l'addestramento finale del modello. La Tabella 4.3 riporta l'elenco completo delle feature selezionate.

Feature selezionata	Importanza
Root Mean Squared	0,2846
Run Entropy	0,2634
Mean Absolute Deviation	0,1598
Skewness	0,0972
Dependence Entropy	0,0341
Gray Level Non Uniformity	0,0270
Size Zone Non Uniformity	0,0203
Small Area Emphasis	0,0182
Gray Level Non Uniformity Normalized	0,0132
Mean	0,0124
Autocorrelation	0,0114
Joint Average	0,0101
Zone Entropy	0,0093
Large Area Emphasis	0,0075
Run Length Non Uniformity Normalized	0,0068
Zone Percentage	0,0065
Small Area Low Gray Level Emphasis	0,0062
Cluster Prominence	0,0060
Small Area High Gray Level Emphasis	0,0059

Tabella 4.3: Importanza delle 19 feature radiomiche di tessitura selezionate dopo scrematura per varianza e correlazione.

Prestazioni del modello XGBoost nella stima globale della SoS

Per valutare le prestazioni del modello XGBoost sono stati considerati il training set, il validation set e il test set simulato, riportando per ciascuno il valore medio della *Mean Signed Deviation (MSD)* accompagnato dalla relativa deviazione standard e dal coefficiente di determinazione R^2 , come sintetizzato in Tabella 4.4. I risultati mostrano valori di R^2 più elevati sul training set rispetto al validation set e al test simulato, con una progressiva riduzione delle prestazioni man mano che ci si allontana dal dominio di addestramento. Per quanto riguarda il test set sperimentale, costituito dalle

Dataset	MSD \pm Dev. Std	R^2
Training set	0,1 \pm 4,4	0,98
Validation set	-7,5 \pm 37,2	0,85
Test set simulato	-21,1 \pm 41,3	0,80

Tabella 4.4: Confronto delle prestazioni del modello tra i diversi set: training, validation e test simulato.

immagini acquisite sui due fantocci, le prestazioni del modello sono state descritte confrontando direttamente la velocità del suono reale con quella stimata da XGBoost per

ciascun campione, riportato nella Tabella 4.5. Questo confronto consente di quantificare l'accuratezza dello stimatore nel dominio sperimentale e di evidenziare l'eventuale scostamento tra i valori reali e quelli predetti.

Campione	SoS reale [m/s]	SoS predetto [m/s]
3B Scientific	1540	1413,2
Fantoccio in PDMS/agarosio	1250	1384,3

Tabella 4.5: Prestazioni del modello sul test set sperimentale. Confronto tra la velocità del suono reale e quella stimata dal modello XGBoost per i due fantocci sperimentali.

Discussione dei risultati

L'analisi della feature importance riportata in Tabella 4.3 mostra che le 19 feature selezionate appartengono a gruppi radiomici eterogenei, indicando che la stima della velocità del suono beneficia sia di descrittori di intensità globale sia di misure di tessitura di ordine superiore. Tra le feature più rilevanti compaiono diversi descrittori *first order* (*Root Mean Squared*, *Mean Absolute Deviation*, *Skewness*, *Mean*), che sintetizzano la distribuzione statistica dei livelli di grigio e suggeriscono come l'informazione sul contenuto medio e sulla dispersione delle intensità sia ancora discriminante per il modello. Accanto a queste, rivestono un ruolo importante le feature derivate da matrici di tessitura, in particolare dai gruppi *GLRLM* (ad esempio *Run Entropy* e *Run Length Non Uniformity Normalized*) e *GLSZM* (*Size Zone Non Uniformity*, *Zone Entropy*, *Zone Percentage*, *Small Area Emphasis*, *Large Area Emphasis*, *Small Area Low/High Gray Level Emphasis*), che caratterizzano la dimensione, l'eterogeneità e la complessità delle regioni strutturalmente omogenee nell'immagine. Per quanto riguarda il gruppo *GLCM*, studiato in dettaglio nel capitolo precedente, nel modello finale risultano selezionate in particolare le feature *Autocorrelation* e *Joint Average*, mentre le altre tre feature *GLCM* individuate nello studio preliminare (*IDN*, *IDMN* e *SA*) sono state eliminate nella fase di scrematura a causa dell'elevata correlazione con altre variabili. Inoltre, le due feature *GLCM* mantenute presentano un'importanza non predominante rispetto ad altri descrittori, a conferma del fatto che, pur mostrando una chiara dipendenza dalla SoS nelle analisi condotte sulle diverse fasi della catena di elaborazione, esse non esauriscono l'informazione utile ai fini della stima. Questo risultato valida a posteriori la scelta metodologica di utilizzare il classificatore XGBoost sull'intero spazio di feature radiomiche, e non limitarsi al solo sottoinsieme di feature *GLCM* precedentemente identificate.

Un'ulteriore considerazione riguarda il possibile fenomeno di overfitting del modello. Come riportato in Tabella 4.4, il coefficiente di determinazione risulta molto elevato sul training set ($R^2 = 0,98$) e si riduce progressivamente su validation set ($R^2 = 0,85$) e test set simulato ($R^2 = 0,80$), accompagnato da un aumento della deviazione standard della MSD. Questo andamento suggerisce la presenza di un moderato calo di prestazioni al passaggio dal dominio di addestramento a quello di validazione e, soprattutto, al test su dati simulati indipendenti; tuttavia, il mantenimento di valori di

R^2 ancora relativamente elevati indica che il modello conserva una capacità predittiva non trascurabile anche su dati non visti.

Il quadro cambia in maniera più marcata nel caso del test sperimentale. Dal confronto tra la velocità del suono reale e quella stimata dal modello sui due fantocci (Tabella 4.5) emerge infatti uno scostamento significativo tra valori misurati e predetti, con una sottostima della SoS per il fantoccio *3B Scientific* e una sovrastima per il fantoccio in PDMS/agarosio. Questo comportamento evidenzia come il modello, pur mostrando buone prestazioni nel dominio simulato, fatichi a generalizzare correttamente a dati sperimentali reali, verosimilmente a causa delle differenze tra le condizioni ideali simulate e le complessità presenti nelle acquisizioni in vitro. In supporto a questa interpretazione, la Figura 4.8 mostra, a titolo esemplificativo, le distribuzioni di due feature selezionate, *Root Mean Squared* e *Skewness*, confrontando i valori assunti nei fantocci con quelli del dataset simulato costituito da immagini a singolo strato: in entrambi i casi, i valori delle feature corrispondenti ai fantocci risultano fortemente distanti dal range coperto dalle immagini simulate. Sebbene in figura siano rappresentate solo queste due feature, un comportamento analogo è stato osservato per la maggior parte delle 19 feature selezionate, indicando un marcato disallineamento tra la distribuzione delle feature nei dati simulati e quella osservata nei dati sperimentali.

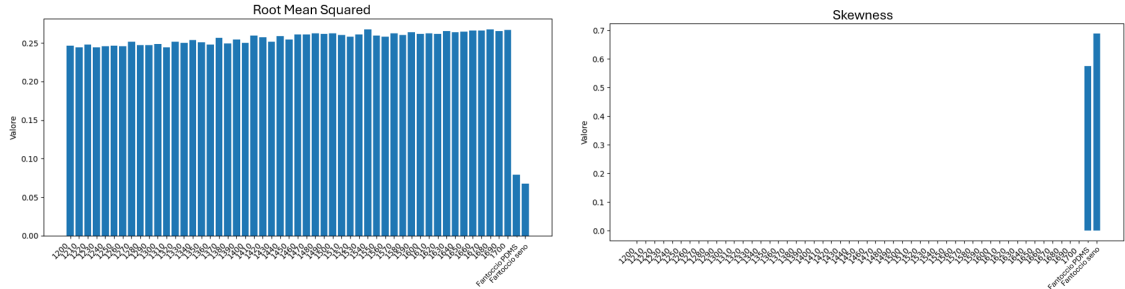


Figura 4.8: Confronto dei valori delle feature nei dati simulati e sperimentali. La figura mostra il confronto tra i valori delle feature *Root Mean Squared* (sinistra) e *Skewness* (destra) calcolati sulle immagini simulate a singolo strato (tutte le barre a sinistra) e sulle due immagini acquisite sui fantocci in vitro (ultime due barre).

