UNIVERSITÀ DEL PIEMONTE ORIENTALE

Dipartimento di Scienze e Innovazione Tecnologica

Corso di Laurea Magistrale in Intelligenza Artificiale e Innovazione Digitale

Tesi di Laurea Magistrale

Explainable AI in Sanità: Storytelling e Modelli Transformer per l'Ottimizzazione del Length of Stay

Relatore
Prof. Giorgio Leonardi

Firmato digitalmente da Giorgio Leonardi
Data: 09.04.2026 12:09:39 CEST
Organizzazione: UNIVERSITA' DEGLI STUDI DEL PIEMONTE ORIENTALE/01943490027

Candidato
Simone Garau

Anno Accademico 2024/2025

Questa opera è distribuita con la licenza **Creative Commons Attribuzione 4.0 Internazionale (CC BY 4.0)**.
È permesso condividere e adattare l'opera anche per scopi commerciali a condizione di attribuire adeguatamente l'autore: Simone Garau.
Testo completo della licenza: https://creativecommons.org/licenses/by/4.0/
Codice legale: https://creativecommons.org/licenses/by/4.0/legalcode

Abstract

L'ottimizzazione del *Length of Stay* (LOS) ospedaliero è una sfida fondamentale per conciliare efficienza economica e qualità delle cure. Questa tesi, sviluppata all'interno del progetto TEXLOS [1], propone un sistema innovativo d'intelligenza artificiale per la classificazione retrospettiva e l'individuazione dei colli di bottiglia logistici responsabili di degenze critiche ($\geq$ 20 giorni) partendo dai *Log degli Eventi* clinici.

L'approccio metodologico estende le metodologie del *Process Mining* tradizionale introducendo il paradigma dello **Storytelling**: una traduzione semantica dei pattern cronologici in narrazioni in linguaggio naturale. Tali testi alimentano un classificatore Transformer (`bert-base-uncased`), addestrato mediante *Focal Loss* per mitigare il severo sbilanciamento dei dati ospedalieri.

Avendo accertato l'apprendimento strutturale del modello, viene risolto il classico problema clinico della scatola nera ("black box"). Si applica una variante adattiva degli **Integrated Gradients** – supportata da una parallela logica di ricomposizione semantica dei token frammentati da BERT – per interrogare a posteriori le scelte della rete. Assegnando matematicamente un punteggio esplicito d'impatto ad ogni singola azione nosocomiale processata, si fornisce al *management* medico un *cruscotto diagnostico di processo* rigoroso per individuare empiricamente le inefficienze ospedaliere.

Indice

1. Introduzione

1.1 Contesto e Motivazione

L'ottimizzazione dei processi sanitari rappresenta oggi una delle sfide principali per le amministrazioni ospedaliere, spinte dalla necessità di coniugare la qualità delle cure con la sostenibilità economica delle strutture. In questo contesto, il controllo e la previsione del *Length of Stay* (LOS), ovvero la durata della degenza ospedaliera dei pazienti, rivestono un ruolo cruciale. Un LOS eccessivamente prolungato non solo grava sulle risorse finanziarie e strutturali dell'ospedale, come la disponibilità di posti letto e il carico di lavoro del personale, ma espone anche i pazienti a un maggiore rischio di infezioni nosocomiali e complicazioni cliniche.

Storicamente, l'analisi dei percorsi clinici si è avvalsa del *Process Mining*, una disciplina che consente di estrarre conoscenza preziosa dai log degli eventi registrati dai sistemi informativi sanitari, esplorando e modellando i flussi di cura. Tuttavia, le metodologie classiche di Process Mining risultano spesso insufficienti nel catturare la complessità semantica e la sequenzialità non lineare intrinseche alle tracce cliniche dei pazienti, spesso frammentate e altamente variabili.

Negli ultimi anni, l'evoluzione dell'Intelligenza Artificiale ha aperto nuove prospettive, in particolare grazie all'introduzione dei modelli basati sull'architettura Transformer [2] e all'avvento dei Large Language Models (LLM). Il presente lavoro adotta proprio questa prospettiva: l'intuizione fondamentale risiede nell'assimilare una sequenza di eventi clinici a una narrazione, traducendo le tracce in veri e propri documenti di testo (*Storytelling*). I modelli linguistici avanzati, come BERT, possiedono infatti la capacità intrinseca di comprendere il contesto bidirezionale di una sequenza temporale, cogliendo dipendenze a lungo termine e sfumature che i modelli statistici tradizionali ignorano.

Nonostante le elevate prestazioni predittive raggiunte dalle reti neurali profonde, la loro natura di "scatola nera" (*black box*) costituisce un ostacolo all'adozione pratica in ambito clinico. È fondamentale chiarire fin dall'inizio il perimetro di applicazione dell'Intelligenza Artificiale in questo lavoro: il classificatore BERT non è concepito come un oracolo predittivo "in tempo reale" per anticipare l'esito degenziale di nuovi pazienti al momento dell'ingresso in ospedale. Piuttosto, esso viene impiegato per analizzare retrospettiva-

mente log *storici* consolidati. Addestrare il modello a riconoscere con grande precisione se una traccia *ha avuto* un Length of Stay maggiore di 20 giorni serve primariamente a comprovare che la rete ha *imparato* a discriminare i pattern clinici.

Solo a fronte di questo solido apprendimento strutturale entra in gioco l'Explainable AI (XAI). Poiché il modello "sa" riconoscere l'esito della degenza, è possibile interrogarlo *a posteriori* per decostruirne le logiche interne. Al medico o all'amministratore ospedaliero servono le motivazioni interpretabili per comprendere l'inefficienza strutturale, l'azione specifica o il collo di bottiglia processuale in un gruppo di pazienti critici.

1.2 Obiettivi della Tesi

Questa tesi si colloca all'interno del progetto di ricerca TEXLOS [1] (*Investigating the Impact of Outpatient Services on Length of Stay: an Easily Interpretable Approach*) e si pone due obiettivi primari fusi in un'unica pipeline di analisi.

Il primo obiettivo è lo sviluppo di un sistema classificatorio basato sull'architettura Transformer per discriminare le degenze che hanno superato una soglia critica, fissata a 20 giorni. Il modello fa leva sull'approccio dello *Storytelling*, trasformando i log in formato XES in narrazioni testuali ed impiegando `bert-base-uncased` per catturare le connessioni logiche tra i vari eventi clinici.

Il secondo e più innovativo obiettivo riguarda l'interpretabilità post-hoc del modello predittivo. Attraverso l'uso di tecniche avanzate di Explainable AI, in particolare l'algoritmo degli Integrated Gradients (IG) e le mappe di attenzione, il lavoro mira a estrapolare e quantificare l'impatto di ogni singola azione clinica sulla durata prolungata della degenza. L'obiettivo ultimo è fornire alle amministrazioni ospedaliere uno strumento diagnostico di processo capace di evidenziare i colli di bottiglia storici, traducendo token sparsi in azioni cliniche leggibili provviste di un punteggio d'impatto rigoroso.

1.3 Struttura dell'Elaborato

Il presente documento è organizzato come segue:

- Il **Capitolo 2** esplora lo stato dell'arte, partendo dall'uso del Process Mining in ambito clinico fino all'impiego del Deep Learning e degli LLM per la classificazione di tracce. Particolare attenzione viene posta ai concetti di Explainable AI e all'approccio Storytelling, citando progetti pionieristici come LEGOLAS.

- Il **Capitolo 3** descrive i passaggi tecnici del lavoro: preprocessing dei dati, costruzione degli embedding, configurazione di BERT, gestione dello sbilanciamento delle classi mediante Focal Loss e setup sperimentale. È incluso anche un esperimento esplorativo di data augmentation tramite LLM generativi.

- Il **Capitolo 4** riporta i risultati della classificazione in termini di accuratezza e metriche di valutazione. Successivamente, dettaglia criticamente l'implementazione adattativa degli Integrated Gradients e illustra il meccanismo di mappatura dei sub-token, mostrando concretamente come i risultati dell'algoritmo permettono di individuare i colli di bottiglia nel processo clinico.
- Il **Capitolo 5** riassume i traguardi raggiunti dalla tesi, evidenzia i limiti metodologici intrinseci dell'approccio proposto, chiarendo la differenza tra correlazione e causalità, e traccia possibili traiettorie per gli sviluppi futuri.

2. Stato dell'Arte

2.1 Process Mining in Ambito Clinico

Il settore sanitario genera quotidianamente moli di dati strutturati e non strutturati che riflettono l'interazione dei pazienti con le strutture ospedaliere, rendendo i *Log degli Eventi* una miniera di informazioni fondamentale per analizzare oggettivamente i percorsi di cura. La gestione e il miglioramento di questi percorsi sono sfide affrontate storicamente attraverso il *Process Mining*.

In ambito clinico, il Process Mining consiste in una serie di tecniche volte a estrarre modelli di processo dai dati (ad esempio tracce esportate in formato XES) per scoprire, monitorare e ottimizzare l'erogazione delle cure mediche e l'efficienza dei servizi. Gli algoritmi di Process Discovery o di Conformance Checking permettono alle amministrazioni di individuare anomalie procedurali e deviazioni dai percorsi terapeutici codificati (Clinical Pathways). Sebbene questi metodi godano di robustezza matematica per mappare il sistema, si sono spesso rivelati rigidi di fronte alle variabili caotiche e all'alta stocasticità dei flussi ospedalieri, che esibiscono dipendenze temporali complesse difficili da cogliere attraverso soli grafi deterministici.

2.2 Evoluzione del Deep Learning: Da RNN a Transformer

La necessità di mitigare le carenze del Process Mining basato su regole si è tradotta nell'integrazione di tecniche di Data Mining esplorativo, giungendo ben presto all'utilizzo del *Deep Learning*.

Inizialmente, l'esigenza di elaborare tracce cliniche di lunghezza variabile ha portato all'impiego delle architetture ricorrenti, in primis Recurrent Neural Networks (RNN) e Long Short-Term Memory (LSTM), poiché naturalmente predisposte per dati sequenziali. Tuttavia, il limite strutturale delle RNN legato alla dispersione del gradiente (*vanishing gradient*) e l'incapacità intrinseca di gestire parallelismi e interdipendenze a lungo raggio in storie cliniche complesse ne hanno fortemente limitato l'efficacia pratica.

Un fondamentale cambio di paradigma avviene nel 2017 con l'introduzione dell'architettura Transformer ad opera di Vaswani et al. [2]. A differenza delle RNN, il Transformer elabora l'intera sequenza in parallelo, senza cicli ricorrenti: il meccanismo cardine, la *Self-Attention*, calcola per ogni token il grado di rilevanza semantica rispetto a tutti gli altri token della sequenza in un unico passaggio. Replicando queste operazioni su più "teste" indipendenti (*Multi-Head Attention*) e stratificandole in profondità, la rete costruisce una rappresentazione contestuale densa dell'intera sequenza, cogliendo dipendenze a lungo raggio che le architetture ricorrenti non riuscivano a catturare.

È importante distinguere tre macro-famiglie di architetture derivate dal Transformer originario:

- **Encoder-only (es. BERT)**: conservano solo la componente in grado di leggere l'intera sequenza in modo bidirezionale. Sono eccellenti per task di classificazione, analisi del sentiment o comprensione del testo, poiché possono catturare il contesto sia a destra che a sinistra di un token senza restrizioni temporali.

- **Decoder-only (es. GPT)**: sono modelli auto-regressivi progettati per generare testo prevedendo il token successivo. Leggono la sequenza in senso unidirezionale, mascherando i token futuri.

- **Encoder-Decoder (es. BART)**: utilizzano entrambe le componenti per task di tipo sequence-to-sequence, come la traduzione automatica.

In questo lavoro di tesi, la scelta è ricaduta su un approccio *Encoder-only* poiché lo scopo primario non è generare nuove tracce cliniche, ma classificarle e comprenderne le dinamiche cronologiche globali, traendo massimo vantaggio dal contesto bidirezionale intero.

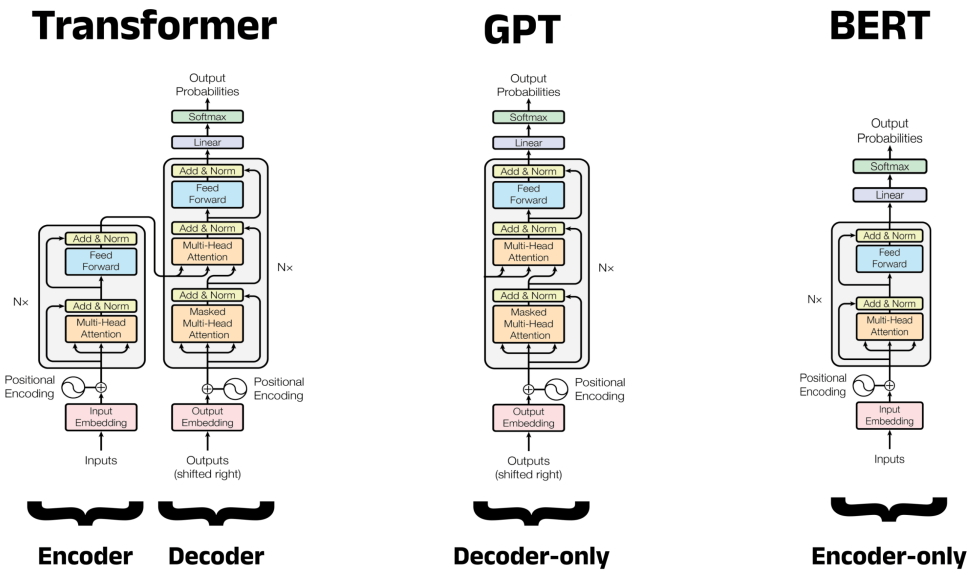

Figura 2.1: Schema riassuntivo dell'architettura Transformer nelle tre forme: Encoder-only, Decoder-only e Encoder-Decoder.

