Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Grafové neurónové siete a ich využitie pri získavaní poznatkov z relačných databáz

Bakalárska práca

2025
Štefan Beluško

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Grafové neurónové siete a ich využitie pri získavaní poznatkov z relačných databáz

Bakalárska práca

Študijný program:	Aplikovaná informatika
Študijný odbor:	Aplikovaná informatika
Školiace pracovisko:	Katedra aplikovanej informatiky
Školiteľ:	doc. RNDr. Tatiana Jajcayová, PhD.
Konzultant:	Dipl. Ing. Dagmar Ceľuchová Bošanská

Bratislava, 2025
Štefan Beluško

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Štefan Beluško
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Grafové neurónové siete a ich využitie pri získavaní poznatkov z relačných databáz
Graph neural networks and their use in obtaining knowledge from relational databases

Anotácia: Grafové údaje sú všade okolo nás - akýkoľvek systém pozostávajúci z entít a vzťahov medzi nimi možno reprezentovať ako graf. Napriek tomu však boli dlho pri pokroku v oblasti hĺbkového učenia (deep learning) do značnej miery ignorované. Údaje sú v bežných informačných systémov štandardne ukladané v relačných databázach, čo tiež predstavuje istú bariériu pri spracovávaní údajov vo forme grafov. Až v posledných rokoch sa grafové neurónové siete (ďalej len "GNNs") de facto stali kľúčovým nástrojom na riešenie reálnych problémov v mnohých úplne odlišných a zdanlivo nesúvisiacich oblastiach, ako je napríklad objavovanie liekov, odporúčacie systémy, predpovedanie dopravy a mnohé ďalšie, v ktorých zlyhávajú tradičnejšie metódy.

Cieľom bakalárskej práce bude zmapovať, v ktorých oblastiach majú GNNs najväčší potenciál a aké sú ich silné a slabé stránky pri riešení problémov v týchto oblastiach. Vzhľadom na to, že obrovské množstvo údajov je uložených v relačných databázach, praktická časť bakalárskej práce sa bude venovať takzvanému relačnému hĺbkovému učeniu, ktoré využíva GNNs na učenie sa užitočných vzorov a vnorení ("embeddings") priamo z relačnej databázy bez akéhokoľvek inžinierstva vstupných premenných ("features"). Tento prístup sa porovná s inými metódami tvorby grafových údajov a ich využitia na predpovede v GNNs.

Cieľ: Ciele bakalárskej práce:
Zmapovanie oblastí využívania GNNs, ich slabé a silné stránky.
Vysvetliť spôsob ako aplikovať GNNs na relačné dáta.
Praktické odskúšanie Relational Deep Learning Benchmark https://relbench.stanford.edu/start/
Využitie vlastného dátového zdroja (pôjde o zjednodušenú verziu databázy MIMIC IV),
Úprava dostupného modelu.
Kvalitatívne porovnanie zvolenej metódy s inými metódami na spracovanie údajov cez GNNs.

Literatúra: Milan Cvitkovic: Supervised Learning on Relational Databases with Graph Neural Networks

30222831

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Matthias Fey2,*, Weihua Hu2,*, Kexin Huang1,*, Jan Eric Lenssen2,3,*, Rishabh Ranjan1,*, Joshua Robinson1,*, Rex Ying2,4, Jiaxuan You2,5, Jure Leskovec1,: Relational Deep Learning: Graph Representation Learning on Relational Databases
William L. Hamilton: Graph Representation Learning

Vedúci: doc. RNDr. Tatiana Jajcayová, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. RNDr. Tatiana Jajcayová, PhD.

Dátum zadania: 24.09.2024

Dátum schválenia: 14.10.2024 doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Čestne vyhlasujem, že celú bakalársku prácu na tému `Grafové neurónové siete a ich využitie pri získavaní poznatkov z relačných databáz`, vrátane všetkých jej príloh a obrázkov, som vypracoval samostatne, a to s použitím literatúry uvedenej v priloženom zozname a nástrojov umelej inteligencie. Vyhlasujem, že nástroje umelej inteligencie som použil v súlade s príslušnými právnymi predpismi, akademickými právami a slobodami, etickými a morálnymi zásadami za súčasného dodržania akademickej integrity.

.....

Bratislava, 2025 Štefan Beluško

Abstrakt

Grafové údaje sú všade okolo nás - akýkoľvek systém pozostávajúci z entít a vzťahov medzi nimi možno reprezentovať ako graf. Napriek tomu však boli dlho pri pokroku v oblasti hĺbkového učenia (deep learning) do značnej miery ignorované. Údaje sú v bežných informačných systémov štandardne ukladané v relačných databázach, čo tiež predstavuje istú bariériu pri spracovávaní údajov vo forme grafov. Až v posledných rokoch sa grafové neurónové siete (ďalej len "GNNs") de facto stali kľúčovým nástrojom na riešenie reálnych problémov v mnohých úplne odlišných a zdanlivo nesúvisiacich oblastiach, ako je napríklad objavovanie liekov, odporúčacie systémy, predpovedanie dopravy a mnohé ďalšie, v ktorých zlyhávajú tradičnejšie metódy.

Práca sa zaoberá zmapovaním oblastí, v ktorých majú GNNs najväčší potenciál, a analýzou ich silných a slabých stránok pri riešení problémov v týchto oblastiach. Práca sa venuje aktuálnym vedeckým otázkam v oblasti efektívneho spracovania grafových dát - odkiaľ ich možno získať, či, ako a do akého formátu ich transformovať, v akých databázach ich ukladať. Vzhľadom na to, že obrovské množstvo údajov je uložených v relačných databázach, praktická časť práce sa venuje takzvanému relačnému hĺbkovému učeniu, ktoré využíva GNNs na učenie sa užitočných vzorov a vnorení (embeddings) priamo z relačnej databázy bez akéhokoľvek inžinierstva vstupných premenných ("features"). Tento prístup sa porovná s inými metódami tvorby grafových údajov a ich využitia na predpovede v GNNs.

Kľúčové slová: grafové neurónové siete, relačné databázy, predikcia klinických výsledkov, MIMIC-IV, RelBench

Abstract

Graph-structured data are all around us—any system composed of entities and relationships between them can be represented as a graph. Despite this, such data were long overlooked in the advancement of deep learning. Information systems typically store data in relational databases, which creates a barrier for processing them as graphs. Only in recent years have Graph Neural Networks (GNNs) emerged as a key tool for solving real-world problems in many diverse and seemingly unrelated domains, such as drug discovery, recommendation systems, traffic prediction, and others where more traditional methods often fail.

The goal of this thesis is to map out the domains where GNNs show the greatest potential, as well as to examine their strengths and limitations in solving problems within those domains. This thesis explores current research challenges in the area of efficient graph data processing—how to obtain such data, whether and how to transform them into suitable formats, and what types of databases to use for storage. Given that a vast amount of data is stored in relational databases, the practical part of the thesis focuses on so-called relational deep learning, which leverages GNNs to learn useful patterns and embeddings directly from relational databases without any manual feature engineering. This approach is compared with other methods for generating graph data and using them in GNN-based prediction tasks.

Keywords: graph neural networks, relational databases, clinical outcome prediction, MIMIC-IV, RelBench

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

V ére digitálneho zdravotníctva sa čoraz viac kladie dôraz na využitie rozsiahlych klinických databáz na podporu rozhodovania, zlepšenie liečby a efektívnejšiu správu pacientov. Jednou z najvýznamnejších verejne dostupných databáz v tejto oblasti je Medical Information Mart for Intensive Care (MIMIC), ktorá poskytuje podrobné a dlhodobo zbierané dáta o pacientoch hospitalizovaných na jednotkách intenzívnej starostlivosti (JIS). Novšia verzia tejto databázy, MIMIC-IV, obsahuje stovky miliónov záznamov usporiadaných v rozsiahlej relačnej štruktúre, ktorá zachytáva komplexné prepojenia medzi pacientmi, hospitalizáciami, procedúrami, medikáciou, vitálnymi funkciami a voľnými textovými poznámkami.

Táto bohatá štruktúra však predstavuje výzvu pre tradičné modely strojového učenia, ktoré sú navrhnuté predovšetkým na spracovanie plochých tabuliek. Klasické prístupy často vyžadujú rozsiahle predspracovanie a inžinierstvo príznakov, čím dochádza k strate časti informácií a možnému narušeniu časových závislostí. Tieto obmedzenia motivujú výskum nových techník, ktoré dokážu efektívne modelovať komplexnú štruktúru údajov.

Grafové neurónové siete (GNN) predstavujú sľubnú alternatívu, ktorá umožňuje explicitne pracovať s prepojeniami medzi entitami. Na rozdiel od tradičných modelov umožňujú GNN zachytiť vzťahy medzi pacientmi, procedúrami, meraniami a ďalšími klinickými udalosťami prostredníctvom grafovej reprezentácie. Táto schopnosť je mimoriadne cenná v doménach, kde sú vzťahy medzi entitami kľúčové, ako je práve zdravotníctvo.

Cieľom tejto práce je preskúmať, do akej miery dokážu grafové neurónové siete využiť štruktúrované údaje z databázy MIMIC-IV na predikciu klinických výsledkov, konkrétne dĺžky pobytu pacienta na JIS. Zameriavame sa na binárnu klasifikačnú úlohu: predikovať, či pacient zostane na JIS tri alebo viac dní. V rámci práce využívame benchmarkový rámec RelBench, ktorý umožňuje trénovanie GNN modelov priamo nad relačnými databázami bez potreby splošťovania dát do jednej tabuľky.

Práca je rozdelená do niekoľkých tematických celkov. V prvej kapitole predstavujeme teoretické východiská: vysvetľujeme pojem závislostne orientovaných údajov, motiváciu pre grafové modelovanie a princípy fungovania grafových neurónových sietí. Uvádzame aj výhody a obmedzenia GNN, osobitne v kontexte klinických údajov, kde

je dôležitá interpretovateľnosť a robustnosť modelov. Detailne opisujeme databázu MIMIC-IV a ilustrujeme, ako možno jej relačnú štruktúru reprezentovať ako graf.

V ďalšej kapitole sa venujeme výskumnému kontextu a cieľom. Analyzujeme výhody učenia na grafoch v porovnaní s tradičnými tabuľkovými modelmi, diskutujeme prístupy založené na grafoch podobnosti medzi pacientmi a uvádzame súvisiace práce, ako napríklad model Graformer, ktorý využíva transformerovú architektúru nad grafom podobnosti pacientov.

Tretia kapitola sa zameriava na implementáciu a experimentálny rámec. Detailne opisujeme postup spracovania údajov z MIMIC-IV pre použitie v RelBenchi – vrátane výberu relevantných pacientov, filtrovania záznamov, čistenia tabuliek a konštrukcie relačného grafu. Následne opisujeme architektúru použitého modelu, embedding textových čŕt, konfiguráciu trénovania a výber hodnotiacich metrík. Špeciálna pozornosť je venovaná návrhu samotnej predikčnej úlohy a zabezpečeniu správneho časového rozdelenia údajov.

Štvrtá kapitola obsahuje výsledky a ich vyhodnotenie. Prezentujeme numerické výsledky modelu, analyzujeme ROC a PR krivky, kalibráciu a maticu zámen. Výsledky ukazujú, že model dokáže s určitou mierou úspešnosti predikovať dĺžku pobytu, no zároveň upozorňujú na vysoký šum v dátach a limitácie zvoleného embeddera. Porovnávame naše výsledky s modelom Graformer, ktorý dosiahol vyššiu výkonnosť.

V diskusii sumarizujeme kľúčové zistenia, hodnotíme prínosy a obmedzenia zvoleného prístupu a navrhujeme konkrétne kroky pre jeho zlepšenie. Práca tiež identifikuje výzvy pri spracovaní veľkých a heterogénnych databáz a zdôrazňuje dôležitosť kombinovania technickej precíznosti s doménovou expertízou.

Táto práca tak predstavuje príspevok k rastúcej oblasti využívania grafových neurónových sietí nad relačnými klinickými údajmi a zároveň slúži ako prípadová štúdia aplikácie moderného GNN rámca na reálnu klinickú databázu. Zároveň upozorňuje na praktické úskalia pri práci s dátami zo zdravotníctva a potrebu modelov, ktorých rozhodnutia sú dostatočne zrozumiteľné a overiteľné aj pre klinických odborníkov. Verím, že skúsenosti prezentované v tejto práci môžu inšpirovať ďalší výskum zameraný na zlepšenie predikčnej presnosti pri zachovaní vysokej transparentnosti a dôveryhodnosti modelu.

Kapitola 1

Teoretické východiská: Grafové modelovanie a GNN v zdravotníctve

Táto kapitola predstavuje teoretické základy potrebné pre porozumenie grafovo orientovaných údajov a ich spracovania pomocou grafových neurónových sietí (GNN). Predstavíme si, čo znamená, že údaje sú orientované na závislosti, prečo sa takéto štruktúry vyskytujú v reálnych doménach a ako ich možno efektívne modelovať. Následne sa zameriame na klinický dataset MIMIC-IV, ktorý je príkladom rozsiahlej databázy s bohatou štruktúrou vzťahov, a ukážeme si, ako sa dá reprezentovať ako graf. V druhej časti kapitoly vysvetlíme princípy fungovania grafových neurónových sietí, ich silné a slabé stránky, a predstavíme možnosti ich využitia v oblasti zdravotníctva. Osobitnú pozornosť venujeme aj interpretovateľnosti modelov, ktorá je kľúčová pri ich nasadzovaní v citlivých oblastiach, ako je zdravotná starostlivosť.

1.1 Závislostne orientované údaje

Grafy sú neodmysliteľnou súčasťou mnohých dátových setov z reálneho sveta, od sociálnych sietí a biológie až po odporúčacie systémy a zdravotníctvo. Závislostne orientované údaje predstavujú dátové štruktúry, v ktorých zohrávajú kľúčovú úlohu explicitné vzťahy medzi entitami a často definujú štruktúru údajov. Na rozdiel od údajov neorientovaných na závislosti, kde sú záznamy väčšinou nezávislé, v súboroch orientovaných na závislosti možno analyzovať prepojenia a interakcie medzi dátovými bodmi.

