



# UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria Industriale — DII

Corso di Laurea Magistrale in Ingegneria Aerospaziale

Curriculum Aeronautico

## Verso la simulazione in tempo reale: utilizzo di modelli AI per la risoluzione accelerata di problemi CFD complessi

**Relatore:**

Prof. Federico Dalla Barba

**Corelatore:**

Ing. Andrea Dal Monte, PhD

**Studente:**

Nicolò Marinelli

Matricola 2122767

Anno Accademico 2025/2026



## ABSTRACT

---

L'evoluzione dei modelli di Deep Learning e il costante incremento della potenza computazionale hanno rivoluzionato i processi di progettazione e analisi in ambito ingegneristico. Nel contesto della fluidodinamica computazionale (CFD), le reti neurali a grafo (Graph Neural Networks - GNN) rappresentano una tecnologia emergente di grande potenziale: fornendo predizioni pressoché istantanee, permettono di superare i limiti temporali dei solutori tradizionali, che spesso richiedono tempi di elaborazione nell'ordine di ore o giorni.

Il presente elaborato si pone l'obiettivo di analizzare i vantaggi e le eventuali limitazioni dell'applicazione delle reti neurali a grafo in ambito CFD. Nello specifico, viene descritta la procedura di generazione del dataset sintetico, successivamente impiegato per l'addestramento di un'architettura neurale basata sulla metodologia X-MeshgraphNet, e la relativa fase di predizione. Per valutare l'efficacia del modello, i risultati ottenuti sono stati confrontati con una simulazione CFD tradizionale al fine di individuarne gli scostamenti.

Il caso applicativo in esame consiste nell'analisi termofluidodinamica di un cabinato elettrico, al cui interno un trasformatore dissipa calore per convezione naturale con l'ambiente circostante. I risultati dimostrano che l'integrazione delle reti neurali nella CFD abbatte significativamente il tempo computazionale. A fronte di un errore di predizione estremamente contenuto, la drastica contrazione delle risorse temporali richieste conferma la validità di questo approccio, che si configura come un nuovo e potente strumento di supporto alle decisioni in tempo reale e alla progettazione interattiva.



# INDICE

---

<b>Abstract</b>	<b>ii</b>
<b>Elenco delle figure</b>	<b>viii</b>
<b>Elenco delle tabelle</b>	<b>ix</b>
<b>1 Necessità delle simulazioni CFD in ambito elettronico</b>	<b>1</b>
1.1 Problematiche e analisi termiche nelle schede a circuito stampato (PCB) . . . . .	1
1.2 Problematiche e analisi termiche nei trasformatori MT/BT . . . . .	2
1.3 Problematiche e analisi termiche nei pacchi batteria dei veicoli . . . . .	3
1.4 Problematiche e analisi termiche dell'ambiente interno dei datacenter (HVAC) . . . . .	4
1.5 Limiti computazionali della CFD e necessità di modelli predittivi accelerati tramite AI . . . . .	6
1.6 Obiettivo e struttura della tesi . . . . .	7
<b>2 Introduzione all'Intelligenza Artificiale e Machine Learning</b>	<b>9</b>
2.1 Le Reti neurali artificiali (ANN) . . . . .	9
2.2 Livelli di una rete neurale . . . . .	10
2.3 L'algoritmo di backpropagation . . . . .	12
2.4 Physics-Informed Neural Networks (PINN) . . . . .	12
2.5 Tipologie di architetture e algoritmi di apprendimento . . . . .	13
2.5.1 Reti Feedforward (FFNN) . . . . .	13
2.5.2 Reti Neurali Ricorrenti (RNN) e LSTM . . . . .	14
2.5.3 Reti Neurali Convoluzionali (CNN) . . . . .	14
2.5.4 Graph Neural Networks (GNN) . . . . .	14
<b>3 Metodologia delle Graph Neural Networks e Integrazione di MeshGraphNet</b>	<b>16</b>
3.1 Reti Neurali a Grafo (GNN) . . . . .	16
3.1.1 Obiettivi del capitolo . . . . .	16
3.1.2 Il paradigma MeshGraphNet . . . . .	16
3.2 Fondamenti di teoria dei grafi . . . . .	17
3.2.1 Rappresentazione della mesh in MeshGraphNet . . . . .	19
3.3 Il meccanismo di Message Passing . . . . .	19
3.4 Stratificazione e propagazione dell'informazione . . . . .	21

3.5	Analisi e task predittivi . . . . .	21
3.6	Spaider e l'integrazione della X-MeshGraphNet . . . . .	22
3.6.1	Processo tecnico di raffinamento . . . . .	23
3.6.2	Vantaggi del metodo . . . . .	23
<b>4</b>	<b>Pre-elaborazione dei dati e addestramento</b>	<b>25</b>
4.1	Pulizia dei dati e suddivisione del dataset . . . . .	25
4.1.1	Data cleaning in ambito fluidodinamico . . . . .	25
4.1.2	Partizionamento del dataset . . . . .	26
4.2	Normalizzazione delle feature in ambito CFD . . . . .	26
4.3	Data augmentation . . . . .	27
4.4	Parametri e iperparametri del modello . . . . .	27
4.4.1	Parametri del modello . . . . .	27
4.4.2	Iperparametri . . . . .	28
4.5	Il Processo di addestramento: forward pass e funzioni di attivazione .	29
4.5.1	Il forward pass . . . . .	29
4.5.2	Le funzioni di attivazione . . . . .	29
4.6	La funzione di perdita (Loss Function) . . . . .	30
4.6.1	Principali funzioni di perdita per la regressione . . . . .	30
4.6.2	La funzione di perdita informata dalla fisica (Physics-Informed Loss) . . . . .	31
4.7	Valutazione e generalizzazione: overfitting e underfitting . . . . .	32
4.7.1	Underfitting (Sottoadattamento) . . . . .	32
4.7.2	Overfitting (Sovraadattamento) . . . . .	32
4.7.3	Il Trade-off tra bias e varianza . . . . .	33
4.7.4	Tecniche per il contrasto dell'overfitting . . . . .	33
<b>5</b>	<b>Caso di analisi</b>	<b>35</b>
5.1	Selezione del caso studio: analisi termica di un cabinato elettrico con trasformatore . . . . .	35
5.1.1	Motivazioni della scelta . . . . .	35
5.1.2	Complessità del dominio . . . . .	35
5.2	Configurazione della simulazione e condizioni al contorno . . . . .	36
5.3	Modelli fisici e turbolenza . . . . .	39
5.4	Proprietà del fluido e condizioni operative . . . . .	39
5.4.1	Modello di gas ideale . . . . .	40
5.4.2	Accelerazione di gravità . . . . .	40
5.4.3	Definizione della densità operativa . . . . .	40

5.5	Metodi di soluzione e discretizzazione spaziale . . . . .	40
5.5.1	Strategia di risoluzione numerica . . . . .	40
5.5.2	Schemi di discretizzazione spaziale . . . . .	41
5.6	Verifica della convergenza e validazione dei risultati . . . . .	41
<b>6</b>	<b>Generazione del Dataset</b>	<b>45</b>
6.1	Definizione dello spazio dei parametri e strategia di campionamento .	45
6.1.1	Metodologia di campionamento: Latin Hypercube Sampling (LHS) . . . . .	46
6.2	Automazione della modellazione geometrica e della <i>Named Selection</i> .	47
6.3	Automazione del meshing e delle simulazioni . . . . .	47
<b>7</b>	<b>Analisi dei risultati della predizione</b>	<b>49</b>
7.1	Metriche di valutazione delle performance: NRMSE . . . . .	49
7.2	Analisi dei risultati e prestazioni della rete neurale . . . . .	49
7.3	Analisi dei Campi Volumetrici (VTU) . . . . .	53
7.3.1	Validazione qualitativa: confronto visivo dei campi di pressione	53
7.3.2	Validazione qualitativa: confronto visivo dei campi di temperatura	56
7.3.3	Validazione qualitativa: confronto visivo dei campi di velocità	58
7.4	Analisi dei campi superficiali (VTP) . . . . .	60
7.4.1	Validazione qualitativa: confronto visivo dei campi di pressione	60
7.4.2	Validazione qualitativa: confronto visivo dei campi di temperatura	61
7.5	Confronto prestazionale e vantaggio computazionale . . . . .	62
<b>8</b>	<b>Conclusioni</b>	<b>64</b>

## ELENCO DELLE FIGURE

---

3.1	Rappresentazione schematica di grafi diretti e non diretti. . . . .	18
3.2	Rappresentazione matrice di adiacenza . . . . .	18
4.1	Rappresentazione grafica di overfitting e underfitting. . . . .	32
5.1	Rappresentazione grafica del dominio in analisi . . . . .	36
5.2	Rappresentazione delle feritoie. . . . .	37
5.3	Residui scalati . . . . .	42
5.4	Temperatura media e massima sulla superficie del trasformatore . . .	42
5.5	Temperatura media e massima sulla porta del cabinato . . . . .	43
5.6	Temperatura media e massima sul soffitto del cabinato . . . . .	43
5.7	Residui scalati . . . . .	44
7.1	Rappresentazione grafica della Loss Function all'avanzare delle epoche per le predizioni superficiali. . . . .	50
7.2	Rappresentazione grafica dell'apprendimento delle diverse variabili nella predizione superficiale. . . . .	51
7.3	Rappresentazione grafica della Loss Function all'avanzare delle epoche.	52
7.4	Confronto del campo di pressione $p$ [Pa] sulla sezione trasversale $x$ . .	54
7.5	Confronto del campo di pressione $p$ [Pa] sulla sezione longitudinale $y$ .	55
7.6	Confronto del campo di pressione $p$ [Pa] sulla sezione orizzontale $z$ . .	55
7.7	Confronto qualitativo del campo di temperatura $T$ [°C] sulla sezione trasversale $x$ . . . . .	56
7.8	Confronto qualitativo del campo di temperatura $T$ [°C] sulla sezione longitudinale $y$ . . . . .	57
7.9	Confronto qualitativo del campo di temperatura $T$ [°C] sulla sezione orizzontale $z$ . . . . .	57
7.10	Confronto qualitativo del campo di velocità $U$ [m/s] sulla sezione trasversale $x$ . . . . .	58
7.11	Confronto qualitativo del campo di velocità $U$ [m/s] sulla sezione longitudinale $y$ . . . . .	59
7.12	Confronto qualitativo del campo di velocità $U$ [m/s] sulla sezione orizzontale $z$ . . . . .	59
7.13	Confronto del campo di pressione $p$ [Pa] sulle superfici del trasformatore.	60
7.14	Confronto del campo di pressione $p$ [Pa] sulle superfici del cabinato. .	61

7.15	Confronto qualitativo del campo di temperatura $T$ [°C] sulle superfici del trasformatore . . . . .	61
7.16	Confronto qualitativo del campo di temperatura $T$ [°C] sulla superficie del cabinato . . . . .	62

## ELENCO DELLE TABELLE

---

5.1	Condizioni al contorno di pressione (Inlet e Outlet) . . . . .	37
5.2	Condizioni termiche per pareti con scambio convettivo . . . . .	38
5.3	Coefficienti di scambio termico derivati dai dati tecnici disponibili in rete. . . . .	38
5.4	Condizioni termiche per componenti del Trasformatore . . . . .	39
5.5	Schemi di discretizzazione spaziale adottati nella simulazione. . . . .	41
6.1	Range di variazione dei parametri per la generazione dei 100 campioni. . . . .	45
7.1	Metriche di errore NRMSE per i set di Training, Validazione e Test (Caso VTP). . . . .	51
7.2	Metriche di errore NRMSE per i set di Training, Validazione e Test (Caso VTU). . . . .	53
7.3	Confronto dei tempi di calcolo: Solutore CFD tradizionale vs Rete Neurale. . . . .	63

# 1. NECESSITÀ DELLE SIMULAZIONI CFD IN AMBITO ELETTRONICO

---

La progressiva miniaturizzazione dei componenti nel settore dei *printed circuit board* (PCB), l'introduzione di normative sempre più stringenti in materia di dissipazione termica nelle cabine dei trasformatori MT/BT, l'aumento della densità di potenza e le crescenti richieste di efficienza nei pacchi batterie per macchine e veicoli, insieme alla necessità di sistemi di raffreddamento sempre più performanti per il condizionamento degli ambienti (HVAC), in particolare nei data center, rendono indispensabile, in fase di progettazione, l'esecuzione di analisi termiche sempre più rapide e accurate. Tali analisi risultano fondamentali per affrontare efficacemente queste criticità, riducendo al contempo i tempi di sviluppo, di validazione sperimentale e test.

## 1.1 Problematiche e analisi termiche nelle schede a circuito stampato (PCB)

Nel caso delle schede a circuito stampato (PCB), una delle principali criticità è rappresentata dal surriscaldamento dei componenti, fenomeno che può causare malfunzionamenti sia a livello del singolo dispositivo sia dell'intero sistema elettronico. L'analisi termica si concentra sulla valutazione dei valori di temperatura raggiunti nelle diverse condizioni operative: il superamento del range di funzionamento ottimale accelera i processi di degrado, riduce la vita utile dei componenti e può condurre al guasto immediato di uno o più elementi, compromettendo così l'affidabilità del sistema.

Le problematiche termiche sono fortemente influenzate dal posizionamento dei componenti, strettamente legato al layout elettronico della scheda. A ciò si aggiungono gli effetti della composizione e della struttura del PCB, determinati dal numero di layer, dalla loro configurazione (in particolare dalla percentuale di rame rispetto al materiale isolante), dallo spessore dei singoli strati e da quello complessivo, nonché dalla distribuzione dei collegamenti verticali (*vias*) tra i vari layer[1, 2].

Per quanto riguarda le applicazioni terrestri, le simulazioni CFD non sono sempre eseguite, sebbene siano fortemente raccomandate. In tali casi, l'approccio più diffuso per la determinazione degli eventuali punti caldi (*hot spot*) consiste nel calcolo della densità di corrente che attraversa le diverse tracce e componenti, stimando successivamente la temperatura raggiunta sulla base di tali valori. In ambito spaziale, invece, le simulazioni CFD risultano ampiamente necessarie, soprattutto a causa dell'assenza di atmosfera, che impedisce lo scambio termico per convezione.

Nell'eventualità che vengano condotte, si adottano prevalentemente simulazioni di tipo CHT (Conjugate Heat Transfer), nelle quali vengono considerate unicamente la conduzione termica all'interno del PCB dei componenti e la convezione verso il fluido circostante (in ambito terrestre).

Una delle principali difficoltà nella simulazione termica dei PCB consiste nella corretta modellazione dei diversi strati. Per molti anni, la letteratura si è concentrata sull'individuazione di metodologie finalizzate alla definizione di un modello equivalente del PCB, trattandolo come materiale omogeneo con un'unica conducibilità termica[3, 4, 5], oppure assegnando valori distinti di conducibilità in-plane e out-of-plane per rappresentarne il comportamento anisotropo.

È stato tuttavia dimostrato che tali approcci conducono frequentemente a mappe termiche imprecise[6, 4]; nei pochi casi in cui si ottengono risultati soddisfacenti, i valori di conducibilità risultano determinati tramite procedure iterative di adattamento basate su tentativi successivi (try and error), rendendo il metodo poco affidabile e difficilmente generalizzabile.

Attualmente, l'approccio più diffuso prevede l'importazione dei modelli ECAD e CAD del PCB all'interno dell'ambiente di simulazione, seguita dalla suddivisione di ciascun layer in celle. Per ogni cella viene calcolata la frazione volumetrica di rame a partire dai dati ECAD e, sulla base di questa, viene determinata la conducibilità termica effettiva. Durante il passaggio dalla mesh dei layer alla mesh CFD si applica quindi un processo di interpolazione, che consente di ottenere una mappatura continua e completa della conducibilità termica dell'intero PCB. Il tempo di calcolo e il livello di dettaglio della mappatura dipendono direttamente dalla risoluzione della mesh termica dei layer[6].

I componenti reali vengono infine modellati come sorgenti di calore, mentre il contatto termico tra componente e scheda è rappresentato mediante un'interfaccia caratterizzata da una resistenza termica, dipendente dal tipo di saldatura o dal materiale adesivo impiegato per il fissaggio. Richiedendo milioni di celle per una simulazione sufficientemente accurata, i tempi di calcolo sono estremamente elevati.

## **1.2 Problematiche e analisi termiche nei trasformatori MT/BT**

I trasformatori di Media e Bassa Tensione (MT/BT) rappresentano i nodi nevralgici delle reti di distribuzione elettrica. Durante il processo di trasformazione, una frazione non trascurabile di energia viene dissipata sotto forma di calore dalle perdite per effetto Joule negli avvolgimenti in rame o alluminio.

La gestione termica di questi dispositivi è una sfida ingegneristica di primaria importanza, resa ancora più necessaria dalle recenti normative europee in materia

di ecodesign (normativa Commission Regulation (EU) 2019/1783 of 1 October 2019 amending Regulation (EU) No 548/2014 on implementing Directive 2009/125/ EC of the European Parliament and of the Council with regard to small, medium and large power transformers.), che impongono limiti stringenti sulle perdite massime consentite. Il superamento delle temperature operative di progetto non solo compromette l'efficienza della macchina, ma accelera drasticamente il degrado dei materiali isolanti (come resine o carta impregnata), riducendo la vita utile del trasformatore e aumentando il rischio di guasti catastrofici e cortocircuiti[7].

Le analisi termo-fluidodinamiche in questo settore si concentrano sullo studio dei moti convettivi naturali o forzati per il fluido di raffreddamento . Storicamente, il dimensionamento termico veniva effettuato mediante reti termiche concentrate o formule empiriche. Ad oggi, la necessità di predire la dissipazione termica e le temperature superficiali raggiunte dai vari componenti con elevata precisione prima della messa a terra dell'impianto, rende la simulazione CFD un requisito fondamentale durante la fase di progettazione[8, 9, 10, 11, 12].

### **1.3 Problematiche e analisi termiche nei pacchi batteria dei veicoli**

La generazione e l'accumulo di calore nelle batterie agli ioni di litio aumentano con l'innalzamento della temperatura operativa durante il ciclo di funzionamento, determinando una riduzione delle prestazioni e l'insorgenza di criticità legate alla sicurezza. In particolare, temperature elevate possono innescare la decomposizione dell'elettrolita, dando origine al fenomeno del thermal runaway (TR), un processo esotermico e autocatalitico che può condurre all'esplosione della batteria. Al contrario, temperature estremamente basse comportano una riduzione dell'energia e della potenza erogabile. Il thermal runaway rappresenta pertanto la principale problematica di sicurezza nei sistemi di accumulo per applicazioni automotive. Inoltre, il funzionamento al di fuori del range termico ottimale accelera i processi di degrado della batteria e ne determina una perdita di efficienza complessiva[13, 14].

La dissipazione termica generata durante il funzionamento della cella viene solitamente descritta mediante un modello analitico, spesso basato sulla equazione di Bernardi, che consente di stimare il calore prodotto dai processi elettrochimici. Un ulteriore approccio, ampiamente diffuso nella modellazione del rilascio di calore durante il thermal runaway, consiste nell'includere le reazioni di decomposizione dell'elettrolita e dei materiali attivi tramite equazioni di tipo Arrhenius, al fine di rappresentare in maniera più realistica la natura fortemente esotermica e autocatalitica del fenomeno[15].

Alcuni studi integrano inoltre ulteriori meccanismi fisici, quali il cortocircuito interno o reazioni chimiche secondarie, trattandoli come sorgenti termiche aggiuntive

che contribuiscono al degrado della cella e all'innesco del TR[14].

Le singole celle vengono generalmente analizzate mediante modelli FEM o modelli analitici, che permettono di determinare la dissipazione termica generata durante il funzionamento. Il valore di potenza termica così ottenuto viene successivamente utilizzato come dato di ingresso per simulazioni CFD bidimensionali o tridimensionali, impiegate per calcolare i campi di temperatura e valutare il comportamento termico del modulo o del pacco batteria.

La tipologia di simulazione dipende dal mezzo di raffreddamento adottato (aria o fluido liquido). Nella maggior parte dei casi si ricorre comunque a simulazioni di tipo CHT, nelle quali la batteria viene modellata attraverso un modello termico concentrato (heat source), senza risolvere nel dettaglio la geometria interna della cella [13].

#### **1.4 Problematiche e analisi termiche dell'ambiente interno dei datacenter (HVAC)**

Con la crescita costante dei data center, inizialmente legata ai servizi IT tradizionali e oggi fortemente accelerata dalle applicazioni di intelligenza artificiale, la gestione termica a livello di sala è diventata un elemento di criticità primaria. L'efficienza energetica e i costi dei sistemi di raffreddamento assumono un ruolo determinante, considerando che il cooling può incidere fino al 40% sui costi operativi complessivi di un data center[16, 17].

L'obiettivo principale è l'incremento dell'efficienza dell'impianto di climatizzazione mantenendo un valore di Power Usage Effectiveness (PUE) inferiore a 1,8. Ciò consente di ridurre la quota di energia non destinata direttamente all'alimentazione dei server blade, con conseguente diminuzione dei costi operativi e delle emissioni di anidride carbonica. A questo si aggiunge l'introduzione di normative sempre più stringenti, che limitano la dispersione energetica e impongono valori minimi per la temperatura dell'aria in ingresso ai rack, al fine di garantire un utilizzo più razionale delle risorse energetiche[18].

Le problematiche termiche in questo contesto sono quindi determinate sia dai vincoli normativi sia dalle esigenze di massimizzazione dell'efficienza energetica. Le normative definiscono limiti di temperatura minima per l'aria in ingresso ai rack e limiti massimi per l'aria in uscita, con l'obiettivo di contenere il consumo energetico dei sistemi di raffreddamento. Tali vincoli risultano coerenti con le strategie aziendali orientate a rendere gli impianti di climatizzazione sempre più efficienti. Le principali criticità riguardano il bypass dell'aria, la miscelazione tra aria fredda e aria calda, l'assenza di un adeguato isolamento tra corridoi freddi e caldi, il posizionamento delle

unità CRAC e CRAH, l'altezza del plenum e del soffitto, nonché la distribuzione delle piastrelle forate o delle bocchette per il raffreddamento dedicato[16, 17, 19].