Parallelamente all'evoluzione architetturale, un passo decisivo verso l'analisi semantica delle tracce ospedaliere è stato il superamento delle classiche codifiche One-Hot, tipiche dell'era pre-Transformer, in favore di *embedding* linguistici contestuali. L'avvento dei Large Language Models (LLM), tra cui Bidirectional Encoder Representations from Transformers (BERT) [3], ha consentito di addestrare vettori profondamente contestuali su vasti corpus di dati testuali non etichettati. Nel caso clinico, ciò si traduce in un modello capace di osservare e valutare un'intera traccia di eventi contemporaneamente e in modo bidirezionale (in parallelo), decodificando inefficienze e derive diagnostiche senza i colli di bottiglia computazionali del parsing puramente passo-passo.

2.3 L'Approccio Storytelling: Il Progetto LEGOLAS

Affinché i modelli Transformer, ed in particolare BERT, possano elaborare con profitto i dati tratti da log storici di Process Mining, si è resa necessaria una forte innovazione nella tecnica di rappresentazione dell'informazione (*embedding*). Nasce da questa esigenza l'approccio *Storytelling*.

L'idea fondamentale, che ispira anche il presente lavoro di tesi, si radica nel recente e pionieristico progetto *LEGOLAS* (Leveraging a Large Language Model to Predict Hospital Admissions of Emergency Department Patients) [4]. L'intuizione principale di LEGOLAS consiste nel superare la mera concatenazione meccanica di eventi e timestamp discreti per spingersi nella generazione di veri e propri documenti di testo coerenti, in linguaggio naturale.

La traccia di un paziente non viene più rappresentata come una sequela di array numerici estratti dal registro dell'ospedale, ma viene trascritta sotto forma di "storia" cro-

nologica, in modo che il testo contenga esplicitamente la semantica degli eventi temporali e delle distanze tra un'azione clinica e l'altra. Poiché BERT è un LLM pre-addestrato sul linguaggio naturale umano, alimentarlo con racconti clinici strutturati ad hoc garantisce un notevole incremento prestazionale in fase di fine-tuning per task di classificazione. Rispetto agli approcci classici basati su rappresentazioni one-hot o array sparsi, questa strategia mitiga i problemi di dimensionalità esplosiva, permettendo al modello di valutare non solo la presenza di un evento, ma di decodificarne il *senso clinico* globale nel percorso complessivo del paziente.

2.4 Explainable AI (XAI) per i Modelli Transformer

Per affrontare la criticità legata alla scarsa interpretabilità dei modelli a scatola nera (*black-box*), l'Explainable AI (XAI) rappresenta oggi una necessità metodologica ineludibile. Nel caso clinico, gli algoritmi intrinsecamente trasparenti come il *Decision Tree* non reggono il confronto prestazionale col Deep Learning, rendendo cruciale sviluppare metodi post-hoc (applicabili a modello addestrato) per interpretare l'architettura Transformer.

Ricorrere alle sole *Mappe di Attenzione* si rivela spesso insufficiente, poiché esse riflettono "pesi" posizionali generici, affetti da *noise* e non direttamente causali rispetto all'output di classificazione finale. Per ottenere metriche di *feature importance* rigorose, la letteratura consiglia l'adozione degli **Integrated Gradients** (IG), un metodo di attribuzione proposto originariamente da Sundararajan et al. [5] nel dominio della *Computer Vision* per quantificare il contributo dei singoli pixel a una predizione visiva. In questa tesi, la tecnica viene traslata al dominio testuale per misurare l'impatto dei singoli *token* elaborati da BERT sulla probabilità di un LOS prolungato.

Gli IG si pongono lo scopo di assegnare un'importanza predittiva alle singole *feature* in ingresso ($x \in \mathbb{R}^n$), partendo da una *baseline* informativa neutra ($x' \in \mathbb{R}^n$), che nel contesto testuale della tesi corrisponde tipicamente ai token di *padding*.

2.4.1 Fondamenti Assiomatici e Debolezza dei Metodi Standard

Il metodo IG è stato formulato per soddisfare rigorosamente due assiomi teorici, spesso falliti dai meccanismi XAI preesistenti:

1. **Sensitivity(a):** Se un input clinico e la *baseline* differiscono per una sola azione medica ma portano a predizioni diverse, tale perturbazione deve ricevere un'attribuzione non nulla. I metodi classici basati sui gradienti standard (e.g. *Guided Back-propagation*) violano questo assioma, poiché il gradiente locale può annullarsi improvvisamente (causa appiattimento delle attivazioni ReLU) ancor prima che la rete registri la variazione predittiva globale.

2. **Implementation Invariance:** Reti funzionalmente equivalenti (le cui uscite coincidono a parità di input) devono produrre identiche attribuzioni, indipendentemente dalla loro stratificazione interna. I metodi che ricorrono a gradienti discreti (come *DeepLift*) falliscono questo test, in quanto la regola della catena differenziale non si generalizza coerentemente nel discreto.

2.4.2 Formulazione Matematica dell'Integrale e Completeness

Gli IG appartengono alla classe dei *Path Methods*, che calcolano l'integrale dei gradienti lungo una curva di interpolazione nello spazio vettoriale. L'algoritmo IG sceglie specificatamente l'unica traiettoria rigorosa per preservare la simmetria formale: un percorso rettilineo (*straightline path*) fra x' e x. Per l'i-esima dimensione dell'input, l'attribuzione continua di un'azione processuale è calcolata come:

$$IntegratedGrads_i(x) ::= (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha \tag{2.1}$$

Dove F è la funzione predittiva globale del Transformer e α è la costante di interpolazione numerica. Questa rigorosa formulazione matematica garantisce il rispetto di un terzo assioma di cruciale importanza: la **Completeness**. Sfruttando il teorema fondamentale del calcolo per gli integrali di linea, la somma esatta di tutte le *feature attribution* assegnate alle parole in ingresso coincide sempre con la differenza matematica tra lo score predittivo predetto per la stringa corrente e lo score della baseline fittizia neutra:

$$\sum_{i=1}^{n} IntegratedGrads_i(x) = F(x) - F(x') \tag{2.2}$$

La rigorosa formulazione matematica, esente da euristiche arbitrarie, fa degli *Integrated Gradients* lo strumento ottimale per interpretare le logiche interne dei modelli a "scatola nera": quantificare l'impatto clinico token per token è la chiave metodologica per fornire alle direzioni sanitarie risposte chiare sui propri colli di bottiglia logistici.

3. Metodologia

3.1 Estrazione e Preprocessing dei Dati

Il punto di partenza dell'esperimento è costituito da log ospedalieri estratti in formato XES. L'obiettivo primario di questa fase iniziale è strutturare e formattare tali log non strutturati per renderli digeribili all'architettura neurale sottostante il Transformer.

3.1.1 Analisi Esplorativa del Dataset (EDA)

Per acquisire contezza della natura e del volume dei log estratti pre-elaborazione, è stata condotta un'Analisi Esplorativa dei Dati (EDA) sviluppando una pipeline automatizzata in Python (basata principalmente sulla libreria `pandas` per l'efficiente gestione e *chunking* massivo delle estrazioni). Il dataset grezzo comprende 7.393 casi clinici unici (singoli pazienti ospitati), all'interno dei quali sono registrati 88.115 eventi granulari totali non etichettati.

Si rileva contestualmente l'ingente sbilanciamento delle classi, cardine della sfida analitica affrontata in questo progetto: su 7.393 pazienti totali, ben 6.584 esibiscono una degenza classificabile come normale (classe 0), e solamente 809 costituiscono campioni d'anomalia con degenza critica ≥ 20 giorni (classe 1).

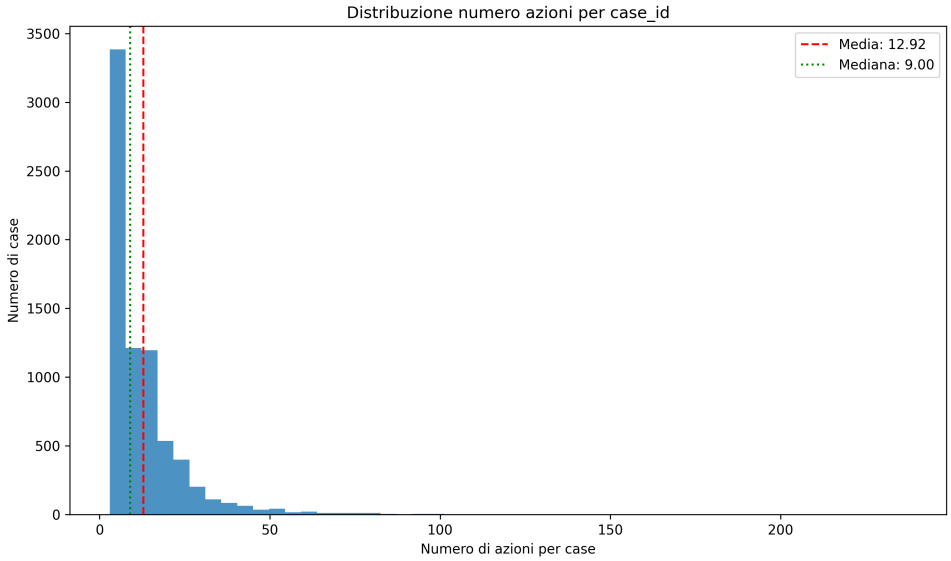

Figura 3.1: Distribuzione statistica globale del numero di azioni processuali registrate per singolo caso clinico. Il grafico si caratterizza per una spiccata asimmetria (right-skewed), evidenziando come una concentrazione primaria di interazioni ospedaliere rapide all'ingresso si contrapponga ad una "lunga coda" indicante pazienti complessi trattenuti in cure iterative stratificate.

Come illustrato in Figura 3.1, il numero di azioni per paziente presenta una forte asimmetria destra (*right-skewness*): la media è di 12.92 azioni, mentre la mediana scende a 9.00, segnalando che la distribuzione è tirata verso l'alto da una "lunga coda" (*long tail*) di pazienti complessi con decine o centinaia di re-interazioni.

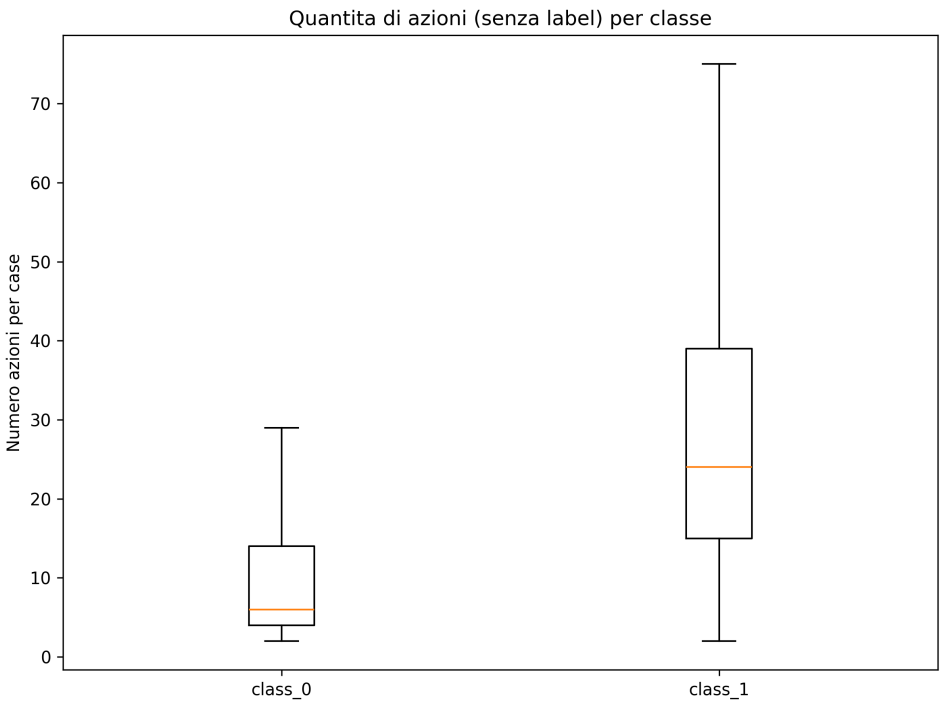

Figura 3.2: Boxplot di confronto diretto tra il numero d'interazioni erogate ai pazienti di classe 0 (degenza limitata) e di classe 1 (degenza prolungata). Risalta ad occhio nudo la traslazione dei quartili a sfavore della classe anomala, gravata intrinsecamente da percorsi ben più caotici, iterativi ed onerosi quantitativamente per la clinica.

