UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

From Comorbidity to Prediction: A Platform for Disease Evolution Analysis Using Graph Neural Networks in Healthcare

RELATORI

Prof.ssa Delfina Malandrino

Prof. Rocco Zaccagnino

Università degli Studi di Salerno

CANDIDATO

Tullio Mansi

Matricola: 0512114647

Anno Accademico 2024-2025

A te che stai leggendo

Abstract

Questa tesi esplora il fenomeno della comorbidità attraverso un'analisi dettagliata dei dati clinici, con l'obiettivo di comprendere le connessioni tra diverse malattie e fornire supporto ai professionisti della salute. Lo studio si sviluppa in più fasi: inizialmente, è stata condotta una valutazione approfondita delle comorbidità e dei pattern presenti nei dati sanitari strutturati.

A seguire, è stata creata ComorGraph, una piattaforma di Visual Analytics e Social Network Analysis (SNA) integrata all'interno di MedMiner, che consente di visualizzare le comorbidità come un grafo. Utilizzando metriche grafiche, la piattaforma permette di individuare malattie centrali e analizzare le loro correlazioni.

Infine, lo studio si completa con l'implementazione di un modulo di intelligenza artificiale basato su Graph Neural Network (GNN), utilizzato per effettuare previsioni sull'insorgenza di nuove patologie nei pazienti. Sebbene questo modulo non sia parte di ComorGraph, fornisce previsioni utili per la gestione delle comorbidità, offrendo un potente strumento di supporto decisionale.

Questa tesi è stata realizzata in

EDGE
RESEARCH LAB

Indice

CAPITOLO 1

Introduzione

1.1 Contesto Applicativo

Lo studio della comorbidità, ovvero la coesistenza di più malattie in un paziente, è diventato un argomento cruciale nella medicina moderna. Comprendere come le patologie si influenzano reciprocamente è fondamentale per migliorare la gestione clinica dei pazienti con condizioni complesse. La crescita esponenziale dei dati sanitari e l'interconnessione tra pazienti e malattie richiede l'uso di strumenti innovativi per analizzare queste relazioni.

La rappresentazione dei dati clinici sotto forma di grafo offre una visione più chiara delle correlazioni tra le malattie. Un grafo permette di modellare pazienti e patologie come nodi, con le connessioni tra malattie rappresentate da archi, facilitando l'analisi della comorbidità tramite metriche di Social Network Analysis (SNA). Tuttavia, gli approcci tradizionali non considerano la dimensione temporale, rendendo difficile prevedere l'evoluzione clinica di un paziente.

Per affrontare queste sfide, è stata sviluppata **ComorGraph**, una piattaforma integrata nell'ecosistema **MedMiner** (progettato per supportare medici e specialisti nell'analisi di casi clinici complessi). **ComorGraph** sfrutta la rappresentazione grafica dei dati clinici per analizzare i pattern di correlazione tra malattie, consentendo una maggiore

precisione nelle prescrizioni terapeutiche e supportando la gestione dell'insorgenza di nuove patologie strettamente correlate a quelle già presenti nel quadro clinico del paziente.
A complemento della piattaforma, è stato condotto uno studio sull'applicazione del **Machine Learning** per facilitare l'analisi di questi pattern. Data la natura grafica della rappresentazione, è stato sviluppato un modello basato su **Graph Neural Networks (GNN)**, in grado di rilevare pattern complessi e meno evidenti, e di prevedere la probabilità che un paziente sviluppi determinate patologie in base al proprio quadro clinico. Questo approccio consente di delineare terapie personalizzate mirate non solo alla cura, ma anche alla prevenzione delle malattie con maggiore probabilità di insorgenza.

1.2 Motivazioni e Obiettivi

La gestione delle comorbidità, specialmente in pazienti con più patologie croniche, richiede strumenti avanzati per identificare le relazioni tra malattie e prevedere possibili sviluppi clinici. Gli approcci tradizionali si concentrano su modelli statici, spesso non sufficienti a cogliere l'interconnessione dinamica tra le patologie né a considerare la storia clinica del paziente nel tempo. Questa limitazione comporta difficoltà per i professionisti sanitari nel prendere decisioni preventive e nel gestire le comorbidità in modo efficiente.
Per rispondere a queste sfide, è stata sviluppata **ComorGraph**, una piattaforma innovativa che combina la rappresentazione grafica dei dati sanitari con algoritmi di intelligenza artificiale per fornire previsioni cliniche personalizzate.

◎ **Our Goal.** L'obiettivo di questo lavoro è stato lo sviluppo di una piattaforma innovativa per la Visual Analytics di dati clinici complessi, offrendo uno strumento avanzato per supportare i medici nella gestione delle comorbidità. In aggiunta, è stata sviluppata una Graph Neural Network (GNN) che consente di calcolare le probabilità di contrarre nuove malattie nei pazienti, migliorando così la capacità di intervenire in modo tempestivo e personalizzato.

1.3 Risultati Ottenuti

La sperimentazione condotta con **ComorGraph** ha prodotto risultati significativi nell'analisi della comorbidità e nella predizione delle malattie. La piattaforma è stata testata utilizzando un dataset di dati clinici reali, permettendo di identificare con precisione le relazioni tra le malattie e di evidenziare patologie particolarmente connesse tra loro.
Grazie all'applicazione delle metriche di **Social Network Analysis (SNA)**, è stato possibile individuare nodi centrali nel grafo delle malattie, fornendo nuove intuizioni sui collegamenti tra le patologie e sul loro impatto nei casi clinici. Inoltre, il modulo di intelligenza artificiale basato su una **Graph Neural Network (GNN)** ha dimostrato una capacità predittiva promettente, consentendo di stimare con buona accuratezza la probabilità di comparsa di una malattia specifica in un paziente nel tempo.
Questi risultati aprono nuove opportunità nell'uso dei grafi e dell'IA per migliorare la comprensione delle comorbidità e supportare i professionisti sanitari nella gestione dei pazienti con condizioni complesse.

1.4 Struttura della Tesi

Questa tesi è suddivisa in cinque capitoli, ognuno dei quali approfondisce aspetti specifici del lavoro di ricerca e sviluppo svolto.

- **Capitolo 2: Background e Stato dell'Arte** – Questo capitolo introduce i concetti chiave e le metodologie esistenti per lo studio della comorbidità tramite grafi, fornendo una panoramica delle tecniche di **Social Network Analysis (SNA)**, **Visual Analytics (VA)** e delle reti neurali grafiche (**GNN**) applicate al contesto medico.

- **Capitolo 3: Metodologia e Implementazione** – Viene descritto il processo di sviluppo della piattaforma **ComorGraph**, con dettagli sull'architettura del sistema, inoltre viene descritto il lavoro svolto sui dati e le metriche di **Social Network Analisys (SNA)** studiate. Il codice sorgente della piattaforma, insieme alla

documentazione dettagliata, è reso disponibile pubblicamente sul repository GitHub al seguente link: `https://github.com/Mall1o/MedMiner`.

- **Capitolo 4: Heterogeneous Graph Neural Networks per la Predizione di Relazioni Paziente-Malattia** - Questo capitolo fornisce una descrizione dettagliata sui modelli di Machine Learning utilizzati, un confronto tra questi e un'analisi dei risultati ottenuti. Il codice sorgente relativo ai modelli di Heterogeneous Graph Neural Networks (HGNN) utilizzati per la predizione delle relazioni Paziente-Malattia, insieme agli esperimenti svolti, è disponibile pubblicamente nel repository GitHub al seguente link: `https://github.com/Mall1o/MedMiner`. Questo permette di riprodurre i risultati e testare ulteriormente i modelli descritti in questo capitolo.

- **Capitolo 5: Conclusioni e Sviluppi Futuri** – Viene discusso l'impatto della piattaforma **ComorGraph** e i potenziali sviluppi futuri, con particolare attenzione alle possibili estensioni del sistema e alle sfide future nel campo della medicina predittiva.

CAPITOLO 2

Background e Stato dell'Arte

2.1 La comorbidità: una sfida clinica complessa

La comorbidità, definita come la coesistenza di due o più malattie croniche in uno stesso individuo, rappresenta una delle maggiori sfide per la medicina moderna. Questo fenomeno non solo complica la gestione clinica dei pazienti, ma impatta anche significativamente sulla qualità della vita e sulle risorse sanitarie. La presenza di più condizioni patologiche contemporanee aumenta il rischio di esiti clinici negativi, quali una maggiore mortalità, un peggioramento della qualità della vita e un aumento dei costi sanitari globali [1].

La distinzione tra comorbidità e multimorbidità, termini spesso usati in modo intercambiabile, è rilevante dal punto di vista clinico. Il termine comorbidità solitamente si riferisce alla presenza di altre malattie in aggiunta a una condizione principale (ad esempio, il diabete associato a ipertensione), mentre la multimorbidità implica l'assenza di un'unica malattia predominante, concentrandosi piuttosto sulla gestione di tutte le condizioni come un insieme [2].

L'aumento dell'aspettativa di vita e l'invecchiamento della popolazione hanno contribuito a un incremento della prevalenza della comorbidità. Questo trend ha portato a un cambiamento nella pratica clinica, con la necessità di passare da un approccio

centrato sulla singola malattia a una gestione olistica delle condizioni del paziente. I medici devono considerare non solo l'interazione tra le malattie stesse, ma anche l'impatto delle terapie su più condizioni contemporaneamente [1].
Oltre agli aspetti clinici, la comorbidità ha conseguenze sociali ed economiche rilevanti. La gestione di pazienti con comorbidità richiede un maggior utilizzo di risorse sanitarie, incluse cure più frequenti e personalizzate, prolungamenti dei tempi di degenza ospedaliera e un impatto diretto sui costi del sistema sanitario [3].

2.2 Social Network Analysis (SNA) nel contesto della comorbidità

La **Social Network Analysis (SNA)** è una tecnica metodologica utilizzata per studiare le relazioni tra entità in una rete. Applicata al campo medico, in particolare nello studio della comorbidità, la SNA permette di visualizzare e analizzare le interazioni tra patologie come se fossero nodi connessi tra loro da archi. Questo approccio aiuta a identificare pattern di comorbidità, ossia insiemi di malattie che tendono a manifestarsi insieme in determinati gruppi di pazienti, e a comprendere come queste patologie si influenzino a vicenda in termini di evoluzione e gravità clinica [4].
In una rete di comorbidità, le malattie più centrali, note come "hub", sono quelle che svolgono un ruolo cruciale nel collegare altre patologie, influenzando così la complessità dei quadri clinici. Le metriche di centralità, come la **betweenness centrality**, che misura quanto un nodo faciliti la comunicazione tra gli altri, e la **degree centrality**, che valuta il numero di connessioni di un nodo, sono essenziali per comprendere quali malattie hanno un impatto maggiore sulla rete di comorbidità. Ad esempio, condizioni come il diabete o le malattie cardiovascolari sono spesso centrali nelle reti di comorbidità, poiché tendono a essere associate a molte altre patologie [5].
L'utilizzo della SNA consente anche di eseguire simulazioni per prevedere l'evoluzione di malattie complesse e supportare la pianificazione delle terapie. Attraverso un'analisi delle connessioni tra le patologie, i ricercatori e i medici possono ottenere una visione più chiara delle interazioni tra malattie e prendere decisioni cliniche più informate. Ciò rappresenta un grande passo avanti verso una medicina più predittiva

e personalizzata [6].

2.3 Visual Analytics nel contesto Clinico

La **Visual Analytics** (VA) rappresenta un'intersezione tra l'analisi dei dati e la visualizzazione interattiva, con l'obiettivo di fornire agli esperti clinici strumenti efficaci per esplorare grandi quantità di dati, come quelli contenuti nei registri elettronici sanitari (EHR). Grazie alla crescente quantità di dati sanitari generati, il ruolo della VA è diventato fondamentale per superare le sfide legate all'**information overload**. Questo sovraccarico di informazioni rende complesso per i medici identificare pattern rilevanti o trarre conclusioni basate su dati non strutturati, specialmente quando si tratta di gestire patologie complesse come le comorbidità.
Nel contesto della **comorbidità**, la VA permette di visualizzare relazioni complesse tra malattie, consentendo ai medici e ai ricercatori di scoprire correlazioni, cluster di patologie e di migliorare la pianificazione terapeutica. Le visualizzazioni interattive forniscono un quadro immediato delle connessioni tra malattie, facilitando il riconoscimento di malattie centrali o "hub" che potrebbero svolgere un ruolo cruciale nel trattamento del paziente [7].
Uno degli aspetti più utili della VA è la sua capacità di combinare analisi statistiche avanzate con rappresentazioni visive intuitive, rendendo più agevole per i medici esplorare i dati e prendere decisioni cliniche più informate. Ad esempio, strumenti di clustering gerarchico permettono di raggruppare i pazienti in base a caratteristiche comuni o a comorbidità, mentre tecniche di filtraggio basate sulla varianza evidenziano le relazioni statisticamente più rilevanti tra patologie. In tal modo, la VA non solo aiuta a esplorare i dati clinici ma supporta anche il processo decisionale attraverso un'interfaccia visiva intuitiva che riduce la complessità delle reti di relazioni [8].
L'integrazione della VA nella ricerca medica e nell'analisi della comorbidità rappresenta una svolta importante per migliorare l'efficienza e la precisione nella diagnosi e trattamento delle malattie croniche complesse.

2.4 Machine Learning e predizione nelle reti di comorbidità

Il **machine learning (ML)** è una branca dell'intelligenza artificiale che consente ai computer di apprendere dai dati senza essere esplicitamente programmati. Attraverso algoritmi complessi, il ML è in grado di individuare pattern nascosti e fare previsioni basate su tali pattern. Le tecniche di ML vengono utilizzate in molte applicazioni, dalla classificazione delle immagini alla predizione di eventi futuri, grazie alla capacità di analizzare grandi quantità di dati e generare modelli predittivi accurati. Tra i metodi più comuni si trovano le reti neurali, gli alberi decisionali, e i modelli di regressione, che sono in grado di adattarsi a problemi molto diversi. Nel contesto delle reti di comorbidità, il machine learning viene applicato per prevedere l'evoluzione delle malattie in pazienti affetti da più condizioni croniche.

Le **reti neurali grafiche (GNN)** rappresentano un'evoluzione recente del machine learning, particolarmente adatta a modellare relazioni complesse come quelle presenti nelle reti sanitarie. Le GNN utilizzano la struttura del grafo, dove i nodi rappresentano pazienti o malattie, e gli archi rappresentano le connessioni tra di essi, come la presenza simultanea di malattie in un paziente o l'associazione tra patologie simili [9].

Applicando le tecniche di ML alle reti di comorbidità, è possibile non solo comprendere le relazioni attuali tra le malattie, ma anche prevedere quali altre condizioni potrebbero svilupparsi in futuro. Studi recenti hanno dimostrato che l'approccio delle GNN può essere utilizzato per effettuare previsioni accurate riguardanti malattie croniche, come il diabete o le malattie cardiovascolari, e per identificare collegamenti latenti tra patologie che potrebbero non essere evidenti con altri metodi [10].

Questi strumenti, insieme ad altre tecniche di machine learning come gli **autoencoder** e l'**embedding delle reti**, stanno migliorando significativamente la capacità predittiva in ambito medico, consentendo di ottimizzare la gestione delle comorbidità e di personalizzare i trattamenti per i pazienti. Il machine learning, quindi, si sta dimostrando un alleato prezioso per la medicina predittiva, fornendo strumenti potenti per anticipare l'insorgenza di nuove malattie e migliorare i risultati clinici [9].

2.5 Graph Neural Networks (GNN) applicate alla predizione delle malattie

Le **Reti Neurali** (neural networks) sono modelli di apprendimento automatico che si ispirano alla struttura del cervello umano, formati da strati di nodi (neuroni) collegati tra loro, in grado di apprendere dai dati e migliorare le loro performance nel tempo. Queste reti vengono utilizzate in diversi ambiti, tra cui la predizione delle malattie, grazie alla loro capacità di modellare e trovare pattern complessi nei dati clinici.