4.3.2 Confronto delle prestazioni nella stima globale della SoS

In questa sezione vengono confrontate le prestazioni del modello XGBoost con quelle ottenute mediante il metodo SNAC, con l'obiettivo di valutarne l'accuratezza relativa in scenari differenti e di identificare in quali condizioni ciascun approccio risulti più affidabile. Il confronto sul test set simulato viene condotto utilizzando le stesse metriche adottate nella fase precedente, ovvero la *Mean Signed Deviation* (MSD) accompagnata dalla relativa deviazione standard e il coefficiente di determinazione R^2 . Le prestazioni vengono analizzate sia sull'intero insieme di test sia sui tre sottoinsiemi che comprendono immagini a singolo strato, a doppio strato verticale e a doppio strato orizzontale,

così da valutare l'influenza della complessità strutturale del mezzo sul comportamento dei due metodi. I risultati ottenuti per il test set di immagini simulate sono riassunti in Tabella 4.6, nella quale, per ciascun sottoinsieme, sono riportati i valori di $MSD \pm STD$ e di R^2 per XGBoost e SNAC.

Sottoinsieme	MSD \pm STD [m/s]		R^2	
	XGBoost	SNAC	XGBoost	SNAC
Intero dataset	-21,1 \pm 42,3*	-78,7 \pm 96,4	0,80	-0,42
Unico strato	-8,4 \pm 36,9*	-7,1 \pm 1,6	0,95	0,99
Due strati orizzontali	-32,1 \pm 51,7	-130,3 \pm 111,7	0,01	-7,66
Due strati verticali	-19,4 \pm 37,3	-81,0 \pm 88,0	0,90	0,14

Tabella 4.6: Confronto delle prestazioni dei metodi XGBoost e SNAC sul test set di immagini simulate.

*Differenza statisticamente significativa tra XGBoost e SNAC secondo test statistico.

Successivamente, l'analisi si sposta sul test sperimentale, dove per entrambi gli approcci viene confrontata direttamente la velocità del suono stimata con il corrispondente valore reale dei fantocci. In questo caso, la Tabella 4.7 riporta i valori di SoS reale e le corrispondenti stime fornite da XGBoost e SNAC per il fantoccio 3B Scientific e per il fantoccio in PDMS/agarosio, mentre la Figura 4.9 mostra, per ciascuna combinazione metodo-fantoccio, l'errore di stima definito come differenza tra SoS predetta e SoS reale. Queste rappresentazioni consentono di quantificare lo scostamento associato ai due metodi nel dominio in vitro, lasciando alla successiva sezione di discussione l'interpretazione dettagliata dei risultati.

Fantoccio	SoS reale	SoS predetta	SoS predetta
	[m/s]	XGBoost [m/s]	SNAC [m/s]
3B Scientific	1540,0	1413,2	1567,2
PDMS/agarosio	1250,0	1384,3	1273,5

Tabella 4.7: Confronto tra la velocità del suono reale e quella stimata dai metodi XGBoost e SNAC sul test set sperimentale.

Discussione dei risultati

L'analisi dei risultati ottenuti sul test set simulato evidenzia differenze significative tra il comportamento del modello XGBoost e quello del metodo SNAC, soprattutto al variare della complessità strutturale delle immagini. Prima di procedere al confronto statistico, gli errori dei due metodi sono stati sottoposti a un test di normalità, il quale ha evidenziato una distribuzione non normale per l'intero dataset, mentre i tre sottoinsiemi presentano errori compatibili con la normalità. Sulla base di questi risultati, per l'intero

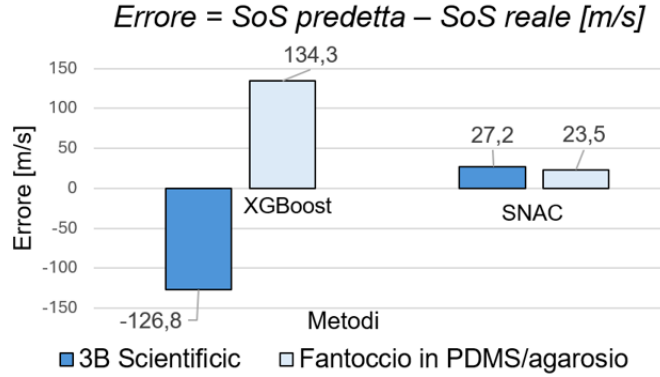


Figura 4.9: Errori di stima della velocità del suono sui fantocci sperimentali. L'errore è definito come differenza tra la SoS predetta e quella reale per ciascun metodo (XGBoost e SNAC) e per ciascun fantoccio.

dataset è stato applicato il test non parametrico per dati appaiati (*Wilcoxon signed-rank*), mentre per i tre sottoinsiemi è stato utilizzato il test parametrico appaiato (*paired Student's t-test*).

I test statistici mostrano che le prestazioni di XGBoost e SNAC risultano significativamente diverse nell'intero dataset e nel sottoinsieme a singolo strato, mentre non emergono differenze statisticamente rilevanti nei casi più complessi, ovvero nei sottoinsiemi a doppio strato orizzontale e a doppio strato verticale.

Considerando l'intero dataset, XGBoost mostra prestazioni complessivamente superiori, con una MSD più contenuta e un coefficiente di determinazione R^2 pari a 0,80, mentre SNAC presenta sia un errore medio più elevato sia un valore negativo di R^2 , indicativo di una scarsa capacità di adattamento al modello di riferimento.

Un comportamento differente emerge nel caso delle immagini a singolo strato, dove i test statistici confermano una differenza significativa tra i due metodi: in questo scenario semplice e coerente con le ipotesi fisiche del metodo, SNAC raggiunge prestazioni molto elevate, con una MSD particolarmente ridotta e un coefficiente di determinazione prossimo all'unità. XGBoost, pur mantenendo valori positivi e accettabili di MSD e R^2 , risulta meno accurato rispetto a SNAC.

La situazione cambia nei sottoinsiemi più complessi, ovvero nelle immagini a doppio strato orizzontale e a doppio strato verticale. In questi casi, i test statistici non evidenziano differenze significative tra i due metodi: le prestazioni di entrambi risultano influenzate dalla presenza di disomogeneità acustiche e variazioni strutturali interne. In particolare, SNAC mostra un peggioramento evidente, con errori medi elevati e valori di R^2 fortemente negativi nel caso degli strati orizzontali. XGBoost, pur degradando rispetto agli scenari più semplici, mantiene valori di MSD più contenuti e un grado di correlazione con la SoS reale nettamente superiore, specialmente nel sottoinsieme a strati verticali, dove R^2 rimane pari a 0,90.

Nel complesso, questi risultati mostrano come i due metodi si comportino in modo

complementare: SNAC eccelle in condizioni ideali o quasi uniformi, mentre XGBoost dimostra una maggiore robustezza e capacità di generalizzazione al crescere della complessità strutturale del mezzo simulato. La presenza di differenze statisticamente significative solo negli scenari meno complessi costituisce un elemento fondamentale per l'interpretazione complessiva delle prestazioni dei due approcci e per la valutazione della loro potenziale applicabilità in contesti più realistici.

La valutazione sul test set sperimentale mette in evidenza un comportamento dei due metodi differente rispetto a quanto osservato nel dominio simulato. Come riportato nella Tabella 4.7, per il fantoccio 3B Scientific il modello XGBoost sottostima marcatamente la velocità del suono, mentre per il fantoccio in PDMS/Agarosio la sovrastima in modo altrettanto rilevante. Questo è ben visibile anche in Figura 4.9, dove gli errori di stima (SoS predetta meno SoS reale) assumono valori di grande ampiezza e di segno opposto per i due fantocchi, indicando una difficoltà del regressore nell'adattarsi in modo coerente alle diverse condizioni sperimentali. Il metodo SNAC, al contrario, mostra errori di entità più contenuta e di segno coerente nei due casi, con una leggera sovrastima della SoS sia per il fantoccio 3B Scientific sia per quello in PDMS/Agarosio. Pur trattandosi di un numero limitato di campioni, questi risultati suggeriscono che, nel dominio in vitro considerato, il metodo fisico SNAC fornisca stime globali della SoS più accurate e stabili rispetto al modello di regressione XGBoost, il quale risente dello scostamento tra le caratteristiche del dataset simulato utilizzato per l'addestramento e quelle delle acquisizioni sperimentali reali.