Táto kapitola opisuje, ako možno tradičné tabuľkové údaje interpretovať ako grafy a ako grafové neurónové siete (GNN) využívajú túto štruktúru na učenie.

Väčšina tradičných datasetov v strojovom učení má podobu multidimenzionálnych tabuliek, ktoré sa označujú ako *nondependency-oriented data*. Avšak, ako sa uvádza v učebnici Data Mining [1]:

"Network data are a very general representation and can be used for solving

> many similarity-based applications on other data types. For example, multidimensional data may be converted to network data by creating a node for each record in the database, and representing similarities between nodes by edges. Such a representation is used quite often for many similarity-based data mining applications, such as clustering. It is possible to use community detection algorithms to determine clusters in the network data and then map them back to multidimensional data."

Klasickým príkladom na ilustráciu tohto konceptu je súbor údajov o Titanicu. Hoci sa s ním tradične zaobchádza ako s tabuľkovými údajmi, možno ho reinterpretovať ako graf: cestujúci (uzly) prepojení vzťahmi, ako sú rodinné väzby, blízkosť kabín alebo zdieľanie lístkov. Táto transformácia na sieťovú reprezentáciu umožňuje vykonať sofistikovanejšie analýzy, ktoré sú založené na prepojeniach medzi entitami – napríklad predpovedanie prežitia na základe sociálnych zoskupení.

Podľa Aggarwala [1] sú sieťové údaje všeobecnou reprezentáciou, ktorá umožňuje aplikácie založené na podobnosti v rôznych typoch údajov. Viacrozmerné údaje, ktoré nie sú orientované na závislosti, možno previesť na sieťové údaje. Pre každý záznam sa vytvorí uzol a medzi uzlami sa vytvoria hrany vyjadrujúce podobnosť alebo vzťahy. Takto možno aj pôvodne nezávislé súbory údajov transformovať do závislostne orientovaných formátov s cieľom odhaliť hlbšie poznatky.

1.2 Dataset MIMIC-IV

Medical Information Mart for Intensive Care IV (MIMIC-IV) [6] je významným príkladom rozsiahleho, verejne dostupného súboru údajov z oblasti zdravotníctva orientovaného na závislosti. MIMIC-IV, ktorý pokrýva viac ako desať rokov (2008 - 2019) hospitalizácií v Beth Israel Deaconess Medical Center, ponúka podrobné klinické údaje vrátane meraní pacientov, liečby, diagnóz, postupov a deidentifikovaných voľných textových poznámok.

MIMIC-IV je uložený v relačnej databázovej štruktúre a jeho štruktúra prirodzene odráža závislosti medzi entitami, ako sú pacienti, hospitalizácie, pobyty na jednotkách intenzívnej starostlivosti, laboratórne merania a klinické poznámky. Táto relačná štruktúra umožňuje výskumníkom modelovať klinické postupy pacientov ako siete, kde uzly predstavujú pacientov, návštevy, liečby alebo dokonca klinické udalosti a hrany predstavujú časové alebo logické prepojenia medzi nimi.

Celkový proces spracovania údajov v MIMIC-IV – od ich pôvodných klinických systémov (napr. MetaVision) až po výslednú deidentifikovanú a modulárne štruktúrovanú databázu – je znázornený na obrázku 1.1. Ten zachytáva ako konverziu formátov a spätnú väzbu používateľov, tak aj rozdelenie údajov do hlavných modulov *hospital*,

icu a *note*, ktoré sú opísané nižšie.

- ***hosp:*** Obsahuje záznamy na úrovni celej nemocnice, ako sú objednávky liekov, presuny pacientov, účtované výkony, predpisovanie a podávanie liekov, laboratórne výsledky a objednávky poskytovateľa. Stĺpec *subject_id* je prítomný vo všetkých tabuľkách a umožňuje prepojenie s demografickými údajmi pacienta v tabuľke *patients*. Stĺpec *hadm_id* je tiež prítomný vo všetkých tabuľkách a predstavuje jednu hospitalizáciu.

- ***icu:*** Obsahuje podrobné údaje o pacientoch na jednotkách intenzívnej starostlivosti (JIS), napríklad vitálne funkcie, podanú liečbu či laboratórne výsledky. Tieto údaje boli zaznamenávané výhradne pomocou klinického systému MetaVision, ktorý bol v čase zberu údajov jediným používaným systémom na jednotkách intenzívnej starostlivosti. Údaje sú usporiadané do tabuliek podľa hviezdicovej schémy – všetky udalosti odkazujú na tabuľku *d_items*, kde sú definované typy záznamov, a na tabuľku *icustays*, ktorá identifikuje jednotlivé pobyty na jednotkách intenzívnej starostlivosti pomocou *stay_id*.

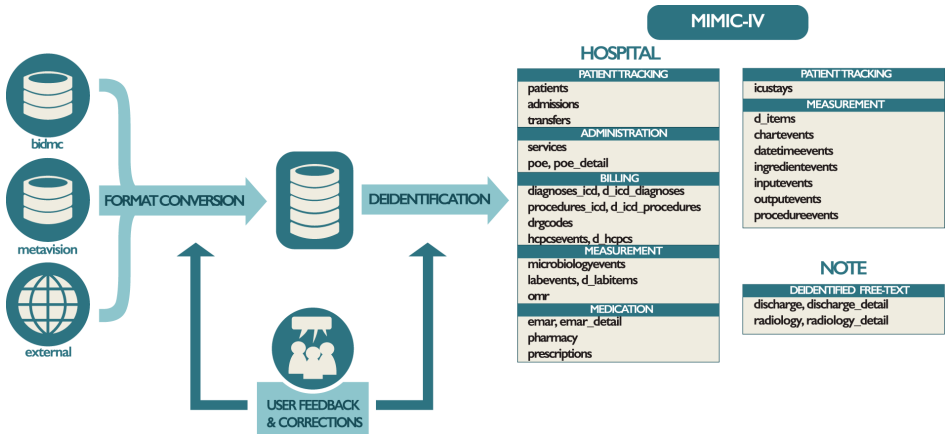
- ***note:*** Modul poznámok obsahuje deidentifikované voľné textové klinické záznamy, rozdelené do dvoch tabuliek: *discharge* a *radiology*. Tabuľka *discharge* obsahuje podrobné prepúšťacie súhrny, ktoré poskytujú prehľad o anamnéze a priebehu hospitalizácie pacienta. Tabuľka *radiology* obsahuje správy rádiológov týkajúce sa vykonaných zobrazovacích vyšetrení, vrátane röntgenových snímok, počítačovej tomografie (CT), magnetickej rezonancie (MRI) a ultrazvuku.

Táto architektúra uľahčuje skúmanie modelov závislosti, napríklad ako sa vyvíjajú rozhodnutia o liečbe počas pobytu na jednotke intenzívnej starostlivosti alebo ako klinické poznámky korelujú s laboratórnymi výsledkami.

Napriek bohatosti systému MIMIC-IV väčšina štúdií využíva len časť dostupných informácií. Plnohodnotné využitie si vyžaduje rozsiahle inžinierstvo príznakov, ktoré je vzhľadom na zložitosť a heterogenitu údajov náročné a zahŕňa dôkladné predbežné spracovanie vrátane časového zosúladenia, normalizácie a ošetrenia chýbajúcich hodnôt.

1.2.1 Výzvy a príležitosti v analýze MIMIC-IV orientovanej na závislosti

Práca s MIMIC-IV je príkladom niekoľkých výziev, ktoré sú neoddeliteľnou súčasťou prieskumu závisle orientovaných údajov:[1]



Obr. 1.1: Vizualizácia pôvodu, spracovania a členenia údajov v databáze MIMIC-IV. Údaje pochádzajú z viacerých zdrojov (napr. BIDMC, MetaVision) a po konverzii a deidentifikácii sú rozdelené do modulov *hospital*, *icu* a *note* [6].

- **Komplexné inžinierstvo príznakov:** Klinické udalosti sú časovo označené, ale asynchrónne zaznamenané v rôznych systémoch, čo si vyžaduje zosúladenie a integráciu.
- **Nedostatočné a zašumené údaje:** Pri zhromažďovaní lekárskych údajov sa často uprednostňuje klinická prax a administratíva – napríklad vykazovanie úkonov pre poisťovne – pred výskumným využitím. To vedie k záznamom, ktoré môžu byť chýbajúce, nadbytočné, nepresné alebo nekonzistentné.
- **Škálovateľnosť:** Veľký objem údajov (státisíce hospitalizácií a pobytov na jednotkách intenzívnej starostlivosti) si vyžaduje efektívne algoritmy schopné zvládnuť relačné spájanie a konštrukciu grafov v širokom rozsahu.
- **Deidentifikácia a etické obmedzenia:** Prísne požiadavky na ochranu osobných údajov komplikujú prepojenie pacientov z rôznych zdrojov údajov.

Tieto výzvy však vyvažujú významné príležitosti. Modelovanie MIMIC-IV ako grafovej štruktúry umožňuje pokročilé aplikácie, ako je identifikácia pacientov s podobnými priebehmi, detekcia odchýlok v klinických postupoch, predikcia výsledkov na základe priebehu liečby a efektívna selekcia pacientskych kohort pre výskumné analýzy.

Prevod relačných štruktúr na grafy je navyše v súlade s modernými prístupmi strojového učenia vrátane grafových neurónových sietí (GNN), ktoré sa ukázali ako sľubné pri učení komplexných závislostí v údajoch zo zdravotníctva.

MIMIC-IV je teda jedinečným a neoceniteľným zdrojom pre výskumníkov, ktorí sa zaujímajú o získavanie údajov orientovaných na závislosti, a ponúka komplexnosť reálneho sveta, bohaté vzťahové štruktúry a hlboký spoločenský vplyv prostredníctvom potenciálnych poznatkov v oblasti zdravotníctva. [6]

Tento dataset je dostupný výskumníkom po absolvovaní tréningu ochrany osobných údajov a podpise licenčnej zmluvy. Údaje sú prístupné prostredníctvom platformy PhysioNet vo formáte CSV, čo zjednodušuje analýzu pomocou moderných nástrojov na spracovanie dát. [5]

Prečo grafová reprezentácia v zdravotníctve?

Zdravotnícke údaje majú prirodzene bohatú štruktúru vzťahov – napríklad medzi pacientmi, hospitalizáciami, procedúrami, laboratórnymi výsledkami či liečbou. Tieto vzťahy nie sú náhodné, ale nesú významné informácie o priebehu ochorení, odpovedi na liečbu a o interakciách medzi rôznymi klinickými faktormi. V tradičných prístupoch sa často ignorujú alebo sa len nepriamo premietajú do vektorovej podoby.

Grafová reprezentácia umožňuje tieto prepojenia modelovať explicitne: uzly reprezentujú entity (pacientov, merania, hospitalizácie) a hrany zachytávajú vzťahy alebo interakcie medzi nimi. To umožňuje algoritmom efektívne „vidieť“ kontext každého prvku v rámci siete, čo vedie k presnejším predikciám a interpretáciám.

V oblasti zdravotníctva je tento prístup obzvlášť výhodný – napríklad dvaja pacienti s podobnou liečbou a diagnózami môžu byť prepojení a ich klinické dráhy porovnávané. Vzťahy medzi liekmi a laboratórnymi výsledkami môžu indikovať interakcie alebo vedľajšie účinky, ktoré by sa inak stratili v tabuľkových reprezentáciách. Grafová reprezentácia tak otvára nové možnosti pre analýzu a personalizovanú medicínu.

1.3 Grafové neurónové siete (GNN)

Grafové neurónové siete (GNN) predstavujú inovatívny prístup v hlbkovom učení, ktorý umožňuje efektívne modelovať a analyzovať údaje s komplexnou grafovou štruktúrou. Na rozdiel od tradičných modelov, ktoré pracujú s údajmi v euklidovských štruktúrach ako sú napríklad mriežky alebo postupnosti, GNN umožňujú modelovanie neeuklidovských štruktúr, v ktorých sú uzly (entity) prepojené hranami (vzťahmi). Mnohé reálne systémy, ako molekuly, sociálne siete, biologické siete či zdravotnícke databázy, sú prirodzene reprezentované ako grafy.

Kľúčovým konceptom GNN je *relational inductive bias* – predpoklad, že štruktúra vzťahov medzi entitami obsahuje významné informácie pre učenie. GNN sú preto navrhnuté tak, aby využívali topológiu grafu a vlastnosti uzlov na vytváranie reprezentácií, ktoré sú nezávislé od poradia susedov a veľkosti grafu. Vychádzajú z princípu *homophily*, podľa ktorého majú susedné uzly tendenciu zdieľať podobné vlastnosti. Modely GNN propagujú informácie medzi prepojenými entitami iteratívnym agregovaním a aktualizovaním reprezentácií uzlov na základe ich susedov, čím zachytávajú lokálne aj globálne štruktúry grafu. Táto schopnosť vytvárať kontextovo bohaté reprezentácie

robí GNN výnimočnými pre úlohy ako klasifikácia uzlov, predikcia hrán, zhlukovanie či interpretácia celej štruktúry grafu.

GNN sú flexibilné a univerzálne – rovnaký architektonický rámec možno aplikovať na rôzne typy grafov, vrátane homogénnych, heterogénnych, dynamických a s atribútmi. Medzi známe implementácie patria Graph Convolutional Networks (GCN), Graph Attention Networks (GAT) či GraphSAGE. Ich základnou črtou je rámec odovzdávania správ *(message passing)*, ktorý umožňuje iteratívne agregovanie informácií od susedov uzla. Tento proces je diferencovateľný a dá sa efektívne optimalizovať pomocou gradientných metód [4].