Le **reti neurali grafiche** (Graph Neural Networks - GNN), in particolare, sono un'evoluzione delle reti neurali che operano su dati strutturati sotto forma di grafo. Un grafo è una rappresentazione di dati costituita da nodi e archi, in cui i nodi rappresentano entità (come malattie o pazienti) e gli archi rappresentano le relazioni tra queste entità. Le GNN sono particolarmente utili quando i dati presentano una natura intrinsecamente connessa, come nel caso delle reti di comorbidità, dove diverse malattie possono essere collegate tra loro da relazioni complesse. Le GNN riescono a catturare queste relazioni per migliorare la comprensione e la predizione delle malattie [11].

Le **Heterogeneous Graph Neural Networks (HeteroGNN)** sono progettate per gestire grafi con nodi e relazioni di diverse tipologie, offrendo un potente strumento per modellare contesti complessi come quelli clinici. Grazie alla capacità di rappresentare e analizzare diverse entità cliniche (ad esempio pazienti, malattie, farmaci) e le loro interazioni, le HeteroGNN risultano fondamentali per predizioni cliniche avanzate, migliorando la comprensione delle relazioni tra malattie e supportando la diagnosi e il trattamento delle comorbidità [11].

Gli strati di convoluzione come **SAGEConv** e **GATConv** sono particolarmente rilevanti in questo contesto. Il **SAGEConv** (Sample and Aggregation) sfrutta campionamenti efficienti dai vicini di ciascun nodo per aggregare informazioni da grandi grafi eterogenei, rendendolo adatto per il trattamento di dati clinici su larga scala. Dall'altro lato, il **GATConv** (Graph Attention Convolution) utilizza meccanismi di attenzione per assegnare pesi diversi ai nodi vicini, catturando così meglio le interazioni cruciali tra entità cliniche, come pazienti e trattamenti, basate sull'importanza

delle connessioni [12].
Questi metodi consentono di migliorare la capacità delle HeteroGNN nel predire con precisione lo sviluppo di malattie e ottimizzare le strategie terapeutiche, fornendo ai medici strumenti di supporto decisionali sempre più sofisticati, capaci di operare su dati eterogenei e complessi.

2.6 Network Analysis e Machine Learning nel contesto sanitario

L'analisi delle reti e il **machine learning** stanno rivoluzionando il settore sanitario, non solo per la diagnosi, ma anche per la prevenzione delle complicazioni cliniche gravi. Uno degli obiettivi principali di queste tecniche è identificare pattern nascosti nei dati sanitari che possano indicare la progressione verso esiti negativi per il paziente. Questo approccio si è dimostrato particolarmente utile in ambiti come l'oncologia e la cardiologia, dove le malattie tendono a interagire in modi complessi e difficili da rilevare con le tecniche tradizionali.
In oncologia, ad esempio, il machine learning è stato utilizzato per analizzare grandi dataset di pazienti, identificando pattern nelle interazioni tra tumori e altre comorbidità che possono peggiorare l'esito clinico. Le tecniche di **Network Analysis** consentono di visualizzare come diverse malattie possano interagire nel corso del tempo, rivelando legami cruciali tra tumori e patologie croniche. Questo tipo di analisi aiuta a prevenire l'aggravarsi della situazione clinica, suggerendo terapie personalizzate e mirate prima che il paziente sviluppi condizioni critiche [13].
Nel campo della cardiologia, tecniche simili vengono applicate per studiare le relazioni tra malattie cardiovascolari e altre patologie comorbide, come il diabete o l'ipertensione. L'uso combinato di **machine learning** e **SNA** consente di identificare pazienti a rischio, prevedendo episodi cardiaci gravi come infarti o insufficienze cardiache sulla base di dati clinici storici e in tempo reale. Questo approccio proattivo supporta i medici nel monitoraggio continuo dei pazienti, migliorando la gestione delle malattie e prevenendo complicazioni acute [13].
Uno degli sviluppi più interessanti è l'uso di **reti neurali grafiche** (GNN) per predire

le interazioni future tra malattie e l'evoluzione della salute del paziente. Le reti GNN possono analizzare l'intera storia clinica di un paziente e prevedere quali condizioni potrebbero svilupparsi sulla base di modelli complessi di comorbidità. Questo è particolarmente utile nella gestione di malattie croniche, dove la previsione accurata delle condizioni future può migliorare la personalizzazione delle cure e ridurre il rischio di peggioramenti improvvisi [13].

2.7 Un caso di studio: comorbidità

Lo studio condotto dai dott. Cavallo, Pagano, De Santis e Capobianco [14] si inserisce nell'ambito dell'analisi della comorbidità attraverso un approccio basato su Big Data e reti di comorbidità. L'uso di dati provenienti dai General Practitioner Records (GPR), che comprendono informazioni sulle prescrizioni mediche e sui dati clinici di routine, consente una visione più ampia delle condizioni cliniche dei pazienti rispetto ai tradizionali Electronic Health Records (EHR), solitamente limitati ai pazienti ospedalizzati per malattie gravi.
Questo aspetto è particolarmente rilevante quando si tratta di condizioni croniche come il diabete, dove la presenza di altre patologie (comorbidità) complica la gestione terapeutica e richiede strategie di prevenzione mirate.
Lo studio ha esaminato un campione di 14.958 pazienti e 1.728.736 prescrizioni raccolte in un arco temporale di 10 anni. I dati riguardavano sia pazienti diabetici che non diabetici, e sono stati utilizzati per creare reti di comorbidità, dove ogni nodo rappresentava un gruppo di diagnosi (codici ICD9-CM) e gli archi indicavano la presenza contemporanea di queste diagnosi nello stesso paziente. Attraverso questo modello di rete, è stato possibile visualizzare e quantificare le associazioni tra diverse malattie e come queste variano in base a fattori come l'età e il sesso del paziente [14].
I risultati dello studio hanno confermato che la comorbidità tende ad aumentare con l'età del paziente e che i pazienti diabetici presentano un pattern di comorbidità significativamente più complesso rispetto ai non diabetici. Questa complessità, rappresentata graficamente, ha permesso di identificare malattie che spesso co-occorrono nei pazienti diabetici, fornendo così un utile strumento per migliorare la gestione clinica di queste persone [14].

In conclusione, l'approccio di rete utilizzato in questo studio ha dimostrato l'efficacia dei dati di prescrizione come strumento autonomo per l'analisi delle comorbidità, permettendo di anticipare trend epidemiologici e di supportare i processi decisionali dei responsabili delle politiche sanitarie. L'utilizzo di tecniche di **network analysis** applicate ai GPR potrebbe rappresentare una strategia scalabile e generalizzabile, applicabile a popolazioni più ampie, migliorando l'efficienza della sanità pubblica e la prevenzione delle complicanze legate alle malattie croniche [14].

2.8 Big Data

2.8.1 Big Data: una panoramica

Il termine *Big Data* si riferisce a grandi volumi di dati che non possono essere elaborati efficacemente utilizzando metodi tradizionali. I Big Data sono caratterizzati dalle cosiddette "cinque V": Volume, Varietà, Velocità, Veridicità e Valore.

Volume: Si tratta della quantità di dati generati ogni secondo in vari ambiti, dai social media ai dispositivi mobili fino a sensori e sistemi IoT. Il volume è forse l'aspetto più emblematico dei Big Data, poiché si parla di petabyte e zettabyte di informazioni prodotte globalmente ogni anno.

Varietà: I dati provengono da diverse fonti e sono di diversi tipi. Possono essere strutturati (ad esempio, tabelle di database), semi-strutturati (log di server web) o non strutturati (immagini, video, testo libero).

Velocità: La velocità si riferisce alla rapidità con cui i dati vengono generati e devono essere elaborati. Con tecnologie come i sensori e le reti di comunicazione, i dati possono essere raccolti e trasmessi in tempo reale.

Veridicità: Questo concetto si riferisce all'accuratezza e all'affidabilità dei dati raccolti. Non sempre i dati generati sono corretti o completi, quindi la gestione dei Big Data richiede tecniche per garantire la loro qualità.

Valore: Alla fine, la vera sfida dei Big Data è riuscire a estrarre informazioni utili e significative per prendere decisioni strategiche, un compito che richiede strumenti avanzati di analisi e capacità predittive.
L'emergere dei Big Data ha avuto un impatto su numerosi settori, dalla finanza alla logistica, ma uno dei campi in cui il loro utilizzo si è dimostrato più promettente è il settore sanitario.

2.8.2 Big Data nel contesto sanitario

Nel settore sanitario, i Big Data rappresentano una risorsa inestimabile per migliorare la diagnosi, la gestione delle malattie e l'efficienza operativa dei sistemi sanitari. La capacità di analizzare grandi quantità di dati provenienti da cartelle cliniche elettroniche (EHR), sensori medici, test di laboratorio, e fonti esterne come dati socio-economici e comportamentali, consente di creare una visione più completa della salute di un paziente o di una popolazione [15].
In particolare, i Big Data hanno cambiato il modo in cui si affrontano le malattie croniche e complesse, come le comorbidità. Analizzando grandi dataset provenienti da popolazioni diverse, i ricercatori possono identificare pattern di malattia, monitorare l'efficacia dei trattamenti e prevedere esiti clinici futuri. Inoltre, i dati raccolti da dispositivi indossabili e sensori consentono di monitorare i pazienti in tempo reale, migliorando la prevenzione e l'intervento tempestivo.
Uno degli esempi più rilevanti dell'applicazione dei Big Data in sanità è l'uso delle reti comorbide per studiare la co-occorrenza delle malattie, come discusso nel nostro caso di studio sulla comorbidità[14]. Grazie alla combinazione di dati clinici e tecniche di analisi di rete, è possibile identificare gruppi di malattie che tendono a manifestarsi insieme, migliorando la capacità di predire l'evoluzione della salute di

un paziente e personalizzare il trattamento.
Oltre alle applicazioni cliniche, i Big Data consentono alle autorità sanitarie di prendere decisioni informate sulla gestione delle risorse, l'allocazione dei fondi e la pianificazione delle politiche sanitarie. Ad esempio, i dati epidemiologici possono essere analizzati per monitorare la diffusione delle malattie infettive e prevedere focolai futuri, consentendo una risposta più efficace e tempestiva.
In sintesi, i Big Data nel contesto sanitario non solo migliorano la qualità delle cure per i singoli pazienti, ma aiutano a ottimizzare l'intero sistema sanitario, rendendolo più efficiente e proattivo.

2.9 Pre-Processing dei Dati

Il *data cleaning* è una fase essenziale in qualsiasi processo di analisi dei dati, soprattutto quando si tratta di dati sanitari. Questa operazione consiste nella rimozione o correzione di dati inconsistenti, duplicati o incompleti, garantendo che i dataset utilizzati per l'analisi siano accurati e affidabili. Nel contesto sanitario, dove le informazioni provengono da molteplici fonti (cartelle cliniche elettroniche, prescrizioni, registri ospedalieri), la pulizia dei dati diventa ancora più critica per prevenire errori nell'analisi e garantire l'integrità delle previsioni basate sui modelli di *machine learning*.
La preparazione dei dataset implica anche la normalizzazione e l'organizzazione dei dati in formati che consentano l'applicazione di tecniche di analisi avanzata, come la *network analysis* e i modelli di *machine learning*. Nel caso della comorbidità, la corretta gestione dei codici diagnostici (come ICD9-CM) e delle prescrizioni mediche è fondamentale per costruire reti di comorbidità affidabili e accurate.

2.10 Data Visualization e manipolazione dei grafi clinici

La *data visualization* è uno strumento cruciale per esplorare e comprendere i pattern nascosti nei dati complessi, specialmente in ambito sanitario. Le tecniche di visualizzazione dei dati consentono di rappresentare graficamente le reti di comorbi-

dità, evidenziando le connessioni tra malattie e pazienti e facilitando l'interpretazione delle interazioni cliniche.

L'uso di grafi, in particolare, è fondamentale per rappresentare visivamente le relazioni tra malattie, rendendo evidenti i nodi centrali (malattie *hub*) e le loro connessioni. Grazie alla visualizzazione grafica, è possibile individuare rapidamente malattie che svolgono un ruolo chiave nella rete di comorbidità e identificare pattern che potrebbero non essere evidenti attraverso l'analisi statistica tradizionale.

Inoltre, la manipolazione dei grafi clinici permette ai ricercatori di esplorare scenari simulativi, modificando le connessioni tra i nodi e osservando come i cambiamenti nelle relazioni tra malattie possano influire sulla salute del paziente. Questi strumenti forniscono un approccio visivo e dinamico all'analisi clinica, migliorando il processo decisionale e facilitando la comunicazione tra medici e ricercatori.

CAPITOLO 3

Medodologia ed Implementazione

3.1 Introduzione

La crescente importanza dei Big Data nel settore sanitario ha trasformato il modo in cui vengono gestiti, analizzati e utilizzati i dati per supportare le decisioni cliniche. Questo fenomeno è particolarmente rilevante nello studio delle comorbidità, che si riferiscono alla coesistenza di più malattie in un singolo paziente. In questo contesto, l'obiettivo del presente capitolo è descrivere il processo metodologico e tecnico che ha portato alla creazione di ComorGraph, una piattaforma progettata per analizzare, visualizzare e prevedere la comorbidità utilizzando tecniche di Social Network Analysis (SNA).

ComorGraph è stata sviluppata non solo come uno strumento di visualizzazione, ma come un sistema avanzato di analisi che consente di esplorare i complessi modelli di comorbidità all'interno di vasti dataset medici. L'intero processo, dalla raccolta dei dati alla loro elaborazione e interpretazione, è stato guidato dalla necessità di fornire un sistema efficiente e scalabile per il settore medico, capace di gestire grandi volumi di dati e supportare i medici nel processo decisionale.

L'implementazione della piattaforma è stata orientata verso scelte tecniche mirate a garantire flessibilità e prestazioni elevate, senza sacrificare la facilità d'uso. In

particolare, il sistema è stato progettato per essere user-friendly e al contempo offrire strumenti analitici potenti, in grado di sfruttare le potenzialità di Neo4j come database a grafo, la reattività di React per l'interfaccia utente, e la robustezza di Python per il backend.
Nel presente capitolo verranno descritti i principali passaggi della progettazione e implementazione di ComorGraph, a partire dai requisiti funzionali fino alla scelta delle tecnologie, con particolare attenzione al processo di ottimizzazione del sistema per gestire le complessità dei dati clinici. Ci saranno in fine due sezioni, quella relativa alla pulizia e analisi dei dati, e quella relativa al calcolo e lo studio delle metriche SNA realive alla piattaforma.

3.2 Requisiti Funzionali

Per garantire che la piattaforma ComorGraph soddisfi le esigenze del contesto medico in cui è stata sviluppata, sono stati definiti dei requisiti funzionali specifici. Questi requisiti definiscono cosa il sistema deve fare per supportare l'analisi delle comorbidità in maniera efficiente, permettendo ai medici di visualizzare e analizzare i dati clinici con precisione.

Obiettivi dei Requisiti Funzionali

- **Visualizzazione grafi clinici**: Permettere la rappresentazione visiva delle relazioni tra pazienti, malattie e prescrizioni, con la possibilità di esplorare i dettagli di ogni nodo e arco.
- **Applicazione di metriche SNA**: Fornire metriche di analisi delle reti come betweenness, closeness e k-core, essenziali per l'analisi delle comorbidità e per identificare malattie centrali o particolarmente connesse.
- **Interattività e analisi temporale**: Consentire l'analisi dinamica dei dati nel tempo, in modo da osservare come le comorbidità si evolvono per ogni paziente.
- **Gestione e caricamento di database personalizzati**: Dare la possibilità di caricare e cambiare database dinamicamente tramite file CSV strutturati.

Tabella dei Requisiti Funzionali

In seguito riporto una porzione della tabella originale dei requisiti funzionali.

