Eötvös Loránd Tudományegyetem
Társadalomtudományi Kar
MESTERKÉPZÉS

Szóbeágyazás, tudásgráfok és GAT neurális háló alkalmazása:

heveny hasnyálmirigy-gyulladás súlyosságának előrejelzése

Konzulens:
Rakovics Márton

Készítette:
Varga Tamás
RRMCZ7
Survey statisztika és adatanalitika

2025. április

EREDETISÉGNYILATKOZAT

Alulírott *Varga Tamás* az ELTE TáTK *survey statisztika és adatanalitika mesterképzés* hallgatója fegyelmi felelősségem tudatában nyilatkozom, és aláírásommal igazolom, hogy a *Szóbeágyazás, tudásgráfok és GAT neurális háló alkalmazása: heveny hasnyálmirigy-gyulladás súlyosságának előrejelzése* című szakdolgozat **saját, önálló szellemi munkám,** az abban hivatkozott, nyomtatott és elektronikus szakirodalom felhasználása a szerzői jogok szabályainak megfelelően történt.

Tudomásul veszem, hogy szakdolgozat esetén plágiumnak számít:

- a szó szerinti idézet vagy fordítás közlése idézőjel és hivatkozás megjelölése nélkül;
- a tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát, valamint a HKR 74/A–C. §-aiban foglalt rendelkezéseket megismertem, és tudomásul veszem, hogy

- a szakdolgozatom szövege a benyújtása után plágiumellenőrző szoftverrel vizsgálható,
- plágium esetén szakdolgozatom értékelés nélkül visszautasításra kerül,
- plágiumvétség esetén fegyelmi eljárás indítható.

Budapest, 2025. április 14.

.....

Varga Tamás

Absztrakt

Dolgozatomban a heveny hasnyálmirigy-gyulladás súlyosságának előrejelzésére tettem kísérletet gráfalapú, figyelem-mechanizmust alkalmazó neurális háló segítségével annak érdekében, hogy bemutassak egy olyan keretrendszert és mélytanulási modellt, amely lehetővé teszi a tabuláris adatok gráfalapú megközelítéssel történő prediktív elemzését. A dolgozat részletesen ismerteti a *Graph Attention Network* modell módszertanát és alkalmazását, kiemelt figyelmet fordítva a kutatási problémára vonatkozó legoptimálisabb paraméterbeállítások és döntési pontok bemutatására. A neurális hálón alapuló elemzés eredményeinek kontextusba helyezése érdekében a felvetett kutatási problémát gépi tanulási módszerekkel elemeztem, és az így kapott eredményeket összevetettem a mélytanulási algoritmus eredményével. Mindezek eredményeképpen a dolgozat hozzájárul a tárgyalt figyelem-mechanizmuson alapuló mélytanulási modell módszertani, illetve gyakorlati megértéséhez.

Kulcsszavak: hasnyálmirigy-gyulladás, klasszifikáció, neurális háló, gráfalapú neurális háló, szóbeágyazás, tudásgráf, Graph Attention Network (GAT)

Tartalomjegyzék

Bevezetés

Napjainkban a mesterséges intelligencia fejlődésével az adatok egyre inkább kiemelt jelentőségű stratégiai erőforrássá váltak, hiszen mind a tudományos kutatásban, mind az üzleti döntéshozatalban kiemelkedő szerepet töltenek be. Az évszázad eleje óta ez a megállapítás különösen igaz az orvostudományra, ahol az elmúlt évtizedekben az adatvezérelt döntéseken alapuló diagnosztikai eljárások kiemelt figyelemnek örvendtek, hiszen az újfajta diagnosztikai rendszerek révén lehetőség nyílt a betegséglefolyások pontosabb modellezésére, illetve a kezelések hatékonyságának növelésére. Az utóbbi években a neurális hálókon alapuló mélytanuló algoritmusok jelentős sikereket értek el különböző, nem szigorúan strukturált adatokon alapuló előrejelzési és diagnosztikai problémák vonatkozásában, így különösen alkalmassá váltak a számottevő adatmennyiséget, illetve kifinomult, összetett kapcsolat-mintázati elemzőképességét igénylő klinikai problémák vonatkozásában.