GNN sa osvedčili najmä pri úlohách, kde sú prítomné:

- **Implicitné vzťahy a závislosti:** Dáta, kde nie sú všetky vzťahy explicitne vyjadrené, ale sú zásadné pre pochopenie štruktúry (napr. priateľstvá v sociálnej sieti).
- **Vysoká dimenzionalita a riedkosť prepojení:** Situácie, kde väčšina uzlov má len málo susedov alebo kde sa v údajoch často vyskytujú chýbajúce či nulové hodnoty.
- **Zložité nelokálne interakcie:** Potreba zachytiť vplyvy, ktoré presahujú priame susedstvá a vyžadujú viacero vrstiev GNN na ich zistenie.

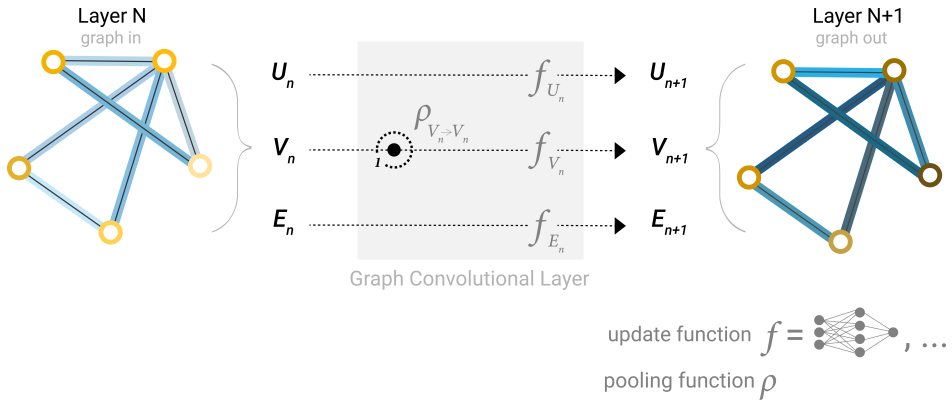
1.3.1 Rámec odovzdávania správ

Základným mechanizmom väčšiny GNN je odovzdávanie správ. Každý uzol aktualizuje svoju reprezentáciu agregovaním parametrov od svojich susedov a ich kombinovaním so svojimi vlastnými. Tento proces sa zvyčajne opakuje vo viacerých vrstvách (iteráciách), čo umožňuje šírenie informácií naprieč grafom.

Typicky sa skladá zo štyroch krokov:

- **Zber parametrov zo susedstva:** Získanie reprezentácií susedných uzlov.
- **Agregácia:** Kombinovanie reprezentácií susedov pomocou funkcie ako súčet, priemer alebo max.
- **Nelineárna transformácia:** Aplikovanie lineárnej vrstvy a nelineárnej funkcie (napr. ReLU).
- **Aktualizácia uzla:** Spojenie transformovaných údajov so stavom uzla a výpočet novej reprezentácie.

Tento štvorkrokový rámec je podrobne opísaný v [4], kde tvorí základnú schému architektúry GNN.



Obr. 1.2: Vizuálna ilustrácia grafovej konvolučnej vrstvy z Distill.pub [9], znázorňujúca mechanizmus odovzdávania správ. Uzly zbierajú informácie zo svojho okolia, ktoré sa kombinujú cez poolingové a aktualizačné funkcie do novej reprezentácie v ďalšej vrstve.

Aby sme lepšie pochopili mechanizmus odovzdávania správ, uvažujme o skupine ľudí, ktorí si vymieňajú názory v kolách. Na začiatku má každá osoba (uzol) svoj vlastný názor (vlastnosti). V každom kole (vrstve) počúvajú svojich priateľov (susedov) a upravujú svoj názor na základe toho, čo počujú. Po niekoľkých kolách názor každej osoby odráža zmes jej vlastných názorov a názorov širšej komunity. GNN formalizujú túto myšlienku pomocou diferencovateľných výpočtov, ktoré sa optimalizujú počas trénovania.

Matematicky sa proces odovzdávania správ pre uzol v vo vrstve k formuluje nasledovne:

$$h_v^{(k)} = \text{UPDATE}^{(k)}\left(h_v^{(k-1)}, \text{AGGREGATE}^{(k)}\left(\{h_u^{(k-1)} : u \in \mathcal{N}(v)\}\right)\right)$$

kde $h_v^{(k)}$ označuje reprezentáciu (embedding) uzla v v k-tej vrstve a $\mathcal{N}(v)$ je množina susedov uzla v.

Táto formulácia zabezpečuje, že reprezentácie uzlov sú invariantné voči poradiu ich susedov, čo je kľúčová požiadavka pre učenie na grafoch. Funkcie $\text{AGGREGATE}^{(k)}$ a $\text{UPDATE}^{(k)}$ sú diferencovateľné moduly, ktoré sa učia počas trénovania modelu.

Takáto architektúra umožňuje *end-to-end* trénovanie pomocou gradientného zostupu, pričom každá vrstva rozširuje pohľad uzla o jeho širšie okolie v grafe.

1.3.2 Použitie GNN

Problémy reálneho sveta často zahŕňajú subjekty, ktoré sú vzájomne prepojené zložitým a netriviálnym spôsobom. Tieto prepojenia kódujú kritické informácie, ktoré sa stratia, ak sa údaje sploštia alebo pretransformujú do bežného formátu. Grafy nám umožňujú explicitne modelovať tieto vzťahy a GNN poskytujú rámec na efektívne učenie sa z nich.

Zoberme si sociálne siete, kde sú používatelia (uzly) prepojení priateľstvami (hranami), alebo molekuly, kde sú atómy (uzly) prepojené chemickými väzbami (hranami). V oboch prípadoch zohráva štruktúra spojení zásadnú úlohu pri určovaní správania a GNN sa snažia túto štruktúru využiť pri učení.

Univerzálnosť GNN viedla k ich prijatiu v rôznych oblastiach:

- **Chémia a biológia:** Predpovedanie molekulárnych vlastností, objavovanie liečiv, predpovedanie proteínových rozhraní.
- **Sociálne siete:** Zisťovanie komunity, odhad vplyvu, odporúčanie obsahu.
- **Znalostné grafy:** Predpovedanie prepojení, klasifikácia entít, sémantické vyhľadávanie.
- **Doprava a mobilita:** Predpovedanie dopravných tokov pomocou cestných sietí.
- **Spracovanie prirodzeného jazyka:** Sémantický rozbor, riešenie koreferencií, odpovedanie na otázky pomocou znalostných grafov.

Tieto príklady poukazujú na širokú využiteľnosť GNN a dôležitosť modelovania vzťahovej štruktúry.

1.3.3 Silné stránky GNN

GNN ponúkajú niekoľko výhod, ktoré z nich robia presvedčivú voľbu:

- **Reprezentačná schopnosť:** Dokážu zachytiť lokálnu aj globálnu štruktúru grafu a vytvárať bohaté vektorové reprezentácie uzlov a grafov.
- **Induktívne učenie:** Dokážu zovšeobecňovať na nevidené uzly alebo grafy.
- **Vyhodnocovanie vzťahov medzi entitami:** Vhodné na úlohy, ktoré si vyžadujú pochopenie a využitie závislostí medzi entitami v grafe.
- **Škálovateľnosť (s variantmi):** Najnovšie architektúry ako GraphSAGE a metódy založené na vzorkovaní lepšie škálujú veľké grafy.

1.3.4 Obmedzenia GNN

Hoci GNN poskytujú silný nástroj na modelovanie grafovo štruktúrovaných údajov, v praxi čelia viacerým výzvam:

- **Výpočtová náročnosť:** Pri veľkých grafoch sa rýchlo zvyšujú pamäťové a výpočtové nároky, najmä pri modeloch, ktoré spracúvajú celé susedstvá.

- **Oversmoothing:** Ak sa GNN trénujú s príliš veľkým počtom vrstiev, reprezentácie uzlov sa môžu zlievať a strácať diskriminatívnu silu.
- **Náchylnosť na overfitting:** GNN môžu ľahko pretrénovať na štruktúru trénovacích grafov, najmä ak je málo dát alebo sú grafy zložité, čo vedie k zníženej schopnosti generalizácie.
- **Závislosť od kvality grafovej štruktúry:** Ak graf obsahuje nesprávne, chýbajúce alebo zašumené hrany, model sa môže naučiť nepresné reprezentácie.

Tieto obmedzenia sú predmetom aktívneho výskumu, ktorý sa zameriava na ich prekonanie pomocou nových architektúr, regularizačných techník, prístupov k vzorkovaniu a pokročilých tréningových stratégií.

1.3.5 Interpretovateľnosť v GNN

Grafové neurónové siete (GNN) dosahujú vynikajúce výsledky v rôznych úlohách nad grafovo štruktúrovanými dátami, avšak ich rozhodovacie procesy sú často neprehľadné. Vzhľadom na rastúce nasadenie GNN v citlivých doménach, ako je zdravotníctvo či biologický výskum, je interpretovateľnosť výstupov modelu mimoriadne dôležitá. Interpretovateľnosť zvyšuje dôveru používateľov a umožňuje odhaliť chyby či dôležité štruktúry v dátach.

GNNExplainer

GNNExplainer je prvý všeobecný, architektúrne nezávislý prístup k vysvetľovaniu rozhodnutí GNN modelov [11]. Pre daný uzol alebo graf vytvára vysvetlenie v podobe subgrafu a podmnožiny čŕt uzlov, ktoré maximalizujú vzájomnú informáciu s predikciou modelu. Optimalizácia prebieha pomocou masky nad hranami a črtami, pričom sa využíva *variational approximation*. Výsledné vysvetlenie reprezentuje najrelevantnejšie štruktúry a črty, ktoré ovplyvňujú dané rozhodnutie modelu. GNNExplainer podporuje aj vysvetlenia pre viacero prípadov (napr. pre celú triedu uzlov) pomocou prototypových subgrafov. Výhodou je univerzálnosť a schopnosť vizualizácie, nevýhodou nutnosť optimalizácie pre každý prípad zvlášť.

PGExplainer

PGExplainer rozširuje možnosti interpretácie využitím parametrického modelu, ktorý sa učí generovať vysvetlenia predikcií [7]. Využíva neurónovú sieť na učenie pravdepodobností hrán, ktoré tvoria kľúčovú podštruktúru grafu relevantnú pre rozhodnutie modelu. Model využíva generatívny prístup na úrovni hrán grafu a umožňuje hromadné

vysvetľovanie predikcií pre viaceré vstupy naraz. Hlavnou výhodou je možnosť aplikovať už natrénovaný model na generovanie vysvetlení aj na nové príklady bez potreby opätovného trénovania, čo zvyšuje efektivitu a umožňuje jeho použitie v induktívnom nastavení. PGExplainer je tak vhodný pre rozsiahle alebo dynamické grafy, kde je dôležitá škálovateľnosť a rýchlosť.

Obe metódy poskytujú dôležité nástroje na pochopenie vnútorného fungovania GNN a prispievajú k ich praktickej využiteľnosti v doménach, kde sú vysvetliteľnosť a dôveryhodnosť modelov nevyhnutné.

Kapitola 2

Výskumné otázky v oblasti

Grafové neurónové siete (GNN) sa stali výkonným rámcom na učenie štruktúrovaných údajov, najmä v oblastiach, kde sú vzťahy medzi entitami kľúčové. V kontexte elektronických zdravotných záznamov sú tieto vzťahy obzvlášť bohaté a rôznorodé. Cieľom nášho výskumu je preskúmať, či sa GNN dokážu efektívne učiť z takýchto komplexných vzťahových štruktúr a ponúknuť zmysluplné predpovede týkajúce sa výsledkov pacientov na jednotkách intenzívnej starostlivosti (JIS).

MIMIC-IV je vynikajúcim príkladom súboru údajov orientovaného na závislosti. Zachytáva heterogénne klinické údaje s vysokým rozlíšením vrátane laboratórnych výsledkov, receptov, klinických poznámok a udalostí na jednotke intenzívnej starostlivosti pre desiatky tisíc hospitalizácií za viac ako desaťročie [6]. Tieto údaje sú vo svojej podstate vzájomne prepojené - to, čo sa deje počas prvých kritických hodín prijatia na JIS, často určuje priebeh zvyšku hospitalizácie. Tradičné tabuľkové modely však často ignorujú vzťahovú štruktúru údajov a jednotlivé zdravotné záznamy posudzujú ako nezávislé jednotky. Zjednodušením relačných údajov do jedinej tabuľky prerušujú časové sekvencie a zastierajú kauzálne prepojenie medzi klinickými udalosťami [2].

Učenie založené na grafoch ponúka alternatívu: transformácia relačných databáz na grafy nám umožňuje zachovať závislosti medzi jednotlivými subjektmi [3].

V takomto grafe uzly predstavujú pacientov a ich udalostí, ako sú hospitalizácie alebo laboratórne testy, a hrany odrážajú vzťahy, ako je časové poradie alebo podobnosti. To umožňuje GNN uvažovať nad interakciami medzi vzdialenými uzlami v grafe a odhaľovať vzory, ktoré nie sú ľahko viditeľné v plochých údajov [4]. To je obzvlášť dôležité pri úlohách, ako je predpovedanie úmrtnosti alebo odhad dĺžky pobytu, kde môžu byť prediktívne jemné vzájomné závislosti medzi včasnými meraniami a následnou liečbou.

Okrem toho predchádzajúci výskum zdôraznil dôležitosť explicitného modelovania týchto interakcií. Aggarwal [1] uvádza, že mnohé aplikácie založené na podobnosti profitujú z konverzie viacrozmerných údajov do sieťovej podoby, pretože sa dajú analyzovať

štruktúry spoločenstiev a vzorce interakcií. V lekárskom prostredí by takéto vzorce mohli zodpovedať liečebným postupom aplikovaným na podobné diagnózy alebo podskupinám pacientov so spoločnými komplikáciami.