Tabella 3.1: Requisiti Funzionali di ComorGraph

ID	Nome	Descrizione	Priorità
RF_01	Visualizzazione Dashboard	L'utente deve poter visualizzare un riepilogo con il numero di pazienti, prescrizioni e malattie.	Alta
RF_02	Visualizzazione Grafo Paziente	L'utente deve poter esplorare il grafo del singolo paziente e la sua storia clinica nel tempo.	Alta
RF_03	Analisi Temporal Slider	L'utente deve poter utilizzare uno slider temporale per analizzare l'andamento clinico del paziente.	Media
RF_04	Visualizzazione Grafo Malattie	L'utente deve poter visualizzare le associazioni tra malattie in un grafo interattivo.	Alta
RF_05	Applicazione Metriche SNA	L'utente deve poter applicare metriche come betweenness, closeness, e k-core sui grafi visualizzati.	Alta
RF_06	Creazione Dinamica Database	L'utente deve poter caricare CSV formattati per creare un nuovo database di analisi.	Media
RF_07	Switch Database	L'utente deve poter cambiare tra diversi dataset caricati per analisi comparative.	Media
RF_08	Pannello Dettagli Interattivo	L'utente deve poter visualizzare i dettagli dei nodi e degli archi selezionati all'interno dei grafi.	Alta

3.3 Requisiti Non Funzionali

I requisiti non funzionali descrivono le caratteristiche di qualità che la piattaforma ComorGraph deve soddisfare per garantire un'esperienza d'uso ottimale e rispondere alle esigenze tecniche del contesto medico.

Obiettivi dei Requisiti Non Funzionali

- **Facilità di utilizzo**: L'interfaccia utente deve essere intuitiva e semplice da utilizzare anche per medici e ricercatori non tecnici.
- **Efficienza**: Le operazioni di analisi sui grafi devono essere eseguite rapidamente, anche in presenza di grandi volumi di dati.
- **Scalabilità**: Il sistema deve poter gestire un numero crescente di dati senza compromettere le prestazioni.
- **Sicurezza**: Poiché si tratta di dati clinici sensibili, il sistema deve garantire che l'accesso ai dati sia sicuro e che le informazioni vengano crittografate.

Tabella dei Requisiti Non Funzionali

Tabella 3.2: Requisiti Non Funzionali di ComorGraph

ID	Nome	Descrizione	Priorità
RNF_01	Facilità di Utilizzo	L'interfaccia deve essere intuitiva e utilizzabile senza una formazione tecnica approfondita.	Alta
RNF_02	Efficienza	Le operazioni di analisi devono essere eseguite rapidamente, anche con dataset di grandi dimensioni.	Alta
RNF_03	Scalabilità	Il sistema deve gestire senza problemi dataset di dimensioni crescenti, fino a milioni di record.	Media
RNF_04	Sicurezza	I dati clinici sensibili devono essere protetti da accessi non autorizzati mediante crittografia.	Alta
RNF_05	Compatibilità	La piattaforma deve essere accessibile tramite i principali browser.	Media

3.4 Motivazione delle Scelte Tecnologiche

Le tecnologie selezionate per lo sviluppo della piattaforma **ComorGraph**: **React**, **Python** e **Neo4j**, rappresentano strumenti moderni e consolidati, ampiamente utilizzati nel settore tecnologico e con una vasta gamma di librerie e strumenti che facilitano lo sviluppo di soluzioni complesse come l'analisi della comorbidità.

React è una libreria frontend moderna e robusta, ampiamente utilizzata per creare interfacce utente dinamiche e interattive. È particolarmente apprezzata per la sua flessibilità, scalabilità e capacità di gestire aggiornamenti efficienti dell'interfaccia, rendendola ideale per la visualizzazione di grafi e dati clinici complessi

in modo intuitivo. La sua vasta comunità di sviluppatori e il supporto di numerose librerie ne fanno una scelta naturale per lo sviluppo di applicazioni avanzate.

Python è stato scelto per il backend grazie alla sua semplicità e potenza nel campo della manipolazione dei dati e del machine learning. È dotato di numerose librerie scientifiche (come **Pandas**, **NumPy**, **Scikit-learn**) che permettono di gestire, analizzare e processare grandi dataset sanitari senza compromessi in termini di flessibilità. Python è ampiamente utilizzato nel mondo accademico e industriale, il che lo rende una scelta ideale per integrare tecniche avanzate di analisi dei dati e machine learning.

Neo4j, il database a grafo, rappresenta una scelta eccellente per lo studio della comorbidità, poiché permette di modellare le relazioni complesse tra malattie, pazienti e prescrizioni in modo naturale ed efficiente. Con un ampio set di funzionalità integrate, come il linguaggio di query **Cypher** e la **Graph Data Science Library**, Neo4j fornisce un supporto diretto per l'applicazione di metriche di Social Network Analysis, consentendo analisi rapide e intuitive di reti complesse.

L'integrazione di queste tecnologie, solide e mature, assicura che la piattaforma possa gestire efficientemente l'analisi della comorbidità e offrire un'esperienza utente avanzata e altamente interattiva.

3.5 Architettura della Piattaforma

3.5.1 Database (Neo4j)

Neo4j è uno dei più avanzati database a grafo attualmente disponibili, utilizzato principalmente per modellare dati con relazioni complesse tra entità. A differenza dei tradizionali database relazionali, che rappresentano le relazioni tramite tabelle e join, Neo4j rappresenta i dati in forma di grafo, dove nodi ed archi costituiscono le entità e le loro connessioni dirette. Questa struttura consente una rappresentazione più naturale e intuitiva di dati che presentano molte interconnessioni, rendendolo ideale

per l'analisi delle reti e delle strutture complesse, come ad esempio le reti sociali o, nel nostro caso, le **relazioni tra pazienti, malattie e prescrizioni**.

Potenzialità di Neo4j nella Rappresentazione dei Grafi

1. **Efficienza nel gestire relazioni complesse**: le operazioni che coinvolgono molte relazioni, come il calcolo di collegamenti tra malattie in un contesto di comorbidità, sono molto più rapide rispetto a database relazionali tradizionali.

2. **Visualizzazione immediata delle connessioni**: semplifica notevolmente l'interpretazione delle relazioni tra entità (ad esempio, i collegamenti tra malattie basate su prescrizioni condivise).

3. **Facilità di interrogazione con Cypher**: linguaggio specificamente progettato per interrogare grafi. Cypher consente di eseguire ricerche complesse. La sintassi è molto più intuitiva rispetto a quella SQL, specialmente per domande complesse che richiederebbero numerosi join in un database relazionale.

4. Supporto a **Graph Data Science Library**: Uno dei principali vantaggi di Neo4j è l'integrazione nativa con la Graph Data Science Library (GDSL), un plugin che fornisce una vasta gamma di algoritmi per l'analisi delle reti sociali e la teoria dei grafi. Questo plugin supporta algoritmi di clustering, centralità, analisi delle componenti connesse, rilevamento di comunità, predizione di link, e molto altro. Alcune delle metriche più utilizzate includono: **Betweeness Centrality**, **Closeness Centrality**, **Page Rank**, algoritmo **K-Core**.

Metodologie di Rappresentazione: Dallo Studio Ereditato alla Nuova Strategia

Lo studio iniziale del dott. Cavallo e Giordano[5], che ha ispirato il nostro lavoro, era basato su una rappresentazione della comorbidità, in cui le malattie erano i nodi principali e le prescrizioni condivise rappresentavano gli archi che collegavano queste malattie. In tale struttura, ogni arco conteneva tutte le informazioni relative alle prescrizioni comuni tra le malattie, rappresentando un approccio che, sebbene logico, ha rivelato significativi limiti man mano che la mole di dati aumentava.

Limiti della Strategia Iniziale Con l'aumentare delle dimensioni del dataset, questo tipo di rappresentazione ha cominciato a mostrare diverse inefficienze:

- **Complessità computazionale**: Con il crescere del numero di nodi (malattie) e degli archi (prescrizioni condivise), le query per l'estrazione di informazioni specifiche sono diventate molto onerose dal punto di vista computazionale. L'esecuzione di queste query comportava tempi di attesa prolungati e l'impossibilità di gestire grandi volumi di dati in modo efficiente.
- **Ridotta scalabilità**: Quando il numero di relazioni tra malattie superava determinate soglie, la gestione delle query diveniva impraticabile, rendendo estremamente difficile estrapolare le informazioni utili dai dati. Ogni prescrizione condivisa creava nuovi archi tra le malattie, il che significava che, con un dataset esteso, il numero di relazioni cresceva in maniera esponenziale.

Questa strategia aveva dunque un'efficienza limitata nella gestione di grandi volumi di dati e, ancor più importante, nella possibilità di ricavare informazioni utili e specifiche riguardanti le relazioni tra pazienti, malattie e prescrizioni.

Passaggio alla Nuova Strategia di Rappresentazione Per risolvere le limitazioni sopra citate, è stata adottata una nuova strategia di rappresentazione che ha permesso di separare meglio i dati, facilitando le operazioni di query e visualizzazione. Questa nuova strategia si basa su una struttura che include tre tipi principali di nodi:

1. **Paziente**: Nodo identificato dal campo univoco *codice fiscale assistito*. Questo nodo rappresenta il paziente e permette di mantenere una relazione diretta con le prescrizioni e le malattie.
2. **Malattia**: Nodo identificato dal codice ICD9-CM, rappresenta ciascuna malattia nel dataset. L'informazione relativa alle malattie è stata mantenuta, ma in una struttura che facilita la correlazione con i pazienti e le prescrizioni.
3. **Prescrizione**: Nodo identificato dal *codice prescrizione*, contenente tutte le informazioni necessarie a descrivere i farmaci o le terapie somministrate al paziente.

Questa rappresentazione a tre nodi ha ridotto drasticamente la complessità delle query e ha permesso di mantenere separati i diversi tipi di informazioni in modo che fossero più facilmente accessibili e interpretabili.

Relazioni tra Nodi Le relazioni tra questi nodi sono state gestite attraverso tre tipi principali di archi:

1. **DIAGNOSTICATO_CON**: Una relazione tra un paziente e una malattia, che indica quali malattie sono state diagnosticate al paziente. Questa relazione non si limita a indicare la presenza della malattia, ma include anche informazioni aggiuntive come la *data della prima diagnosi*, *data dell'ultima diagnosi* e il *conteggio delle diagnosi ripetute*.

2. **CURATA_CON**: Questa relazione collega le malattie alle prescrizioni, indicando quali farmaci sono stati utilizzati per trattare specifiche patologie. Questa informazione è fondamentale per analizzare l'efficacia dei trattamenti e per correlare le malattie tra di loro attraverso prescrizioni comuni.

3. **ASSOCIATA_A**: Una relazione tra malattie che rappresenta la comorbidità, ovvero la tendenza di determinate malattie a presentarsi contemporaneamente nello stesso paziente.
   È doveroso citare che la realizzazione di questa relazione è basata sul medesimo studio[5] in cui due malattie sono associate tra di loro quando vengono trattate contemporaneamente attraverso la stessa prescrizione per lo stesso paziente.

Riduzione della Complessità Computazionale La nuova strategia di rappresentazione ha consentito una notevole riduzione della complessità computazionale. Con la separazione dei nodi e la semplificazione delle relazioni, le query per estrarre informazioni specifiche sono diventate molto più rapide ed efficienti. Inoltre, la rappresentazione a tre nodi ha permesso una visualizzazione più intuitiva e comprensibile delle relazioni tra pazienti, malattie e prescrizioni, rendendo più agevole l'analisi dei pattern di comorbidità.

Limiti Hardware e Scelte del Dataset Ridotto Nonostante la nuova struttura di rappresentazione abbia ridotto in modo significativo la complessità delle operazioni sul database, il volume dei dati rimaneva ancora una sfida. Gestire 17 milioni di record

su un database locale ha portato a limitazioni hardware, con difficoltà nella gestione delle risorse e nel tempo di esecuzione delle query. Per questo motivo, è stata presa la decisione di lavorare su un dataset ridotto per la piattaforma, riducendo il numero di record a circa 40.000 entry. Questo dataset è stato creato con un approccio bilanciato che ha mantenuto comunque la rappresentatività dei dati clinici per l'analisi della comorbidità. I dettagli relativi alla creazione di questo dataset verranno trattati nella sezione 3.6.

Vantaggi della Nuova Rappresentazione Questa rappresentazione ha consentito:

- Una maggiore efficienza nell'elaborazione delle query.
- Un miglioramento delle capacità di visualizzazione e analisi.
- La possibilità di eseguire calcoli di metriche di Social Network Analysis (SNA) direttamente all'interno di Neo4j grazie all'uso della **Graph Data Science Library (GDSL)**, che include metriche essenziali come *degree centrality*, *betweenness*, *closeness*, *PageRank* e *K-core*.

In conclusione, la nuova struttura di rappresentazione ha permesso di ottenere un notevole miglioramento delle prestazioni e dell'efficienza nell'analisi delle relazioni comorbili tra malattie, garantendo al contempo la possibilità di gestire e visualizzare dati clinici complessi in modo ottimizzato.

3.5.2 Backend (Python)

Il backend della piattaforma *ComorGraph* è basato su *Python*, con un'architettura organizzata secondo il pattern *Service-Model-Routes*. Questo approccio permette una chiara separazione delle responsabilità, facilitando la gestione, la manutenzione e l'espansione del codice.

Struttura: Service-Model-Routes

Models: Qui vengono definiti i modelli che interagiscono direttamente con il database *Neo4j*. Ogni funzione presente nel modello esegue operazioni *CRUD* (Create, Read, Update, Delete) o query particolari sul database.

- **graph_model.py**: gestisce il recupero dei grafi per pazienti, prescrizioni e malattie. Questo file contiene le query dirette per *Neo4j* che estraggono i dati in base a vari criteri.
- **utils_model.py**: definisce funzioni ausiliarie, come il conteggio di pazienti, prescrizioni e malattie, e metriche di analisi della rete (degree centrality, betweenness, closeness, etc.).

Services: Contiene la logica di business che utilizza i metodi dei modelli per eseguire operazioni specifiche. I servizi agiscono come intermediari tra i modelli e le rotte, incapsulando la logica di alto livello.

- **graph_service.py**: gestisce l'interazione con *graph_model.py*, fornendo servizi per il recupero dei grafi (paziente, prescrizione, malattia).
- **utils_service.py**: si occupa di elaborare i dati di contatori e metriche e fornisce funzionalità di caricamento *CSV* per la creazione dinamica di database.

Routes: Qui si definiscono le *API REST* che gestiscono le richieste *HTTP* da parte del frontend. Le rotte richiamano i servizi e ritornano i dati in formato *JSON*.

- **graph_routes.py**: definisce le rotte per ottenere i grafi per pazienti, prescrizioni e malattie.
- **utils_routes.py**: gestisce le *API* per il caricamento dei *CSV* e per ottenere informazioni generali come i contatori di pazienti, prescrizioni, e malattie.

Gestione del Database Neo4j

Il file *extensions.py* si occupa della connessione con *Neo4j*. Qui viene implementata la logica per stabilire la connessione con il database e gestire il passaggio tra database dinamici.

- **init_app**: Inizializza la connessione al database con le credenziali appropriate.
- **switch_database**: Permette di cambiare il database attivo, utile nel contesto della piattaforma per gestire diversi dataset clinici.
- **close**: Chiude la connessione al database.

Librerie Utilizzate

Alcune delle librerie fondamentali del backend includono:

Flask: Utilizzato per creare l'*API RESTful* che comunica con il frontend. *Flask* è un micro-framework che consente di gestire facilmente richieste *HTTP*.

Flask-CORS: Gestisce le politiche di *Cross-Origin Resource Sharing*, permettendo al frontend (che potrebbe essere su un dominio differente) di accedere alle risorse del backend senza problemi di sicurezza legati al *CORS*.

Neo4j: Il driver *Python* per connettersi a *Neo4j* e gestire le query a grafo, che vengono eseguite per recuperare dati complessi sulle relazioni di comorbidità.

Werkzeug: Utilizzata per funzioni di sicurezza come *secure_filename*, che garantisce che i file caricati abbiano nomi sicuri.