In questo ambito, le analisi CFD vengono eseguite in maniera estensiva. Le simulazioni sono condotte a livello di sala e mirano a studiare il comportamento termico e fluidodinamico dell'intero ambiente. Gli aspetti principali oggetto di analisi includono:

- la **profondità del plenum** e il suo effetto sulla distribuzione dell'aria;
- le **temperature dell'ambiente**, con verifica del rispetto dei limiti in ingresso e in uscita dai rack;
- l'**altezza del soffitto** e la sua influenza sui percorsi dell'aria calda;
- la **disposizione dei rack e delle unità CRAC/CRAH**, determinante per la corretta circolazione e stratificazione dell'aria;
- la **miscelazione tra aria calda e aria fredda** e i fenomeni di ritorno caldo;
- la **forma e le dimensioni delle piastrelle forate**, che influenzano la portata d'aria dedicata;
- il **bypass dell'aria** e le inefficienze dovute a percorsi non controllati;
- le **soluzioni di isolamento dei corridoi freddi e caldi**;
- le **nuove strategie di utilizzo delle unità CRAC/CRAH**, come la ventilazione lungo la linea dei rack o quella dedicata al singolo rack;
- le **metodologie di raffreddamento a liquido**, sempre più diffuse in presenza di elevate densità di potenza.

In queste simulazioni, le unità CRAC e CRAH vengono generalmente modellate come volumi semplificati (black box) dotati di condizioni al contorno di inlet e outlet, oppure come porous region, per simulare la resistenza aerodinamica interna. I rack sono modellati come blocchi volumetrici con salto termico assegnato, utilizzando condizioni di ingresso e uscita dell'aria fornite dal produttore, senza risolvere nel dettaglio la geometria interna dei server blade. Le piastrelle forate sono rappresentate tramite porous region o baffle, alle quali viene imposta una determinata resistenza al moto dell'aria. L'altezza del plenum e quella del soffitto vengono infine parametrizzate, al fine di valutare l'impatto di tali grandezze sulla distribuzione dei flussi d'aria e sul campo di temperatura[19, 16].

Questa impostazione consente di individuare le principali inefficienze, ottimizzare la disposizione della sala e valutare nuove soluzioni di raffreddamento, mantenendo un modello numerico computazionalmente efficiente e senza la necessità di simulare nel dettaglio la geometria interna delle unità CRAC, CRAH e dei server blade

### **1.5 Limiti computazionali della CFD e necessità di modelli predittivi accelerati tramite AI**

Sebbene le simulazioni termo-fluidodinamiche rappresentino oggi uno standard industriale imprescindibile, il loro impiego sistematico nelle fasi di ottimizzazione iterativa del *design* (come la scelta del *layout* in un PCB o il dimensionamento di un sistema di raffreddamento a liquido per batterie) incontra un limite operativo insormontabile: l'elevato costo computazionale. La risoluzione numerica delle equazioni di Navier-Stokes su domini discretizzati con milioni di celle richiede l'impiego prolungato di infrastrutture HPC (*High Performance Computing*), con tempi di calcolo che possono variare da diverse ore a svariati giorni per una singola configurazione geometrica.

Questo collo di bottiglia temporale rende di fatto impraticabile l'esplorazione estensiva dello spazio di progettazione. Non potendo simulare centinaia di varianti geometriche in tempi utili, i progettisti sono spesso costretti ad accontentarsi di soluzioni sub-ottimali o a ripiegare su campagne sperimentali basate su iterazioni di *trial and error*, le quali comportano costi di prototipazione elevati e tempi di validazione altrettanto dilatati.

L'introduzione dell'Intelligenza Artificiale (AI) – e in particolare del *Deep Learning* – nell'ambito della fluidodinamica computazionale si inserisce esattamente in questo divario, operando un radicale cambio di paradigma. Una volta completata la fase di addestramento (fase *training*, onerosa dal punto di vista computazionale ma eseguita *una tantum*), un modello neurale è in grado di effettuare una predizione completa del campo di velocità e temperatura e non solo, in pochi secondi (fase di *inferenza*). Questa riduzione drastica del tempo di analisi, che abbatte i costi computazionali di interi ordini di grandezza, sblocca la possibilità di effettuare ottimizzazioni in tempo reale.

Rispetto all'approccio classico, l'uso di reti neurali (come quelle basate sui grafi) offre inoltre costi operativi nettamente inferiori, pur mantenendo un livello di accuratezza predittiva paragonabile a quello delle simulazioni RANS (*Reynolds-Averaged Navier-Stokes*). L'accuratezza finale del modello surrogato dipende tuttavia dalla qualità dei dati utilizzati per il *training*: qualora la rete venga addestrata sfruttando dataset derivati da simulazioni ad alta fedeltà come LES (*Large Eddy Simulation*) o DNS (*Direct Numerical Simulation*), è possibile ottenere in fase predittiva un dettaglio della fisica

dei fluidi superiore a quello delle classiche simulazioni industriali RANS, unendo di fatto l'accuratezza dell'alta fedeltà alla velocità dell'intelligenza artificiale.

## 1.6 Obiettivo e struttura della tesi

Alla luce delle limitazioni computazionali intrinseche ai solutori tradizionali, l'obiettivo principale del presente lavoro di tesi è valutare e validare l'impiego del *Deep Learning* – nello specifico le *Graph Neural Networks* (GNN) – come modello surrogato per l'accelerazione delle simulazioni termo-fluidodinamiche.

In particolare, la ricerca si concentra su un caso di studio applicato all'ambito del raffreddamento elettronico, impiegando l'architettura neurale sviluppata dalla *start-up* Spaider, la quale si basa su un'evoluzione multiscala del paradigma *MeshGraphNet*. Per dimostrare la validità di tale approccio, il lavoro persegue i seguenti obiettivi specifici:

- **Analisi dell'architettura neurale:** Comprendere e illustrare matematicamente come la topologia di una mesh fluidodinamica non strutturata possa essere elaborata da una GNN tramite il meccanismo di *message passing*.
- **Gestione del dato fisico:** Strutturare una *pipeline* rigorosa per la pulizia, la normalizzazione e l'eventuale *augmentation* dei dati grezzi derivanti dalle simulazioni CFD di *ground truth*, rendendoli idonei all'addestramento.
- **Validazione quantitativa:** Confrontare direttamente le predizioni termiche e fluidodinamiche restituite dal modello di Intelligenza Artificiale con i risultati ottenuti tramite solutori CFD tradizionali.
- **Valutazione delle performance:** Quantificare l'effettivo *speed-up* computazionale (riduzione dei tempi di inferenza) e il compromesso tra l'accuratezza fisica del risultato e il risparmio di risorse hardware.

## Struttura dell'elaborato

Per affrontare metodicamente tali obiettivi, la trattazione è stata organizzata secondo la seguente struttura logica:

- Il **Capitolo 2** introduce i concetti fondamentali dell'Intelligenza Artificiale, del *Machine Learning* e delle Reti Neurali Artificiali, evidenziando i limiti delle architetture classiche nel trattamento dei domini spaziali non strutturati.

- Il **Capitolo 3** entra nel merito del *Deep Learning* , esponendo la teoria dei grafi e l'architettura *MeshGraphNet*, per poi presentare l'approccio multiscala alla base del solutore Spaider.
- Il **Capitolo 4** analizza le delicate fasi di *pre-processing* dei dati, definendo le metodologie di pulizia, partizionamento, regolarizzazione e addestramento del modello, ponendo particolare attenzione al contrasto dell' *overfitting*.
- Il **Capitolo 5** presenta il caso in analisi e il *setup* delle simulazioni utilizzate per generazione del *dataset*.
- Il **Capitolo 6** descrive la procedura di generazione del dataset sintetico, approfondendo la strategia di *Latin Hypercube Sampling* (LHS) impiegata per garantire un'esplorazione uniforme dello spazio dei parametri geometrici e fisici. Viene inoltre illustrata l'automazione dell'intero workflow tramite PyFluent e i compromessi ingegneristici adottati per bilanciare l'accuratezza fisica con la sostenibilità computazionale del modello neurale.
- Il **Capitolo 7** espone l'analisi quantitativa e qualitativa dei risultati predittivi, utilizzando la metrica NRMSE per garantire una valutazione adimensionale delle performance. Vengono confrontati i processi di addestramento su domini superficiali (VTP) e volumetrici (VTU). Infine, l'elaborato dimostra lo straordinario vantaggio computazionale della metodologia proposta.
- Il **Capitolo 8** sintetizza le conclusioni dell'indagine, confermando l'efficacia delle Graph Neural Networks nel trasformare la progettazione tecnica in un processo interattivo in tempo reale. Vengono discussi i vantaggi dell'approccio ibrido, in cui l'AI funge da inizializzatore per accelerare la convergenza dei solutori classici. La trattazione si conclude delineando le prospettive future, orientate all'espansione del dataset e all'integrazione di modelli informati dalla fisica (PINN) per una superiore coerenza dei risultati

## 2. INTRODUZIONE ALL'INTELLIGENZA ARTIFICIALE E MACHINE LEARNING

---

L'Intelligenza Artificiale (AI) rappresenta un vasto ambito della computer science dedito allo sviluppo di sistemi capaci di eseguire compiti complessi che, tradizionalmente, richiederebbero l'intervento dell'intelligenza umana, quali la risoluzione di problemi, il processo decisionale e il riconoscimento di pattern. A differenza dei software basati su regole rigide e deterministiche, i moderni sistemi di AI traggono la loro efficacia dalla capacità di apprendere direttamente dai dati, migliorando le proprie prestazioni attraverso l'esperienza. All'interno di questo vasto ecosistema, è possibile delineare una gerarchia di specializzazione:

**Machine Learning (ML):** È il cuore operativo dell'AI moderna. Consiste in un insieme di tecniche e algoritmi — come la regressione lineare, le *Support Vector Machines* (SVM), gli alberi decisionali e il clustering — che permettono ai computer di identificare relazioni statistiche nei dati senza essere esplicitamente programmati per ogni scenario.

**Deep Learning (DL):** Rappresenta un'evoluzione del Machine Learning basata sull'impiego di reti neurali artificiali “profonde” (dotate di numerosi strati)[20]. Il Deep Learning è particolarmente efficace nell'elaborare dati ad alta dimensionalità e non strutturati, come immagini, segnali vocali e, nel contesto della presente tesi, campi fisici complessi derivanti da simulazioni, grazie alla capacità di apprendere rappresentazioni gerarchiche dei dati.

### 2.1 Le Reti neurali artificiali (ANN)

Le Reti Neurali Artificiali (ANN) costituiscono l'architettura fondamentale del Deep Learning. Ispirate parzialmente alla struttura biologica dei neuroni cerebrali, queste reti sono composte da una serie di unità di calcolo elementari, chiamate nodi o neuroni, organizzate in una struttura stratificata (**Multi-Layer Perceptron**):

- **Input Layer:** riceve le informazioni grezze dal dataset.
- **Hidden Layers (Strati nascosti):** eseguono trasformazioni matematiche intermedie per estrarre caratteristiche rilevanti dai dati.
- **Output Layer:** produce la previsione o la classificazione finale.

Dal punto di vista matematico, una rete neurale ha lo scopo di apprendere una funzione complessa  $f$  che mappi un vettore di input  $\mathbf{x} \in \mathbb{R}^n$  (ad esempio, le coordinate di una mesh o le condizioni al contorno) verso una risposta predetta  $\hat{\mathbf{y}} \in \mathbb{R}^m$  (come la pressione o la velocità del fluido):

$$\hat{\mathbf{y}} = f(\mathbf{x}; \theta) \tag{2.1}$$

dove  $\theta$  rappresenta l'insieme dei parametri interni (pesi e bias) che la rete ottimizza durante l'addestramento. Ciò che distingue le reti neurali dai modelli statistici tradizionali è la loro capacità di eseguire trasformazioni non lineari concatenate; queste sono necessarie per superare i limiti della linearità, sfruttando apposite *funzioni di attivazione* che regolano il passaggio di informazioni tra uno strato e l'altro secondo regole matematiche predefinite. Questa proprietà è essenziale per la modellazione dei fenomeni fluidodinamici, intrinsecamente caratterizzati da forti non-linearità e gradienti elevati che non potrebbero essere catturati da modelli lineari semplici.

## 2.2 Livelli di una rete neurale

Un'architettura neurale standard è organizzata in una struttura sequenziale di neuroni interconnessi, distribuita su tre tipologie principali di livelli: *Input*, *Hidden* e *Output*.

### A. Il livello di input (Input Layer)

Il livello di input funge da interfaccia tra il dominio fisico esterno e il modello matematico; rappresenta la sezione di ingresso delle informazioni. A differenza degli strati successivi, questo livello è **passivo**: non esegue calcoli attivi (ovvero non possiede pesi  $w$  o bias  $b$ ), ma ha il compito di ricevere, normalizzare e trasmettere i dati grezzi.

Il numero di neuroni è determinato univocamente dalla dimensionalità del problema. Nel contesto di una simulazione fluidodinamica (CFD), il vettore di input  $\mathbf{x}$  è solitamente composto da:

- **Coordinate Spaziali e Temporali** ( $x, y, z, t$ ): definizione del punto nel dominio spazio-temporale.
- **Parametri Operativi**: grandezze adimensionali come il Numero di Reynolds ( $Re$ ), il Numero di Mach ( $Ma$ ) o l'angolo d'attacco ( $\alpha$ ).

Poiché tali variabili presentano ordini di grandezza eterogenei, la fase di **normalizzazione** in questo strato è fondamentale per evitare il predominio numerico di alcune *feature*, che renderebbe la rete “cieca” rispetto ai parametri con ordine di grandezza inferiore.

## B. I livelli nascosti (Hidden Layers)

I livelli nascosti costituiscono il motore computazionale della rete. Attraverso una serie di trasformazioni affini seguite da funzioni di attivazione non lineari, il cui compito è decidere se attivare o disattivare la comunicazione tra neuroni, questi strati estraggono progressivamente rappresentazioni astratte dei dati.

Ogni strato nascosto compie una trasformazione dello spazio delle *feature*: l’obiettivo è mappare dati intrinsecamente non lineari e complessi (come i campi di moto turbolenti) in uno spazio in cui la relazione con l’output risulti modellabile. Grazie alla presenza di molteplici strati e di attivazioni come **ReLU** (*Rectified Linear Unit*) o **Tanh**, il modello beneficia del **Teorema di Approssimazione Universale**. Tale principio stabilisce che una rete con una struttura sufficientemente profonda e ampia può approssimare con arbitraria precisione qualunque funzione continua, incluse le soluzioni delle equazioni di *Navier-Stokes*.

## C. Il livello di output (Output Layer)

Il livello di output traduce le rappresentazioni interne della rete in variabili fisiche concrete. In un tipico problema di regressione fluidodinamica, il vettore di output  $\hat{\mathbf{y}}$  sarà composto dalle variabili di interesse:

$$\hat{\mathbf{y}} = [u, v, p]^T \quad (2.2)$$

dove  $u$  e  $v$  sono le componenti della velocità e  $p$  è la pressione locale. La scelta della funzione di attivazione in questo strato è dettata dalla natura del target:

- **Lineare (Identità)**: è la scelta standard in CFD, poiché non impone vincoli al range dei valori in uscita.
- **Softmax**: utilizzata qualora il compito sia la classificazione del regime di moto (es. laminare vs turbolento).

Mentre nei livelli nascosti l’attivazione serve a introdurre non-linearità per far “imparare” alla rete relazioni complesse, nel livello di output la funzione ha un ruolo di vincolo matematico. Serve a garantire che i numeri in uscita abbiano senso rispetto a ciò che stai cercando di predire. Un aspetto cruciale è la **denormalizzazione**: i valori

predetti devono essere riconvertiti nelle unità di misura fisiche originali (es. *Pa* o *m/s*) per l'utilizzo ingegneristico.

### 2.3 L'algoritmo di backpropagation

Lo sviluppo successivo è il meccanismo fondamentale che permette alla rete di apprendere dai propri errori, ossia la Backpropagation (retro-propagazione dell'errore)[20]. Una volta erogato l'output, si calcola l'errore di predizione (Loss Function  $\mathcal{L}$ ), l'algoritmo procede a ritroso verso l'input per determinare l'impatto di ogni peso  $w$  sull'errore finale. Questo processo si basa sulla Regola della Catena (Chain Rule), che permette di calcolare il gradiente della funzione di perdita rispetto a un generico peso  $w_{ij}$  tra i neuroni  $i$  e  $j$ :

$$\frac{\partial \mathcal{L}}{\partial w_{ij}} = \frac{\partial \mathcal{L}}{\partial \text{out}_j} \cdot \frac{\partial \text{out}_j}{\partial \text{net}_j} \cdot \frac{\partial \text{net}_j}{\partial w_{ij}}$$

Dove le due componenti fondamentali dell'elaborazione di un singolo nodo sono:

- $\text{net}_j$  rappresenta l'**input netto** (o somma pesata) del neurone  $j$ .
- $\text{out}_j$  rappresenta l'**output attivato** del neurone  $j$ .

Di conseguenza, l'equazione calcola la correzione del peso moltiplicando tre derivate parziali che agiscono a catena:

1.  $\frac{\partial \mathcal{L}}{\partial \text{out}_j}$  valuta come varia l'errore totale in funzione dell'output finale del neurone  $j$ .
2.  $\frac{\partial \text{out}_j}{\partial \text{net}_j}$  rappresenta la pendenza della funzione di attivazione, misurando come varia l'output del neurone al variare del suo input grezzo.
3.  $\frac{\partial \text{net}_j}{\partial w_{ij}}$  quantifica l'impatto del singolo peso  $w_{ij}$  sull'input netto del neurone.

Attraverso questo circuito di feedback correttivo, i pesi vengono aggiornati iterativamente: le connessioni che portano a previsioni corrette vengono rafforzate, mentre quelle che contribuiscono all'errore vengono attenuate.

### 2.4 Physics-Informed Neural Networks (PINN)

Un approccio al Machine Learning basato esclusivamente sulla minimizzazione dell'errore sui dati ( $\mathcal{L}_{data}$ ) presenta un limite strutturale noto come problema della scatola nera (black-box). In tale scenario, la rete potrebbe individuare un'ottima approssimazione

statistica dei campioni di addestramento, ma generare soluzioni fisicamente inconsistenti, ad esempio predicendo campi di velocità che violano il principio di conservazione della massa. Per ovviare a questo problema, nella fluidodinamica moderna si adotta il paradigma delle Physics-Informed Neural Networks (PINN) [21]. In questa architettura, la fisica del problema non è solo un riferimento esterno, ma viene codificata direttamente all'interno della Loss Function Composta. In questo modo, il modello è costretto a rispettare le leggi della fisica che descrivono il fenomeno, nel caso della fluidodinamica sono le equazioni differenziali di Navier-Stokes. Nelle architetture più avanzate, come le Physics-Informed Neural Networks (PINNs), il livello di output riveste un ruolo duale. Oltre a fornire la previsione, i valori di  $u, v, p$  vengono derivati rispetto agli input  $(x, y)$  mediante la Differenziazione Automatica. Questo permette di verificare in tempo reale il rispetto dei vincoli fisici, come l'equazione di continuità per un fluido incomprimibile:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

Qualora il residuo di tale equazione sia diverso da zero, la rete riceve un segnale di errore aggiuntivo, forzando i parametri interni a convergere verso una soluzione che sia non solo statisticamente coerente con i dati, ma anche fisicamente valida.

## 2.5 Tipologie di architetture e algoritmi di apprendimento

Le reti neurali possono essere categorizzate in base alla topologia delle connessioni e alle modalità con cui l'informazione fluisce dai nodi di input a quelli di output. La scelta dell'architettura è strettamente legata alla natura dei dati e alla struttura spaziale o temporale del problema fisico analizzato[20].

### 2.5.1 Reti Feedforward (FFNN)

Le reti Feedforward rappresentano l'architettura più lineare: l'informazione viaggia esclusivamente in una direzione, dall'input verso l'output, senza cicli o connessioni ricorrenti. Ogni strato è densamente connesso al successivo, motivo per cui queste reti sono spesso chiamate Fully Connected. In ingegneria, le FFNN sono utilizzate come approssimatori puntuali di funzioni. Se applicate alla CFD, esse possono mappare una coordinata  $(x, y)$  su un valore di pressione  $p$ , trattando però ogni punto come un'entità isolata. Il loro limite principale risiede nell'incapacità di cogliere relazioni spaziali complesse tra elementi vicini o di elaborare dati strutturati come mesh e griglie.

### 2.5.2 Reti Neurali Ricorrenti (RNN) e LSTM

Per la modellazione di fenomeni fisici transitori (unsteady), dove lo stato del fluido al tempo  $t$  dipende fortemente dagli istanti precedenti, si utilizzano le Reti Neurali Ricorrenti. Queste possiedono cicli interni che permettono all'informazione di persistere nel tempo. L'evoluzione più robusta di questa categoria è rappresentata dalle reti LSTM (Long Short-Term Memory), capaci di gestire memorie a lungo termine e di prevenire il problema della scomparsa del gradiente, rendendole ideali per simulazioni temporali complesse.

### 2.5.3 Reti Neurali Convoluzionali (CNN)

Le CNN hanno rivoluzionato il trattamento dei dati strutturati in griglie regolari. Attraverso l'uso di filtri (kernel) che scorrono sopra i dati, queste reti estraggono pattern spaziali come bordi, gradienti e strutture vorticose. Ha quindi la capacità di individuare le diverse caratteristiche di un'immagine, dopo aver scartato le caratteristiche meno importanti (rumore), le caratteristiche più rilevanti passano al nodo successivo. I dati fin qua ottenuti vengono ricostruiti per piccole porzioni della mappa caratteristiche. Dopo cicli di analisi e ricostruzione, la rete restituisce una predizione sull'immagine osservata, che verrà usata come probabilità per ogni possibile classificazione. Questa metodologia può essere usata per la classificazione di immagini e individuazione della posizione degli oggetti. In ambito fluidodinamico, un campo di moto può essere interpretato come un'immagine. Le CNN eccellono nell'identificare strutture coerenti del flusso, ma presentano un limite invalicabile: richiedono griglie euclidee regolari. Questo le rende difficilmente applicabili alle mesh non strutturate tipiche della CFD industriale, dove la connettività tra le celle è variabile.