Tale diversità emerge chiaramente analizzando le durate per classe (Figura 3.2): appare evidente il divario numerico, con la classe 1 (long-LOS) caratterizzata da percorsi mediani molto più lunghi e stratificati rispetto ai pazienti con degenti brevi. Questa evidenza giustifica l'adozione di un'architettura avanzata come BERT. A differenza di una stringa Markoviana classica che frammenterebbe i calcoli su sequenze enormi, l'architettura basata su *Self-Attention* garantisce un'elaborazione in parallelo per mappare le dipendenze logiche tra decine di step ospedalieri.

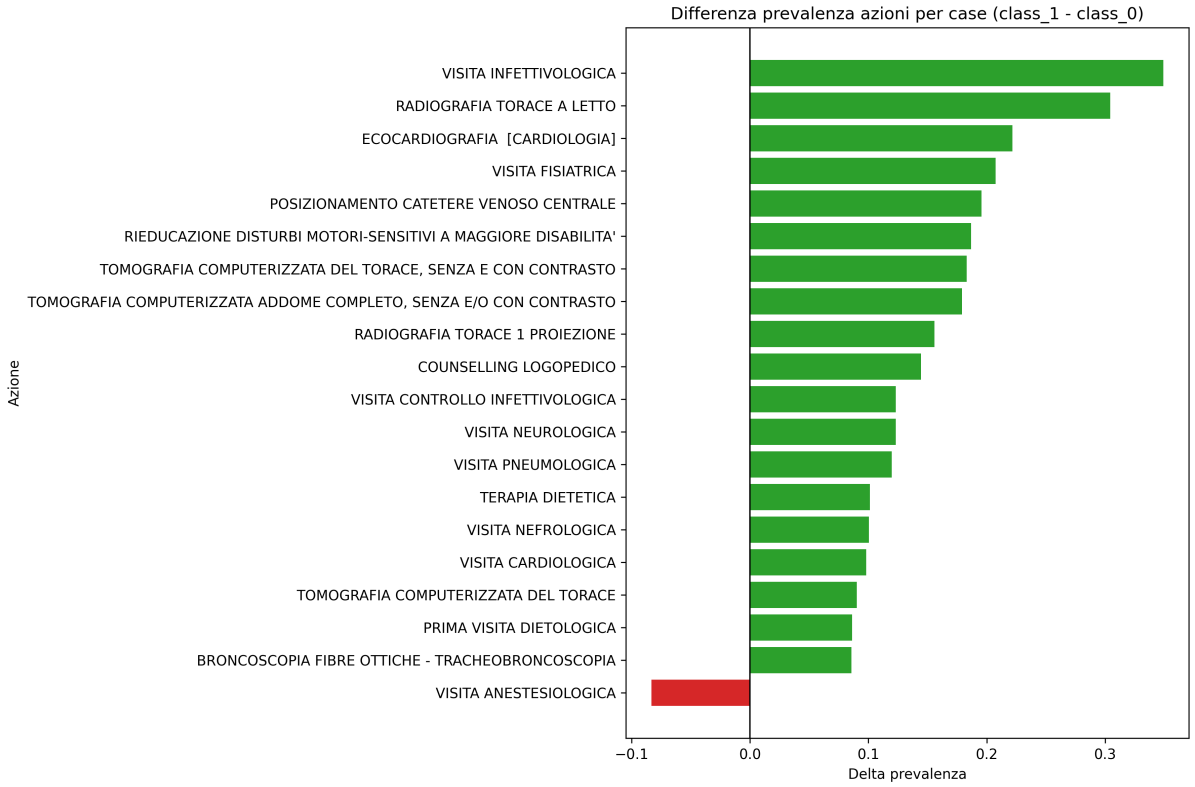

Figura 3.3: Panoramica estrapolata dei delta di prevalenza percentuale (sbilanciamento) per le singole azioni ospedaliere fortemente correlate in modo endogeno alla classe d'emergenza logistica 1.

Un'ulteriore esplorazione evidenzia la distribuzione asimmetrica delle frequenze delle singole procedure (Figura 3.3). Risultano particolarmente pronunciati gap associati ad alta probabilità critica per eventi clinici come la "Visita Infettivologica", la "Radiografia Torace a Letto" o il "Posizionamento Catetere Venoso Centrale". Pur trattandosi di statistiche esplorative prive di valore causale formale, tali osservazioni anticipano i pattern che il modello imparerà a pesare durante il training e che verranno interpretati tramite XAI nel Capitolo 4.

3.1.2 Approcci e Formattazione degli Embedding

Poiché il task ingegneristico mira strettamente ad accoppiare eventi clinici elaborati a corretto formato ed interfacciandoli poi col *Learning* del Transformer, è stata condotta una meticolosa opera preliminare testando tre filosofie distinte di *embedding*:

1. **Azioni cliniche + Binning Temporale:** In questo approccio, il preprocessing concatena coppie formate dall'azione clinica e da un token temporale speciale. Il timing continuo tra le azioni viene discretizzato (*binning*) in categorie predefinite. Un esempio di traccia processata è il seguente:


```
{"case_id": "5870415",
 "tokens": ["Complete abdomen ultrasound", "_TIME_DELTA_START_",
            "Computerized tomography of the skull",
            "_TIME_DELTA_START_", "Complete abdomen ultrasound",
            "_TIME_DELTA_LONG_", "Computerized tomography of the skull",
            "_TIME_DELTA_MEDIUM_"],
 "label_id": 0
}
```


   La scelta delle soglie di discretizzazione non è arbitraria ma *data-driven*: le categorie sono state definite analizzando la distribuzione empirica dei delta temporali nel dataset, riportata in Tabella 3.1. La distribuzione rivela una struttura bimodale con la grande maggioranza degli intervalli concentrata entro la finestra dei 7 giorni (97.0%), e una coda esigua di eventi molto dilazionati. Le soglie sono state quindi fissate sui breakpoint clinici naturali — un'ora, un giorno, sette giorni e trenta giorni — che separano fasi di cura operativamente distinte: accertamenti rapidi, degenza standard, ricovero prolungato e cronicità.

Token	Intervallo	Conteggio	Freq.
_TIME_DELTA_START_	$\Delta = 0$ s	5 842	6.9%
_TIME_DELTA_IMMEDIATE_	$0 < \Delta \leq 1$h	19 456	23.1%
_TIME_DELTA_SHORT_	$1\text{h} < \Delta \leq 1\text{gg}$	29 887	35.4%
_TIME_DELTA_MEDIUM_	$1\text{gg} < \Delta \leq 7\text{gg}$	26 649	31.6%
_TIME_DELTA_LONG_	$7\text{gg} < \Delta \leq 30\text{gg}$	2 296	2.7%
_TIME_DELTA_VERY_LONG_	$\Delta > 30\text{gg}$	233	0.3%
Totale delta calcolati		84 363	100%

Tabella 3.1: Distribuzione empirica dei delta temporali nel dataset e corrispondente token di discretizzazione assegnato. Il totale è inferiore agli 88.115 eventi grezzi poiché il primo evento di ciascuna traccia non possiede un evento precedente da cui calcolare il delta.

La funzione `categorize_time_delta_vectorized`, implementata con `numpy` per un processing completamente vettorializzato sull'intero dataset, traduce i delta continui nei token speciali secondo questa logica:

```
def categorize_time_delta_vectorized(deltas):
    conditions = [
        deltas == 0,                                    # Delta nullo: eventi simultanei
        (deltas > 0) & (deltas <= 3600),                # 0-1 ora:    Immediato
        (deltas > 3600) & (deltas <= 86400),            # 1h-1gg:     Breve
        (deltas > 86400) & (deltas <= 604800),          # 1-7gg:      Medio
        (deltas > 604800) & (deltas <= 2592000),        # 7-30gg:     Lungo
    ]
    choices = [
        '_TIME_DELTA_START_',         # delta = 0
        '_TIME_DELTA_IMMEDIATE_',     # 0 < delta <= 1h
        '_TIME_DELTA_SHORT_',         # 1h < delta <= 1gg
        '_TIME_DELTA_MEDIUM_',        # 1gg < delta <= 7gg
        '_TIME_DELTA_LONG_',          # 7gg < delta <= 30gg
    ]
    return np.select(conditions, choices, default='_TIME_DELTA_VERY_LONG_')
```

2. **Array concatenati (Azioni + Distanze numeriche):** Questa seconda forma di preprocessing costruisce l'embedding affiancando due array distinti: uno contenente esclusivamente la sequenza testuale delle azioni cliniche e un array parallelo numerico (*float*) esprimente la distanza in secondi dall'evento precedente. Un esempio:

```
{"case_id": "5870415",
 "tokens": ["Complete abdomen ultrasound", "Computerized tomography of the skull",
            "Complete abdomen ultrasound", "Computerized tomography of the skull"],
```

```
 "time_deltas": [0.0, 0.0, 728160.0, 178980.0],
 "label_id": 0
}
```


3. **Storytelling in linguaggio naturale:** L'approccio finale, che costituisce il core metodologico del progetto TEXLOS [1] e che è stato preferito ai precedenti, si rifà al pionieristico lavoro di traduzione semantica dei log del progetto LEGOLAS [4]. Anziché alimentare il classificatore con una combinazione mista di vettori numerici e codici, gli eventi e i vincoli cronologici logici della traccia sono tradotti letteralmente in frasi inglesi strutturate grammaticalmente, producendo interi paragrafi coerenti narranti il percorso diagnostico/terapeutico di ciascun paziente.


```
{"case_id": "5870415",
 "story": "The patient, 87 years old, underwent a series of examinations and
 interventions during hospitalization.
After 0 seconds, the following examinations were performed simultaneously:
Complete abdomen ultrasound and Computerized tomography of the skull.
After 728,160 seconds, Complete abdomen ultrasound was performed.
After 178,980 seconds, Computerized tomography of the skull was performed.",
"label_id": 0
}
```


3.1.3 Dettagli Implementativi dell'Algoritmo di Storytelling

Per colmare la distanza che intercorre tra i log estratti in formato grezzo e i testi discorsivi forniti in addestramento, è stata sviluppata un'infrastruttura di *parsing* ed iniezione in linguaggio Python. Dal punto di vista informatico, il processo si divide in due fasi computazionali: la strutturazione in memoria del file XES e la generazione cronologica della storia narrata.

Nella prima fase entra in gioco il modulo `xes_parser.py`. Affrontando l'ingente volume di eventi clinici asincroni tipici di un'analisi di Process Mining, l'algoritmo si destreggia tra la libreria per l'ingestione dei log (`pm4py`) e le più ottimali e leggere architetture dati di `pandas`. Quest'ultimo processa massivamente il dataset suddividendo mediante *groupby* e parallelismi nativi le righe afferenti ai medesimi pazienti; si procede poi convertendo in formato standard i *timestamp* (le date d'esecuzione dell'atto medico), le stringhe degli attori coinvolti ed instradando così ogni frammento cronologico entro uno strato informativo solido (oggetto *PatientTrace*).

La fase autoriale vera e propria è governata dalla classe `StoryGenerator` (all'interno dello script `story_generator.py`). Le specifiche di traduzione che intercettano queste *trace* vertono su tre stringenti requisiti ingegneristici:

- **Gestione del Tempo (Time Deltas):** L'evoluzione clinica viene misurata estraendo rigorosamente il *delta temporale in secondi* che separa l'azione attuale dalla precedente operazione registrata. Questo gap continuo viene iniettato nel template e calcolato esplicitamente mediante un blando *parsing* aritmetico (`current_time - prev_time`), oggettivando la scansione temporale. Nessuna approssimazione logica interviene nel testo finale; i secondi marcano senza preconcetti le tempistiche di risposta dei triage o la dilatazione cronica della degenza (si passa da "0 secondi" per l'esordio iniziale, a svariate centinaia di migliaia di secondi per tac e risonanze d'esito fatte la settimana d'uscita).

- **Gestione degli Eventi Simultanei:** Non è inusuale, nei processi su scala giornaliera densi come in ospedale, rilevare molteplici interazioni registrate col medesimo identico *timestamp*. Un esempio classico si verifica con batterie di check-up del sangue incodificate alla stessa ora informatica dal personale. Per evitare grave ridondanza lessicale che sovraccaricherebbe indebitamente l'asse d'attenzione di BERT con frasi ripetute, il generatore intercetta internamente gli aggregati (tramite la funzione `_group_simultaneous_events`), fondendoli all'interno di specifici *bullet points* o paragrafi discorsivi compatti nel seguente formato: *"After X seconds, the following examinations were performed simultaneously: Action A, and Action B"*.

