UNIVERSITÀ LUM GIUSEPPE DEGENNARO

Dipartimento di Ingegneria

Corso di Laurea in Ingegneria Informatica
per la Transizione Digitale (L-8)

TESI DI LAUREA

IN

"Machine Learning e Artificial Intelligence"

Progettazione di una piattaforma cloud per sistemi di supporto clinico odontoiatrico basati su Intelligenza Artificiale

Relatore:

Prof. Ing. Giuseppe Loseto

Laureando:

Giuseppe Giorgio / 17568

Correlatori:

Prof. Giuseppe Troiano

Dott. Francesco Fanelli

Anno Accademico: 2025/2026

Indice

Indice Immagini

Indice Tabelle

Introduzione

L'introduzione dell'*Intelligenza Artificiale* (IA) sta progressivamente ridefinendo gli standard operativi in odontoiatria e, in particolar modo, nella parodontologia. Il trattamento del paziente parodontale impone infatti la raccolta e l'elaborazione di numerosi dati, che spaziano dai parametri clinici rilevati durante il sondaggio fino all'interpretazione dei reperti radiografici. L'esecuzione di tali procedure genera un notevole carico operativo e cognitivo: il clinico si trova spesso costretto a interrompere l'esame per compilare la cartella, oppure deve necessariamente ricorrere al supporto di un assistente.

Per superare questi inconvenienti operativi è possibile integrare diverse tecnologie emergenti in moderne piattaforme decisionali progettate per supportare i professionisti del settore parodontale e implantologico integrando modelli generativi e sistemi di visione artificiale basati su tecniche di *Machine Learning* (ML). Come riferimento architetturale è stato utilizzato il progetto *PerioGPT*, che impiega *Large Language Models* (LLM) combinati con un'architettura di *Retrieval-Augmented Generation* (RAG). Questo approccio vincola la generazione delle risposte alla consultazione diretta della letteratura scientifica aggiornata, superando così le tipiche imprecisioni e i limiti di conoscenza dei modelli generalisti.

Lo scopo di questo lavoro di tesi consiste nel progettare un ecosistema digitale integrato, pensato per l'assistenza clinica in parodontologia. L'architettura proposta si articola in tre moduli indipendenti, studiati per snellire l'operatività quotidiana dello studio:

- *Supporto decisionale basato sull'evidenza*: un sistema conversazionale con architettura RAG in grado di estrarre informazioni da un database di documenti medici, restituendo al clinico risposte puntuali e ancorate alla letteratura.

- *Automazione della cartella parodontale*: un'interfaccia di dettatura vocale hands-free. L'impiego combinato di algoritmi *Speech-to-Text* e tecniche di *Natural Language Processing* (NLP) permette di tradurre il parlato dell'operatore e popolare la cartella clinica in tempo reale.

- *Diagnostica per ortopantomografia*: la valutazione del rischio parodontale su *ortopantomografie* (OPT) è affidata a un modello *Vision Transformer* (DeiT3). Per garantire

l'interpretabilità clinica dei risultati, l'inferenza è accompagnata da tecniche di *Explainable Artificial Intelligence* (XAI); queste generano mappe visive delle aree che hanno determinato la decisione della rete neurale, un requisito oggi indispensabile per l'accettazione degli algoritmi in ambito medico.

In particolare, la tesi è strutturata in quattro capitoli principali:

1. Il *primo capitolo* analizza lo stato dell'arte dell'IA in ambito clinico, in particolare nell'odontoiatria. Descrive l'impiego dei Large Language Models in ambito clinico, i sistemi Speech-to-Text e l'importanza della XAI per la trasparenza diagnostica.

2. Il *secondo capitolo* dettaglia l'architettura della piattaforma cloud proposta. Illustra l'integrazione tra modelli generativi locali e remoti, la pipeline di trascrizione vocale e l'interfaccia per visualizzare le spiegazioni a supporto del medico.

3. Il *terzo capitolo* valuta l'usabilità del sistema proposto in scenari simulati. Riporta l'analisi delle prestazioni, discute la sostenibilità dell'infrastruttura e delinea le possibilità di integrazione con gli attuali sistemi informativi sanitari.

4. Il *quarto capitolo* discute le conclusioni del lavoro.

Capitolo 1

Intelligenza Artificiale in ambito medico

1.1 Large Language Models e modelli generativi

Gli LLMs costituiscono un sottoinsieme avanzato delle architetture di *Deep Learning*, specificamente ingegnerizzato per l'elaborazione, la comprensione e la generazione del linguaggio naturale. Dal punto di vista teorico, l'attuale stato dell'arte dei modelli linguistici si fonda sul superamento delle classiche *Recurrent Neural Network* (RNN) e *Long Short-Term Memory* (LSTM) a favore dell'architettura Transformer, mostrata in Figura 1.1 e introdotta formalmente da Vaswani et al. nel 2017 [1].

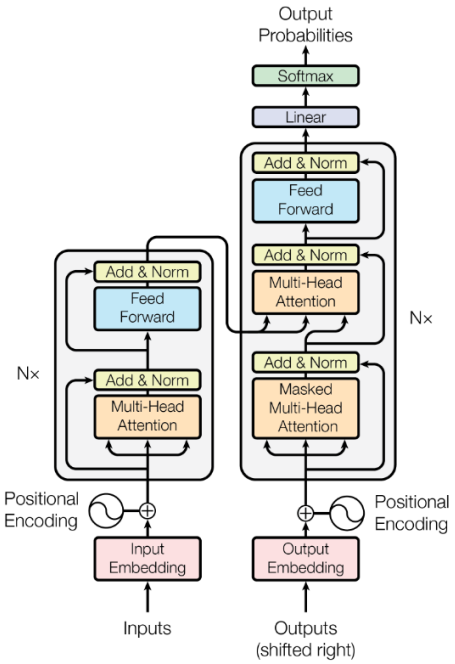

Figura 1.1: Architettura Transformer [1]

Il nucleo di questa architettura è rappresentato dal meccanismo matematico di *self-attention*. Tale approccio computazionale consente al modello di valutare e pesare la rilevanza di ogni singolo elemento (token) rispetto all'intera sequenza di testo in input.

Il processo di creazione di un LLM si articola tipicamente in due macro-fasi sequenziali:

- **Pre-training**: il modello viene sottoposto ad un addestramento non supervisionato su enormi blocchi testuali. L'obiettivo della rete è prettamente statistico e consiste nel Next-Token Prediction, ovvero il calcolo della distribuzione di probabilità del token successivo data una sequenza precedente. Durante questa fase, la rete astrae e assimila la sintassi, la grammatica e un'estesa conoscenza generale.

- **Fine-tuning**: la rete pre-addestrata viene rifinita su dataset specifici. L'impiego di tecniche supervisionate e algoritmi di *Reinforcement Learning from Human Feedback* (RLHF) specializza il modello nell'esecuzione di istruzioni complesse, in base al contesto di riferimento.

1.1.1 Stato dell'arte in ambito medico: vantaggi e criticità

Storicamente, l'applicazione dell'Intelligenza Artificiale in odontoiatria si è concentrata sull'analisi di dati visivi tramite Reti Neurali Convoluzionali per compiti di narrow AI, come l'identificazione di lesioni cariose o la misurazione della perdita ossea [3]. La transizione odierna verso l'*Intelligenza Artificiale Generativa* o Generative AI (GenAI) permette di superare l'analisi di immagini isolate, automatizzando la stesura di referti medici, sintetizzando cartelle cliniche elettroniche e offrendo sistemi di supporto decisionale clinico.

Tuttavia, l'applicazione clinica diretta di questi sistemi computazionali incontra profondi limiti epistemologici. La conoscenza di un LLM è racchiusa nei suoi pesi parametrici e "congelata" al termine dell'addestramento *(knowledge cut-off)*. In discipline come l'odontoiatria, affidarsi a informazioni statiche rappresenta un rischio clinico inaccettabile. Inoltre, la natura probabilistica della generazione favorisce le "allucinazioni": la produzione di contenuti sintatticamente plausibili ma fattualmente errati [3].

Uno studio del 2025 ha valutato architetture LLM (tra cui DeepSeek, Claude 3.5, GPT-4o e Gemini) nell'estrazione di dati numerici complessi per meta-analisi odontoiatriche [2]. La ricerca ha evidenziato come l'incremento del carico cognitivo faccia lievitare l'errore sistemico. Modelli più sicuri (come Claude e DeepSeek) tendono all'"omissione silenziosa" (missed data) in caso di incertezza, mentre altri (come Gemini) hanno manifestato allucinazioni pure, alterando l'integrità scientifica dei dati estratti [2]. Questo conferma come, in medicina, la supervisione umana resti imprescindibile.

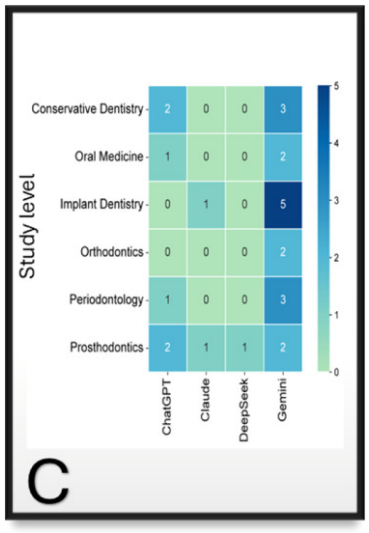

Figura 1.2: Heatmap degli errori [2]

La heatmap in Figura 1.2 mostra il numero di errori (allucinazioni e dati mancanti) nei diversi ambiti odontoiatrici per ciascun modello.

1.1.2 Architetture Retrieval-Augmented Generation

Per superare i limiti architetturali degli LLM, è stato sviluppato il paradigma RAG [7]. Questo approccio ibrido unisce le capacità di comprensione e generazione del linguaggio naturale degli LLM con sistemi di *Information Retrieval* (IR) ad alta precisione, permettendo al modello di accedere dinamicamente a database esterni aggiornati prima di generare una risposta.

L'infrastruttura RAG si articola in una pipeline complessa che collega due macro-processi: il recupero (retrieval) e la generazione. La fase preliminare e fondamentale è l'indicizzazione (indexing): viene selezionata una raccolta di documenti di dominio specifico, che nel caso odontoiatrico può includere abstract di PubMed, revisioni sistematiche, protocolli istituzionali, libri di testo specializzati e cartelle cliniche elettroniche anonimizzate. Poiché i testi integrali di questi documenti superano spesso il limite della "*context window*" (quantità massima di testo che il modello può elaborare in una singola iterazione), il materiale viene suddiviso in unità semantiche più piccole attraverso un processo denominato *chunking*.

Ogni frammento di testo (chunk) viene successivamente processato da un modello di embedding (Figura 1.3) che mappa il contenuto semantico del testo in un vettore numerico continuo all'interno di uno spazio ad alta dimensionalità. Questi vettori vengono archiviati in database vettoriali specializzati. L'eleganza di questo spazio matematico risiede nel fatto che frammenti di testo con significati clinici simili verranno posizionati spazialmente vicini, indipendentemente dalla precisa sintassi utilizzata. Quando un clinico sottopone una query,

questa subisce lo stesso processo di embedding e viene proiettata nello spazio vettoriale. Il sistema calcola quindi la prossimità geometrica (spesso misurata tramite la similarità coseno definita come $\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$) per recuperare i frammenti ("top-k chunks") più pertinenti per risolvere il quesito medico.

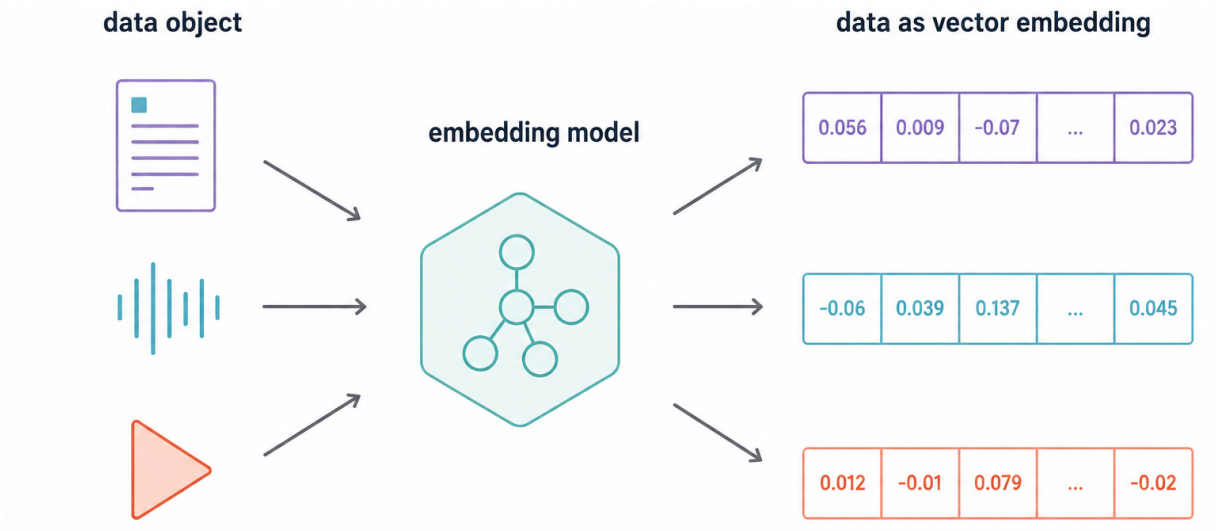

Figura 1.3: Modello di Embedding

L'efficacia clinica di un sistema RAG odontoiatrico, la cui pipeline generica è mostrata in Figura 1.4, dipende intrinsecamente dall'algoritmo di recupero delle informazioni impiegato. La letteratura più recente (2024-2026) classifica le architetture RAG in tre varianti evolutive principali [7]:

- **Naive RAG**: questa architettura di base utilizza prevalentemente metodi di Sparse Retrieval, i quali si fondano sulla sovrapposizione lessicale esatta tra la query dell'utente e i documenti archiviati. Impiega algoritmi statistici classici, come TF-IDF (Term Frequency-Inverse Document Frequency) e BM25, per assegnare pesi maggiori alle parole che appaiono frequentemente in un singolo documento ma raramente nell'intera collezione documentale. Pur garantendo velocità di esecuzione estreme (con latenze di inferenza di circa 120 millisecondi), alta interpretabilità computazionale e l'assenza di necessità di addestramento specifico, i metodi di Sparse Retrieval falliscono nel cogliere le sfumature semantiche. In ambito biomedico questo rappresenta un limite critico, data l'incapacità del sistema di gestire l'ampia rete di sinonimi (es. "malattia parodontale" vs "parodontite cronica"), acronimi tipici dell'odontoiatria.

- **Advanced RAG**: nasce per superare la rigidità lessicale del modello Naive attraverso l'implementazione del *Dense Retrieval*. I retriever densi (come il *Dense Passage*

Retrieval (DPR) utilizzano encoder neurali per mappare il testo in uno spazio di embedding semantico, permettendo al sistema di recuperare documenti rilevanti anche in assenza delle parole esatte usate dal clinico. L'apice ingegneristico di questa categoria è rappresentato dall'*Hybrid Retrieval*, un approccio che fonde i vantaggi di entrambe le metodologie tramite tecniche di *cross-encoding* e *Reciprocal Rank Fusion* (RRF). Test empirici indicano che una pipeline ibrida (DPR + BM25 + cross-encoder) garantisce massimi livelli di precisione nel recupero documentale (Precision), abbattendo al contempo il tasso di allucinazioni generative a percentuali inferiori al 5.8% [8]. Tuttavia, questa architettura comporta un elevato costo computazionale e una marcata suscettibilità al *domain shift* (un calo prestazionale che si verifica elaborando testi linguisticamente eterogenei, come gli appunti telegrafici di una cartella clinica confrontati con la prosa formale degli articoli accademici).

- **Modular RAG**: estende le capacità dell'*Advanced RAG* scomponendo la pipeline in moduli indipendenti e riconfigurabili (*Search*, *Memory*, *Routing*, *Rewrite*, *Rerank*, *Fusion*) collegati attorno a un blocco RAG centrale. Il retriever (tipicamente un encoder denso duale) e il generatore vengono addestrati in modo disaccoppiato: il primo tramite loss contrastiva sui documenti, il secondo tramite massima verosimiglianza sui top-k documenti recuperati, così da poter sostituire o ri-addestrare singole componenti senza modificare l'intero sistema [7]. In questo quadro, l'integrazione di ontologie cliniche e *Knowledge Graph* nei moduli di ricerca e fusione è stata testata con successo in scenari a forte impatto clinico: uno studio del 2025, basato su un'ontologia proposta dalla community Open Biological and Biomedical Ontology (OBO) Foundry, mostra come una struttura di dati rigorosamente ontologica a monte del processo generativo permetta di mitigare il ragionamento *black-box* tipico degli LLM [9].

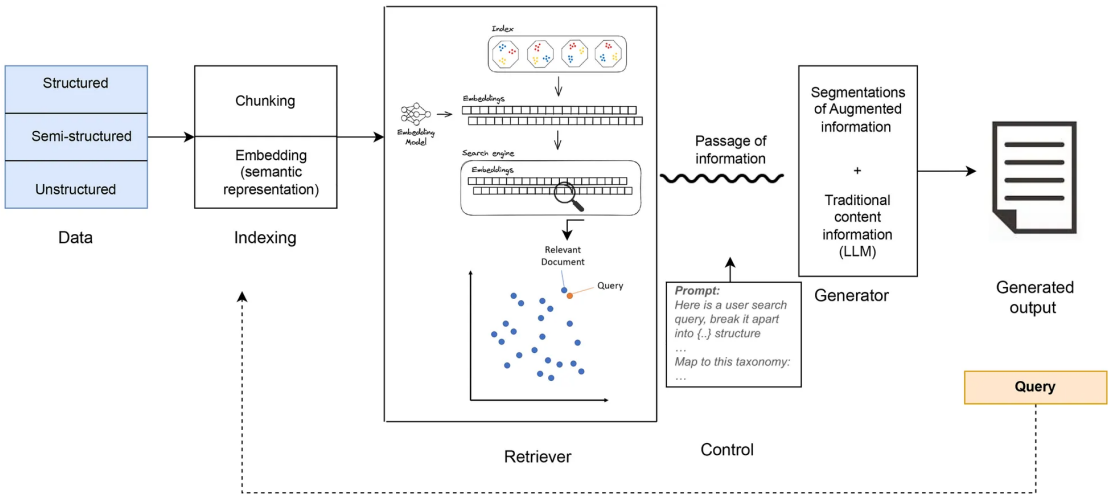

Figura 1.4: RAG Pipeline [3]

1.1.3 Applicazioni in Odontoiatria e Parodontologia

Nell'ambito dell'odontoiatria, l'impiego di metodologie basate sull'Apprendimento Automatico si è inizialmente focalizzato sulla classificazione di immagini tramite Reti Neurali Convoluzionali, utili per la misurazione radiografica della perdita ossea alveolare e per la stadiazione automatica della parodontite [10]. L'inserimento dei LLMs per compiti di elaborazione del linguaggio e decision-making costituisce uno sviluppo recente e in rapida espansione all'interno della pratica clinica [7]. Tuttavia, l'applicazione degli LLM generalisti alla parodontologia presenta ancora limiti evidenti. Quando vengono testati su decisioni cliniche specialistiche (come la scelta terapeutica per i difetti delle forcazioni secondo le linee guida della European Federation of Periodontology (EFP)) i modelli non modificati forniscono spesso indicazioni obsolete o in contrasto con le evidenze scientifiche più recenti [3]. Questi errori dimostrano che la complessità della parodontologia richiede infrastrutture dedicate: risulta indispensabile unire le capacità di ragionamento e fluidità dei modelli linguistici con l'affidabilità di database scientifici costantemente aggiornati, un traguardo raggiungibile proprio attraverso l'architettura RAG.