### 2.5.4 Graph Neural Networks (GNN)

Le Graph Neural Networks rappresentano la risposta al limite strutturale delle CNN. Esse operano direttamente su domini non euclidei, dove i dati sono rappresentati come grafi (nodi e archi). Nel contesto della presente tesi, la mesh di calcolo viene trattata come un grafo in cui i nodi rappresentano i centroidi o i vertici delle celle e gli archi le interconnessioni fisiche tra di essi. Attraverso il meccanismo di Message Passing, ogni nodo aggrega le informazioni provenienti dai vicini immediati e aggiorna il proprio stato interno. Ripetendo questo processo attraverso più strati, la rete "diffonde" l'informazione spaziale in tutto il dominio, permettendo di mappare la geometria complessa del problema direttamente sulle variabili fisiche (ad esempio

pressione e velocità). Questo approccio garantisce predizioni ultra-rapide rispettando fedelmente la topologia originale del problema fisico.

### 3. METODOLOGIA DELLE GRAPH NEURAL NETWORKS E INTEGRAZIONE DI MESHGRAPHNET

---

#### 3.1 Reti Neurali a Grafo (GNN)

Le **Graph Neural Networks (GNN)** rappresentano una classe di architetture di Deep Learning progettate per operare direttamente su dati strutturati come grafi. A differenza delle reti neurali classiche, le GNN non sono vincolate a domini euclidei regolari; tale caratteristica le rende lo strumento ideale per l'analisi di mesh computazionali in ambito fluidodinamico, dove la geometria del dominio è intrinsecamente non strutturata.

##### 3.1.1 Obiettivi del capitolo

Il presente capitolo si pone l'obiettivo di fornire una panoramica concettuale e funzionale delle Reti Neurali a Grafo, privilegiando la comprensione del loro funzionamento logico e delle loro potenzialità applicative rispetto a una trattazione puramente analitica, che esulerebbe dalle finalità del presente lavoro. Si intende illustrare come la topologia di una mesh fisica possa essere tradotta in un linguaggio matematico comprensibile per un algoritmo di apprendimento, permettendo la modellazione di fenomeni complessi.

##### 3.1.2 Il paradigma MeshGraphNet

Tra le architetture allo stato dell'arte, un ruolo di rilievo è ricoperto da **MeshGraphNet**[22]. Si tratta di un modello di rete neurale a grafi specificamente progettato per la simulazione di sistemi fisici, in cui la struttura della mesh viene convertita fedelmente in un grafo.

L'idea cardine di questo modello risiede nell'impiego del paradigma di *message passing* tra i nodi: questo meccanismo permette di propagare le informazioni riguardanti gli stati fisici — quali velocità, pressione o temperatura — attraverso l'intero dominio nel corso del tempo. Grazie a questo approccio, MeshGraphNet è in grado di catturare efficacemente sia le interazioni locali che quelle globali all'interno di sistemi governati da equazioni differenziali alle derivate parziali (PDE), offrendo una flessibilità che i solutori tradizionali o le reti neurali classiche difficilmente riescono a eguagliare con la medesima efficienza computazionale.

### 3.2 Fondamenti di teoria dei grafi

Un grafo rappresenta una struttura matematica astratta utilizzata per modellare relazioni e interazioni tra entità discrete[23]. Formalmente, un grafo  $G$  è definito come una tupla composta da un insieme di nodi (o vertici)  $V$  e un insieme di archi (o *edges*)  $E$ :

$$G = (V, E) \quad (3.1)$$

Per arricchire la descrizione del sistema, è possibile associare degli attributi (o *feature*) sia ai nodi che agli archi:

- **Attributi dei nodi:** Definiti dalla matrice  $\mathbf{X} \in \mathbb{R}^{n \times d}$ , dove  $n$  è il numero di nodi e  $d$  la dimensione dello spazio delle caratteristiche. Il vettore  $\mathbf{h}_v \in \mathbb{R}^d$  rappresenta le caratteristiche specifiche del nodo  $v$ . Se, ad esempio, per ogni nodo si conoscono le coordinate spaziali  $(x, y, z)$ , la temperatura ( $T$ ) e la densità ( $\rho$ ), allora  $d = 5$ . La matrice  $\mathbf{X}$  sarà una tabella con  $n$  righe e 5 colonne.
- **Attributi degli archi:** Definiti dalla matrice  $\mathbf{X}_e \in \mathbb{R}^{m \times c}$ , dove  $m$  è il numero di archi e  $c$  la dimensione delle relative caratteristiche. Il vettore  $\mathbf{e}_{ij} \in \mathbb{R}^c$  codifica le informazioni relative alla connessione tra il nodo  $i$  e il nodo  $j$ . Come informazioni, l'arco potrebbe contenere la distanza euclidea tra due centroidi della mesh, il versore normale della faccia che li separa o il coefficiente di diffusione. Questi dati aiutano la rete a capire la direzione e l'intensità degli scambi di quantità di moto o calore.

A seconda della natura delle relazioni, i grafi possono essere classificati in due categorie principali:

- **Diretti (Orientati):** In cui ogni arco possiede un verso specifico, indicando una relazione unidirezionale. Questa struttura è fondamentale per modellare flussi di informazioni o quantità fisiche dotate di orientamento.
- **Non diretti:** In cui la relazione tra due nodi è intrinsecamente bidirezionale. Un grafo non diretto può essere interpretato come un caso speciale di grafo diretto in cui, per ogni connessione, esiste una coppia di archi con direzioni opposte. Algebricamente, un grafo è non diretto se e solo se la sua matrice di adiacenza è simmetrica ( $\mathbf{A} = \mathbf{A}^T$ ).

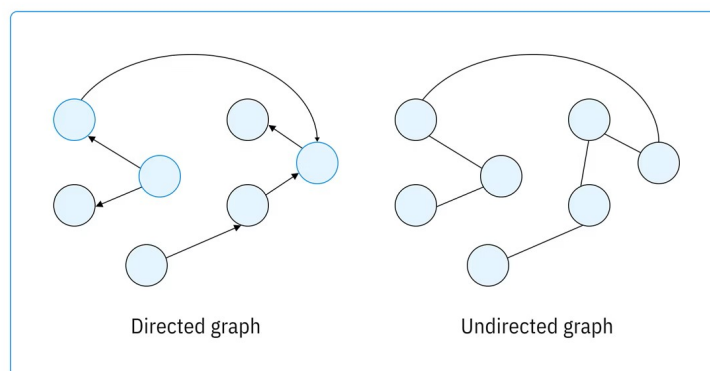


Figura 3.1: Rappresentazione schematica di grafi diretti e non diretti.

### La matrice di adiacenza

La connettività di un grafo con  $n$  nodi può essere sintetizzata algebricamente attraverso la matrice di adiacenza  $\mathbf{A} \in \{0, 1\}^{n \times n}$ . Un elemento  $A_{ij} = 1$  indica la presenza di un arco tra i nodi  $i$  e  $j$ , mentre  $A_{ij} = 0$  indica l'assenza di connessione. In termini matematici la matrice di adiacenza  $\mathbf{A}$  è una matrice  $n \times n$  tale per cui  $A_{ij} = 1$  se  $e_{ij} \in E$  e  $A_{ij} = 0$  se  $e_{ij} \notin E$ . [24]. In ambito GNN, la matrice non è solo una tabella

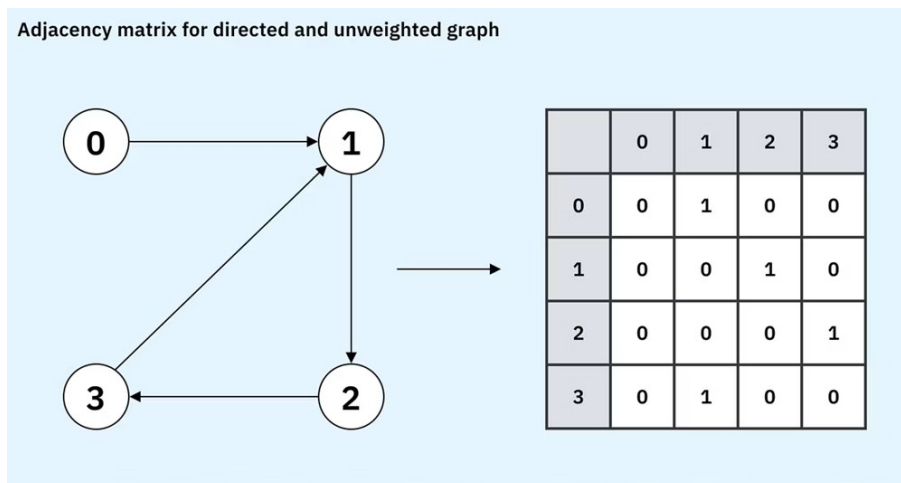


Figura 3.2: Rappresentazione matrice di adiacenza

statica di dati, ma agisce attivamente come operatore di diffusione. Essa guida il modo in cui le informazioni fisiche si propagano localmente (tra celle adiacenti) durante un singolo livello di message passing. Moltiplicando iterativamente lo stato dei nodi per tale matrice attraverso i vari strati della rete, diventa possibile diffondere l'informazione a distanze progressivamente maggiori, rispettando sempre la reale topologia del dominio fisico..

## Invarianza alla permutazione

Una proprietà fondamentale delle GNN è l'invarianza alla permutazione. Poiché in un grafo (e in una mesh) l'ordine dei nodi nella memoria del computer è arbitrario, la rete deve produrre lo stesso output indipendentemente da come i nodi vengono indicizzati. Questo garantisce che il modello apprenda le relazioni strutturali e non una sequenza numerica casuale.

### 3.2.1 Rappresentazione della mesh in MeshGraphNet

Nel framework di *MeshGraphNet*[22], la mesh di calcolo viene trattata come un grafo  $G = (V, E)$ , dove la topologia del dominio fisico viene preservata attraverso la seguente mappatura:

- **Nodi ( $V$ ):** L'insieme dei nodi corrisponde ai vertici o ai centoridi delle celle della mesh. Ogni nodo  $i \in V$  è caratterizzato da un vettore di *feature*  $\mathbf{h}_i \in \mathbb{R}^d$ , che memorizza le grandezze fisiche rilevanti quali la velocità  $\mathbf{v}_i \in \mathbb{R}^3$ , la pressione  $p_i \in \mathbb{R}$  o altre variabili di stato (ad esempio temperatura, tensori degli sforzi o flag relativi alle condizioni al contorno).
- **Archi ( $E$ ):** Ogni arco  $(i, j) \in E$  possiede un vettore di *feature*  $\mathbf{e}_{ij} \in \mathbb{R}^k$ , che codifica le relazioni spaziali tra i nodi nel dominio discretizzato. Queste includono comunemente il vettore posizione relativa  $\mathbf{x}_j - \mathbf{x}_i$  e la relativa distanza euclidea  $\|\mathbf{x}_j - \mathbf{x}_i\|$ , parametri essenziali affinché il modello possa apprendere l'invarianza per traslazione e la metrica del problema.

## 3.3 Il meccanismo di Message Passing

Il cuore operativo delle GNN è il meccanismo di *Message Passing* [22, 25], un processo iterativo che permette ai nodi di scambiare informazioni con il proprio vicinato locale  $\mathcal{N}(i)$ . Per ogni strato  $t$  della rete, il processo si articola in tre fasi principali:

1. **Fase di messaggio (Message Step):** Per ogni arco  $(i, j)$ , viene generato un messaggio  $\mathbf{m}_{ij}$  basandosi sugli stati dei nodi collegati e sulle proprietà dell'arco stesso:

$$\mathbf{m}_{ij}^{(t)} = \phi_e \left( \mathbf{h}_i^{(t)}, \mathbf{h}_j^{(t)}, \mathbf{e}_{ij} \right) \quad (3.2)$$

I singoli termini che compongono la formula sono definiti come segue:

- $\mathbf{m}_{ij}^{(t)}$ : rappresenta il **vettore messaggio**, ovvero l'informazione inviata dal nodo  $j$  al nodo  $i$  allo strato  $t$ . In un contesto fluidodinamico, esso può essere interpretato come il contributo del flusso che attraversa l'interfaccia tra due celle computazionali.
- $(t)$ : l'apice tra parentesi indica lo **strato della rete** (*layer*) o l'iterazione attuale del processo di *message passing*. Trattandosi di un processo iterativo, le informazioni vengono raffinate e si evolvono a ogni passaggio attraverso la struttura neurale.
- $\phi_e$ : indica la **funzione messaggio** (o *edge function*). Rappresenta l'operatore matematico preposto a combinare i dati dei nodi connessi per generare il messaggio; solitamente, tale funzione è implementata tramite un *Multi-Layer Perceptron* (MLP).
- $\mathbf{h}_i^{(t)}$  e  $\mathbf{h}_j^{(t)}$ : sono i **vettori di stato** (*hidden features*) rispettivamente del nodo ricevente  $i$  e del nodo sorgente  $j$ . Essi contengono le variabili fisiche locali (quali pressione e velocità) o le relative rappresentazioni latenti estratte dalla rete durante l'addestramento.
- $\mathbf{e}_{ij}$ : rappresenta le **caratteristiche dell'arco** (*edge features*). Si tratta di informazioni statiche relative alla connessione che rimangono invariate tra i diversi strati.

2. **Fase di aggregazione (Aggregation Step)**: Il nodo  $i$  raccoglie i messaggi provenienti da tutti i vicini  $j \in \mathcal{N}(i)$ , dove  $\mathcal{N}(i)$  denota l'insieme dei nodi nelle vicinanze di  $i$ :

$$\mathbf{M}_i^{(t)} = \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}^{(t)} \quad (3.3)$$

Questo passaggio rappresenta l'equivalente neurale dell'integrazione dei flussi sulle facce di una cella, operazione tipica del metodo ai Volumi Finiti .

3. **Fase di aggiornamento (Update Step)**: Il nodo aggiorna il proprio stato combinando l'informazione attuale con il messaggio aggregato:

$$\mathbf{h}_i^{(t+1)} = \phi_v \left( \mathbf{h}_i^{(t)}, \mathbf{M}_i^{(t)} \right) \quad (3.4)$$

Dove  $\mathbf{h}_i^{(t+1)}$  è il vettore delle *feature* aggiornato. Tale operazione consente a ogni nodo di incorporare informazioni dal proprio vicinato, catturando così le interazioni locali e l'evoluzione dello stato fisico.

### 3.4 Stratificazione e propagazione dell’informazione

Il processo di *message passing* sopra descritto viene ripetuto attraverso molteplici strati della rete neurale utilizzando parametri addestrabili distinti per ogni livello. Con ogni strato successivo, l’informazione viene propagata attraverso regioni progressivamente più ampie del grafo[25].

Dopo  $L$  strati di *message passing*, ogni nodo  $i$  ha aggregato informazioni provenienti da nodi distanti fino a  $L$  “salti” (*hops*) nel grafo. Questa architettura stratificata permette a *MeshGraphNet* di catturare sia le interazioni locali (tra celle adiacenti) che le dipendenze globali del sistema fisico. La regola di aggiornamento globale per il nodo  $i$  allo strato  $t$  può essere sintetizzata nella seguente espressione:

$$\mathbf{h}_i^{(t+1)} = \phi_u^{(t)} \left( \mathbf{h}_i^{(t)}, \sum_{j \in \mathcal{N}(i)} \phi_m^{(t)} \left( \mathbf{h}_i^{(t)}, \mathbf{h}_j^{(t)}, \mathbf{e}_{ij} \right) \right) \quad (3.5)$$

dove  $\phi_m^{(t)}$  e  $\phi_u^{(t)}$  sono le funzioni apprendibili (reti neurali) specifiche per lo strato  $t$ . Al termine degli  $L$  strati, il grafo computazionale per l’aggiornamento di ciascun nodo risulta autocontenuto: sia i calcoli diretti (*forward*) che quelli di retro-propagazione (*backward*) dipendono esclusivamente dal suo intorno locale di  $L$  salti e dalle relative caratteristiche di input associate.

Dopo  $T$  iterazioni (o strati), lo stato finale  $\mathbf{h}_i^{(T)}$  avrà incorporato informazioni provenienti da un intorno di raggio  $L$ , permettendo alla rete di catturare fenomeni globali partendo da interazioni puramente locali.

### 3.5 Analisi e task predittivi

Le GNN consentono di operare a diversi livelli di astrazione a seconda dell’obiettivo ingegneristico e della natura del problema fisico analizzato[23]:

**Task a livello di nodo (*Node-level*):** Gli output si riferiscono alle attività di regressione e classificazione dei nodi. Rappresenta l’approccio più diffuso nell’ambito della fluidodinamica computazionale. L’obiettivo è associare a ogni singola cella o nodo della mesh il valore predetto di una variabile fisica puntuale. In questo caso, l’output della GNN è un campo scalare o vettoriale distribuito su tutto il dominio, come il campo di pressione  $p$  o il vettore velocità  $\mathbf{u}$ .

**Task a livello di arco (*Edge-level*):** Gli output riguardano le attività di classificazione dei bordi e di predizione dei collegamenti. Questo livello riguarda la predizione

di grandezze definite sulle interfacce tra gli elementi del volume di controllo. Viene impiegato per stimare flussi di scambio o interazioni locali. Esempi tipici in ambito termofluidodinamico includono la stima dei flussi di calore attraverso una superficie o il calcolo degli sforzi di taglio (*wall shear stress*) in prossimità delle pareti solide.

**Task a livello di grafo (*Graph-level*):** In questo scenario, le informazioni locali estratte dai singoli nodi vengono aggregate attraverso un'operazione di *global pooling* (o *readout*) per ottenere un unico valore rappresentativo dell'intero dominio. In ambito aerodinamico, ciò risulta fondamentale per la previsione di coefficienti globali di performance, quali il coefficiente di portanza ( $C_L$ ) o di resistenza ( $C_D$ ) relativi a un profilo alare o a un corpo immerso.

### 3.6 Spaider e l'integrazione della X-MeshGraphNet

La rete neurale implementata dalla start-up *Spaider* si basa sull'architettura **MeshGraphNet**, tra le architetture allo stato dell'arte per la risoluzione di problemi fisici complessi tramite *deep learning*. Tuttavia, questi modelli presentano limitazioni intrinseche, tra cui criticità nella scalabilità e difficoltà nella gestione delle interazioni a lungo raggio, provocando una possibile perdita di accuratezza nelle strutture multiscala.

Come verrà illustrato nei capitoli successivi, il problema analizzato in questa tesi coinvolge mesh composte da milioni di celle; l'impiego di una GNN standard comporterebbe la necessità di risorse hardware proibitive a causa dell'elevato numero di passaggi di messaggio necessari per propagare le informazioni lungo l'intero dominio computazionale. Per affrontare questa sfida, è stata introdotta **X-MeshGraphNet**[25], un'estensione scalabile e multiscala progettata per superare i limiti strutturali del modello originale.

In questo approccio multiscala, per quanto concerne la struttura del grafo non viene mantenuta solo la connettività locale della mesh, ma vengono aggiunte connessioni di scala più ampia. Nodi collegati a distanze maggiori permettono di catturare interazioni a diverse scale di grandezza, garantendo un addestramento più accurato. Nelle simulazioni multiscala, sono necessari diversi livelli di fedeltà per catturare sia le dinamiche globali che quelle locali .

X-MeshGraphNet introduce un processo di generazione di grafi multiscala in cui le nuvole di punti grezze vengono raffinate iterativamente. Ogni livello della gerarchia rappresenta un sottoinsieme del precedente, consentendo al modello di catturare le interazioni in modo efficiente. Nel caso dell'aerodinamica di un veicolo,

per far sì che l'informazione viaggi, ad esempio, dal muso alla coda, servirebbero troppi passaggi di messaggi; il sistema multi-scala crea invece una gerarchia di visioni che permette di elaborare simultaneamente il dettaglio e l'insieme.

### 3.6.1 Processo tecnico di raffinamento

Basandosi sul caso studio dell'aerodinamica automobilistica presente nel documento redatto da Mohammad Amin Nabian et al.(2024), il processo si articola in tre fasi:

1. **Creazione della gerarchia (Campionamento a livelli):** Il sistema genera diverse versioni della nuvola di punti a partire dal file CAD originale:

- **Livello 1 (Grossolano):** Una nuvola con risoluzione ridotta (es. 500.000 punti) per catturare la sagoma generale.
- **Livello 2 (Medio):** Una risoluzione intermedia (es. 1 milione di punti), dove i punti del Livello 1 sono un sottoinsieme di quelli del Livello 2.
- **Livello 3 (Fine):** Massima risoluzione (es. 2 milioni di punti) per catturare i dettagli minimi della geometria.

2. **Connessioni locali e a lungo raggio:** La rete stabilisce gli archi in modo strategico:

- **Nei livelli fini:** I nodi sono molto vicini, permettendo alla rete di apprendere le interazioni locali, come i vortici d'aria dietro piccoli componenti.
- **Nei livelli grossolani:** Poiché i punti sono radi, un singolo salto tra nodi copre una distanza fisica maggiore, permettendo di catturare il comportamento del flusso globale.

3. **Raffinamento iterativo:** L'informazione fluisce tra i livelli. La rete inizia l'elaborazione sui livelli grossolani per definire la trama generale del flusso; questa informazione viene poi passata ai livelli fini, che utilizzano il contesto globale per raffinare la previsione locale.

### 3.6.2 Vantaggi del metodo

L'integrazione di X-MeshGraphNet introduce benefici rivoluzionari per la simulazione fisica:

- **Efficienza:** La visione a lungo raggio è garantita da pochi nodi strategici nei livelli alti, evitando miliardi di connessioni superflue.

- **Precisione:** L'abbinamento tra scale diverse riduce l'errore di validazione rispetto ai grafi a livello singolo.
- **Velocità:** Permette di ottenere precisione paragonabile alla CFD professionale in una frazione del tempo, eliminando la necessità di gestire mesh multiple a diverse risoluzioni.

## 4. PRE-ELABORAZIONE DEI DATI E ADDESTRAMENTO

---

L'efficacia di un modello di Machine Learning è strettamente correlata alla qualità dei dati. Questi modelli, composti da molteplici livelli di elaborazione, apprendono rappresentazioni dei dati con diversi gradi di astrazione [20]. Prima di procedere all'addestramento vero e proprio, le informazioni devono attraversare una pipeline di pre-elaborazione strutturata, mirata a eliminare rumore e bias che potrebbero compromettere l'accuratezza delle previsioni. Mentre dataset di alto livello garantiscono output affidabili in molteplici domini applicativi, la varietà dei dati è altrettanto determinante: solo una base dati eterogenea permette al modello di sviluppare una reale capacità di generalizzazione. In assenza di diversificazione, si incorre infatti nell'overfitting, un errore sistematico in cui l'algoritmo memorizza le specificità del dataset di addestramento anziché apprenderne le logiche [26].