Spiegazione del Flusso di Dati

Il backend interagisce con il database *Neo4j* principalmente attraverso due servizi: *graph_service.py* e *utils_service.py*. Questi servizi chiamano i modelli che eseguono le query *Neo4j*, restituendo al frontend i grafi relativi a pazienti, prescrizioni, malattie e le metriche di analisi (come *betweenness centrality* e *page rank*).
Le rotte definite in *routes* ricevono le richieste *HTTP* dal frontend, richiamano i servizi appropriati, e ritornano i risultati come *JSON*, che saranno utilizzati per la visualizzazione grafica nel frontend.

Integrazione con il Frontend

L'integrazione tra frontend (*React*) e backend (*Python*) avviene tramite le *API RESTful*. *React* invia richieste *HTTP GET* o *POST* al backend, che risponde con dati strutturati. Questi dati vengono poi visualizzati in modo interattivo grazie a librerie come *Vis.js* per la gestione dei grafi, garantendo che il medico possa visualizzare relazioni complesse tra malattie, pazienti e prescrizioni.
Questo approccio, basato su servizi modulari è una chiara separazione tra business logic e gestione dei dati, garantisce una manutenzione facilitata e una scalabilità del

sistema, consentendo alla piattaforma di gestire grandi quantità di dati clinici in modo efficiente e strutturato.

3.5.3 Frontend (React)

Struttura e Architettura del Frontend

La struttura del frontend è organizzata in una serie di moduli distinti, che facilitano lo sviluppo collaborativo e la suddivisione delle responsabilità del codice. Le cartelle principali includono:

- **assets**: Questa cartella contiene tutte le risorse statiche utilizzate dall'applicazione, come immagini e icone. La gestione delle risorse in modo centralizzato contribuisce a mantenere l'applicazione organizzata e facilmente scalabile.
- **components**: Il cuore dell'architettura del frontend è rappresentato dai componenti *React*, che comprendono elementi come *GraphComponent* (dedicato alla visualizzazione dei grafi clinici), *Sidebar*, e *DetailsPanel*. Ogni componente segue il principio di responsabilità singola, il che significa che ogni modulo è progettato per eseguire una funzione specifica e ben definita.
- **pages**: Le pagine dell'applicazione includono viste come *Homepage*, *PatientPage*, *GraphPage*, e *PrescriptionPage*. Queste pagine gestiscono le diverse sezioni della piattaforma, connesse tra loro tramite il *React Router*, che permette una navigazione fluida e dinamica tra i vari moduli della piattaforma.
- **services**: I servizi gestiscono le richieste asincrone al backend per il recupero di dati. Questo approccio consente una separazione tra la logica di presentazione e la logica di accesso ai dati, migliorando la manutenibilità del codice.

Gestione della Visualizzazione dei Grafi

Uno dei punti di forza della piattaforma *ComorGraph* è la capacità di visualizzare reti complesse che rappresentano le relazioni tra pazienti, malattie e prescrizioni. Questo è reso possibile grazie a *Vis.js*, libreria leader per la gestione dei grafi.
Il *GraphComponent* gestisce la visualizzazione dei nodi (che rappresentano pazienti,

malattie o prescrizioni) e degli archi (che rappresentano le relazioni tra di loro). La logica sottostante è contenuta in *GraphComponentLogic.js*, dove i dati ricevuti dal backend vengono trasformati in strutture grafiche interattive.
Ogni nodo è rappresentato visivamente con un'icona e un colore distintivo per facilitarne l'identificazione immediata. I nodi paziente, malattia e prescrizione vengono rappresentati con colori distinti: rosso (Paziente), blu (Malattia) e verde (Prescrizione). Questa scelta facilita la comprensione delle relazioni tra i dati clinici in modo visivo e intuitivo.
Il grafo è dinamico e interattivo: l'utente può zoomare, spostarsi e cliccare sui nodi e archi per ottenere dettagli aggiuntivi. Inoltre, è disponibile la funzionalità per calcolare le metriche *SNA* (*Social Network Analysis*) e applicarle sul grafo corrente, che viene aggiornato ridimensionando i nodi malattia in base al valore della metrica desiderata.

Gestione delle Richieste al Backend

La comunicazione tra il frontend e il backend è gestita da una serie di *service* definiti in file separati, come *graphDataService.js* e *utilsDataService.js*. Questi servizi utilizzano il metodo *fetch* per inviare richieste al backend e recuperare i dati necessari per la visualizzazione e l'interazione nell'interfaccia utente.

- **graphDataService**: gestisce le richieste per ottenere il grafo completo o i singoli grafi per pazienti, malattie e prescrizioni.
- **utilsDataService**: si occupa di fornire i risultati delle metriche di *SNA* e della gestione dei dati generali, come: il caricamento di nuovi file *CSV*, la creazione dinamica del database e il recupero delle statistiche globali dell'applicazione.

Vite come Strumento di Build

Il frontend di *ComorGraph* è stato sviluppato utilizzando *Vite*, un moderno strumento di build per applicazioni web. *Vite* è stato scelto in quanto offre numerosi vantaggi rispetto a strumenti più tradizionali come *Create React App* (CRA):

Velocità di sviluppo: *Vite* utilizza *ESBuild* per gestire la fase di sviluppo, il che si traduce in tempi di build e reload significativamente più rapidi, specialmente per progetti di grandi dimensioni. Permette di effettuare modifiche real-time, salvare ed effettuare la compilazione solo del modulo richiesto, visualizzando quasi immediatamente i risultati della modifica.

Ottimizzazione del codice: Grazie al suo sistema di bundling, *Vite* ottimizza il codice per la produzione in modo più efficiente, riducendo i tempi di caricamento e migliorando le performance dell'applicazione.

Supporto avanzato per le librerie: *Vite* supporta nativamente molte librerie moderne e permette una configurazione flessibile tramite il file *vite.config.js*, consentendo una personalizzazione ottimale del processo di build.

In questo contesto, *Vite* è stato scelto non solo per la sua rapidità ma anche per la sua capacità di gestire applicazioni con un elevato numero di dipendenze, come quella di *ComorGraph*, che si affida a numerose librerie per la gestione grafica e la visualizzazione dei dati clinici.
Con questo approccio, la piattaforma *ComorGraph* offre una soluzione robusta e scalabile per l'analisi della comorbidità, utilizzando le migliori tecnologie per garantire performance elevate, una user experience fluida, e l'interattività richiesta per l'analisi di dati clinici complessi.

3.6 Pre-processing e Bilanciamento del Dataset Clinico

Nell'ambito della costruzione della piattaforma **ComorGraph**, il pre-processing dei dati ha rappresentato una fase preliminare cruciale per garantire l'affidabilità e la qualità delle analisi successive. Il lavoro trattato nella tesi del mio collega Barba Gianfranco è concentrato in particolare sulla pulizia e il bilanciamento di un vasto dataset clinico, composto da informazioni sanitarie. Questo processo ha assicurato che i dati fossero pronti per le successive analisi predittive, che costituiscono una parte fondamentale del progetto. Il dataset iniziale, composto da milioni di record, rifletteva una complessa eterogeneità di variabili, inclusi dati demografici, prescrizioni mediche

e diagnosi di patologie. L'obiettivo del pre-processing era quindi duplice: da un lato, migliorare la qualità dei dati attraverso operazioni di pulizia; dall'altro, garantire un bilanciamento ottimale delle variabili chiave, per evitare distorsioni e bias nei modelli predittivi successivi.

Normative e Contesto dei Dati I dati utilizzati per l'analisi sono stati ottenuti grazie alla collaborazione con cinque diverse ASL della regione Campania, sotto la supervisione del dott. Pierpaolo Cavallo. Questa raccolta di informazioni cliniche ha fornito una solida base per costruire un dataset in grado di riflettere le comorbilità tra malattie e le relative prescrizioni mediche. I dati comprendono diverse tipologie di informazioni, tra cui prescrizioni, dati di fragilità, anagrafici e informazioni sui medici, ciascuna suddivisa in file CSV specifici. È importante osservare già da adesso che tutte le informazioni sensibili dei pazienti e medici sono già state fornite in forma anonima dalla sorgente, motivo per cui non è stata necessaria alcuna crittografia dei dati ricevuti.
La raccolta di questi dati clinici è stata resa possibile dal processo di digitalizzazione delle prescrizioni mediche, avviato in seguito al **Decreto Legge n. 269/2003**, convertito nella **Legge n. 326/2003**, che ha istituito il **Sistema Tessera Sanitaria**. A partire dal **2004**, la digitalizzazione ha facilitato una gestione più efficiente delle informazioni sanitarie, ma ha anche posto l'esigenza di una revisione approfondita dei dati storici. Il dataset includeva infatti dati risalenti a periodi precedenti al 2004, molti dei quali soggetti a errori dovuti al trasferimento manuale delle informazioni dai registri cartacei ai sistemi digitali. Ciò ha reso necessario un rigoroso processo di data cleaning, che ha comportato la correzione di errori nei codici diagnostici e nei campi demografici, nonché il controllo della coerenza interna del dataset.

Lavoro di Pre-processing e Bilanciamento dei Dati Il lavoro di pre-processing dei dati ha richiesto l'esecuzione di una serie di operazioni specifiche, tra cui la rimozione di valori mancanti, la correzione di errori nei codici **ICD9-CM**, e la gestione delle righe duplicate. Particolare attenzione è stata data alla variabile relativa al **codice diagnostico ICD9-CM**, dove si è riscontrata la presenza di errori tipografici che avrebbero potuto compromettere l'accuratezza delle analisi. Attraverso un processo

di sostituzione automatica e verifica incrociata con fonti scientifiche, sono stati corretti gli errori più comuni, come la sostituzione errata della lettera "O" al posto del numero "0".

Oltre alla pulizia, è stato necessario garantire che il dataset fosse **bilanciato** in termini di variabili fondamentali come età, sesso, e numero di prescrizioni per paziente. Queste variabili presentavano una distribuzione non omogenea nel dataset originale, che avrebbe potuto influenzare negativamente le analisi successive. Per esempio, la predominanza di pazienti anziani nel dataset avrebbe potuto portare a distorsioni nell'analisi delle comorbilità. Per correggere questo squilibrio, sono state adottate tecniche di **resampling**, sia attraverso l'oversampling di classi sottorappresentate sia con l'undersampling delle classi sovrarappresentate.

Il risultato di questa fase ha prodotto un dataset equilibrato, in grado di rappresentare adeguatamente la popolazione clinica e di supportare efficacemente le successive fasi di analisi e modellazione. La rimozione di outlier e la gestione delle variabili demografiche ha migliorato significativamente la robustezza del dataset, rendendolo una solida base per l'analisi mediante SNA e la costruzione dei modelli di machine learning.

Conclusioni Per una trattazione più specifica e approfondita delle tecniche di pre-processing applicate al dataset clinico, è possibile consultare la tesi del collega Barba Gianfranco. Tale documento esplora nel dettaglio i processi di data cleaning e bilanciamento, fornendo una base solida per lo sviluppo di un dataset affidabile e robusto. Questo lavoro rappresenta un passaggio fondamentale per le analisi predittive trattate nel Capitolo 6, dove verranno impiegate tecniche avanzate di machine learning su grafi eterogenei per l'analisi delle comorbilità. Grazie a un accurato pre-processing, il dataset risulta adatto all'identificazione di pattern clinici complessi e offre una prospettiva innovativa per l'analisi dei dati sanitari.

3.7 Social Network Analysis (SNA) per l'Analisi della Comorbidità

La Social Network Analysis (SNA) è una metodologia utilizzata per studiare le interazioni tra entità in un sistema complesso attraverso l'uso di grafi. In ambito medico, e più specificamente nello studio della comorbidità, la SNA permette di rappresentare le malattie come nodi di una rete, con relazioni che le collegano in base alla loro co-presenza nei pazienti. Questo approccio visivo e quantitativo fornisce un'analisi approfondita dei pattern di comorbidità, individuando come certe malattie tendano a manifestarsi insieme e come queste interazioni possano influenzare la gestione clinica.

Social Network Analysis su ComorGraph Nella piattaforma ComorGraph, la SNA è stata applicata per analizzare le reti di comorbidità tra patologie, facilitando l'individuazione di malattie centrali o connesse strategicamente ad altre. Le metriche principali trattate nella piattaforma sono:

Degree Centrality Misura il numero di connessioni dirette che un nodo ha con altri nodi, indicando malattie che tendono a presentarsi frequentemente con altre patologie.

Betweenness Centrality Identifica i nodi che fungono da "ponti" tra diverse sezioni della rete, segnalando malattie che collegano cluster distinti di patologie.

Closeness Centrality Valuta la vicinanza di un nodo a tutti gli altri, evidenziando malattie che possono influenzare rapidamente altre patologie nella rete.

PageRank Utilizzato per misurare l'importanza relativa di un nodo in base alla qualità delle sue connessioni, non solo al numero, e permette di identificare malattie con un'influenza clinica rilevante.

K-Core Algoritmo che individua gruppi di malattie fortemente interconnesse tra loro, facilitando l'identificazione di cluster di patologie che coesistono frequentemente in determinati gruppi di pazienti.

Conclusioni Per un'analisi più approfondita e dettagliata sull'applicazione della Social Network Analysis (SNA) e delle relative metriche nel contesto medico e all'interno della piattaforma ComorGraph, si rimanda alla tesi del collega Barba Gianfranco, che tratta l'argomento in maniera specifica e tecnica. Tale documento esplora i dettagli tecnici e le applicazioni cliniche della SNA, fornendo una prospettiva più ampia sulle potenzialità di questa metodologia nella pratica medica.

CAPITOLO 4

Heterogeneous Graph Neural Networks per la Predizione di Relazioni Paziente-Malattia

4.1 Introduzione

Negli ultimi anni, l'analisi delle reti complesse ha acquisito un ruolo di crescente rilevanza in diversi ambiti della ricerca scientifica, tra cui quello sanitario. In particolare, l'adozione di tecniche di Social Network Analysis (SNA) applicate alla salute ha aperto nuove prospettive nello studio delle malattie e delle loro correlazioni, permettendo di comprendere meglio le dinamiche della comorbidità tra pazienti affetti da patologie multiple. Tale approccio consente di individuare relazioni non evidenti tra patologie, che possono offrire preziose informazioni per la predizione di nuovi eventi patologici, migliorando l'efficacia delle decisioni cliniche e dei trattamenti. In questo contesto, le reti neurali grafiche (Graph Neural Networks, GNN) rappresentano una delle tecniche di punta per modellare, analizzare e predire relazioni in insiemi di dati strutturati a grafo. Le GNN, attraverso il meccanismo di propagazione delle informazioni tra i nodi e gli archi del grafo, sono in grado di catturare le dipendenze locali tra gli oggetti rappresentati e di apprendere rappresentazioni latenti significative per compiti di classificazione e predizione. In particolare, le reti neurali grafiche

eterogenee (Heterogeneous Graph Neural Networks, HeteroGNN) si dimostrano particolarmente efficaci in scenari complessi come quello della sanità, dove differenti tipologie di entità (ad esempio, pazienti e malattie) interagiscono tra loro attraverso relazioni complesse.
Il lavoro descritto in questo capitolo si inserisce nel filone di ricerca che applica le GNN per lo studio della comorbidità. Il fine principale è quello di sviluppare modelli capaci di predire la probabilità che un paziente sviluppi una determinata patologia, basandosi sulle diagnosi pregresse e su un ampio insieme di dati clinici. A tal proposito, sono stati implementati e confrontati due modelli di HeteroGNN che si differenziano per il tipo di convoluzione grafica adottata e la complessità della rappresentazione del grafo:

- Il primo modello è basato su convoluzioni eterogenee con SAGEConv, una variante delle reti neurali grafiche che aggrega le informazioni dei nodi vicini per generare rappresentazioni delle entità. Questo modello adotta una struttura più semplice e basilare del grafo, in cui i nodi rappresentano pazienti e malattie, con archi che indicano la relazione di diagnosi tra paziente e malattia. Tale struttura, sebbene efficace, non cattura alcuni aspetti temporali che potrebbero influenzare la progressione clinica delle malattie.