4.3.3 Modello XGBoost per la stima locale della velocità del suono

Dataset e spazio delle feature

Per la valutazione quantitativa del modello XGBoost nella stima locale della velocità del suono, è stato definito un dataset costituito da campioni ottenuti mediante una suddivisione verticale delle immagini ecografiche. In questo contesto, per stima locale si intende la ricostruzione della SoS su patch verticali costituite da due colonne e tutte le righe dell'immagine, così da preservare la risoluzione assiale e catturare le variazioni locali del contenuto strutturale lungo la direzione laterale. Le patch estratte non presentano alcuna sovrapposizione, assicurando l'indipendenza dei campioni generati da ciascuna immagine.

Il dataset complessivo è costituito da 1120 patch simulate e 16 patch sperimentali. Analogamente alla procedura adottata per la stima globale della SoS, la suddivisione del dataset è stata effettuata per immagine, evitando rigorosamente che patch appartenenti alla stessa immagine venissero suddivise tra training, validation e test set, così da prevenire fenomeni di *data leakage*. Il *training-validation set* comprende complessivamente 952 patch simulate, impiegate sia per l'addestramento del modello sia per la validazione interna nell'ambito della procedura di ottimizzazione degli iperparametri. Il *test set simulato* è composto da 168 patch, mentre il *test set sperimentale* include 16 patch estratte dalle immagini acquisite in vitro. Come già descritto per il caso della

stima globale, tale scelta consente di valutare sia le prestazioni del modello in condizioni controllate sia la sua capacità di generalizzare su dati sperimentali reali. Per ciascuna patch locale sono state calcolate 93 feature radiomiche di tessitura, derivate dagli stessi gruppi descrittivi impiegati nella stima globale della SoS e descritte nel capitolo metodologico. In questo caso, tuttavia, tali feature sono state estratte localmente, ovvero applicando i descrittori sull'intera estensione verticale della patch, così da caratterizzare le variazioni del contenuto strutturale su scala laterale.

Lo spazio delle feature iniziale risulta quindi costituito da 93 variabili per ciascuna delle 1120 patch simulate e 16 patch sperimentali, generando un dataset numericamente consistente e adeguato all'addestramento di un modello di regressione basato su tecniche di apprendimento supervisionato. Le feature sono state successivamente impiegate nelle fasi di selezione e addestramento del modello XGBoost, come descritto nei paragrafi seguenti.

Tuning iperparametri

Il tuning degli iperparametri del modello XGBoost per la stima locale della velocità del suono è stato effettuato mediante la procedura di *5-fold cross-validation*. In particolare, il dataset è stato suddiviso in cinque fold e, a rotazione, quattro fold sono stati utilizzati per l'addestramento e uno per la validazione. Per ogni combinazione di iperparametri è stato addestrato un modello sul training set e ne è stata valutata la prestazione sul relativo validation set, utilizzando come metrica di ottimizzazione l'errore assoluto medio (MAE) calcolato sui valori locali di SoS stimati a livello di patch.

Al termine della cross-validation, è stato selezionato il set di iperparametri che minimizzava il MAE medio sui cinque fold. Tale configurazione è stata quindi assunta come ottimale per il modello XGBoost nella stima locale della SoS. La Tabella 4.8 riporta in forma sintetica gli iperparametri ottimali individuati per il modello XGBoost proposto nella stima locale della SoS.

Iperparametro	Valore ottimale
Profondità massima degli alberi (<code>max_depth</code>)	4
Tasso di apprendimento (<code>learning_rate</code>)	0,05
Gamma (<code>gamma</code>)	0
Regolarizzazione L2 (<code>reg_lambda</code>)	20
Regolarizzazione L1 (<code>reg_alpha</code>)	1
Soglia di correlazione (<code>soglia_corr</code>)	0,95
Soglia di varianza (<code>var_threshold</code>)	0,01
Numero di alberi (<code>n_estimators</code>)	1000
Frazione di campioni per albero (<code>subsample</code>)	0,8
Frazione di feature per albero (<code>colsample_bytree</code>)	1,0

Tabella 4.8: Iperparametri ottimali del modello XGBoost per la stima locale della SoS ottenuti tramite 5-fold cross validation.

Training finale e selezione delle feature

Una volta individuata, mediante procedura di 5-fold cross-validation, la combinazione ottimale degli iperparametri del modello XGBoost per la stima locale della velocità del suono, è stato eseguito l'addestramento finale utilizzando l'intero training-validation set.

Prima della fase di addestramento, è stata effettuata la selezione delle feature sul solo insieme di training-validation, al fine di ridurre la dimensionalità del problema e limitare il rischio di overfitting. In particolare, sono state applicate due procedure sequenziali:

- *Scrematura per bassa varianza*: sono state eliminate le feature con varianza inferiore a 10^{-2} .
- *Scrematura per alta correlazione*: tra le feature rimanenti, sono state rimosse le variabili altamente correlate, utilizzando una soglia di correlazione pari a 0.95; in ciascuna coppia (o gruppo) di feature con correlazione superiore alla soglia è stata mantenuta la variabile con maggiore rilevanza secondo la *feature importance* di XGBoost calcolata sul training-validation set.

Lo spazio delle feature iniziale, costituito da 93 descrittori radiomici di tessitura calcolati localmente sulle patch, è stato così ridotto a un sottinsieme di 49 feature, che rappresentano le variabili effettivamente utilizzate per l'addestramento finale del modello. Le importanze assegnate dal modello XGBoost alle feature selezionate sono state analizzate per identificare i descrittori maggiormente discriminanti nella stima locale della SoS. Poiché il numero totale di variabili selezionate è pari a 49, in Tabella 4.9 sono riportate, a titolo esemplificativo, le 20 feature con importanza maggiore, sebbene tutte le 49 feature siano state impiegate per l'addestramento finale del modello.

Prestazioni del modello XGBoost nella stima locale della SoS

Per valutare le prestazioni del modello XGBoost nella stima locale della velocità del suono sono stati considerati il training-validation set, il test set simulato e il test set sperimentale. Per ciascun insieme di dati è stato calcolato il valore medio della *Mean Signed Deviation* (MSD), accompagnato dalla relativa deviazione standard, e il coefficiente di determinazione R^2 , entrambi valutati sui campioni locali (patch) che compongono il dataset. I risultati sono sintetizzati in Tabella 4.10.

Nel caso del training-validation set, la MSD e il coefficiente R^2 forniscono una misura della capacità del modello di adattarsi ai dati utilizzati per l'addestramento, mentre le prestazioni sul test set simulato consentono di valutare la capacità di generalizzazione del regressore su immagini non viste ma appartenenti allo stesso dominio simulato.

Per quanto riguarda il test sperimentale, ciascuna immagine ecografica dei due fantocci è stata suddivisa in 8 patch verticali, all'interno delle quali è stata stimata localmente la velocità del suono mediante il modello XGBoost. Poiché la SoS reale è uniforme all'interno di ciascun fantoccio, tutte le patch associate alla stessa immagine condividono lo stesso valore di riferimento, riportato in Tabella 4.11 nella colonna *SoS*

Feature selezionata	Importanza
Energy	0,4800
Gray Level Non Uniformity	0,1156
Mean Absolute Deviation	0,0898
Robust Mean Absolute Deviation	0,0680
Large Dependence Emphasis	0,0265
Long Run Emphasis	0,0230
Kurtosis	0,0203
Difference Entropy	0,0189
Imc2	0,0146
Large Dependence High Gray Level Emphasis	0,0138
Sum Entropy	0,0110
Root Mean Squared	0,0105
Gray Level Non Uniformity	0,0104
Dependence Non Uniformity	0,0095
Large Dependence Low Gray Level Emphasis	0,0091
Complexity	0,0082
Run Entropy	0,0065
Minimum	0,0059
Short Run High Gray Level Emphasis	0,0054
Run Variance	0,0050

Tabella 4.9: Importanza delle 20 feature radiomiche di tessitura più rilevanti per il modello XGBoost nella stima locale della velocità del suono.

Dataset	MSD \pm Dev. Std	R ²
Training-validation set	0,01 \pm 2,08	0,99
Test set simulato	-1,20 \pm 46,61	0,88

Tabella 4.10: Confronto delle prestazioni del modello XGBoost nella stima locale della SoS tra i diversi set: training-validation e test simulato.

reale. A partire dalle stime locali ottenute sulle singole patch, per ogni fantoccio è stato quindi calcolato il valore medio della SoS predetta e la relativa deviazione standard, così da ottenere una misura sintetica dell'accuratezza locale del modello e della variabilità intra-fantoccio. La Tabella 4.11 riassume tale confronto tra la velocità del suono reale e quella stimata dal modello XGBoost nel caso sperimentale.