### 4.1 Pulizia dei dati e suddivisione del dataset

Il *dataset* grezzo (*raw data*) derivante da solutori CFD presenta frequentemente criticità intrinseche, quali valori nulli, ridondanze informative o errori di campionamento dovuti a instabilità numeriche locali. La fase di *data cleaning* è pertanto indispensabile per sanare queste imperfezioni prima che il dato venga somministrato al modello.

#### 4.1.1 Data cleaning in ambito fluidodinamico

Nello specifico contesto della fluidodinamica computazionale, la pulizia del dato implica:

- **Smorzamento delle oscillazioni:** Calcolo della media sulle ultime iterazioni della simulazione.
- **Rimozione dei non-convergenti:** Esclusione delle simulazioni che non hanno raggiunto i criteri di convergenza prefissati, onde evitare che la rete apprenda campi fisici instabili.
- **Gestione delle singolarità:** Correzione o rimozione di valori non fisici (es. pressioni negative estreme o velocità divergenti) in prossimità di spigoli vivi o singolarità della mesh.
- **Filtraggio del rumore:** Eliminazione di eventuali fluttuazioni numeriche ad alta frequenza che non rappresentano la fisica reale del problema ma sono artefatti della discretizzazione.

### 4.1.2 Partizionamento del dataset

Una volta ripulito, il dataset viene ripartito in tre frazioni distinte e indipendenti. Questa separazione è cruciale per prevenire il *data leakage* e per monitorare rigorosamente la capacità di generalizzazione del modello:

1. **Training Set (Set di addestramento):** Rappresenta la porzione maggiore del dataset (tipicamente il 70-80%). Viene utilizzato attivamente dall'algoritmo di ottimizzazione per regolare i parametri interni (pesi e *bias*) e trovare la migliore corrispondenza tra previsioni e dati reali.
2. **Validation Set (Set di convalida):** Impiegato durante la fase di addestramento per l'ottimizzazione degli iperparametri e per monitorare l'insorgenza di fenomeni di *overfitting*. Se l'errore sul *validation set* inizia ad aumentare mentre quello sul *training set* continua a scendere, il processo viene interrotto (*Early Stopping*).
3. **Test Set:** Riservato esclusivamente alla valutazione finale e imparziale. Questo set non entra mai in contatto con il modello durante l'addestramento, fornendo una stima realistica delle prestazioni della GNN su geometrie o condizioni di flusso completamente inedite.

## 4.2 Normalizzazione delle feature in ambito CFD

Un passaggio critico e particolarmente rilevante nel contesto della fluidodinamica computazionale è la **normalizzazione** (o *scaling*) delle variabili. In ambito fisico, i dati di input presentano spesso ordini di grandezza estremamente divergenti: si consideri, a titolo di esempio, la discrepanza numerica tra i valori di pressione espressi in Pascal ( $10^5$ ) e le componenti della velocità espresse in m/s. Se forniti al modello senza alcun trattamento, i valori con magnitudo maggiore dominerebbero il calcolo della *loss function* e l'aggiornamento dei gradienti, rendendo l'addestramento instabile o impedendo del tutto la convergenza numerica.

Le tecniche di riscaldamento più diffuse in letteratura sono:

- **Min-Max scaling:** Trasla e comprime i dati in un intervallo predefinito, solitamente  $[0, 1]$  o  $[-1, 1]$ , mediante la formula:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4.1)$$

Questa tecnica è utile quando i dati hanno confini ben definiti e non presentano *outlier* estremi che potrebbero "schiacciare" la distribuzione degli altri valori.

- **Standardizzazione (Z-score):** Trasforma i dati affinché presentino media nulla ( $\mu = 0$ ) e varianza unitaria ( $\sigma = 1$ ):

$$z = \frac{x - \mu}{\sigma} \quad (4.2)$$

La standardizzazione è spesso preferita in ambito CFD poiché risulta più robusta in presenza di valori anomali (come picchi di pressione locali o singolarità numeriche) e facilita la dinamica di apprendimento degli attivatori non lineari.

Garantire che ogni variabile contribuisca equamente all'apprendimento non solo accelera la discesa del gradiente, ma migliora drasticamente la capacità della rete di mappare correttamente le correlazioni fisiche tra i diversi campi scalari e vettoriali del fluido.

### 4.3 Data augmentation

Il processo di *Data Augmentation* consiste nella generazione artificiale di nuovi campioni a partire dai dati esistenti, con l'obiettivo principale di incrementare la robustezza e la capacità di generalizzazione dei modelli di *machine learning*. L'addestramento di architetture profonde richiede set di dati ampi e variegati; tuttavia, il reperimento di simulazioni CFD reali sufficientemente diversificate può risultare oneroso a causa dell'elevato costo computazionale dei solutori ad alta fedeltà o delle limitazioni nell'accesso a database proprietari[27].

L'incremento dei dati permette di ovviare a queste limitazioni apportando modifiche controllate ai campioni originali. L'applicazione strategica del *Data Augmentation*[27] non solo amplia artificialmente la dimensione del *training set*, ma agisce come una forma di regolarizzazione, riducendo drasticamente il rischio di *overfitting* e garantendo che il modello apprenda le leggi fisiche sottostanti piuttosto che le singole configurazioni geometriche presenti nel dataset iniziale.

### 4.4 Parametri e iperparametri del modello

Il funzionamento di un modello di *Machine Learning* è regolato da due categorie distinte di variabili: i **parametri**, che vengono appresi automaticamente durante l'addestramento, e gli **iperparametri**, definiti a priori dal progettista.

#### 4.4.1 Parametri del modello

I parametri rappresentano le variabili interne che il modello modifica iterativamente per minimizzare l'errore tra le proprie previsioni e i dati reali (fase di *fitting*). La loro

quantità e configurazione determinano la capacità del modello di catturare relazioni complesse all'interno del dataset. In una rete neurale, i parametri principali sono:

- **Pesi ( $w$ ):** Quantificano la forza delle connessioni tra i neuroni dei diversi strati. Essi determinano l'influenza che un determinato input esercita sull'output finale. Inizialmente impostati su valori casuali di piccola entità, i pesi vengono aggiornati durante l'addestramento affinché la rete impari a riconoscere pattern e correlazioni tra le variabili.
- **Bias ( $b$ ):** Rappresentano termini di soglia o di *offset* che permettono di traslare la funzione di attivazione indipendentemente dagli input. Il *bias* garantisce che il neurone possa generare un output significativo anche in presenza di input nulli o molto bassi, aumentando la flessibilità e la capacità di generalizzazione del modello.

Mentre un numero elevato di parametri permette di modellare fenomeni estremamente complessi, un'eccessiva complessità parametrica rispetto alla quantità di dati disponibili può condurre al fenomeno dell'*overfitting*.

#### 4.4.2 Iperparametri

Gli iperparametri sono variabili configurate prima dell'inizio del processo di apprendimento; essi non vengono appresi dai dati, ma guidano e vincolano l'algoritmo di ottimizzazione.

Tra i principali si annoverano:

- **Learning rate (Tasso di apprendimento):** Esso definisce l'ampiezza del passo compiuto dall'algoritmo di ottimizzazione (tipicamente basato sul gradiente) verso il minimo della funzione di perdita (loss function). Nel contesto specifico delle GNN, il learning rate regola l'entità degli aggiornamenti apportati ai pesi delle matrici di trasformazione durante la fase di backpropagation. Un valore troppo elevato può causare instabilità o divergenza, mentre un valore troppo basso rallenta eccessivamente la convergenza.
- **Batch size (Dimensione del lotto):** Rappresenta la suddivisione del training set e indica il numero di campioni di addestramento analizzati dal modello prima di procedere all'aggiornamento dei parametri. L'uso di *mini-batch* permette di bilanciare l'efficienza computazionale e la stabilità del gradiente.
- **Numero di epoche:** Rappresenta il numero totale di passaggi completi dell'intero set di dati attraverso il modello durante l'addestramento.

- **Architettura della rete:** Include la profondità del modello (numero di strati o *layers*) e la sua ampiezza (numero di neuroni o canali per strato). Nel contesto delle GNN, questo include anche il numero di passaggi di propagazione dei messaggi (*message passing steps*) tra i nodi del grafo.

#### 4.5 Il Processo di addestramento: forward pass e funzioni di attivazione

L'addestramento di una rete neurale è un processo iterativo finalizzato all'ottimizzazione dei parametri interni per minimizzare l'errore predittivo. La fase iniziale di ogni iterazione è rappresentata dal *Forward Pass* (o propagazione in avanti).

##### 4.5.1 Il forward pass

Durante il *forward pass*, i dati di input fluiscono attraverso i diversi strati della rete. In ogni neurone, i dati vengono combinati linearmente con i pesi e i bias associati secondo l'operazione:

$$z = \sum_{i=1}^n w_i x_i + b \quad (4.3)$$

dove  $w_i$  rappresenta i pesi,  $x_i$  rappresenta il dato in input e  $b$  il bias. Il risultato di questa combinazione lineare viene elaborato da una funzione di attivazione non lineare, producendo l'output del neurone che diventerà l'input per lo strato successivo. Questo processo prosegue fino allo strato di output, dove la rete genera una previsione. Tale valore viene infine confrontato con il dato reale (*ground truth*) tramite una *loss function*, che quantifica l'errore commesso.

##### 4.5.2 Le funzioni di attivazione

La funzione di attivazione determina se e in che misura il segnale ricevuto da un neurone debba essere trasmesso ai livelli successivi. Senza di essa, una rete neurale si comporterebbe come un semplice modello lineare, incapace di apprendere relazioni complesse. L'introduzione della non-linearità conferisce alle reti la proprietà di "approssimatori universali", caratteristica fondamentale nella CFD per modellare le forti non-linearità delle equazioni di Navier-Stokes, come i fenomeni turbolenti o gli elevati gradienti di pressione.

Le funzioni di attivazione più comuni includono:

- **ReLU (Rectified Linear Unit):** Standard per gli strati intermedi, definita come  $f(x) = \max(0, x)$ . Se il numero in ingresso ( $x$ ) è positivo, il neurone lo lascia passare esattamente così com'è. Se il numero in ingresso ( $x$ ) è negativo, il neurone lo azzerava, restituendo semplicemente un valore nullo.

- **Sigmoide:** Comprime l'output nell'intervallo  $[0, 1]$ , definita da  $\sigma(x) = \frac{1}{1+e^{-x}}$ . Oggi è meno usata negli strati nascosti a causa della saturazione dei gradienti.
- **Tanh (Tangente Iperbolica):** Produce output tra  $-1$  e  $1$ , centrando i dati intorno allo zero.
- **Softmax:** Utilizzata nello strato di output per problemi di classificazione, trasforma i valori in una distribuzione di probabilità (somma pari a 1).
- **Lineare:** Utilizzata tipicamente nello strato di output per problemi di regressione (comune nella CFD), dove non si desidera limitare l'intervallo del valore predetto.

## 4.6 La funzione di perdita (Loss Function)

La *Loss Function* rappresenta la metrica utilizzata per quantificare lo scostamento tra le previsioni del modello e i valori reali (*ground truth*). L'obiettivo dell'addestramento è la minimizzazione di tale funzione tramite la regolazione iterativa dei parametri. Matematicamente, se le previsioni sono accurate, il valore della perdita tende a zero. Il calcolo avviene solitamente su un *mini-batch* di dati, ottenendo un valore medio utilizzato per aggiornare i pesi.

Le funzioni di perdita si dividono in:

- **Funzioni di classificazione:** per variabili discrete.
- **Funzioni di regressione:** per variabili continue, fondamentali per le GNN applicate alla CFD per predire campi scalari o vettoriali (pressione, velocità, temperatura).

### 4.6.1 Principali funzioni di perdita per la regressione

- **Errore quadratico medio (MSE - Mean Squared Error):** Nota come *L2 Loss*, calcola la media dei quadrati delle differenze:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.4)$$

Penalizza pesantemente gli *outlier* ed è ideale per alta precisione numerica.

- **Errore assoluto medio (MAE - Mean Absolute Error):** Definita come *L1 Loss*, misura la media delle differenze assolute:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4.5)$$

Risulta più robusto in presenza di valori anomali.

- **Root mean squared error (RMSE):** Rappresenta la radice quadrata dell'MSE:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (4.6)$$

Mantiene la stessa unità di misura della variabile *target* (es. *m/s*).

- **Mean squared logarithmic error (MSLE):** Utile per dati con ampi intervalli o crescita esponenziale:

$$MSLE = \frac{1}{n} \sum_{i=1}^n (\ln(1 + y_i) - \ln(1 + \hat{y}_i))^2 \quad (4.7)$$

#### 4.6.2 La funzione di perdita informata dalla fisica (Physics-Informed Loss)

Nella CFD, le sole metriche statistiche possono non garantire previsioni fisicamente plausibili. Si ricorre quindi al paradigma delle *Physics-Informed Neural Networks* (PINN)[21], dove la perdita totale è una combinazione convessa:

$$L_{total} = w_{data}L_{data} + w_{phys}L_{phys} \quad (4.8)$$

Dove:

- $L_{data}$ : Errore classico (es. MSE) rispetto ai dati di addestramento.
- $L_{phys}$ : Residuo delle equazioni differenziali alle derivate parziali (PDE). Per la CFD, impone leggi di conservazione come l'equazione di continuità:

$$\mathcal{R} = \nabla \cdot \mathbf{u} = 0 \quad (4.9)$$

Tale formulazione non si limita all'equazione di continuità, ma può integrare, a seconda delle necessità di modellazione, anche i residui delle equazioni della quantità di moto e dell'energia.

$$L_{phys} = \frac{1}{n} \sum_{i=1}^n |\mathcal{R}_i|^2 \quad (4.10)$$

## 4.7 Valutazione e generalizzazione: overfitting e underfitting

L'obiettivo ultimo dell'addestramento di un modello di *Machine Learning* non è la semplice precisione sui dati noti, ma la sua capacità di **generalizzazione**: l'attitudine a produrre previsioni accurate su dati nuovi e mai visti in precedenza. In questo contesto, si possono verificare due fenomeni degenerativi opposti: l' *underfitting* e l' *overfitting*[26].

### 4.7.1 Underfitting (Sottoadattamento)

L' *underfitting* si manifesta quando il modello è troppo semplice per catturare la relazione tra le variabili di input e output. Ciò può dipendere da un numero insufficiente di parametri, da una scarsa durata dell'addestramento o da un set di *feature* non rappresentativo, ciò significa un numero di variabili troppo basso per descrivere completamente il comportamento del caso in esame. Un modello in *underfitting* è caratterizzato da un elevato errore sia nel *training set* che nel *test set*, ed è accompagnato da un elevato *bias* (distorsione), ovvero quando il modello introduce assunzioni errate che gli impediscono di approssimare correttamente la funzione *target*. Nel grafico sottostante è rappresentato nel lato sinistro dove l'errore nel Test set non scende sufficientemente.

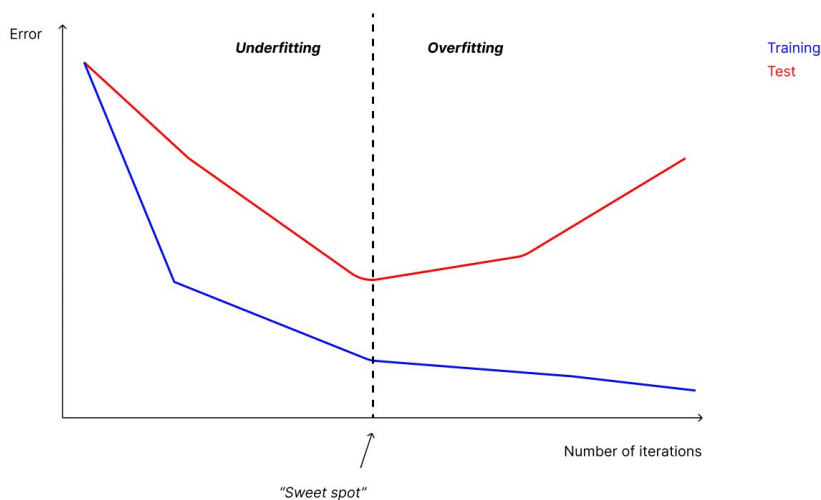


Figura 4.1: Rappresentazione grafica di overfitting e underfitting.

### 4.7.2 Overfitting (Sovraadattamento)

Al contrario, l' *overfitting* si verifica quando il modello presenta un'eccessiva complessità rispetto alla quantità di dati disponibili o viene addestrato per un tempo troppo

prolungato. In questa condizione, l'algoritmo non apprende le logiche sottostanti il fenomeno, ma inizia a "memorizzare" il rumore e le fluttuazioni casuali specifiche del *training set*. Un modello in *overfitting* presenta un basso errore di addestramento ma un elevato errore di test con un'alta varianza; in questo caso, le previsioni cambiano drasticamente al variare dei dati di input, rendendo il modello instabile. Il lato destro della Figura 4.1 illustra un classico scenario di sovraadattamento: mentre la curva dell'errore sul set di addestramento converge verso valori minimi, l'errore sul test set subisce un evidente incremento, segno che la rete sta perdendo la propria capacità di generalizzazione.

#### 4.7.3 Il Trade-off tra bias e varianza

Il processo di ottimizzazione cerca di trovare il punto di equilibrio ideale (*sweet spot*) tra *bias* e varianza. Aumentando la complessità del modello, il *bias* tende a diminuire ma la varianza aumenta, elevando il rischio di *overfitting*. La validazione incrociata (*Cross-Validation*) e il monitoraggio costante delle *performance* sul *validation set* sono gli strumenti principali per identificare questo punto critico.

#### 4.7.4 Tecniche per il contrasto dell'overfitting

Per mitigare l' *overfitting* e migliorare la robustezza del modello, si adottano diverse strategie di regolarizzazione e gestione dei dati[26]:

- **Arresto Anticipato (Early Stopping):** Consiste nell'interrompere l'addestramento nel momento in cui l'errore sul *validation set* smette di diminuire e inizia a risalire, segnale che il modello sta iniziando a memorizzare il rumore.
- **Regolarizzazione ( $L1$  e  $L2$ ):** Introduce una penalità nella *Loss Function* basata sulla grandezza dei pesi del modello. La regolarizzazione  $L2$  (*Ridge*) tende a mantenere i pesi piccoli, mentre la  $L1$  (*Lasso*) può azzerare i pesi meno rilevanti, effettuando una selezione automatica delle caratteristiche.
- **Dropout:** Tecnica specifica per le reti neurali che consiste nello "spegnere" casualmente una percentuale di neuroni durante ogni iterazione di addestramento. Questo impedisce ai neuroni di sviluppare co-dipendenze eccessive, forzando la rete a imparare *pattern* più robusti.
- **Data Augmentation:** Come ampiamente discusso nella Sezione 4.3, questa tecnica previene la memorizzazione del rumore fornendo nuovi campioni al modello. In ambito fluidodinamico, ciò si traduce tipicamente in operazioni

spaziali come rotazioni, riflessioni o scalature delle mesh computazionali, a patto che tali trasformazioni rispettino scrupolosamente le simmetrie fisiche del problema.

- **Ensembling:** Combina le previsioni di diversi modelli indipendenti (tramite tecniche di *Bagging* o *Boosting*) per ottenere un risultato finale più stabile e accurato, riducendo sensibilmente la varianza complessiva.

## 5. CASO DI ANALISI

---

### 5.1 Selezione del caso studio: analisi termica di un cabinato elettrico con trasformatore

Il caso selezionato per la validazione del modello è l'analisi termica semplificata di una cabina di trasformazione elettrica, all'interno della quale è posizionato un trasformatore soggetto a dissipazione di potenza. La scelta di tale scenario è dettata da precise esigenze di natura tecnica e strategica, finalizzate a testare i limiti e le potenzialità dell'architettura neurale in contesti non puramente aerodinamici.

#### 5.1.1 Motivazioni della scelta

La selezione di questo specifico dominio fisico risponde a tre obiettivi principali:

1. **Affidabilità e benchmarking:** Il caso era già stato oggetto di studi approfonditi all'interno della azienda, disponendo di una validazione numerica. Questo garantisce una base solida per iniziare la fase di *data augmentation*.
2. **Transizione verso la fisica multivariabile:** L'analisi termica di una cabina elettrica introduce una complessità fisica superiore, in cui i campi di moto (velocità e pressione) sono intrinsecamente accoppiati ai gradienti di temperatura. Questo scenario permette di valutare la capacità della GNN di gestire uno spazio delle *feature* più ampio.
3. **Stress-Test per l'architettura neurale:** Il caso è stato scelto per valutare la risposta della rete neurale a simulazioni fisiche più complesse e il vantaggio, in termini di tempo, che si ottiene utilizzando questa metodologia rispetto alle simulazioni numeriche tradizionali.

#### 5.1.2 Complessità del dominio

A differenza dei profili alari standard precedentemente testati dalla rete, la cabina elettrica presenta un'elevata densità di celle in prossimità delle zone di scambio termico, con una mesh che raggiunge circa i 4,5 milioni di elementi. Questa configurazione è ideale per validare la scalabilità della **X-MeshGraphNet**, verificando se l'approccio multiscala sia in grado di ricostruire correttamente i pennacchi termici e le zone di ristagno d'aria all'interno del vano. La densità della mesh è stata determinata a seguito di un'analisi di sensitività, ricercando un compromesso ottimale tra

l'accuratezza dei risultati numerici e i vincoli dimensionali imposti dall'architettura della rete neurale nella fase di elaborazione successiva.

## 5.2 Configurazione della simulazione e condizioni al contorno

Il dominio computazionale, illustrato nella figura sottostante, rappresenta un vano tecnico adibito a cabina elettrica. Al suo interno è alloggiato un trasformatore, modellato geometricamente come un solido semplificato, il quale agisce come sorgente termica primaria, dissipando un flusso di calore superficiale costante ( $q''$ ) su tutta la sua estensione.