- Il secondo modello adotta una struttura più complessa basata su GATConv (Graph Attention Networks), che sfrutta un meccanismo di attenzione per pesare in modo diverso i nodi vicini. La particolarità di questo modello è l'aggiunta di un secondo tipo di relazione che collega i pazienti tra loro attraverso una dimensione temporale. Questa relazione temporale, che riflette la sequenza cronologica delle diagnosi, introduce un aspetto più "clinico" nella struttura del grafo, fornendo una rappresentazione più dettagliata dell'evoluzione della salute del paziente nel tempo. L'inclusione di questa feature permette al modello di cogliere la progressione delle malattie e di migliorare la capacità predittiva rispetto a eventi futuri.

L'obiettivo di questo capitolo è dunque presentare la metodologia utilizzata per l'implementazione di entrambi i modelli, descrivendo il dataset, le tecniche di preprocessing, le architetture delle reti e il processo di addestramento. Inoltre, verranno

discussi i risultati ottenuti, valutando le prestazioni dei due modelli attraverso un confronto approfondito delle metriche di performance. In particolare, l'analisi si focalizzerà su:

La costruzione del grafo eterogeneo in cui i nodi rappresentano pazienti e malattie, e gli archi rappresentano le diagnosi e, nel caso del secondo modello, anche la relazione temporale tra pazienti.

L'utilizzo di tecniche di train-test split e negative sampling per la suddivisione del grafo e la creazione di archi negativi per bilanciare il dataset.

Il confronto delle architetture basate su SAGEConv e GATConv valutandone le capacità di apprendimento e la precisione predittiva in termini di accuracy, ROC AUC, precision, recall e f1-score.

Attraverso questo confronto, si mira a fornire una valutazione empirica dell'efficacia di differenti architetture grafiche nel campo della predizione delle malattie, evidenziando i punti di forza e di debolezza di ciascun modello. Tali informazioni potranno contribuire a migliorare le capacità predittive dei modelli di machine learning nel contesto sanitario, favorendo un approccio personalizzato alla cura e alla gestione delle malattie complesse.

4.2 Descrizione del Dataset

4.2.1 Origine e Struttura del Dataset

Il dataset utilizzato per lo sviluppo dei modelli di machine learning è stato ottenuto da un insieme di dati sanitari contenenti informazioni su prescrizioni mediche e diagnosi associate a un gruppo di pazienti già precedentemente filtrati e puliti come descritto nel capitolo 3. Questi dati, derivati dalle cartelle cliniche, includono malattie classificate tramite il sistema di codifica ICD9-CM (International Classification of Diseases, Ninth Revision, Clinical Modification) e prescrizioni mediche rappresentate dai codici. Oltre a queste informazioni, sono disponibili anche dati anagrafici e demografici sui pazienti, che possono influire sulla loro evoluzione clinica. Le principali colonne presenti nel dataset includono:

Nome	Descrizione
CODICE_FISCALE_ASSISTITO	Un identificativo univoco anonimizzato che rappresenta ciascun paziente.
DATA_PRESCRIZIONE	La data in cui è stata emessa la prescrizione medica.
ICD9_CM	Il codice ICD9-CM che identifica la malattia diagnosticata.
CODICE_PRESCRIZIONE	Il codice del farmaco o del trattamento assegnato al paziente.
DESCRIZIONE_PRESCRIZIONE	Una descrizione testuale del farmaco prescritto o del trattamento.
ANNO_NASCITA	L'anno di nascita del paziente.
CAP_RESIDENZA	Il codice di avviamento postale della residenza del paziente, utile per eventuali analisi geografiche.
SESSO	Il sesso del paziente (M per maschio, F per femmina).
DISEASE_LABEL	Un'etichetta che categorizza ulteriormente la malattia in base a classificazioni cliniche specifiche.

Utilizzo di Dataset di Differenti Dimensioni Per l'addestramento e la valutazione dei modelli, sono stati utilizzati due dataset di dimensioni differenti, in funzione delle esigenze computazionali dei due modelli. Questo approccio ha permesso di bilanciare la complessità dei modelli con la capacità di elaborazione disponibile, garantendo comunque risultati significativi.

Modello basato su SAGEConv Per il modello basato su SAGEConv, è stato invece utilizzato un dataset più ampio, composto da 1 milione di record. La struttura del modello SAGEConv, che non utilizza meccanismi di attenzione ma un'aggregazione media delle informazioni dai nodi vicini, è meno onerosa dal punto di vista computazionale, consentendo di gestire un volume di dati maggiore. Questo ha permesso di sfruttare un dataset più ricco, aumentando la capacità del modello di apprendere le caratteristiche latenti dei dati e migliorare la

precisione nella predizione delle malattie. Utilizzando un dataset più ampio, il modello SAGEConv è stato in grado di catturare una maggiore varietà di relazioni tra pazienti e malattie, aumentando la sua generalizzabilità.

Modello basato su GATConv Per il modello con GATConv, è stato utilizzato un dataset ridotto, composto da 100.000 record. La scelta di questo sottoinsieme ridotto è stata dettata da considerazioni di natura computazionale: il modello GATConv richiede una gestione intensiva della memoria a causa del meccanismo di attenzione applicato ai nodi vicini. Questo processo risulta oneroso in termini di tempo di calcolo e consumo di memoria, soprattutto quando si utilizzano dataset di grandi dimensioni. Per tale motivo, l'adozione di un dataset ridotto ha consentito di ottimizzare i tempi di addestramento, mantenendo comunque una complessità sufficiente per ottenere risultati significativi. Nonostante la dimensione ridotta, il dataset da 100.000 record conserva la varietà di malattie e pazienti necessari per una valutazione accurata delle prestazioni del modello.

Giustificazione dell'Utilizzo di Dataset Differenti L'adozione di due dataset di dimensioni diverse risponde a precise necessità computazionali. Il modello GATConv, pur essendo più sofisticato nella gestione delle interazioni tra nodi grazie al meccanismo di attenzione, richiede maggiori risorse computazionali, soprattutto in presenza di un elevato numero di nodi e archi. Questo rende l'addestramento su dataset di grandi dimensioni proibitivo per alcune configurazioni hardware. D'altra parte, il modello SAGEConv, grazie alla sua architettura meno complessa, ha consentito di elaborare un dataset più ampio, sfruttando una maggiore quantità di informazioni per migliorare le capacità predittive. Tale approccio ha permesso di sfruttare al meglio le risorse disponibili, garantendo una buona efficienza computazionale senza sacrificare la qualità delle analisi e delle predizioni.

4.2.2 Selezione delle Colonne Rilevanti

Per l'implementazione dei modelli, è stato necessario selezionare le colonne del dataset più rilevanti ai fini dell'analisi. Le colonne **CODICE_FISCALE_ASSISTITO**

e **ICD9_CM** sono state utilizzate per costruire la mappatura univoca tra pazienti e malattie, trasformando ciascun valore categoriale in un indice numerico. La colonna **DATA_PRESCRIZIONE** è stata convertita in formato datetime per gestire correttamente le informazioni temporali, mentre le colonne **SESSO** e **ANNO_NASCITA** sono state preprocessate per creare caratteristiche aggiuntive sui pazienti, come il calcolo dell'età alla data della prescrizione. Nello specifico:

CODICE_FISCALE_ASSISTITO è stato trasformato in un indice numerico che rappresenta ciascun paziente in modo univoco all'interno del grafo.

ICD9_CM è stato anch'esso codificato numericamente per rappresentare ogni malattia come un nodo distinto nel grafo.

DATA_PRESCRIZIONE è stata utilizzata per calcolare l'età del paziente al momento della diagnosi, una caratteristica che è stata normalizzata per essere inclusa nelle feature dei nodi pazienti.

4.2.3 Preprocessing dei Dati

Il dataset originale conteneva dati non normalizzati, pertanto è stato necessario applicare una serie di tecniche di preprocessing per garantire la qualità dei dati e migliorare l'efficacia dei modelli.

Normalizzazione delle Feature Le caratteristiche numeriche sono state standardizzate utilizzando lo StandardScaler. Questo ha permesso di garantire che le variabili numeriche (come età e codifica del sesso) avessero una media pari a zero e una deviazione standard pari a uno, migliorando così la convergenza del modello durante l'addestramento. Per i nodi pazienti, la feature Sesso è quella interessata ed è stata codificata come valore numerico (0 per maschio, 1 per femmina). Per i nodi malattie, le feature normalizzate includono:

ICD9_CM Il codice della malattia, normalizzato per garantire una distribuzione uniforme.

Etichetta Codificata e normalizzata per rappresentare la classificazione clinica della malattia.

4.2.4 Costruzione del Grafo Eterogeneo

Una volta preprocessate le caratteristiche dei pazienti e delle malattie, è stato costruito un grafo eterogeneo per ciascun modello. La struttura del grafo varia leggermente tra i due modelli:

- Nel primo modello (basato su SAGEConv), il grafo è stato costruito in modo semplice e basilare, con due tipi di nodi:
  - **Pazienti**: Ogni paziente è rappresentato da un nodo con caratteristiche come sesso ed età.
  - **Malattie**: Ogni malattia è rappresentata da un nodo basato sul codice ICD9_CM e l'etichetta clinica associata. Gli archi collegano pazienti e malattie sulla base delle diagnosi presenti nel dataset. Ogni arco tra un paziente e una malattia rappresenta una relazione di diagnosi.
- Nel secondo modello (basato su GATConv), è stata aggiunta una relazione temporale tra i pazienti, che riflette l'evoluzione clinica nel tempo:
  - **Archi temporali tra pazienti**: Oltre alla relazione diagnosi tra paziente e malattia, il secondo modello introduce un secondo tipo di arco che collega i pazienti tra loro sulla base della sequenza cronologica delle diagnosi. Questa relazione temporale arricchisce il grafo con un'informazione dinamica, permettendo al modello di catturare potenzialmente la progressione delle malattie e di migliorare la predizione futura di eventi patologici.

4.2.5 Suddivisione dei Dati per l'Addestramento

Una volta costruito il grafo eterogeneo, il dataset è stato suddiviso in training, validation e test set, utilizzando una tecnica di train-test split che divide casualmente gli archi del grafo:
Training set: Il 70% degli archi è stato utilizzato per l'addestramento del modello.
Validation set: Il 15% degli archi è stato riservato per la fase di validazione, per monitorare le prestazioni del modello durante l'addestramento.
Test set: Il restante 15% degli archi è stato utilizzato per la valutazione finale del

modello, garantendo che i risultati non fossero influenzati da overfitting o da un leakage del dataset di training.
Per bilanciare il dataset, sono stati generati anche archi negativi utilizzando la tecnica del negative sampling, in cui vengono create coppie di pazienti e malattie non collegate, al fine di garantire un numero comparabile di esempi positivi e negativi durante l'addestramento.

4.3 Modello 1: HeteroGNN con SAGEConv

4.3.1 Architettura del Modello

L'approccio SAGEConv (Sample and AggregatE Convolutional Networks) permette di aggregare le informazioni dai nodi vicini durante il processo di convoluzione. Questo consente al modello di apprendere rappresentazioni latenti per ciascun nodo tenendo conto delle sue connessioni all'interno del grafo. Nel caso specifico, ogni nodo che rappresenta un paziente non viene descritto solo dalle sue caratteristiche individuali, come età, sesso e le diagnosi ricevute, ma anche dalle informazioni derivate dalle malattie con cui è collegato e, indirettamente, dalle malattie che sono state diagnosticate a pazienti vicini all'interno del grafo. In altre parole, se un paziente è connesso a una malattia, e altri pazienti nel grafo sono connessi a malattie simili o correlate, l'aggregazione di SAGEConv consente al modello di tener conto di queste connessioni indirette e di arricchire la rappresentazione del nodo paziente con informazioni provenienti dal contesto più ampio della rete locale. Questo meccanismo permette al modello di catturare le interazioni tra pazienti e malattie, costruendo rappresentazioni più complete e robuste rispetto a quelle che si baserebbero solo sulle caratteristiche individuali del singolo nodo.

Composizione del Grafo Nel modello basato su SAGEConv, il grafo è costruito in modo eterogeneo, rappresentando pazienti e malattie come nodi distinti, con le loro rispettive caratteristiche. Il grafo è formato da due tipi principali di nodi:

Nodi Paziente Ogni nodo rappresenta un singolo paziente all'interno del dataset. Le informazioni associate a ciascun paziente vengono utilizzate come feature per

descriverne le caratteristiche personali e cliniche. Le feature includono attributi come:

- **Età**: Calcolata sottraendo l'anno di nascita del paziente (presente nella colonna ANNO_NASCITA) dalla data della diagnosi (indicata nella colonna DATA_PRESCRIZIONE). Questa feature fornisce un'indicazione temporale del momento in cui il paziente è stato diagnosticato con una malattia specifica.
- **Sesso**: Codificato come un valore numerico binario (0 per maschi e 1 per femmine), normalizzato per garantire una distribuzione equilibrata e adeguata durante l'addestramento del modello.

Nodi Malattia Ogni nodo rappresenta una malattia diagnosticata, codificata secondo il sistema ICD9-CM (International Classification of Diseases, Ninth Revision, Clinical Modification). Ogni malattia viene mappata a un codice univoco presente nella colonna ICD9_CM del dataset. Le feature associate ai nodi malattia includono:

- **Codice ICD9-CM**: Il codice numerico che identifica la malattia in modo univoco.
- **Etichetta clinica**: Un'ulteriore classificazione della malattia, che offre una descrizione più dettagliata o una categorizzazione specifica. Questa feature è codificata e normalizzata per essere inclusa nelle rappresentazioni numeriche del grafo.

Gli archi nel grafo collegano i nodi paziente ai nodi malattia attraverso la relazione di diagnosi. Ogni arco rappresenta il fatto che un determinato paziente è stato diagnosticato con una specifica malattia. Gli archi possono essere intesi come collegamenti diretti che trasmettono informazioni tra pazienti e malattie. La struttura del grafo in questo modello è basilare e diretta: ogni arco rappresenta una semplice connessione tra un paziente e una malattia diagnosticata, senza ulteriori informazioni o relazioni aggiuntive tra i pazienti. Non vengono inclusi dati temporali che possano indicare, ad esempio, l'ordine cronologico delle diagnosi o l'evoluzione clinica del paziente nel tempo. Il grafo costruito in questo modello è relativamente semplice, poiché si

concentra esclusivamente sulle connessioni tra pazienti e malattie. Non vengono incorporate informazioni temporali, né vengono modellate relazioni più complesse tra pazienti. Questo approccio, sebbene efficace per una prima analisi delle relazioni di diagnosi, ha il limite di non catturare dinamiche temporali o interazioni più intricate tra pazienti, che potrebbero arricchire ulteriormente la capacità predittiva del modello. Questa mancanza viene affrontata nel secondo modello, che aggiunge una componente temporale tra pazienti.

Livelli di Convoluzione Il modello è stato strutturato su tre livelli di convoluzione eterogenea, ognuno dei quali utilizza SAGEConv per aggregare le informazioni dai nodi vicini. Ogni livello di convoluzione prende in input le rappresentazioni dei nodi dal livello precedente e applica un meccanismo di aggregazione per combinare le feature del nodo con quelle dei nodi vicini. L'aggregazione dei nodi vicini avviene secondo il meccanismo di mean aggregation, in cui le feature dei nodi vicini vengono mediate e combinate per produrre una nuova rappresentazione per ciascun nodo. In questo modo, il modello riesce a catturare le interazioni tra pazienti e malattie a più livelli di profondità. Dopo ogni livello di convoluzione, viene applicato un meccanismo di Dropout, con una probabilità di 0.2, per ridurre il rischio di overfitting. Questo processo permette di escludere casualmente alcuni nodi e archi durante l'addestramento, rendendo il modello più robusto e meno dipendente da singole connessioni.