Szakdolgozatom során arra a kérdésre keresem a választ, hogy a gráfalapú neurális háló – azon belül is egy speciális, figyelem-mechanizmust használó modell – miként használható az akut hasnyálmirigy-gyulladás súlyosságnak előrejelzésében. A hasnyálmirigy-gyulladás a magas halálozási arány okán komoly kihívást jelent az orvostudomány számára, így az időben megállapított és lehetőleg minél pontosabb diagnózis kiemelt fontosságú. Munkám során első lépésként röviden bemutatom a konkrét egészségügyi probléma vonatkozásában történt korábbi előrejelzési módszereket, majd rátérek arra, hogy az egészségügyi adatok esetén miként építhető fel egy szóbeágyazáson alapuló, gráf topológiát használó, különböző forrású és szerkezetű adatokat integráló keretrendszer. A szakdolgozat elsődleges célja a keretrendszer módszereinek részletes bemutatása, valamint a neurális hálón alapuló modell egy már kipróbált adatbázison történő alkalmazása annak vizsgálatával, hogy a felvázolt előrejelzési rendszer milyen eredményt képes elérni a hagyományos, gépi tanuláson alapuló módszerekkel összehasonlítva.

1. Elméleti háttér

Ebben a fejezetben elsőként definiálom a kutatási kérdést, illetve kitérek a téma kapcsán született hazai, illetve nemzetközi kutatási előzményekre, melyek a probléma felvetésének alapját jelentették. A kutatási kérdés pontos definiálása után az elemzés keretrendszerét ismertetem, ahol bemutatom, hogy az eddig használt tabuláris adatstruktúrán alapuló elemzési módszerek helyett az adatokat milyen módon lehet áttranszformálni gráf adatszerkezetté. A fejezet utolsó, és egyben legjelentősebb részét a gráfalapú neurális hálók, és azon belül is figyelem-mechanizmuson alapuló *Graph Attention Network* neurális háló ismertetése teszi ki.

1.1. Kutatási probléma definíciója és relevanciája

A kutatási probléma a heveny hasnyálmirigy-gyulladás súlyosságának neurális hálóval történő előrejelzésére vonatkozik, így a statisztikai módszerek ismertetése előtt fontosnak tartom a problémát szolgáltató kórkép általános bemutatását, illetve a súlyosság fogalmának konceptualizálását.

Az Amerikai Egyesült Államok Orvostudományi Nemzetközi Könyvtára (2023) alapján a heveny hasnyálmirigy-gyulladás (*Acute Pancreatitis*, röviden AP) egy potenciálisan halálos kimenettel járó gasztroenterológiai betegség, mely a hasnyálmirigy hirtelen kialakult gyulladásával jár. A betegség esetén megkülönböztetendő a krónikus lefolyású és tartós gyulladás, illetve a heveny (vagy más néven akut) gyulladás, mely egy minden előjel nélkül kialakuló, hirtelen lefolyású kórkép.

A gyulladás súlyosságának kategorizálására a felülvizsgált Atlanta klasszifikáció (Banks et al, 2013) alapján három kategóriát különböztet meg a szakirodalom:

- **enyhe lefolyás:** gyors, komplikációmentes lefutás, szervelégtelenség nélkül
- **közepes lefolyás:** két napon belül elmúló szervelégtelenség
- **súlyos lefolyás:** két napon túl múló, akár több szervre kiterjedő szervelégtelenség

A statisztikai modellezés során a súlyosság bináris változóként kerül előállításra, bináris klasszifikációs problémát eredményezve. A súlyosság szempontjából a közepes és súlyos lefolyású csoport összevonásra kerül, kialakítva ezzel a súlyos csoportot, míg az enyhe lefolyású csoport a nem súlyos kategóriát képviseli.

A modellezés szempontjából ezenfelül érdemes szem előtt tartani, hogy az orvostudomány mai álláspontja alapján a betegség kialakulásának legjelentősebb okai az epekő kialakulása – ami elakadás esetén elzárja a hasnyálmirigy-vezetéket, ezzel megakadályozva a hasnyálmirigynedv elfolyását – és az alkoholfogyasztás.