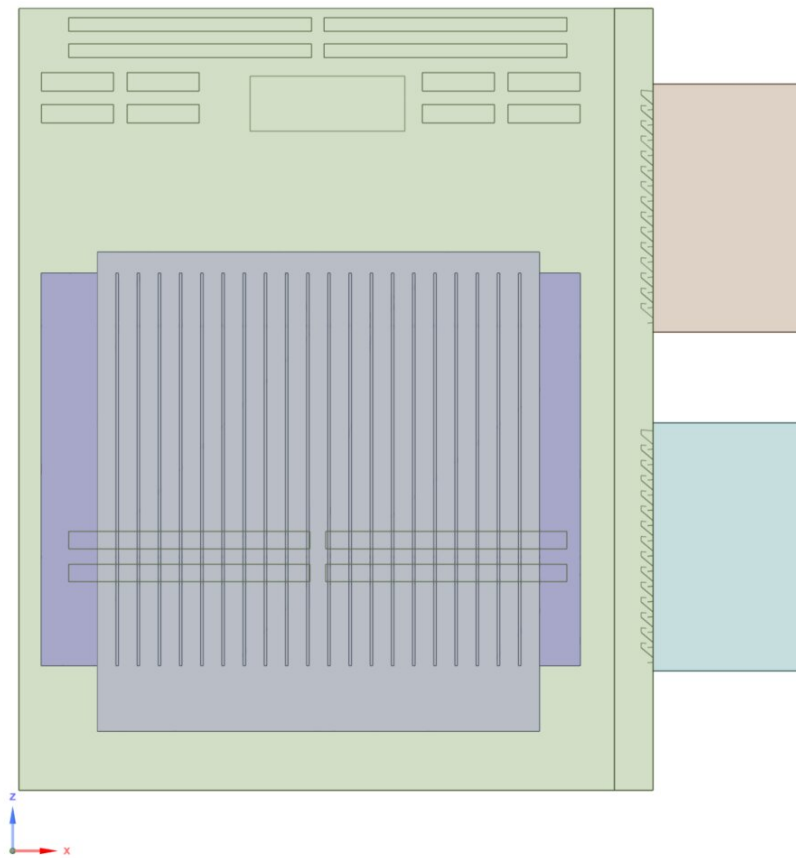


Figura 5.1: Rappresentazione grafica del dominio in analisi

La complessità fenomenologica del caso è definita da un insieme di condizioni al contorno (*Boundary Conditions*) eterogenee, strutturate per simulare con accuratezza un ambiente operativo reale. Le superfici dissipative, identificate nelle pareti laterali, gestiscono lo scambio termico convettivo con l'interno del cabinato e integrano aperture strategiche che fungono da sezioni di ingresso (*inlet*) e uscita (*outlet*) per il

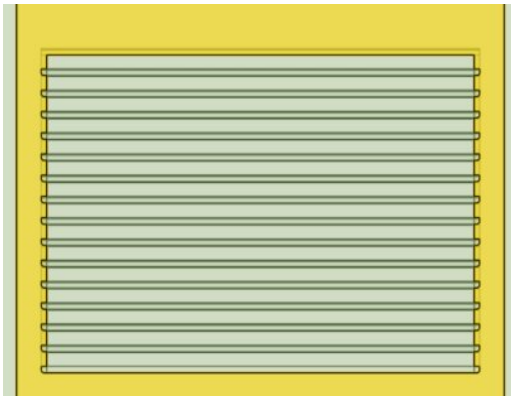
ricircolo naturale dell'aria. La tipologia di boundary e valori utilizzati sono riportati nella tabella sottostante.

Tabella 5.1: Condizioni al contorno di pressione (Inlet e Outlet)

Zone Name	Type	Gauge Pressure [Pa]	Temperature [K]
pressure-inlet-1	Pressure Inlet	0 (Total)	<i>current_temperature</i>
pressure-outlet-1	Pressure Outlet	0 (Static)	<i>current_temperature</i>
outlet-1	Pressure Outlet	0 (Static)	<i>current_temperature</i>
outlet-2	Pressure Outlet	0 (Static)	<i>current_temperature</i>

In questa formulazione, il termine *current\_temperature* si riferisce ai diversi valori di temperatura ambiente impiegati nella simulazione, estratti dai cento campioni generati tramite la tecnica *Latin Hypercube Sampling* (LHS).

A questa configurazione si aggiunge la porta frontale, dotata di feritoie di ventilazione che oltre ad assolvere alla loro naturale funzione, incrementano deliberatamente la complessità topologica della mesh.



(a) Rappresentazione frontale delle feritoie



(b) Rappresentazione laterale delle feritoie

Figura 5.2: Rappresentazione delle feritoie.

Per ottimizzare lo sforzo computazionale senza compromettere la generalità del modello, sono stati introdotti due piani di simmetria (*symmetry plane*), mentre la restante parte del volume è delimitata dal soffitto (*roof*), dal basamento (*ground*), dalla porta (*door*) e dalla parete laterale (*walls*). Per le diverse boundary condition, la tipologia e i valori di riferimento utilizzati sono i seguenti:

Tabella 5.2: Condizioni termiche per pareti con scambio convettivo

Zone Name	$h$ [W/m <sup>2</sup> K]	$T_\infty$ [K]	Spessore [m]	Emissività
Basamento	7.889	<i>curr_temp</i>	0.0050	0.9
Soffitto	4.673	<i>curr_temp</i>	0.1360	0.9
Pareti	5.217	<i>curr_temp</i>	0.1000	0.9
Porta	8.000	<i>curr_temp</i>	0.0015	0.9

Per garantire che la risposta termica del modello sia coerente con le specifiche prestazionali del componente reale, la caratterizzazione dei flussi di calore sulle superfici non è stata affidata esclusivamente a correlazioni teoriche standard, ma è stata definita sulla base di dati sperimentali e tecnici. In particolare, i valori dei coefficienti di scambio termico adottati per le pareti del cabinato sono stati estratti da database specialistici disponibili in rete.

Tali dati, che riflettono le proprietà convettive dei materiali, sono stati sintetizzati nella Tabella 5.3.

Tabella 5.3: Coefficienti di scambio termico derivati dai dati tecnici disponibili in rete.

Componente	Spessore [mm]	$\lambda$ [W/mK]	$R_{materiale}$ [m <sup>2</sup> K/W]	$\alpha$ esterna [W/m <sup>2</sup> K]	$R_{tot}$ [m <sup>2</sup> K/W]	$\alpha_{equivalente}$ [W/m <sup>2</sup> K]
Soffitto (calcestruzzo)	136	1.5	0.091	8	0.216	4.637
Basamento (calcestruzzo)	200	1.5	0.133	8	0.258	3.871
Pareti (calcestruzzo)	100	1.5	0.067	8	0.192	5.217
Porta (acciaio)	60.5	1.5	0.000	8	0.125	7.998

I valori sono stati calcolati con le seguenti formule:

Il calcolo della resistenza termica del materiale viene effettuato come:

$$R_{material} = \frac{s}{\lambda} \left[ \frac{\text{m}^2\text{K}}{\text{W}} \right] \quad (5.1)$$

La resistenza totale del componente, considerando il contributo convettivo esterno ( $\alpha_{ext}$ ), è data da:

$$R_{tot} = R_{material} + \frac{1}{\alpha_{ext}} \left[ \frac{\text{m}^2\text{K}}{\text{W}} \right] \quad (5.2)$$

Il coefficiente di scambio termico equivalente risulta quindi:

$$\alpha_{equivalent} = \frac{1}{R_{tot}} \left[ \frac{\text{W}}{\text{m}^2\text{K}} \right] \quad (5.3)$$

La sorgente termica principale del sistema, costituita dal trasformatore, è stata modellata definendo le condizioni al contorno riassunte nella Tabella 5.4:

Tabella 5.4: Condizioni termiche per componenti del Trasformatore

Zone Name	Heat Flux [W/m <sup>2</sup> ]	Internal Emissivity
wall-trasformatore	<i>potenza</i>	0.9

In linea con l'architettura di automazione del workflow, il valore della potenza dissipata non è configurato come un dato statico, bensì come una variabile dinamica la cui entità viene estratta sequenzialmente dal file JSON per ciascuno dei cento campioni del dataset.

Infine, il dominio originale è stato integrato con apposite estrusioni volumetriche nelle zone di inlet ed outlet, in corrispondenza delle zone di afflusso e deflusso, necessarie per garantire il corretto sviluppo del campo di velocità e prevenire instabilità numeriche derivanti da effetti di riflessione o ricircoli indesiderati in prossimità delle sezioni di ingresso ed uscita del fluido.

L'insieme di queste variabili crea un campo di moto turbolento caratterizzato da forti accoppiamenti termofluidodinamici, rappresentando un banco di prova ideale per verificare se la rete neurale sia in grado di sostituire i solutori tradizionali nella previsione della distribuzione di temperatura e dei flussi d'aria interni.

### 5.3 Modelli fisici e turbolenza

La simulazione integra lo scambio termico e gli effetti della turbolenza con le seguenti impostazioni:

- **Equazione dell'energia**
- **Modello di turbolenza:**  $k - \omega$  SST (Shear Stress Transport).
- **Modello di radiazione:** Surface-to-Surface (S2S).
  - *Surface Clusters:* 800 facce globali per cluster.
  - *Frequenza di calcolo:* Ogni 5 iterazioni.

### 5.4 Proprietà del fluido e condizioni operative

In questa sezione vengono definiti i parametri termofisici del fluido e le costanti ambientali necessarie per la risoluzione del campo di moto e del galleggiamento (*buoyancy*).

### 5.4.1 Modello di gas ideale

Il fluido (aria) è trattato come un gas ideale, dove la densità  $\rho$  è funzione della pressione assoluta e della temperatura locale secondo l'equazione di stato:

$$\rho = \frac{p_{abs}}{R_{air} \cdot T} \quad (5.4)$$

dove  $R_{air} \approx 287.05 \text{ J}/(\text{kg} \cdot \text{K})$  è la costante specifica del gas.

### 5.4.2 Accelerazione di gravità

Per l'attivazione dei termini di sorgente gravitazionale nelle equazioni della quantità di moto, viene definito il vettore accelerazione  $\vec{g}$  lungo l'asse verticale:

$$\vec{g} = \begin{bmatrix} 0 & 0 & -9.81 \end{bmatrix}, \text{ m/s}^2 \quad (5.5)$$

### 5.4.3 Definizione della densità operativa

Al fine di modellare correttamente le spinte convettive ed evitare instabilità numeriche nel calcolo del gradiente di pressione idrostatica, è definita una densità operativa di riferimento  $\rho_{op}$  basata sulle condizioni iniziali del dominio:

$$\rho_{op} = \frac{p_{op}}{R_{air} \cdot T_{\infty}} \quad (5.6)$$

#### Parametri impostati:

- *Operating Pressure* ( $p_{op}$ ): Pressione di riferimento del sistema.
- *Reference Temperature* ( $T_{\infty}$ ): Temperatura ambiente corrente.
- *Operating Density*: Abilitata e impostata al valore calcolato  $\rho_{op}$ .

## 5.5 Metodi di soluzione e discretizzazione spaziale

Per la risoluzione numerica delle equazioni di governo, è stato adottato un approccio basato sulla pressione con accoppiamento delle variabili di moto. Di seguito sono riportati i dettagli della strategia di calcolo e degli schemi di discretizzazione spaziale implementati.

### 5.5.1 Strategia di risoluzione numerica

La risoluzione del sistema di equazioni differenziali alle derivate parziali è stata affidata al solutore **Coupled**. A differenza degli algoritmi di tipo segregato, questo

metodo risolve simultaneamente le equazioni di continuità e della quantità di moto, garantendo una convergenza più robusta e una maggiore stabilità numerica.

Per quanto concerne la discretizzazione dei gradienti, è stato adottato lo schema **Least Squares Cell Based**, il quale offre un ottimo compromesso tra accuratezza numerica e costo computazionale, minimizzando gli errori di diffusione sui volumi di controllo. Infine, per favorire la stabilità del calcolo durante le iterazioni, è stato impiegato il metodo **Pseudo Time Method** con una formulazione **Global Time Step**. Questa configurazione permette al solutore di avanzare verso la soluzione stazionaria in modo controllato, adattando il passo temporale fittizio alla scala globale del dominio e prevenendo divergenze precoci nelle fasi iniziali della simulazione.

### 5.5.2 Schemi di discretizzazione spaziale

Per garantire un'accuratezza del secondo ordine e gestire correttamente i gradienti di pressione dovuti alla convezione naturale, è stata definita la seguente configurazione:

Tabella 5.5: Schemi di discretizzazione spaziale adottati nella simulazione.

<b>Variabile / Equazione</b>	<b>Schema di Discretizzazione</b>
Pressione	PRESTO!
Densità	<i>Second Order Upwind</i>
Quantità di Moto	<i>Second Order Upwind</i>
Energia Cinetica Turbolenta ( $k$ )	<i>Second Order Upwind</i>
Dissipazione Specifica ( $\omega$ )	<i>Second Order Upwind</i>
Energia	<i>Second Order Upwind</i>

## 5.6 Verifica della convergenza e validazione dei risultati

La verifica della convergenza rappresenta un passaggio metodologico di fondamentale importanza per garantire che i risultati derivanti dalla simulazione numerica siano validi e fisicamente consistenti. In questo studio, il raggiungimento della convergenza non è stato affidato esclusivamente al monitoraggio dei residui delle equazioni di governo, ma è stato confermato attraverso un'analisi multicriterio.

In prima istanza, si è verificato che i residui normalizzati relativi alle equazioni di continuità, quantità di moto, energia e turbolenza decrescessero fino a raggiungere valori nell'ordine di  $10^{-3}$  o inferiori Figura 5.3, assicurando così una stabilità numerica globale del sistema. Tuttavia, per superare i limiti intrinseci dei soli residui, la

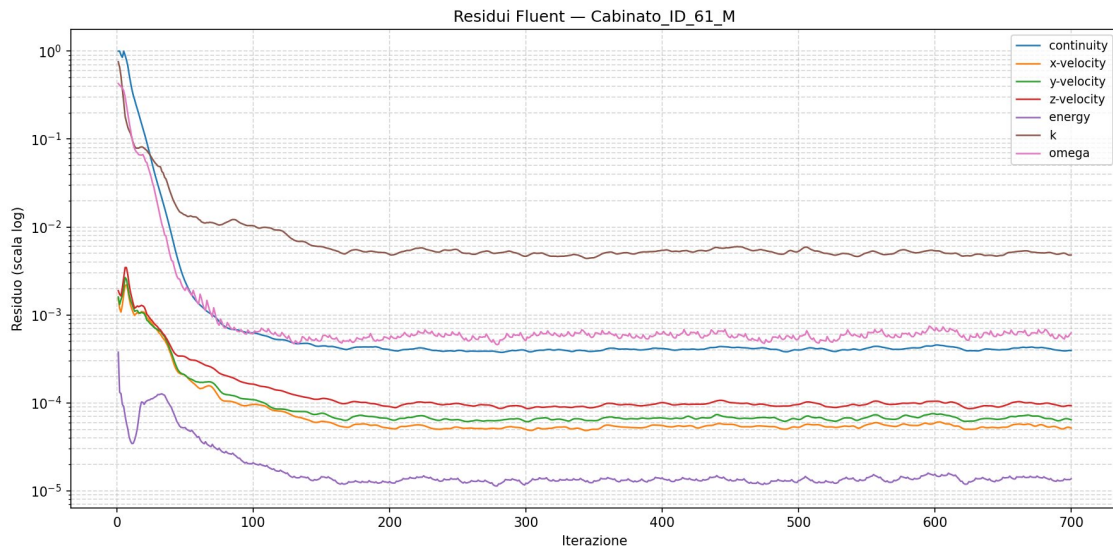


Figura 5.3: Residui scalati

validazione è stata estesa all'analisi di variabili scalari d'interesse. Nello specifico, sono stati monitorati i grafici relativi alla temperatura media e alla temperatura massima su diverse superfici chiave del dominio computazionale, integrando tale verifica con il controllo dei valori medi di temperatura rilevati in dieci punti di monitoraggio (*probes*), distribuiti strategicamente all'interno del campo fluido.

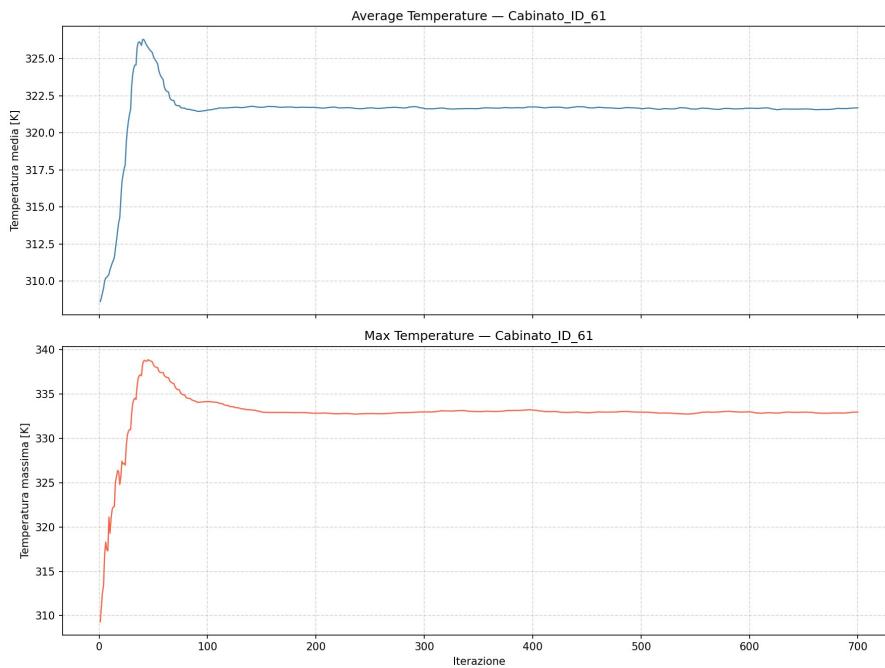


Figura 5.4: Temperatura media e massima sulla superficie del trasformatore

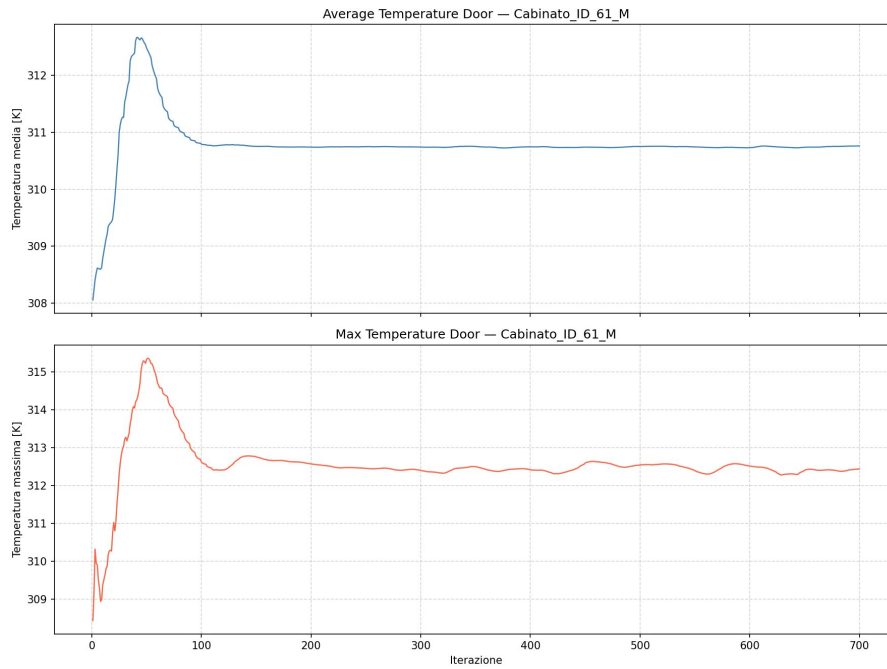


Figura 5.5: Temperatura media e massima sulla porta del cabinato

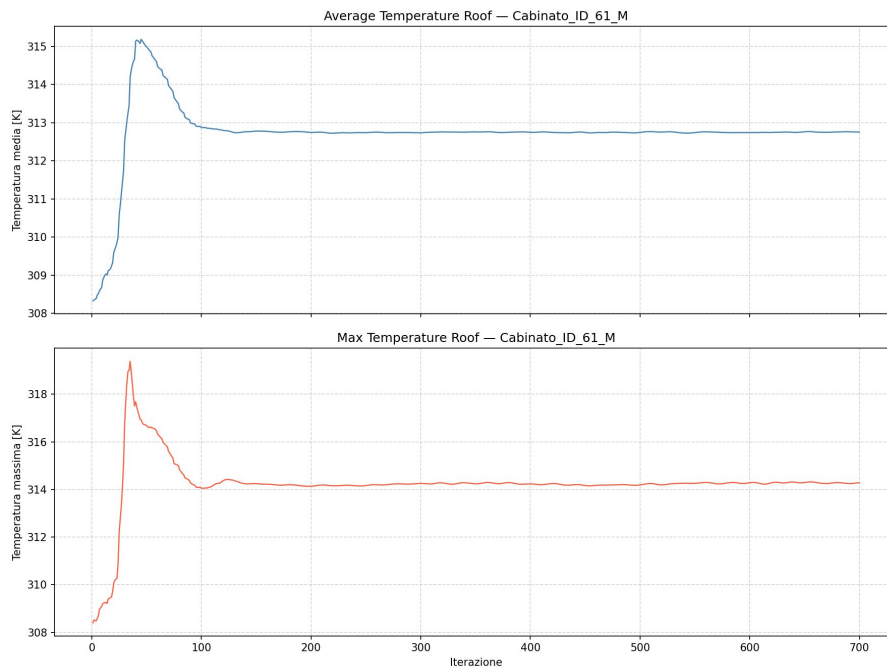


Figura 5.6: Temperatura media e massima sul soffitto del cabinato

La soluzione è stata considerata convergente solo quando tali parametri hanno mostrato un andamento asintotico, come mostrato nelle figure 5.4,5.5,5.6 , stabilizzandosi su valori costanti al progredire delle iterazioni. Contestualmente, è stata verificata la correttezza della risoluzione dello strato limite termico in prossimità delle pareti

attive, accertandosi che il valore del  $y^+$  si attestasse in prossimità dell'unità ( $y^+ \approx 1$ )  
Figura 5.7, condizione necessaria per la validità del modello di turbolenza adottato.