4.3.2 Funzione di Perdita e Ottimizzazione

Funzione di Perdita La funzione di perdita scelta per il modello è la Binary Cross Entropy Loss (BCELoss), una funzione comunemente utilizzata nei compiti di classificazione binaria. In questo caso, la funzione confronta la probabilità predetta per ogni arco (ossia, la probabilità che esista una connessione tra un paziente e una malattia) con la vera etichetta (1 per gli archi effettivi, 0 per gli archi generati tramite negative sampling). L'addestramento del modello ha richiesto la generazione di archi negativi utilizzando la tecnica del negative sampling. Gli archi negativi rappresentano coppie di nodi paziente-malattia che non sono direttamente connessi, al fine di bilanciare il

dataset durante l'addestramento e garantire che il modello impari a distinguere sia le connessioni esistenti che quelle non esistenti.

Ottimizzazione Per ottimizzare il modello, è stato utilizzato l'algoritmo Adam (Adaptive Moment Estimation), uno dei metodi di ottimizzazione più popolari nel campo del deep learning grazie alla sua capacità di adattare dinamicamente i tassi di apprendimento per ciascun parametro. L'algoritmo Adam combina i vantaggi di due approcci classici: il Gradient Descent Stocastico e l'RMSProp, mantenendo una stima esponenzialmente decrescente della media dei gradienti e dei loro quadrati.

Impostazioni dell'Ottimizzatore Learning Rate: È stato scelto un learning rate iniziale pari a 0.001. Questo valore rappresenta la velocità con cui l'algoritmo Adam aggiorna i pesi del modello durante l'addestramento. Un valore troppo alto del learning rate potrebbe portare a una convergenza instabile, mentre un valore troppo basso potrebbe rallentare il processo di apprendimento. La scelta di 0.001 bilancia queste due necessità, consentendo una velocità di apprendimento moderata che permette al modello di avvicinarsi gradualmente al minimo della funzione di perdita. Weight Decay: Un parametro cruciale per regolarizzare il modello e prevenire l'overfitting è il weight decay (decadimento del peso), impostato a 1e-4. Il weight decay è una forma di regolarizzazione L2 che penalizza i pesi troppo grandi, riducendo la possibilità che il modello si adatti eccessivamente ai dati di training. Questa penalizzazione riduce la complessità del modello, favorendo soluzioni con pesi più piccoli e robusti, evitando così che il modello apprenda troppo in dettaglio le fluttuazioni casuali nei dati di addestramento.
L'algoritmo Adam si è dimostrato particolarmente efficace in questo contesto poiché adatta automaticamente i tassi di apprendimento per ciascun parametro, il che risulta utile in scenari con grafo eterogenei, dove i nodi (pazienti e malattie) hanno feature con scale di valore e dinamiche diverse, in più utilizza una combinazione di momenti di primo e secondo ordine (media e varianza dei gradienti) per aggiornare i pesi, il che rende l'ottimizzazione più stabile e veloce rispetto agli algoritmi di discesa gradiente standard.

Scheduler di Tipo ReduceLROnPlateau Per ottimizzare ulteriormente il processo di addestramento, è stato impiegato uno scheduler di tipo ReduceLROnPlateau, che riduce automaticamente il learning rate quando la funzione di perdita non mostra miglioramenti significativi per un certo numero di epoche consecutive, indicativo di uno stallo nel processo di apprendimento. Il meccanismo si basa sul monitoraggio continuo della funzione di perdita: se, per 10 epoche consecutive, non si riscontrano progressi, lo scheduler riduce il learning rate di un fattore pari a 0.5. Questa riduzione graduale consente al modello di compiere aggiustamenti più piccoli, facilitando il raggiungimento di un minimo locale più accurato ed evitando oscillazioni che ostacolano la convergenza.

Vantaggi dell'Approccio L'uso combinato di Adam e ReduceLROnPlateau ha fornito diversi vantaggi significativi durante l'addestramento:

- **Velocità di Convergenza**: Grazie alla gestione dinamica del learning rate e alla capacità di Adam di trattare gradienti rumorosi, il modello ha raggiunto la convergenza in meno epoche, riducendo il rischio di overfitting nelle fasi iniziali.
- **Prevenzione degli Stalli**: Lo scheduler ReduceLROnPlateau ha abbassato il learning rate quando la funzione di perdita non migliorava più, permettendo al modello di proseguire l'apprendimento senza interventi manuali, migliorando ulteriormente le prestazioni.
- **Maggiore Precisione**: La riduzione graduale del learning rate ha consentito al modello di fare aggiustamenti più accurati nelle fasi finali, facilitando il raggiungimento di un minimo locale ottimale senza oscillazioni dovute a gradienti elevati.
- **Efficienza Computazionale**: Questo approccio ha ridotto il numero di epoche necessarie, evitando sprechi computazionali e diminuendo il rischio di overfitting, grazie al weight decay e al fine-tuning automatico del learning rate.

In conclusione, la combinazione di Adam, weight decay e ReduceLROnPlateau ha ottimizzato le prestazioni del modello, migliorando la velocità di convergenza e la capacità di generalizzazione senza necessità di frequenti interventi manuali.

4.3.3 Addestramento e Valutazione del Modello

Train-Test Split Il dataset, come descritto in precedenza, è stato suddiviso in tre parti per garantire una valutazione accurata e robusta delle prestazioni del modello. Il 70% dei dati è stato utilizzato per il training set, che ha fornito al modello la base su cui apprendere le relazioni tra pazienti e malattie. Il 15% dei dati è stato riservato al validation set, il cui scopo era quello di monitorare le prestazioni del modello durante l'addestramento e aiutare a prevenire fenomeni di overfitting. Infine, il restante 15% è stato utilizzato per il test set, con cui valutare definitivamente la capacità del modello di generalizzare a dati non visti in precedenza. Durante la fase di addestramento, il modello ha imparato a distinguere tra gli archi veri e quelli falsi. Gli archi veri rappresentavano connessioni reali tra un paziente e una malattia, basate sulle informazioni diagnostiche contenute nel dataset. Al contrario, gli archi falsi (o negativi) sono stati creati artificialmente tramite una tecnica chiamata negative sampling, che genera coppie di nodi paziente-malattia non connessi tra loro nel grafo. L'obiettivo del modello era imparare a riconoscere quali connessioni esistevano effettivamente e quali erano invece inesistenti. Il validation set è stato di fondamentale importanza per monitorare il comportamento del modello durante l'addestramento. Grazie all'uso di questo insieme di dati, è stato possibile verificare se il modello stava imparando a generalizzare correttamente o se stava cominciando ad adattarsi eccessivamente ai dati di training, un fenomeno noto come overfitting. Attraverso questa valutazione continua sulle prestazioni nel validation set, è stato possibile aggiustare i parametri di addestramento e intervenire tempestivamente nel caso in cui le prestazioni iniziassero a deteriorarsi.

Ciclo di Addestramento Il processo di addestramento del modello ha seguito un ciclo ben definito, articolato in diverse fasi che hanno permesso al modello di apprendere dalle informazioni presenti nel grafo. La prima fase è stata il forward

pass, durante la quale le feature di ciascun nodo, sia dei pazienti che delle malattie, sono state fatte passare attraverso i vari livelli di convoluzione del modello. Ogni livello di convoluzione ha permesso di aggregare le informazioni dai nodi vicini, creando così una rappresentazione latente per ciascun nodo. Queste rappresentazioni contengono informazioni più ricche e complesse rispetto alle semplici feature di partenza, poiché tengono conto delle connessioni tra i nodi e delle caratteristiche dei loro vicini nel grafo. Dopo aver ottenuto queste rappresentazioni latenti, il modello è passato alla fase di predizione dei link. In questa fase, ha utilizzato le rappresentazioni dei nodi per predire la probabilità di esistenza di un arco tra un determinato paziente e una specifica malattia. In altre parole, il modello ha cercato di capire se un paziente, basandosi sulle sue caratteristiche e sulle sue connessioni con altre malattie e pazienti, fosse effettivamente collegato a una determinata malattia. Una volta effettuate le predizioni, è stata calcolata la funzione di perdita utilizzando la BCELoss (Binary Cross Entropy Loss). Questa funzione ha confrontato le probabilità predette dal modello con le etichette reali degli archi, cioè ha verificato quanto il modello fosse vicino o lontano dal predire correttamente la presenza o l'assenza di un arco tra paziente e malattia. Infine, nella fase di backpropagation, gli errori (ossia la differenza tra le predizioni e i valori reali) sono stati propagati all'indietro attraverso la rete, aggiornando i pesi del modello. Questo processo ha permesso al modello di apprendere dagli errori commessi e di migliorare le predizioni successive. Ogni volta che i pesi venivano aggiornati, il modello diventava progressivamente più accurato nel predire la presenza di connessioni tra i nodi. Un elemento chiave durante l'addestramento è stato l'uso del Dropout, applicato tra i vari livelli di convoluzione. Questo meccanismo ha contribuito a migliorare la generalizzazione del modello, evitando che si adattasse eccessivamente a specifiche connessioni del grafo. Escludendo casualmente alcune connessioni durante l'addestramento, il Dropout ha reso il modello più robusto e meno incline a overfitting, migliorandone così la capacità di generalizzare su dati non visti.

4.3.4 Risultati del Modello 1

Metriche di Valutazione Le prestazioni del modello sono state valutate utilizzando una serie di metriche comuni per i compiti di classificazione binaria, tra cui:
ROC AUC: L'area sotto la curva ROC, che misura la capacità del modello di distinguere tra archi veri e archi negativi. **Accuracy**: La percentuale di predizioni corrette. **Precision e Recall**: Metriche che valutano la precisione del modello nel predire connessioni esistenti e la sua capacità di identificare correttamente tutti i collegamenti reali. **F1-Score**: Una combinazione di precision e recall per fornire una misura più bilanciata delle prestazioni complessive del modello.

Risultati Il modello basato su SAGEConv ha mostrato buone prestazioni complessive in termini di accuratezza predittiva e gestione del processo di addestramento, specialmente considerando la complessità e la dimensione del dataset utilizzato, che comprendeva 1 milione di record. Le valutazioni chiave del modello sono state monitorate principalmente attraverso la loss e l'accuracy durante l'intero ciclo di addestramento, mentre le altre metriche come AUC-ROC, precision, recall e F1-score sono state calcolate in modo dettagliato alla fine del ciclo di addestramento, utilizzando il set di test.

Evoluzione della Loss e dell'Accuracy durante l'Addestramento Nel corso delle 100 epoche, il modello ha mostrato una progressiva riduzione della loss, che misura la differenza tra le predizioni del modello e i valori reali degli archi presenti nel grafo. Alla 10ª epoca, il valore della loss era pari a 0.6923, mentre l'accuracy (percentuale di predizioni corrette) era ancora bassa ma migliorava costantemente. Il modello stava ancora imparando a distinguere tra gli archi veri (paziente $\rightarrow$ malattia) e gli archi negativi creati tramite negative sampling.
Con il progredire delle epoche, sia la loss che l'accuracy hanno continuato a migliorare. Alla 20ª epoca, la loss è scesa a 0.6823 e l'accuracy ha iniziato a mostrare miglioramenti significativi. Questa tendenza è continuata fino alla 50ª epoca, dove la loss è scesa ulteriormente a 0.6101, segnalando che il modello stava apprendendo le connessioni nel grafo in modo più efficiente. Alla 80ª epoca, la loss ha raggiunto 0.5897, e l'accuracy ha continuato ad aumentare.

Alla fine dell'addestramento, alla 100ª epoca, la loss si è stabilizzata attorno a 0.5812, con un'accuracy complessivamente elevata, segnalando che il modello aveva imparato a riconoscere correttamente le relazioni tra pazienti e malattie con una buona affidabilità. Le metriche di precision e accuracy sono state calcolate durante questa fase sia sul training set che sul validation set. Ciò ha permesso di monitorare le prestazioni del modello in tempo reale durante l'addestramento, verificando che il modello non stesse sovradattandosi ai dati del training set e fosse capace di generalizzare anche sul validation set.

Valutazione Finale sulle Metriche di AUC-ROC, Precision, Recall e F1-Score sul Set di Test Alla fine del ciclo di addestramento, le metriche chiave di valutazione del modello, tra cui AUC-ROC, precision, recall e F1-score, sono state calcolate utilizzando il test set per garantire una valutazione finale accurata e indipendente dai dati visti durante l'addestramento. Alla 100ª epoca, il valore di AUC-ROC è stato pari a 0.8332 sul test set, indicando una buona capacità del modello di distinguere tra archi veri e falsi. Questo valore è particolarmente significativo, poiché un AUC superiore a 0.80 riflette una solida capacità di discriminare le connessioni corrette nel grafo eterogeneo. La precision finale, calcolata sul test set, ha mostrato che il modello era in grado di predire connessioni reali tra pazienti e malattie con un'alta percentuale di correttezza. Questo significa che la maggior parte degli archi predetti come veri corrispondevano effettivamente a connessioni esistenti. Il recall ha evidenziato la capacità del modello di identificare correttamente la maggior parte delle connessioni esistenti, limitando i falsi negativi. L'F1-score, una metrica che combina precision e recall in un'unica valutazione, ha confermato le buone prestazioni generali del modello. Un F1-score bilanciato come quello ottenuto indica che il modello non solo ha predetto correttamente le connessioni, ma è riuscito a mantenere un equilibrio tra la capacità di riconoscere tutte le connessioni esistenti e il numero di falsi positivi.

Conclusioni sui Risultati In sintesi, il modello SAGEConv ha appreso efficacemente le relazioni complesse tra pazienti e malattie, mostrando miglioramenti costanti nelle metriche di loss e accuracy durante l'addestramento. Le valutazioni finali sul test set hanno confermato la sua efficacia, con un AUC-ROC di 0.8332 e un buon bilancia-

mento tra precision e recall. L'uso del Dropout e dello scheduler ReduceLROnPlateau ha evitato il sovradattamento, migliorando la generalizzazione del modello su dati non visti. Questi risultati dimostrano la solidità del modello per futuri sviluppi nella predizione delle comorbilità e nell'analisi di grafi sanitari eterogenei.

4.4 Modello 2: HeteroGNN con GATConv

4.4.1 Architettura del Modello

L'approccio GATConv (Graph Attention Networks) introduce un meccanismo di attenzione che permette al modello di assegnare importanza variabile alle connessioni tra nodi durante la convoluzione, migliorando così la qualità delle rappresentazioni latenti. Ogni nodo paziente è descritto non solo dalle proprie caratteristiche, come età e diagnosi, ma anche dalle malattie a cui è collegato. Grazie all'attenzione, il modello può attribuire maggiore peso alle connessioni paziente-malattia più rilevanti clinicamente. Il modello cattura anche le connessioni temporali tra pazienti, tracciando l'evoluzione delle condizioni cliniche nel tempo, utile per monitorare lo sviluppo di malattie croniche. GATConv consente di pesare diversamente le relazioni paziente-malattia e paziente-paziente, selezionando quelle più significative nel contesto clinico. In sintesi, l'uso di GATConv permette al modello di concentrare l'attenzione sulle connessioni più rilevanti, fornendo una rappresentazione del grafo più accurata e clinicamente significativa rispetto a modelli che non considerano tali pesi adattivi.

Composizione del Grafo Il grafo costruito per il modello HeteroGNN con GATConv è decisamente più complesso rispetto a quello del primo modello basato su SAGEConv, poiché introduce una serie di elementi che aggiungono profondità e una maggiore interpretazione clinica dei dati. In questo grafo, sono presenti due tipologie di nodi principali:

Nodi Paziente Ogni nodo rappresenta un paziente del dataset, caratterizzato da attributi come l'età e il sesso, codificati come feature numeriche. Ogni nodo paziente può avere collegamenti sia con le malattie diagnosticate che con altri

nodi paziente, creando una rete temporale che tiene traccia della storia clinica di ogni individuo.

Nodi Malattia I nodi malattia rappresentano le diagnosi ricevute dai pazienti, codificate tramite il codice ICD9-CM e arricchite da un'etichetta clinica che descrive in modo più dettagliato la malattia.