A probléma relevanciáját növeli az a tény is, hogy az AP a páciensek mellett az ellátórendszerekre is jelentős terhet nyújt. Roberts és szerzőtársai (2017) által végzett európai országokra vonatkozó metaanalízis 1989 és 2015 közötti kutatásokat összegezte azzal a megállapítással, hogy 100 000 főre vetítve évente 4.6 és 100 közötti lakos esetén alakul ki a betegség. A tág intervallum arra enged következtetni, hogy az országok között jelentős szórás figyelhető meg az előfordulási gyakoriságban, azonban a tanulmány gyakoriságra vonatkozó megállapításai – az országonként eltérő jelentési rendszerek miatt – inkább csak a probléma mértékének érzékeltetésre alkalmazhatóak. Ezzel szemben a részletesebb, nagyobb mintán megvizsgált etiológiai vizsgálat erőteljes régiós mintázatokat mutat, epekő dominanciával a mediterrán és nyugat-európai térségben (Olaszország, Görögország, Spanyolország), míg alkohol dominanciával észak- és kelet-európai térségben (Magyarország, Finnország, Románia), ami azt eredményezi a kutatási probléma vonatkozásában, hogy az alkoholfogyasztási szokásokat a modellbe mindenképpen érdemes bevonni.

Li és szerzőtársai (2019) a betegség etiológiája mellett részletesen vizsgálták a kor szerint standardizált előfordulási és mortalitási mutatókat, illetve a szociodemográfiai jellemzők betegségre gyakorolt hatását. Fontos megállapításuk volt, hogy a betegség előfordulási gyakorisága a kelet-európai térségben világszinten az egyik legmagasabb, valamint amíg a világ többi részén az elmúlt 30 évben egy stabil csökkentő tendencia volt megfigyelhető, addig a kelet-európai régióban 11 százalékos növekedés volt tapasztalható az esetszámokban, valamint a halálozási arány is a legmagasabbak között szerepel.

1.2. Korábbi kutatások, felhasznált adatbázis

A gyulladás súlyosságának időben történő felismerése a betegség lefolyása szempontjából kiemelt fontosságú tényező, így az idő múlásával kialakult az igény egy könnyen kivitelezhető előrejelzési rendszer kialakítására. A problémára számos modell került előállításra, azonban a szakdolgozat szempontjából kiemelten fontos Kui és szerzőtársai (2022) által közölt modell, ami az EASY-APP nevet viseli.

A szerzők tanulmányukban szintetizálták a korábban használt predikciós módszereket azzal a megállapítással, hogy nincsen egy olyan kitüntetett laboratóriumi paraméter, ami képes lenne konzisztens módon, már a betegség megjelenésének pillanatában előrejelezni a gyulladás súlyosságának lefolyását. A gyulladásos folyamatokra termelt C-reaktív protein (CRP) szint emelkedése korrelál a betegség súlyosságával, azonban az emelkedés csak a panaszok első megjelenése után 48 órával történik, így a paraméter a betegség súlyosságának beutaláskor történő előrejelzésére nem alkalmazható. Ezenfelül a további, többváltozós modellek is mind olyan paramétereket használnak, melyek nehezen érhetőek el a beutalás utáni kritikus 48-72 órában, ezáltal korai prevencióra nem alkalmasak. Ezzel szemben az EASY-APP módszer olyan változók alapján kísérli meg a modellezést, amelyek könnyen, már a beutaláskor elérhetőek, potenciálisan kialakítva ezzel egy olyan előrejelzési modellt, mely már a korai szakaszban is képes a súlyosság előrejelzésére.

1.2.1. EASY-APP adatbázis

Az elemzés a kutatási problémát szolgáltató Kui és szerzőtársai (2022) által publikált cikk alapjául szolgáló adatbázison alapul, és arra a kérdésre keresi a választ, hogy a cikkben szereplő gépi tanulási alapokon nyugvó megoldásokhoz képest milyen eredmény elérésre képes a gráfalapú, figyelem-mechanizmuson alapuló neurális háló. Mivel a szakdolgozat elsődleges célja a GAT neurális háló, mint módszer bemutatása és alkalmazása, ezért a publikált adatbázis kapcsán elfogadásra került az a feltevés, hogy az már egy kipróbált, előrejelzési modell építésére alkalmas adattömeg, így a dolgozat nem tér ki részletesen az adatok keletkezési módjára, illetve a klinikai adatminőség kérdésére, pusztán egy statisztikai jellegű leíró bemutatást biztosít.