- **Indipendenza dal Protocollo di Triage:** Il generatore proposto si differenzia strutturalmente dal progetto LEGOLAS, il quale era progettato per interpolare un insieme fisso di variabili fisiologiche standard da Pronto Soccorso (frequenza respiratoria, temperatura corporea, scala del dolore, ecc.), compilando un template narrativo predefinito. Nel contesto di TEXLOS, i pazienti attraversano reparti eterogenei con procedure cliniche non standardizzate, prive di una nomenclatura universale codificata. Per gestire questa variabilità, il generatore adotta dizionari interni flessibili e **regole di template matching dinamiche**, agnostiche rispetto alla tipologia di azione clinica, che mappano la tassonomia locale dell'ospedale ai costrutti discorsivi del Transformer senza vincolare l'algoritmo a un sottoinsieme predefinito di sensori o misurazioni.

A scopo esemplificativo, il seguente frammento di pseudo-codice descrive la routine adoperata dal software in `story_generator.py` per snellire testualmente (*trimming* sintattico) il sovraffolamento degli stadi simultanei ed iniettare i *time-delta* per una traccia degenziale prima dell'invio in NLP ad estrapolazione avvenuta:

```
def create_narrative_from_trace(trace_events):
    sorted_events = sort_by_timestamp(trace_events)
    grouped_events = group_simultaneous_events(sorted_events)

    narrative_paragraphs = []
    prev_time = grouped_events[0].timestamp

    for group in grouped_events:
        time_elapsed = calculate_seconds_diff(prev_time, group.timestamp)

        if len(group) == 1:
            # Singolo evento testuale formattato
            activity_desc = apply_medical_template(group[0].activity)
            text = f"After {time_elapsed} seconds, {activity_desc} was performed."
        else:
            # Piu' azioni collassate per alleggerimento dell'Attention
            activities = [apply_medical_template(e.activity) for e in group]
            concat_acts = join_with_commas_and(activities)
            text = f"After {time_elapsed} seconds, the following examinations " \
                   f"were performed simultaneously: {concat_acts}."

        narrative_paragraphs.append(text)
        prev_time = group.timestamp # Sposta l'ancora

    return "\n\n".join(narrative_paragraphs)
```

3.2 Architettura del Modello e Limitazioni dei BERT Clinici

Alla base del sistema di previsione giace `bert-base-uncased`, un Large Language Model reso disponibile pubblicamente sulla piattaforma HuggingFace. Questo Transformer, limitato alla componente Encoder, viene sottoposto a Fine Tuning aggiungendo al termine dell'architettura standard una testa di classificazione (*classification head*) volta a predire la natura binaria del problema: degenza normale (sotto i 20 giorni) contro degenza critica (uguale o superiore a 20 giorni).

Il primo istinto sperimentale è stato chiaramente quello di testare modelli *Domain-Specific*, confrontando le prestazioni di varianti pre-addestrate unicamente su corpus clinici – quali `biobert`, `pubmedBert`, `blue-bert` e `clinical-bert` – con quelle di `bert-base-uncased`. L'ipotesi sottostante era che un vocabolario medico radicato *ab initio* avrebbe accelerato la decodifica delle diagnosi. L'evidenza empirica ha sorprendentemente sancito prestazioni sovrapponibili tra i vari modelli e, anzi, una *Balanced Accuracy* leggermente inferiore per i modelli clinici rispetto al canonico modello generalista.

Tale anomalia si spiega analizzando l'altissima specificità terminologica del dataset della tesi: i modelli addestrati su dataset clinici (come MIMIC) sono spesso altamente specifici e ottimizzati per una formale nomenclatura standard inglese. Nel nostro caso, il nomenclatore locale dell'ospedale era ricco di acronimi, contrazioni proprietarie e formulazioni non standardizzate, vanificando almeno parzialmente il vantaggio della pregressa conoscenza medica dei pesi. Questo disallineamento è critico soprattutto a livello di *tokenizer*: i tokenizer dei vari modelli clinici, agendo su un vocabolario molto più verticale, frammentavano le parole spurie del dataset della tesi perdendo efficacia. Di contro, il tokenizer base di `bert-base-uncased` si è dimostrato molto più elastico, resiliente e capace di preservare sufficiente compattezza semantica pur al cospetto di un dominio estraneo.

3.3 Giustificazione Architetturale: Analisi del Trade-off Computazionale

In fase di definizione dell'architettura neurale, è stata condotta un'indagine esplorativa scalando il modello base verso la sua variante maggiorata `bert-large-uncased` (circa 340 milioni di parametri contro i 110 milioni originari). L'intento era tracciare la scalabilità dell'accuratezza su una rete più stratificata.

Un test preliminare su una singola esecuzione ha restituito metriche di rilievo sulla classe critica (LOS $\geq$ 20 giorni), registrando una Precision del 73.65% e una Recall dell'84.36%. Tuttavia, pur evidenziando un lieve incremento prestazionale grezzo rispetto alla versione base, l'esperimento ha fatto emergere severi limiti di scalabilità e di *data plateauing*. Da un lato, l'aumento dimensionale del modello non ha prodotto un salto di qualità proporzionale al costo computazionale, suggerendo che un bacino di circa 7.000 tracce cliniche non sia sufficiente a saturare la capacità di astrazione di un'architettura "Large", portando di fatto la rete in una fase di plateau informativo.

Dall'altro lato, i vincoli hardware hanno reso la sperimentazione ingegneristicamente insostenibile. La saturazione della VRAM ha imposto una drastica riduzione della *batch size*, precipitata dai 22 campioni gestibili dal modello base ad appena 4 campioni per il modello large. Questo crollo ha dilatato esponenzialmente i tempi di addestramento, rendendo materialmente inapplicabile la *Stratified 5-Fold Cross Validation*, requisito im-

prescindibile delineato in questa tesi per garantire la robustezza statistica del classificatore dinanzi a uno sbilanciamento di classi così severo.

Di conseguenza, il modello `bert-base-uncased` è stato eletto come "ottimo paretiano" del progetto, offrendo il compromesso ideale tra capacità predittiva, sostenibilità hardware e garanzia di validazione metodologica rigorosa.

3.4 Focal Loss e Sbilanciamento delle Classi

Il dataset presenta uno dei paradossi strutturali più frequenti della Process Analytics in sanità: lo *sbilanciamento estremo delle classi*. Di fatto, le casistiche in cui il paziente sfiora o sfora una degenza continua di venti giorni (classe 1 - Critico) costituiscono una netta e fisiologica minoranza anomala nei log degli eventi rispetto alla travolgente maggioranza dei passaggi di routine o dimissioni rapide (classe 0 - Normale).

Una prima fase di addestramento gestita dalla standard *Binary Cross Entropy* (BCE) ha confermato l'incapacità del modello di apprendere in modo bilanciato: il classificatore tendeva a privilegiare la classe dominante, producendo di fatto una predizione prossima al *ZeroR* e penalizzando la sensibilità sui casi critici — esattamente quelli di primario interesse per le amministrazioni ospedaliere.

Per correggere questo bias strutturale e ricalibrare il contributo al gradiente in modo differenziale, è stata adottata in fase di fine-tuning la funzione di costo denominata **Focal Loss** [6]. Questa funzione mutua la sua origine nel dominio della *Computer Vision*, dove nacque per correggere il massiccio sbilanciamento tra uno sconfinato spazio di *background* e piccoli oggetti da identificare. Applicata al processing di tracce cliniche, la Focal Loss riduce il peso computazionale (*loss contribution*) degli esempi facili e sovrabbondanti, in favore dei campioni mal classificati.

3.4.1 Formulazione Matematica della Focal Loss

Il punto di partenza è la funzione di costo *Cross Entropy* (CE) per la classificazione binaria. Definito $y \in \{\pm 1\}$ come la classe ground-truth e $p \in [0, 1]$ come la probabilità stimata dal modello per la classe $y = 1$, la formulazione classica è:

$$CE(p, y) = \begin{cases} -\log(p) & \text{se } y = 1 \\ -\log(1-p) & \text{altrimenti.} \end{cases} \tag{3.1}$$

Per convenienza notazionale, definendo p_t:

$$p_t = \begin{cases} p & \text{se } y = 1 \\ 1-p & \text{altrimenti,} \end{cases} \tag{3.2}$$

la CE si può riscrivere compattamente come $CE(p_t) = -\log(p_t)$.

Un metodo comune per mitigare lo sbilanciamento puro introduce un fattore di peso $\alpha \in [0, 1]$ per la classe 1 e $1 - \alpha$ per la classe -1. Definendo α_t in modo analogo a p_t, si ottiene la *α-balanced Cross Entropy*:

$$CE(p_t) = -\alpha_t \log(p_t) \tag{3.3}$$

Tuttavia, pur bilanciando l'importanza numerica tra positivi e negativi, questa formulazione non differenzia tra esempi *facili* (*easy negatives*) ed esempi *difficili* (*hard examples*). Per abbattere il peso sproporzionato degli innumerevoli esempi facili (i pazienti con degenze standard facilmente prevedibili), Lin et al. introducono un fattore modulante $(1 - p_t)^\gamma$, dove $\gamma \geq 0$ è un iperparametro noto come *parametro di focalizzazione* (*focusing parameter*). La formulazione completa della **Focal Loss** (nella sua variante α-balanced pratica) diviene quindi:

$$FL(p_t) = -\alpha_t (1 - p_t)^\gamma \log(p_t) \tag{3.4}$$

Questa funzione gode di proprietà analitiche fondamentali per il nostro task:

- Se un esempio ospedaliero è mal classificato o raro (quindi p_t è piccolo), il fattore modulante $(1 - p_t)^\gamma$ tende a 1 e la *loss contribution* rimane inalterata, garantendo l'apprendimento.
- Se un esempio è facile e ben classificato ($p_t \to 1$), il fattore modulante tende a 0, penalizzando (*down-weighting*) fortemente l'impatto sul gradiente globale.
- Il parametro γ regola il tasso di questa penalizzazione (addolcendo l'iper-dominanza della classe 0 nel nostro dataset), esonerando l'analista dalla necessità di applicare euristiche forzate e distruttive a valle, come un aggressivo *undersampling* che distruggerebbe preziosi pattern temporali globali.

L'effetto della parametrizzazione γ si evince analizzando visivamente l'andamento della funzione per vari scenari, confrontando la Focal Loss con la Cross Entropy standard (Figura 3.4).

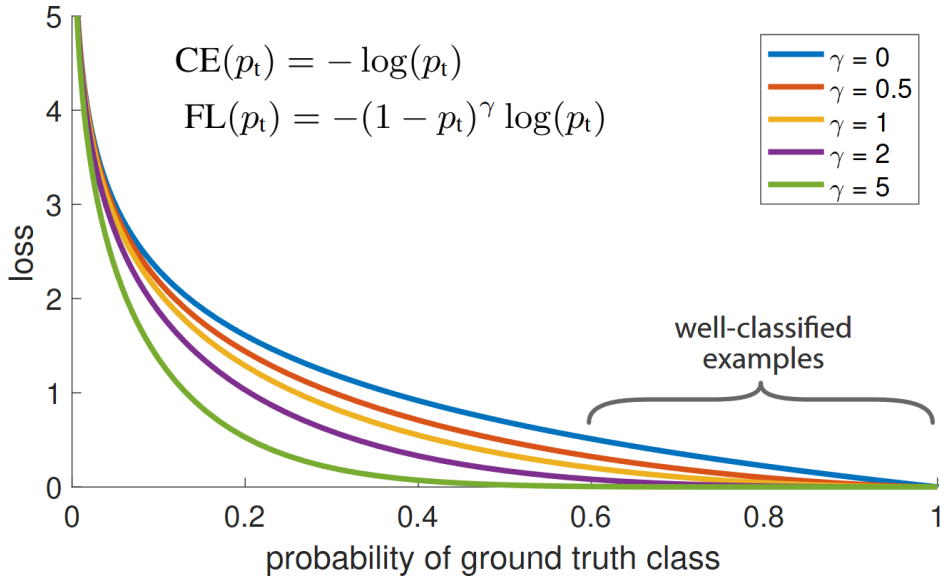

Figura 3.4: Confronto tra la Cross Entropy (CE) e diverse configurazioni della Focal Loss. Sull'asse delle ascisse è riportata la probabilità effettiva stimata dal modello, $p_t \in [0,1]$, mentre sull'asse delle ordinate figura il valore della funzione di costo. Si noti come, al crescere dell'iperparametro di focalizzazione γ (es. $\gamma = 2,5$), la loss addebitata agli esempi facili e ben classificati ($p_t \to 1$) subisca un drastico schiacciamento rispetto al decadimento più lento della CE. Questo espediente permette all'osservazione di focalizzarsi dinamicamente sugli esempi più critici ed errati.

Una volta garantita la capacità discriminatoria del modello su classi sbilanciate, si procede all'analisi della sua interpretabilità tramite le tecniche descritte nel capitolo successivo.

3.5 Setup Sperimentale e Hyperparameter Tuning

L'intera pipeline di esperimenti e addestramento supervisionato è stata materialmente elaborata e portata a convergenza su un setup hardware performante, incentrato su una workstation munita di CPU AMD Ryzen 9 5900X, 64 GB di RAM e GPU NVIDIA GeForce RTX 4090 (consentendo l'aumento efficiente delle batch size e l'addestramento distribuito del grafo tensoriale Transformer).