Discussione dei risultati

L'analisi della feature importance riportata in Tabella 4.9 mostra che le 20 feature più rilevanti per la stima locale della velocità del suono appartengono a gruppi radiomici eterogenei, indicando come il modello faccia uso congiunto di descrittori di intensità e

Fantoccio	SoS reale [m/s]	SoS predetta locale [m/s]
3B Scientific	1540	1420,7 \pm 6,1
PDMS/agarosio	1250	1377,4 \pm 4,9

Tabella 4.11: Confronto tra la velocità del suono reale e quella stimata dal modello XGBoost nella stima locale della SoS sul test set sperimentale. Per ciascun fantoccio, la SoS predetta locale è riportata come media \pm deviazione standard delle stime ottenute sulle 8 patch che compongono l'immagine.

di tessitura di ordine superiore. In particolare, tra le feature con importanza maggiore si osserva una forte presenza di descrittori *first order* (*Energy*, *Mean Absolute Deviation*, *Robust Mean Absolute Deviation*, *Kurtosis*, *Root Mean Squared*, *Minimum*), che sintetizzano le proprietà statistiche locali dei livelli di grigio all'interno della patch e suggeriscono come l'informazione sul contenuto energetico e sulla dispersione delle intensità sia ancora altamente discriminante per il modello anche nella stima locale. Accanto a queste, rivestono un ruolo importante le feature derivate da matrici di tessitura. Tra i descrittori associati ai gruppi GLDM e GLSZM compaiono, ad esempio, *Gray Level Non Uniformity*, *Dependence Non Uniformity*, *Large Dependence Emphasis*, *Large Dependence High Gray Level Emphasis* e *Large Dependence Low Gray Level Emphasis*, che caratterizzano la dimensione, l'eterogeneità e la complessità delle regioni strutturalmente omogenee nella direzione laterale. Dal gruppo GLCM provengono invece *Difference Entropy*, *Sum Entropy* e *Imc2*, che descrivono rispettivamente il disordine locale e il grado di correlazione tra coppie di pixel adiacenti. Infine, i descrittori GLRLM e NGTDM (*Long Run Emphasis*, *Short Run High Gray Level Emphasis*, *Run Entropy*, *Run Variance*, *Complexity*) catturano la lunghezza e la distribuzione delle *runs* di livelli di grigio, fornendo una misura della granularità e della complessità del pattern tessiturale lungo la patch. Nel complesso, la combinazione di questi gruppi di feature conferma che la stima locale della SoS beneficia della descrizione congiunta delle caratteristiche di intensità e delle proprietà tessiturali di ordine superiore dell'immagine. Un'ulteriore considerazione riguarda il possibile fenomeno di overfitting del modello nella stima locale della velocità del suono. Come riportato in Tabella 4.10, il coefficiente di determinazione risulta molto elevato sul training-validation set ($R^2 = 0,99$) e si riduce sul test set simulato ($R^2 = 0,88$), accompagnato da un marcato aumento della deviazione standard della MSD (da 2,08 a 46,61 m/s). Questo andamento suggerisce la presenza di un moderato calo di prestazioni al passaggio dal dominio di addestramento a quello di test su dati simulati indipendenti, verosimilmente legato alla maggiore variabilità locale introdotta dal campionamento per patch e dalla complessità strutturale delle immagini non viste. Tuttavia, il mantenimento di valori di R^2 ancora relativamente elevati indica che il modello conserva una capacità predittiva non trascurabile anche su dati locali indipendenti, pur manifestando un incremento della dispersione dell'errore rispetto al training-validation set.

Un comportamento analogo, ma ancora più marcato, si osserva nel caso sperimentale anche per la stima locale della velocità del suono. Come riportato in Tabella 4.13, la

stima media locale della SoS ottenuta dal modello XGBoost presenta uno scostamento evidente rispetto ai valori reali per entrambi i fantocci, nonostante la ridotta deviazione standard all'interno delle patch indichi una buona coerenza interna delle predizioni. In particolare, per il fantoccio 3B Scientific si osserva una sottostima sistematica, mentre per il fantoccio in PDMS/agarosio si evidenzia una sovrastima delle velocità locali. Questo comportamento è coerente con quanto già discusso per la stima globale della SoS: la discrepanza tra le predizioni del modello sui dati simulati e su quelli sperimentali deriva principalmente dall'elevata differenza nella distribuzione delle feature radiomiche tra i due domini. Le feature estratte dalle immagini in vitro risultano infatti collocate in regioni di valore significativamente diverse rispetto a quelle osservate sulle simulazioni, come precedentemente evidenziato. Tale disallineamento implica che il modello, pur avendo appreso correttamente la relazione tra tessitura e velocità del suono all'interno del dominio simulato, fatichi a generalizzare quando applicato a dati reali caratterizzati da condizioni acustiche, livelli di rumore, eterogeneità e proprietà tessiturali differenti. Nel contesto della stima locale, questa problematica risulta ulteriormente amplificata poiché il modello opera su porzioni estremamente ristrette dell'immagine, dove le variazioni strutturali tipiche del dominio sperimentale possono produrre valori delle feature ancora più distanti da quelli presenti nel dataset di addestramento. Di conseguenza, anche in presenza di una buona consistenza locale delle predizioni, la differenza sistematica in termini di SoS predetta riflette principalmente l'inadeguato allineamento statistico tra feature simulate e sperimentali.

4.3.4 Confronto delle prestazioni nella stima locale della SoS

In questa sezione vengono confrontate le prestazioni dei modelli XGBoost, SNAC e della rete neurale nella stima *locale* della velocità del suono, utilizzando gli stessi dataset di test impiegati nella valutazione del modello di regressione, ovvero il test set simulato e il test set sperimentale. L'obiettivo è valutare l'accuratezza dei tre approcci sia in condizioni controllate, sia su immagini acquisite in vitro, analizzando le metriche globali di errore.

Il confronto quantitativo è condotto inizialmente sul test set simulato utilizzando le metriche già introdotte nel capitolo precedente, ovvero la *Mean Signed Deviation (MSD)* con la relativa deviazione standard e il coefficiente di determinazione R^2 calcolati a livello di patch. Le prestazioni dei tre metodi vengono analizzate sia sull'intero dataset sia suddividendo il test set in sottosezioni caratterizzate da differente complessità strutturale dell'immagine: immagini a singolo strato, doppio strato verticale o doppio strato orizzontale. I risultati sono riportati in Tabella 4.12, che consente di confrontare direttamente l'accuratezza dei tre modelli nelle diverse condizioni simulate.

Successivamente, l'analisi viene estesa al dominio sperimentale confrontando, per ciascun fantoccio, il valore reale della velocità del suono con le corrispondenti stime locali ottenute dai tre metodi. Le immagini sperimentali sono state suddivise in 8 patch verticali e, per ogni fantoccio, la SoS predetta è stata calcolata come media delle stime locali sulle patch. La Tabella 4.13 riporta quindi i valori di SoS reale e le

Sottoinsieme	MSD \pm STD [m/s]			R^2		
	XGBoost	SNAC	OFN	XGBoost	SNAC	OFN
Intero dataset	$-1,2 \pm 46,6$	$-20,1 \pm 53,6^*$	$-4,9 \pm 64,7$	0,88	0,78	0,77
Unico strato	$10,0 \pm 17,1$	$4,2 \pm 1,8^*$	$2,6 \pm 20,5^\dagger$	0,98	0,99	0,98
Due strati orizzontali	$-11,3 \pm 67,0$	$55,3 \pm 69,9$	$-23,0 \pm 74,4^\dagger$	0,17	-1,3	-0,08
Due strati verticali	$-5,5 \pm 37,7$	$-8,4 \pm 4,7$	$5,44 \pm 74,1$	0,94	0,97	0,77

Tabella 4.12: Confronto delle prestazioni dei metodi XGBoost, SNAC e OFN sul test set di immagini simulate.

*Differenza statisticamente significativa tra XGBoost e SNAC secondo test statistico.

† Differenza statisticamente significativa tra XGBoost e OFN secondo test statistico.

corrispondenti stime locali dei metodi XGBoost, SNAC e della rete neurale espresse come media \pm deviazione standard, così da quantificare non solo l'errore medio, ma anche la variabilità intra-fantoccio delle predizioni.

Nel complesso, questo confronto quantitativo permette di identificare i punti di forza e i limiti dei tre approcci nella stima locale della SoS, fornendo indicazioni utili sulla loro robustezza e generalizzabilità.