Per rispondere a questa necessità clinica, è stato introdotto PerioGPT: un modello GPT-4o specializzato e ri-addestrato ("*reinstructed*") per il dominio parodontale tramite un'infrastruttura RAG [11]. Presentata nella sua versione iniziale nel 2025 come proposta di ricerca sul *Journal of Clinical Periodontology* [11], la piattaforma ha dimostrato un'accuratezza dell'81,16% nel test dell'*American Academy of Periodontology*, superando ampiamente i chatbot generalisti e fornendo ai clinici uno strumento decisionale evidence-based affidabile.

1.2 Natural Language Processing e Speech-to-Text

La documentazione clinica rappresenta il fondamento informativo essenziale per garantire la continuità delle cure, la comunicazione interdisciplinare e la tutela medico-legale all'interno delle moderne infrastrutture sanitarie. Storicamente gestita tramite archivi cartacei, ha subito un'importante transizione verso le cartelle cliniche elettroniche e le rispettive versioni specialistiche odontoiatriche. Questa digitalizzazione ha centralizzato le informazioni, ma ha generato colli di bottiglia operativi imprevisti, dovuti principalmente all'inserimento manuale dei dati. I tradizionali metodi di interazione tramite tastiera e mouse si sono rivelati onerosi in termini di tempo, incrementando il carico cognitivo del clinico e introducendo un potenziale rischio di errori di trascrizione [12]. Per risolvere queste inefficienze, l'informatica biomedica si è orientata verso l'integrazione di due paradigmi tecnologici complementari: i sistemi di *Automatic Speech Recognition* (ASR), comunemente noti come Speech-to-Text, e gli algoritmi di Natural Language Processing. La pipeline tecnologica prevede una sequenza operativa ben definita, rappresentata in Figura 1.5. In una prima fase, i modelli ASR

convertono il segnale acustico vocale del clinico in una trascrizione testuale grezza. Successivamente, i moduli NLP elaborano tale trascrizione testuale destrutturata per estrarre entità cliniche, relazioni e parametri numerici, classificandoli all'interno di schemi di dati predefiniti. L'unione e la sincronizzazione di queste due tecnologie abilita la modalità operativa *hands-free*, consentendo al medico o all'igienista di dettare i parametri in tempo reale durante l'esame, eliminando di fatto le interruzioni burocratiche dal flusso di lavoro primario [13].

I sistemi di riconoscimento vocale automatico hanno attraversato una profonda trasformazione architetturale nel corso dell'ultimo decennio. I modelli di prima e seconda generazione si basavano su approcci puramente statistici, impiegando i *Modelli Nascosti di Markov* (HMM) in combinazione con i *Modelli di Miscela Gaussiana* (GMM). Tali architetture, sebbene efficaci in ambienti di registrazione controllati e per vocabolari di base, presentavano limitazioni insormontabili nella modellazione di contesti linguistici a lungo raggio e nella resistenza alle variazioni fonetiche. L'attuale stato dell'arte si fonda sull'implementazione intensiva di Reti Neurali Profonde e, più recentemente, sulle architetture come *Transformer*, *Conformer* e il modello *OpenAI Whisper*. A differenza dei sistemi sequenziali tradizionali, l'architettura *Transformer* elabora l'audio sfruttando complessi meccanismi di *self-attention*. Questo paradigma computazionale consente alla rete di ponderare l'importanza di specifici segmenti acustici rispetto all'intero flusso audio, permettendo di catturare dipendenze linguistiche complesse e abbassando drasticamente le metriche di errore, come la *Word Error Rate* (WER) [14].

L'applicazione e il rilascio in produzione dei sistemi ASR in ambito sanitario, e specificatamente nel dominio odontoiatrico, richiede tuttavia il superamento di sfide ambientali notevoli. Le sale operative odontoiatriche sono caratterizzate da un elevato inquinamento acustico di fondo, generato dalla strumentazione dinamica in uso (quali turbine, micromotori e aspiratori chirurgici). A ciò si aggiunge l'alterazione dello spettro vocale causata dall'utilizzo di Dispositivi di Protezione Individuale (come mascherine mediche) da parte del clinico. Di conseguenza, per garantire una trascrizione affidabile, il segnale audio deve essere processato con tecniche avanzate di *data augmentation* (come la perturbazione della velocità) e modelli neurali di soppressione del rumore ambientale [15]. Prendendo spunto dalle architetture di frontiera sviluppate per il riconoscimento vocale in scenari *mission-critical* ad altissima interferenza, il segnale audio misto catturato dal microfono viene innanzitutto trasformato dal dominio del tempo a quello della frequenza tramite la Trasformata di Fourier a Breve Termine (Short-Time Fourier Transform, STFT). La conversione matematica necessaria per fornire l'input alla rete neurale estrattiva è descritta dalla seguente equazione:

$$X[k] = \sum_{n=0}^{N-1} x[n] \times e^{-\frac{2\pi k n}{N}} \tag{1.1}$$

Questa formula descrive la Trasformata Discreta di Fourier (Discrete Fourier Transform, DFT) di base; applicandola iterativamente su finestre temporali sovrapposte si ottiene la STFT, che genera lo spettrogramma bidimensionale. In essa, N indica i campioni temporali totali, $x[n]$ l'n-esimo campione e $X[k]$ il coefficiente spettrale risultante. Fornendo questi spettri in input a modelli Deep Learning (come gli encoder U-Net), il sistema impara a mappare e isolare il rumore di fondo [13]. L'addestramento su tali scenari garantisce sistemi ASR altamente robusti, capaci di trascrivere fedelmente il complesso gergo specialistico anche in ambienti rumorosi [15]. Sottoporre la rete neurale a questa diversificata distribuzione di scenari acustici assicura la creazione di sistemi ASR estremamente robusti, capaci di garantire una trascrizione affidabile in ambienti rumorosi e di riconoscere senza ambiguità il complesso gergo specialistico.

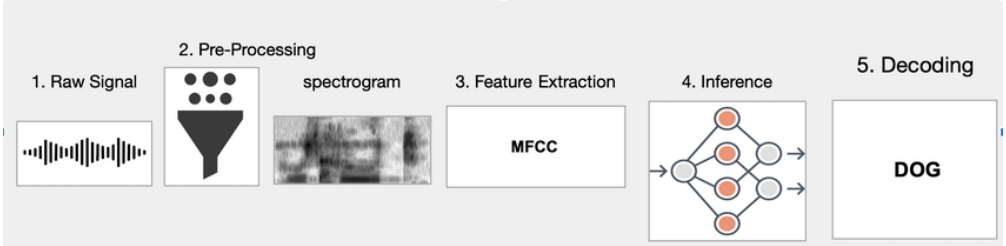

Figura 1.5: Pipeline di Automatic Speech Recognition [4]

La conversione accurata del segnale vocale in testo costituisce soltanto il primo modulo del processo di automazione. La trascrizione generata si presenta infatti come una "narrazione clinica non strutturata", ovvero una sequenza testuale priva di etichette logiche, che necessita di un'elaborazione semantica per divenire intellegibile e computabile per i sistemi informativi sanitari. Il Natural Language Processing interviene in questa fase per decodificare la semantica del linguaggio umano e mappare le informazioni su vettori strutturati [12]. Le tecniche convenzionali di NLP applicate al campo biomedico si sono storicamente affidate ad algoritmi di *Named Entity Recognition* (NER) e *Relation Extraction* (RE). Questi modelli venivano addestrati in modo supervisionato per individuare specifiche categorie di parole all'interno del testo, come nomi di patologie, riferimenti anatomici e misurazioni cliniche. Sebbene precisi, tali sistemi basati su regole risultavano rigidi e incapaci di adattarsi a formulazioni linguistiche atipiche o dialettali.

L'integrazione recente dei LLMs ha ridefinito le capacità operative dell'estrazione dati clinica. Queste reti neurali generative, avendo assimilato i fondamenti della linguistica e della medicina su vastissime fonti documentali, permettono di estrarre informazioni altamente strutturate sfruttando tecniche di *prompt engineering*. In un tipico scenario clinico odontoiatrico vocale, l'algoritmo NLP riceve in input una stringa narrativa destrutturata (ad esempio: "*elemento uno sei, sondaggio mesiale cinque millimetri con sanguinamento*").

Il modello elabora il testo, isola l'identificativo anatomico ("dente 1.6"), classifica il valore numerico associandolo al parametro di riferimento ("PD mesiale: 5") ed estrae le condizioni cliniche collaterali ("DoP - Sanguinamento al Sondaggio: presente").

Queste entità, una volta astratte dal testo libero, vengono convertite in tempo reale in formati dato standardizzati, interoperabili e tipizzati, come stringhe in formato JavaScript Object Notation (JSON), pronti per popolare direttamente le matrici di un database relazionale o i campi visivi di una cartella clinica elettronica. Questa architettura software elimina la necessità di interfacce intermedie e di validazioni manuali post-visita, automatizzando interamente il flusso del dato, dall'acquisizione vocale fino all'archiviazione strutturata.

1.3 Explainable Artificial Intelligence in ambito sanitario

L'integrazione sistematica dell'Intelligenza Artificiale all'interno dei flussi di lavoro clinici ha subito una trasformazione radicale nell'ultimo decennio, evolvendo da componente sperimentale a pilastro essenziale della diagnostica avanzata e della telemedicina [16]. In particolare, l'adozione di modelli di Deep Learning ha permesso di raggiungere livelli di accuratezza senza precedenti in compiti quali la segmentazione di immagini radiologiche e la classificazione istopatologica [6]. Tuttavia, questa capacità computazionale superiore è stata ottenuta a costo della trasparenza, portando alla diffusione di modelli definiti "*black box*". Tali strutture algoritmiche presentano una logica interna inaccessibile o indecifrabile per l'operatore umano, rendendo impossibile tracciare il percorso logico che conduce dall'input (es. una radiografia) all'output (es. una diagnosi di parodontite) [17].

In ambito sanitario, l'opacità dei modelli non rappresenta solo un limite tecnico, ma un ostacolo etico e legale che coinvolge la sicurezza del paziente e la responsabilità professionale del medico [16]. La Explainable Artificial Intelligence emerge come il campo di ricerca dedicato allo sviluppo di metodologie atte a rendere i risultati dell'IA comprensibili, trasparenti e giustificabili. L'obiettivo è garantire che le decisioni automatizzate possano essere validate e integrate criticamente nel giudizio clinico [18].

La tassonomia attuale riguardante la XAI divide le tecniche esistenti in due macrocategorie fondamentali in base al momento e alla modalità con cui l'interpretabilità viene integrata nel ciclo di vita del modello [19].

- *Tecniche Ante-hoc*: note come tecniche di *intrinsic interpretability*, prevedono la progettazione di modelli intrinsecamente interpretabili fin dalla fase di sviluppo. In questo caso, la struttura del modello è concepita per essere comprensibile e la spiegazione coincide direttamente con il suo funzionamento (ad esempio regressione lineare e alberi decisionali), rendendo possibile analizzare in modo trasparente il processo che porta dall'input all'output [19].

- *Tecniche Post-hoc*: note come metodi di *extrinsic interpretability*, vengono applicate a modelli già addestrati, tipicamente complessi e non interpretabili (black-box), con l'obiettivo di fornire spiegazioni a posteriori senza modificarne la struttura o i parametri. Questi approcci mirano ad approssimare o analizzare il comportamento del modello e trovano particolare applicazione nei sistemi di classificazione e supporto decisionale in ambito clinico, come la diagnostica per immagini e l'analisi di dati testuali sanitari. In tali contesti, sono state sviluppate diverse tecniche specializzate, tra cui:

  - *Saliency Maps e Grad-CAM*: tecniche utilizzate principalmente in modelli per immagini che producono mappe di calore sovrapposte all'input originale. In particolare, il *Gradient-weighted Class Activation Mapping* (Grad-CAM) sfrutta i gradienti associati a una specifica classe (ad esempio una patologia) rispetto agli ultimi layer convoluzionali per evidenziare le regioni dell'immagine che hanno contribuito maggiormente alla predizione. Questo consente di ottenere una localizzazione qualitativa delle aree rilevanti per la classificazione [5].
  - *Feature Attribution (SHAP e LIME)*: metodi model-agnostic che assegnano un punteggio di importanza alle caratteristiche di input rispetto a una specifica predizione. La tecnica SHapley Additive exPlanations (SHAP) si basa su principi della teoria dei giochi per stimare il contributo marginale di ciascuna feature, mentre Local Interpretable Model-agnostic Explanations (LIME) costruisce un modello interpretabile locale che approssima il comportamento del modello complesso in prossimità dell'istanza considerata. In ambito clinico, queste tecniche permettono di analizzare l'influenza di variabili testuali o tabellari (ad esempio parametri clinici) sul risultato del modello [20].
  - *Attention Rollout*: tecnica applicata a modelli basati su architetture Transformer che consente di analizzare la propagazione dell'informazione attraverso i meccanismi di attenzione. Combinando le matrici di attenzione tra i diversi layer, il rollout permette di stimare come l'importanza dei token di input contribuisca alla rappresentazione finale, offrendo una visualizzazione del flusso informativo che porta alla decisione del modello [21].

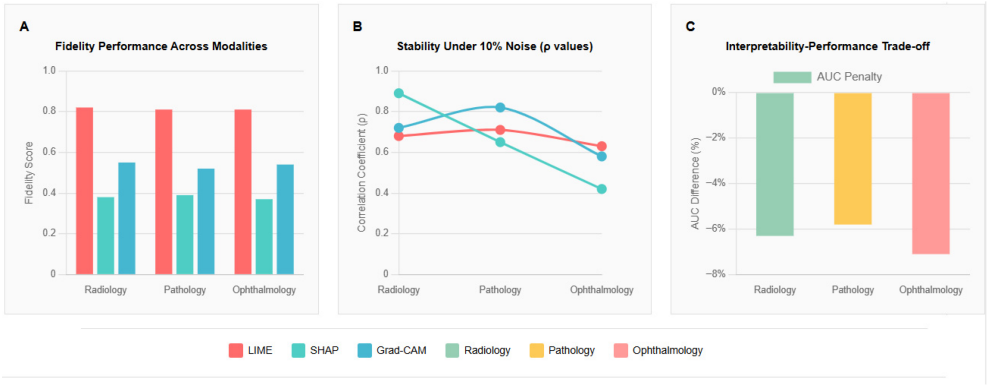

Figura 1.6: Confronto tra tecniche di XAI in ambito di imaging medico [5]

L'architettura di un sistema sanitario orientato alla trasparenza richiede una pipeline integrata che colleghi l'elaborazione del dato alla sua visualizzazione interpretativa. Seguendo i moderni paradigmi di ingegneria del software medico, il flusso informativo si articola in tre stadi: l'acquisizione del segnale (audio, testo o immagine), l'inferenza tramite modelli predittivi e lo stadio di spiegazione (*Explainer Layer*).

In questa pipeline, lo stadio di spiegazione opera in parallelo o sequenzialmente al modello principale. Nel caso dell'elaborazione di dati strutturati (come parametri clinici o misurazioni ambientali), l'algoritmo XAI restituisce un set di coefficienti di importanza, come gli *SHAP values*. Come illustrato in Figura 1.7, queste tecniche permettono di visualizzare graficamente l'impatto positivo o negativo di ogni singola variabile sulla previsione algoritmica globale. Analogamente, nel caso dell'analisi radiografica, quando un modello di visione (come un *Vision Transformer*) identifica un rischio patologico, il modulo genera simultaneamente una visualizzazione spaziale sotto forma di mappa di salienza.

Questi dati di spiegazione vengono infine aggregati in una dashboard clinica, fornendo al medico non solo il risultato numerico sintetico, ma l'intera evidenza analitica necessaria per la validazione. Questo approccio metodologico riduce drasticamente il rischio di accettazione acritica dell'output algoritmico (*automation bias*) e favorisce la scoperta di eventuali correlazioni errate apprese dal modello durante la fase di addestramento [6].

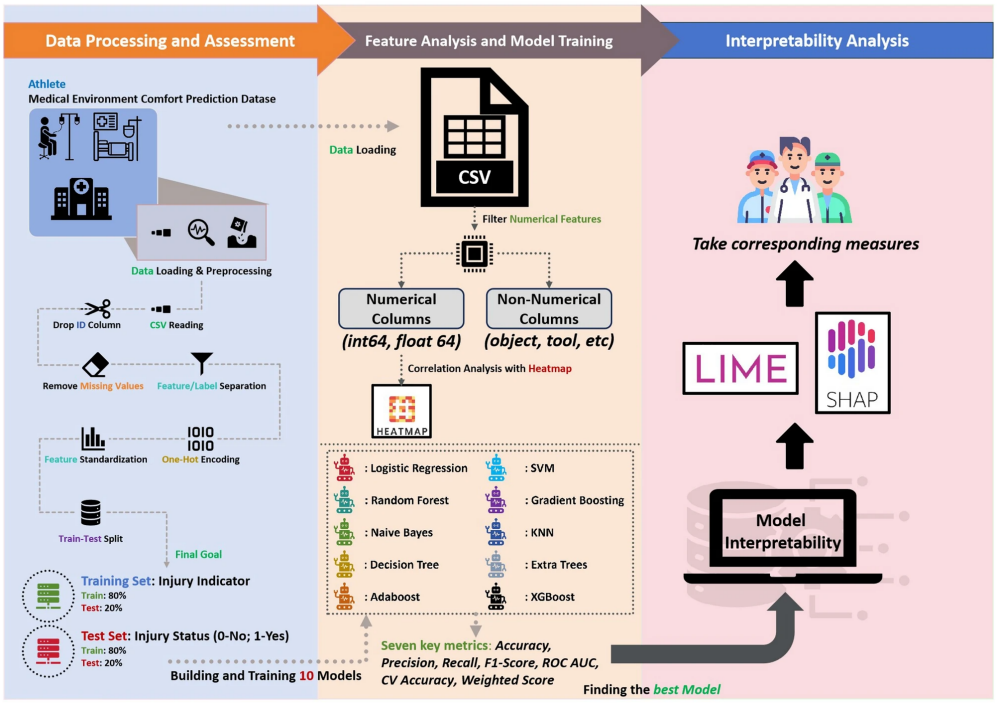

Figura 1.7: Esempio di XAI per un modello di Machine Learning in ambito medico [6]

Capitolo 2

Architettura della piattaforma cloud

L'architettura logica del sistema proposto illustrata in Figura 2.1, che si propone come evoluzione della piattaforma PerioGPT [11], si fonda su un paradigma cloud-native e sul principio ingegneristico di netta separazione delle responsabilità (Separation of Concerns). Tale approccio isola rigorosamente il livello di presentazione dall'infrastruttura di calcolo intensivo e dalla gestione persistente dei dati vettoriali. Questa suddivisione logica garantisce un'elevata manutenibilità del codice sorgente e consente una scalabilità orizzontale indipendente per ciascun modulo operativo [7]. La piattaforma è stata ingegnerizzata per elaborare flussi di dati multimodali in tempo reale, integrando l'Intelligenza Artificiale direttamente nel flusso di lavoro clinico odontoiatrico. Il sistema gestisce in modo sinergico l'acquisizione e la decodifica del segnale vocale, l'analisi di immagini radiografiche come OPT e l'astrazione semantica di documentazione clinica testuale destrutturata. L'ecosistema distribuito si articola su una rigorosa architettura a microservizi. L'infrastruttura bilancia dinamicamente l'elaborazione su risorse hardware locali (Edge Computing), necessaria per la rigorosa tutela della privacy del paziente, con l'interrogazione di servizi cloud remoti ad altissime prestazioni per i compiti di ragionamento clinico complesso e generalizzazione astratta.

A livello strutturale, la dicotomia tra livello di presentazione e il livello di elaborazione permette di mantenere interfacce fortemente reattive anche durante l'esecuzione di task computazionalmente onerosi (come l'inferenza di modelli trasformativi per la Computer Vision). Questa marcata modularità facilita l'integrazione di approcci basati su Continuous Integration e Continuous Delivery/Deployment (CI/CD), il tempestivo aggiornamento dei Knowledge Graph e dei database scientifici, mantenendo il supporto diagnostico costantemente aderente alle linee guida parodontali presenti in letteratura [3].