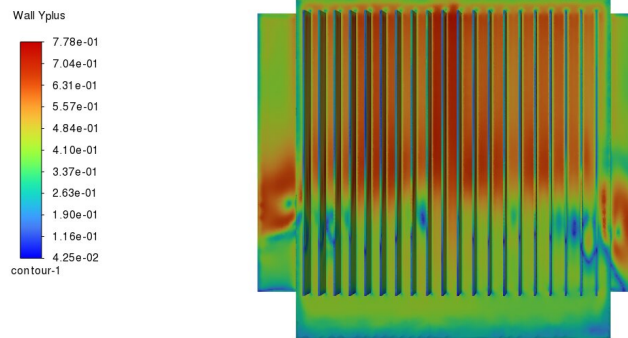


Figura 5.7: Residui scalati

Infine, un ulteriore e decisivo criterio di controllo ha riguardato la verifica della conservazione dell'energia all'interno dell'intero dominio. Si è proceduto analizzando il bilancio dei flussi termici, accertandosi che il valore del flusso totale di potenza netta (differenza tra potenza immessa ed uscente) fosse il più vicino possibile allo zero. La validità fisica della simulazione è stata dunque confermata dalla discrepanza estremamente ridotta tra questi flussi rispetto alla potenza totale scambiata, garantendo la coerenza del modello rispetto ai principi della termodinamica.

## 6. GENERAZIONE DEL DATASET

---

### 6.1 Definizione dello spazio dei parametri e strategia di campionamento

Al fine di costituire un dataset rappresentativo composto da 100 campioni (*samples*) per l'addestramento della rete neurale, è stata definita una strategia di variazione controllata basata su cinque grandezze indipendenti. Tali variabili, selezionate per riflettere sia la flessibilità geometrica che la variabilità operativa del sistema, si suddividono in tre parametri dimensionali relativi al cabinato (lunghezza, larghezza e altezza) e due parametri legati alla fisica del problema (temperatura ambiente e potenza dissipata). Mentre gli intervalli delle dimensioni geometriche sono stati stabiliti sulla base di vincoli di ingombro spaziale tipici, la lunghezza varia tra 1800.00 mm e 2700.00 mm, la larghezza tra 1200.00 mm e 2200.00 mm e l'altezza tra 2400.00 mm e 2900.00 mm — la definizione dei range per le grandezze fisiche ha seguito rigorosi criteri ambientali e normativi. L'intervallo della temperatura ambiente (10.0°C - 40.0°C) è stato infatti determinato sulla base dei rilievi meteorologici minimi e massimi registrati nell'anno precedente al presente studio presso il sito di installazione previsto. Per quanto concerne la potenza dissipata dal trasformatore, il range operativo è stato fissato tra 1000 W e 12000 W, in stretta conformità con i requisiti tecnici imposti dal *Regolamento (UE) 2019/1783 della Commissione del 1° ottobre 2019*, il quale integra e modifica la direttiva 2009/125/CE in merito alle specifiche di progettazione ecocompatibile dei trasformatori di piccola, media e grande potenza. I limiti inferiori (*Lower Bound*), superiori (*Upper Bound*) e i valori base utilizzati per la generazione del dataset sono riassunti nella Tabella 6.1.

Tabella 6.1: Range di variazione dei parametri per la generazione dei 100 campioni.

Parametro	Unità	Lower Bound	Upper Bound	Valore Base
Lunghezza	[mm]	1800.00	2700.00	2095.00
Larghezza	[mm]	1200.00	2200.00	1541.00
Altezza	[mm]	2400.00	2900.00	2584.00
Temperatura	[°C]	10.0	40.0	27.0
Potenza Dissipata	[W]	1000	12000	1600

### 6.1.1 Metodologia di campionamento: Latin Hypercube Sampling (LHS)

Per garantire che i 100 campioni generati rappresentino in modo uniforme l'intero dominio multidimensionale definito dai parametri geometrici e fisici, si è adottata la tecnica del *Latin Hypercube Sampling* (LHS). A differenza di un semplice campionamento casuale (Monte Carlo), il metodo LHS è una tecnica di tipo *space-filling* progettata per distribuire i punti in modo che ogni variabile sia campionata in ogni sua zona del range operativo.

Il principio di funzionamento del LHS prevede la suddivisione dell'intervallo di variazione di ciascuna delle  $k$  dimensioni (nel nostro caso  $k = 5$ ) in  $n$  intervalli di uguale probabilità, dove  $n$  corrisponde al numero di campioni desiderati ( $n = 100$ ). Il campionamento viene effettuato in modo tale che, per ogni riga e ogni colonna dell'ipercubo multidimensionale, sia presente un solo punto di campionamento.

L'adozione di questa strategia è fondamentale nella creazione di un dataset destinato all'addestramento di reti neurali. Essa permette infatti di massimizzare l'informazione ottenuta da ogni singola simulazione CFD, riducendo drasticamente il numero di test necessari per mappare accuratamente la risposta termica del cabinato al variare delle dimensioni e della potenza dissipata.

L'applicazione pratica della metodologia di campionamento è stata affidata a uno script sviluppato in linguaggio Python, progettato per automatizzare la generazione dei punti nello spazio dei parametri e garantire la riproducibilità del dataset. Il flusso di lavoro prevede l'utilizzo di file in formato JSON sia per la fase di input che per quella di output, facilitando l'interoperabilità tra lo script di campionamento e il solutore CFD.

Nello specifico, la procedura riceve in ingresso un file di configurazione contenente i domini di variazione delle grandezze geometriche e fisiche. Sulla base di questi limiti, lo script esegue l'algoritmo *Latin Hypercube Sampling*, restituendo un file JSON che elenca i 100 campioni generati. Ogni elemento del dataset così ottenuto è identificato da un codice univoco (*ID*) e contiene i valori precisi delle cinque variabili indipendenti estratti all'interno dei rispettivi intervalli.

A titolo esemplificativo, viene riportata di seguito la struttura tipica di un record generato dallo script:

```
{
  "id": 3,
  "Lunghezza": 2090.271351890581,
  "Larghezza": 1842.7844340762965,
  "Altezza": 2700.7448232565694,
```

```
"Temperatura": 11.94016771147285,  
"Potenza_dissipata": 10333.417118616457  
}
```

## 6.2 Automazione della modellazione geometrica e della *Named Selection*

La generazione delle cento varianti geometriche previste per la costituzione del dataset è stata interamente automatizzata all'interno dell'ambiente **Ansys SpaceClaim 2024**, sfruttando le potenzialità dell'interfaccia di scripting integrata in linguaggio **Python**. Il workflow di automazione ha inizio con l'esecuzione dello script, programmato per operare in modalità iterativa su un modello di riferimento denominato *STARTER*. Lo script legge sequenzialmente il file **JSON** e traduce i valori numerici di lunghezza, larghezza e altezza in modifiche topologiche dirette sulla geometria base. Un elemento cardine di questa procedura risiede nella gestione sistematica delle **Named Selections**: parallelamente alla modifica delle quote, il codice **Python** provvede ad associare etichette identificative univoche alle diverse superfici e ai volumi del dominio. Tale operazione risulta determinante per le successive fasi di *meshing* e di setup del solutore, in quanto l'assegnazione automatica dei nomi alle *boundary* permette al software di riconoscere e applicare — autonomamente o tramite script dedicati — le corrette condizioni al contorno, eliminando la necessità di interventi manuali e garantendo l'integrità del processo di calcolo per l'intero dataset.

## 6.3 Automazione del *meshing* e delle simulazioni

L'intero flusso di lavoro, dalla generazione della mesh alla risoluzione numerica, è stato automatizzato mediante l'integrazione di **Ansys Fluent 2024** con l'estensione **PyFluent**. Questa architettura ha permesso di gestire il calcolo in modo massivo attraverso l'impiego di un cluster ad alte prestazioni (*HPC - High Performance Computing*), garantendo l'esecuzione sequenziale dei 100 campioni senza necessità di supervisione manuale.

Per la fase di pre-processamento, è stata adottata la metodologia *Watertight Geometry*, un workflow strutturato che assicura la chiusura volumetrica del dominio e la qualità degli elementi. Al fine di catturare correttamente i fenomeni fluidodinamici locali, sono stati applicati dei *local sizing* in regioni critiche, con particolare attenzione alle feritoie di ventilazione delle porte e alle superfici del trasformatore. In questa fase, lo script interviene inoltre per correggere alcune assegnazioni automatiche delle *boundary conditions* effettuate dal software, le quali risultavano incoerenti con la fisica del problema, garantendo così la corretta imposizione dei vincoli al contorno.

Una scelta progettuale specifica ha riguardato la modellazione dello strato limite. Sul trasformatore, principale sorgente termica del sistema, è stato inserito un *boundary layer* mirato a garantire un valore di  $y^+ \approx 1$ , condizione necessaria per una risoluzione accurata del gradiente termico e del sottostrato laminare. Al contrario, per le restanti pareti del cabinato, si è scelto di non implementare lo strato limite. Tale decisione rappresenta un consapevole compromesso ingegneristico: l'aggiunta di *inflation layers* su tutte le superfici avrebbe incrementato eccessivamente il numero di elementi della mesh, rendendo il peso del dataset difficilmente gestibile per la successiva fase di addestramento della rete neurale.

Sebbene questa semplificazione comporti una relativa perdita di accuratezza globale, essa risulta coerente con gli obiettivi della presente tesi. Lo scopo principale del lavoro, infatti, non risiede nell'ottenimento di una simulazione CFD di assoluta precisione numerica, bensì nella dimostrazione dell'efficacia e dei vantaggi derivanti dall'applicazione di modelli basati sull'intelligenza artificiale alle simulazioni fluidodinamiche. Una volta completata la generazione della mesh, lo script procede automaticamente al setup fisico e alla risoluzione numerica, applicando i modelli e gli schemi di discretizzazione precedentemente descritti nel capitolo relativo alla configurazione della simulazione.

## 7. ANALISI DEI RISULTATI DELLA PREDIZIONE

---

### 7.1 Metriche di valutazione delle performance: NRMSE

Per l'analisi dei risultati predittivi della rete neurale è stato adottato il *Normalized Root Mean Square Error* (NRMSE) il quale rappresenta la radice dell'errore quadratico medio normalizzato. La scelta di questa metrica deriva dalla necessità di superare i limiti intrinseci del semplice RMSE, il cui valore, essendo espresso nella medesima unità di misura dei dati originali, risulta strettamente dipendente dalla scala numerica del fenomeno osservato. Mentre l'RMSE classico non permette di stabilire in modo assoluto se l'entità di un errore sia trascurabile o significativa senza una profonda conoscenza del contesto fisico, l'NRMSE consente di svincolare la valutazione della performance del modello dalla sua unità di misura. Nello specifico, è stata applicata la normalizzazione per intervallo, dividendo l'RMSE per la differenza tra il valore massimo e il valore minimo dei dati osservati secondo la formulazione:

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \quad (7.1)$$

Tale approccio risulta fondamentale per garantire l'imparzialità del confronto tra modelli o dataset operanti su ordini di grandezza differenti; in assenza di normalizzazione, infatti, un modello applicato a un dataset con valori elevati, presenterebbe quasi certamente un errore quadratico medio molto più alto rispetto a un modello operante su scale ridotte, pur possedendo potenzialmente una capacità predittiva analoga o superiore. L'NRMSE fornisce dunque un indicatore adimensionale che rende confrontabili le prestazioni del sistema indipendentemente dalla scala numerica dei dati termofluidodinamici trattati.

### 7.2 Analisi dei risultati e prestazioni della rete neurale

In questa fase vengono presentati i risultati derivanti da due distinti processi di addestramento della rete neurale: il primo condotto sull'intero dominio fluido mediante file in formato **VTU**, e il secondo focalizzato esclusivamente sulle variabili superficiali tramite file **VTP**. Poiché l'architettura della rete richiede specificamente questi formati per la gestione dei dati basata su grafi, è stato sviluppato uno script Python dedicato che, sfruttando la libreria `pyvista`, provvede alla conversione automatizzata dei file nativi di Fluent (`.cas` e `.dat`).

L'analisi dei risultati relativi al caso *VTP*, sintetizzati nella Tabella 7.1 e rappresentati nella figura 7.1, evidenzia le prestazioni del modello attraverso tre raggruppamenti

distinti di dati: *Training*, *Validation* e *Test*. Nel set di **Training**, composto da 32 grafi per un totale di oltre 11 milioni di nodi, si osservano i valori di errore più contenuti. Tale riduzione è fisiologica, poiché i parametri della rete vengono ottimizzati iterativamente proprio per minimizzare la *loss function* su questo specifico insieme di dati.

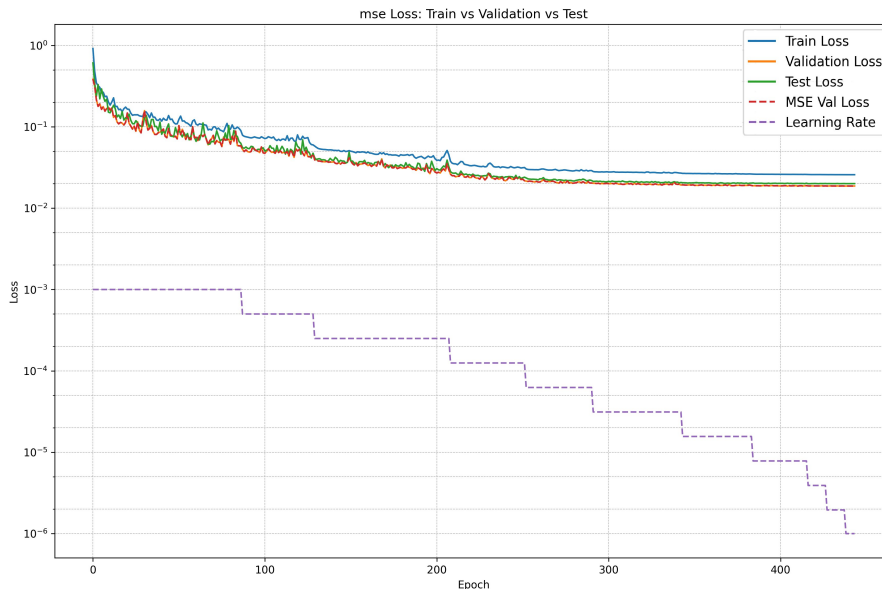


Figura 7.1: Rappresentazione grafica della Loss Function all'avanzare delle epoche per le predizioni superficiali.

Il set di **Validation** interviene invece durante la fase di addestramento come strumento di controllo esterno: mettendo alla prova il modello su dati non utilizzati per l'aggiornamento dei pesi, permette di verificare se la rete stia effettivamente apprendendo le regole fisiche e matematiche generali o se stia semplicemente memorizzando i dati di addestramento (*overfitting*). La coerenza riscontrata tra gli errori di *Training* e *Validation* conferma l'ottima capacità di generalizzazione del modello. Infine, le prestazioni definitive sono valutate sul set di **Test**, un raggruppamento di dati analizzato una sola volta al termine del processo. Con un errore NRMSE sulla temperatura pari al 3.45%, il modello dimostra una notevole precisione predittiva, confermando la validità dell'approccio basato su reti neurali.

L'analisi dell'andamento delle curve di *loss* riportate nella Figura 7.2 permette di identificare diverse fasi cruciali nell'evoluzione del modello. Nella fase iniziale del processo, si osserva un gradiente di apprendimento estremamente elevato: la rete neurale identifica rapidamente i macro-pattern geometrici e le leggi fisiche fondamentali che governano il sistema, traducendo tale comprensione in una drastica riduzione dell'errore globale.

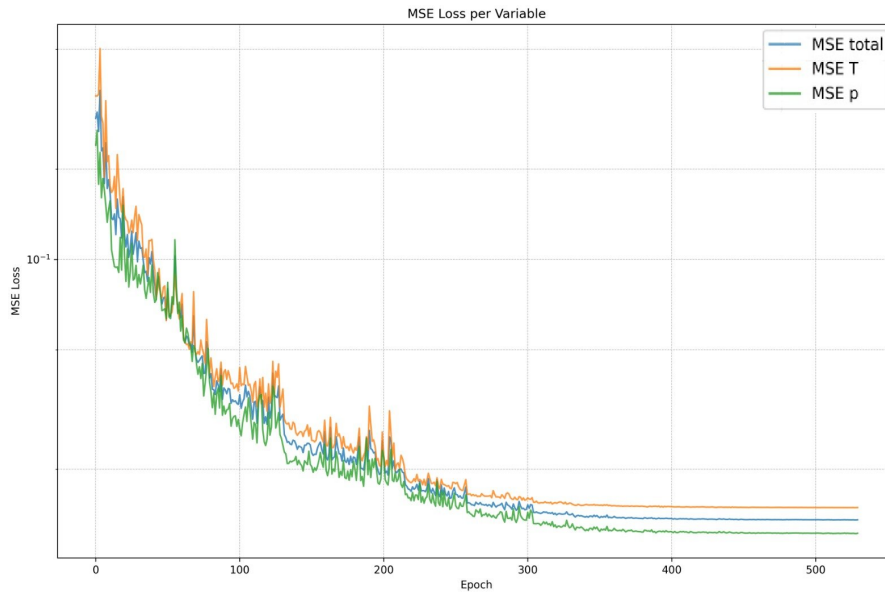


Figura 7.2: Rappresentazione grafica dell'apprendimento delle diverse variabili nella predizione superficiale.

In una fase intermedia, si evidenziano continue risalite della curva (tra l'epoca 100 e 200), queste sono dovute al metodologia di addestramento, dove la rete non analizza contemporaneamente tutti i *sample*, ma a pacchetti e l'ingresso di un nuovo campione che si discosta sensibilmente dai precedenti, provoca una nuova taratura dei pesi del modello. Contestualmente si assiste a una divergenza tra i campi fisici trattati: la curva relativa alla pressione inizia a distaccarsi in modo sistematico, stabilizzandosi su valori stabilmente inferiori rispetto a quelli della temperatura, indicando una maggiore accuratezza nella predizione del campo scalare relativo alla pressione.

Tabella 7.1: Metriche di errore NRMSE per i set di Training, Validazione e Test (Caso VTP).

Metrica	Training Set	Validation Set	Test Set
NRMSE Temperatura ( $T$ )	0.0306	0.0303	0.0345
NRMSE Pressione ( $p$ )	0.0143	0.0133	0.0149
RMSE Combined (Range)	0.0239	0.0234	0.0266
Numero di Grafi	32	4	5
Nodi Totali	11.498.852	1.431.773	1.799.399

Il secondo processo di addestramento ha riguardato l'intero dominio fluido attraverso l'utilizzo di file in formato **VTU**. In questo scenario, la rete neurale è chiamata a

predire un numero superiore di campi fisici rispetto al caso superficiale, includendo non solo pressione e temperatura, ma anche le tre componenti del vettore velocità ( $U_x, U_y, U_z$ ) e il relativo modulo ( $|U|$ ).

Come evidenziato dalle metriche riportate nella Tabella 7.2, si riscontra un sensibile peggioramento delle prestazioni predittive nel dominio volumetrico rispetto a quello superficiale. Tale tendenza trova una chiara corrispondenza visiva nell'andamento delle curve di *loss* riportate in Figura 7.3.

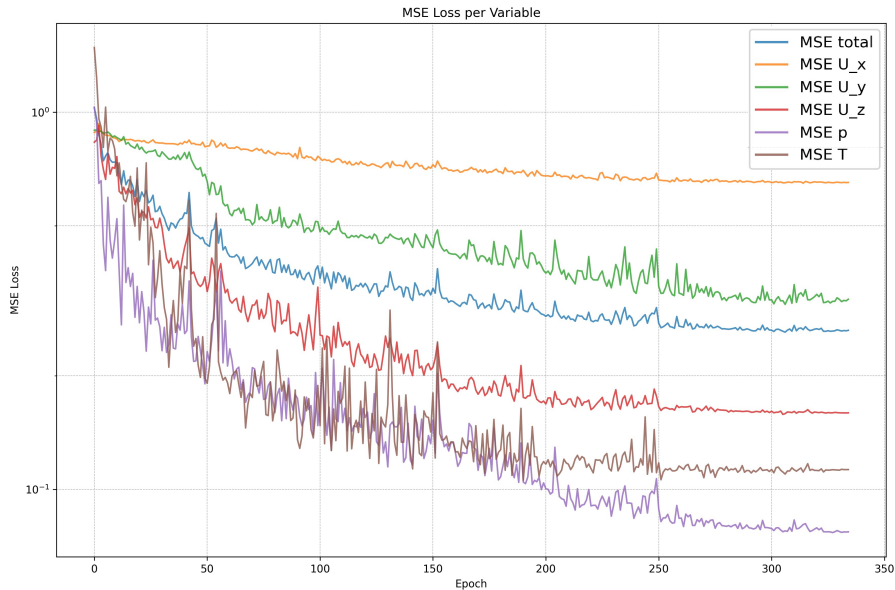


Figura 7.3: Rappresentazione grafica della Loss Function all'avanzare delle epoche.

Le curve relative alla pressione ( $p$ ) e alla temperatura ( $T$ ) mostrano una decrescita estremamente rapida, raggiungendo i valori di errore più bassi del dataset. Risulta evidente come la rete trovi relativamente agevole l'apprendimento della distribuzione termica e pressoria, grandezze che, in regime stazionario, presentano una distribuzione spaziale più regolare e legata alla macro-geometria del sistema. Al contrario, le componenti della velocità ( $U_x, U_y, U_z$ ) mostrano una resistenza molto maggiore alla minimizzazione dell'errore. In particolare, la componente  $U_x$  (evidenziata in arancione nel grafico) presenta il valore residuo più elevato.

La discrepanza riscontrata non è imputabile esclusivamente alla dimensione del dataset — che pur conta oltre 72 milioni di nodi nel set di *training* — ma trova una precisa giustificazione nella natura fisica del problema. In presenza di regimi turbolenti e caotici, caratterizzati da alte frequenze spaziali e strutture vorticoso instabili, la rete incontra un limite intrinseco di risoluzione, difatti il fenomeno analizzato è caratterizzato da un regime di **convezione naturale** in cui le dimensioni

geometriche del cabinato e i gradienti termici imposti determinano numeri di **Reynolds** ( $Re$ ) e di **Rayleigh** ( $Ra$ ) estremamente elevati.

Questi parametri indicano la transizione verso un flusso pienamente turbolento, intrinsecamente non stazionario. Sebbene la simulazione CFD sia stata condotta con un solutore stazionario (*steady-state*), il campo di moto reale presenta una natura caotica e instabile; la semplificazione numerica della simulazione cattura dunque un valore mediato di un fenomeno fluttuante. Per la rete neurale, l'apprendimento e la generalizzazione di tali eterogeneità spaziali sul volume 3D risultano significativamente più complessi rispetto alla mappatura dei soli dati superficiali, portando a un incremento dell'errore NRMSE, che sul modulo della velocità ( $|U|$ ) raggiunge circa il 6.67% nel set di test.