2.1 Výskumná otázka a ciele práce

Preto táto práca skúma hlavnú otázku: Môžu GNN natrénované na grafových reprezentáciách MIMIC-IV lepšie predpovedať výsledky pacientov v porovnaní s tradičnými tabuľkovými prístupmi? Okrem predikčnej presnosti je v klinickom prostredí rovnako dôležitá interpretovateľnosť modelov. GNN modely môžu byť vnímané ako „čierne skrinky“, a preto sa vyvíjajú techniky ako GNNExplainer či PGExplainer [11, 7], ktoré umožňujú identifikovať podgrafy a črty najviac ovplyvňujúce rozhodovanie modelu. Táto interpretovateľnosť je kľúčová pre dôveru regulátorov a poskytovateľov zdravotnej starostlivosti, ako aj pre praktické nasadenie v zdravotníctve.

2.2 Dodatočná výskumná otázka

Okrem hlavného cieľa práce sa venujeme aj doplňujúcej výskumnej otázke: Je možné efektívne využiť grafové neurónové siete priamo nad relačnou štruktúrou databázy MIMIC-IV, v ktorej sú údaje pôvodne uložené? Alebo je nevyhnutné venovať výraznú pozornosť návrhu grafu, definícii entít a inžinierstvu vstupných premenných? Táto otázka reflektuje praktickú výzvu pri použití GNN v kontexte zdravotníctva a síce, že samotná relačná štruktúra databázy nemusí byť priamo vhodná na učenie bez ďalšieho spracovania alebo obohatenia o doménové poznatky. Preto je cieľom práce vyhodnotiť praktickú využiteľnosť relačného hlbokého učenia na reálnych klinických údajoch, pochopiť podmienky, za ktorých sú takéto modely úspešné, a poukázať na výhody modelovania založeného na grafoch v strojovom učení v zdravotníctve.

Zameriavame sa tak nielen na predikčnú presnosť, ale aj na širšie praktické a metodologické aspekty návrhu týchto modelov, ktoré sú kľúčové pre ich reálne nasadenie.

2.3 GNNs založené na grafoch podobnosti

Jedným z prístupov k použitiu grafových neurónových sietí (GNN) v kontexte údajov o zdravotnej starostlivosti je vytvorenie grafu podobnosti nad pacientmi. V tomto prostredí každý uzol predstavuje pacienta a hrany kódujú párové podobnosti na základe klinických vlastností. Táto myšlienka sa zhoduje so širšou koncepciou v dolovaní údajov, kde možno ľubovoľné typy údajov previesť na sieťové reprezentácie na učenie založené na podobnosti [1]. Krok konštrukcie grafu je kritický, pretože kvalita a

sémantika výsledných hrán výrazne ovplyvňuje následné učenie.

Navrhnúť účinnú metriku podobnosti však nie je triviálne. Klinické údaje sú heterogénne, zašumené a často neúplné, čo spôsobuje, že štandardné metriky vzdialenosti, ako je euklidovská alebo kosínusová podobnosť, sú nedostatočné. Okrem toho definovanie podobnosti medzi pacientmi zahŕňa rozhodnutia špecifické pre danú oblasť, ako napríklad, ktoré vlastnosti zahrnúť (napr. kódy diagnóz, laboratórne hodnoty, demografické údaje) a ako zohľadniť časové vzory alebo chýbajúce údaje.

Najnovšie modely ako Graphormer [8] skúmajú použitie grafov populácie pacientov založených na podobnosti na nekontrolované predtrénovanie. Hoci využívajú štruktúru na úrovni populácie na zlepšenie predpovedania výsledkov pacientov, hlbšie sa nezaoberajú výzvou konštrukcie zmysluplných grafov podobnosti. Toto obmedzenie je spoločné mnohým prácam v tejto oblasti, v ktorých sa predpokladá, že graf je daný, alebo je skonštruovaný pomocou jednoduchých heuristík bez dôsledného vyhodnotenia toho, ako jeho návrh ovplyvňuje výkonnosť modelu alebo interpretovateľnosť. V mnohých prípadoch sú takéto grafy podobnosti vytvárané na základe kvantitatívnych atribútov pacientov, pričom sa používa niektorá metrika vzdialenosti (napr. Euklidovská alebo Mahalanobisova), podľa ktorej sa následne pre každý uzol vyberie k najbližších susedov (k-NN) a vytvoria sa hrany. Takýto prístup je podrobne popísaný v [1].

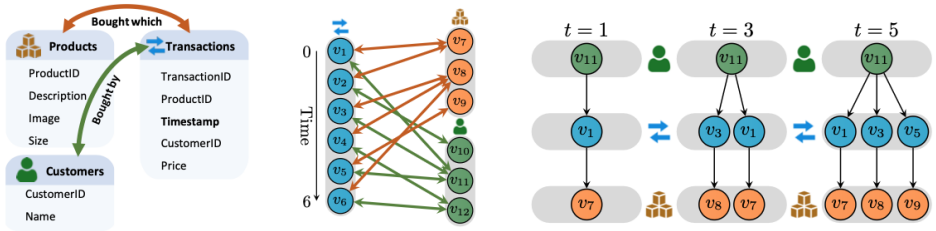
GNN založené na podobnosti sú preto sľubným, ale zatiaľ nedostatočne preskúmaným smerom. Vývoj principiálnych funkcií podobnosti, ktoré zohľadňujú konkrétne úlohy ako je klasifikácia uzla alebo predpovedanie existencie hrany, a integrácia doménových znalostí do konštrukcie grafov zostávajú otvorenými výzvami. Ich riešenie by mohlo viesť k vernejším a zovšeobecnenejším reprezentáciám pacientov, najmä v podmienkach, kde nie sú ľahko dostupné relačné údaje alebo keď interakcie medzi pacientmi ponúkajú cenné induktívne zaujatie.

2.4 Grafové učenie nad relačnými databázami

Ďalší sľubný prístup zahŕňa trénovanie neurónových sietí priamo na relačných databázach. Túto paradigmu umožňujú grafové neurónové siete (GNN), ktoré môžu prirodzene pracovať s relačnými údajmi tým, že považujú entity za uzly a vzťahy primárneho a cudzieho kľúča za hrany. Na rozdiel od tradičných prístupov, ktoré si vyžadujú sploštenie relačných údajov do jednej tabuľky - čo často vedie k strate informácií - GNN zachovávajú pôvodnú štruktúru a umožňujú učenie *end-to-end* nad schémami s viacerými tabuľkami [3, 2].

Tento prístup je dobre ilustrovaný v nedávnej práci [3], kde sa databáza s viacerými tabuľkami (napr. zákazníci, transakcie, produkty) transformuje na rôzne úrovne grafovej abstrakcie. Ako je znázornené na obrázku 2.1, proces sa začína grafom schémy

(a), ktorý zachytáva, ako tabuľky súvisia prostredníctvom kľúčov. Potom sa vytvorí relačný graf entít (b), kde sa každý riadok v každej tabuľke stáva uzlom a spojenia založené na kľúčoch tvoria hrany. Nakoniec sa pre každú predpoveď dynamicky vytvorí graf výpočtu (c), ktorý umožňuje GNN šíriť informácie z okolitého relačného kontextu.



(a) Schema Graph (b) Relational Entity Graph (c) Computation Graphs for different time t

Obr. 2.1: Tri úrovne grafových reprezentácií pre relačné databázy, upravené z [3]: (a) Graf schémy, kde sú tabuľky prepojené prostredníctvom vzťahov primárnych a cudzích kľúčov. (b) Graf relačných entít, v ktorom sú jednotlivé riadky reprezentované ako uzly a prepojené podľa vzťahov založených na kľúčoch. (c) Výpočtové grafy v rôznych časových krokoch t, ktoré ukazujú, ako GNN môžu agregovať informácie z okolia uzla v priebehu času na úlohy, ako je napríklad predikcia.

Táto reprezentácia založená na grafoch umožňuje neurónovým modelom štruktúrovane a expresívne spracovávať komplexné relačné údaje. Týmto prístupom sa vyhýbame potrebe ručne vytváraných vstupných premenných a namiesto toho sa spoliehame na induktívne zovšeobecňovanie, ktoré je schopné zachytiť zložité interakcie medzi entitami. GNN modely môžu využiť štruktúru relácií ako prostriedok na propagáciu informácií medzi entitami a ich kontextom, čo je kľúčové pri spracovaní klinických údajov s viacnásobnými vzťahmi a hierarchickými štruktúrami.

Avšak aplikácia grafového učenia na rozsiahle, heterogénne databázy ako MIMIC-IV predstavuje zásadnú výskumnú výzvu. Klinické dáta v tejto databáze sú často riedke, zašumené, obsahujú chýbajúce hodnoty a trpia nejednotným kódovaním. Okrem toho sa v praxi vyskytujú problémy, ako sú nekonzistentné časové pečiatky medzi tabuľkami, rôzne jednotky merania (napr. mg/dL vs mmol/L), duplicitné alebo neštruktúrované klinické poznámky a nejednoznačne definované kategórie údajov. Tieto faktory komplikujú predspracovanie a modelovanie, a zvyšujú riziko zlej generalizácie modelu [3, 2].

Výzvou nie je len návrh architektúry modelu, ale aj zabezpečenie transparentnosti, interpretovateľnosti a replikovateľnosti v celom spracovateľskom reťazci. Vzhľadom na množstvo heuristických rozhodnutí počas predspracovania môže dôjsť k neúmyselnej strate alebo deformácii relevantných informácií.

Ostáva preto otvorenou otázkou, či grafové neurónové siete dokážu efektívne spracovať takéto komplexné a neúplné súbory klinických údajov, ktoré sú rozdelené medzi viaceré tabuľky a často obsahujú nepravidelnosti. Dôležité je aj porovnanie týchto

metód s alternatívnymi prístupmi – napríklad prístupmi založenými na grafoch podobnosti, kde sú pacienti reprezentovaní ako uzly a medzi nimi sú vytvárané hrany na základe vypočítanej podobnosti ich klinických charakteristík. Táto podobnosť môže byť odvodená z rôznych aspektov, ako sú demografické údaje, diagnózy, výsledky laboratórnych testov alebo liečebné postupy. V praxi to znamená, že pacienti, ktorí majú podobné priebehy hospitalizácie alebo klinické nálezy, budú v grafe silnejšie prepojení. Takéto grafy môžu byť vytvorené pomocou metód, ako sú metriky vzdialenosti (napr. kosínusová alebo Mahalanobisova vzdialenosť a následné určenie najbližších susedov), alebo prostredníctvom učenia podobnosti pomocou autoenkodérov či iných techník reprezentácie. Hoci tieto grafy často nevychádzajú zo schémy databázy, umožňujú zachytiť latentné vzťahy medzi pacientmi a sú vhodné najmä v scenároch, kde nie je k dispozícii explicitná relačná štruktúra alebo je príliš zložitá na priamu analýzu. Takéto grafy, aj keď často konštruované heuristicky, umožňujú využiť sémantickú blízkosť medzi pacientmi aj bez explicitného modelovania relačných štruktúr. [8]

Tento kontrast medzi relačnými a podobnostnými prístupmi zároveň otvára priestor na hlbšie skúmanie ich výhod, obmedzení a potenciálu ich kombinácie v rámci hybridných modelov pre klinické predikcie.

Kapitola 3

Implementácia a vyhodnotenie modelu

V tejto kapitole opisujeme, ako sme navrhli a implementovali experiment s využitím RelBench na vyhodnotenie prediktívneho výkonu GNN na klinicky štruktúrovanom súbore údajov inšpirovanom systémom MIMIC-IV.

3.1 Relačné hĺbkové učenie a RELBENCH Benchmark

Tradičné modely strojového učenia na relačných údajoch často vyžadujú rozsiahle manuálne inžinierstvo príznakov – najmä spájaním a agregáciou tabuliek do jednej plochej štruktúry. Tento prístup je časovo náročný, náchylný na chyby a často vedie k strate informácií. Na prekonanie týchto obmedzení bola navrhnutá nová paradigma nazvaná relačné hĺbkové učenie (RDL) [3]. RDL zavádza komplexný prístup k učeniu, ktorý priamo pracuje s relačnými databázami s viacerými tabuľkami tak, že ich reprezentuje ako relačné grafy entít, kde každý riadok zodpovedá uzlu grafu a vzťahy primárneho a cudzieho kľúča definujú hrany.

Tento prístup eliminuje potrebu manuálneho inžinierstva príznakov využitím grafových neurónových sietí (GNN) na šírenie informácií naprieč relačnými štruktúrami. Kľúčovou zložkou RDL je časové obmedzenie, ktoré zabezpečuje, aby počas odovzdávania správ prúdili informácie len od entít so skoršími časovými značkami, čím sa zachováva kauzalita udalostí a zabezpečí, že model čerpá informácie výlučne z tréningových a validačných dát. Táto konštrukcia chráni pred únikom informácií, najmä v oblastiach, kde je kľúčové zachovanie časovej následnosti udalostí, ako je zdravotníctvo.

S cieľom podporiť reprodukovateľnosť a uľahčiť empirický výskum v tejto novej paradigme autori predstavujú RELBENCH, súbor benchmarkov a Python na vyhodnocovanie modelov strojového učenia na relačných údajoch. RELBENCH poskytuje:

- Rôznorodý súbor relačných dát z reálnych oblastí, ako je elektronický obchod a online fóra,
- preddefinované prediktívne úlohy s príslušnými tréningovými tabuľkami,

- nástroje na transformáciu relačných údajov do relačných grafov,
- podporu pre časové rozdelenie údajov a odovzdávanie správ,

Každá prediktívna úloha v systéme RELBENCH je definovaná pomocou tréningovej tabuľky obsahujúcej značky vypočítané z historických údajov spolu s cudzími kľúčmi a časovými značkami. Tieto trénovacie tabuľky umožňujú širokú škálu úloh učenia s učiteľom, ako je klasifikácia uzlov, regresia a predikcia prepojení. Architektúra modelu sa zvyčajne skladá z (1) enkodérov špecifických pre tabuľku, ktoré konvertujú riadky na počiatočné vektorové reprezentácie uzlov, (2) relačno-časových vrstiev GNN na odovzdávanie správ a (3) výstupnej vrstvy špecifickej pre danú úlohu, ktorá vytvára predpovede z konečných osadení uzlov. Celá táto architektúra sa trénuje *end-to-end*.