Az EASY-APP adatbázist a hasnyálmirigy-gyulladással kórházba beutalt, 18 éven felüli, írásban belegyező nyilatkozatot adó betegek alkotják. Az eredeti populációs méretet alkotó 1.388 főből 2 beteg a későbbiekben visszamonda a belegyező nyilatkozatát, míg 202 személyt adathiány miatt kellett kizárni, így végül az elemzésbe bevont betegszám 1.184 főre redukálódott. Az elemzésbe 12 ország 22 különböző egészségügyi intézményében kezelt betege került bevonásra, azonban megállapítható, hogy az adatbázisba bekerült személyek döntő többsége (79%-a) magyarországi intézménybe, azon belüli is a Pécsi Tudományegyetem Klinikájára (584 fő) beutalt beteg. (Kui et al., 2022)

A mintába került személyek alapvető demográfiai adatait, kórelőzményeit, illetve néhány, a korábbi kutatások alapján a kór kialakulásában meghatározó paraméterének értékét az alábbi táblázat foglalja össze:

KATEGÓRIA	ÉRTÉK	TARTOMÁNY	ADAT-MINŐSÉG[1]
DEMOGRÁFIAI JELLEMZŐK			
Nem (férfiak százaléka)	58.1%	Nő / férfi	100%
Átlagos életkor	55.7 (16.6)	[17, 95]	100%
Átagos BMI érték	27.98 (5.86)	[14.8, 50.4]	99%
KÓRELŐZMÉNY			
Rendszeresen alkoholt fogyaszt (%)	54.0%	Igen / nem	100%
Dohányzik (%)	34.4%	Igen / nem	100%
Átlagos hasi fájdalom hossza (óra)	36.8 (40.4)	[1, 168]	98%
BEUTALÁSKOR RÖGZÍTETT ÉRTÉKEK			
Hasfali érzékenység jelen (%)	89.6%	Igen / nem	99%
Átlagos testhőmérséklet (Celsius)	36.7 (0.46)	[34.8, 39.0]	98%
Átlagos szisztolés vérnyomás (Hgmm)	141.9 (22.5)	[75, 220]	100%
Átlagos diasztolés vérnyomás (Hgmm)	85.2 (14.3)	[40, 191]	100%
Átlagos pulzusszám	83.9 (16.5)	[41, 153]	100%
Átlagos légzésszám	17.7 (3.7)	[10, 45]	99%
LABOR PARAMÉTEREK			
Átlagos C-reaktív fehérje szint (mg/L)	49.76 (74.5)	[0.07, 515]	100%
Átlagos kreatinin szint (µmol/L)	85.8 (46.7)	[36, 706]	100%
Átlagos glükóz szint (mmol/L)	8.23 (3.48)	[2.53, 43.29]	100%
Átlagos urea (karbamid) szint (mmol/L)	6.32 (3.85)	[0.98, 40.09]	100%
Átlagos fehérvérsejtszám (G/L)	12.78 (5.05)	[1.32, 52.70]	100%

1. táblázat Összefoglaló átlagos értékek, zárójelben a szórással

Forrás: Kui et al., 2022 alapján saját szerkesztés

[1] *Az adatminőség az adathiány mértékét jelzi, 100% esetén nem merül fel adathiány*

1.3. Keretrendszer ismertetése

Ebben a fejezetben a kutatási problémára javasolt elemzési keretrendszer kerül bemutatásra, mely az adatbázis tabuláris jellegű adatait konvertálja át a végső, neurális háló által feldolgozható gráfalapú adatszerkezetté.

1.3.1. Gráf struktúra létrehozása: szóbeágyazás és CUI2VEC

A szóbeágyazási vektortérmodellek az elmúlt évek során kiemelten kutatott területté váltak, hiszen segítségükkel hatékonyan reprezentálhatóak ritka és magas dimenziós szöveges adatok. A kezdeti, szótárakon alapuló *one-hot* kódolás sok egyedi szó esetén rendkívül erőforrásigényesnek, valamint a szavak közötti hasonlóság kifejezésére is alkalmatlannak bizonyult. A szóbeágyazási módszerek központi feltevése Harris (1954) tollából származik, akinek disztribúciós hipotézise úgy fogalmazott, hogy egy adott szó jellemezhető a környezetével, vagyis a hasonló jelentésű szavaknak hasonló kontextussal kell rendelkezniük. A szóbeágyazási vektortérmodellek e hipotézis alapján arra tesznek kísérletet, hogy ezt a kontextuális hatást egy alacsonyabb dimenziós vektortérben modellezzék.