Tabella 7.2: Metriche di errore NRMSE per i set di Training, Validazione e Test (Caso VTU).

<b>Metrica</b>	<b>Training Set</b>	<b>Validation Set</b>	<b>Test Set</b>
NRMSE $U_x$	0.0325	0.0328	0.0339
NRMSE $U_y$	0.0211	0.0211	0.0215
NRMSE $U_z$	0.0370	0.0417	0.0405
NRMSE Pressione ( $p$ )	0.0234	0.0293	0.0245
NRMSE Temperatura ( $T$ )	0.0319	0.0509	0.0406
NRMSE Modulo Velocità ( $ U $ )	0.0640	0.0664	0.0667
RMSE Combined (Range)	0.0298	0.0366	0.0332
Nodi Totali	72.653.045	8.992.308	9.227.682

### 7.3 Analisi dei Campi Volumetrici (VTU)

#### 7.3.1 Validazione qualitativa: confronto visivo dei campi di pressione

A completamento dell'analisi quantitativa, viene presentata una validazione qualitativa attraverso il confronto visivo tra i risultati delle simulazioni CFD tradizionali e le predizioni generate dalla rete neurale. Nelle immagini riportate nelle Figure 7.4 7.5 7.6, vengono messe a confronto tre diverse sezioni del dominio relative al campo di pressione  $p$  [Pa]: per ogni coppia, l'immagine a sinistra rappresenta il target (*Ground Truth* da CFD), mentre quella a destra mostra l'output della rete neurale. Come suggerito dall'analisi numerica della *loss function*, il campo di

pressione non ha rappresentato una criticità per il processo di apprendimento. Il modello è in grado di riprodurre con estrema fedeltà la distribuzione spaziale delle pressioni, identificando correttamente sia i gradienti principali che le zone di ristagno. Sebbene si possa riscontrare una lieve semplificazione nei dettagli più minuti in prossimità dell'alettatura del trasformatore, il comportamento globale è catturato con una precisione tale da rendere le coppie di immagini quasi indistinguibili. Questo eccellente accordo visivo conferma l'affidabilità del valore di NRMSE ottenuto, dimostrando che la rete ha appreso non solo la complessità geometrica del cabinato ma anche la fisica sottostante che regola il trasporto del fluido.

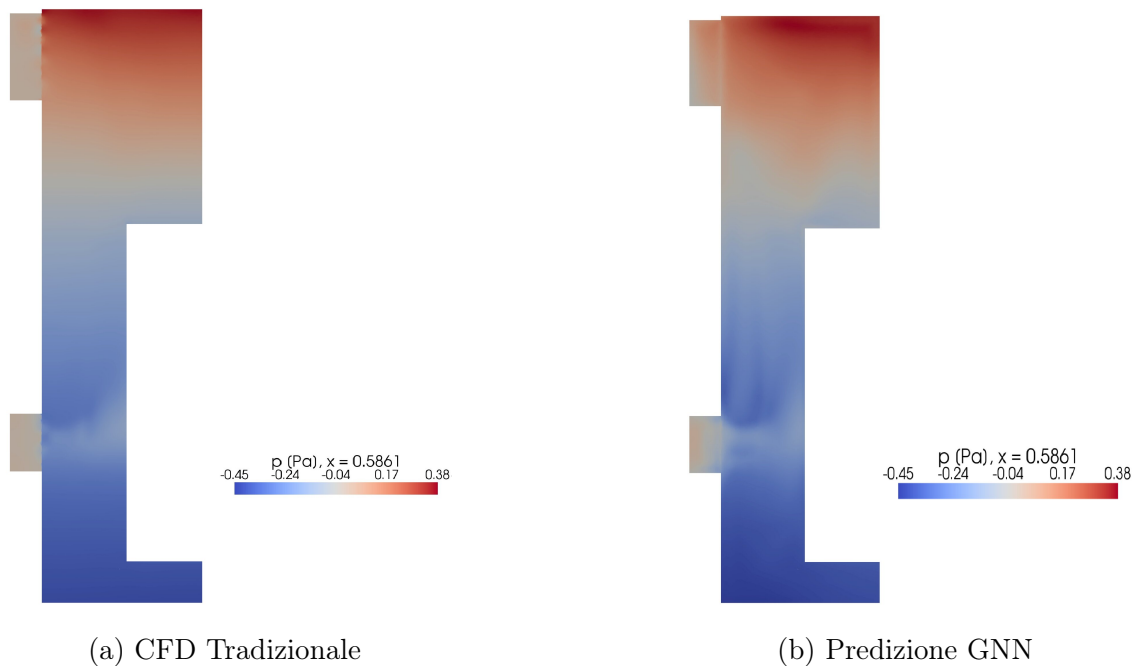


Figura 7.4: Confronto del campo di pressione  $p$  [Pa] sulla sezione trasversale  $x$ .

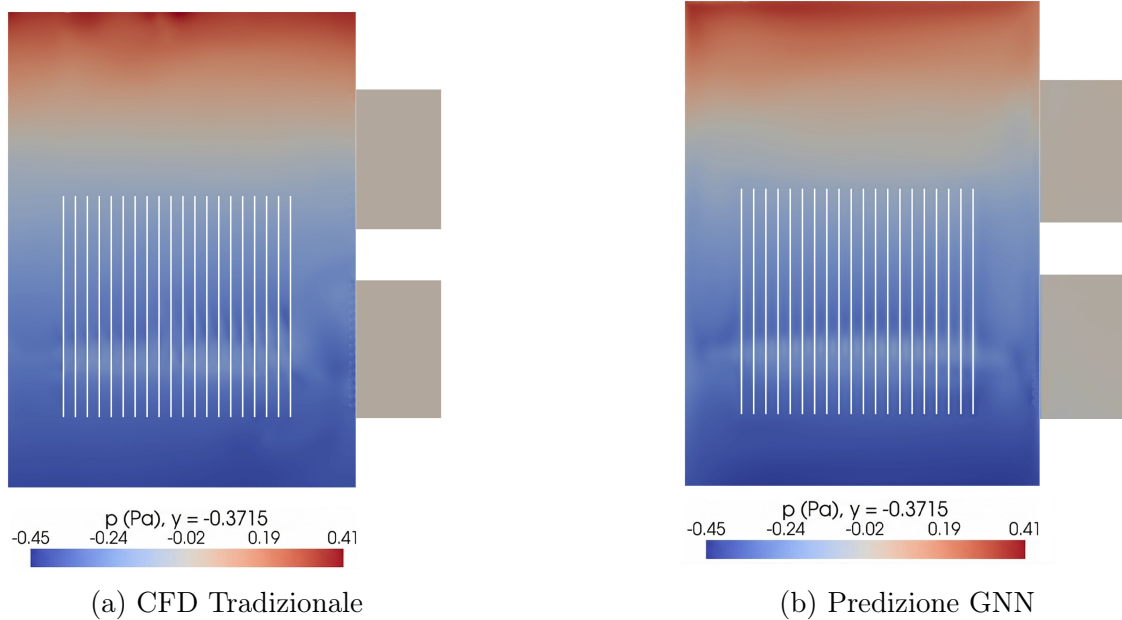


Figura 7.5: Confronto del campo di pressione  $p$  [Pa] sulla sezione longitudinale  $y$ .

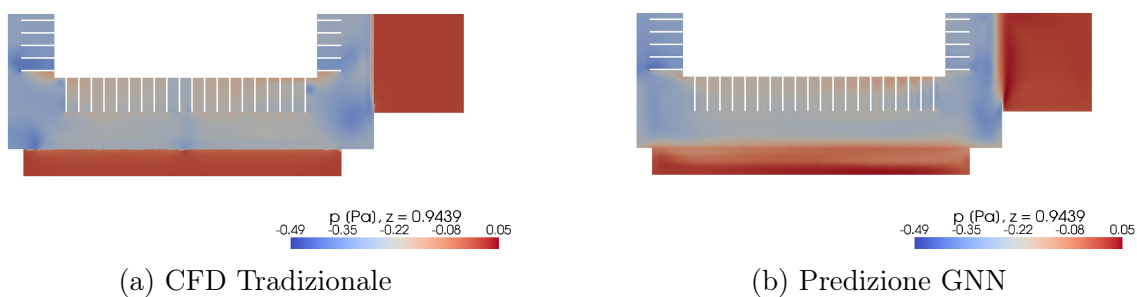


Figura 7.6: Confronto del campo di pressione  $p$  [Pa] sulla sezione orizzontale  $z$ .

### 7.3.2 Validazione qualitativa: confronto visivo dei campi di temperatura

Analogamente a quanto osservato per i campi di pressione, l'analisi termica conferma le medesime considerazioni metodologiche. La rete neurale esibisce una notevole capacità predittiva nel catturare la distribuzione termica macroscopica e l'andamento globale del campo. Tuttavia, emergono alcune limitazioni nella risoluzione accurata dei dettagli locali e nell'individuazione dei gradienti termici più severi, questo è dovuto soprattutto ai dati in input alla rete, infatti, come si può notare dalle immagini, lo scostamento della predizione avviene principalmente nella zona ad elevata turbolenza, dove di fatto le informazioni del dataset sono più eterogenee e di conseguenza la rete non è in grado di apprendere con precisione la fisica sottostante.

Come si può osservare dal confronto delle immagini nelle sezioni longitudinali e orizzontali (Fig. 7.8 e 7.9), la GNN individua correttamente le zone di ristagno e i percorsi dei flussi caldi, pur perdendo parte della nitidezza nei punti di massima variazione termica in prossimità delle pareti attive.

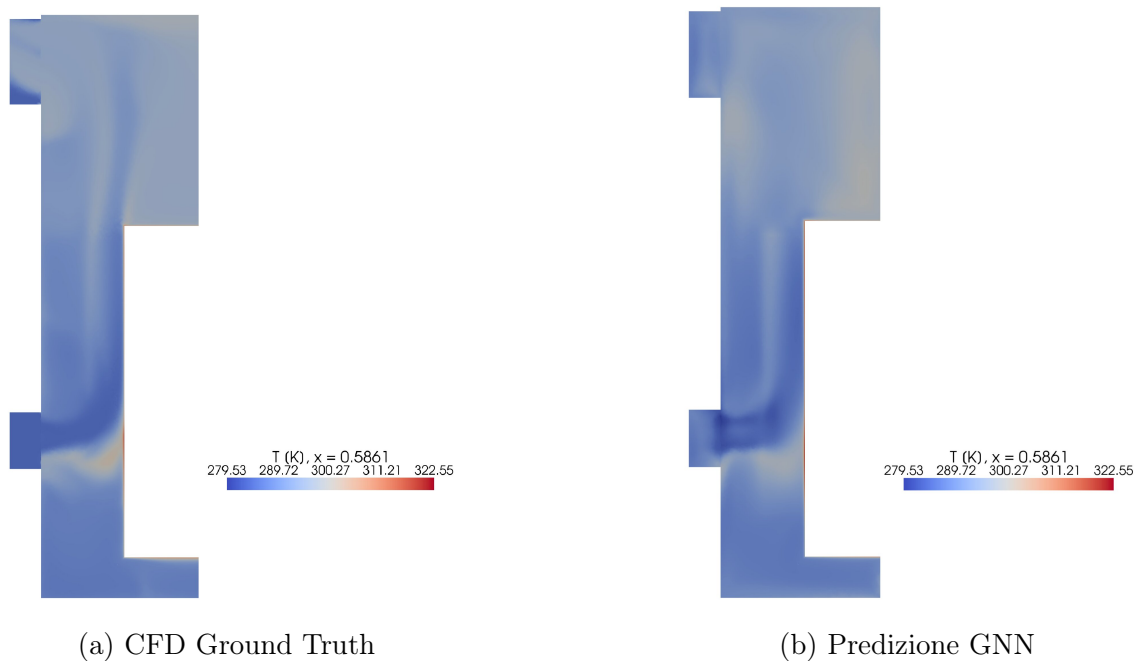


Figura 7.7: Confronto qualitativo del campo di temperatura  $T$  [°C] sulla sezione trasversale  $x$ .

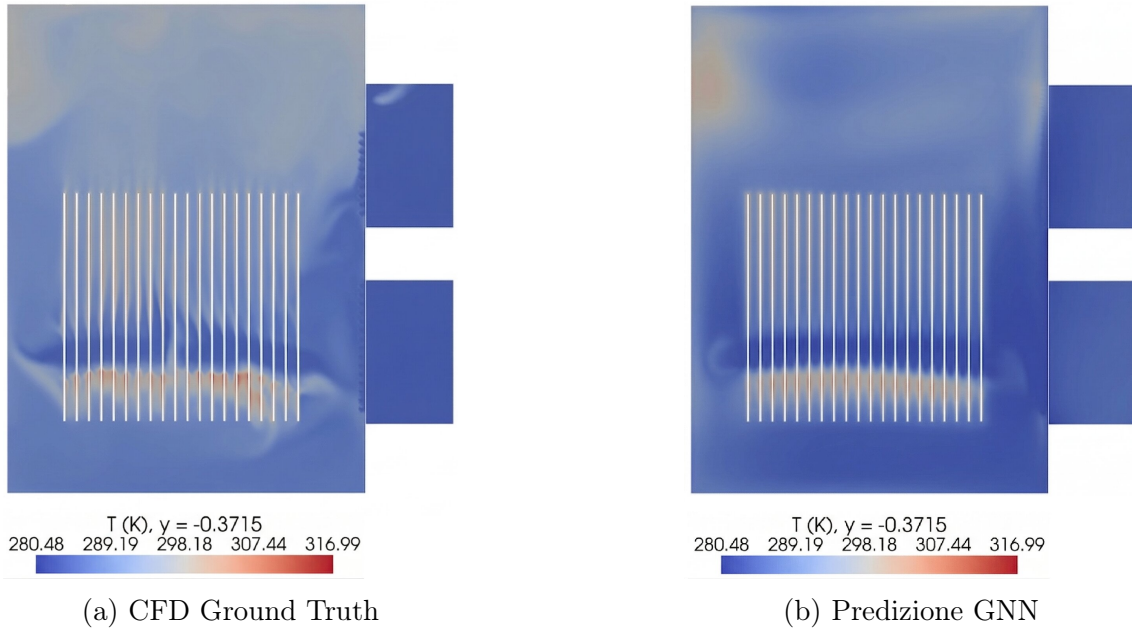


Figura 7.8: Confronto qualitativo del campo di temperatura  $T$  [°C] sulla sezione longitudinale  $y$ .

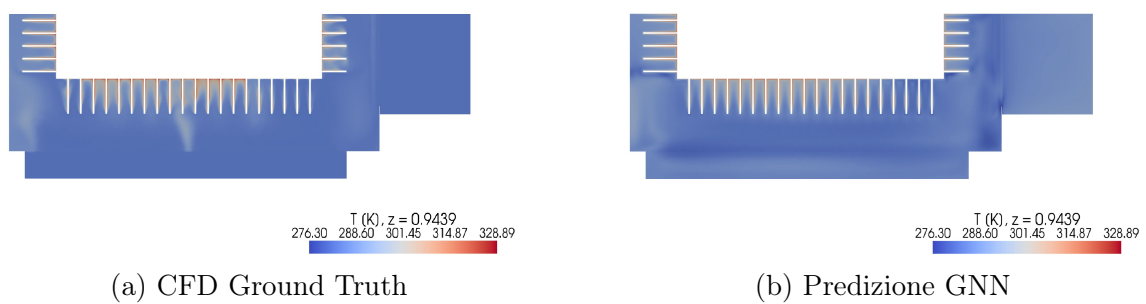


Figura 7.9: Confronto qualitativo del campo di temperatura  $T$  [°C] sulla sezione orizzontale  $z$ .

### 7.3.3 Validazione qualitativa: confronto visivo dei campi di velocità

La validazione qualitativa dei campi di velocità, riportata in Figura 7.10,7.11,7.12, evidenzia in modo inequivocabile il comportamento della rete neurale nei confronti della variabilità del campo di velocità. Dal confronto tra le simulazioni CFD (sinistra) e le predizioni GNN (destra), emerge come il modello sia in grado di identificare con precisione i macro-flussi e i percorsi preferenziali dell'aria all'interno del cabinato, individuando correttamente le zone di accelerazione tra le alette e l'effetto camino indotto dalla spinta termica. Tuttavia, è evidente un marcato effetto di *smoothing* (smussamento) nelle predizioni della rete, soprattutto nelle zone dove l'intensità della turbolenza è elevata. Mentre il solutore CFD cattura la natura turbolenta del moto, la rete neurale restituisce una versione "mediata" del campo di moto. Questo fenomeno spiega lo scostamento numerico del 6.67% rilevato nel set di test. Questo risultato conferma che, per flussi ad alti numeri di Rayleigh e Reynolds simulati come stazionari, la rete neurale eccelle nel fornire una panoramica generale del sistema, pur presentando limiti nella risoluzione dettagliata. Da sottolineare anche come la rete neurale abbia avuto delle difficoltà nell'analisi dei flussi in prossimità delle feritoie della porta, indicando la necessità di andare a raffinare ulteriormente la struttura del grafo in quell'area.

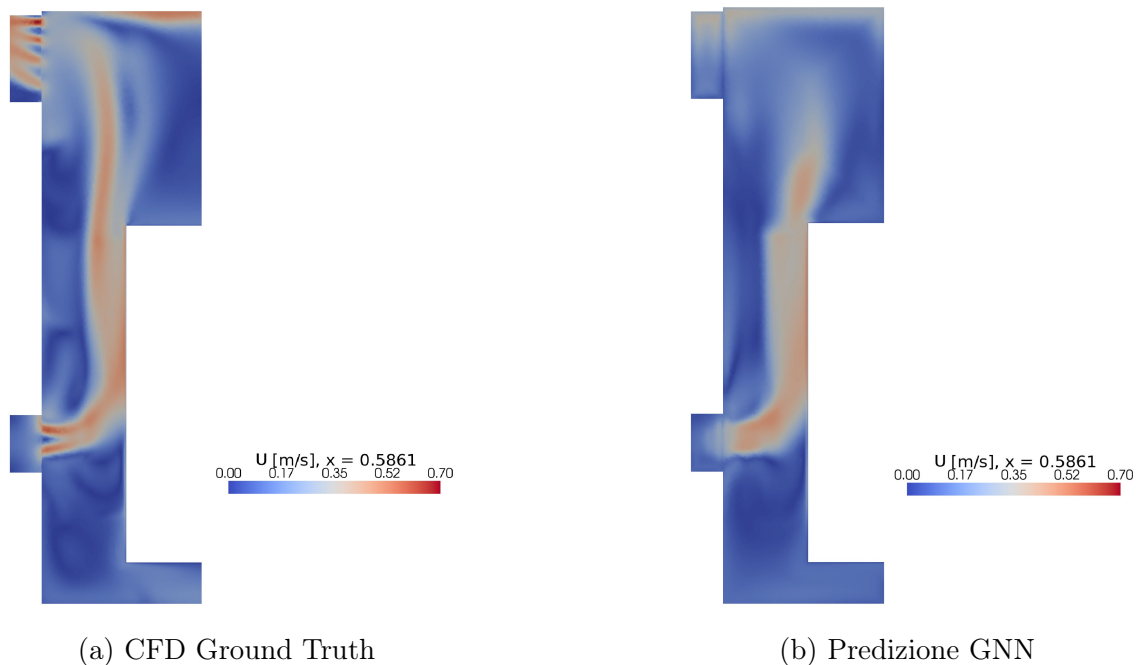


Figura 7.10: Confronto qualitativo del campo di velocità  $U$  [m/s] sulla sezione trasversale  $x$ .

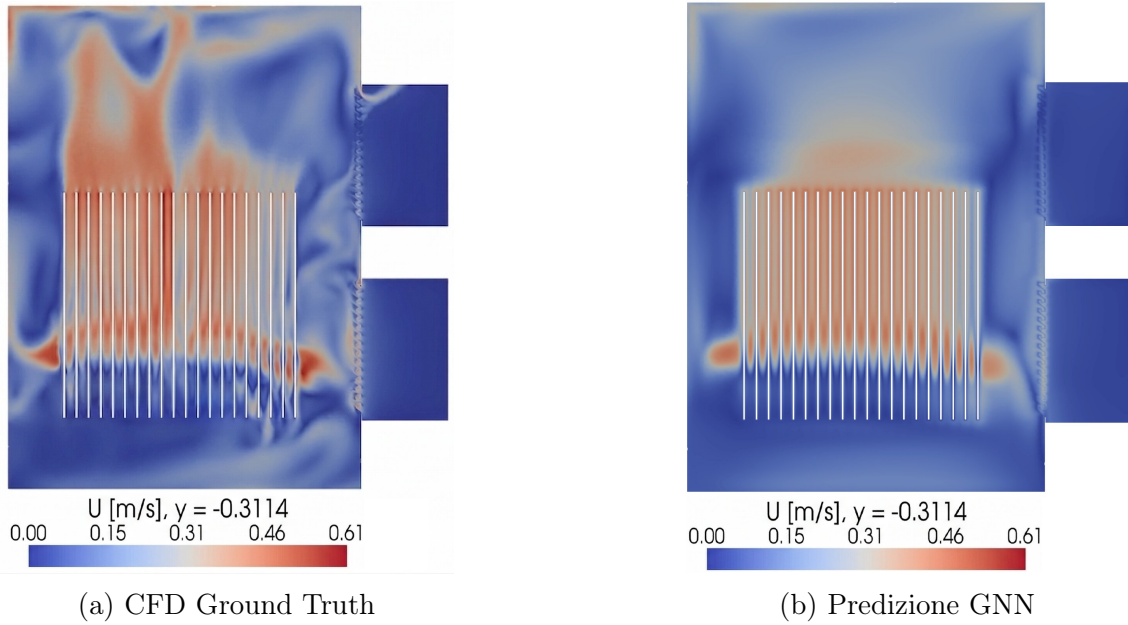


Figura 7.11: Confronto qualitativo del campo di velocità  $U$  [m/s] sulla sezione longitudinale  $y$ .