Nasledujúce časti sa zaoberajú našou vlastnou definíciou úlohy, prípravou tréningovej tabuľky, konštrukciou grafu relačných entít a postupom trénovania a vyhodnocovania modelu.

3.1.1 Príprava a spracovanie údajov pre RelBench

Táto časť opisuje praktický proces prípravy klinických údajov z databázy MIMIC-IV na použitie v rámci frameworku RelBench. Na rozdiel od tradičných prístupov, ktoré vyžadujú manuálne spájanie tabuliek a tvorbu príznakov, RelBench umožňuje modelovanie priamo nad relačným grafom vytvoreným na základe primárnych a cudzích kľúčov [3]. Tento prístup zabezpečuje, že vzťahy medzi entitami sú zachované, čím sa minimalizuje strata informácií.

V tejto časti detailne opisujeme jednotlivé kroky:

- výber a filtrovanie relevantných záznamov (pacientov a pobytov na JIS),
- napojenie súvisiacich tabuliek pomocou primárnych a cudzích kľúčov,
- čistenie údajov a odstránenie stĺpcov s potenciálnym únikom informácií,
- vytvorenie dátových rámcov (`DataFrame`) z jednotlivých tabuliek,
- definovanie časového stĺpca pre každú tabuľku, čo je nevyhnutné pre správne časové delenie a tvorbu relačného grafu,
- obalenie datasetu do triedy `Mimic_Dataset` kompatibilnej s RelBench,
- časové rozdelenie údajov na trénovaciu, validačnú a testovaciu množinu na základe distribúcie ICU časov,
- vytvorenie predikčnej úlohy

Týmto postupom sme vytvorili vstupné dáta pre experimenty s grafovým učením bez nutnosti ručného inžinierstva príznakov.

Výber pacienta a pobytu na JIS

Náš súbor údajov sme obmedzili na pacientov s jasne definovaným prvým prijatím na jednotku intenzívnej starostlivosti (JIS) tým, že sme doň zahrnuli len tých, ktorých záznamy spĺňali sadu klinických a časových obmedzení a obmedzení konzistentnosti údajov. Konkrétne sme pre každého pacienta vybrali prvý pobyt v nemocnici a prvý pobyt na jednotke intenzívnej starostlivosti (`hospstay_seq = 1` a `icustay_seq = 1`), čím sme zabezpečili porovnateľnosť medzi klinickými situáciami. Zároveň sme kvôli výpočtovým obmedzeniam obmedzili veľkosť trénovacej množiny na maximálne 20 000 pacientov.

Na ďalšie zlepšenie kvality údajov a zníženie variability sme použili viacero kritérií filtrovania. Vylúčili sme záznamy s chýbajúcimi alebo nulovými identifikátormi (`hadm_id`, `stay_id`) a dĺžku pobytu na JIS (`los_icu`). Zahrnutí boli len pacienti, ktorí mali v čase prijatia 15 rokov alebo viac, a obmedzili sme pobyty na JIS na tie, ktoré trvali od 36 do 240 hodín, aby sme vylúčili odľahlé hodnoty a nepravdepodobné záznamy.

Táto skupina pacientov bola získaná dotazom do tabuľky `icustay_detail`, ktorá spája demografické údaje a údaje špecifické pre JIS. Dotaz SQL je uvedený v algoritme 3.1. Vráti jednoznačnú množinu záznamov o pacientoch spolu s nasledujúcimi atribútmi: `subject_id` (jedinečné ID pacienta), `hadm_id` (ID prijatia do nemocnice), `stay_id` (ID pobytu na JIS), `gender` (pohlavie pacienta), zaokrúhlený `admission_age` (vek pacienta pri prijatí do nemocnice), a `race` (etnický pôvod).

Tieto vlastnosti sa použili ako základ pre konštrukciu grafu a na odvodenie atribútov uzlov v neskorších fázach experimentu.

Napojenie súvisiacich tabuliek

Po výbere pacientov a ich pobytov na JIS sme sa zamerali na napojenie súvisiacich tabuliek pomocou primárnych a cudzích kľúčov. Táto fáza je kľúčová pre zachovanie relačnej štruktúry dát, ktorú využíva framework RelBench na konštrukciu relačného grafu.

Pracovali sme s tabuľkami uvedenými v tabuľke 3.1, ktoré boli vybrané na základe ich významu pri modelovaní klinického priebehu pacienta. Tabuľku `chartevents` sme následne rozdelili a vytvorili z nej verziu `chartevents_numeric`, ktorá obsahuje iba záznamy s číselnými hodnotami. Tento krok bol nevyhnutný, keďže pôvodná tabuľka obsahovala heterogénne údaje rôzneho typu (textové aj číselné) v jednom stĺpci, čo by komplikovalo spracovanie dát.

Všetky tieto tabuľky obsahujú stĺpec `subject_id`, ktorý slúži ako základný identifikátor pacienta. Ako môžeme vidieť na obrázku 3.1, tabuľky `admissions` a `icustays` sú prepojené cez `hadm_id` a `stay_id`, čím umožňujú zreťazenie hospitalizačných a JIS

Algoritmus 3.1: SQL dotaz na výber prvej hospitalizácie a pobytu na JIS

```
SELECT DISTINCT ON (i.subject_id)
    i.subject_id,           -- Unique patient identifier
    i.hadm_id,              -- Unique hospital admission ID
    i.stay_id,              -- Unique ICU stay ID
    i.gender,               -- Patient gender
    ROUND(i.admission_age) AS age, -- Age of the patient
    i.race                  -- Patient race/ethnicity
FROM mimiciv_derived.icustay_detail i
WHERE i.hadm_id IS NOT NULL
    AND i.los_icu IS NOT NULL AND i.stay_id IS NOT NULL
    AND i.hospstay_seq = 1 AND i.icustay_seq = 1
    AND i.admission_age >= {min_age} AND i.los_icu >= {min_day}
    AND (i.outtime >= (i.intime + INTERVAL '{min_dur} hours'))
    AND (i.outtime <= (i.intime + INTERVAL '{max_dur} hours'))
```

pobytov. Tabuľka `chartevents_numeric` obsahuje podrobné klinické údaje viazané na konkrétny `stay_id`, zatiaľ čo tabuľka `procedureevents` zaznamenáva procedúry vykonané počas pobytu na JIS. Tabuľka `d_items` je referenčná tabuľka s metaúdajmi o položkách vyskytujúcich sa v udalostných tabuľkách, napojená pomocou `itemid`. Tieto tabuľky sú prepojené pomocou primárnych a cudzích kľúčov, ktoré definujú vzťahy medzi nimi.

Každú tabuľku sme načítavali pomocou dynamicky generovaného SQL dotazu, založeného na predchádzajúcom výbere pacientov a ich pobytov na JIS. Načítané boli iba riadky, ktoré zodpovedali týmto filtrovaným identifikátorom.

Keďže tabuľky `patients` a `icustays` už obsahovali len požadované záznamy, nebolo potrebné ich ďalej filtrovať, a preto boli z tohto procesu vynechané.

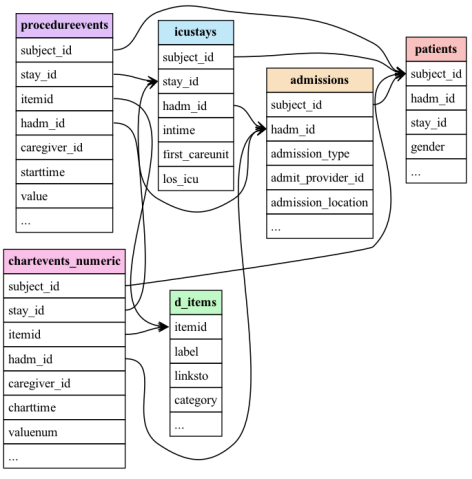
Na načítanie ostatných tabuliek sme využili SQL šablónu `sql_template`, uvedenú v Algoritme 3.2. Filtrovanie podľa stĺpcov `stay_id` a `hadm_id` bolo generované automaticky v závislosti od ich prítomnosti v konkrétnej tabuľke, čo sme zisťovali pomocou funkcie `get_columns_names`, ktorá vracia zoznam názvov stĺpcov.

Na základe vygenerovaného SQL dotazu bol pre každú tabuľku vytvorený nový `DataFrame` pomocou funkcie `querier.query`, ktorá dotaz vykoná nad databázou vrátane zadaných parametrov `template_vars`.

Všetky načítané tabuľky sme následne uložili do slovníka `tables_df`, kde kľúčom je názov tabuľky a hodnotou zodpovedajúci `DataFrame`. V ďalšom kroku sme definovali prepojenia medzi tabuľkami pomocou primárnych a cudzích kľúčov, ako bolo popísané v predchádzajúcej časti. Tento proces zabezpečuje, že všetky relevantné tabuľky sú

Tabuľka 3.1: Prehľad použitých tabuliek z MIMIC-IV

Tabuľka	**Popis**
`patients`	Demografické údaje pacientov, ako je pohlavie a vek.
`admissions`	Informácie o hospitalizáciách – typ prijatia, miesto prijatia, dátumy.
`icustays`	Záznamy o pobyte na JIS vrátane času prijatia a trvania.
`procedureevents`	Záznamy o liečebných a diagnostických výkonoch počas JIS.
`chartevents_numeric`	Číselné klinické merania, ako napr. srdcová frekvencia, tlak atď.
`d_items`	Metaúdaje o položkách vyskytujúcich sa v ostatných tabuľkách (napr. `itemid`, `label` pre konkrétnu procedúru).



Obr. 3.1: Vizualizácia relačných prepojení medzi použitými tabuľkami z databázy MIMIC-IV, ktoré slúžia ako základ pre konštrukciu relačného grafu. Každá tabuľka je reprezentovaná ako entita so svojimi kľúčovými atribútmi.

Algoritmus 3.2: Prepojenie tabuliek podla prímarnych a cudzich kľučov

```
sql_template =
    SELECT *
    FROM {schema}.{table_name}
    WHERE subject_id IN {subject_ids}

for schema, tables in tables_with_schema_key.items():
    for (table_name,) in tables:
        if table_name in {'patients', 'icustays'}:
            continue # skip tables that are already filtered
        columns = get_columns_names(
            table_name=table_name, schema=schema
            )
        if 'stay_id' in columns:
            query_string += ' AND stay_id IN {stay_ids}'
        if 'hadm_id' in columns:
            query_string += ' AND hadm_id IN {hadm_ids}'
        query_string += ";"

        template_vars = build_template_vars(table_name, schema)

        tables_df[table_name] = querier.query(
            query_string=sql_template,
            extra_template_vars=template_vars
            )
```

správne prepojené a pripravené na ďalšie spracovanie a analýzu.

Čistenie a predspracovanie údajov

Aby sme znížili dimenzionalitu a zabránili úniku údajov, systematicky sme vynechávali stĺpce, ktoré buď neboli užitočné na modelovanie, alebo obsahovali informácie o výsledkoch po ukončení. Príkladom sú časové pečiatky prepustenia alebo úmrtia, príznaky výsledkov nemocnice a nadbytočné metaúdaje. Stĺpce boli vyradené pomocou vopred definovaného slovníka pre každú tabuľku, ako je uvedené v Algoritme 3.3. Napríklad v tabuľke `admissions` sme vylúčili atribúty, ako `admittime`(čas prijatia), `dischtime`(čas prepustenia), `deathtime`(čas úmrtia), `hospital_expire_flag`(indikátor úmrtia v nemocnici), ako aj `race`(etnický pôvod), `discharge_location`(miesto prepustenia) a ča-

Algoritmus 3.3: Zoznam stĺpcov odstránených z jednotlivých tabuliek pre účely čistenia a prevencie úniku informácií

```
{
    "admissions":          ["admittime", "dischtime","deathtime",
                            "edouttime","edregtime",
                            "hospital_expire_flag",
                            "discharge_location", "race" ],
    "chartevents":         ["storetime"],
    "procedureevents" :    ["storetime", "endtime","location",
                             "location_category", "location",
                             "locationcategory", "linkorderid" ,
                             "originalamount", "originalrate" ],
    "d_items" :            ["abbreviation"]
}
```

sové pečiatky pohotovostných oddelení.

Tento krok orezávania je kľúčový, aby sa pri trénovaní nepoužívali budúce informácie a aby sa zabezpečilo, že proces konštrukcie grafu bude brať do úvahy len časovo platné hrany.

Zabalenie datasetu a časové rozdelenie

Po filtrovaní a vyčistení údajov boli spracované tabuľky zabalené do vlastnej podtriedy `relbench.base.Dataset` s názvom `Mimic_Dataset`. Tento krok je nevyhnutný na zabezpečenie kompatibility s frameworkom RelBench, ktorý očakáva špecifickú štruktúru a metódy pre spracovanie relačných grafov.

Ako je znázornené v ukažke algoritmu 3.4, pre každú tabuľku v slovníku `tables_df` (vytvorenom podľa algoritmu 3.2) sme vytvorili objekt triedy `Table` z knižnice RelBench. Tento objekt obsahuje samotné dáta (`df`), primárny kľúč (`pkey_col`), cudzí kľúč odkazujúci na inú tabuľku (`fkey_col_to_pkey_table`) a časový stĺpec (`time_col`), ktorý je kľúčový pre časové rozdelenie a konštrukciu grafu.