Ci sono due tipi di archi che collegano i nodi:

Paziente → Malattia (diagnosi) Questi archi rappresentano il fatto che un determinato paziente sia stato diagnosticato con una specifica malattia. A differenza del primo modello, questi collegamenti includono un peso che indica la frequenza o la ricorrenza di quella malattia nel paziente, il che rende il grafo più informativo. Se un paziente è stato diagnosticato con la stessa malattia più volte, il peso dell'arco aumenta. Questo permette al modello di catturare una maggiore rilevanza clinica di alcune malattie rispetto ad altre.

Paziente → Paziente (temporale) Questo tipo di arco connette lo stesso paziente a se stesso, ma in momenti temporali differenti. La presenza di questa connessione temporale è ciò che rende il grafo più dinamico e clinicamente significativo rispetto al modello precedente. Le date delle diagnosi sono state codificate come un valore temporale che viene utilizzato per collegare i nodi paziente in base alla sequenza temporale delle diagnosi. Questo permette di rappresentare la progressione delle malattie nel tempo e di studiare l'evoluzione delle condizioni cliniche di un paziente, elemento cruciale nel contesto medico. Questa dimensione temporale non era presente nel modello basato su SAGEConv e costituisce un'importante aggiunta, poiché consente di modellare la cronologia degli eventi clinici. La possibilità di collegare i nodi paziente anche attraverso le connessioni temporali arricchisce il modello, permettendo di analizzare non solo lo stato attuale del paziente, ma anche come la sua salute si sia evoluta nel tempo.

In sintesi, il grafo costruito per il modello GATConv è più complesso e sofisticato rispetto al modello precedente. Non solo tiene conto delle caratteristiche dei nodi paziente e malattia, ma introduce anche pesi sugli archi per riflettere la ricorrenza delle

malattie e connessioni temporali tra i nodi paziente, fornendo così una visione più completa e clinicamente rilevante delle storie mediche dei pazienti. Questo permette al modello di estrarre informazioni più dettagliate e di catturare meglio l'evoluzione delle condizioni di salute nel tempo, migliorando la qualità delle predizioni rispetto a un grafo statico.

Livelli di Convoluzione Il modello HeteroGNN con GATConv è composto da tre livelli di convoluzione eterogenea, ognuno dei quali utilizza il meccanismo di attenzione per gestire le relazioni tra Paziente $\rightarrow$ Malattia (diagnosi) e Paziente $\rightarrow$ Paziente (temporale). Ogni livello impiega l'aggregazione media per combinare le feature dei nodi vicini e utilizza tecniche di regolarizzazione come dropout e weight decay per migliorare la generalizzazione e prevenire l'overfitting.
Primo livello di convoluzione: In questa fase, il modello applica l'attenzione per convolvere le feature dei nodi considerando entrambe le relazioni (diagnosi e temporale). L'aggregazione media combina le informazioni dai nodi adiacenti, consentendo a ogni paziente di integrare le diagnosi e l'evoluzione temporale. Il dropout regolarizza il processo, riducendo la dipendenza da connessioni specifiche.
Secondo livello di convoluzione: Questo livello raffina ulteriormente le rappresentazioni latenti dei nodi, continuando a combinare le due relazioni. Le rappresentazioni diventano più ricche di informazioni cliniche e temporali, fornendo una visione più completa della progressione del paziente. Il dropout e il weight decay continuano a prevenire il sovradattamento.
Terzo livello di convoluzione: L'ultimo livello perfeziona le rappresentazioni latenti, convolvendo simultaneamente le relazioni diagnostiche e temporali. Grazie all'attenzione, le connessioni clinicamente rilevanti vengono enfatizzate, permettendo al modello di costruire rappresentazioni finali robuste e significative.
In sintesi, il modello utilizza l'aggregazione media, il dropout e il weight decay in ogni livello per bilanciare le due relazioni, enfatizzando le connessioni più rilevanti grazie al meccanismo di attenzione. Questo permette al modello di catturare in modo efficace sia le informazioni cliniche sia quelle temporali tra pazienti e malattie.

Decoder Dopo le tre convoluzioni, il modello utilizza un decoder per predire la probabilità che un paziente sia associato a una malattia. Il dot product viene calcolato tra le rappresentazioni latenti del paziente e della malattia, e il risultato passa attraverso una funzione sigmoide per ottenere una probabilità compresa tra 0 e 1. Questo rappresenta la probabilità che un paziente abbia effettivamente contratto una malattia specifica.

4.4.2 Funzione di Perdita e Ottimizzazione

Nel modello HeteroGNN con GATConv, l'ottimizzazione è stata effettuata utilizzando una funzione di perdita basata su BCEWithLogitsLoss, combinata con un optimizer AdamW per l'aggiornamento dei pesi. Questo approccio è stato scelto per gestire la natura binaria del task di predizione dei link (paziente $\rightarrow$ malattia), in cui il modello deve distinguere tra archi veri e archi negativi (generati artificialmente).

Funzione di Perdita La funzione di perdita utilizzata è BCEWithLogitsLoss (Binary Cross-Entropy with Logits Loss), che è una scelta comune per problemi di classificazione binaria. Questa funzione calcola la cross-entropia tra le predizioni del modello (valori logit) e le etichette reali degli archi, garantendo che il modello apprenda a distinguere correttamente tra archi positivi (paziente diagnosticato con una malattia) e archi negativi (paziente non associato a una malattia). Inoltre, per gestire lo squilibrio tra archi positivi e negativi, è stato utilizzato un peso di classe. Il modello ha bilanciato l'influenza delle classi positive e negative attraverso il parametro pos_weight, calcolato in modo da dare più peso agli archi positivi rispetto a quelli negativi, data la maggiore frequenza di archi negativi generati durante il training.

Ottimizzazione con AdamW Per aggiornare i pesi del modello è stato utilizzato l'optimizer AdamW, una variante dell'algoritmo Adam che incorpora esplicitamente il weight decay per migliorare la regolarizzazione. A differenza di Adam, che combina la stima del gradiente con il momento, AdamW separa il weight decay dalla correzione del gradiente, applicandolo direttamente ai pesi durante l'aggiornamento. Questo approccio previene l'accumulo eccessivo dei pesi, migliorando la regolarizzazione. Nel nostro caso, il learning rate iniziale è stato impostato a 0.001 con un

weight decay di 1e-4. Questa configurazione ha permesso di bilanciare la velocità di apprendimento con la stabilità dei pesi, garantendo una buona convergenza del modello e prevenendo la crescita eccessiva dei parametri, che può avvenire in assenza di regolarizzazione adeguata.

ReduceLROnPlateau Per migliorare ulteriormente la convergenza, è stato utilizzato lo stesso scheduler ReduceLROnPlateau adottato nel modello precedente. Questo scheduler riduce automaticamente il learning rate quando la perdita non mostra miglioramenti per un certo numero di epoche consecutive, prevenendo eventuali stalli e permettendo al modello di esplorare in modo più preciso lo spazio dei parametri. In combinazione con AdamW e BCEWithLogitsLoss, lo scheduler ha garantito un equilibrio tra efficienza e accuratezza, migliorando le prestazioni predittive e riducendo il rischio di overfitting.

4.4.3 Addestramento e Valutazione del Modello

Il processo di addestramento e valutazione del modello HeteroGNN con GATConv è stato progettato per garantire un'accurata predizione dei collegamenti tra nodi paziente e malattia, tenendo conto anche dell'evoluzione temporale delle malattie. Il dataset è stato suddiviso in training set (70%), validation set (15%) e test set (15%), per garantire una valutazione robusta delle prestazioni del modello.

Fasi del ciclo di addestramento Il ciclo di addestramento del modello si articola in diverse fasi principali, ciascuna essenziale per migliorare progressivamente le predizioni:
Forward pass: Le feature dei nodi paziente e malattia vengono elaborate attraverso i tre livelli di convoluzione eterogenea, come descritto in precedenza. Il meccanismo di attenzione di GATConv assegna maggiore peso alle connessioni più rilevanti, mentre l'aggregazione media combina le informazioni dei nodi vicini, generando rappresentazioni latenti che catturano sia le informazioni cliniche che quelle temporali.
Predizione dei link: Utilizzando queste rappresentazioni latenti, il modello predice la probabilità di esistenza di un arco tra nodi paziente $\rightarrow$ malattia, per gli archi positivi (esistenti) e negativi (generati artificialmente) per bilanciare l'addestramento.

Calcolo della perdita: La funzione BCEWithLogitsLoss viene utilizzata per calcolare la perdita, confrontando le predizioni con le etichette reali. L'ottimizzazione avviene tramite AdamW, che combina la gestione dinamica del learning rate con la regolarizzazione tramite weight decay.
Backpropagation e aggiornamento: Gli errori vengono propagati indietro per aggiornare i pesi del modello, con il dropout che riduce il rischio di overfitting. Lo scheduler ReduceLROnPlateau monitora la perdita e riduce il learning rate quando non si riscontrano miglioramenti, come già descritto in precedenza.
In questo modo, il modello è in grado di migliorare continuamente le proprie predizioni, bilanciando l'ottimizzazione e la regolarizzazione in ogni fase.

Addestramento e Valutazione su Set di Training e Validation Durante l'addestramento del modello HeteroGNN con GATConv, il sistema ha progressivamente imparato a distinguere tra gli archi veri (che rappresentano connessioni reali tra paziente e malattia) e quelli negativi (generati artificialmente). Questa fase è stata attentamente monitorata per valutare sia la capacità del modello di apprendere dai dati di training, sia la sua capacità di generalizzazione tramite il validation set. Nelle prime fasi, il modello ha mostrato una certa variabilità nelle prestazioni, con un'accuratezza oscillante dovuta alla complessità del compito e alla natura intricata dei dati clinici. Tuttavia, con il progredire delle epoche, il modello ha iniziato a convergere verso prestazioni più stabili.

- **Epoca 1**: Il training è iniziato con una train loss di 0.4470 e una train accuracy del 93.85%, mentre la val accuracy era significativamente più bassa, al 66.70%. Questa differenza era attesa nelle fasi iniziali, poiché il modello stava ancora imparando a classificare correttamente gli archi.

- **Epoche 2-4**: Durante queste epoche, l'accuratezza del modello sul training set è rimasta stabile, con una media del 66-85%, e la val accuracy ha mostrato lievi miglioramenti, fino al 66.83%. Nonostante ciò, la train loss ha iniziato a ridursi, segno che il modello stava lentamente convergendo.

- **Epoca 5**: Il primo miglioramento significativo si è registrato con una val accuracy di 94.94%, accompagnata da una train loss di 0.4469. Ciò ha indicato che il modello stava migliorando la sua capacità di predizione, riducendo gli errori.
- **Epoche 6-10**: Durante queste epoche, l'accuracy sul training set si è mantenuta elevata (91-93%) e la val accuracy ha mostrato una stabilità intorno al 66-95%, con oscillazioni occasionali. La train loss è rimasta su livelli bassi (circa 0.4467), riflettendo una progressiva riduzione degli errori.
- **Epoche 11-20**: Verso la fine dell'addestramento, il modello ha continuato a mantenere una val accuracy elevata, raggiungendo nuovamente il 94.94% in più epoche. La train loss si è stabilizzata intorno a 0.4465, mentre la train accuracy ha continuato a migliorare fino a raggiungere il 94.63% nell'ultima epoca.

Questi risultati confermano che il modello ha acquisito una maggiore stabilità e capacità di predizione, distinguendo efficacemente tra archi positivi e negativi nel contesto clinico, con una buona generalizzazione sui dati di validazione.

Risultati Finali Alla fine del ciclo di addestramento di 20 epoche, il modello ha mostrato una val accuracy massima del 94.94% e una train accuracy di 94.63%. Questi risultati indicano che il modello ha imparato efficacemente a distinguere gli archi veri da quelli negativi, migliorando sensibilmente le sue prestazioni rispetto alle prime fasi dell'addestramento.

Valutazione sul Test Set Al termine delle 20 epoche di addestramento, il modello è stato valutato sul test set utilizzando diverse metriche per misurare le sue prestazioni:
AUC-ROC (Area Under the Curve): Il modello ha raggiunto un'AUC di 0.9247, dimostrando un'eccellente capacità di distinguere tra archi positivi (reali) e negativi (falsi). Questo valore indica una solida capacità discriminativa nel contesto clinico.
Accuracy: Sul test set, il modello ha ottenuto un'accuracy del 94.97%, mostrando una buona capacità di predizione complessiva.
Precision e Recall: Il modello ha raggiunto una precision del 92.99%, segnalando che la maggior parte delle connessioni previste come vere erano effettivamente corrette.

Anche il recall, che misura la capacità di identificare correttamente le connessioni reali, è stato bilanciato, garantendo un basso numero di falsi negativi.
F1-Score: La F1-Score, che combina precision e recall, è stata del 96.36%, evidenziando un equilibrio tra la capacità di predire correttamente le connessioni esistenti e l'evitare falsi positivi.
In generale, la curva ROC ha confermato la capacità del modello di bilanciare bene il compromesso tra falsi positivi e falsi negativi. Le ottime prestazioni del modello sul test set dimostrano la sua efficacia nel predire correttamente i collegamenti nel grafo, sfruttando sia le informazioni cliniche che quelle temporali.

Valutazione Finale sulle Metriche Il modello HeteroGNN con GATConv è stato valutato utilizzando le metriche chiave della classificazione binaria: AUC-ROC, precision, recall e F1-score, ottenendo risultati elevati che indicano un buon apprendimento e generalizzazione. Il risultato di AUC-ROC di 0.9247 riflette la capacità del modello di distinguere correttamente tra connessioni vere e false, superando il confine comune di 0.80 per una buona capacità predittiva. Tuttavia, è importante considerare che la relativa semplicità del dataset potrebbe aver contribuito a questo risultato elevato.
Precision ha dimostrato la capacità del modello di identificare correttamente le connessioni reali tra pazienti e malattie, con una buona accuratezza nelle predizioni. Anche recall ha evidenziato la capacità del modello di ridurre i falsi negativi, indicando che la maggior parte delle connessioni rilevanti sono state riconosciute correttamente.
L'F1-score ha confermato le buone prestazioni generali del modello, mostrando un equilibrio efficace tra precision e recall. Tuttavia, è importante notare che, con un dataset più complesso o caratterizzato da variabili cliniche aggiuntive, i risultati potrebbero variare, e le prestazioni potrebbero non essere così elevate in contesti più sfidanti.
In sintesi, il modello ha dimostrato un'elevata capacità di predizione, anche se la complessità del dataset può influire sulle sue prestazioni in contesti più dettagliati e ricchi di informazioni cliniche.

Conclusioni sui Risultati In sintesi, il modello HeteroGNN con GATConv ha dimostrato di saper apprendere in modo estremamente efficace le relazioni tra pazienti e malattie presenti nel grafo. Le metriche di loss e accuracy hanno mostrato un miglioramento continuo e stabile durante tutto il processo di addestramento, segnalando che il modello ha appreso le connessioni in modo progressivo. Tuttavia, l'elevata performance ottenuta, con valori di AUC-ROC e altre metriche molto vicini all'ottimo, solleva il dubbio che il problema possa essere meno caratterizzato del necessario. In altre parole, il dataset utilizzato potrebbe non contenere informazioni sufficientemente ricche da sfidare appieno il potenziale del modello. Se i dati clinici fossero più caratterizzati, includendo ad esempio variabili aggiuntive come informazioni sulla gravità delle malattie, trattamenti prescritti, dati longitudinali più dettagliati o fattori di rischio specifici, le prestazioni del modello potrebbero cambiare. In questo contesto più complesso, il modello potrebbe incontrare sfide maggiori, e le metriche di performance potrebbero fornire un quadro più realistico delle capacità del modello di gestire un problema realmente difficile. Nonostante ciò, l'attuale implementazione del modello ha dimostrato la sua forza nel riconoscere schemi e connessioni all'interno di un dataset clinico eterogeneo, offrendo ottime potenzialità per applicazioni future nella predizione delle comorbilità e nell'analisi di grafi sanitari. Questo evidenzia che, sebbene il modello sia molto efficace, la qualità e la complessità dei dati utilizzati sono cruciali per sfruttarne al massimo le capacità e ottenere risultati che riflettano più accuratamente la complessità reale del problema clinico.