Per preservare l'integrità e il realismo della diagnostica probabilistica al cospetto dello sbilanciamento delle classi, la valutazione accademica si basa unicamente su iterazioni cicliche di *Stratified K-Fold Cross Validation*. Questo garantisce che ogni singola split prodotta preservi identiche proporzioni di pazienti anomali/ordinari rispetto al database globale, smaltendo bias d'addestramento.

L'impostazione degli iperparametri del modello neurale è sfociata da un processo combinato:

- L'esplorazione automatica bayesiana condotta massivamente dal framework `Optuna`.

- Un minuzioso *tuning manuale* a posteriori. Affinché i risultati non scivolassero in asimmetrie dimensionali non volute, le alterazioni manuali degli intervalli innescati da Optuna si sono preoccupate di preservare costanti le "proporzioni architetturali" auree del transformer, quali i rapporti intrinseci tra hidden layers, attention heads e scaling rate dello stack.

3.5.1 Convergenza del Modello e Early Stopping

A seguito del setup definito, l'addestramento della configurazione finale `bert-base-uncased` ha evidenziato un comportamento stabile. Come illustrato nella Figura 3.5, l'andamento mostra una discesa costante della *training loss*, mentre la *validation loss* raggiunge il suo minimo all'epoca 10. Nelle epoche successive la curva di validazione inizia a divergere, indicando l'innesco dell'*overfitting*. Grazie alla tecnica dell'*early stopping* (impostata con un Δ minimo di 1×10^{-3} su una *patience* di 5 epoche), il training si è interrotto all'epoca 15; i pesi del modello sono stati quindi ripristinati all'ottimo dell'epoca 10, garantendo la massima capacità di generalizzazione sui dati non visti per la successiva fase di test.

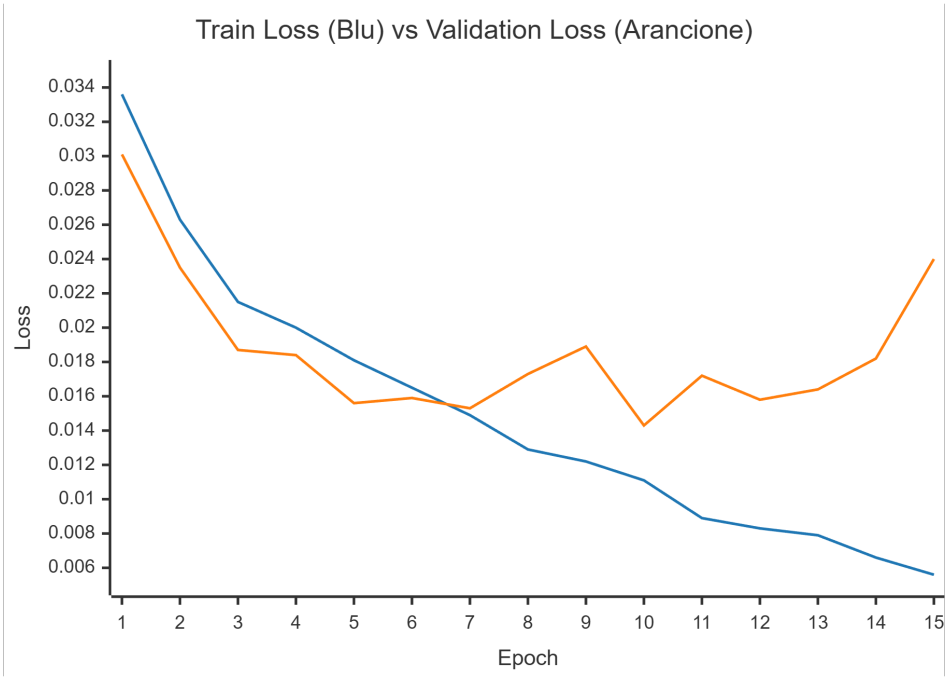

Figura 3.5: Curva di apprendimento (Training vs Validation Loss) del modello definitivo durante il processo di fine-tuning.

3.6 Indagine Esplorativa: Data Augmentation con LLM

Nel tentativo di robustire l'astrazione di BERT sulla classe minoritaria (long-LOS), si è tentato un originale esperimento di estensione del dataset (*Data Augmentation*) avvalendosi dell'intelligenza generativa degli attuali LLM generativi (decoder-only).

La pipeline esplorativa prevedeva l'impiego locale di modelli generativi allo stato dell'arte, nello specifico `phi4:14b` e `nous-hermes2:34b`, eseguiti tramite l'ambiente open-source `Ollama`. Per inibire le allucinazioni e circoscrivere l'effusione creativa dell'esposizione testuale, la *Temperature* è stata fissata al minimo rigido ($T = 0.0$).

Il layer di codice Python istruiva severamente le reti a limitarsi ad un mero lavoro di riformattazione narrativa, passando iterativamente le coordinate base della traccia log:

```
prompt = f"""You are an expert physician who writes discursive clinical
narratives in English for BERT/NLP analysis.

IMPORTANT - MANDATORY MEDICAL TERMINOLOGY:
You must use EXACTLY the medical terminology specified in the clinical vocabulary.

PATIENT DATA:
- Information: {patient_desc}

MEDICAL EVENTS SEQUENCE (already standardized):
{events_text}

SPECIFIC RULES:
1. USE EXACTLY the provided activity names (already standardized)
2. Introduction with age and medical context
3. Separate paragraph for each temporal group
4. Format numbers: "13,380 seconds" (with commas for thousands)
5. Concurrent events: "the following examinations were performed simultaneously:"
6. Connectors: "The [activity] was performed", "After X seconds [activity] was executed"
7. Paragraphs separated by blank line
8. Keep medical terminology IDENTICAL to provided terminology
9. Always conclude with "At the end of the clinical pathway, the medical events were concluded."
10. NO translations or paraphrases of medical activity names
11. All text must be in ENGLISH"""
```

Nonostante l'imbragatura direzionale e sintattica imposta a livello di framework, l'ingestione e la rielaborazione da parte delle reti ha portato a delle devianze gravose. A titolo di dimostrazione, viene qui incollata la narrativa sintetica prodotta dal modello `phi4:14b` a fronte del caso ospedaliero ID "5870415" (contenente originariamente 4 atti nosocomiali primari ed oggettivi estratti dai registri):


```
{
  "case_id": "5870415",
  "story_text": "The patient, 87 years old, underwent a series of examinations...
The Complete abdomen ultrasound was performed as the initial examination to
assess abdominal structures for any abnormalities or pathologies that might
require further investigation... This non-invasive imaging technique provided
valuable insights... After 0 seconds, the following examinations were performed
```

```
simultaneously: Computerized tomography of the skull and Complete abdomen ultrasound. The Coputerized tomography of
the skull was conducted to evaluate any potential intracranial issues such as
hemorrhages, tumors, or other abnormalities... After 728,160 seconds
(approximately 8 days and 7 hours), another Complete abdomen ultrasound was
executed. This follow-up examination aimed to monitor any changes...
At the end of the clinical pathway, the medical events were concluded."
}
```


L'esempio documenta i limiti intrinseci nella generazione del testo impiegata per questo scopo predittivo. Nonostante le rigide indicazioni tecniche nel prompt testuale, il modello `phi4:14b` ha generato un'ampia **allucinazione semantica**, arricchendo indebitamente l'output testuale: il LLM ha aggiunto descrizioni procedurali e teoriche arbitrarie sul motivo degli esami (*"to evaluate any potential intracranial issues such as hemorrhages"*), che risultano rumore puro ai fini di classificazione ospedaliera.

Test successivi svolti con un LLM dotato di un maggior numero di parametri (`nous-hermes2:34b`) hanno prodotto storie più sintetiche e rispettose delle regole sintattiche del prompt, immettendo tuttavia alterazioni concettuali sottili e altrettanto problematiche:


```
{
  "case_id": "5870415",
  "story_text": "The patient, an 87-year-old individual, underwent a series of
examinations... The first event was a complete abdomen ultrasound...
Simultaneously, a Computerized tomography scan of the skull without contrast medium was executed.
After 728160 seconds, another complete abdomen ultrasound was conducted.
Following this, a Computerized tomography scan of the skull with contrast medium was carried out..."
}
```


In questo secondo test, il modello linguistico ha dedotto arbitrariamente dettagli non presenti nel database clinico, specificando la presenza o l'assenza del contrasto in una diagnostica di routine (*"with contrast medium / without contrast medium"*). Benché semanticamente pertinenti in un caso cardio-vascolare, queste integrazioni scostano la traccia artificiale dall'estrazione originale, alterando il vocabolario disponibile a valle e compromettendo la confrontabilità con le tracce reali.

La convalida empirica e i successivi processi di *fine-tuning* incrociato hanno sancito il fallimento dell'ipotesi predetta. L'avvalersi dello *Storytelling* sintetico arricchito da tali artefatti informativi ha comportato un degrado strutturale delle metriche computazionali. La rumorosità lessicale iniettata dai Large Language Models confonde l'attenzionalità temporale del Transformer, peggiorandone la capacità selettiva e compromettendo di fatto i valori di *Balanced Accuracy*. Per via di queste problematiche, l'adozione testuale definitiva in questa tesi usa esclusivamente script di parsing lineari (`story_generator.py`), assicurando una perfetta aderenza matematica ed enumerabile al tracciato loggato originario.

4. Risultati

4.1 Metriche di Classificazione

La validazione del modello predittivo `bert-base-uncased`, architettato sull'embedding a *Storytelling* per la classificazione dicotomica sul Length of Stay lungo, ha prodotto le seguenti metriche di classificazione.

Al termine del processo di hyperparameter tuning e addestramento supervisionato, la metrica chiave di valutazione, la **Balanced Accuracy**, si è assestata su valori di circa l'**88%**. Questo risultato assume particolare rilievo considerando il gravoso sbilanciamento delle classi in input: la Balanced Accuracy, di fatto media aritmetica delle accuratezze (Sensibilità e Specificità) sulle due classi in esame (Casi normali vs Casi patologici), testimonia che il classificatore ha evitato di **collassare sulla classe maggioritaria** (la degenza breve). Un ruolo critico nel conseguimento di quest'alta soglia sensibile è ascrivibile all'adozione differenziale e decisa della Focal Loss, che dinamicamente forzava il focus iterativo di addestramento primariamente sulle tracce rare a LOS anomalo, contribuendo a limitare i falsi negativi e mitigando il bias di predizione introdotto dagli esemplari maggioritari.

4.2 Confronto Prestazionale degli Embedding

Prima di eleggere lo *Storytelling* come soluzione definitiva, il modello è stato regolarmente addestrato e testato sulle altre due formulazioni di embedding discusse (Coppie Azione-Token Temporale e Array Concatenati). La Tabella 4.1 riassume il confronto sulle principali metriche di classificazione binaria. Lo Storytelling ha garantito un balzo in avanti, specialmente nella metrica chiave di *Balanced Accuracy*.

Embedding	**Accuracy**	**Balanced Accuracy**	**F1-Score**	**Precision**	**Recall**	**Specificity**
Azioni + Binning	91.34%	87.16%	67.35%	57.23%	81.82%	92.51%
Array Concatenati	87.29%	83.44%	57.40%	45.24%	78.51%	88.36%
Storytelling	92.61%	89.69%	71.72%	61.54%	85.95%	93.42%

Tabella 4.1: Confronto delle prestazioni di classificazione per le tre tecniche di embedding testate.

Per facilitare l'analisi visiva delle performance, la Figura 4.1 espone graficamente i differenziali riportati in tabella.

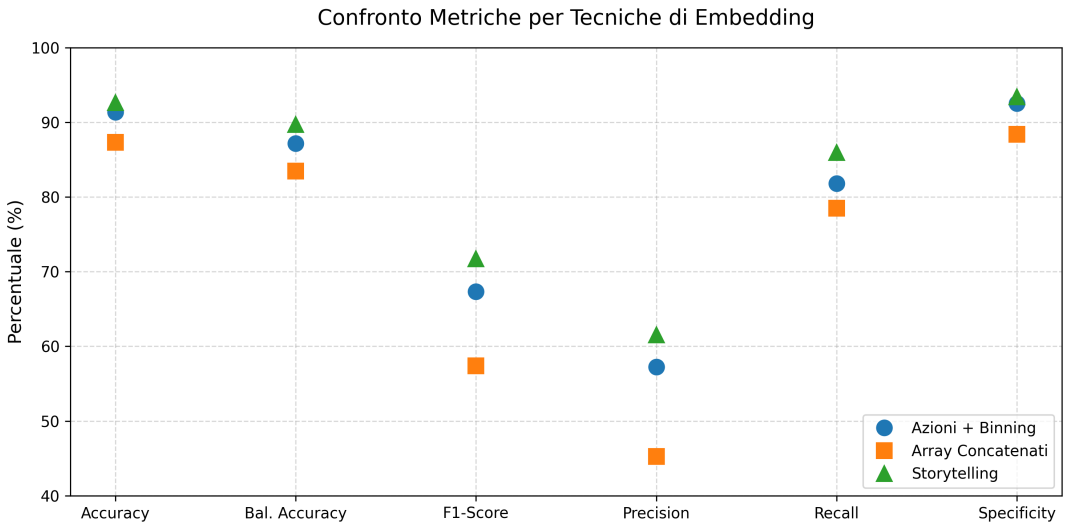

Figura 4.1: Rappresentazione grafica a dispersione delle metriche di classificazione a confronto per i tre metodi di embedding testati: Storytelling, Azioni + Binning e Array Concatenati. Il grafico illustra visivamente i risultati tabellari, evidenziando il vantaggio prestazionale garantito dall'approccio narrativo su tutte le metriche chiave, in particolar modo sulla Balanced Accuracy e sull'F1-Score.