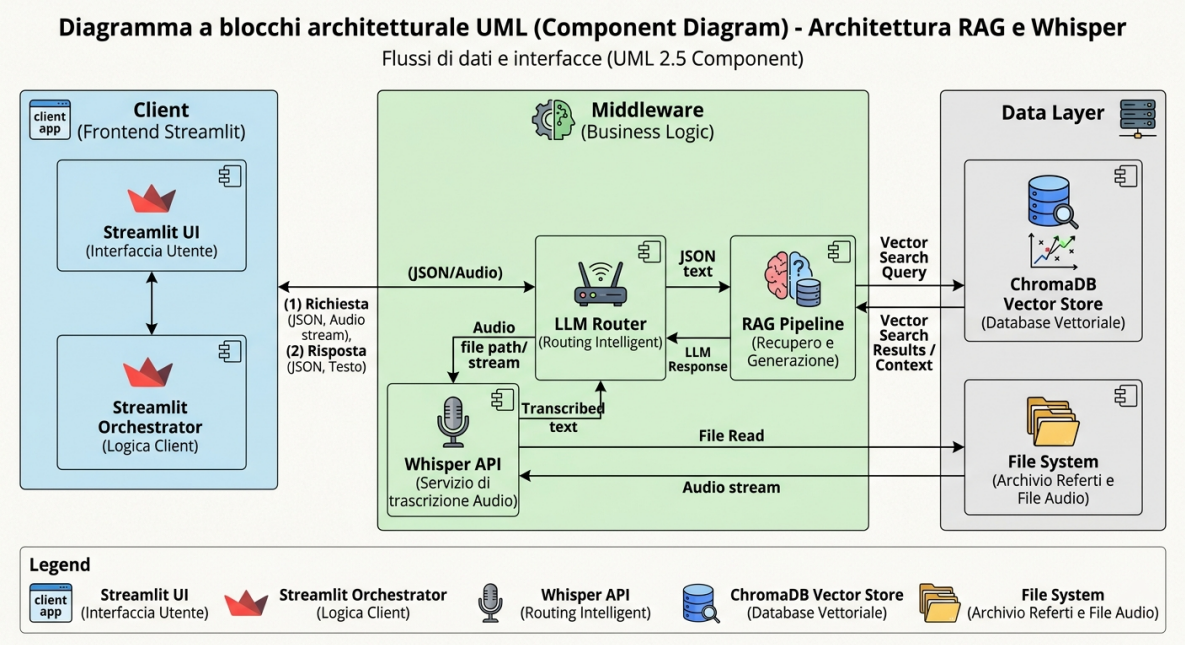

Figura 2.1: Diagramma delle componenti della piattaforma

2.1 Funzionalità principali e flussi informativi

Il livello di presentazione del sistema è stato interamente sviluppato impiegando il framework Streamlit[1]. Sebbene originariamente concepito per la prototipazione rapida in ambito *Data Science*, in questo progetto il framework è stato spinto verso architetture di livello enterprise, fungendo da vero e proprio orchestratore reattivo *event-driven*. Questo modulo incapsula la complessità degli script di inferenza neurale dietro un'interfaccia utente fluida, progettata per operare in contesti hands-free all'interno delle sale odontoiatriche.

Dal punto di vista dell'ingegneria del software, la sfida principale nell'utilizzo di architetture web stateless per applicazioni mediche risiede nella preservazione del contesto diagnostico. Per ottemperare a questa necessità, la coerenza logico-temporale della sessione viene mantenuta tramite un'implementazione avanzata dello stato *(Session State)*. La struttura dati residente in memoria, istanziata come *ChatMemoryBuffer*, traccia l'intera cronologia della conversazione *(message history)*. Questo meccanismo architetturale permette all'agente conversazionale di risolvere le anafore e contestualizzare i parametri di sondaggio parodontale forniti sequenzialmente dal medico, superando i limiti fisiologici della *context window* dei singoli prompt.

L'infrastruttura è stata ottimizzata per prevenire il blocco del thread principale (UI blocking), un evento critico durante il caricamento di indici vettoriali di grandi dimensio-

[1]https://docs.streamlit.io

ni. È stata implementata una logica di caricamento in background *Background Loading* che utilizza thread separati e primitivi di sincronizzazione come *threading.Lock*. La funzione *_load_index_background* permette di inizializzare l'indice senza interrompere la reattività dell'interfaccia, mentre un sistema di caching globale *_GLOBAL_INDEX_CACHE* garantisce la persistenza dell'indice tra i vari ricaricamenti della pagina, riducendo drasticamente i tempi di attesa dell'utente.

Un ulteriore elemento di ottimizzazione riguarda le strategie di *pre-warming* delle risorse. Subito dopo il caricamento dell'indice, il sistema avvia processi paralleli per caricare le pagine dell'indice *Hierarchical Navigable Small World* (HNSW) nella *page cache* del sistema operativo e per pre-calcolare la cache dei metadati dei documenti *(_prewarm_docs_cache)*. Questo approccio garantisce che la prima interrogazione effettuata dal clinico riceva una risposta istantanea. Le chiamate verso le reti neurali generative avvengono tramite stream di token asincroni: l'interfaccia si aggiorna progressivamente man mano che il modello auto-regressivo produce l'output, garantendo un feedback visivo ininterrotto essenziale per l'accettazione del software in ambito clinico.

2.2 Backend e integrazione dei servizi cloud

Il backend infrastrutturale, responsabile dell'orchestrazione logica e del routing delle chiamate algoritmiche, è stato implementato sfruttando l'ecosistema Python[2]. Il linguaggio rappresenta lo standard industriale de facto per lo sviluppo di architetture di Machine Learning, offrendo un livello di astrazione ideale per l'integrazione di libreria calcolo tensoriale e framework NLP.

Il core logico della piattaforma agisce come un *API Gateway* interno, gestendo payload informativi multimodali. L'infrastruttura di integrazione astrae la complessità dei singoli moduli funzionali, coordinando:

- *Modulo ASR:* Flussi audio grezzi vengono acquisiti e convertiti in array NumPy, per poi essere processati da architetture *Sequence-to-Sequence* come Whisper per la trascrizione del parlato clinico.
- *Modulo di Visione Artificiale*: Matrici di pixel derivanti da OPT vengono processate tramite architetture *Vision Transformer* per la classificazione del rischio parodontale, integrate con moduli di spiegabilità XAI basati su algoritmi SHAP e GradCAM.
- *Pipeline RAG*: Query testuali vengono instradate verso motori di ricerca vettoriale che supportano backend multipli, permettendo all'utente di scegliere tra ChromaDB (per persistenza su disco) e *SimpleVectorStore* (per esecuzioni rapide in RAM.

[2]https://www.python.org/

Le comunicazioni tra i moduli locali e le interfacce di programmazione delle applicazioni *Application Programming Interface* (API) basate su cloud avvengono seguendo il paradigma RESTful. Per garantire la resilienza del sistema in ambienti operativi mission-critical, sono state implementate routine di Fault Tolerance e meccanismi di retry esponenziale per gestire in modo trasparente l'eventuale latenza di rete (network latency) o i limiti di frequenza (rate limiting) imposti dai provider esterni [22]. Un componente chiave è il wrapper *_CachedEmbedding*, che implementa una cache LRU *(Least Recently Used)* per gli embedding delle query. Questo accorgimento evita di richiamare inutilmente i servizi di embedding per domande identiche, riducendo i costi operativi e migliorando la latenza di risposta.

L'architettura supporta una gestione dinamica del provider di intelligenza artificiale. Attraverso l'uso di variabili d'ambiente caricate tramite *dotenv*, la piattaforma può alternare l'uso di modelli remoti ad alte prestazioni con modelli *(On-Premises)*. Questa flessibilità è fondamentale per garantire la sovranità dei dati e la conformità normativa, permettendo di processare dati sensibili in isolamento locale quando richiesto dalle politiche della struttura sanitaria.

2.3 Integrazione di modelli generativi remoti e locali

La natura intrinsecamente sensibile dei dati trattati in ambito medico impone vincoli architetturali severi, rendendo la scelta del provider di intelligenza artificiale una decisione non solo tecnica, ma anche deontologica e legale. Per bilanciare l'esigenza di disporre di capacità di ragionamento di frontiera con l'obbligo di conformità normativa alle normative vigenti sulla protezione dei dati sanitari, la piattaforma adotta una sofisticata topologia ibrida per l'erogazione dei modelli generativi. Il backend infrastrutturale astrae completamente il livello del provider, implementando un'architettura modulare che permette di iniettare dinamicamente le dipendenze per l'inferenza locale *(On-Premises)* o remota (*Cloud-based*). Questa flessibilità è governata centralmente attraverso il modulo di configurazione dell'applicativo, che permette all'utente o all'amministratore di selezionare il motore di inferenza più idoneo per lo specifico task clinico [23]; come illustrato in Figura 2.2. L'integrazione è stata progettata seguendo tre pilastri fondamentali:

1. *Sovranità del dato*: la possibilità di processare informazioni sensibili in isolamento locale, minimizzando il trasferimento verso terze parti e mantenendo il controllo operativo all'interno del dominio tecnico–organizzativo del titolare, in coerenza con i principi di minimizzazione, riservatezza e sicurezza del trattamento dei dati sanitari previsti dalle normative applicabili nel contesto giurisdizionale di riferimento.

2. *Eccellenza nel ragionamento*: l'accesso ai modelli più avanzati del settore per compiti di sintesi e analisi scientifica complessa.

3. *Resilienza operativa*: la capacità del sistema di operare anche in condizioni di connettività internet limitata o assente, sfruttando le risorse hardware locali.

Per l'esecuzione di compiti analitici ad alta complessità computazionale, come la sintesi di casi clinici eterogenei o l'interrogazione profonda della letteratura scientifica indicizzata, il sistema delega il calcolo al modello GPT-4o attraverso chiamate API verso i server di OpenAI[3]. Questa architettura SaaS *(Software as a Service)* sfrutta cluster di GPU remoti di immensa capacità, dimostrando una precisione superiore rispetto ai chatbot generalisti e raggiungendo un tasso di accuratezza dell'81.16% in test specialistici [11]. Nel backend della piattaforma, tale integrazione è gestita dalla funzione *build_openai_llm*, che configura il modello con parametri stocastici rigidamente controllati: la variabile *Temperature* (che indica il grado di "creatività" del modello) è impostata a 0 per garantire un output deterministico e rigoroso, essenziale per la sicurezza clinica, mentre il parametro *request_timeout* assicura la resilienza del sistema in caso di latenza di rete. Ogni interazione è preceduta da una direttiva di sistema, definita come *_IDENTITY_PROMPT*, che vincola il modello a operare esclusivamente come assistente specializzato in parodontologia e implantologia.

Parallelamente, la piattaforma affronta le criticità inerenti alla protezione dei dati sanitari offrendo una modalità di esecuzione completamente *off-grid*, isolata dalla rete esterna. Tramite l'integrazione del runtime locale Ollama[4], un orchestratore per modelli generativi *open-weight* che espone una REST API compatibile e gestisce il ciclo di vita dei modelli su macchina locale, il sistema è in grado di caricare direttamente nella VRAM dell'hardware locale modelli *Open-Weight*, quali le architetture Llama 3 o varianti DeepSeek quantizzate. L'utilizzo di tecniche di quantizzazione dei pesi, tipicamente a 4 o 8 bit, riduce drasticamente l'impronta computazionale in memoria senza compromettere eccessivamente le capacità di ragionamento, consentendo l'inferenza su normali workstation cliniche. La funzione *build_ollama_llm* implementa un controllo preventivo della connettività al server locale e configura una finestra di contesto specifica *(num_ctx)* pari a 8192 token, bilanciando la memoria a breve termine del modello con le risorse hardware disponibili. Questa modalità garantisce che l'elaborazione dei dati coperti da segreto medico, come anagrafiche e cartelle cliniche, non abbandoni mai il perimetro di sicurezza cibernetica dello studio odontoiatrico [23].

Questa architettura ibrida permette un routing intelligente: il sistema può inviare richieste anonimizzate riguardanti la letteratura scientifica globale ai modelli cloud per ottenere le risposte più aggiornate, mentre le attività che coinvolgono dati identificativi del paziente vengono gestite interamente dai modelli locali [24].La versatilità di Ollama, che espone una REST API compatibile, facilita questa commutazione dinamica tramite una logica di factory

[3]https://openai.com
[4]`https://ollama.com`

nel codice, permettendo al clinico di selezionare il motore di inferenza più idoneo tramite l'interfaccia utente senza richiedere modifiche al backend. Inoltre, il sistema implementa una gestione trasparente dello streaming dei token, permettendo all'utente di visualizzare la risposta man mano che viene generata, migliorando l'usabilità complessiva della piattaforma in scenari clinici in tempo reale.

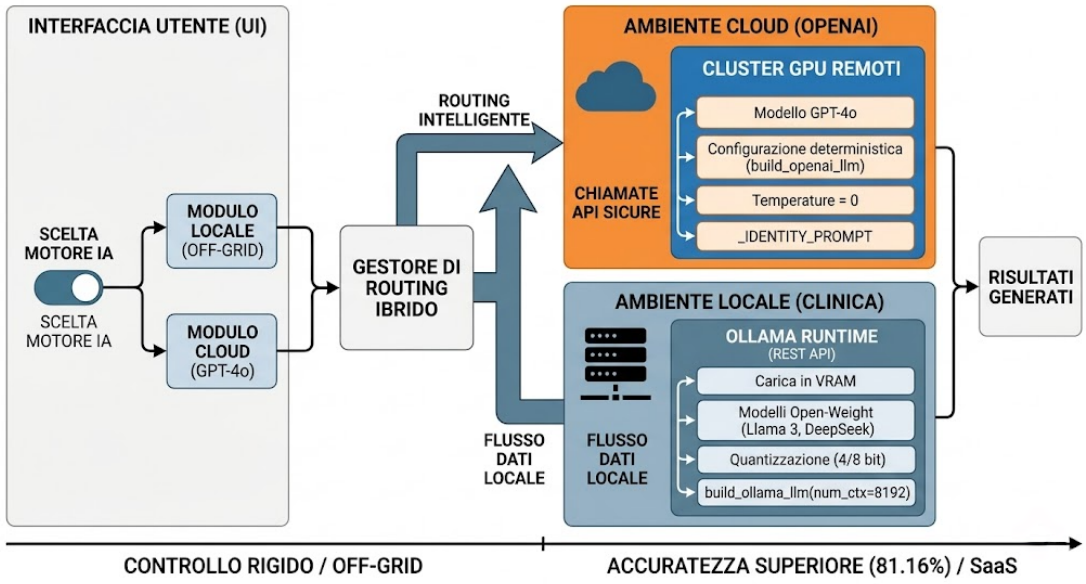

Figura 2.2: Gestione dei flussi informativi nella piattaforma

Il fulcro decisionale evidence-based della piattaforma proposta è costituito dalla sua infrastruttura RAG, governata tramite il framework avanzato LlamaIndex[5]. Questa pipeline inverte il classico paradigma generativo: anziché affidarsi alla conoscenza parametrica congelata nei pesi della rete neurale al momento del pre-training, l'LLM viene vincolato sintatticamente a rispondere esclusivamente sul contesto documentale iniettato dinamicamente.

Il motore di Information Retrieval è stato implementato istanziando ChromaDB[6], un *Vector Database* progettato per l'ottimizzazione delle query ad alta dimensionalità. La letteratura clinica (linee guida, revisioni sistematiche, ecc.) subisce una fase di caricamento *(ingestion)*. Il testo grezzo viene segmentato in unità semantiche *(chunking)* tramite algoritmi di partizionamento gerarchico che preservano l'integrità del contesto (ad esempio evitando la troncatura di tabelle a cavallo di due pagine). Successivamente, modelli di embeddings (come *nomic-embed-text* in esecuzione locale) mappano ciascun frammento in un vettore denso a moltiplicità N-dimensionale.

L'indicizzazione all'interno di ChromaDB avviene tramite l'impiego di grafi HNSW, che permettono una ricerca approssimata dei vicini *Approximate Nearest Neighbor* computazio-

[5]https://docs.llamaindex.ai/
[6]https://docs.trychroma.com/

nalmente efficiente, valutando la metrica della similarità coseno tra il vettore della query del medico e l'intero spazio documentale [7].

Tuttavia per superare le sfide poste dal linguaggio medico specialistico, PerioGPT implementa tecniche di Advanced RAG tramite l'adozione di un *Hybrid Retrieval* che affianca alla ricerca densa algoritmi statistici basati sulle occorrenze lessicali esatte *(Sparse Retrieval, come BM25).* I risultati contrastanti dei due motori di ricerca vengono successivamente fusi matematicamente tramite algoritmi di *Reciprocal Rank Fusion* (RRF). L'aggiunta in pipeline di un ulteriore nodo di post-elaborazione basato su *Cross-Encoder* valuta la correlazione esatta tra i frammenti fusi e la domanda clinica originaria, grazie al processo di *Re-ranking*, filtrando il rumore e massimizzando la precisione del contesto testuale fornito all'LLM. Questa ingegnerizzazione algoritmica garantisce tassi di allucinazione tendenti allo zero, un requisito non negoziabile in ambito medico.

2.4 Acquisizione audio e trascrizione automatica

La transizione digitale in ambito odontoiatrico, specialmente nella parodontologia clinica, è storicamente molto attenta alla stringente necessità di mantenere l'assoluta sterilità del campo operatorio. Durante la compilazione della cartella parodontale clinica, il professionista è costretto a interrompere continuamente l'ispezione per annotare i dati manualmente, o in alternativa necessita della presenza costante di un assistente, incrementando il carico cognitivo. Per risolvere tale criticità strutturale, la piattaforma PerioGPT è stata ingegnerizzata con un modulo integrato per l'interazione *hands-free.* Il sistema propone un'architettura basata su microservizi che trasla il paradigma di data entry dall'input fisico (tastiera o schermo tattile) all'input vocale [25]. La pipeline algoritmica progettata si articola in tre macro-fasi sequenziali strettamente accoppiate e rappresentata in Figura 2.3:

1. *Acquisizione e pre-elaborazione del segnale vocale*: cattura del flusso audio analogico, digitalizzazione e filtraggio del rumore e campionamento ottimizzato.

2. *Trascrizione*: inferenza tramite modelli di Automatic Speech Recognition basati su architetture a trasformatori per la decodifica dell'onda acustica in sequenze testuali.

3. *Astrazione semantica strutturata*: impiego di reti neurali per il Natural Language Processing al fine di mappare il testo destrutturato in vettori o oggetti JSON *(JavaScript Object Notation)* rigidamente tipizzati, direttamente integrabili nel frontend clinico.