Fantoccio	SoS reale	SoS predetta	SoS predetta	SoS predetta
	[m/s]	XGBoost [m/s]	SNAC [m/s]	OFN [m/s]
3B Scientific	1540,0	$1420,7 \pm 6,1$	$1563,4 \pm 2,7$	$1503,3 \pm 4,0$
PDMS/agarosio	1250,0	$1377,4 \pm 4,9$	$1271,7 \pm 3,5$	$1492,1 \pm 10,5$

Tabella 4.13: Confronto tra la velocità del suono reale e quella stimata dai metodi XGBoost, SNAC e OFN sul test set sperimentale.

Discussione dei risultati

L'analisi dei risultati locali riportati in Tabella 4.12 è stata preceduta da una valutazione statistica delle differenze tra i metodi, condotta confrontando il modello proposto XGBoost con i due approcci di riferimento SNAC e OFN. Per ciascun confronto sono stati dapprima analizzati gli errori mediante il test di normalità di *Shapiro-Wilk*, al fine di selezionare il test statistico più appropriato. Nel confronto tra XGBoost e SNAC, gli errori risultano non normalmente distribuiti in tutti i sottoinsiemi eccetto quello a singolo strato; di conseguenza è stato applicato il test non parametrico di *Wilcoxon* per l'intero dataset, per il doppio strato orizzontale e per il doppio strato verticale, mentre per il sottoinsieme a singolo strato è stato adottato il *paired Student's t-test*. Nel confronto tra XGBoost e OFN, invece, la normalità è risultata verificata soltanto per il sottoinsieme a singolo strato, per il quale è stato quindi impiegato un *paired t-test*, mentre per gli altri tre sottoinsiemi è stato applicato il test di *Wilcoxon*.

I risultati mostrano che XGBoost differisce in modo statisticamente significativo dal metodo SNAC nell'intero dataset e nel sottoinsieme a singolo strato, mentre non emergono differenze significative nei casi a doppio strato orizzontale e verticale. Nel

confronto tra XGBoost e OFN, una differenza significativa è stata riscontrata nel sottoinsieme a singolo strato, mentre gli altri tre sottoinsiemi non mostrano scostamenti statisticamente rilevanti.

L'esame dei valori di MSD e R^2 mette in luce un comportamento differenziato dei tre modelli al variare della complessità strutturale delle immagini simulate. Considerando l'intero dataset, il modello XGBoost mostra una Mean Signed Deviation moderata ($\text{MSD} = -1,2 \text{ m/s}$, con segno invertito rispetto alla stima reale) e una dispersione contenuta ($46,6 \text{ m/s}$), accompagnate dal coefficiente di determinazione più elevato ($R^2 = 0,88$). Il metodo SNAC presenta invece una marcata sottostima ($\text{MSD} = -20,1 \text{ m/s}$) e una deviazione standard leggermente superiore ($53,6 \text{ m/s}$), con un R^2 pari a 0,78. L'operatore neurale di Fourier (OFN) mostra un bias negativo più contenuto ($\text{MSD} = -4,9 \text{ m/s}$) ma una variabilità maggiore ($64,7 \text{ m/s}$) e un coefficiente di determinazione paragonabile a quello di SNAC ($R^2 = 0,77$). Nel complesso, sul totale delle immagini XGBoost fornisce stime locali della SoS più coerenti e meno distorte rispetto ai due metodi di confronto.

La scomposizione del test set nelle diverse sottosezioni permette di evidenziare più chiaramente le differenti aree di forza dei tre approcci. Nel caso delle immagini a singolo strato, SNAC evidenzia un comportamento pressoché non distorto ($\text{MSD} = 4,2 \pm 1,8 \text{ m/s}$) e un coefficiente di determinazione quasi unitario ($R^2 = 0,99$), superiore a quello ottenuto sia da XGBoost ($\text{MSD} = 10,0 \pm 17,1 \text{ m/s}$, con segno invertito rispetto alla stima reale, $R^2 = 0,98$) sia da OFN ($\text{MSD} = 2,6 \pm 20,5 \text{ m/s}$, $R^2 = 0,98$). In questo scenario semplice, pienamente coerente con le assunzioni del metodo fisico, SNAC risulta il più accurato; XGBoost e OFN raggiungono prestazioni paragonabili in termini di correlazione, ma con una maggiore dispersione delle stime.

All'aumentare della complessità strutturale, il quadro si modifica sensibilmente. Nelle immagini a doppio strato orizzontale, XGBoost mantiene una deviazione media contenuta ($\text{MSD} = -11,3 \pm 67,0 \text{ m/s}$) e un coefficiente di determinazione superiore a quello degli altri due metodi ($R^2 = 0,17$). SNAC mostra una marcata sovrastima ($\text{MSD} = 55,3 \pm 69,9 \text{ m/s}$) con R^2 negativo ($R^2 = -1,3$), mentre OFN tende a una sottostima sistematica ($\text{MSD} = -23,0 \pm 74,4 \text{ m/s}$) con un R^2 leggermente negativo ($R^2 = -0,08$). Ciò suggerisce che, in presenza di discontinuità laterali e strutture sovrapposte, XGBoost riesce ad adattarsi meglio alla variabilità locale del mezzo, mentre sia il metodo fisico sia l'operatore neurale incontrano maggiori difficoltà nel rappresentare correttamente tali configurazioni.

Nel caso delle immagini a doppio strato verticale, tutti i metodi recuperano prestazioni elevate in termini di capacità predittiva. XGBoost mostra un errore medio contenuto ($\text{MSD} = -5,5 \pm 37,7 \text{ m/s}$) e un coefficiente di determinazione pari a 0,94, mentre SNAC mantiene bias limitato ($\text{MSD} = -8,4 \pm 4,7 \text{ m/s}$) e un R^2 molto elevato (0,97). OFN presenta un leggero bias positivo ($\text{MSD} = 5,4 \pm 74,1 \text{ m/s}$), ma con una variabilità decisamente maggiore e un R^2 inferiore (0,77). In questo scenario, SNAC risulta il più accurato e stabile, XGBoost si conferma competitivo, mentre OFN tende a produrre stime più disperse, pur mantenendo una correlazione accettabile con i valori di riferimento.

Il confronto sul test sperimentale, sintetizzato in Tabella 4.13, conferma e rafforza il

quadro emerso dall'analisi simulata, evidenziando differenze significative tra i tre metodi considerati. Per entrambi i fantocci, le stime fornite da SNAC risultano complessivamente le più vicine ai valori reali della SoS, con un errore sistematico contenuto e una deviazione standard ridotta tra le patch. XGBoost tende invece a introdurre un bias più marcato, mentre l'operatore neurale di Fourier (OFN) mostra un comportamento intermedio, con accuratezza variabile a seconda del materiale.

Nel caso del fantoccio *3B Scientific*, XGBoost sottostima la velocità del suono ($1420,7 \text{ m/s}$ contro 1540 m/s), mentre SNAC fornisce una leggera sovrastima ($1563,4 \text{ m/s}$), ma di entità limitata e accompagnata da una basse dispersione intra-fantoccio. OFN, al contrario, produce una stima più prossima al valore reale rispetto a XGBoost ($1503,3 \text{ m/s}$), pur mostrando una variabilità maggiore rispetto a SNAC.

Per il fantoccio in PDMS/Agarosio, XGBoost presenta una chiara sovrastima ($1377,4 \text{ m/s}$ rispetto ai 1250 m/s reali), mentre SNAC fornisce nuovamente una stima molto accurata ($1271,7 \text{ m/s}$) con una dispersione ridotta. In questo caso, OFN fornisce la sovrastima più pronunciata ($1492,1 \text{ m/s}$), suggerendo una limitata capacità di adattamento alle proprietà acustiche del materiale.

Queste osservazioni risultano coerenti con quanto discusso in precedenza riguardo alle differenze tra dominio simulato e dominio sperimentale. Le feature radiomiche estratte dalle immagini in vitro presentano distribuzioni sensibilmente diverse rispetto a quelle utilizzate durante l'addestramento del modello XGBoost, determinando un marcato *domain shift* che ne compromette la capacità di generalizzazione. L'OFN mostra una sensibilità simile, risentendo anch'esso delle variazioni non modellate nelle caratteristiche delle immagini sperimentali.