4.2.1 Analisi Critica: Il Trade-Off tra Precision e Recall

Nell'ottica clinica applicata, le evidenze restituite dalla tabella non devono essere valutate unicamente sotto la lente della *Balanced Accuracy*, seppur eccellente. Il comportamento profondo del modello è nitidamente leggibile scomponendo il risultato nelle metriche di **Precision (61.54%)** e **Recall (85.95%)** ottenute con la variante *Storytelling*.

Il parametro Recall (Sensibilità) si dimostra volutamente prominente: la rete è stata addestrata e calibrata – anche mediante l'impiego della Focal Loss – per captare con elevata affidabilità i casi positivi, ovvero l'insorgenza di anomalie logistiche (pazienti con lunghissimo Length of Stay). Questo si traduce in una ridottissima generazione di *falsi negativi*. Nel contesto direzionale di una struttura nosocomiale, un falso negativo costituisce un danno economico e gestionale severo, in quanto concorre ad una categorizzazione ottimistica di un iter che sfocerà invece in percorsi laboriosi, saturando letti d'appoggio imprevisti, complicazioni inaspettate o dimissioni abortite.

Di contro, la *Precision* al 61.54% implica un margine intrinseco di **falsi positivi**. Dal punto di vista pratico, ciò comporta che una porzione decrescente di degenti classificati come "ad alto rischio di LOS prolungato" dall'intelligenza artificiale concluderà in realtà il proprio percorso nei tempi previsti. In questo specifico dominio di indagine, il *trade-off* è non solo ampiamente giustificato, ma ricercato proattivamente: il costo etico ed

amministrativo legato all'attivazione di monitoraggi supplementari e allocation preventiva su un "falso allarme" è ordini di grandezza inferiore rispetto al dramma di trascurare e smarrire analiticamente un paziente esposto ad altissime complessità di cura. Questo bilanciamento matematico certifica quindi non solo la robustezza formale di BERT, ma la reale fattibilità della sua adozione quale *Decision Support System* proattivo per le direzioni sanitarie, massimizzando strutturalmente la tutela clinica al prezzo di parziali ma tollerabili sovrastime logistiche d'allerta.

A ulteriore riprova delle differenze di convergenza, si riportano le **Matrici di Confusione** relative ai test set di ciascun approccio. La capacità dello Storytelling di limitare i falsi negativi patologici giustifica l'investimento computazionale del fine-tuning linguistico.

Array Concatenati

		Predetto	
		Normale	Critico
Reale	Normale	873	115
	Critico	26	95

Binning

		Predetto	
		Normale	Critico
Reale	Normale	914	74
	Critico	22	99

Storytelling

		Predetto	
		Normale	Critico
Reale	Normale	923	65
	Critico	17	104

Tabella 4.2: Matrici di confusione confrontate sui tre approcci di embedding.

4.3 Explainable AI: L'Adattamento degli Integrated Gradients

Mentre le metriche computazionali provano la **capacità predittiva** del modello generico per la classificazione dei log ospedalieri, l'aspetto metodologicamente distintivo del progetto risiede nella trasparenza offerta dalle interpretazioni di classe post-hoc.

4.3.1 Giustificazione del Metodo e Gradient Shattering

Per interpretare le logiche decisionali di BERT, lo studio ha adottato il rigore degli **Integrated Gradients (IG)**. La scelta di questo algoritmo interpretativo primario, scartando alternative euristiche storiche diffuse in letteratura, è dettata da stringenti limiti matematici e di dominio. Tecniche come Grad-CAM risultano intrinsecamente legate alle features map spaziali delle Reti Convoluzionali (CNN) e si adattano con grandi limiti deduttivi alla rigidità vettoriale dei meccanismi asincroni di *Self-Attention* alla base del transformer testuale. Analogamente, approcci appoggiati all'estrazione di gradienti standard (Vanilla Gradient Descent) o fondati su perturbazioni con vettori di iniezione a rumore continuo (SmoothGrad, Blur IG), pur confermandosi campioni d'analisi su *Computer Vision*, non si interfacciano coerentemente alla natura discreta imposta dal *Word Embedding* di un modello linguistico. A riguardo, un confronto visuale astratto sulle metriche (Figura

4.2) rivela plasticamente il disturbo di calcolo associato a pattern basici che verrebbe inevitabilmente trasferito sulla parola di testo clinica.

Un ulteriore limite del calcolo iterativo classico della retropropagazione standard è il **Gradient Shattering** (frantumazione del gradiente). Nelle reti neurali profonde, i gradienti calcolati localmente attraverso funzioni di attivazione non lineari (come la *GELU*) possono annullarsi o divergere in punti specifici della rete, producendo attribuzioni localmente incoerenti e prive di significato globale [5]. Valutando invece l'integrale definito lungo l'intero percorso dalla baseline all'input reale, gli Integrated Gradients mediano queste oscillazioni, restituendo per ogni sub-token un'attribuzione netta che riflette il contributo effettivo all'output di classificazione finale.

Figura 4.2: Confronto tra metodi di Feature Attribution XAI applicati all'individuazione di un input "cane". I metodi Vanilla Gradient e Guided BackProp producono maschere ad alto rumore geometrico, prive di valore interpretativo. Per evitare lo stesso problema sul corpus clinico testuale, si è adottato il calcolo integrale degli Integrated Gradients.

4.3.2 Approssimazione Discreta e Completeness

Dal punto di vista implementativo, il calcolo continuo dell'integrale degli IG non può essere risolto in forma chiusa per reti neurali profonde. L'integrale viene valutato in modo discreto tramite una somma di Riemann con m step di interpolazione. La formula approssimata, adottata come standard nelle librerie come *Captum*, è la seguente:

$$IntegratedGrads_i^{approx}(x) ::= (x_i - x'_i) \times \frac{1}{m} \sum_{k=1}^{m} \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \tag{4.1}$$

Nell'adattamento matematico di tale equazione per sequenze linguistiche discrete, il termine x' detiene il ruolo nevralgico della cosiddetta **Baseline**. Tale parametro funge da punto d'origine astratto, ideato per simulare al classificatore *BERT* una condizione di totale assenza di informazione testuale. Poiché i token di testo non possono essere azzerati matematicamente con un semplice scalare nullo, si è resa necessaria una preparazione tensoriale algoritmica. La baseline è stata creata replicando il *Padding Token* (`[PAD]`, identificativo $ID = 0$) fino a eguagliare la lunghezza della traccia elaborata (l'input reale x). Questo vettore viene propagato in *feedforward* lungo il layer di `word_embeddings` del modello, ottenendo la sua rappresentazione nello spazio latente numerico. Fissare la baseline in questo spazio continuo garantisce dunque all'algoritmo di interpolazione uno "zero semantico" affidabile da cui partire per approssimare l'ingresso testuale.

Per valutare matematicamente l'affidabilità delle importanze estratte e assicurarsi che l'approssimazione discreta dell'integrale sia adeguata, la spiegazione sfrutta l'assioma essenziale della **Completeness** (Completezza). Esso stabilisce che la somma algebrica dei gradienti attribuiti a tutti i singoli token debba eguagliare fedelmente la differenza tra l'output predetto dal modello per l'input in esame $f(x)$ e l'output per la *baseline* isolata $f(x')$. Per svincolare la discrepanza dalla magnitudo della predizione originaria del neurone finale, tale accuratezza calcola un vincolo dimensionale quantificato dall'*Errore Relativo*:

$$\text{Rel\_Error} = \frac{|f(x) - f(x') - \sum \text{Attributions}|}{|f(x) - f(x')|} \tag{4.2}$$

Quando questo errore cala sotto una rigidissima tolleranza del 5% (Rel_Error $\leq$ 0.05), si ritiene di aver raggiunto la chiusura integrativa e pertanto definita la formale **convergenza** dell'algoritmo.

Per forzare la convergenza, la programmazione ha previsto uno script Python contente una subroutine (*algoritmo adattivo*) che raddoppia l'indice incrementale della base approssimatrice degli iper-parametri di calcolo. Partendo di prassi con una configurazione base di test ($m = 50$), il modulo controlla che lo scarto esibito rientri nelle quote minime. Se l'errore travalica il limite, la diagnostica ripiega sul raddoppio dei passaggi (*steps doubling*) ripetendo totalmente la funzione. Su queste dinamiche protocollari la maggioranza fisiologica (90%) delle derivate giunge a convergenza definitiva a un passo iterativo di precisione quantificati statisticamente a **1500 steps**. Di contro, tracce anamnestiche insolitamente vaste spingono questo automatismo in escalation arrivando a esigere persino frazioni minime per bloccare l'errore raggiungendo perciò massimali esecutivi di **5500 steps**.

Questo elevato valore iterativo richiesto per convergere palesa la grande differenza computazionale tra NLP e gli algoritmi di Explainability in Computer Vision, i quali si risolvono spesso con appena 50 o 100 *steps*. In campo visivo l'interpolazione transita lungo lo spettro dei pixel da aree vuote per sfumature matematiche coese; al contrario, il *Word Embedding* archiviato nel *transformer* è una griglia matematica astratta a 768 dimensioni molto ripida e disgiunta tra occorrenze semantiche ben polarizzate limitando severamente il passo di scarto ammissibile senza causare deformità volumetriche all'integrale effettivo. L'alta frizione derivativa esige pertanto l'enorme fittezza numerica del campionamento ($m > 1500$) per superare l'assioma.

4.3.3 Bottleneck Computazionale

L'alto numero di *step* introduce un severo sovraccarico prestazionale legato al disallineamento operativo di memoria tra CPU e GPU. La classe `IntegratedGradients` della libreria Captum processa il campionamento adottando per default la quadratura standard di **Gauss-Legendre**; una soluzione puramente algebrica per definire i vertici spaziali non equidistanziati che vincola però 'numpy' a generare sulla CPU matrici dense di ampiezza $m \times m$ valutandone gli autovalori iterativamente. Allo standard clinico di **1500 steps**, questa decomposizione frenava l'infrastruttura sprecando fino a 15 secondi per passaggio disattivando contestualmente l'offload grafico della GPU RTX 4090 durante lo svuotamento. A soglie critiche da **5500 steps** provocava puntualmente il crash strutturale matematico infrangendo il *timeout*.

A livello codice si è pertanto ovviato imponendo forzatamente il calcolo *standard* d'ispezione alla formula di **Riemann Trapezoid** ('method='riemann_trapezoid'). Tale implementazione numerica modella i segmenti con equispazialità costante: evitando del tutto lo studio logico matriciale e riversando costanti lineari vettorializzando al cento per cento il carico sulla GPU (*pure-GPU arithmetic*). Ai regimi iper-densi testati l'approssimazione differenziale di quest'ottimizzazione rispetto al gaussiano crolla sotto una tolleranza irrisoria dello 0.1%; fornendo insomma garanzie formali intatte risolvendo però criticamente i limiti hardware **riportando i tempi d'estrazione clinica a circa 2.5 secondi a singolo sample**.

4.4 Ricomposizione dei Token e Analisi dell'Impatto Clinico

4.4.1 Il Limite Euristico della Tokenizzazione Sub-Word

L'applicazione diretta degli Integrated Gradients su uno *Storytelling* elaborato tramite BERT manifesta tuttavia un profondo limite euristico per un decisore medico umano.

BERT sfrutta un tokenizer sub-word (tipicamente il *WordPiece*), spezzando semanticamente le parole rare, gli acronimi nosocomiali ed i tag composti generati in fase di data embedding in atomi non autosufficienti concettualmente (sub-token). L'algoritmo IG standard attribuirebbe un punteggio di importanza a singoli sub-token privi di significato semantico autonomo (es. frammenti di acronimi o codici di procedura), rendendo complessa l'interpretazione clinica e vanificando per il management il pregio diagnostico ed ottimizzativo del framework.

4.4.2 Strategia di Aggregazione (Aggregation Strategy)

La presente proposta risolve radicalmente l'impasse introducendo il meccanismo cruciale della **Ricomposizione Semantica Post-Calcolo** dei token frammentati, che governa la conversione dello score grezzo dell'attribuzione matematica in metrica clinica operativa.