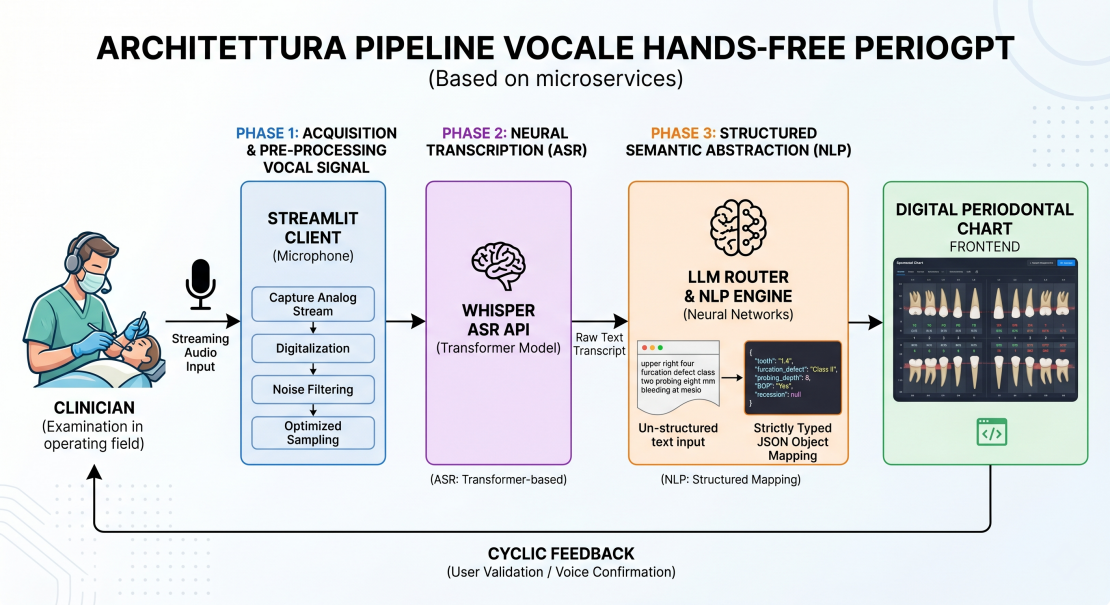

Figura 2.3: Pipeline per la trascrizione voice-to-text

2.4.1 Integrazione del modello per il riconoscimento vocale

La fase di conversione dell'input vocale in testo grezzo è demandata a Whisper, un modello di reti neurali di tipo fondazionale *(Foundation Model)* sviluppato per il riconoscimento vocale[7]. Da un punto di vista dell'ingegneria architetturale, Whisper è strutturato secondo un paradigma Encoder-Decoder basato su architettura Transformer Sequence-to-Sequence.

Il flusso audio in ingresso, acquisito tramite l'interfaccia reattiva in Python, viene preliminarmente ricampionato a una frequenza di 16.000 Hz. Successivamente, il segnale monodimensionale viene trasformato nel dominio del tempo-frequenza calcolando uno spettrogramma di Mel a 80 canali. L'elaborazione di questo spettrogramma viene iniettata nel blocco Encoder, costituito da layer convoluzionali seguiti da blocchi di attenzione multi-testa *(Multi-Head Self-Attention)*. Questa topologia matematica consente al modello di isolare efficacemente le feature acustiche rilevanti. Il blocco Decoder auto-regressivo converte poi le rappresentazioni latenti in una sequenza di token testuali.

L'adozione di tale architettura risolve due problematiche ingegneristiche fondamentali nell'ecosistema odontoiatrico:

- *Robustezza al rumore di fondo (Noise Robustness)*: Gli studi clinici sono caratterizzati da elevati livelli di inquinamento acustico causati da strumentazione meccanica (es. micromotori, turbine ad alta velocità, aspiratori chirurgici). I meccanismi di attenzione di Whisper permettono alla rete di assegnare pesi inferiori ai pattern di rumore bianco o meccanico, focalizzando il calcolo tensoriale sulle formanti della voce umana.

[7]https://huggingface.co/docs/transformers/en/model_doc/whisper

- *Generalizzazione sul dominio specifico (Domain Shift)*: Il vocabolario parodontale è denso di terminologia iper-specialistica e riferimenti topografici atipici (es. "mesio-vestibolare", "sondaggio", "sanguinamento", "recessione"). Il pre-addestramento multilingua e massivo del modello garantisce tassi di Word Error Rate *(WER)* inferiori rispetto alle architetture acustiche tradizionali (come gli *Hidden Markov Models*), senza richiedere alcun fine-tuning addizionale sui dati clinici locali [26].

A livello infrastrutturale, per minimizzare la latenza di trasduzione ed evitare fenomeni di bottleneck computazionale sull'hardware locale, il sistema inoltra i frammenti audio compressi (es. formato *.wav* o *.webm*) verso gli endpoint RESTful di OpenAI. Questo approccio basato su cloud computing riduce il tempo di inferenza a poche frazioni di secondo, abilitando una dettatura in tempo reale *(Real-Time Dictation)* percepita come istantanea dall'operatore sanitario.

2.4.2 Elaborazione NLP per l'estrazione e tipizzazione dei dati

La mera trascrizione fonetica dell'audio clinico produce una stringa di testo lineare e destrutturata *(es. "inserisci dente uno sei profondità tasca cinque millimetri con sanguinamento e placca")*. Tale formato risulta inutilizzabile per un database relazionale o per la generazione dinamica di un'interfaccia clinica vettoriale. Pertanto, la piattaforma implementa uno stadio intermedio di post-elaborazione algoritmica basato sul Natural Language Processing avanzato.

Il testo generato dall'ASR viene inoltrato a un Large Language Model, specificamente istanziato con un System Prompt restrittivo. [27]. L'obiettivo ingegneristico di questo modulo è eseguire due task simultanei di estrazione dell'informazione *(Information Extraction)*:

1. *Named Entity Recognition (NER)*: Identificazione e classificazione di entità biomediche specifiche all'interno della stringa trascritta.

2. *Relation Extraction (RE)*: Creazione di associazioni logico-matematiche tra le entità individuate.

Per garantire che la rete neurale produca un output deterministicamente interfacciabile con l'architettura logica del software, il modello viene forzato a operare in modalità strutturata (JSON Mode). Il prompting algoritmico vincola l'LLM a mappare le entità estratte seguendo uno schema di serializzazione predefinito. Il sistema identifica le seguenti variabili cliniche:

- *Identificativo topografico*: Il numero dell'elemento dentario secondo la nomenclatura internazionale FDI *(FDI World Dental Federation notation)*, convertendo locuzioni come "dente quattro punto cinque" nell'intero 45.

- *Profondità di sondaggio (Probing Pocket Depth, PPD)*: Estrazione dei valori numerici in millimetri e mappatura sui siti specifici del dente (es. disto-vestibolare, centro-vestibolare, mesio-vestibolare).

- *Indici di infiammazione (Bleeding on Probing/Plaque Index, BOP/PI)*: Riconoscimento di parametri booleani. Termini come "sanguina" o "presenza di placca" vengono trasformati in vettori logici true per le chiavi Bleeding on Probing e Plaque Index.

- *Recessione gengivale (REC) e Forcazioni (FURC)*: Mappatura dei valori numerici di retrazione tissutale e del grado di compromissione delle biforcazioni radicolari.

Questo processo di astrazione semantica agisce come un parser intelligente. Una volta generato, l'oggetto JSON viene trasmesso asincronamente al modulo di orchestrazione grafico. Il livello di frontend intercetta il payload strutturato e, tramite logiche di state management interne, aggiorna dinamicamente i componenti visivi (Odontogramma vettoriale) sulla cartella clinica digitale (Figura 2.4). Tale approccio annulla l'errore umano di trascrizione asincrona e permette al medico di focalizzarsi interamente sulla morfologia dei tessuti molli, garantendo una fluida e sicura integrazione tra Artificial Intelligence generativa e strumentazione diagnostica convenzionale.

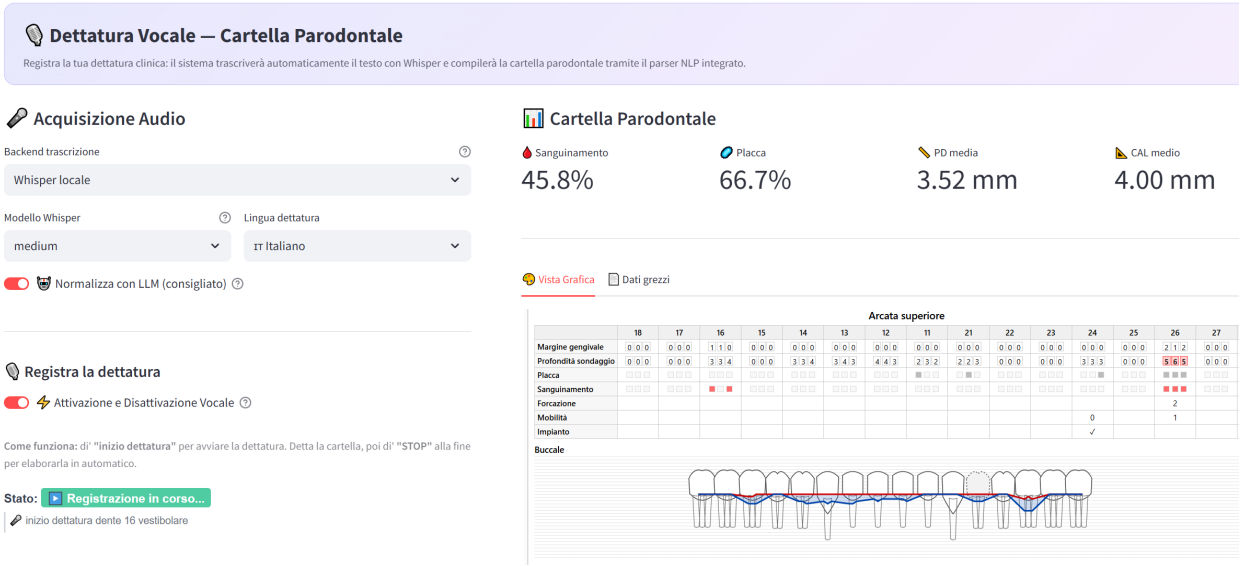

Figura 2.4: Modulo di dettatura della cartella parodontale

2.5 Visualizzazione delle spiegazioni a supporto del medico

L'integrazione sistematica di modelli di *Artificial Intelligence* nei flussi di lavoro clinici odontoiatrici introduce sfide epistemologiche complesse, strettamente legate alla trasparenza e all'interpretabilità dei processi decisionali algoritmici. I sistemi di supporto clinico d'avanguardia, basati su tecniche di *Deep Learning*, sono tipicamente classificati dalla comunità informatica come architetture "Black Box". Tale denominazione deriva dall'elevata complessità topologica delle reti, caratterizzate da milioni di parametri e dalla natura non lineare delle trasformazioni applicate ai dati durante i passaggi attraverso i *layer* nascosti.

In ambito medico e chirurgico, l'incapacità strutturale di interpretare la logica interna di una rete neurale stratificata rappresenta un ostacolo critico all'adozione della tecnologia, minando la fiducia del clinico e compromettendo potenzialmente la sicurezza del paziente. Questa opacità computazionale risulta dogmaticamente inaccettabile per la pratica odontoiatrica *evidence-based*, dove ogni singola decisione diagnostica o terapeutica deve essere necessariamente ancorata a evidenze cliniche verificabili, tracciabili e scientificamente giustificabili.

Oltre alle implicazioni etico-professionali, l'esigenza di trasparenza risponde a requisiti normativi stringenti definiti dalla giurisprudenza europea tramite il *General Data Protection Regulation* (GDPR). In particolare, l'articolo 22 del suddetto regolamento istituisce il cosiddetto "diritto alla spiegazione", sancendo la prerogativa del paziente di ricevere delucidazioni significative, comprensibili e logiche in merito ai trattamenti informatici automatizzati che producono effetti giuridici o clinici sulla sua persona. Un sistema di supporto decisionale non conforme a tali principi andrebbe incontro all'esclusione dai contesti clinici regolamentati.

Per ottemperare a queste necessità legali e cliniche, la piattaforma *PerioGPT* affronta il limite intrinseco dei modelli complessi attraverso l'implementazione sistematica di moduli di *Explainable Artificial Intelligence* (XAI), schematizzati in Figura 2.5. Il sistema ingegnerizzato supera il problema dell'opacità algoritmica restituendo un feedback analitico e visivo direttamente nell'interfaccia utente grafica. Questo approccio permette all'odontoiatra di calibrare la propria fiducia nei confronti della macchina, trasformando il modello neurale da un semplice oracolo inconfutabile a uno strumento collaborativo e interattivo per il supporto diagnostico avanzato.

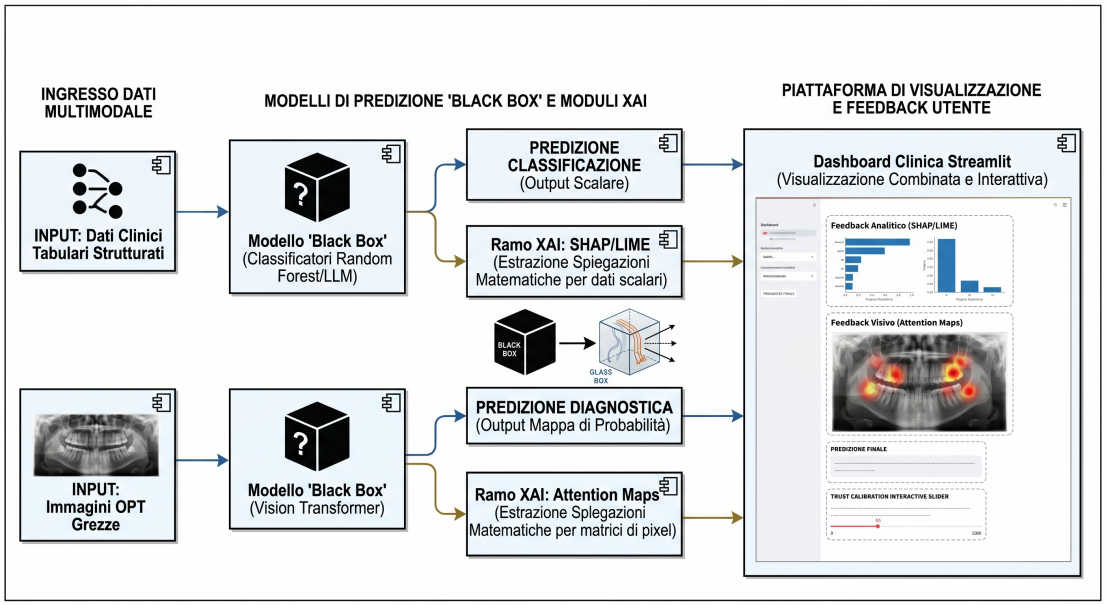

Figura 2.5: Workflow di XAI integrato nella piattaforma

2.5.1 Algoritmi di spiegabilità post-hoc

Per decodificare e interpretare le predizioni effettuate sui parametri clinici scalari estratti dalla cartella parodontale (quali la profondità di sondaggio, l'indice di sanguinamento, l'età anagrafica e le abitudini sistemiche come il tabagismo), il sistema implementa metodi computazionali di spiegabilità *Post-Hoc* e *Model-Agnostic*. Queste due definizioni indicano che gli algoritmi intervengono in una fase successiva all'addestramento e all'inferenza del modello principale, analizzando empiricamente la relazione di dipendenza tra le perturbazioni degli input e le fluttuazioni dell'output, senza richiedere alcun accesso alla topologia interna, ai gradienti o ai pesi della rete neurale analizzata.

Il primo motore di spiegabilità integrato è basato sull'algoritmo *SHapley Additive exPlanations* (SHAP). Tale approccio matematico fonda le sue radici nella teoria dei giochi cooperativi, elaborata inizialmente dall'economista Lloyd Shapley. L'algoritmo concettualizza ogni caratteristica clinica (feature) come un "giocatore" che partecipa a un gioco cooperativo, in cui il premio finale è costituito dal valore della predizione diagnostica. L'obiettivo computazionale di SHAP è assegnare a ciascuna variabile clinica un valore di importanza marginale (il Valore di Shapley) che rispetti rigorosi assiomi matematici di equità, simmetria e additività locale. A livello discorsivo, la formulazione calcola il contributo di una specifica variabile effettuando una media ponderata dei suoi contributi marginali su tutte le possibili permutazioni e combinazioni di sottoinsiemi di feature. Questo meccanismo esplorativo garantisce che il peso assegnato a fattori interagenti (ad esempio, la sinergia deleteria

tra fumo e indice di placca elevato nel rischio parodontale) venga ripartito in modo equo e matematicamente inappuntabile.

Complementarmente a SHAP, che eccelle nella coerenza globale ma richiede tempi di computazione elevati per set di dati estesi, il backend impiega l'algoritmo *Local Interpretable Model-agnostic Explanations* (LIME). LIME è stato ingegnerizzato per massimizzare la velocità di interpretazione attraverso la generazione di modelli surrogati lineari locali. Il funzionamento algoritmico prevede che il sistema perturbi deliberatamente i dati clinici del paziente in esame, introducendo rumore statistico per creare un set di campioni sintetici nell'intorno spaziale del dato originale [28]. Successivamente, la "Black Box" viene interrogata su questi campioni perturbati. Su questo intorno locale pesato (utilizzando kernel esponenziali per calcolare la distanza dal dato reale), viene addestrato un modello intrinsecamente interpretabile, come una semplice regressione lineare, che approssima fedelmente il comportamento della rete neurale complessa limitatamente a quella specifica regione dello spazio delle feature. Il risultato finale, esposto al medico tramite grafici a barre bidirezionali, consiste in una mappa di importanza locale dove ogni metrica parodontale è associata a un peso quantitativo che ne palesa l'influenza positiva o negativa rispetto all'esito diagnostico.

2.5.2 Visualizzazione di mappe di salienza tramite Vision Transformer

La decodifica visiva automatizzata delle indagini radiologiche bidimensionali, come le ortopantomografie (OPT), rappresenta il dominio applicativo della Computer Vision all'interno della piattaforma. L'analisi è affidata a un'architettura di frontiera denominata Data-efficient Image Transformers *(DeiT3)*. Questa rete rappresenta un superamento concettuale delle tradizionali CNN. A differenza delle CNN, che processano l'immagine applicando filtri locali per estrarre feature gerarchiche limitate da un campo recettivo ristretto, i Vision Transformer *(ViT)* processano l'immagine in modo olistico, adottando un approccio computazionale basato esclusivamente sul meccanismo di Self-Attention.

Dal punto di vista dell'ingegneria dei tensori, il processo di inferenza inizia con il partizionamento della matrice di pixel dell'immagine radiografica in una sequenza rigorosa di piccoli riquadri adiacenti e non sovrapposti, denominati patch (tipicamente di dimensione 16x16 pixel). Attraverso un layer di proiezione lineare, ogni singola patch viene appiattita e proiettata in uno spazio vettoriale continuo ad alta dimensionalità, assumendo il ruolo di token. Per preservare l'informazione topografica della radiografia, ai token viene sommato un Positional Encoding (codifica posizionale), dopodiché la sequenza viene elaborata analogamente a come i modelli linguistici processano le stringhe di testo.

Il nucleo tecnologico dell'architettura DeiT3 è rappresentato dai blocchi di Multi-Head Self-Attention. Durante il forward pass, il modello calcola matematicamente il grado di

correlazione visiva (i punteggi di attenzione) tra ogni patch dell'arcata dentaria e tutte le restanti patch dell'immagine, moltiplicando tensori denominati Query, Key e Value. Questo meccanismo permette alla rete di stabilire dipendenze a lungo raggio: ad esempio, il modello può correlare un difetto osseo in un quadrante con la simmetria anatomica del quadrante opposto.

Per rendere spiegabile questa decisione, il sistema estrae la matrice dei pesi di attenzione derivante dall'ultimo blocco di astrazione del trasformatore, o applica algoritmi avanzati come il Gradient-Driven Multi-Head Attention Rollout (GMAR). Da questi valori tensoriali vengono generate delle Mappe di Salienza (Saliency Maps), comunemente note in ambito radiologico come mappe di calore (Heatmaps).

Sfruttando le primitive grafiche della libreria OpenCV, il backend normalizza queste matrici e le sovrappone in semitrasparenza (tramite fusione alpha-blending) all'immagine radiografica originale visualizzata a schermo. Le aree anatomiche che hanno condotto la rete neurale a formulare il sospetto diagnostico (come zone di radiotrasparenza indicanti riassorbimento osseo verticale, o spigoli radiopachi suggestivi di depositi di tartaro sottogengivale) vengono "illuminate" con gradienti di colori caldi. Questo meccanismo di Explainable AI puramente visivo permette all'odontoiatra di validare in frazioni di secondo la deduzione algoritmica, orientando lo sguardo umano direttamente verso le regioni critiche. L'impiego dei Vision Transformer, integrati con mappe di salienza precise, abbatte il rischio di falsi negativi dovuti alla fatica percettiva dell'operatore e fornisce un formidabile strumento di comunicazione visiva per la compliance del paziente.