SNAC, basandosi su un modello fisico della propagazione acustica, risulta meno influenzato da tali variazioni e mantiene prestazioni più stabili anche nel dominio reale. Nel complesso, l'analisi congiunta delle Tabelle 4.12 e 4.13 mostra quindi come XGBoost e OFN possano risultare competitivi nel dominio simulato, soprattutto in strutture complesse, mentre SNAC mantenga una maggiore robustezza e affidabilità nella stima locale della SoS quando si passa a dati sperimentali reali.

Conclusioni e sviluppi futuri

Conclusioni

In questo lavoro di tesi è stata analizzata la possibilità di stimare la velocità del suono (SoS) nei tessuti mediante analisi radiomica di tessitura applicata ai segnali pre-beamforming, con l'obiettivo di ottenere sia una stima globale sia una stima locale della SoS da integrare in futuri schemi di beamforming adattativo.

Nella prima fase è stato mostrato come i parametri di tessitura dipendano in maniera non trascurabile dalle scelte di calibrazione adottate nella Digital Scan Conversion. Il confronto tra diverse configurazioni di interpolazione ha evidenziato che sia la risoluzione assiale sia quella laterale contribuiscono alla variabilità delle feature radiomiche e dei relativi coefficienti di determinazione. Questo risultato ha motivato la scelta metodologica di spostare l'analisi dal dominio dell'immagine ricostruita al dominio pre-DSC, operando direttamente sui segnali RF. In questo modo, si è ridotta la variabilità introdotta dalle operazioni di ricostruzione, rendendo più robusto e interpretabile il confronto tra i diversi stimatori di SoS basati su grandezze pre-beamforming (SNAC, OFN e XGBoost).

La seconda fase ha evidenziato che, in ambito locale, la relazione tra feature GLCM e velocità del suono è fortemente dipendente dalla regione di analisi. In particolare, quando l'analisi è limitata a porzioni controllate del mezzo, cinque feature GLCM (Autocorrelation, IDN, Joint Mean, IDMN e Sum Average) hanno mostrato un andamento caratteristico e una chiara sensibilità alla variazione di SoS. Tuttavia, tale comportamento tende a diluirsi quando il calcolo viene esteso all'intera immagine, dove l'eterogeneità spaziale introduce rumore e riduce la significatività statistica. Il confronto tra mappe pre e post beamforming ha inoltre mostrato che quattro di queste cinque feature — tutte tranne l'Autocorrelazione — mantengono una buona coerenza tra le due condizioni, indicando che una parte rilevante dell'informazione tessiturale è già presente nei dati pre-beamforming e rafforzando l'idea di utilizzare un regressore che operi direttamente su tali segnali. Queste considerazioni hanno guidato la progettazione del modello di regressione.

La terza fase ha permesso di valutare l'efficacia di un modello di regressione basato su Extreme Gradient Boosting (XGBoost) per la stima globale e locale della SoS su un ampio dataset simulato e su dati sperimentali in vitro. L'analisi della feature importance ha mostrato che il modello sfrutta congiuntamente descrittori di primo ordine e feature di tessitura di ordine superiore, confermando che l'informazione utile alla stima

della SoS non è riconducibile a un singolo gruppo radiomico, ma alla combinazione di più famiglie di descrittori.

Nel dominio simulato, XGBoost ha dimostrato buone capacità predittive, con valori di R^2 elevati sia per la stima globale sia per quella locale e un errore medio contenuto, soprattutto in presenza di strutture stratificate complesse. Il confronto con il metodo SNAC ha mostrato che, considerando l'intero dataset simulato, XGBoost fornisce stime globali mediamente meno distorte rispetto al regressore fisico, mentre SNAC tende a risultare più accurato negli scenari strutturalmente semplici e caratterizzati da un'unica velocità del suono. L'applicazione dei test statistici ha confermato queste osservazioni: nella stima globale, differenze significative tra XGBoost e SNAC emergono principalmente quando si considera l'intero dataset simulato e il sottoinsieme di immagini a singolo strato, mentre nei casi eterogenei i due metodi mostrano prestazioni statisticamente sovrapponibili.

Per quanto riguarda la stima locale, il confronto è stato esteso anche all'operatore neurale di Fourier (OFN), impiegato come ulteriore termine di riferimento. L'analisi statistica, condotta previo test di normalità per selezionare opportunamente il test parametrico o non parametrico, ha evidenziato differenze significative tra XGBoost-SNAC e tra XGBoost-OFN nei soli scenari meno complessi, mentre nei sottoinsiemi a maggiore eterogeneità strutturale le prestazioni risultano statisticamente comparabili. Questi esiti rafforzano l'interpretazione qualitativa dei risultati: SNAC tende a essere molto accurato in condizioni semplici, OFN mostra prestazioni intermedie, mentre XGBoost mantiene una buona affidabilità nelle configurazioni più articolate.

Quando l'analisi è stata estesa al dominio sperimentale, le prestazioni di XGBoost hanno evidenziato un marcato degrado sia nella stima globale sia in quella locale. Le SoS predette per i due fantocci mostrano infatti bias sistematici opposti (sottostima per il fantoccio 3B Scientific e sovrastima per quello in PDMS/agarosio), nonostante la buona coerenza interna delle predizioni sulle patch. L'analisi delle distribuzioni delle feature ha messo in luce un forte disallineamento tra i valori radiomici dei dati simulati e quelli in vitro, segnalando uno shift di dominio che limita la capacità di generalizzazione del modello addestrato unicamente su simulazioni. In questo scenario, il metodo SNAC ha mantenuto prestazioni decisamente più stabili, confermando una maggiore robustezza del modello fisico in presenza di condizioni non ideali. L'operatore neurale di Fourier ha mostrato un comportamento non uniforme: buone prestazioni locali sul fantoccio 3B Scientific, ma scarsa capacità di adattamento al fantoccio in PDMS/agarosio.

Nel complesso, i risultati ottenuti dimostrano che l'analisi radiomica di tessitura applicata ai segnali pre-beamforming rappresenta un approccio promettente per la stima globale e locale della velocità del suono, in particolare nel dominio simulato e in presenza di strutture complesse. Allo stesso tempo, l'integrazione dei test statistici ha confermato che le differenze osservate tra i metodi non sono soltanto qualitative, ma supportate da un'evidenza quantitativa coerente con la natura dei dati: XGBoost risulta significativamente diverso dai metodi di riferimento solo negli scenari più semplici, mentre nelle configurazioni ad alta complessità strutturale le prestazioni dei diversi approcci risultano statisticamente comparabili. Infine, l'analisi evidenzia la necessità

di affrontare in modo sistematico il problema della trasferibilità tra domini simulati e reali e suggerisce che l'integrazione di modelli di apprendimento automatico con approcci fisici espliciti, come SNAC, insieme all'esplorazione preliminare di operatori neurali come OFN per la stima locale, possa costituire una strategia efficace per future metodologie di imaging ecografico adattativo.

Limitazioni

Il lavoro svolto presenta alcune limitazioni che è importante esplicitare.

In primo luogo, il dataset utilizzato per l'addestramento e la validazione del modello XGBoost è sbilanciato a favore delle simulazioni, mentre il numero di immagini in vitro è limitato a due fantocci, con conseguente ridotta rappresentatività del dominio sperimentale.

Inoltre, l'analisi ha messo in evidenza una marcata differenza nella tessitura tra le immagini simulate e quelle acquisite in vitro, che si traduce in distribuzioni delle feature radiomiche poco sovrapponibili e in un significativo shift di dominio.

Un ulteriore limite è rappresentato dall'assenza di una ground truth locale per la SoS nei dati reali: le valutazioni delle prestazioni dei modelli sui fantocci si basano su valori medi noti del mezzo, senza una mappa di riferimento ad alta risoluzione spaziale. Infine, la potenziale trasferibilità clinica dell'approccio proposto non è stata ancora dimostrata, poiché lo studio è stato condotto esclusivamente su fantocci e simulazioni e non sono stati considerati dati provenienti da pazienti.

Approcci futuri

Alla luce di queste considerazioni, diversi sviluppi futuri appaiono naturali.

Un primo passo consiste nell'ampliamento del dataset sperimentale, acquisendo un numero maggiore di immagini in vitro, includendo fantocci con geometrie e proprietà acustiche più eterogenee e, in una fase successiva, dati clinici reali.

Parallelamente, sarà importante progettare simulazioni più realistiche, in grado di ridurre il gap radiomico tra immagini simulate e in vitro, ad esempio introducendo modelli di rumore, attenuazione e complessità strutturale più vicini a quelli osservati nelle acquisizioni reali.