Al fine di intercettare analiticamente gli atti medici da ricostruire, il modulo (*Clinical Action Aggregator*) importa il dizionario in uso (`translation_cache.json`). Cosciente che i record clinici affondano radici nella stesura infermieristica italiana ma giungono formattati per la somministrazione al modello in lingua inglese, la procedura Python usa il dizionario incrociandone i valori: la stringa in uscita anglofona è individuata e aggregata alla sua naturale controparte italiana per l'allocazione al clinico del valore esatto dell'importanza vettoriale isolata. Tale macro-fase è pilotata in RAM seguendo l'accorpamento additivo diviso in due logiche consequenziali:

1. **Step A: Da Embedding Dimension a Token Score.** Allo stato primitivo di estrazione back-propagativa post-tuning, l'algoritmo IG classico invocato da Captum non restituisce un singolo valore numerico per ciascun tag alfabetico decodificato, ma diffonde ed espone un valore di calcolo derivato per ogni singola coordinata del vettore denso di codifica. Nell'architettura specifica da noi in analisi (`bert-base-uncased`), questa scomposizione si declina in ben 768 dimensioni continue distinte per ogni frammento di sillaba. Poiché l'intento applicativo è valutarne l'ingombro processuale finale (la predizione del Length of Stay), le 768 *features score* occulte devono imperativamente subire una compressione quantitativa: il processo effettua vettorialmente una macro-sommatoria algoritmica che attraversa e somma orizzontalmente i valori distribuiti nell'asse tensoriale di profondità (matematicamente, il sistema effettua un'operazione deduttiva di *reduction* imponendo `sum(dim=-1)` sulla libreria estesa di estrazione). Tramite lo Step A, questa astrazione collassa restituendo un unico scalare oggettivo riflettente l'impatto integrale olistico adoperato dal modello attenzionale per ogni unità.

2. **Step B: Da Sub-word frammentato a Parola Clinica Intera.** Una volta isolati gli score scalari per ciascun frammento, è necessario ricomporre la terminologia

clinica originale sommando i contributi dei sub-token correlati. Il tokenizer *WordPiece* di BERT segnala i frammenti secondari con il prefisso `##`: lo script individua queste sequenze e somma algebricamente i loro punteggi al token radice precedente (*root sum*), ricostruendo così la parola clinica intera con un unico score aggregato. Come passo finale, i token artificiali `[CLS]` e `[SEP]` vengono esclusi dalla raccolta degli score: pur essendo funzionali al fine-tuning della rete, non corrispondono ad alcuna azione clinica reale e la loro inclusione nel report diagnostico introdurrebbe rumore privo di valore interpretativo.

Conducendo quest'azione meticolosa su un vocabolario testuale raggruppabile attorno ai parametri cardine documentabili dell'ospedale (esami, visite, trattamenti, ecc.), si assegna rigorosamente un punteggio univoco ed aggregato dinamicamente allo specifico atto medico effettivo intra-processuale reale in maniera coerente.

4.4.3 Risoluzione End-to-End: Il caso "Cardiology visit"

A titolo dimostrativo e per cementificare *end-to-end* i costrutti discussi, si consideri la mappatura dell'evento di log processato e testato: **"Cardiology visit"**. Essendo la traduzione semantica delle degenze effettuata in lingua inglese per assecondare la sintassi per cui è stato istruito `bert-base-uncased`, la ricostruzione *post-hoc* risulta fondamentale e modella testualmente le 6 sotto-fasi matematiche IG presentate poc'anzi:

1. **Tokenizzazione e Baseline:** Il testo d'ingresso viene prelazionato dal Layer e sfaldato dal *WordPiece* tokenizer in `[CLS, card, ##iology, visit, SEP]`. Per generare l'esca matematica su cui far orbitare la derivata parziale rispetto agli input formattati di riga, viene allocato il *baseline tensor* costituito da un vettore di pari dimensionalità ma saturato esclusivamente dal fattore nullo Padding (ID 0): `[PAD, PAD, PAD, PAD, PAD]`.

2. **Interpolazione a Step Finita:** Per congiungere la baseline fittizia all'input paziente reale in esame, l'integrale approssimato crea $m = 50$ frazioni spaziali equidistanti. In ciascuno step discreto α_i (che corre tra 0.00 e la prossimità asintotica 1.00) avviene la fusione interpolata e pesata nel continuo per l'indagine algebrica: $\text{Baseline}_{emb} + \alpha_i \times (\text{Input}_{emb} - \text{Baseline}_{emb})$.

3. **Attribuzione Sparsa Ponderata:** Su ciascuno dei 50 steps spaziotemporali è valutato iterativamente il *gradiente* attenzionale di logica contro il nodo predittivo finale del modello. Terminate le fasi di inferenza passiva e raccolta retroattiva, l'integrazione numerica distribuisce come ritorno un tensore matematicamente irragionevole per il singolo uomo: ogni sub-token testuale processato (tra cui `card` e `##iology`) espone difatti sulla griglia ben 768 valori *float* nativi.

4. **Aggregazione Step A (Collasso Logit):** L'azione coesiva interviene tramite la macro-riduzione $\Sigma_{dim=-1}$, annullando algebricamente l'orizzonte tensoriale isolando l'intensità direzionale netta. Supponendo, a titolo d'esempio esplicativo, di quantificare un attrito generato di +12.45 in capo a `card`, una sfumatura concettuale d'apporto di +7.83 isolata sul suffisso `##iology` e uno spiccato interesse computazionale quantificabile a +18.92 ad uso stringa `visit`. Viene qui forzata anche l'identificazione selettiva in esubero: i sub-token `CLS` e `SEP` vengono destituiti dalla pipeline a questo esatto traguardo.

5. **Aggregazione Step B (Cucitura Lessicale):** Lo script scansiona la sequenza e individua i sub-token marcati con `##`: il punteggio di `##iology` (+7.83) viene sommato a quello del token radice `card` (+12.45), producendo il termine clinico ricomposto **"Cardiology"** con score aggregato $12.45 + 7.83 = 20.28$. Il token `visit`, non essendo un frammento secondario, mantiene il proprio punteggio invariato (+18.92).

Quest'azione algebrica mirata restituisce un set di **macro-scores unici per atto medico coerente** (Figura 4.3), che l'architettura espone interamente decodificato e quantificato, superando le mutilazioni linguistiche del Transformer e favorendone lo sviluppo ed ispezione manageriale per le direzioni sanitarie attive.

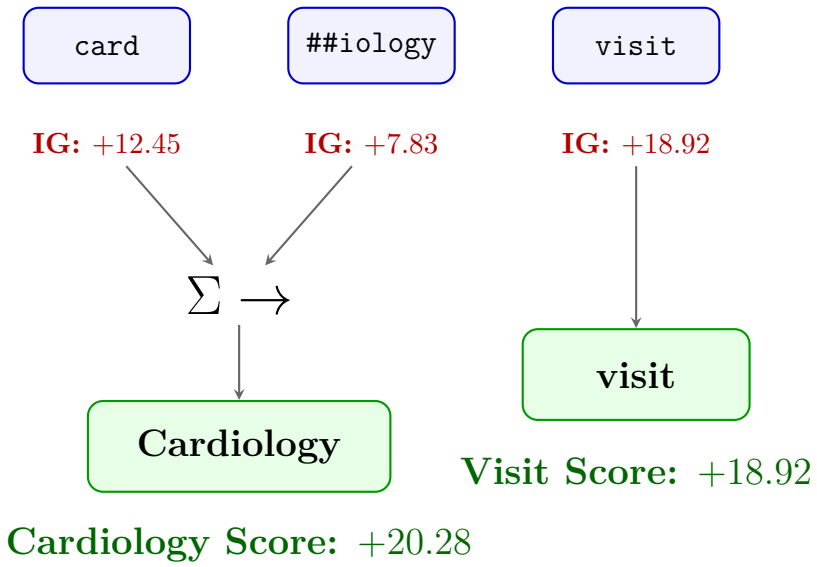

Figura 4.3: Ricomposizione post-calcolo della refertazione "Cardiology visit". Il processo addiziona computazionalmente (Step B) gli score vettoriali scalari dei sub-token frammentati interdipendenti (`card` e `##iology`). Il token isolato `visit` trasferisce a valle la sua integrità quantitativa.

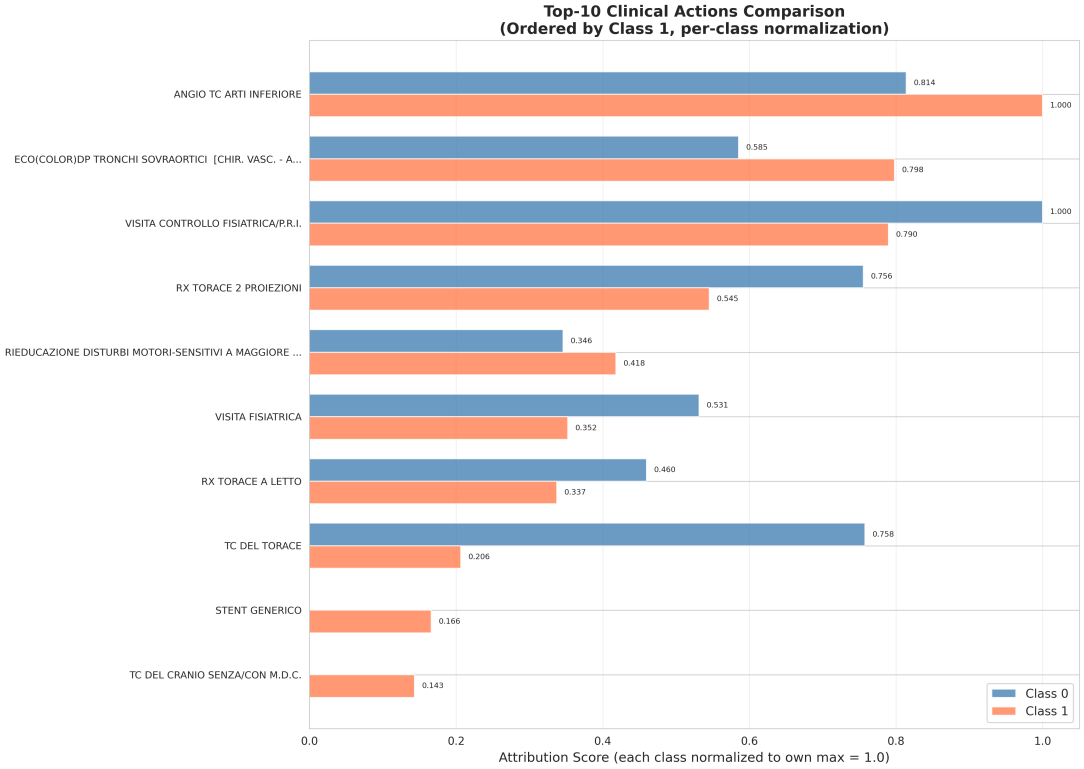

Figura 4.4: Istogramma logaritmico d'estrazione delle metriche d'impatto clinico sulle Top Actions associate dai Gradienti allo spreco del LOS. Il grafico aggrega i dati vettoriali relativi ad un campione mirato di 109 pazienti transitati presso il reparto operativo 701 (Cardiochirurgia) dell'Azienda Ospedaliera di Alessandria. La mancanza di azioni associate alla classe 0 è dovuta al fatto che non ci sono stati pazienti con degenza ≤ 20 che hanno subito le medesime.

4.4.4 Applicazione Pratica: Analisi sulle Degenze di Cardiochirurgia

Il potenziale di questa disaggregazione lessicale è drastico e restituisce istogrammi e *heatmap* (mappe di calore intra-traccia) leggibili, affidabili e matematicamente bilanciate. A titolo dimostrativo (Figure 4.4 e 4.5), sono state estratte e renderizzate le logiche attenzionali relative a un pool di 109 degenze afferenti al Reparto 701 di Cardiochirurgia (Ospedale di Alessandria). Attraverso lo studio dei punteggi aggregati sulle esecutive raggruppabili di questo dipartimento, i decisori aziendali sono ora muniti di un cruscotto semantico **affidabile**. Questo strumento permette di localizzare i veri determinanti processuali che prolungano materialmente le degenze cliniche, **superando il limite interpretativo tipico dei modelli black-box**.