Capitolo 3

Valutazione del sistema in scenari simulati

La valutazione prestazionale del sistema prescinde dalla specificità di un singolo esame odontoiatrico, adottando invece una metodologia di collaudo trasversale. I test sono stati ingegnerizzati per simulare un'ampia eterogeneità di richieste: dall'interrogazione profonda di linee guida cliniche e protocolli terapeutici (RAG intensivo), fino all'estrazione strutturata di parametri diagnostici tramite input vocale destrutturato.

L'ambiente di test è stato standardizzato per riflettere i vincoli hardware reali di uno studio dentistico medio. I collaudi sono stati eseguiti su una workstation equipaggiata con una CPU a 6 core, 32 GB di RAM di sistema e una GPU NVIDIA GeForce GTX 1050 Ti dotata di 4 GB di VRAM. Questa specifica limitazione hardware (4 GB di memoria video) rappresenta uno stress-test fondamentale: essa impone vincoli stringenti sull'impronta di memoria dei modelli locali, permettendo di valutare scientificamente la sostenibilità operativa dell'approccio On-Premises quantizzato rispetto ai servizi Cloud-Native.

3.1 Analisi comparativa dei modelli on-premises

In continuità con l'impostazione metodologica descritta in precedenza, la seguente sezione analizza comparativamente le prestazioni dei modelli linguistici locali integrati sulla piattaforma PerioGPT, con l'obiettivo di valutare la sostenibilità di un approccio interamente On-Premises in uno studio odontoiatrico reale. L'analisi si basa su un insieme di benchmark dedicati, eseguiti su un set di dieci task clinici parodontali rappresentativi dei casi d'uso principali della piattaforma.

Dal punto di vista architetturale, la piattaforma è stata configurata in modalità ibrida ma, ai fini del benchmark, è stato forzato l'utilizzo dei soli modelli serviti tramite *Ollama* in esecuzione locale, disattivando temporaneamente i provider remoti. In questo modo è stato

possibile isolare l'impatto prestazionale degli LLM On-Premises all'interno della pipeline applicativa, mantenendo invariati i parametri di generazione (temperatura, massimo numero di token, finestra di contesto) per tutte le combinazioni modello–task.

Sono stati confrontati tre modelli linguistici open-weight quantizzati, distribuiti tramite *Ollama* e selezionati in base alla loro compatibilità con l'hardware disponibile e al diverso profilo di compromesso tra velocità e capacità di ragionamento:

- `llama3.2 3B`: modello compatto della famiglia *Llama 3*, ottimizzato per l'esecuzione su GPU con VRAM limitata.
- `deepseek-r1 1.5B`: modello di ragionamento a bassa dimensione, progettato per massimizzare throughput e capacità logica su hardware modesto.
- `qwen2.5 3B`: modello general-purpose bilanciato, selezionato per verificare la capacità di mantenere un'elevata aderenza a vincoli di output strutturato pur rimanendo eseguibile in locale.

Il set di benchmark comprende dieci task clinici (T01–T10), definiti per coprire trasversalmente i moduli chiave della piattaforma proposta:

- *T01 - Estrazione JSON sondaggio parodontale* (`JSON`): conversione della descrizione del sondaggio in struttura JSON tipizzata per l'odontogramma.
- *T02 - Classificazione diagnosi parodontale* (`JSON`): mappatura automatica dei parametri clinici su una diagnosi parodontale codificata secondo linee guida.
- *T03 - Piano di trattamento Step-by-Step* (`JSON`): generazione di un piano terapeutico strutturato in fasi, con output serializzato.
- *T04 - Riassunto referto clinico* (testo): sintesi in linguaggio naturale di un referto clinico complesso.
- *T05 - Estrazione multi-dente da dettatura* (`JSON`): parsing di una dettatura libera con più elementi dentari e relativi parametri associati.
- *T06 - Domanda RAG su parodontologia* (testo + RAG): interrogazione della base documentale in *ChromaDB* con generazione di risposta ancorata alla letteratura.
- *T07 - Interpretazione indici di igiene orale* (`JSON`): strutturazione di indici parodontali in formato computabile.
- *T08 - Valutazione rischio impianti* (`JSON`): valutazione strutturata del rischio implantare a partire da anamnesi e parametri clinici.

- *T09 - Spiegazione per il paziente* (testo): riformulazione empatica e comprensibile del caso clinico.
- *T10 - Estrazione farmaci da anamnesi* (`JSON`): estrazione di principi attivi e posologie da un'anamnesi farmacologica libera.

La distinzione tra task con output strutturato (`JSON`) e task con output testuale libero è cruciale, poiché consente di valutare sia la capacità di ragionamento semantico sia la disciplina del modello nel rispettare uno schema formale di serializzazione, requisito fondamentale per l'integrazione con la cartella clinica digitale.

Per ogni combinazione modello–task sono state raccolte le seguenti metriche:

- *Time To First Token (TTFT)*: tempo, in secondi, tra l'invio della richiesta e la generazione del primo token, percepito dall'utente come latenza iniziale.
- *Tempo totale di generazione*: durata complessiva dell'inferenza fino alla produzione dell'ultimo token.
- *Throughput (Tokens Per Second, TPS)*: rapporto tra numero di token generati e tempo totale, indicativo della velocità di streaming.
- *JSON Accuracy*: percentuale di run in cui, per i task `JSON`, la risposta del modello è sintatticamente valida e aderente allo schema previsto.
- *Errori e run completati*: numero di errori di sistema o fallimenti di inferenza, utile per valutare la robustezza operativa.

La Tabella 3.1 riporta le metriche medie per ciascun modello.

Tabella 3.1: Riepilogo delle metriche di valutazione per ciascun LLM locale.

Modello	TTFT medio [s]	Tempo tot. medio [s]	TPS medio	JSON Acc. [%]	Errori	Run OK
`deepseek-r1 1.5B`	10.849	14.58	43.2	29	0	10
`llama3.2 3B`	7.351	19.65	19.2	71	0	10
`qwen2.5 3B`	7.313	20.11	23.2	100	0	10

Dai dati emerge come `deepseek-r1 1.5B` privilegi in modo aggressivo la velocità di generazione: a fronte di un TTFT medio più elevato, il modello presenta il tempo totale medio di inferenza più contenuto e un throughput medio pari a 43.2 token/s, ma una JSON Accuracy limitata al 29%. Al contrario, `qwen2.5 3B` garantisce una JSON Accuracy del 100%, con tutti i task strutturati completati in formato valido, a fronte di tempi medi di TTFT e di generazione comparabili a quelli degli altri due modelli.

Il modello `llama3.2 3B` si colloca in una posizione intermedia: TTFT medio leggermente migliore di `deepseek-r1` e allineato a `qwen2.5`, throughput inferiore, ma una JSON

Accuracy pari al 71 %, che implica comunque una quota non trascurabile di risposte non immediatamente utilizzabili nei task strutturati. In un flusso *hands-free* in cui il JSON alimenta direttamente la cartella parodontale, la necessità di rigenerare o correggere manualmente gli output invalidi ha un impatto diretto sul carico operativo del clinico.

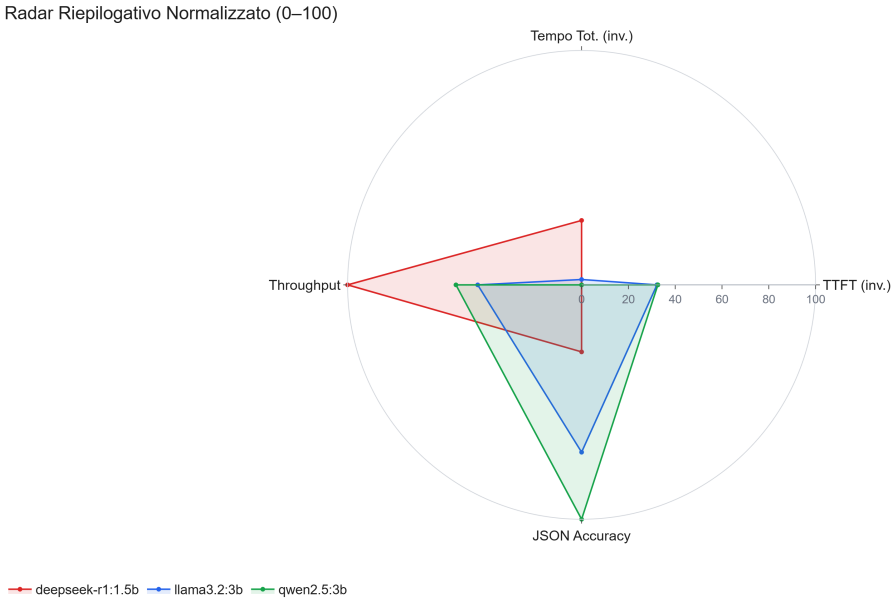

Figura 3.1: Confronto tra i tre LLM locali

Nel grafico radar di Figura 3.1, `deepseek-r1 1.5B` massimizza throughput e tempo totale inverso, ma collassa sull'asse dell'accuratezza JSON; `llama3.2 3B` descrive un profilo più bilanciato ma con JSON Accuracy inferiore a `qwen2.5`; `qwen2.5 3B` occupa una posizione di equilibrio, con un poligono più regolare e un vantaggio netto sull'accuratezza JSON, pur mantenendo tempi di risposta comparabili.

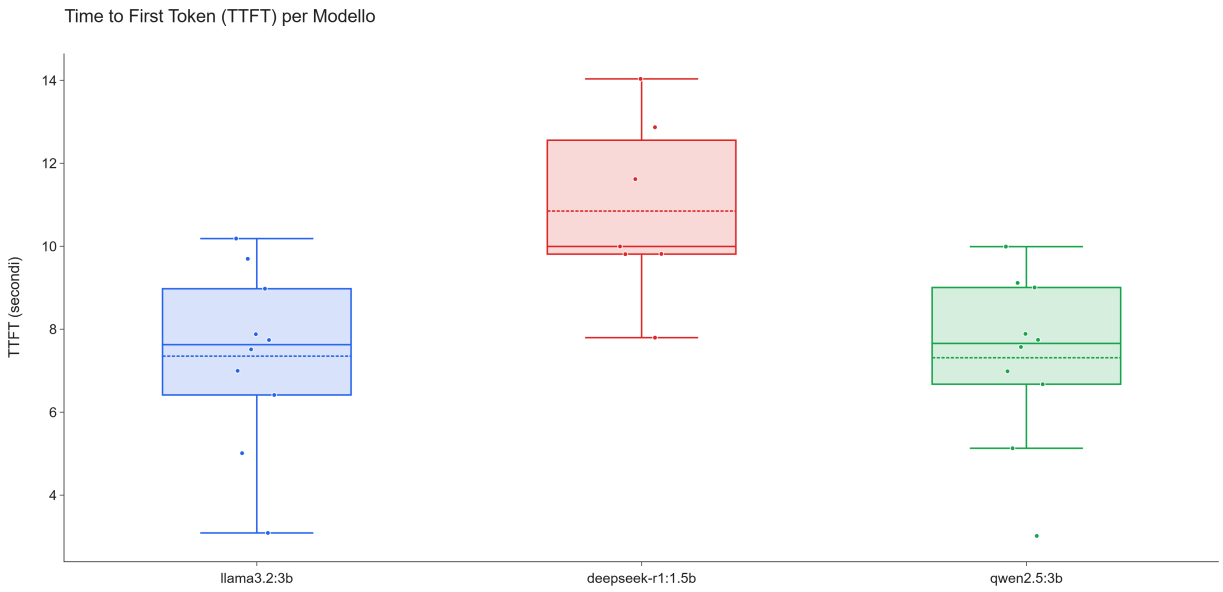

Figura 3.2: Confronto TTFT per LLM locale

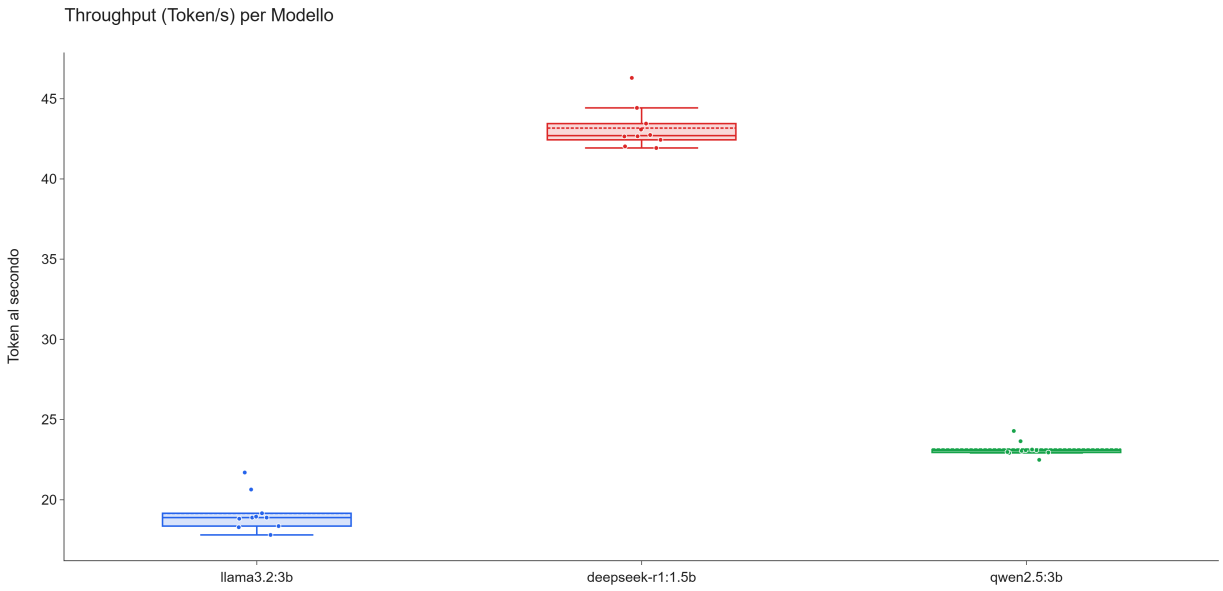

Figura 3.3: Confronto TPS per LLM locale

Il box plot del TTFT, illustrato in Figura 3.2, evidenzia come `llama3.2 3B` e `qwen2.5 3B` presentino distribuzioni molto simili, con mediane nell'ordine di pochi secondi e variabilità contenuta, mentre `deepseek-r1 1.5B` mostra alcuni valori più elevati, legati all'inizializzazione dell'inferenza per taluni task. Viceversa, il box plot del throughput (Figura 3.3) conferma la netta superiorità di `deepseek-r1 1.5B`, con valori stabilmente superiori ai 40 token/s, rispetto ai circa 19–23 token/s degli altri due modelli.

In termini di esperienza d'uso, ciò significa che, una volta superata la fase iniziale (TTFT), `deepseek-r1` restituisce risposte molto rapidamente, caratteristica utile per referti lunghi o spiegazioni estese al paziente, ma che va bilanciata con la bassa affidabilità nel rispetto del formato strutturato nei task `JSON`.

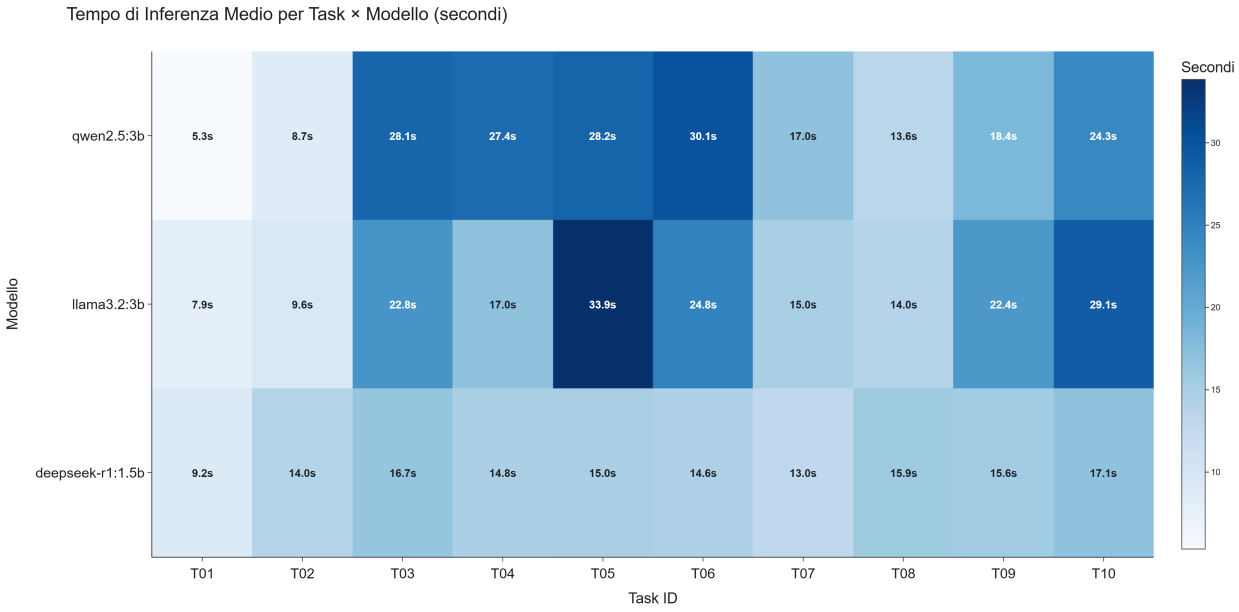

Figura 3.4: Confronto TTFT medio per task clinico e LLM locale

La heatmap di Figura 3.4 mostra come le latenze più elevate si concentrino nei task strutturati più complessi (in particolare T03, T05 e T10), in cui i modelli devono generare JSON articolati a partire da descrizioni cliniche ricche. In tali scenari tutti i modelli sperimentano un aumento dei tempi di elaborazione, ma le differenze relative restano coerenti con quanto osservato nei box plot: `deepseek-r1` tende a TTFT più variabile, mentre `llama3.2` e `qwen2.5` mantengono un comportamento più stabile.

Un aspetto peculiare emerge nei task RAG-intensivi (ad esempio T06): qui la latenza complessiva non dipende solo dal modello, ma anche dal tempo necessario per il retrieval semantico in *ChromaDB* e per le operazioni di re-ranking. In questi casi le differenze tra modelli tendono a ridursi, suggerendo che ulteriori ottimizzazioni della pipeline RAG potrebbero portare benefici più significativi rispetto alla sola sostituzione dell'LLM locale.