Un ulteriore filone di ricerca riguarda l'integrazione di tecniche di adattamento di dominio e di regolarizzazione specifiche per migliorare la robustezza dei modelli di apprendimento automatico al passaggio da simulazioni a dati sperimentali.

Infine, la valutazione della trasferibilità clinica dell'approccio richiederà test su immagini di pazienti, al fine di verificare l'impatto reale della stima globale e locale della SoS sulla qualità dell'immagine ecografica e sull'accuratezza diagnostica.

Bibliografia

- [1] Jørgen Arendt Jensen. Medical ultrasound imaging. *Progress in Biophysics and Molecular Biology*, 93:153–165, 2007. doi:10.1016/j.pbiomolbio.2006.07.009.
- [2] Marie Ploquin, Adrian Basarab, and Denis Kouamé. Resolution enhancement in medical ultrasound imaging. *Journal of Medical Imaging*, 2, Jan-Mar 2015. doi:10.1117/1.JMI.2.1.017001.
- [3] Denise Gentile, Filippo Molinari, Kristen Meiburger, and Francesco Marzola. *Sviluppo di un metodo automatico basato sul deep learning per la valutazione automatica del nervo ottico in immagini ultrasonografiche transorbitali*. PhD thesis, Politecnico di Torino, 2021. Tesi di dottorato. pp. 7, 8, 11.
- [4] Jeff Powers and Frederick Kremkau. Medical ultrasound systems. *Interface Focus*, 1:477–489, 2011. doi:10.1098/rsfs.2011.0027.
- [5] Giulia Matrone, Alessandro Ramalli, and Piero Tortoli. Ultrasound b-mode imaging: Beamforming and image formation techniques. *Applied Sciences*, 9:2533, 2019. doi:10.3390/app9122533.
- [6] Sofia De Dionigi. Studio e simulazione di sistemi di micro-beamforming per ecografia 2d/3d con sonda cmut. Tesi di laurea, Politecnico di Milano, 2014.
- [7] Aria Ashir, Saeed Jerban, Victor Barrère, Yuanshan Wu, Sameer B. Shah, Michael P. Andre, and Eric Y. Chang. Skeletal muscle assessment using quantitative ultrasound: A narrative review. *Sensors*, 23:4763, 2023. doi:10.3390/s23104763.
- [8] Kristen M. Meiburger. *Quantitative Ultrasound and Photoacoustic Imaging for the Assessment of Vascular Parameters*. PoliTO Springer Series. Springer, 1 edition, 2017. pp. 8-9.
- [9] Giuseppe Bergamin. *Analisi dei sistemi ecodoppler tecniche color e power*. Laurea triennale in ingegneria biomedica, Università di Padova, Facoltà di Ingegneria, 2012.
- [10] Alberto Campo. *Simulazione di immagini ad ultrasuoni multitessuto: confronto tra immagini cliniche e simulate a diversi stadi patologici*. PhD thesis, Politecnico di Torino, Facoltà di Ingegneria Biomedica, 2019.
- [11] Giacomo Viola. *Segmentazione e caratterizzazione automatica del nervo ottico in immagini ecografiche transorbitali per la valutazione di disturbi neurologici*. Tesi di laurea, Politecnico di Torino, Torino, ottobre 2019.

- [12] Bradley Treeby, Ben Cox, and Jiri Jaros. *k-Wave: A MATLAB toolbox for the time domain simulation of acoustic wave fields*. k-Wave team, London (UK), 1.1 edition, August 2016.
- [13] Qingyun Cao. Curriculum design of art higher vocational education based on artificial intelligence assisted virtual reality technology. *Security and Communication Networks*, pages 1–9, 2022. doi:10.1155/2022/3535068.
- [14] Vincenzo Perrot, Maxime Polichetti, François Varray, and Damien Garcia. Quindi pensi di poter usare il das? un punto di vista sul beamforming a ritardo e somma. *Ultrasonics*, 111, march 2021.
- [15] Moein Mozaffarzadeh, Masume Sadeghi, Ali Mahloojifar, and Mahdi Orooji. Double-stage delay multiply & sum beamforming algorithm applied to ultrasound medical imaging. *Ultrasound in Medicine & Biology*, 44:677–686, 2018. doi:10.1016/j.ultrasmedbio.2017.10.020.
- [16] Nicola Casali. Sviluppo e generalizzazione di un modello U-Net per il beamforming e la segmentazione simultanea di immagini ad ultrasuoni. Master’s thesis, Politecnico di Torino, 2021.
- [17] Chi Hyung Seo and Jesse T. Yen. Sidelobe suppression in ultrasound imaging using dual apodization with cross-correlation. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 55:2198–2210, 2008.
- [18] Drake A. Guenther and William F. Walker. Progettazione ottimale dell’apodizzazione per ultrasuoni medici utilizzando i minimi quadrati vincolati parte i: teoria. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 54:332–342, Febbraio 2007. doi:10.1109/tuffc.2007.247.
- [19] M. Bae, A. K. M. A. Hasan, and H. K. Kim. Fast minimum variance beamforming based on legendre polynomials. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 63:1422–1431, 2016. doi:10.1109/TUFFC.2016.2591623.
- [20] G. Matrone, P. L. Sarti, A. Quaglia, and M. L. Mischi. The delay multiply and sum beamforming algorithm in ultrasound b-mode medical imaging. *IEEE Transactions on Medical Imaging*, 34:940–949, 2015. doi:10.1109/TMI.2014.2371235.
- [21] G. Matrone, A. Ramalli, J. D’Hooge, P. Tortoli, and G. Magenes. A comparison of coherencebased beamforming techniques in high-frame-rate ultrasound imaging with multi-line transmission. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 67:329–340, Feb. 2020. doi:10.1109/TUFFC.2019.2945365.
- [22] Silvia Seoni, Giulia Matrone, and Kristen M. Meiburger. Texture analysis of ultrasound images obtained with different beamforming techniques and dynamic ranges – a robustness study. *Ultrasonics*, 131, 2023. doi:10.1016/j.ultras.2023.106940.
- [23] Patricia O’Gorman, Iacopo Chiavacci, and H Knipe. Propagation speed. *Radiopaedia.org*, 2016. URL: <https://radiopaedia.org/articles/46915>, doi:10.53347/rID-46915.

- [24] Michiya Mozyumi and Hideyuki Hasegawa. Adaptive beamformer combined with phase coherence weighting applied to ultrafast ultrasound. *Applied Sciences*, 8:204, 2018. doi:10.3390/app8020204.
- [25] Junseob Shin and Jesse T. Yen. Synergistic enhancements of ultrasound image contrast with a combination of phase aberration correction and dual apodization with cross-correlation. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 59:2089–2101, 2012.
- [26] M. H. Cho, L. H. Kang, J. S. Kim, and S. Y. Lee. An efficient sound speed estimation method to enhance image resolution in ultrasound imaging. *Ultrasonics*, 49:774–778, 2009. doi:10.1016/j.ultras.2009.06.005.
- [27] Changhan Yoon, Yuhwa Lee, Jin Ho Chang, Tai kyong Song, and Yangmo Yoo. In vitro estimation of mean sound speed based on minimum average phase variance in medical ultrasound imaging. *Ultrasonics*, 51:795–802, 2011. doi:10.1016/j.ultras.2011.03.007.
- [28] Martin E. Anderson and Gregg E. Trahey. *The direct estimation of sound speed using pulse-echo ultrasound*, volume 104. 1998. doi:10.1121/1.423889.
- [29] Hideyuki Hasegawa and Ryo Nagaoka. Studio iniziale su un fantasma per la stima della velocità del suono nel mezzo utilizzando la coerenza tra i segnali di eco ricevuti. *J Med Ultrason*, 46:297–307, 2019. doi:10.1007/s10396-019-00936-4.
- [30] Jaejin Lee, Yangmo Yoo, Changhan Yoon, and Tai kyong Song. Un metodo di stima della velocità media del suono computazionalmente efficiente basato su una valutazione della qualità della messa a fuoco per l’imaging ecografico medico. *Elettronica*, 8:1368, 2019. doi:10.3390/electronics8111368.
- [31] Pat De la Torre, Di Xiao, and Alfred C.H. Yu. One shot, one sos: A real-time, single-shot global speed of sound estimator. *Ultrasonics*, 153, 2025. doi:10.1016/j.ultras.2025.107612.
- [32] Lixin Gong Vijay Shamdasani Tsuyoshi Mitake Tetsuya Hayashi Yongmin Kim Siddhartha Sikdar, Ravi Managuli. A single mediaprocessor-based programmable ultrasound system. *IEEE Transactions on Information Technology in Biomedicine*, 7, 2003. doi:10.1109/TITB.2003.808512.
- [33] Rene Y. Choi, Aaron S. Coyner, Jayashree Kalpathy-Cramer, Michael F. Chiang, and J. Peter Campbell. Introduction to machine learning, neural networks, and deep learning. *Translational Vision Science Technology*, 9:14, 2020.
- [34] Università LUM Giuseppe Degennaro. Machine learning, 2025. URL: <https://lum.it/machine-learning/>.
- [35] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, Burlington, MA, USA, 3rd edition, 2011.
- [36] Rita Maestri. Metodiche di deep learning e applicazioni all’imaging medico: la radiomica, 2017.
- [37] Hamdy F. F. Mahmoud. Parametric versus semi and nonparametric regression models. *arXiv preprint arXiv:1906.10221*, 2019.
- [38] Mary Akinyemi, Chika Yinka-Banjo, Ogban-Asuquo Ugot, and Akwarandu Nwachuku. Estimating the time-lapse between medical insurance reimbursement with non-parametric regression models. *University of Lagos Working Paper*, 2023.