Na určenie časových hraníc pre trénovaciu, validačnú a testovaciu množinu sme zoradili ICU pobyty podľa ich začiatku (`intime`) a rozdelili ich v pomere 70,% na trénovanie, 15,% na validáciu a 15,% na testovanie. Tento spôsob delenia zabezpečuje, že model sa učí na historických údajoch a vyhodnocuje sa na novších prípadoch, čím sa predchádza úniku informácií do trénovacej fázy.

Algoritmus 3.4: Vytvorenie objektov `Table` pre každú tabuľku s definovanými kľúčmi a časovým stĺpcom

```
    tables[table_name] = Table(
            df= tables_df[table_name].df,
            fkey_col_to_pkey_table=fkey_col_to_pkey_table,
            pkey_col=pkey_col,
            time_col=time_col
        )
```

3.1.2 Vytvorenie klasifikačnej úlohy

V tomto projekte sme definovali klasifikačnú úlohu `ICULengthOfStayTask`, ktorá predstavuje binárnu klasifikáciu zameranú na predikciu dĺžky pobytu na jednotke intenzívnej starostlivosti (JIS). Cieľom je určiť, či prvý pobyt pacienta na JIS bude trvať tri a viac dní. Úloha je implementovaná ako podtrieda abstraktnej triedy `EntityTask` z knižnice RelBench a je definovaná typom `TaskType.BINARY_CLASSIFICATION`. Využíva stĺpce identifikujúce entitu a čas hospitalizácie (`subject_id`, `intime`) a ako cieľový atribút používa `los_icu_binary`, ktorý nadobúda hodnotu 1, ak dĺžka pobytu na JIS dosahuje aspoň 3 dni, inak má hodnotu 0.

Časová granularita úlohy je nastavená na jeden deň (`timedelta`). Model je trénovaný na základe údajov dostupných počas prvých 24 hodín od prijatia pacienta na JIS, pričom cieľom je predikovať, či dĺžka pobytu presiahne tri dni. Časové hodnoty používané pri výpočtoch sú preberané priamo zo vstupného datasetu a rozdelenie na trénovaciu, validačnú a testovaciu množinu je odvodené na základe časových pečiatok reálnych hospitalizácií.

Základná metóda `make_table`, uvedená v algoritme 3.5, predstavuje povinnú súčasť definície každej úlohy odvodené od triedy `EntityTask`. Táto metóda vykonáva dotaz nad vstupnými tabuľkami databázy a spája ICU pobyty s pacientmi a časovými pečiatkami. Výsledkom je tabuľka obsahujúca trojicu: `timestamp`, `subject_id` a `los_icu_binary`, ktorá slúži ako vstup pre klasifikačný model.

Na hodnotenie výkonu modelu sú použité štandardné metriky pre binárnu klasifikáciu: priemerná presnosť (average precision), presnosť (accuracy), F1 skóre a ROC AUC.

Takto definovaná úloha vytvára základ pre konštrukciu grafu a následné trénovanie modelu nad relačnými údajmi.

Algoritmus 3.5: Definícia úlohy predikcie predĺženého pobytu na JIS

```
def make_table(Database, timestamps) -> Table:
    df = duckdb.sql(
        SELECT
            t.timestamp as intime,
            p.subject_id,
            CASE
                WHEN i.los_icu >= 3 THEN 1
                ELSE 0
            END as los_icu_binary
        FROM
            timestamp_df t
        LEFT JOIN icu i
            ON i.intime > t.timestamp
            AND i.intime <= t.timestamp +
            INTERVAL '{self.timedelta}'
        LEFT JOIN patients p
        ON
            i.subject_id = p.subject_id
    ).df()
```

3.1.3 Konštrukcia grafu

Po definovaní klasifikačnej úlohy sme vytvorili relačný graf, ktorý slúži ako vstup pre grafovú neurónovú sieť. Táto etapa je kľúčová pre zabezpečenie toho, aby model vedel efektívne využiť informácie rozptýlené naprieč viacerými tabuľkami databázy.

Na zakódovanie textových stĺpcov do číselnej podoby sme použili embedder založený na predtrénovanom modeli `MiniLM`. Tento model umožňuje previesť textové hodnoty do vektorov, ktoré môžu byť následne spracované neurónovou sieťou. Embedder sme implementovali ako triedu `GloveTextEmbedding`, ktorá slúži ako obal nad modelom `SentenceTransformer`. V rámci konfigurácie sme ho vložili do objektu `TextEmbedderConfig`, ktorý je súčasťou štandardného nastavenia v RelBench.

Takto definovaný embedder bol použitý pri generovaní relačného grafu pomocou funkcie `make_pkey_fkey_graph`, ktorá automaticky vytvorila relačný graf z databázových tabuliek na základe primárnych a cudzích kľúčov. Vstupom do tejto funkcie je aj špecifikácia typov stĺpcov (`col_to_stype_dict`), ktorá pomáha správne identifikovať textové a číselné stĺpce.

Výsledkom tohto procesu je objekt `data`, ktorý reprezentuje konečný relačný graf pripravený na trénovanie modelu. Zároveň je vygenerovaný aj slovník `col_stats_dict` obsahujúci štatistiky o jednotlivých stĺpcoch. Príslušná implementácia je uvedená v algoritme 3.6.

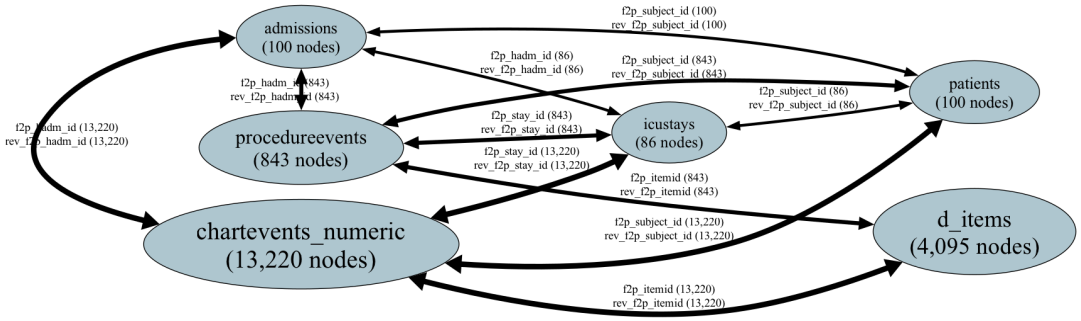
Celkovú štruktúru výsledného relačného grafu po aplikovaní embedingu znázorňuje obrázok 3.2. Tento graf bol vytvorený pomocou funkcie `make_pkey_fkey_graph`, ako je opísané v algoritme 3.6, pričom sme využili vlastnú implementáciu textového embeddera `GloveTextEmbedding` založeného na modeli `MiniLM`. Vizualizácia bola vytvorená pre podmnožinu 100 pacientov a slúži na ilustráciu prepojení medzi tabuľkami. Uzly v grafe reprezentujú jednotlivé tabuľky a hrany znázorňujú vzťahy medzi entitami definované prostredníctvom primárnych a cudzích kľúčov. Každá hrana je označená typom prechodu a počtom výskytov v zátvorke. Z grafu je zrejmé, že najviac prepojení vzniká medzi tabuľkami `chartevents_numeric`, `procedureevents` a `patients`, čo poukazuje na komplexnosť a bohatosť klinických údajov v databáze MIMIC-IV.

3.1.4 Architektúra modelu

Pri výbere architektúry modelu sme sa rozhodli využiť štandardnú implementáciu poskytnutú v knižnici RelBench, ktorá je navrhnutá špeciálne pre prácu s relačnými grafmi. Model kombinuje viacero komponentov, ktoré spolu vytvárajú plne funkčnú architektúru pre predikciu na relačných dátach.

Model pozostáva z nasledujúcich častí:

- **HeteroEncoder** – modul, ktorý vytvára počiatočné reprezentácie uzlov na zá-



Obr. 3.2: Vizualizácia výsledného relačného grafu vytvoreného pomocou funkcie `make_pkey_fkey_graph`. Uzly reprezentujú tabuľky z databázy MIMIC-IV a hrany znázorňujú vzťahy medzi entitami na základe primárnych a cudzích kľúčov. Počet výskytov jednotlivých typov prechodov je uvedený v zátvorkách. Graf bol vytvorený na podmnožine 100 pacientov s využitím embeddera založeného na modeli `MiniLM`.

Algoritmus 3.6: Konštrukcia relačného grafu pomocou GNN a textového embeddera

```
class GloveTextEmbedding:
    def __init__(self, device: Optional[torch.device] = None):
        self.model = SentenceTransformer(
            "sentence-transformers/all-MiniLM-L6-v2",
            device=self.device,
        )
text_embedder_cfg = TextEmbedderConfig(
    text_embedder=GloveTextEmbedding(device=device),
)

data, col_stats_dict = make_pkey_fkey_graph(
    mimic_db,
    col_to_stype_dict=col_to_stype_dict,
    text_embedder_cfg=text_embedder_cfg,
)
```

Algoritmus 3.7: Inicializácia modelu a optimalizátora

```
model = Model(
    data=data,
    col_stats_dict=col_stats_dict,
    num_layers=3,
    channels=128,
    out_channels=1,
    aggr="sum",
    norm="batch_norm",
).to(device)

optimizer = torch.optim.Adam(
    model.parameters(),
    lr=0.00005,
    weight_decay=1e-4
)
```

klade tabuľkových atribútov,

- **HeteroTemporalEncoder** – pridáva informácie o relatívnom čase medzi uzlami, čo umožňuje modelu zachytiť časovú dynamiku údajov,
- **HeteroGraphSAGE** – hlavná GNN architektúra, ktorá šíri informácie naprieč uzlami a hranami relačného grafu,
- **MLP hlava (head)** – výstupná vrstva, ktorá na základe konečných reprezentácií uzlov určí finálnu predikciu.

Model podporuje aj voliteľné komponenty ako „shallow embeddings“ (priame embeddingy pre vybrané typy uzlov) a ID-awareness, ktorá umožňuje explicitne zvýrazniť identitu uzla pri predikcii.

Model sme vytvorili s tromi GNN vrstvami (`num_layers=3`) a šírkou skrytých vrstiev 128 (`channels=128`). Ako agregačná funkcia bola zvolená suma (`aggr="sum"`), a na stabilizáciu trénovania bola použitá normalizácia `batch_norm`.

Optimalizácia modelu prebiehala pomocou algoritmu `Adam` s počiatočnou hodnotou učenia `0.00005` a penalizáciou zložitosti modelu (`weight_decay=1e-4`). Počet trénovacích epoch bol nastavený na 50. Celý postup inicializácie modelu a optimalizátora je uvedený v algoritme 3.7. Architektúra je trénovaná *end-to-end*, pričom parametre sa optimalizujú spoločne v rámci celej siete.

3.1.5 Tréning a vyhodnocovanie

Tréning a vyhodnocovanie modelu prebiehalo štandardným spôsobom vo viacerých epochách. V každej epoche sa najprv vypočítala trénovacia strata, potom sa model vyhodnotil na validačnej množine a výsledné metriky sa zaznamenali.

Hodnotenie sa realizovalo pomocou funkcie `evaluate`, ktorá porovnáva predikované hodnoty s cieľovými značkami a vracia viaceré klasifikačné metriky, ktoré boli definované v sekcii 3.1.2, ako napríklad `accuracy`, `F1 score`, `average precision` a `ROC AUC`.

Počas tréningu sme sledovali validačné skóre podľa zvolenej tunovacej metriky, ktorá bola explicitne nastavená ako `tune_metric = average_precision`. Táto voľba znamená, že ako hlavné kritérium pre výber najlepšieho modelu počas tréningu bola zvolená metrika `average_precision` (priemerná presnosť), ktorá hodnotí presnosť modelu pri rôznych prahoch klasifikácie. Táto metrika je obzvlášť vhodná pre nerovnovážne dátové sady a poskytuje robustnejšie hodnotenie schopnosti modelu oddeľovať pozitívne a negatívne triedy.

Počas tréningu sme po každej epoche porovnali hodnotu tunovacej metriky na validačnej množine s doteraz najlepším dosiahnutým výsledkom. Ak bola nová hodnota lepšia, aktuálne parametre modelu sa skopírovali do premennej `state_dict`, čím sa uchoval najvýkonnejší model na základe danej metriky.

Po dokončení všetkých epoch sme načítali stav modelu, ktorý dosiahol najlepší výsledok na validačnej množine podľa metriky `average_precision`. Táto metrika, určená ako tunovacia, hodnotí schopnosť modelu správne identifikovať pozitívne triedy pri rôznych rozhodovacích prahoch.

Akonáhle sa zaznamenalo najlepšie validačné skóre, model sa uložil a následne načítal pomocou objektu `state_dict`. Na konci tréningu prebehlo finálne vyhodnotenie výkonnosti modelu na validačnej aj testovacej množine. Predikcie boli uložené do súborov pre prípadnú ďalšiu analýzu.

3.1.6 Praktické problémy a riešenia

Počas implementácie experimentálneho rámca sme narazili na viacero praktických výziev, ktoré si vyžadovali úpravu pôvodného plánu alebo dodatočné ladenie.

Na začiatku sme experimentovali s úlohou predikcie úmrtnosti počas hospitalizácie. Táto úloha sa však ukázala ako nevhodná vzhľadom na výrazne nevyvážené rozdelenie tried — úmrtí bolo v dátach relatívne málo, čo viedlo k slabým výsledkom a nízkej stabilite modelu. Následne sme sa pokúsili formulovať úlohu ako regresiu dĺžky pobytu v nemocnici, no aj táto stratégia narážala na problémy, najmä pokiaľ ide o interpretáciu výstupov a hodnotenie výkonnosti modelu. Nakoniec sme zvolili binárnu klasifikáciu predikcie dĺžky pobytu na JIS dlhšieho ako tri dni, ktorá poskytla stabilnejšie výsledky

a jasné hodnotiace kritériá.