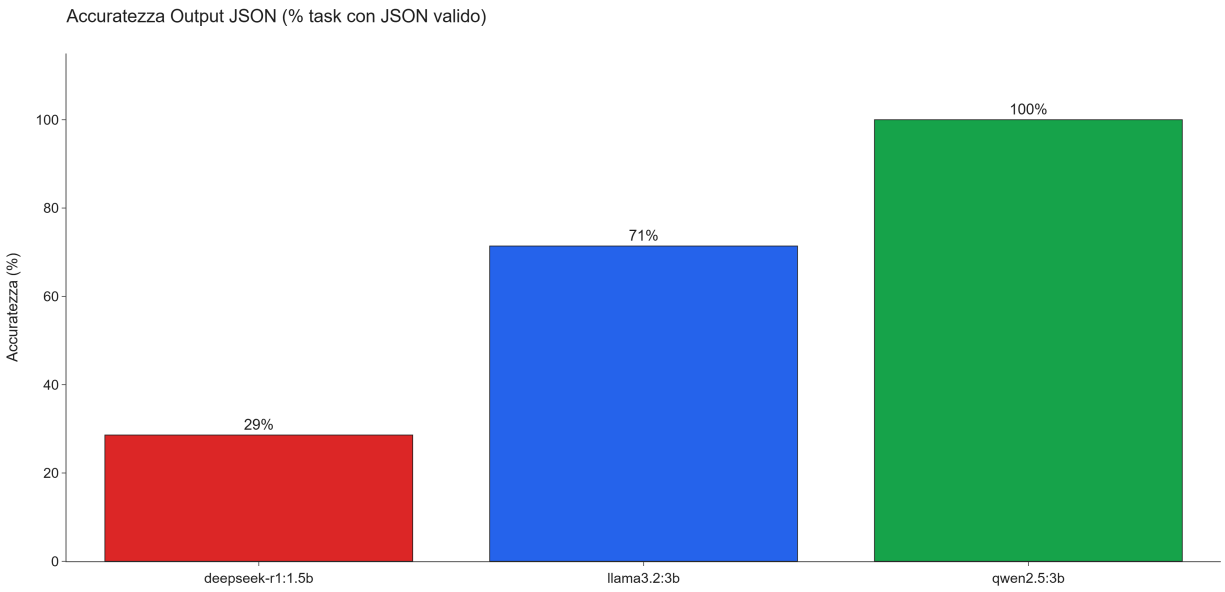

Figura 3.5: Confronto accuratezza per ciascun LLM locale

Come sintetizzato dalla Tabella 3.1 e dalla Figura 3.5, `qwen2.5 3B` raggiunge una JSON Accuracy del 100% nei sette task strutturati, risultando l'unico modello a produrre sempre output direttamente consumabili dal backend della cartella clinica. `llama3.2 3B` si ferma al 71%, mentre `deepseek-r1 1.5B` scende al 29%, generando frequentemente JSON incompleti o non validi che richiedono interventi manuali o rigenerazioni.

In un contesto operativo *hands-free*, in cui il clinico si affida alla dettatura vocale per popolare la cartella in tempo reale, questi risultati rendono evidente come la scelta del modello locale non possa basarsi esclusivamente su metriche di velocità, ma debba privilegiare la robustezza della serializzazione strutturata.

Nel complesso, i risultati dimostrano che, anche con una GPU da 4GB, è possibile eseguire localmente modelli linguistici in grado di supportare sia task di generazione testuale sia task di estrazione strutturata con tempi di risposta compatibili con l'uso in studio. L'adozione di modelli quantizzati, combinata con una pipeline RAG ottimizzata e con il pre-caricamento degli indici vettoriali, permette di mantenere un TTFT medio inferiore a circa 10s e un throughput sufficiente a non introdurre colli di bottiglia percepibili nel processo operativo.

D'altra parte, la forte variabilità dell'accuratezza JSON tra i modelli suggerisce una suddivisione dei ruoli: modelli come `deepseek-r1 1.5B` risultano adatti a task puramente testuali o di consultazione rapida, in cui la struttura del dato non è critica, mentre `qwen2.5 3B` emerge come candidato naturale per tutti i task che alimentano direttamente la cartella parodontale, l'anamnesi farmacologica e i moduli di esportazione verso sistemi informativi sanitari esterni.

In prospettiva, la combinazione tra un'architettura ibrida Cloud/On-Premises, un'infrastruttura RAG avanzata e una selezione mirata degli LLM locali consente alla piattaforma PerioGPT di mantenere un elevato grado di sovranità del dato senza rinunciare alla qualità del supporto decisionale. I benchmark presentati in questa sezione forniscono quindi una base quantitativa per giustificare le scelte architetturali alla base del sistema e per orientare future estensioni, quali l'introduzione di nuovi modelli open-weight e la specializzazione tramite tecniche di distillazione sul dominio parodontale.

3.2 Valutazione del modulo di Information Retrieval

In continuità con l'impostazione adottata per la valutazione degli LLM locali, questa sezione analizza le prestazioni del sottosistema di `Information Retrieval` che alimenta la pipeline RAG della piattaforma PerioGPT, con particolare attenzione alla capacità di recuperare rapidamente documenti clinicamente pertinenti a partire da query specialistiche in parodontologia. L'obiettivo è quantificare il beneficio di strategie di retrieval più sofisticate rispetto a un approccio puramente denso, tenendo conto sia della qualità dei risultati sia della latenza introdotta nel flusso di interrogazione.

La valutazione è stata condotta confrontando due strategie di retrieval applicate alla stessa base documentale clinica indicizzata in `ChromaDB`, mantenendo costanti il contenuto della knowledge base, il numero di documenti restituiti e le query di test.

- *Retriever denso*: indicizzazione dei documenti tramite embedding densi, con ricerca dei vicini più prossimi nello spazio vettoriale (`Dense Vector Store`).
- *Retriever ibrido con reranker*: combinazione di uno strato lessicale tipo BM25 con un livello di retrieval denso e un reranker neurale addestrato specificamente per la similarità testuale biomedica (`dense + BM25 + cross-encoder reranker`).

Entrambe le strategie restituiscono per ogni query i `top-5` frammenti di documento; la valutazione è stata eseguita su dieci query cliniche (Q01-Q10) progettate per coprire casi d'uso tipici (linee guida parodontali, protocolli terapeutici, gestione del rischio implantare, interpretazione di indici clinici). In questa fase l'LLM non interviene: si misura esclusivamente la qualità del recupero e la latenza del componente IR.

Per ciascuna strategia di retrieval sono state calcolate le seguenti metriche aggregate e riassunte in Tabella 3.2:

- *Hit Rate*: proporzione di query per cui almeno uno dei documenti rilevanti è presente nei primi cinque risultati.

- *Mean Reciprocal Rank (MRR)*: media della *reciprocal rank* del primo documento rilevante; misura la posizione media del risultato corretto all'interno della lista.
- *Latenza di retrieval*: tempo medio necessario per eseguire la sola fase di recupero (densa o ibrida), esclusa la generazione LLM.
- *Latenza totale*: tempo medio complessivo della fase IR, comprensivo dell'eventuale reranking neurale.
- *Hits/Query*: numero medio di query per le quali il sistema recupera almeno un documento rilevante nei top-5.

Tabella 3.2: Riepilogo delle metriche di Information Retrieval per le due strategie valutate.

Strategia IR	**Hit Rate**	**MRR**	**Latenza retrieval [s]**	**Latenza totale [s]**	**Hits / Query**
Retriever denso	0.70	0.575	0.0557	0.0557	7 / 10
Retriever ibrido + reranker	0.80	0.620	0.0574	0.2688	8 / 10

Rispetto al retriever puramente denso, la strategia ibrida aumenta l'Hit Rate dal 70% all'80% e l'MRR da 0.575 a 0.620, indicando che i documenti corretti non solo compaiono più frequentemente nei `top-5`, ma tendono anche a posizionarsi più in alto nella lista. La latenza della sola fase di retrieval rimane pressoché invariata (circa 56–57ms), mentre la latenza totale aumenta sensibilmente a causa del reranker neurale, passando da circa 0.056s a circa 0.269s, pur restando al di sotto della soglia percettiva di una consultazione interattiva.

Per visualizzare l'impatto del reranking in termini di qualità del recupero, è possibile rappresentare Hit Rate e MRR tramite un grafico a barre affiancate (Figura 3.6).

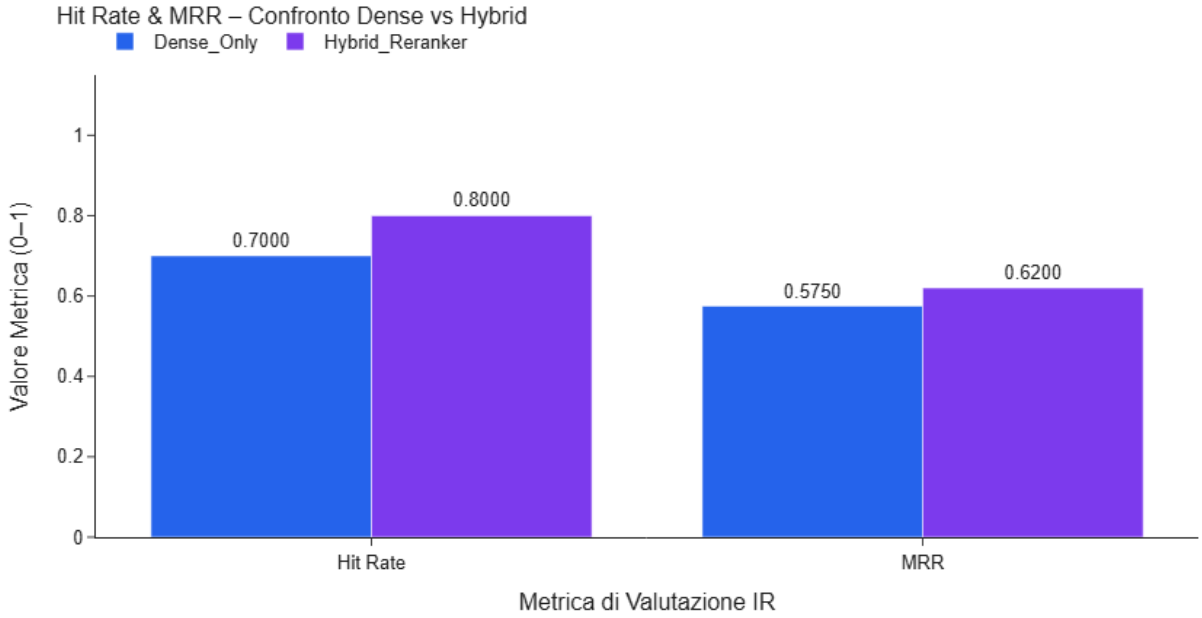

Figura 3.6: Confronto Hit Rate e MRR tra retriever denso e retriever ibrido con reranker

Nel grafico si osserva come la strategia ibrida migliori entrambe le metriche di retrieval, con un incremento di 0.10 in Hit Rate (da 0.70 a 0.80) e di 0.045 in MRR (da 0.575 a 0.620), a parità di base documentale e di numero di documenti restituiti. Questo significa che, in media, il primo documento davvero utile per il clinico si colloca più vicino alla cima della lista dei risultati quando viene applicato il reranking neurale.

Per comprendere meglio come le due strategie si comportino sulle singole query cliniche, è utile esaminare una heatmap di Hit/Miss per query e retriever (Figura 3.7), dove ogni cella indica se tra i `top-5` è presente almeno un documento rilevante.

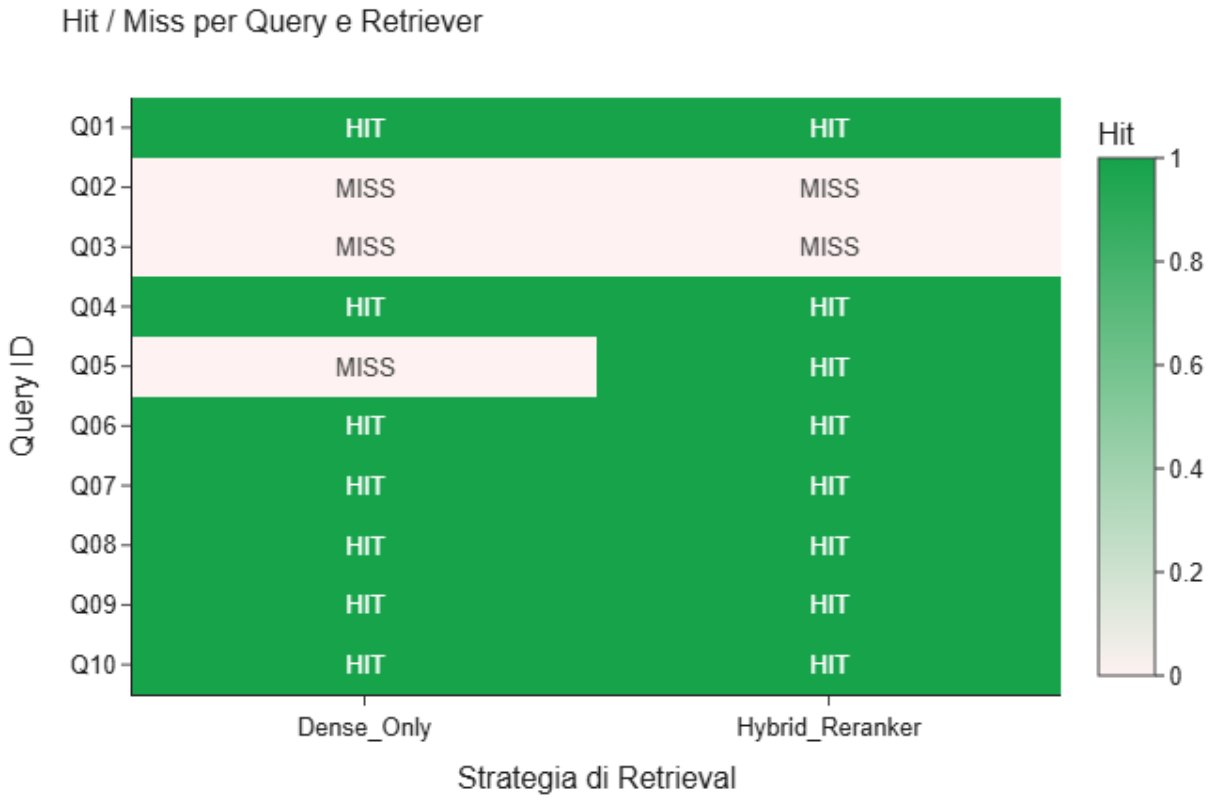

Figura 3.7: Heatmap di Hit/Miss per ciascuna query clinica e strategia di retrieval

La heatmap evidenzia che il retriever denso fallisce il recupero corretto su un sottoinsieme non trascurabile di query, mentre la strategia ibrida riduce il numero di *miss*, trasformando alcune query da fallimento a successo grazie alla combinazione di segnali lessicali e semantici. Questo comportamento è particolarmente rilevante per le domande che utilizzano formulazioni meno canoniche o che si appoggiano a sinonimi tipici del linguaggio clinico quotidiano.

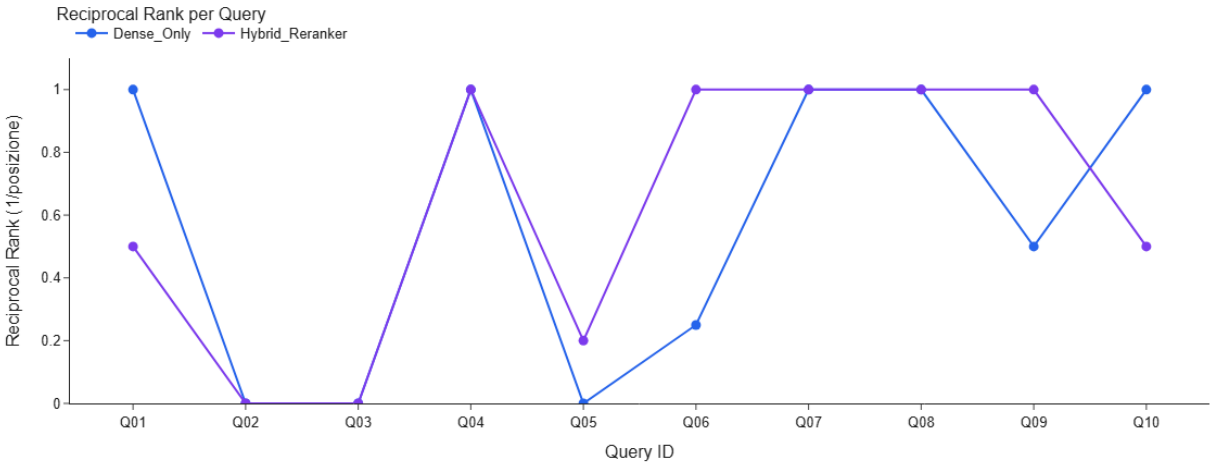

Figura 3.8: Reciprocal Rank per query per retriever denso e retriever ibrido con reranker

Dalla curva di *reciprocal rank*, illustrata in Figura 3.8, emerge che la strategia ibrida migliora marcatamente il posizionamento dei documenti per alcune query difficili, portando il risultato corretto dalla parte bassa della lista ai primissimi posti, mentre su altre query il comportamento dei due retriever rimane sostanzialmente equivalente. In pochi casi si osserva un leggero peggioramento del rank (ad esempio il documento corretto scende dal primo al secondo posto), ma senza impatto pratico significativo per il clinico.

L'impatto del reranking neurale sulla latenza viene analizzato tramite un box plot che confronta la distribuzione dei tempi di risposta per la sola fase di retrieval e per la latenza totale (Figura 3.9).

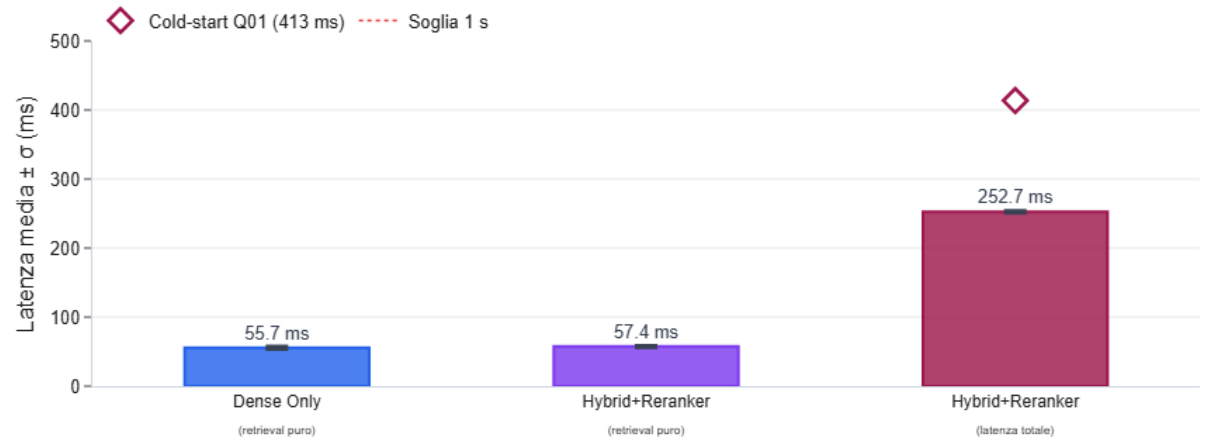

Figura 3.9: Distribuzione della latenza di retrieval e latenza totale per le due strategie di IR

Il grafico a barre mostra che i tempi medi della sola fase di retrieval puro sono quasi sovrapponibili per le due strategie, con valori pari a circa 55.7ms per il retriever denso e 57.4ms per la configurazione `Hybrid+Reranker`, e una deviazione standard contenuta in

entrambi i casi. La barra relativa alla latenza totale della pipeline `Hybrid+Reranker` si attesta invece intorno ai 252.7ms, valore comunque significativamente inferiore alla soglia di 1s indicata nel grafico e quindi compatibile con una consultazione interattiva. Il marcatore a rombo evidenzia il caso di `cold-start` sulla query Q01 (circa 413ms), che rappresenta lo scenario peggiore osservato ma rimane anch'esso ampiamente al di sotto della soglia critica di un secondo.

Dal punto di vista dell'architettura della piattaforma, i risultati suggeriscono che la pipeline RAG dovrebbe adottare la strategia ibrida con reranker come configurazione predefinita per tutte le query cliniche in cui la priorità è massimizzare il recupero di evidenze pertinenti, accettando un moderato aumento della latenza in cambio di una maggiore qualità del contesto fornito all'LLM. In scenari in cui la reattività è estremamente critica e il contenuto informativo richiesto è relativamente semplice (ad esempio consultazioni rapide su definizioni o protocolli molto standardizzati), il retriever denso può comunque rappresentare un'opzione più leggera e sufficiente. In sintesi, la valutazione dell'Information Retrieval conferma che l'investimento in una pipeline IR avanzata (ibrida e dotata di reranking neurale) ha un impatto più marcato sulla riduzione delle allucinazioni e sull'affidabilità delle risposte rispetto alla sola ottimizzazione del modello linguistico, con un overhead di latenza che resta pienamente gestibile nel contesto d'uso clinico previsto dalla piattaforma PerioGPT.