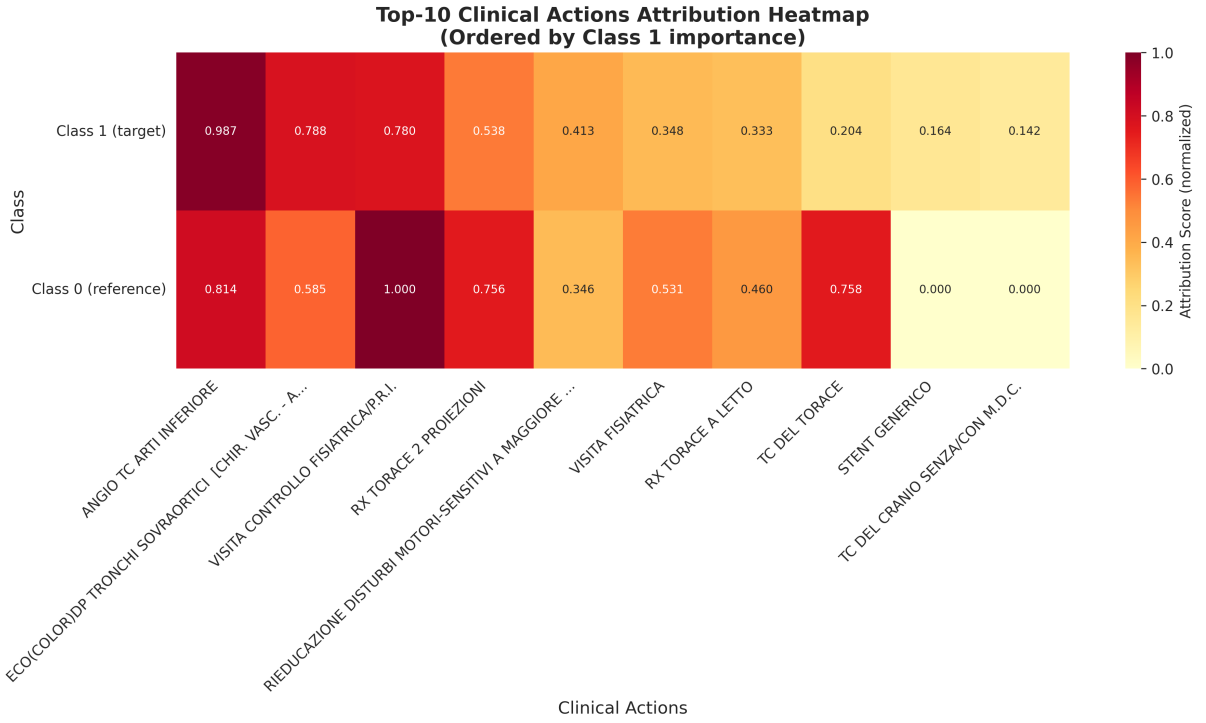

Figura 4.5: Estratto visivo della heatmap Storytelling generata per una traccia in degenza di Cardiochirurgia (Ospedale di Alessandria): l'intensità della sfumatura cromatica isola visualmente e semanticamente lo snodo cronologico ed il collo di bottiglia processuale all'interno del fluire della documentazione del paziente.

4.5 Fattibilità Computazionale e Costi di Deployment

Per valutare la concreta adozione del framework in un ambiente di produzione clinico (deployment aziendale), è stato condotto uno studio parallelo sull'hardware a disposizione

e l'investimento di *runtime*. Le prove sono state misurate impiegando un nodo composto da una singola GPU Nvidia RTX 4090, processore AMD Ryzen 9 5900X (12 core, 24 thread) e 64GB di memoria RAM.

La complessità computazionale asintotica della fase di calcolo e di *fine-tuning* incarna la classica topologia delle architetture Transformer. Semplificandone le componenti in base all'architettura interna, risulta esprimibile come:

$$\mathcal{O}(L^2 \cdot H + L \cdot H^2) \tag{4.3}$$

dove L (impostato a 512 direttamente da Google) riflette la lunghezza massimale della traccia di log passata alla meccanica di *self-attention*, mentre H (768 dimensioni in `bert-base-uncased`) ingombra il computo relativo al layer denso di *feed-forward*.

Al netto del pre-training linguistico intrinseco, lo scoglio di computazione primario per il *deployment* si insinua storicamente nell'innesco degli algoritmi iterativi post-hoc, quali gli *Integrated Gradients*. La complessità estrattiva per l'architettura elaborata è fedelmente approssimabile in:

$$\mathcal{O}(N \cdot K \cdot T \cdot L \cdot H^2) \tag{4.4}$$

dove l'infrastruttura attraversa un bacino di popolarità pari a $N = 1109$ campioni in log, processandoli lungo $K = 2$ direttrici di estrazione dicotomiche avvalendosi attivamente di un delta fine pari a $T = 1500$ steps interpolari. Questi vettori decadranno come poc'anzi sui rami attenzionali base della rete (L ed H).

A fronte di questi scenari computazionali formalmente sfidanti, a livello empirico le metriche *wall-clock* registrate sui server in fase sperimentale si sono rivelate altamente compatibili con una pipeline aziendale agile:

- **Generazione delle storie (Parsing XES testuale):** 1 minuto.
- **Fine-Tuning di BERT (su 5 epoche d'addestramento):** 6 minuti e 30 secondi.
- **Estrazione IG estesa (Analisi completa su 109 tracce loggate):** 8 minuti e 16 secondi.

Questi riscontri temporali sono risultati sostanzialmente stabili e confrontabili trasversalmente a tutti gli approcci di embedding studiati, a parità di dataset.

Processare analiticamente la degenza ospedaliera e l'impatto clinico interpretato per più di mille pazienti in attesa dimissionaria richiede circa 80 minuti di *overhead*. L'utilizzo differito di tali logiche garantisce all'ecosistema *TEXLOS* un'immediata valenza direzionale. In un framework real-world integrato ad un ospedale, innescando l'esportazione automatizzata in asincrono al termine della fascia notturna tramite un trigger temporizzato sul database asincrono o su istanze *cloud ad-hoc*, è possibile consegnare le liste di

monitoraggio priorizzato ed i corrispondenti grafici d'ispezione al *case manager* e ai direttori dipartimentali ad inizio mattinata, preservando i reparti dall'acquisto di hardware di calcolo live sproporzionato ed avvalorando drasticamente i benefici assistenziali attesi all'introduzione empirica.

5. Conclusioni e Sviluppi Futuri

5.1 Sintesi del Lavoro

Il presente lavoro di tesi, inquadrato all'interno del più ampio progetto di ricerca TEXLOS [1], ha affrontato la sfida di diagnosticare ed interpretare i colli di bottiglia nei processi ospedalieri responsabili dell'estensione critica del Length of Stay (LOS) dei pazienti.

Attraverso una rigorosa elaborazione metodologica, si è dimostrato il **significativo** potenziale diagnostico derivante dall'innovativo accostamento tra:

1. L'approccio **Storytelling**, capace di **rappresentare** una traccia evento clinica in linguaggio naturale semantico e **mitigare le criticità legate alla dimensionalità** dei dati.

2. Il classificatore **`bert-base-uncased`**, che è risultato **robusto rispetto alla variabilità terminologica** locale rispetto ai cloni "ClinicalBERT", raggiungendo agilmente un solido grado di accuratezza supervisionata bilanciata (circa l'88%), mitigando al contempo il severo sbilanciamento di classe ospedaliero in virtù della *Focal Loss*.

3. Una variante adattiva degli algoritmi **XAI** (Explainable AI), basata sugli **Integrated Gradients** e implementata con una logica di convergenza adattiva descritta nel Capitolo 4.

Quest'ultimo punto, corroborato dallo sforzo implementativo e computazionale della ricomposizione logica pre- e post-sub-tokenizzazione di BERT in azioni nosocomiali complete, rappresenta il contributo più originale dell'elaborato. Si fornisce concretamente alle amministrazioni sanitarie e ai reparti un cruscotto visivo e tabulare, uno strumento d'indagine in cui punteggi d'impatto rigorosi attribuiscono responsabilità precise alle anomalie ed agli specifici atti clinici, traducendo complesse matrici di gradienti profondi in diagnostica processuale esplicita.

5.2 Limiti dell'Approccio

Pur restituendo metriche solide dinanzi a configurazioni di classe disomogenee, la pipeline qui analizzata è soggetta ad intimi e non negabili limiti concettuali ed empirici. Il primo campanello d'avviso deriva dal parziale fallimento dell'esperimento di data augmentation tramite LLM generativi (es. Ollama, `phi-4`, `nous-hermes2`). Il rumore lessicale intrinseco introdotto da questi modelli linguistici ha degradato la rappresentazione spaziale appresa dal classificatore. Questo dimostra che, attualmente, le pipeline in questo dominio necessitano di dipendere da record ospedalieri originali, aderenti univocamente ai nomenclatori clinici.

In ottica più ampia, va ribadito un importante limite epistemologico riconducibile all'Intelligenza Artificiale odierna. L'architettura proposta, le Attention Maps e la tecnica degli Integrated Gradients restituiscono **metriche e mappe di forte correlazione statistica**. Tali algoritmi documentano matematicamente la covarianza tra l'evento clinico in esame e un incremento nella previsione del *Length of Stay*, ma **non stabiliscono una relazione causale diretta**. In un ambiente di *Process Mining* ospedaliero, l'IA è un sofisticato strumento d'allarme, ma l'imputazione causale e clinica definitiva richiede sempre la **valutazione esclusiva** dello specialista umano.

5.3 Lezioni Apprese e Generalizzabilità del Modello

5.3.1 La Sfida della Scalabilità: Il limite operativo di BERT-Large

Un contributo critico emerso dalla fase sperimentale riguarda l'analisi della scalabilità del modello. I test condotti con l'architettura `bert-large-uncased` hanno rivelato un plateau prestazionale precoce: l'incremento del numero di parametri non ha prodotto benefici proporzionali sulla capacità discriminativa. Questo suggerisce che per un dataset di circa 7.000 tracce, la complessità di BERT-base rappresenti il limite superiore di utilità informativa, oltre il quale si riscontra una saturazione dei pattern apprendibili a fronte di costi computazionali (VRAM) non sostenibili.

5.3.2 Prospettive di Generalizzabilità Clinica

Sebbene la sperimentazione si sia concentrata su una selezione di 20 DRG (Diagnosis Related Groups), la pipeline implementata — basata sulla traduzione deterministica dei log in linguaggio naturale — risulta intrinsecamente generalizzabile ad altri contesti clinici. Il modello ha dimostrato di poter astrarre concetti medici universali senza la necessità di

pre-addestramento specifico su corpus clinici (come ClinicalBERT), rendendo la soluzione facilmente esportabile in diverse unità operative.

5.4 Sviluppi Futuri

L'approccio semantico *Storytelling* + IG delinea direzioni promettenti per i futuri sviluppi nel campo del Process Mining sanitario. Nello specifico panorama del progetto TEXLOS [1], una ramificazione potenziale concerne non tanto il consolidamento della classificazione finale (l'end-point predittivo netto), bensì l'anticipazione online della XAI.

Tra gli sviluppi futuri si auspica la transizione dall'analisi "post-mortem" (su paziente dimesso e traccia ormai conclusa) verso una classificazione "runtime" dinamica, in cui gli Integrated Gradients parziali vengono calcolati in diretta ai vari checkpoint quotidiani della degenza. Approcci di tipo *Next-Activity-Prediction* corredati da estrazioni XAI ad alta densità consentirebbero all'amministrazione un aggiustamento di rotta predittivo in tempo reale durante i processi di cura, inaugurando logiche di controllo manageriale proattive per l'efficienza clinica.

Bibliografia

[1] Roberta Bellini, Simone Garau, Riccardo Gualiumi, Giorgio Leonardi, Stefania Montani, Manuel Striani, and Cristian Zanelli. Investigating the impact of outpatient services on length of stay: an easily interpretable approach. *Frontiers in Artificial Intelligence*, 9, 2026. ISSN 2624-8212. doi: 10.3389/frai.2026.1746547. URL https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2026.1746547.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[4] Valeria Pasquadibisceglie, Annalisa Appice, Giovanna Castellano, and Corrado Mencar. Leveraging a large language model (llm) to predict hospital admissions of emergency department patients. *Expert Systems with Applications*, page 128224, 2025. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2025.128224.

[5] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[6] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):318–327, 2017.

Ringraziamenti

TODO

Dettagli implementativi

In questa appendice si forniscono dettagli aggiuntivi sull'implementazione dei modelli e degli esperimenti presentati nella tesi.

.1 Configurazione Hardware

L'addestramento e la valutazione del modello BERT per la classificazione delle tracce cliniche, insieme alle computazionalmente costose sessioni di Explainable AI (Integrated Gradients), sono state condotte sulla seguente workstation:

- CPU: AMD Ryzen 9 5900X
- GPU: NVIDIA GeForce RTX 4090 (24 GB VRAM)
- RAM: 64 GB
- Sistema Operativo: Linux Ubuntu 24.04.3 LTS (ambiente WSL su Windows)

L'elevata capacità di VRAM della GPU è risultata fondamentale per l'impiego del modello Transformer `bert-base-uncased` e per garantire l'efficiente esecuzione dell'algoritmo IG Adattivo fino a 5500 steps di campionamento.

.2 Configurazione Software e Linguaggi

L'intera pipeline metodologica, dal preprocessing dei log XES fino all'analisi dei risultati XAI, è stata sviluppata in ambiente Python:

- Python 3.11.13
- PyTorch 2.8 (come framework principale per il Deep Learning e il Fine Tuning)
- HuggingFace `transformers` (per caricamento e gestione del modello BERT)
- Captum 0.8 (libreria per il calcolo degli Integrated Gradients)
- Optuna 4.5 (per l'ottimizzazione bayesiana degli iperparametri)

- Scikit-learn 1.7.2 (per le metriche di valutazione come *Balanced Accuracy* e lo Stratified K-Fold CV)

.3 Dettagli sull'Hyperparameter Tuning

Il processo di tuning, gestito da Optuna e rifinito manualmente, aveva l'obiettivo di ottimizzare le prestazioni rispettando le architetture originali del Transformer. Di seguito sono riportati gli iperparametri finali del modello classificatore:

- Base model: `bert-base-uncased` (12 layers, 768 hidden size, 12 attention heads)
- Batch size: Adattiva (limite VRAM)
- Optimizer: AdamW
- Funzione di costo: Focal Loss (γ e α bilanciati per amplificare l'errore sulla classe $LOS \geq 20$)
- Validation strategy: Stratified 5-Fold Cross Validation