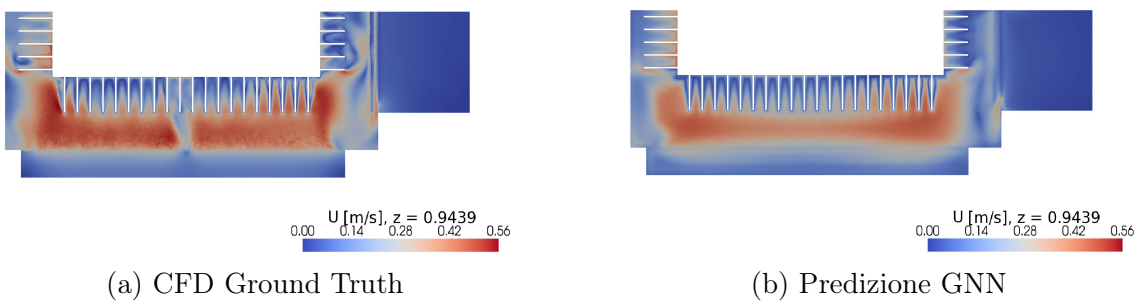


Figura 7.12: Confronto qualitativo del campo di velocità  $U$  [m/s] sulla sezione orizzontale  $z$ .

## 7.4 Analisi dei campi superficiali (VTP)

### 7.4.1 Validazione qualitativa: confronto visivo dei campi di pressione

In questa specifica analisi, la predizione esibisce una fedeltà superiore, caratterizzata da scostamenti minimi e quasi trascurabili rispetto ai dati di riferimento (*ground truth*). Tuttavia, si conferma la tendenza del modello a una lieve perdita di dettaglio in alcune zone. Tale fenomeno, pur non inficiando la validità globale della soluzione, è verosimilmente riconducibile a una cardinalità del dataset non ancora ottimale per la saturazione della capacità di apprendimento della rete.

In altre parole, pur con 100 campioni, la GNN opera una regolarizzazione statistica che predilige la stabilità della soluzione media rispetto alla cattura delle singularità fisiche più minute. È ragionevole ipotizzare che un ulteriore incremento del numero di *samples* nel dataset di addestramento permetterebbe al modello di mappare con maggiore incisività anche i gradienti più severi, riducendo ulteriormente quell'effetto di smussamento riscontrato nelle zone critiche del dominio.

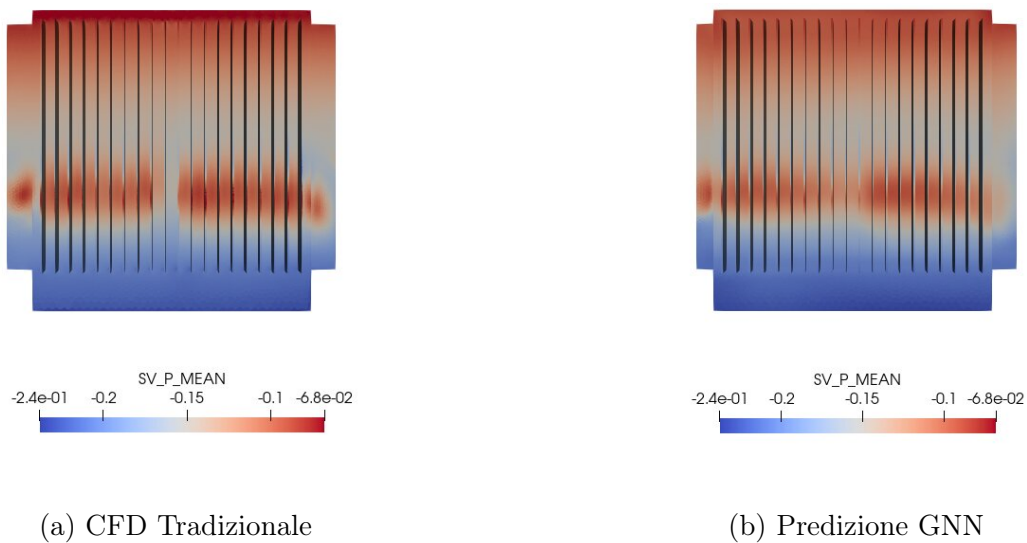


Figura 7.13: Confronto del campo di pressione  $p$  [Pa] sulle superfici del trasformatore.

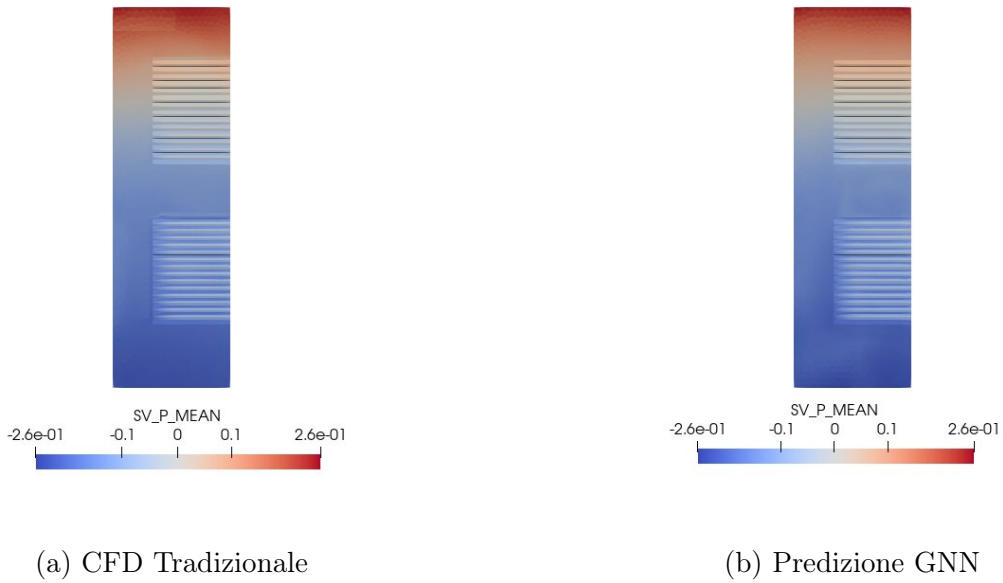


Figura 7.14: Confronto del campo di pressione  $p$  [Pa] sulle superfici del cabinato.

#### 7.4.2 Validazione qualitativa: confronto visivo dei campi di temperatura

Le considerazioni esposte per il campo di pressione trovano piena conferma anche nell'analisi della distribuzione termica. La rete neurale dimostra un'eccellente capacità predittiva nel mappare la fisica globale del sistema, identificando con precisione le zone termiche. Tuttavia, analogamente a quanto riscontrato in precedenza, si rileva una limitata risoluzione nei dettagli di piccola scala e una tendenza allo *smoothing*.

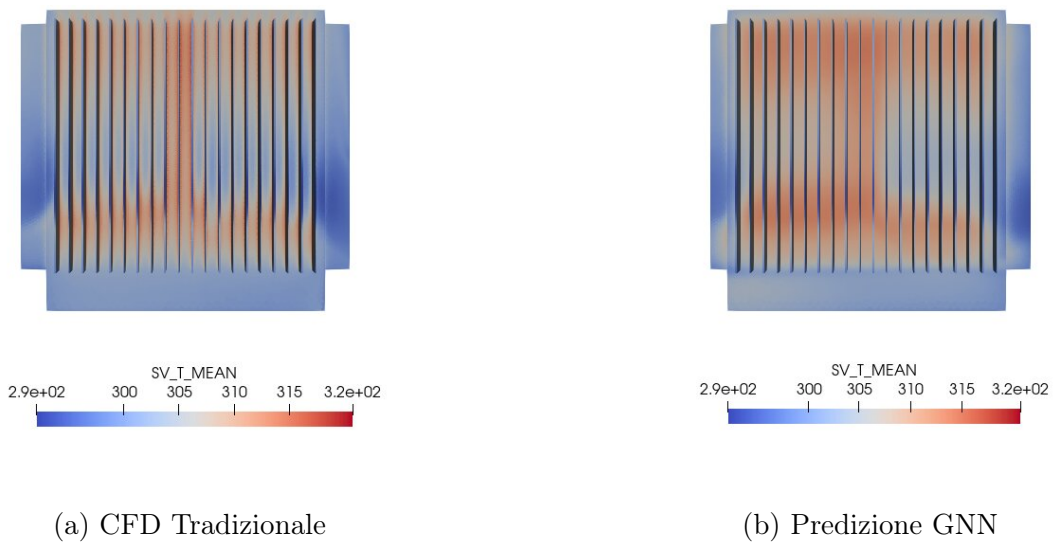
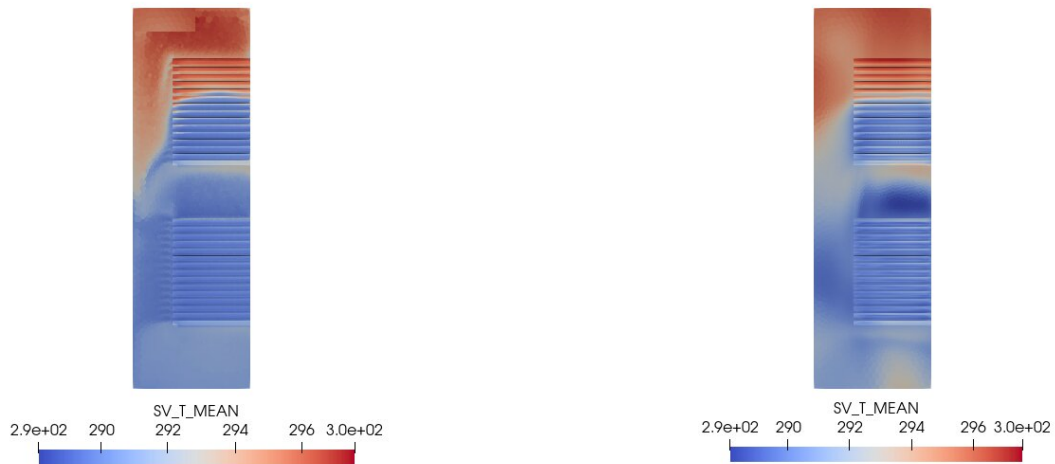


Figura 7.15: Confronto qualitativo del campo di temperatura  $T$  [°C] sulle superfici del trasformatore



(a) CFD Tradizionale

(b) Predizione GNN

Figura 7.16: Confronto qualitativo del campo di temperatura  $T$  [ $^{\circ}\text{C}$ ] sulla superficie del cabinato

## 7.5 Confronto prestazionale e vantaggio computazionale

Il confronto tra i dataset **VTP** (superficiale) e **VTU** (volumetrico) delinea un chiaro trade-off tra precisione e complessità del dominio. Mentre il modello superficiale garantisce un'accuratezza estremamente elevata ( $NRMSE_T < 3.5\%$ ), il modello volumetrico risente della natura turbolenta e intrinsecamente non stazionaria del flusso interno, portando l'errore sul modulo della velocità a circa il 6.67%. Tuttavia, la valutazione dell'efficacia di tale metodologia non può prescindere dall'analisi dei costi computazionali.

Come riassunto nella Tabella 7.3, il solutore CFD tradizionale operante su cluster HPC richiede tempi significativi per la convergenza di ogni singolo caso (si sottolinea che il tempo indicato è relativo alla sola convergenza, non viene perciò considerato il periodo necessario al datasampling), dovendo risolvere iterativamente le equazioni RANS su milioni di nodi. Al contrario, una volta completata la fase di addestramento, la rete neurale è in grado di generare la predizione dell'intero campo termofluidodinamico in pochi secondi (*inference time*) con necessità di risorse hardware minime (1 core e 100 Mb di Ram).

Tabella 7.3: Confronto dei tempi di calcolo: Solutore CFD tradizionale vs Rete Neurale.

Fase del Processo	Solutore CFD (Fluent)	Rete Neurale (GNN)
Preparazione e Meshing	~ 4-5 min	
Risoluzione (HPC Cluster 32 core)	~ 110-130 min / caso	-
Inference VTU (Predizione)	-	≈ <b>4 secondi</b>
Inference VTP (Predizione)	-	< <b>1 secondo</b>

*\*Escluso il tempo di addestramento iniziale.*

Questo abbattimento drastico dei tempi di calcolo del 99%, rappresenta il vero valore aggiunto della metodologia proposta. Sebbene si accetti un lieve scostamento numerico rispetto al solutore CFD, la capacità della rete neurale di fornire risultati quasi istantanei abilita scenari precedentemente preclusi alla CFD tradizionale. Un ulteriore vantaggio ottenuto dalla predizione, è che questa può essere utilizzata come inizializzazione del campo fluido per una simulazione CFD tradizionale andando ad eliminare l'errore analizzato in precedenza.

## 8. CONCLUSIONI

---

Il presente lavoro di tesi ha dimostrato con successo l'efficacia delle *Graph Neural Networks* (GNN), e nello specifico dell'architettura **X-MeshGraphNet**, come strumento per la risoluzione di problemi termofluidodinamici complessi. L'obiettivo primario è stato quello di evidenziare come l'intelligenza artificiale possa trasformare radicalmente la fase di progettazione tecnica, offrendo un'alternativa ai metodi di calcolo tradizionali laddove la rapidità di analisi è un fattore critico.

L'analisi condotta fornisce evidenze significative circa la validità del paradigma *AI-driven*, il cui vantaggio primario risiede nel drastico abbattimento dei tempi di attesa. Laddove un solutore CFD tradizionale richiede ore o persino giorni di calcolo intensivo, il modello neurale è in grado di fornire una predizione completa in pochi secondi; tale *speed-up* trasforma radicalmente la simulazione, rendendolo uno strumento di progettazione interattiva in tempo reale.

Parallelamente, i test hanno evidenziato un bilancio ottimale tra accuratezza e rapidità, con errori di predizione contenuti entro il 4% per i campi superficiali e un massimo del 6.67% per quelli volumetrici. Questi scostamenti, localizzati prevalentemente sulle componenti di velocità, risultano ampiamente accettabili nelle fasi preliminari di *design*, in cui la comprensione qualitativa e quantitativa dei macro-pattern fisici prevale sulla necessità di estrema precisione decimale. Infine, l'efficacia della metodologia è avvalorata dalla possibilità di adottare un approccio ibrido, in cui l'output della rete funge da mappa di inizializzazione per i solutori classici. Questa sinergia consente di raffinare la predizione dell'intelligenza artificiale in pochi istanti di calcolo numerico, riducendo drasticamente il numero di iterazioni necessarie alla convergenza finale del software tradizionale e garantendo, al contempo, la massima affidabilità del risultato.

In un'ottica di superamento delle attuali limitazioni, in particolare per quanto concerne la perdita di dettaglio nella predizione, si delineano due direzioni di sviluppo primarie. Da un lato, l'incremento dell'ampiezza del *dataset* appare fondamentale per affinare la sensibilità del modello verso i regimi turbolenti e i fenomeni caotici, potenziando la risoluzione locale attraverso una base informativa più estesa. Dall'altro, l'integrazione di reti neurali informate dalla fisica (PINN) permetterebbe di incorporare le leggi di conservazione e i vincoli fluidodinamici direttamente nel processo di ottimizzazione. Questo avanzamento metodologico assicurerebbe una coerenza fisica superiore e una maggiore robustezza nella generalizzazione, garantendo l'affidabilità del modello anche a fronte di geometrie inedite non precedentemente esplorate durante l'addestramento.

## BIBLIOGRAFIA

---

- [1] Nadezhda Kafadarova e Anna Andonova. “PCB thermal design improvement through thermal vias”. In: *Recent Advances in Circuits, Systems, Electronics, Control and Signal Processing (CSECS)*. Canary Islands, Spain, 2009, pp. 14–16.
- [2] Yanfeng Shen et al. “Thermal modeling and design optimization of PCB vias and pads”. In: *IEEE Transactions on Power Electronics* 35.1 (2019), pp. 882–900.
- [3] Minh-Nhat Nguyen et al. “Calculation limits of the homogeneous effective thermal conductivity approach in modeling of printed circuit board”. In: *20th International Workshop on Thermal Investigations of ICs and Systems*. IEEE, 2014, pp. 1–6.
- [4] Rafael Lopes Costa e Valeri Vlassov. “Simulation method for effective thermal conductivity determination of complex boards”. In: *2<sup>o</sup> Workshop em Engenharia e Tecnologia Espaciais*. 2011.
- [5] Cheng Gao et al. “Steady-state thermal analysis and layout optimization of DC/DC converter”. In: *Prognostics and System Health Management Conference (PHM-2014 Hunan)*. IEEE, 2014, pp. 405–409.
- [6] Eric Monier-Vinard et al. “State of the art of thermal characterization of electronic components using Computational Fluid Dynamic tools”. In: *International Journal of Numerical Methods for Heat & Fluid Flow* (2016). DOI: 10.1108/HFF-10-2016-0380.
- [7] R. Bouhaddiche, S. Bouazabia e I. Fofana. “Thermal Modelling of Power Transformer”. In: *19th IEEE International Conference on Dielectric Liquids (ICDL)*. Manchester, United Kingdom: IEEE, 2017, pp. 1–4. ISBN: 978-1-5090-4877-9.
- [8] Hui Chen et al. “Numerical simulation and optimisation design for ventilation and heat dissipation in high-temperature and high-load indoor substations”. In: *Case Studies in Thermal Engineering* 59 (2024), p. 104502. DOI: 10.1016/j.csite.2024.104502.
- [9] Mohamad Kanaan e Khaled Chahine. “CFD study of ventilation for indoor multi-zone transformer substation”. In: *International Journal of Heat and Technology* 36.1 (2018), pp. 95–101. DOI: 10.18280/ijht.360112.

- [10] Fan Ou et al. “Numerical Study on Thermal Characteristics of Cabin-mounted Dry-type Transformers”. In: *2022 2nd International Conference on Electrical Engineering and Mechatronics Technology (ICEEMT)*. IEEE, 2022, pp. 129–132. DOI: 10.1109/ICEEMT56362.2022.9862615.
- [11] Fabio Nardecchia et al. “CFD Study of the Impact of an Electrical Power Transformer on a Historical Building: Assessment and Solutions”. In: *Applied Sciences* 14.23 (2024), p. 10827. DOI: 10.3390/app142310827.
- [12] Haomai Zhang et al. “Optimizing air inlet designs for enhanced natural ventilation in indoor substations: A numerical modelling and CFD simulation study”. In: *Case Studies in Thermal Engineering* 59 (2024), p. 104408. DOI: 10.1016/j.csite.2024.104408.
- [13] P. Satheysh Pavai et al. “Numerical analysis of the thermal management strategies of electric vehicle battery: A review”. In: *Archives of Computational Methods in Engineering* 31.5 (2024), pp. 3051–3090.
- [14] S. Hoelle, S. Zimmermann e O. Hinrichsen. “3D Thermal Simulation of Thermal Runaway Propagation in Lithium-Ion Battery Cell Stack: Review and Comparison of Modeling Approaches”. In: *Journal of The Electrochemical Society* 170.6 (2023), p. 060516. DOI: 10.1149/1945-7111/acd966.
- [15] Paolo Cicconi, Daniele Landi e Michele Germani. “Thermal analysis and simulation of a Li-ion battery pack for a lightweight commercial EV”. In: *Applied Energy* 192 (2017), pp. 159–177.
- [16] Hongjie Lu, Zhongbin Zhang e Liu Yang. “A review on airflow distribution and management in data center”. In: *Energy and Buildings* 179 (2018), pp. 264–277. DOI: 10.1016/j.enbuild.2018.08.055.
- [17] Wen-Xiao Chu e Chi-Chuan Wang. “A review on airflow management in data centers”. In: *Applied Energy* 236 (2019), pp. 179–195. DOI: 10.1016/j.apenergy.2018.11.070.
- [18] M. Acton, J. Booth e D. Paci. *2025 Best Practice Guidelines for the EU Code of Conduct on Data Centre Energy Efficiency*. Report EUR 40267. Joint Research Centre (JRC), European Commission, 2025.
- [19] Emelie Wibron, Anna-Lena Ljung e T. Staffan Lundström. “Computational Fluid Dynamics Modeling and Validating Experiments of Airflow in a Data Center”. In: *Energies* 11.3 (2018), p. 646. DOI: 10.3390/en11030646.
- [20] Yann LeCun, Yoshua Bengio e Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444. DOI: 10.1038/nature14539.

- [21] Maziar Raissi, Paris Perdikaris e George E Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* 378 (2019), pp. 686–707. DOI: 10.1016/j.jcp.2018.10.045.
- [22] Tobias Pfaff et al. “Learning Mesh-Based Simulation with Graph Networks”. In: *International Conference on Learning Representations (ICLR)*. 2021.
- [23] Zonghan Wu et al. “A Comprehensive Survey on Graph Neural Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (2021), pp. 4–24. DOI: 10.1109/TNNLS.2020.2978386.
- [24] Peter W. Battaglia et al. “Relational inductive biases, deep learning, and graph networks”. In: *arXiv preprint arXiv:1806.01261* (2018).
- [25] Mohammad Amin Nabian et al. “X-MeshGraphNet: Scalable Multi-Scale Graph Neural Networks for Physics Simulation”. In: *arXiv preprint arXiv:2411.17164* (2024).
- [26] Tim Mucci. *Cosa sono l’overfitting e l’underfitting?* IBM Think Topics. IBM. 2024. URL: <https://www.ibm.com/it-it/think/topics/overfitting-vs-underfitting> (visitato il giorno 31/03/2025).
- [27] IBM Education. *Cos’è l’addestramento del modello?* IBM Think Topics. IBM. 2024. URL: <https://www.ibm.com/it-it/think/topics/model-training> (visitato il giorno 31/03/2026).
- [28] Nikolay Vakrilov Vakrilov, Diana Velkova Stoyanova e Nadezhda Miteva Kafadarova. “Numerical thermal analysis of the PCB construction impact”. In: *XI National Conference with International Participation (ELECTRONICA)*. IEEE, 2020, pp. 1–5.
- [29] Alvaro Sanchez-Gonzalez et al. “Learning to Simulate Complex Physics with Graph Networks”. In: *International Conference on Machine Learning (ICML)*. PMLR. 2020, pp. 8459–8468.
- [30] Michael D McKay, Richard J Beckman e William J Conover. “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code”. In: *Technometrics* 21.2 (1979), pp. 239–245.
- [31] IBM Education. *Cos’è la rete neurale?* IBM Think Topics. IBM. 2024. URL: <https://www.ibm.com/it-it/think/topics/neural-networks> (visitato il giorno 31/03/2025).