4.5 Confronto tra i due Modelli

In questa sezione viene presentato un confronto tra i due modelli di Graph Neural Network (GNN) sviluppati: il modello basato su SAGEConv e il modello basato su GATConv. Entrambi i modelli sono stati progettati per affrontare il problema della predizione delle connessioni tra pazienti e malattie utilizzando un grafo eterogeneo, ma differiscono nelle loro architetture, complessità e performance. Il confronto si concentrerà su vari aspetti chiave, tra cui l'architettura del modello, l'uso delle

relazioni nel grafo, la capacità di apprendimento, le prestazioni in termini di metriche di valutazione e la robustezza ai dati meno caratterizzati.

4.5.1 Architettura dei Modelli

Il modello SAGEConv si basa sull'aggregazione delle informazioni dai nodi vicini, utilizzando il meccanismo di convoluzione per apprendere rappresentazioni latenti per ciascun nodo. Questo approccio si rivela efficace per cogliere le connessioni locali all'interno del grafo, poiché ogni nodo aggrega le informazioni dai suoi vicini diretti, integrando le feature individuali con le informazioni provenienti dalle malattie associate.
Il modello GATConv, d'altra parte, introduce un meccanismo di attenzione (attention mechanism), che permette al modello di assegnare diversi pesi alle connessioni tra i nodi. In altre parole, non tutte le connessioni vengono trattate in modo uguale: il modello può dare maggiore rilevanza a certe connessioni paziente-malattia rispetto ad altre, basandosi sulla loro importanza clinica. Questo livello aggiuntivo di complessità permette al modello di catturare meglio le sfumature delle interazioni tra pazienti e malattie, in particolare in contesti dove alcune connessioni potrebbero essere più significative di altre. Inoltre, il modello GATConv utilizza relazioni temporali tra i nodi paziente, aggiungendo un contesto cronologico che arricchisce ulteriormente la rappresentazione del grafo, fornendo una maggiore valenza clinica ai dati.

4.5.2 Complessità del Grafo e delle Relazioni

Il grafo utilizzato dal modello SAGEConv è relativamente semplice, con due tipi principali di nodi (pazienti e malattie) e una sola relazione tra di essi: la diagnosi. Ogni arco rappresenta una connessione diretta tra un paziente e una malattia senza ulteriori informazioni aggiuntive come il tempo o il peso delle relazioni. Questo rende il grafo più basilare, ma anche meno espressivo dal punto di vista clinico, in quanto non cattura la dinamica temporale o la progressione delle malattie.
Al contrario, il grafo utilizzato dal modello GATConv è più complesso e ricco di informazioni. Oltre alle connessioni paziente-malattia, vengono introdotti anche archi temporali tra i nodi paziente, che permettono di seguire la progressione delle

malattie e delle diagnosi nel tempo. Inoltre, gli archi paziente-malattia sono dotati di pesi, che rappresentano la ricorrenza di una malattia per uno stesso paziente, un'informazione che aggiunge un livello di dettaglio significativo. Questa struttura più avanzata rende il modello GATConv più adatto a rappresentare scenari clinici complessi, dove il tempo e la frequenza delle diagnosi giocano un ruolo cruciale.

4.5.3 Prestazioni e Robustezza

In termini di prestazioni, entrambi i modelli hanno dimostrato di saper apprendere efficacemente le relazioni tra pazienti e malattie. Tuttavia, il modello GATConv ha ottenuto risultati migliori nelle metriche di valutazione. In particolare, ha raggiunto un AUC-ROC di 0.9247, che indica un'ottima capacità di distinguere tra connessioni vere e false. Anche le altre metriche, come precision, recall e F1-score, sono risultate molto elevate, con una precision di 0.9299 e un F1-score di 0.9636. Questi risultati suggeriscono che il meccanismo di attenzione e l'integrazione delle relazioni temporali hanno permesso al modello di cogliere più dettagli e sfumature nelle connessioni del grafo, migliorando la sua capacità predittiva.
Il modello SAGEConv, pur avendo prestazioni solide, ha ottenuto risultati leggermente inferiori. Il suo AUC-ROC finale è stato di 0.8332, comunque indicativo di una buona capacità predittiva, ma inferiore rispetto al modello GATConv. Questo risultato è probabilmente dovuto alla struttura più semplice del grafo, che non integra né pesi né informazioni temporali, limitando in parte la capacità del modello di cogliere interazioni più complesse tra pazienti e malattie.
Tuttavia, è importante considerare che i risultati ottenuti dal modello GATConv sono influenzati anche dal fatto che i dati utilizzati non erano estremamente caratterizzati. Sebbene il modello abbia mostrato prestazioni eccezionali, questo potrebbe riflettere un problema relativamente semplice, dove le connessioni sono più facili da identificare. Se i dati fossero stati più complessi, con informazioni cliniche più dettagliate e variabili aggiuntive, i risultati potrebbero essere diversi, e il modello potrebbe affrontare sfide più significative.

4.5.4 Robustezza ai Dati Poco Caratterizzati

Un aspetto importante da considerare è la robustezza dei modelli di fronte a dati meno caratterizzati. Entrambi i modelli hanno ottenuto ottimi risultati nonostante il dataset utilizzato non fosse particolarmente ricco di dettagli clinici. Tuttavia, è ragionevole ipotizzare che il modello GATConv sarebbe stato in grado di gestire meglio un dataset più complesso grazie al suo meccanismo di attenzione e all'integrazione delle informazioni temporali. La capacità di questo modello di dare peso alle connessioni più rilevanti e di tenere conto della dinamica temporale lo rende potenzialmente più robusto in scenari più complessi.
Il modello SAGEConv, pur essendo meno sofisticato, ha dimostrato una buona capacità di generalizzare anche con dati poco caratterizzati. Tuttavia, la mancanza di una componente temporale e di un meccanismo di attenzione potrebbe limitare la sua capacità di apprendere relazioni più complesse se il dataset fosse stato più dettagliato.

4.5.5 Considerazioni Finali

In conclusione, entrambi i modelli hanno dimostrato di essere efficaci nella predizione delle connessioni paziente-malattia, con il modello GATConv che si è rivelato superiore grazie alla sua maggiore complessità architettonica. L'inclusione delle relazioni temporali e del meccanismo di attenzione ha permesso a GATConv di ottenere prestazioni migliori rispetto al più semplice modello SAGEConv, che ha comunque raggiunto risultati soddisfacenti. È importante sottolineare che l'elevata performance del modello GATConv è influenzata dalla natura relativamente semplice dei dati utilizzati. Future implementazioni su dataset più complessi e ricchi di informazioni cliniche permetteranno di valutare meglio le potenzialità di questi modelli nel contesto clinico reale, offrendo una visione più completa delle loro capacità predittive.

CAPITOLO 5

Conclusioni e Sviluppi Futuri

5.1 Conclusioni

Nel corso di questo lavoro di tesi, abbiamo sviluppato **ComorGraph**, una piattaforma avanzata per l'analisi della comorbidità, basata su tecniche di **Social Network Analysis (SNA)** e progettata per migliorare la gestione di dati clinici complessi. **ComorGraph** si inserisce all'interno dell'ecosistema MedMiner e consente di visualizzare le relazioni tra malattie come un grafo interattivo, applicando metriche grafiche per individuare patologie centrali nel quadro clinico di un paziente.
L'obiettivo principale della piattaforma è quello di offrire uno strumento pratico per i medici, che possono così analizzare la comorbidità in modo visivo e dinamico, identificando pattern rilevanti e supportando il processo decisionale clinico. Il dott. Pierpaolo Cavallo ha approvato e apprezzato **ComorGraph**, sottolineando il suo potenziale utilizzo da parte dei professionisti sanitari per migliorare la diagnosi e il trattamento delle comorbidità.
Parallelamente alla piattaforma, lo studio ha esplorato l'applicazione di modelli di **Graph Neural Networks (GNN)** per la predizione di nuove patologie nei pazienti. Sebbene questo modulo di intelligenza artificiale non faccia parte integrante di **ComorGraph**, ha mostrato risultati promettenti nel prevedere l'insorgenza di nuove

malattie basandosi su pattern di comorbidità già presenti. In particolare, i modelli **HeteroGNN con SAGEConv** e **GATConv** hanno offerto prestazioni interessanti, dimostrando la potenziale utilità di questa tecnologia per supportare il lavoro clinico in futuro.

5.2 Sviluppi Futuri

Nonostante **ComorGraph** rappresenti già un avanzamento significativo nello studio delle comorbidità grazie alla sua capacità di analizzare reti complesse di malattie, ci sono alcuni sviluppi futuri che potrebbero migliorare ulteriormente la piattaforma, offrendo ai professionisti sanitari strumenti ancora più potenti e specifici.
Uno dei principali obiettivi futuri è l'integrazione del modulo di **intelligenza artificiale** direttamente all'interno della piattaforma, permettendo così un'analisi predittiva più precisa ed efficiente. In questo modo, **ComorGraph** potrà evolversi in uno strumento completo che non solo consente di visualizzare e analizzare le reti di comorbidità, ma anche di anticipare potenziali sviluppi patologici per ogni singolo paziente.
Un secondo aspetto fondamentale che sarà esplorato è l'implementazione di un modello basato su **Temporal Graph Neural Networks (TemporalGNN)**. L'introduzione della dimensione temporale permetterà di studiare i **pattern temporali**, le periodicità e la frequenza con cui determinate malattie si manifestano nei pazienti. Questo potrebbe offrire nuove prospettive nel comprendere la progressione delle patologie e nel predire l'evoluzione della salute di un paziente con un livello di dettaglio molto più alto.
Il terzo potenziale sviluppo è la creazione di un **null model** della rete di comorbidità. Un null model è una rete casuale che conserva alcune proprietà strutturali della rete reale, ma rimuove altre proprietà, consentendo di eseguire **analisi comparative** per identificare quali caratteristiche emergono naturalmente dalla struttura del grafo e quali sono significative rispetto alla casualità. Questo strumento potrebbe fornire ulteriori insight sulle dinamiche delle malattie nelle reti di comorbidità.
Infine, si prevede di includere funzionalità per **evidenziare relazioni ricorrenti**

tra malattie. Queste informazioni potranno essere utilizzate per avvisare gli utenti quando malattie che tendono a verificarsi insieme appaiono nel grafico del paziente, migliorando la gestione della diagnosi e prevenendo potenziali complicazioni future. Questi sviluppi rappresentano opportunità per migliorare la piattaforma, pur mantenendo la sua solidità attuale, e per offrire funzionalità avanzate che potrebbero portare a nuove scoperte e ottimizzazioni nel campo della medicina predittiva e della gestione clinica delle comorbidità.

Bibliografia

[1] J. M. Valderas, B. Starfield, B. Sibbald, C. Salisbury, and M. Roland, "Defining comorbidity: Implications for understanding health and health services," *The Annals of Family Medicine*, vol. 7, no. 4, pp. 357–363, 2009. [Online]. Available: https://www.annfammed.org/content/7/4/357 (Citato alle pagine 5 e 6)

[2] R. G. V. M. Prosperina, Ed., *Challenges in Social Network Research: Methods and Applications*. Springer, 2020. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-31463-7 (Citato a pagina 5)

[3] R. Brown and E. Thorsteinsson, *Comorbidity: What Is It and Why Is It Important?* Cham: Springer International Publishing, 2020, pp. 1–22. [Online]. Available: https://doi.org/10.1007/978-3-030-32545-9_1 (Citato a pagina 6)

[4] D. Chambers, P. Wilson, C. Thompson, and M. Harden, "Social network analysis in healthcare settings: A systematic scoping review," *PLOS ONE*, vol. 7, no. 8, p. e41911, 2012. [Online]. Available: https://doi.org/10.1371/journal.pone.0041911 (Citato a pagina 6)

[5] G. Giordano, M. De Santis, S. Pagano, G. Ragozini, M. P. Vitale, and P. Cavallo, *Association Rules and Network Analysis for Exploring Comorbidity Patterns in Health Systems*. Cham: Springer International Publishing, 2020, pp. 63–78. [Online].

Available: https://doi.org/10.1007/978-3-030-31463-7_5 (Citato alle pagine 6, 22 e 24)

[6] R. M. Payton J. Jones and R. J. McNally, "Bridge centrality: A network approach to understanding comorbidity," *Multivariate Behavioral Research*, vol. 56, no. 2, pp. 353–367, 2021, pMID: 31179765. [Online]. Available: https://doi.org/10.1080/00273171.2019.1614898 (Citato a pagina 7)

[7] C.-W. Huang, R. Lu, U. Iqbal, S.-H. Lin, P. A. A. Nguyen, H.-C. Yang, C.-F. Wang, J. Li, K.-L. Ma, Y.-C. J. Li, and W.-S. Jian, "A richly interactive exploratory data analysis and visualization tool using electronic medical records," *BMC Medical Informatics and Decision Making*, vol. 15, no. 1, p. 92, Nov 2015. [Online]. Available: https://doi.org/10.1186/s12911-015-0218-7 (Citato a pagina 7)

[8] N. Rostamzadeh, S. S. Abdullah, and K. Sedig, "Visual analytics for electronic health records: A review," *Informatics*, vol. 8, no. 1, 2021. [Online]. Available: https://www.mdpi.com/2227-9709/8/1/12 (Citato a pagina 7)

[9] H. Lu and S. Uddin, "Embedding-based link predictions to explore latent comorbidity of chronic diseases," *Health Information Science and Systems*, vol. 11, no. 1, p. 2, 2022. [Online]. Available: https://doi.org/10.1007/s13755-022-00206-7 (Citato a pagina 8)

[10] R. J. Woodman, B. Koczwara, and A. A. Mangoni, "Applying precision medicine principles to the management of multimorbidity: the utility of comorbidity networks, graph machine learning, and knowledge graphs," *Frontiers in Medicine*, vol. 10, p. 1302844, 2024. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fmed.2023.1302844/full (Citato a pagina 8)

[11] S. Biswas, K. D. Chaudhuri, P. Mitra, and K. S. Rao, "Relation predictions in comorbid disease centric knowledge graph using heterogeneous gnn models," in *Bioinformatics and Biomedical Engineering*, I. Rojas, O. Valenzuela, F. Rojas Ruiz, L. J. Herrera, and F. Ortuño, Eds. Cham: Springer Nature Switzerland, 2023, pp. 343–356. (Citato a pagina 9)

[12] R. Bing, G. Yuan, M. Zhu, F. Meng, H. Ma, and S. Qiao, "Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications," *Artificial Intelligence Review*, vol. 56, no. 8, pp. 8003–8042, 2023. [Online]. Available: https://doi.org/10.1007/s10462-022-10375-2 (Citato a pagina 10)

[13] N. Shahid, T. Rappon, and W. Berta, "Applications of artificial neural networks in health care organizational decision-making: A scoping review," *PLOS ONE*, vol. 14, no. 2, p. e0212356, 2019. [Online]. Available: https://doi.org/10.1371/journal.pone.0212356 (Citato alle pagine 10 e 11)

[14] P. Cavallo, S. Pagano, M. De Santis, and E. Capobianco, "General practitioners records are epidemiological predictors of comorbidities: An analytical cross-sectional 10-year retrospective study," *Journal of Clinical Medicine*, vol. 7, no. 8, p. 184, 2018. [Online]. Available: https://doi.org/10.3390/jcm7080184 (Citato alle pagine 11, 12 e 13)

[15] A. O. Adeniyi, C. A. Okolo, T. Olorunsogo, and O. Babawarun, "Leveraging big data and analytics for enhanced public health decision-making: A global review," *GSC Advanced Research and Reviews*, vol. 18, no. 2, pp. 450–456, 2024. [Online]. Available: https://doi.org/10.30574/gscarr.2024.18.2.0078 (Citato a pagina 13)