- [39] Impara Facilmente le Statistiche. Che cos'è: regressione non parametrica, 2025. URL: <https://it.statisticseasily.com/glossario/what-is-nonparametric-regression-explained/>.
- [40] Bradley J. Erickson, Panagiotis Korfiatis, Zeynettin Akkus, and Timothy L. Kline. Machine learning for medical imaging. *Radiographics*, 37:505–515, 2017. doi:10.1148/rg.2017160130.
- [41] Omar Burzio. Machine learning per la qualità nell'industria 4.0, 2022. Corso di Laurea in Ingegneria Informatica.
- [42] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA, USA, 1984.
- [43] Maria Laura Iovino. Riprogettazione di un processo aziendale di logistica, 2020. Corso di Laurea Magistrale in Ingegneria della Produzione Industriale e dell'Innovazione Tecnologica.
- [44] IBM. Cos'è il boosting?, 2024. URL: <https://www.ibm.com/it-it/think/boosting>.
- [45] Alessandro Felice. Applicazione di modelli di machine learning ad albero decisionale con r per il credit scoring nel settore elettronico. Tesi di laurea magistrale in ingegneria gestionale, Politecnico di Torino.
- [46] Angelo Tripodi. Interpretabilità dei risultati ottenuti dall'applicazione di algoritmi di machine learning ai sistemi di rating interni: analisi del modulo cash-flow del modello sme retail di intesa sanpaolo. Tesi di laurea magistrale, Politecnico di Torino, 2019.
- [47] IBM. Cos'è la regolarizzazione?, 2023. URL: <https://www.ibm.com/it-it/think/topics/regularization>.
- [48] Alexander Moore and Max Bell. Xgboost, una nuova tecnica di intelligenza artificiale spiegabile, nella previsione dell'infarto miocardico: uno studio di coorte di una biobanca del regno unito. *Clinical Medicine Insights: Cardiology*, 16:11795468221133611, 2022. doi:10.1177/11795468221133611.
- [49] IBM. Che cos'è xgboost?, 2023. URL: <https://www.ibm.com/it-it/think/topics/xgboost>.
- [50] 3B Scientific GmbH. Breast phantom for ultrasound and clinical training. <https://www.3bscientific.com/>, 2024.
- [51] 3B Scientific. Modello per seno singolo con tumori benigni, 2025.
- [52] T. Wang, C. Zhao, M. Xu, and X. Wang. Development of tissue-mimicking phantoms using pdms and agarose for photoacoustic imaging applications. *Journal of Biomedical Optics*, 24:105002, 2019. doi:10.1117/1.JBO.24.10.105002.
- [53] M. O. Culjat, D. Goldenberg, P. Tewari, and R. S. Singh. A review of tissue substitutes for ultrasound imaging. *Ultrasound in Medicine and Biology*, 36:861–873, 2010. doi:10.1016/j.ultrasmedbio.2010.02.012.
- [54] G. Brunetti, K. M. Meiburger, and L. Cavigli. Hierarchical materials based on polydimethylsiloxane as tissue-like phantoms for photoacoustic imaging and diffusive optics. In *Proceedings of SPIE 13308, Photons Plus Ultrasound: Imaging*

- and Sensing 2025*, San Francisco, CA, USA, 2025. URL: <https://iris.cnr.it/handle/20.500.14243/543105>.
- [55] Verasonics. Transduttori verasonics, 2025. URL: <https://verasonics.com/verasonics-transducers/>.
 - [56] Jorgen Arendt Jensen. A program for simulating ultrasound systems. *Medical & Biological Engineering & Computing*, 34:351–353, 1996.
 - [57] G. Castellano, L. Bonilha, L. M. Li, and F. Cendes. Texture analysis of medical images. *Clinical Radiology*, 59:1061–1069, 2004. doi:10.1016/j.crad.2004.07.008.
 - [58] Astrid Schneider, Gerhard Hommel, and Maria Blettner. Linear regression analysis. part 14 of a series on evaluation of scientific publications. *Dtsch Arztebl Int*, 107:776–782, Nov 2010. doi:10.3238/arztebl.2010.0776.
 - [59] Diana Elisa Chiari. *Applicazione di tecniche di analisi statistica e texture analysis all’imaging di risonanza magnetica di pazienti affetti da glioblastoma multiforme*. Tesi di laurea magistrale in fisica, Alma Mater Studiorum - Università di Bologna, 2014.
 - [60] J. J. M. van Griethuysen, A. Fedorov, C. Parmar, A. Hosny, N. Aucoin, V. Narayan, R. G. H. Beets-Tan, J. C. Fillon-Robin, S. Pieper, and H. J. W. L. Aerts. Computational radiomics system to decode the radiographic phenotype. *Cancer Research*, 77:104–107, 2017. doi:10.1158/0008-5472.CAN-17-0339.
 - [61] C. E. Massarzo. Estrazione di features, 2024. URL: <https://radiologica.net/estrazione-di-features/#::~:~:text=Gray%2DLevel%20Size%20Zone%20Matrix,grigio%2C%20indipendentemente%20dall’orientamento>.
 - [62] Xingwu He. Sound speed optimization based on fuzzy sets using image texture as quality factors. *BIO Web of Conferences*, 8, 2017. doi:10.1051/bioconf/20170803014.
 - [63] Kibo Nam, Mehnoosh Torkzaban, Dina Halegoua-DeMarzio, Corinne E. Wessner, and Andrej Lyshchik. Improving diagnostic accuracy of ultrasound texture features in detecting and quantifying hepatic steatosis using various beamforming sound speeds. *Phys Med Biol.*, 68, 2024. doi:10.1088/1361-6560/acb635.
 - [64] Alessandra Nardella. *Simulazione di immagini ecografiche transorbitali come Data Augmentation per la segmentazione delle strutture del nervo ottico*. PhD thesis, Politecnico di Torino, 2024.
 - [65] A. V. Kvostikov, A. S. Krylov, and U. R. Kamalov. Ultrasound image texture analysis for liver fibrosis stage diagnostics. *Pattern Recognition and Image Analysis*, 2015. doi:10.1134/S0361768815050059.
 - [66] Xingwu He. Sound speed optimization based on fuzzy sets using image texture as quality factors. In *BIO Web of Conferences*, volume 8. Chengdu Normal University, Chengdu, Sichuan, China, 2017. doi:10.1051/bioconf/20170803014.
 - [67] Kibo Nam, Mehnoosh Torkzaban, Dina Halegoua-DeMarzio, Corinne E. Wessner, and Andrej Lyshchik. Improving diagnostic accuracy of ultrasound texture features in detecting and quantifying hepatic steatosis using various beamforming sound speeds. *Physiological Measurement and Biology*, 68, 2024. doi:

- 10.1088/1361-6560/acb635.
- [68] D. Hyun, A. Wiacek, S. Goudarzi, S. Rothlübbers, A. Asif, K. Eickel, Y. C. Eldar, J. Huang, M. Misch, H. Rivaz, D. Sinden, R. J. G. van Sloun, H. Stroh, and M. A. L. Bell. Deep learning for ultrasound image formation: CUBDL evaluation framework and open datasets. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 68:3466–3483, 2021.