Mindegyik szóbeágyazási modell közös tulajdonsága, hogy a beágyazások feltanítása jelentős méretű korpuszon történik meg, azonban az egészségügyi adatok esetén egy ilyen jellegű korpusz előállítása személyiségjogi kérdések, illetve az adatok különböző formátuma miatt nehezen kivitelezhető. Ezenfelül az adatok különböző formátuma azt is eredményezi, hogy a népszerű szóbeágyazási módszerek (word2vec, GloVe) nem alkalmazhatók közvetlenül ezek az adatokon, mindenképpen valamilyen egységesítő transzformációs lépésre van szükség.

Az előző bekezdésben említett problémák áthidalására került kidolgozásra Beam és szerzőtársai (2020) által a CUI2VEC módszer, ami egységes klinikai terminológiákból (CUI) készít vektortér beágyazásokat. A módszer előnye, hogy a korábbi vektortérmodellek korlátozott nagyságú korpuszai helyett a CUI2VEC modell egy többmillió nagyságú, különböző forrásokból származó korpuszon került feltanításra, lehetővé téve ezzel a különböző forrásból származó terminológiák egységes keretrendszerbe történő foglalását.

A szóbeágyazási modell az alábbi három különböző forrásból származó adatokon került feltanításra:

- 60 millió főt számláló, 2008 és 2015 között időszakot lefedő amerikai biztosítási kárigény adatbázis
- a Stanford Egyetem 20 millió klinikai feljegyzéséből származó klinikai fogalmak együtt-előfordulási mátrixa
- 1.7 millió szabadon elérhető PubMed folyóiratcikk

Az adatbázis sokszínűségéből fakadóan – biztosítási kódok, szöveges feljegyzések, orvosi beavatkozások kódjai – első körben szükséges volt ezeknek a különböző adatstruktúráknak az egységes koncepcióba konvertálására. Ez a konverzió az *Unified Medical Language System* (UMLS) szótára alapján egy egységes koncepció térbe (CUI) történt meg, a különböző formátumú adatpontok felcímkézésével és egységes szótárba történő elhelyezésével. Miután ez megtörtént, egy CUI-CUI együtt előfordulási mátrix került létrehozásra, ahol a mátrix oszlopai, illetve sorai is CUI kódokból álltak, így a mátrix elemei két fogalom együttes előfordulását kvantifikálták. Az együttjárás különbözőképpen került definiálásra a különböző források esetén. A folyóiratcikkek esetén egy 10 szóból álló mozgóablakkal került a kontextus megállapításra, míg a biztosítási adatoknál a CUI párokat az azon páciensek számával definiálták, akik esetén a fogalompár mindkét eleme 30 napos időablakon belül megjelenik. Utolsó lépésként az egységes előfordulási mátrix átalakításra került SSPMI mátrix formába a lenti összefügés alapján, ami után a mátrix már alkalmassá vált az SVD módszerrel történő faktorizálásra. (Beam et al., 2020)

$$PMI(w,c) \;=\; \frac{p(w,c)}{p(w) * p(c)}$$

$$SPPMI(w,c) = max(PMI(w,c) - log(k),\ 0)$$

A PMI mátrix azt határozza meg, hogy két CUI együttes előfordulása mennyire váratlan a független előfordulásokhoz képest. A PMI mátrix értékei egy $log(k)$ értékkel is eltolásra kerülnek, ahol a k paraméter a word2vec eljárás negatív mintavételezési paraméterét takarja. A képlet alapján így csak azok az értékek maradnak meg, melyek meghaladják a $log(k)$ küszöbértéket, vagyis erős kapcsolattal rendelkeznek.