3.3 Scalabilità e sostenibilità della piattaforma

La transizione di sistemi basati su Intelligenza Artificiale dalla fase prototipale al rilascio in ambienti di produzione clinica impone un'analisi rigorosa dei compromessi (trade-off) tra l'elaborazione locale (Edge Computing) e le architetture distribuite (Cloud Computing). In ambito sanitario, tale dicotomia non si limita esclusivamente alle prestazioni algoritmiche, ma coinvolge intrinsecamente la sostenibilità computazionale dell'hardware, la fattibilità economica nel lungo periodo e l'aderenza alle normative sulla protezione dei dati sensibili. Le sezioni seguenti discutono le strategie implementate per superare i colli di bottiglia hardware e valutano la sostenibilità della piattaforma PerioGPT attraverso un'analisi del *Total Cost of Ownership* (TCO) in contesti operativi reali.

3.3.1 Ottimizzazione hardware e limitazioni

Dal punto di vista dell'hardware locale, la piattaforma è stata validata su una workstation dotata di CPU a 6 core, 32GB di RAM di sistema e GPU `NVIDIA GeForce GTX 1050 Ti` con soli 4GB di VRAM GDDR5. Questa configurazione rappresenta uno scenario volutamente conservativo: consente di simulare le condizioni operative di uno studio odontoiatrico medio,

ma impone vincoli stringenti sulla dimensione dei modelli caricabili e sulla gestione della memoria durante l'inferenza.

Per rendere eseguibili in locale modelli da 3 miliardi di parametri su una GPU con 4GB di VRAM è stata adottata una strategia di quantizzazione aggressiva in formato `GGUF` a 4-bit (preset `Q4_K_M`) per i modelli `llama3.2 3B` e `qwen2.5 3B`. In condizioni full-precision, un modello di queste dimensioni richiederebbe diversi gigabyte di memoria grafica solo per i pesi; la quantizzazione riduce questo footprint a circa 2.1GB, rendendo possibile il caricamento dell'intero modello nella VRAM della 1050 Ti senza ricorrere a tecniche di offloading più invasive.

L'adozione di una quantizzazione aggressiva a 4-bit è in linea con quanto proposto in letteratura per ridurre il footprint di memoria dei modelli Transformer mantenendo prestazioni accettabili [29].

L'adozione del formato `GGUF` si inserisce in un compromesso ingegneristico ben preciso: si rinuncia a una piccola frazione di accuratezza numerica in cambio della possibilità di eseguire inferenza locale su hardware non professionale. I benchmark riportati nelle sezioni precedenti mostrano come, per il dominio parodontale considerato, questa perdita sia ampiamente accettabile rispetto al guadagno in termini di tempi di risposta e di controllo sui dati.

Nei modelli Transformer autoregressivi, ad ogni passo di generazione vengono calcolate e memorizzate le coppie di vettori `Key` e `Value` per ciascun token del contesto; l'insieme di queste strutture è indicato come `KV Cache` e permette al modello di riutilizzare i calcoli già effettuati, evitando di rielaborare l'intera sequenza a ogni nuovo token.

La Figura 3.10 riassume l'allocazione tipica della VRAM durante l'inferenza locale di `qwen2.5-3B` quantizzato a 4-bit sulla `GTX 1050 Ti`. Su un totale di 4096MB disponibili, circa 500MB sono occupati dal sistema operativo e da processi di background, 2100MB sono riservati ai pesi del modello quantizzato e circa 1000MB vengono impiegati dalla `KV Cache` e dalla finestra di contesto, lasciando appena 496MB di margine libero.

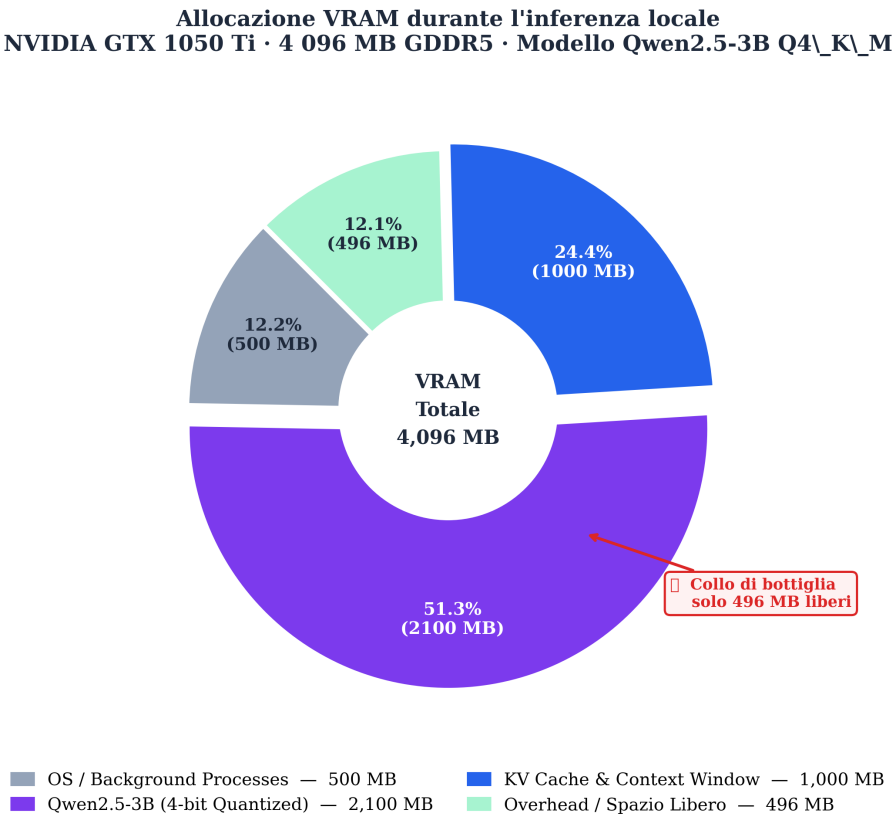

Figura 3.10: Allocazione tipica della VRAM durante l'inferenza locale

La `KV Cache` costituisce il principale fattore dinamico nel consumo di VRAM: a ogni nuovo token generato, il modello memorizza le rappresentazioni intermedie necessarie a riutilizzare i calcoli nei passi successivi. Studi recenti sulla caratterizzazione della `KV Cache` mostrano che, una volta allocati i pesi del modello, la quota predominante della memoria GPU viene assorbita proprio dalla cache delle chiavi e dei valori, con effetti diretti su throughput e latenza in presenza di contesti lunghi e richieste concorrenti [30]. All'aumentare della lunghezza del contesto, la dimensione della cache cresce linearmente fino a saturare quasi completamente lo spazio disponibile. In questo scenario la GPU opera sistematicamente vicino al limite, con un "cuscinetto" di meno di 500MB che deve essere condiviso con possibili picchi di utilizzo dovuti al sistema e al driver grafico.

Quando la conversazione si prolunga o la finestra di contesto viene spinta a valori elevati, la dimensione della `KV Cache` supera la capacità effettiva di VRAM destinabile al modello e alle strutture ausiliarie. In questi casi il runtime di inferenza ricorre automaticamente a meccanismi di *offloading* verso la RAM di sistema: una parte dei tensori viene spostata dalla VRAM alla memoria principale e accessibile tramite il bus PCIe.

Questo comportamento consente al modello di continuare a funzionare anche in presenza di contesti molto lunghi, ma introduce un collo di bottiglia evidente: l'accesso alla RAM

di sistema ha una latenza e una banda passante nettamente inferiori rispetto alla VRAM, con un impatto diretto sui tempi di generazione. In pratica, quando il contesto supera una certa soglia, la GPU non è più il solo componente limitante; la velocità di inferenza diventa progressivamente vincolata dal throughput del bus PCIe e dalla gestione della memoria da parte del sistema operativo.

Per mitigare questo collo di bottiglia, la piattaforma adotta diverse strategie:

- *Limitazione della finestra di contesto effettiva*: nei task in cui non è necessario mantenere l'intera cronologia della sessione, il backend tronca o riassume le parti più vecchie della conversazione, riducendo la quantità di stato che deve essere mantenuta nella `KV Cache`.
- *Separazione dei flussi*: la dettatura della cartella parodontale e le interrogazioni RAG vengono gestite come flussi logici distinti, evitando di sommare in un unico prompt sequenze testuali molto lunghe e di natura eterogenea.
- *Pre-caricamento controllato dei modelli*: i modelli locali vengono caricati una sola volta e mantenuti caldi in memoria, minimizzando i costi di allocazione/deallocazione della VRAM e riducendo il rischio di frammentazione.

Grazie a queste strategie, la piattaforma riesce a sfruttare al massimo una GPU entry-level come la `GTX 1050 Ti`, mantenendo latenze accettabili nella maggior parte degli scenari clinici simulati. Allo stesso tempo, l'analisi dell'allocazione di memoria evidenzia chiaramente come l'infrastruttura operi al margine delle proprie capacità: un incremento significativo della finestra di contesto, del numero di sessioni concorrenti o della dimensione dei modelli richiederebbe l'adozione di GPU con maggiore dotazione di VRAM o l'introduzione di ulteriori meccanismi di offloading e compressione.

3.3.2 Scalabilità dell'infrastruttura Cloud-Native

Per superare i severi vincoli computazionali dell'hardware locale senza abdicare ai requisiti normativi imposti dal Regolamento Generale sulla Protezione dei Dati (GDPR), l'architettura della piattaforma PerioGPT è stata progettata secondo un paradigma `Cloud-Native`. In ambito sanitario, l'adozione del cloud richiede che la protezione dei dati sia considerata già in fase di modellazione dell'architettura, con particolare attenzione a concetti come `data residency`, minimizzazione del dato e tracciabilità delle operazioni di trattamento [31, 32]. L'uso diretto di API pubbliche generaliste (ad esempio `OpenAI GPT-4o`) per l'elaborazione di dati clinici non anonimizzati esporrebbe lo studio odontoiatrico a rischi di perdita di sovranità sul dato, dipendenza dal fornitore e incertezza sul rispetto dei vincoli giurisdizionali.

Nel contesto europeo, la discussione sull'adozione dell'Intelligenza Artificiale in sanità non può essere disgiunta dal tema della `sovranità digitale`: la capacità degli Stati e delle istituzioni di mantenere il controllo effettivo sui dati, sulle infrastrutture che li elaborano e sui modelli che ne determinano l'interpretazione. Il quadro normativo definito da GDPR, NIS2 e dal futuro `EU AI Act` spinge verso soluzioni in cui l'elaborazione di dati clinici avvenga su infrastrutture soggette a giurisdizione europea, con garanzie forti di `data residency`, tracciabilità e valutazione preventiva dell'impatto privacy (DPIA) [31].

In questa prospettiva, le infrastrutture di `Sovereign AI` ospitate in `Virtual Private Cloud` (VPC) dedicati rappresentano un compromesso equilibrato tra flessibilità del cloud pubblico e controllo tipico dell'on-premise. I VPC vengono allocati su cluster GPU localizzati in data center europei, con isolamento logico dei tenant, cifratura end-to-end e politiche di accesso fortemente granulari; il traffico di dati clinici non lascia mai il perimetro giurisdizionale concordato e resta soggetto alle sole normative UE. Dal punto di vista dell'utente clinico, l'esperienza rimane quella di un servizio `Cloud-Native` elastico; dal punto di vista regolatorio, l'infrastruttura si comporta come un'estensione controllata del dominio sanitario, rafforzando la sovranità digitale dell'organizzazione.

L'orchestrazione dei modelli open-weight (ad esempio `Llama 3` o `Qwen`) all'interno del VPC avviene tramite piattaforme `MLOps` enterprise o stack containerizzati basati su `Docker Swarm` o `Kubernetes`, che permettono di scalare orizzontalmente il numero di repliche dedicate agli LLM e ai servizi di `Retrieval-Augmented Generation`. In caso di picchi di interrogazioni simultanee provenienti da più poltrone o da più sedi, il cluster può effettuare il provisioning dinamico di nuovi container mantenendo tempi di latenza stabili, senza esporre il dato clinico a provider extra-UE né rinunciare al controllo sul ciclo di vita dei modelli.

Il modello operativo previsto per PerioGPT è intrinsecamente ibrido. Le componenti che trattano dati particolarmente sensibili e che beneficiano della bassa latenza locale (come la dettatura vocale della cartella parodontale e la gestione della `KV Cache` dei modelli compressi) sono eseguite sulla workstation dello studio. Al contrario, i carichi di lavoro `compute-intensive` e meno sensibili alla latenza di rete (ad esempio l'addestramento di modelli personalizzati, la generazione massiva di report o la validazione batch di linee guida cliniche) possono essere delegati al cluster Sovereign AI ospitato nel VPC.

Questa separazione consente di:

- ridurre il rischio di esposizione dei dati identificativi, mantenendo in locale le informazioni strettamente riconducibili al paziente e trasferendo al VPC solo rappresentazioni pseudonimizzate o aggregate;
- sfruttare l'elasticità del cloud per operazioni episodiche ma intensamente computazionali, senza dover sovradimensionare l'hardware dello studio;

- mantenere un punto di controllo centralizzato sulle politiche di accesso, il ciclo di vita dei modelli e l'osservabilità del sistema,requisito essenziale per una futura certificazione in ambito `Medical Device Regulation` (MDR).

In questa prospettiva, l'infrastruttura locale funge da "edge intelligente" che pre-filtra, contestualizza e anonimizza i dati, mentre il VPC Sovereign AI si occupa delle operazioni più pesanti garantendo al contempo che nessuna informazione esca dal perimetro giurisdizionale stabilito. Ciò è coerente con la strategia europea per l'Intelligenza Artificiale, che mira a coniugare adozione diffusa dell'AI e rafforzamento dell'autonomia tecnologica del continente [33].

Per valutare la sostenibilità economica di questa architettura ibrida è stato elaborato un modello di `Total Cost of Ownership` (TCO) su un orizzonte di 24 mesi confrontando due scenari estremi:

1. *Scenario Cloud-Native puro*: utilizzo esclusivo di API LLM commerciali (es. `GPT-4o`) con tariffazione a consumo, senza investimento iniziale in hardware dedicato.

2. *Scenario On-Premises dedicato*: acquisto di un server con GPU di fascia media (ad esempio `RTX 4060`) e deployment dei modelli open-weight in locale, con costi operativi limitati a energia e manutenzione.

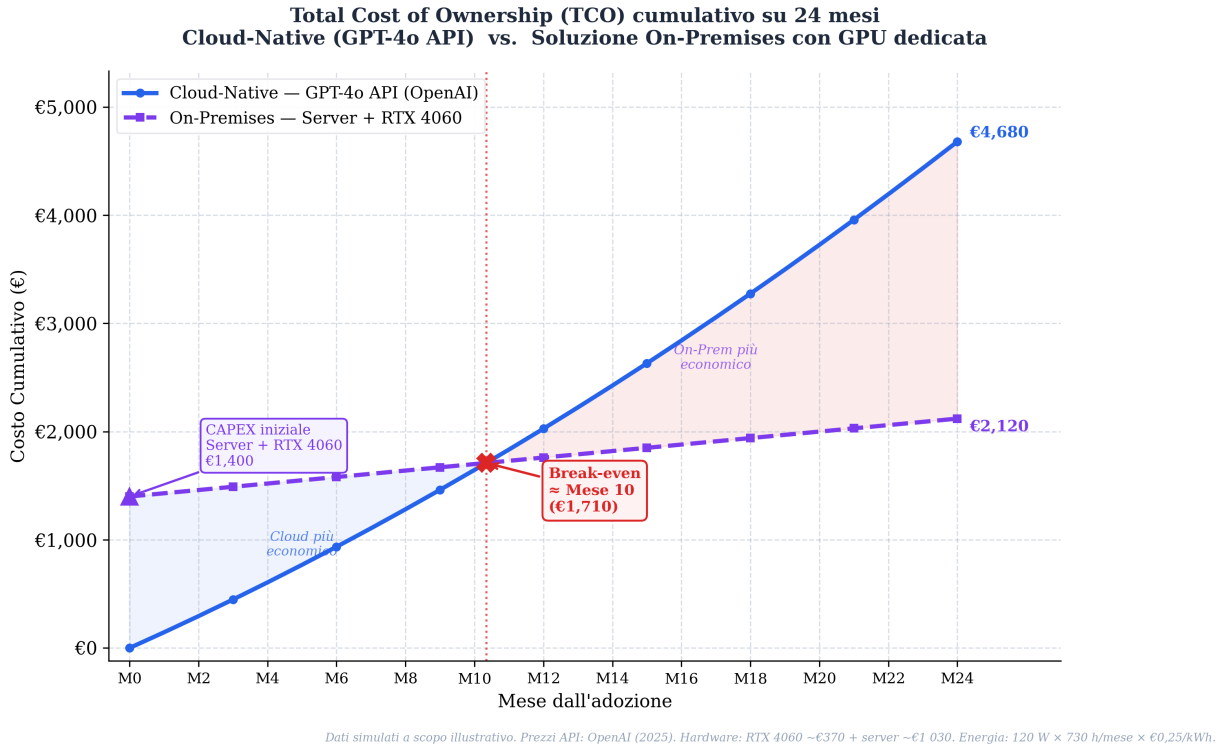

Figura 3.11: Proiezione cumulativa a 24 mesi del Total Cost of Ownership

Nello scenario Cloud-Native, la barriera all'ingresso è praticamente nulla (costo iniziale pari a 0 euro al mese M0), caratteristica che rende questa soluzione particolarmente appetibile per piccoli studi che desiderano sperimentare l'uso dell'Intelligenza Artificiale senza impegnare subito capitale. Tuttavia, la curva del TCO cresce linearmente con il volume di token elaborati, come illustrato in Figura 3.11, raggiungendo un valore stimato di circa 4680 euro dopo 24 mesi, ipotizzando un carico di lavoro costante e non trascurabile.

Nello scenario On-Premises, al contrario, è richiesto un investimento iniziale in conto capitale (CAPEX) dell'ordine di 1400 euro per il server e la GPU dedicata, cui si sommano costi operativi mensili contenuti (consumo energetico e manutenzione di base). La curva del TCO risulta quindi più ripida nei primi mesi, ma cresce molto più lentamente nel lungo periodo, raggiungendo un costo cumulativo stimato di circa 2120 euro al termine dei 24 mesi.

Il punto di pareggio (break-even point) tra i due scenari si colloca intorno al decimo mese, a quota approssimativamente 1710 euro, oltre il quale la soluzione On-Premises diventa economicamente più vantaggiosa per volumi di utilizzo simili. Questo comportamento è coerente con quanto riportato in recenti studi di analisi cost–benefit sull'adozione di LLM on-premise, che evidenziano come il deployment locale di modelli open-source possa superare in convenienza economica i servizi API commerciali già in orizzonti temporali di pochi mesi quando il volume di richieste è elevato e stabile [34].

L'analisi del TCO suggerisce che nessuno dei due approcci, preso singolarmente, sia ottimale per tutti i profili di utilizzo. Per studi odontoiatrici di piccole dimensioni o in fase esplorativa, l'adozione iniziale di un modello Cloud-Native puro consente di minimizzare il rischio finanziario e di valutare l'impatto clinico-organizzativo dell'introduzione della piattaforma. Al crescere del volume di interrogazioni e della centralità della soluzione nel flusso di lavoro, tuttavia, il costo cumulativo del cloud tende a superare la soglia di convenienza, spostando l'ago della bilancia verso un investimento in infrastruttura dedicata (locale o in Sovereign Cloud).