Ďalšou výzvou bol výber vhodných tabuliek z databázy MIMIC-IV. Pri modelovaní sme sa rozhodli vynechať určité tabuľky (napr. `labevents`), ktoré boli príliš rozsiahle alebo redundantné, a naopak sme upravili tabuľku `chartevents` tak, aby obsahovala len číselné hodnoty. Týmto spôsobom sme vytvorili verziu `chartevents_numeric`, čím sme sa vyhli problémom s rôznymi typmi údajov v jednom stĺpci (napr. kombinácia textu a čísiel).

Z technického hľadiska bolo dôležité správne navrhnúť časové rozdelenie trénovacej, validačnej a testovacej množiny, aby nedošlo k úniku informácií. Pracovali sme výhradne s historickými údajmi a overovali, že model je testovaný výlučne na neskorších prípadoch ako tie, na ktorých sa učil.

Pôvodne sme plánovali použiť model `average_word_embeddings_glove.6B.300d`, ktorý je menej náročný na výpočtové zdroje, no neposkytoval dostatočne kvalitné reprezentácie pre naše potreby. Preto sme prešli na model `MiniLM` zo série `sentence-transformers`, ktorý sa ukázal ako vyvážené riešenie s dobrým pomerom medzi kvalitou výstupov a výpočtovou náročnosťou. Model bol bez problémov spustiteľný v dostupných podmienkach a poskytol kvalitné výsledky pri enkódovaní textových atribútov.

Tieto praktické skúsenosti významne prispeli k stabilite a reprodukovateľnosti celého experimentálneho rámca.

Kapitola 4

Výsledky, porovnanie a diskusia

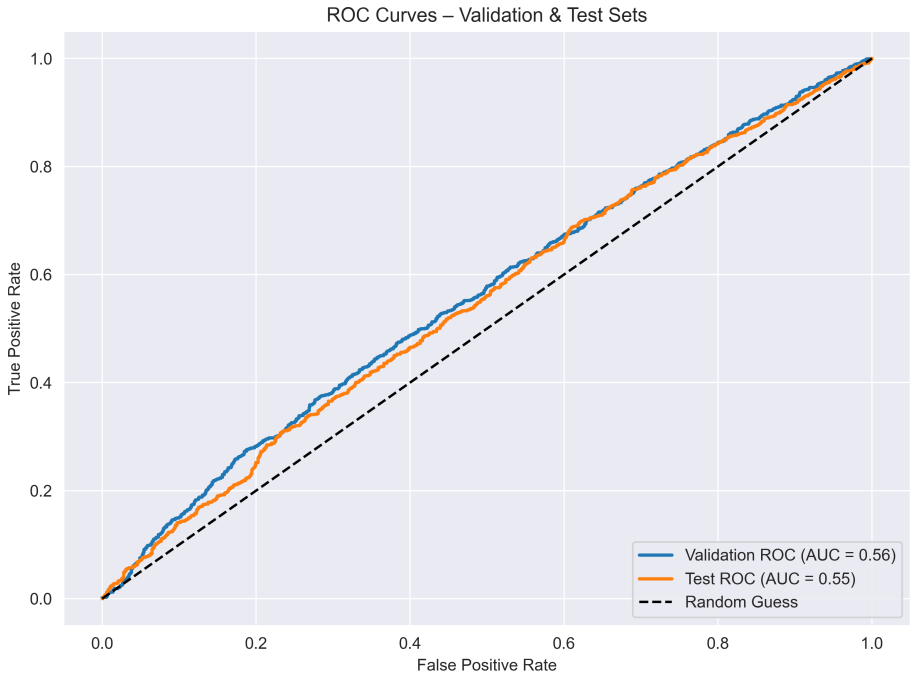
V tejto kapitole prezentujeme výsledky experimentu zameraného na predikciu dĺžky pobytu pacientov na jednotke intenzívnej starostlivosti (JIS) pomocou modelu založeného na frameworku RelBench. Model bol trénovaný nad klinickými údajmi z databázy MIMIC-IV, kde sme použili binárnu klasifikáciu – cieľom bolo predpovedať, či pobyt pacienta na JIS bude trvať aspoň 3 dni.

Hoci sme využili grafovú architektúru schopnú spracovávať komplexné relačné štruktúry, dosiahnuté metriky naznačujú len mierne nadúrovňovú výkonnosť, porovnateľnú s náhodným klasifikátorom. Model dosiahol maximálne skóre ROC AUC na testovacej množine na úrovni 0,55, čo svedčí o nízkej schopnosti diskriminácie medzi triedami. Nízke hodnoty F1 skóre a priemerná absolútna chyba (MAE – Mean Absolute Error) ukazujú na značnú neurčitosť a slabú presnosť predikcií. Jedným z dôvodov môže byť vysoký šum v údajoch, keďže klasifikačné triedy boli vyvážené.

Nasledujúce sekcie rozoberajú výsledky modelu, ich vizualizáciu a porovnanie s inými existujúcimi prístupmi.

4.1 Vyhodnotenie a vizualizácia výsledkov

Tabuľka 4.1 sumarizuje výkonnosť modelu RelBench-GNN na trénovacej, validačnej a testovacej množine. Model dosiahol najvyššiu presnosť na validačnej množine (55,8 %) a najvyššiu hodnotu metriky priemernej presnosti (Average Precision, AP) na testovacej množine (0,506). Táto metrika AP hodnotí presnosť modelu v závislosti od úplnosti – teda ako dobre vie model zachytiť pozitívne prípady naprieč rôznymi prahmi rozhodovania. Ďalšou kľúčovou metrikou je plocha pod ROC krivkou (Receiver Operating Characteristic Area Under Curve, ROC AUC), ktorá vyjadruje schopnosť modelu rozlišovať medzi pozitívnymi a negatívnymi prípadmi bez ohľadu na konkrétny prah. Hodnota ROC AUC sa pohybovala od 0,501 (trénovacia množina) po 0,559 (validačná množina), čo naznačuje slabú výkonnosť modelu, len mierne nad úrovňou náhodného



Obr. 4.1: Porovnanie ROC kriviek modelu na validačnej a testovacej množine.

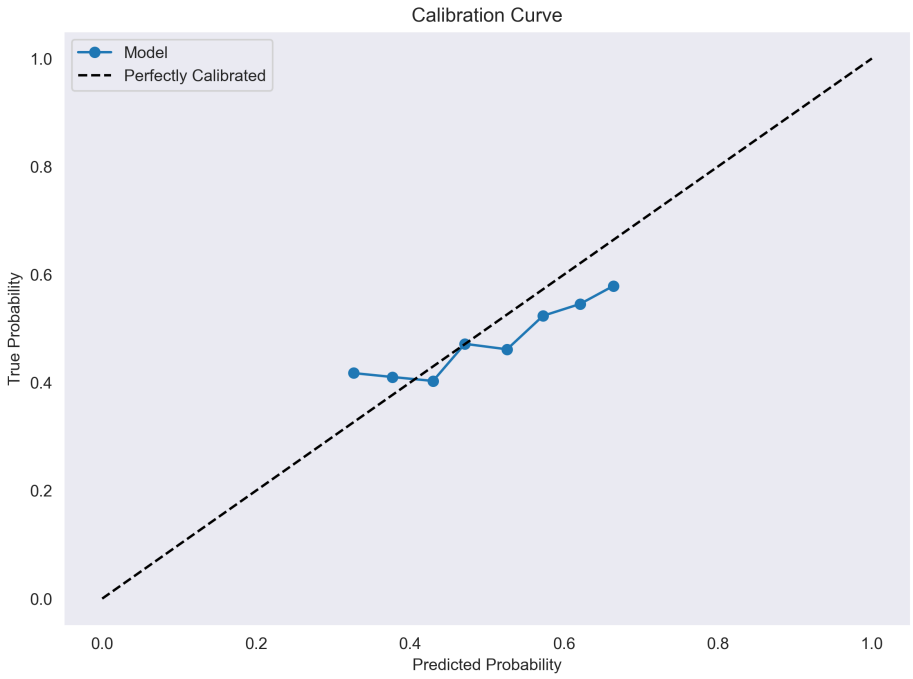
klasifikátora.

Tabuľka 4.1: Výkonnosť modelu RelBench-GNN na jednotlivých množinách

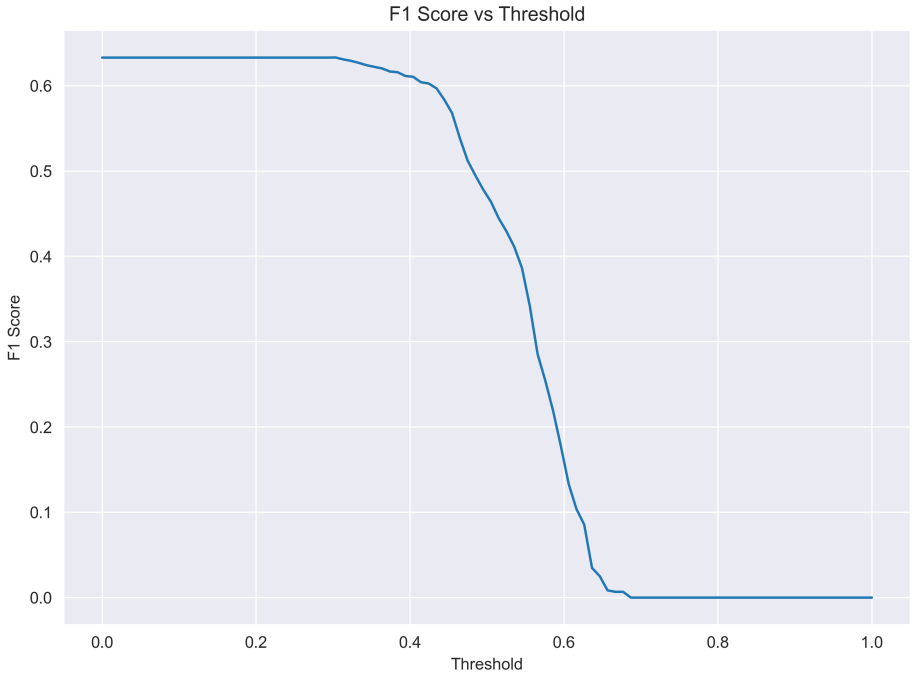
Množina	**Presnosť (Accuracy)**	**F1 skóre**	**AP**	**ROC AUC**
Tréningová	0,532	0,098	0,469	0,501
Validačná	0,558	0,111	0,498	0,559
Testovacia	0,546	0,100	0,506	0,548

Obrázok 4.1 zobrazuje ROC krivku na validačnej a testovacej množine. Hodnota AUC približne 0,55 naznačuje len mierne lepšie výsledky než náhodný klasifikátor a poukazuje na slabú schopnosť modelu rozlišovať medzi pozitívnymi a negatívnymi prípadmi. Kalibračná krivka na obrázku 4.2 ilustruje odhadované pravdepodobnosti – väčšina bodov sa nachádza pod ideálnou diagonálou, čo poukazuje na systematické podceňovanie výskytu pozitívnej triedy. Závislosť F1 skóre od rozhodovacieho prahu je znázornená na obrázku 4.3; najvyššia hodnota sa dosahuje pri veľmi nízkom prahu ($\sim$0,05), čo svedčí o slabej rozhodovacej schopnosti modelu.

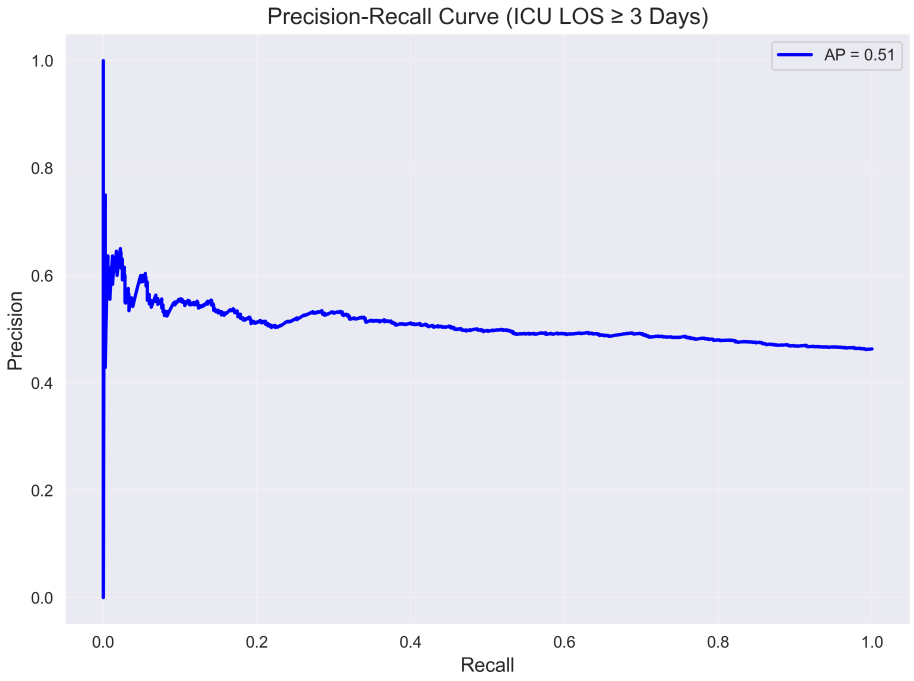
Precision-Recall krivka na obrázku 4.4 poskytuje detailnejší pohľad na výkonnosť pri detekcii pozitívnych prípadov. Hodnota AP (0,51) je len mierne vyššia než náhodný výber, čo opäť svedčí o slabom výkone. Na obrázku 4.5 vidíme maticu zámen pri rozhodovacom prahu 0,5. Model častejšie klasifikuje pacientov do kategórie kratšieho pobytu, pričom až 650 prípadov s LOS $\geq$ 3 dní bolo nesprávne zaradených do negatívnej



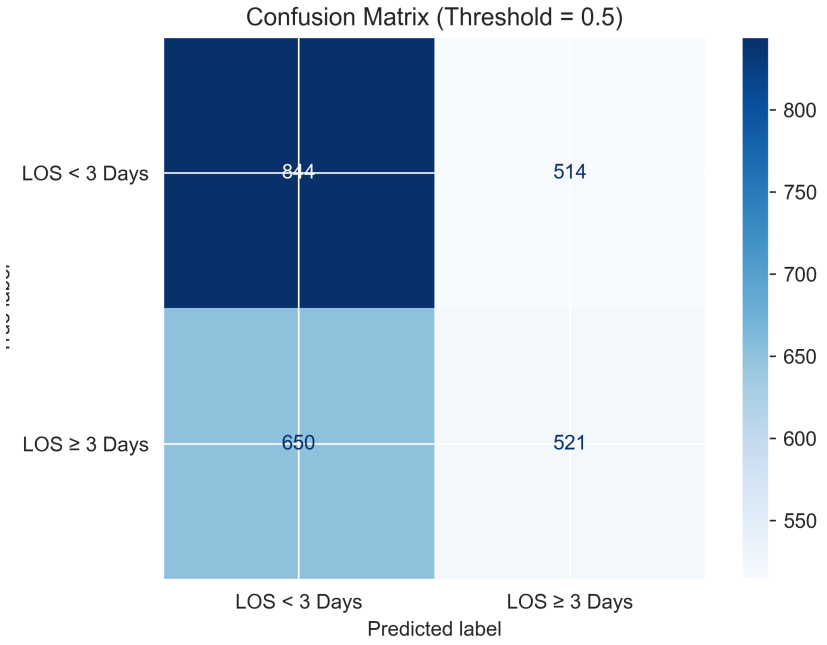
Obr. 4.2: Porovnanie predikovaných a skutočných pravdepodobností.