In questo contesto, l'architettura ibrida proposta per PerioGPT si configura come una strategia di compromesso ottimale:

- i task latency-sensitive e a forte contenuto di dati identificativi (dettatura della cartella, aggiornamento in tempo reale dello status parodontale) sono gestiti dall'hardware locale, già ammortizzato e posto sotto il pieno controllo dello studio;
- i compiti ad alto consumo computazionale ma meno sensibili alla latenza (ad esempio la generazione massiva di documenti o la sperimentazione di nuovi modelli) possono essere eseguiti su cluster Sovereign AI elastici, attivati solo quando necessario;
- la possibilità di migrare progressivamente parte dei carichi verso un VPC dedicato consente di adattare l'infrastruttura alle esigenze dello studio, mantenendo una exit

`strategy` chiara rispetto ai fornitori di API pubbliche e riducendo il rischio di `vendor lock-in`.

In prospettiva, tale architettura abilita anche scenari multi-centro, in cui più studi o strutture sanitarie condividono lo stesso cluster Sovereign AI, beneficiando di economie di scala sui costi di GPU e manutenzione, senza rinunciare alla separazione logica dei dati e alla piena tracciabilità delle operazioni di trattamento. In questo senso, PerioGPT non si limita a essere un prototipo applicativo, ma rappresenta una possibile `blueprint` per piattaforme di `Clinical AI` pienamente conformi al quadro regolatorio europeo e all'agenda di sovranità digitale dell'Unione.

3.4 Possibili estensioni ed integrazione con sistemi sanitari

I risultati ottenuti nelle analisi precedenti mostrano come la piattaforma PerioGPT sia in grado di supportare in modo efficace la raccolta strutturata del dato parodontale, la consultazione della letteratura tramite RAG e la generazione di spiegazioni comprensibili per il paziente. In questa sezione vengono delineate le possibili estensioni dell'ecosistema proposto e le modalità con cui il sistema potrebbe essere integrato all'interno di infrastrutture sanitarie esistenti, quali cartelle cliniche elettroniche (*Electronic Health Record*, EHR), sistemi informativi ospedalieri (*Hospital Information System*, HIS) e registri clinici multi-centro.

Dal punto di vista funzionale, le estensioni riguardano principalmente tre direzioni:

- l'evoluzione della pipeline di *Information Retrieval* verso configurazioni di tipo *Modular RAG* e l'utilizzo di *Knowledge Graph* clinici per query semantiche avanzate;
- l'interoperabilità del dato prodotto dalla piattaforma con gli standard internazionali per la sanità digitale (FHIR, HL7, openEHR);
- il rafforzamento degli aspetti di auditabilità e tracciabilità, in un'ottica di futura certificazione come *Medical Device Software* conforme al quadro regolatorio europeo.

3.4.1 Modular RAG e Knowledge Graph

L'attuale implementazione di PerioGPT adotta una pipeline RAG classica, basata su un singolo indice vettoriale in `ChromaDB` e su un retriever ibrido (`dense + BM25 + reranker`) che opera sull'intero corpus documentale. Questa configurazione ha dimostrato di migliorare sensibilmente le prestazioni di *Information Retrieval* rispetto al solo retriever denso, come descritto nel paragrafo 3.2, ma presenta alcuni limiti nel momento in cui si voglia scalare la

base di conoscenza a domini multipli (parodontologia, implantologia, parodontite aggressiva, protocolli chirurgici, linee guida di igiene orale, ecc.).

Una prima estensione naturale consiste nell'adottare un approccio di tipo *Modular RAG*, in cui il corpus viene suddiviso in collezioni logiche (indici separati per linee guida, articoli di ricerca, protocolli interni di studio, documentazione di dispositivi medici) e il retriever viene reso consapevole del tipo di informazione richiesto. In questo scenario, la piattaforma potrebbe:

- selezionare dinamicamente il sotto-indice più appropriato in base al tipo di domanda (ad esempio distinguendo tra domande normative, quesiti clinici e richieste di spiegazione per il paziente);
- combinare i risultati provenienti da più retriever specializzati (ad esempio un retriever ottimizzato per documenti brevi e uno per linee guida di grandi dimensioni), migliorando sia l'accuratezza sia la robustezza alle formulazioni non standard;
- applicare politiche di aggiornamento differenziate (indici aggiornati in tempo reale per i protocolli interni, aggiornamenti periodici per la letteratura scientifica).

Un secondo asse evolutivo riguarda l'integrazione di *Knowledge Graph* (KG) clinici a supporto del retrieval. Mentre gli indici vettoriali rappresentano il contenuto testuale in uno spazio continuo, un KG permette di modellare esplicitamente entità, relazioni e gerarchie (ad esempio la relazione fra parodontite, fattori di rischio sistemici, terapie causali, indici clinici e outcome nel tempo). L'adozione di ontologie mediche standardizzate (come `SNOMED CT`, `ICD-10` o codifiche odontoiatriche specifiche) consentirebbe di:

- formulare query multi-hop del tipo "mostra la letteratura che collega il fumo, la parodontite generalizzata e il fallimento implantare";
- disambiguare termini polisensi o ambigui (ad esempio distinguere tra parodontite cronica e aggressiva, oppure tra recessione e perdita di attacco clinico);
- abilitare forme di reasoning simbolico leggero a supporto delle raccomandazioni terapeutiche, mantenendo comunque la generazione testuale affidata all'LLM.

Dal punto di vista architetturale, un tale ampliamento può essere perseguito senza stravolgere il disegno complessivo della piattaforma: il modulo RAG viene evoluto in un *Retrieval Orchestrator* capace di interrogare sia gli indici vettoriali sia il *Knowledge Graph*, fondendo poi le evidenze in un unico contesto che viene passato al modello linguistico. Questa modularità risulta particolarmente utile in scenari multi-centro, in cui ogni struttura sanitaria può mantenere un proprio KG locale arricchito con protocolli interni, mantenendo al contempo una base di conoscenza scientifica condivisa.

3.4.2 Interoperabilità e standardizzazione dei dati

Affinché PerioGPT possa essere integrata in un ecosistema sanitario reale, è necessario che il dato prodotto dalla piattaforma sia interoperabile con le principali cartelle cliniche elettroniche e con i sistemi informativi di studio o ospedalieri. Attualmente la piattaforma utilizza strutture `JSON` ottimizzate per l'elaborazione interna (ad esempio per rappresentare il sondaggio parodontale); un passo successivo consiste nel definire una mappatura sistematica tra tali strutture e gli standard di interoperabilità affermati in ambito sanitario.

In particolare:

- lo stato parodontale e i piani di trattamento potrebbero essere mappati su risorse `FHIR` come `Observation`, `Condition`, `Procedure` e `CarePlan`, rendendo possibile lo scambio di informazioni con EHR conformi a questo standard;
- le codifiche diagnostiche e procedurali potrebbero essere allineate a sistemi di classificazione riconosciuti (ad esempio `ICD-10` per le diagnosi sistemiche e codifiche odontoiatriche nazionali per le procedure), facilitando la rendicontazione amministrativa e la ricerca clinica;
- i referti radiologici e le valutazioni su ortopantomografie potrebbero essere collegati a metadati compatibili con `DICOM`, consentendo l'integrazione con i *Picture Archiving and Communication System* (PACS) già in uso presso le strutture sanitarie.

Dal punto di vista implementativo, ciò richiede l'introduzione di un livello di *Data Mapping* che agisca come *Anti-Corruption Layer* (ACL), fungendo da traduttore bidirezionale tra il dominio interno della piattaforma (pensato per l'elaborazione da parte degli LLM) e il dominio esterno degli standard sanitari. Questo livello potrebbe essere configurato tramite template dichiarativi, così da adattare la piattaforma a diversi contesti regionali o nazionali senza modificare il core applicativo.

Un ulteriore aspetto di interoperabilità riguarda la gestione del consenso e delle preferenze del paziente. L'integrazione con i sistemi informativi esistenti permetterebbe di recuperare e rispettare automaticamente le impostazioni di consenso informato registrate a livello di EHR (ad esempio limitazioni sull'uso secondario dei dati per ricerca o addestramento di modelli), contribuendo a rafforzare la conformità normativa e la fiducia nell'utilizzo di sistemi di *Clinical AI* in odontoiatria.

Capitolo 4

Conclusioni

All'interno del presente lavoro di tesi è stata progettata un'infrastruttura di *Clinical AI* specializzata per la parodontologia basata sul progetto PerioGPT. La piattaforma integra modelli linguistici di grandi dimensioni e pipeline RAG in un'architettura ibrida *Cloud-Native* e *On-Premises*, garantendo la totale sovranità del dato. Il sistema automatizza la dettatura della cartella clinica, struttura i parametri in formato `JSON`, interroga basi documentali specialistiche e genera spiegazioni trasparenti per il paziente, tramite tecniche di XAI, operando con latenze compatibili con la pratica ambulatoriale.

L'analisi sperimentale ha validato le prestazioni dei modelli locali (tramite `Ollama`) su hardware vincolato (NVIDIA GTX 1050 Ti con 4 GB di VRAM). L'adozione della quantizzazione a 4-bit (formato `GGUF`) e l'ottimizzazione della `KV Cache` hanno permesso di eseguire l'inferenza di modelli *open-weight* da 3 miliardi di parametri, individuando in `qwen2.5-3B` l'architettura ottimale per l'estrazione `JSON` senza allucinazioni. Contestualmente, la valutazione del modulo di *Information Retrieval* ha certificato che un approccio ibrido con *re-ranking* neurale massimizza metriche chiave quali Hit Rate e MRR rispetto al solo recupero denso, introducendo un *overhead* computazionale trascurabile e compatibile con un utilizzo interattivo. L'infrastruttura ibrida proposta delega l'elaborazione dei task a bassa latenza e ad alta sensibilità (come l'aggiornamento dello *status* parodontale) all'hardware locale, riservando i carichi computazionalmente intensivi a cluster *Sovereign AI* ospitati in *Virtual Private Cloud* (VPC) dedicati. L'analisi del *Total Cost of Ownership* (TCO) a 24 mesi ha quantificato la sostenibilità economica di tale approccio: sebbene un'architettura esclusivamente *Cloud-Native* azzeri i costi di *setup* iniziali, l'esecuzione *On-Premises* diviene economicamente vantaggiosa nel medio periodo, ottimizzando il rapporto tra spese in conto capitale e costi operativi.

Infine, sono state definite le direttive architetturali per l'integrazione della piattaforma nei sistemi informativi sanitari preesistenti. La transizione verso paradigmi di *Modular RAG* e *Knowledge Graph* clinici consente di mappare ontologie specialistiche complesse, mentre

la serializzazione delle strutture `JSON` sui protocolli `FHIR`, `HL7`, `openEHR` e `DICOM` garantisce la piena interoperabilità globale con cartelle cliniche elettroniche (EHR) e sistemi PACS. In conclusione, PerioGPT dimostra che l'implementazione sinergica di LLM, pipeline RAG e infrastrutture *Sovereign AI* converte i modelli predittivi in strumenti clinici sicuri, trasparenti e pienamente conformi al quadro regolatorio europeo sulla protezione dei dati.

Bibliografia

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

[2] Vito Carlo Alberto Caponio, Alejandro I Lorenzo-Pouso, Marco Magalhaes, Aiman Ali, Daniela Adamo, Nicola Cirillo, Rosa María López-Pintor, and Gennaro Musella. Accuracy of llms to retrieve numeric data for meta-analysis in dentistry. *Journal of Dentistry*, page 106245, 2025.

[3] Tahereh Firoozi, Rojin Adabdokht, and Hollis Lai. Retrieval augmented generation in dentistry: potentials, applications, and future directions. *Frontiers in Dental Medicine*, 6:1760990, 2026.

[4] Hannaneh B Pasandi and Haniyeh B Pasandi. Evaluation of automated speech recognition systems for conversational speech: A linguistic perspective. *arXiv preprint arXiv:2211.02812*, 2022.

[5] Yashbir Singh, Quincy A Hathaway, Varekan Keishing, Sara Salehi, Yujia Wei, Natally Horvat, Diana V Vera-Garcia, Ashok Choudhary, Almurtadha Mula Kh, Emilio Quaia, et al. Beyond post hoc explanations: a comprehensive framework for accountable ai in medical imaging through transparency, interpretability, and explainability. *Bioengineering*, 12(8):879, 2025.

[6] Changsheng Zhang and Linjun Liu. Machine learning prediction model for medical environment comfort based on shap and lime interpretability analysis. *Scientific Reports*, 15(1):39269, 2025.

[7] Fnu Neha, Deepshikha Bhati, and Deepak Kumar Shukla. Retrieval-augmented generation (rag) in healthcare: A comprehensive review. *AI*, 6(9):226, 2025.

[8] Krzysztof Wołk. Evaluating retrieval-augmented generation variants for clinical decision support: Hallucination mitigation and secure on-premises deployment. *Electronics*, 14(21):4227, 2025.

[9] Michael DeBellis, Nivedita Dutta, Jacob Gino, and Aadarsh Balaji. Integrating ontologies and large language models to implement retrieval augmented generation. *Applied Ontology*, 19(4):389–407, 2024.

[10] Selen Erkul, Yunus Özden, İdil Özden, Bengü Kaya, Burcu Bal, and ZEYNEP ÖZKURT-KAYAHAN. The use of artificial intelligence in different dental applications. *Cyprus Journal of Medical Sciences*, 10(4), 2025.

[11] Francesco Fanelli, Muhammad Saleh, Pasquale Santamaria, Khrystyna Zhurakivska, Luigi Nibali, and Giuseppe Troiano. Development and comparative evaluation of a reinstructed gpt-4o model specialized in periodontology. *Journal of Clinical Periodontology*, 52(5):707–716, 2025.

[12] Kory Kreimeyer, Matthew Foster, Abhishek Pandey, Nina Arya, Gwendolyn Halford, Sandra F Jones, Richard Forshee, Mark Walderhaug, and Taxiarchis Botsis. Natural language processing systems for capturing and standardizing unstructured clinical information: a systematic review. *Journal of biomedical informatics*, 73:14–29, 2017.

[13] Yi-June Lo, Yung-Chun Chang, Chien-Hung Chen, Hsuan-Po Wang, Hao-Chen Wang, Yoichi Ohiro, and Chih-Yuan Fang. Evaluation of the performance of an artificial intelligence model in recognizing the habitual mixed language in taiwan for generating periodontal charting text reports. *Journal of Dental Sciences*, 2025.

[14] R Karthick. End-to-end asr conformers: Revolutionizing hearing-to-speech-to-writing language processing frameworks. 2026.

[15] Peiyuan Jiang, Weijun Pan, Jian Zhang, Teng Wang, and Junxiang Huang. A robust conformer-based speech recognition model for mandarin air traffic control. *Computers, Materials & Continua*, 77(1), 2023.

[16] Ibomoiye Domor Mienye, George Obaido, Nobert Jere, Ebikella Mienye, Kehinde Aruleba, Ikiomoye Douglas Emmanuel, and Blessing Ogbuokiri. A survey of explainable artificial intelligence in healthcare: Concepts, applications, and challenges. *Informatics in Medicine Unlocked*, 51:101587, 2024.

[17] Ioannis Givisis, Dimitris Kalatzis, Christos Christakis, and Yiannis Kiouvrekis. Comparing explainable ai models: Shap, lime, and their role in electric field strength prediction over urban areas. *Electronics*, 14(23):4766, 2025.

[18] William Van Woensel, Floriano Scioscia, Giuseppe Loseto, Oshani Seneviratne, Evan Patton, Samina Abidi, and Lalana Kagal. Explainable clinical decision support: towards patient-facing explanations for education and long-term behavior change. In

International conference on artificial intelligence in medicine, pages 57–62. Springer, 2022.

[19] Xiaoming Liu, Danni Huang, Jingyu Yao, Jing Dong, Litong Song, Hui Wang, Chao Yao, and Weishen Chu. From black box to glass box: A practical review of explainable artificial intelligence (xai). *AI*, 6(11):285, 2025.

[20] Shahab S Band, Atefeh Yarahmadi, Chung-Chian Hsu, Meghdad Biyari, Mehdi Sookhak, Rasoul Ameri, Iman Dehzangi, Anthony Theodore Chronopoulos, and Huey-Wen Liang. Application of explainable artificial intelligence in medical health: A systematic review of interpretability methods. *Informatics in Medicine Unlocked*, 40:101286, 2023.

[21] Sehyeong Jo, Gangjae Jang, and Haesol Park. Gmar: gradient-driven multi-head attention rollout for vision transformer interpretability. pages 582–587, 2025.

[22] Abhinav Kumar, Abhishek Verma, Sandeep Kumar Dubey, and Anuj Singh. Secure Cloud Native Architecture for AI Enabled Health Data Management and Interoperability. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, 14(4), 2026.

[23] John Rong Hao Tay, Dian Yi Chow, Yi Rong Ivan Lim, and Ethan Ng. Enhancing patient-centered information on implant dentistry through prompt engineering: a comparison of four large language models. *Frontiers in Oral Health*, 6:1566221, 2025.

[24] Alessio Bucaioni, Martin Weyssow, Junda He, Yunbo Lyu, and David Lo. A functional software reference architecture for llm-integrated systems. pages 1–5, 2025.

[25] Kefei Liu and Meizhu Liu. Synthetic doctor-patient dialogue generation for robust medical asr: A scalable pipeline for vocabulary expansion and privacy preservation. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 525–534, 2026.

[26] Sujal Chondhekar, Vasanth Murukuri, Rushabh Vasani, Sanika Goyal, Rajshree Badami, Anushree Rana, Sanjana SN, Karthik Pandia, Sulabh Katiyar, Neha Jagadeesh, et al. When de-noising hurts: A systematic study of speech enhancement effects on modern medical asr systems. *arXiv preprint arXiv:2512.17562*, 2025.

[27] Subhabrata Mukherjee, Markel Sanz Ausin, Kriti Aggarwal, Debajyoti Datta, Shanil Puri, Woojeong Jin, Tanmay Laud, Neha Manjunath, Jiayuan Ding, Bibek Paudel, et al. Perfecting human-ai interaction at clinical scale. turning production signals into safer, more human conversations. *arXiv preprint arXiv:2603.29893*, 2026.

[28] Osama Khattak, Ahmed Shawkat Hashem, Mohammed Saad Alqarni, Raha Ahmed Shamikh Almufarrij, Amna Yusuf Siddiqui, Rabia Anis, Shahzad Ahmad, Muhammad Amber Fareed, Osama Shujaa Alothmani, Lama Habis Samah Alkhershawy, et al. Deep learning applications in dental image-based diagnostics: A systematic review. In *Healthcare*, volume 13, page 1466. MDPI, 2025.

[29] Shih-yang Liu, Zechun Liu, Xijie Huang, Pingcheng Dong, and Kwang-Ting Cheng. Llm-fp4: 4-bit floating-point quantized transformers. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 592–605, 2023.

[30] Jie Ye, Jaime Cernuda, Avinash Maurya, Xian-He Sun, Anthony Kougkas, and Bogdan Nicolae. Characterizing the behavior and impact of kv caching on transformer inferences under concurrency. In *2025 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 1191–1202. IEEE, 2025.

[31] M Emilia Cambronero, Miguel A Martínez, Luis Llana, Ricardo J Rodríguez, and Alejandro Russo. Towards a gdpr-compliant cloud architecture with data privacy controlled through sticky policies. *PeerJ Computer Science*, 10:e1898, 2024.

[32] Dimitra Georgiou and Costas Lambrinoudakis. Data protection impact assessment (dpia) for cloud-based health organizations. *Future Internet*, 13(3):66, 2021.

[33] Tessel Van Oirsouw. Ai, digital sovereignty, and the eu's path forward: A case for mission-oriented industrial policy. *Ash Center for Democratic Governance and Innovation, Harvard Kennedy School. Retrieved May*, 5:2025, 2023.

[34] Guanzhong Pan, Vishal Chodnekar, Abinas Roy, and Haibo Wang. A cost-benefit analysis of on-premise large language model deployment: Breaking even with commercial llm services. *arXiv preprint arXiv:2509.18101*, 2025.