Obr. 4.3: Vývoj F1 skóre pri rôznych prahoch.



Obr. 4.4: Presnosť a citlivosť modelu pre LOS $\geq$ 3 dní.



Obr. 4.5: Matica zámen pre rozhodovací prah 0,5.

triedy.

4.2 Porovnanie s inými prístupmi

Tabuľka 4.2: Porovnanie výkonnosti RelBench-GNN a Graformeru pri predikcii LOS

Model	**Dataset**	**AP**	**ROC AUC**
RelBench-GNN	MIMIC-IV	0,506	0,548
Graformer [8]	MIMIC-III/MIMIC-IV	0,714	0,777

Tabuľka 4.2 porovnáva výkonnosť modelu RelBench-GNN s Graformerom, pokročilým modelom založeným na transformerovej architektúre, ktorý využíva maskované predtrénovanie nad populačným grafom pacientov. Hoci pôvodné výsledky Graformeru boli reportované na staršej verzii databázy MIMIC-III, v rámci tímu Alistiq sme implementáciu [10] prispôsobili pre databázu MIMIC-IV a otestovali prístup použitý v Graformeri. Výsledky sa ukázali byť porovnateľné s pôvodne publikovanými, čo potvrdzuje relevantnosť tohto porovnania aj pre MIMIC-IV. Graformer dosahuje výrazne lepšie hodnoty metriky AP aj ROC AUC, čo poukazuje na výhody komplexnejších architektúr a rozsiahlejšieho predtrénovania. Tento rozdiel zároveň odráža obmedzenia nášho prístupu, ktorý používa jednoduchšie embeddingy a spracováva relačné dáta bez dodatočného inžinierstva vstupných premenných.

4.3 Zhodnotenie výskumnej otázky

Ako bolo formulované v sekcii 2.1, hlavnou otázkou tejto práce bolo, či grafové neurónové siete (GNN) trénované na reprezentáciách databázy MIMIC-IV dokážu presnejšie predpovedať klinické výsledky pacientov v porovnaní s tradičnými prístupmi. Výsledky nášho experimentu naznačujú, že priame trénovanie GNN na úplnej relačnej štruktúre MIMIC-IV bez predspracovania a výberu príznakov zatiaľ nevedie k uspokojivej výkonnosti. Hoci model využíva bohatú sieť vzťahov medzi entitami, vysoká miera zašumenia a rozsah dát pravdepodobne znižujú jeho efektivitu. Odpoveď na výskumnú otázku je preto v tomto kontexte skôr záporná: bez dôslednej prípravy vstupov a doménovo špecifického výberu informácií sa GNN modely na tak komplexných dátach ukazujú ako neefektívne.

Okrem hlavného cieľa sme sa zaoberali aj doplňujúcou výskumnou otázkou, či je možné efektívne využiť grafové neurónové siete priamo nad relačnou štruktúrou databázy MIMIC-IV. Výsledky ukazujú, že bez dôkladnej definície entít, typov vzťahov a

výberu relevantných čŕt je samotná relačná reprezentácia nedostatočná pre trénovanie výkonného GNN modelu. To znamená, že grafová štruktúra odvodená priamo zo schémy databázy síce poskytuje formálny základ, ale bez doménového obohatenia a návrhu zohľadňujúceho klinický kontext nedokáže zachytiť podstatné vzťahy pre predikciu. Táto práca tak poukazuje na potrebu starostlivého návrhu grafu, čo predstavuje kľúčový krok pri uplatňovaní relačného učenia v zdravotníctve.

4.4 Diskusia

Zistenia ukazujú, že napriek využitiu grafového modelu RelBench-GNN nedošlo k výraznému zlepšeniu v predikcii dĺžky pobytu pacientov na JIS. Výkonnosť modelu, vyjadrená najmä metrikami ROC AUC a F1 skóre, sa drží len tesne nad úrovňou náhodného klasifikátora. To naznačuje, že model má nízku schopnosť diskriminovať medzi pacientmi s kratším a dlhším pobytom, pričom veľká časť pozitívnych prípadov ostáva neodhalená.

Jedným z hlavných problémov je vysoká miera šumu v klinických údajoch. Hoci triedy boli vyvážené, dáta obsahovali rôznorodé a často redundantné záznamy, čo mohlo negatívne ovplyvniť schopnosť modelu generalizovať. Navyše, použitie predtrénovaného embeddera `MiniLM` síce zjednodušilo spracovanie vstupov, no bez doménovej optimalizácie nebolo dostatočne informatívne pre klinickú predikciu.

Ďalším obmedzením bola absencia inžinierstva príznakov a predspracovania, ktoré sú bežnou súčasťou dobrej praxe pri práci s databázou MIMIC, napríklad v rámci frameworku `mimic-extract` [10]. Tento nástroj okrem iného agreguje merané veličiny z `chartevents` do hodinových blokov, vykonáva dopĺňanie chýbajúcich hodnôt a zabezpečuje normalizáciu dát, čím výrazne zvyšuje kvalitu vstupov pre modelovanie. Model spracovával celé relačné schémy bez filtrácie či normalizácie, čo síce podporuje automatizáciu, ale zároveň vedie k zahlteniu irelevantnými dátami. V porovnaní s modelmi ako Graformer, ktoré využívajú maskované pretrénovanie a transformerové architektúry, je tento prístup výrazne menej sofistikovaný.

Pre budúci výskum a zlepšenie výkonnosti modelu odporúčame:

- obohatiť graf o ďalšie medicínsky významné tabuľky,
- zvážiť predspracovanie dát na zníženie šumu a redundancie,
- prípadne kombinovať relačný prístup s grafmi podobnosti medzi pacientmi,
- testovať výkon modelu na väčšej a čistejšej podmnožine dát,
- a preskúmať robustnejšie GNN architektúry vrátane heterogénnych alebo transformerových modelov.

Záver

Cieľom tejto práce bolo preskúmať, do akej miery dokážu grafové neurónové siete (GNN) efektívne pracovať s relačnými databázami v kontexte klinických údajov, konkrétne pri komplexnej predikcii dĺžky pobytu pacientov na jednotke intenzívnej starostlivosti (JIS) pomocou databázy MIMIC-IV. Práca nadväzuje na súčasný výskum relačného hlbokého učenia a čerpá z benchmarkového rámca RelBench, ktorý umožňuje trénovanie GNN modelov priamo nad relačnými štruktúrami bez potreby ich transformácie do jednej tabuľky.

V úvodných kapitolách sme predstavili teoretické východiská grafového modelovania, princípy fungovania GNN a špecifiká práce so závislostne orientovanými údajmi v zdravotníctve. Detailne sme opísali štruktúru databázy MIMIC-IV a odôvodnili potrebu zachovania jej relačnej formy pri spracovaní pomocou GNN. Na základe preštudovanej literatúry, ako aj inšpirácie z modelov Graformer a PGExplainer, sme formulovali výskumné otázky zamerané na presnosť a praktickú realizovateľnosť takéhoto prístupu.

Navrhli sme binárnu klasifikačnú úlohu predikcie predĺženého pobytu na JIS (LOS ≥ 3 dní), pripravili sme klinické dáta pomocou frameworku RelBench a implementovali GNN architektúru schopnú spracovávať komplexný relačný graf odvodený zo štruktúry databázy. Model sme trénovali na výbere z MIMIC-IV a jeho výkon sme vyhodnotili pomocou metrík ako F1 skóre, average precision (AP) a ROC AUC.

Výsledky ukázali, že model dosahuje len mierne lepší výkon ako náhodná klasifikácia – najlepšie skóre AP na testovacej množine bolo 0,506 a ROC AUC 0,548. Napriek využitiu pokročilého grafového spracovania nedošlo k výraznému zlepšeniu oproti náhodnej klasifikácii. Získané výsledky naznačujú, že priamy prístup bez inžinierstva čŕt a doménovej optimalizácie je v prípade MIMIC-IV nedostatočný. Na rozdiel od modelu Graformer, ktorý dosahuje AP nad 0,7, náš model narážal na obmedzenia spôsobené šumom, redundanciou údajov a nevhodnou inicializáciou embeddingov.

Z týchto pozorovaní vyplýva, že úspešné nasadenie GNN na relačné databázy v zdravotníctve si vyžaduje:

- premyslený výber a inžinierstvo vstupných príznakov,
- využitie doménovo špecifických embedderov (napr. ClinicalBERT),

- a možno aj hybridné prístupy, ktoré kombinujú relačný graf s grafmi podobnosti medzi pacientmi.

Práca ukazuje potenciál frameworku RelBench pri vývoji reprodukovateľných experimentov na relačných štruktúrach a zároveň identifikuje jeho obmedzenia. Hoci náš model neprekonal tradičné prístupy, skúsenosti z jeho implementácie poskytujú dôležité poznatky pre budúci vývoj relačne orientovaných GNN systémov.

Do budúcna vidíme priestor najmä v kombinácii predtrénovaných embedderov s dôkladne navrhnutou grafovou štruktúrou založenou na klinickej expertíze. Tiež by bolo vhodné porovnať výkon RelBench-GNN s jednoduchšími, ale doménovo vyladenými modelmi (napr. logistická regresia s inžinierstvom príznakov), aby sme lepšie pochopili, v ktorých prípadoch má grafové učenie reálny prínos.

Zhrnutie hlavných prínosov práce

Táto práca priniesla viacero výskumných aj osobných prínosov:

- Implementovali sme kompletný experimentálny pipeline pre trénovanie grafových neurónových sietí (GNN) nad relačnou verziou databázy MIMIC-IV s využitím frameworku RelBench.
- Overili sme, že automatizovaný prístup bez inžinierstva vstupných premenných naráža na limity pri zdravotníckych údajoch – najmä kvôli ich zašumenosti a redundantnosti.
- Identifikovali sme konkrétne slabé miesta prístupu RelBench-GNN a poskytli návrhy na jeho zlepšenie v kontexte klinických predikčných úloh.
- Ukázali sme, že pre efektívne využitie GNN nad zdravotníckymi dátami je potrebná kombinácia doménového porozumenia a technického návrhu štruktúry grafu.
- Získali sme prehľad o možnostiach a obmedzeniach trénovania GNN nad komplexnými relačnými databázami.
- Z praktického hľadiska som nadobudol skúsenosti s implementáciou GNN modelov, prácou s relačnými dátami, spracovaním klinických databáz a vyhodnocovaním prediktívnych modelov.
- Výsledky tejto práce môžu byť prínosom aj pre samotných autorov frameworku RelBench zo Stanfordu, s ktorými spolupracujeme na oficiálnom zverejnení datasetu a predikčnej úlohy založenej na databáze MIMIC-IV. Zaradenie tejto úlohy do benchmarkovej sady vzbudilo značný záujem v komunite.

Literatúra

[1] Charu C Aggarwal et al. *Data mining: the textbook*, volume 1. Springer, 2015.

[2] Milan Cvitkovic. Supervised learning on relational databases with graph neural networks. *arXiv preprint arXiv:2002.02046*, 2020.

[3] Matthias Fey, Weihua Hu, Kexin Huang, Jan Eric Lenssen, Rishabh Ranjan, Joshua Robinson, Rex Ying, Jiaxuan You, and Jure Leskovec. Relational deep learning: Graph representation learning on relational databases. *arXiv preprint arXiv:2312.04615*, 2023.

[4] William L Hamilton. *Graph representation learning*. Morgan & Claypool Publishers, 2020.

[5] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. Mimic-iv. *PhysioNet. Available online at: https://physionet.org/content/mimiciv/1.0/(accessed August 23, 2021)*, pages 49–55, 2020.

[6] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1, 2023.

[7] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33:19620–19631, 2020.

[8] Chantal Pellegrini, Nassir Navab, and Anees Kazi. Unsupervised pre-training of graph transformers on patient population graphs. *Medical Image Analysis*, 89:102895, 2023.

[9] Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B Wiltschko. A gentle introduction to graph neural networks. *Distill*, 6(9):e33, 2021.

[10] Shirly Wang, Matthew BA McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C Hughes, and Tristan Naumann. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM conference on health, inference, and learning*, pages 222–235, 2020.

[11] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.