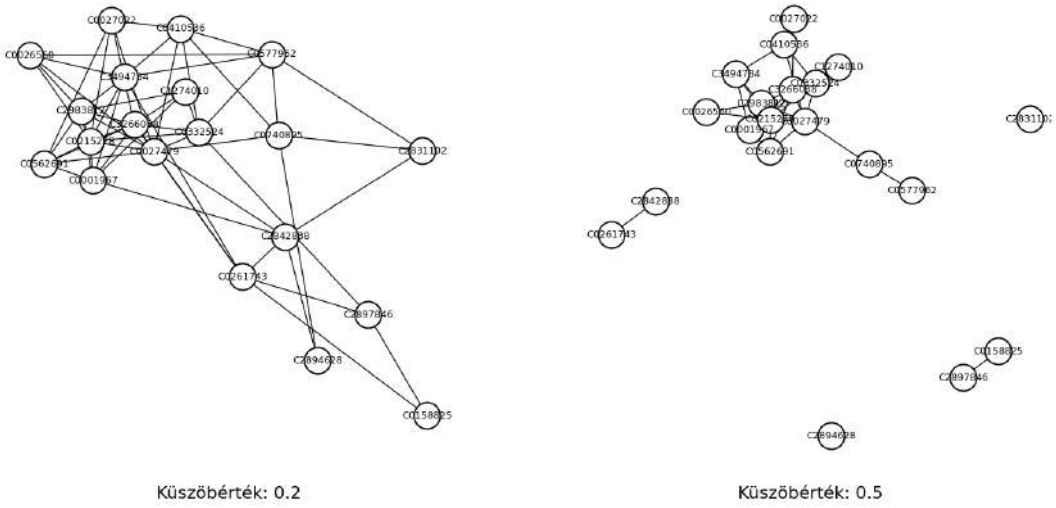
1.3.2. Tudásgráf kialakítása

A dolgozatban ismertetett módszer újdonságát az jelenti, hogy a tabuláris adatok az előző fejezetben ismertetett magas-dimenziószámú klinikai koncepciótér segítségével gráf formátumú szerkezetté kerülnek átkonvertálásra. E logika mögött az az elgondolás áll, hogy a koncepciótérbe beágyazott fogalmak elhelyezkedése, illetve egymáshoz viszonyított pozíciója többletinformációt hordoz egy egyszerű tabuláris struktúrához képest, így egy gráf alapokon nyugvó algoritmussal még egy plusz potenciális információval szolgáló réteg vonható be, ami a várakozások alapján képes a modell mintázatfelismerő, és ezáltal előrejelző képességének további pontosítására.

Egy koncepciókat tartalmazó szóbeágyazási vektortérmodell különböző módszerekkel alakítható át gráf formátummá. Az összes modell esetén azonban egységes az a tény, hogy a kialakuló gráfban a csúcsokat a fogalmak, azaz jelen esetben a különböző klinikai koncepciókat megtestesítő CUI kódok alkotják. Abban az esetben, ha a vektortérmodell nagy mennyiségű fogalmat tartalmaz, akkor a csúcsok összevonása ajánlott, amit valamilyen szinonimitást mérő mértékkel lehet megtenni. Abban az esetben, ha a fogalmak többértelmű kifejezéseket is tárolhatnak, akkor szemantikai elemzés segítségével a fogalmak szükség szerint tovább is bonthatóak. (Winge, 2018)

A modellek közötti legfőbb különbség a csúcsokat összekötő élek kialakításában mutatkozik meg. A naív módszer alapján az összes csúcs összekötésre kerül egy, a magas dimenziós vektortérben értelmezett távolságmérték (pl. euklideszi vagy koszinusz távolság) segítségével, ahol a kialakuló élek súlyozásra kerülnek a csúcsok közötti távolságokkal. Ez a módszer azonban több szempontból is előnytelen. Egyrészt az így kialakuló n darab csúcsot tartalmazó teljes gráf $n(n-1)/2$ élet fog tartalmazni, ami nagy modellek esetén jelentős tárolási igénnyel jár, valamint az így kialakuló struktúra nem reflektál a szemantikai modellek ritka hálózatszerkezetére, így nem képes a fogalmak közötti struktúra modellezésére. (Winge, 2018)

A naív modell ezáltal rávilágít arra, hogy szükséges az élek számának valamilyen módon történő korlátozása annak érdekében, hogy egy ritkább, a fogalmak közötti kapcsolatokat kifejezőbben reprezentálni képes gráf kerüljön kialakításra. Az első módszer a csúcsok közötti távolságok tekintetében bevezet egy globális küszöbértéket, így csak azok az élek kerülnek letárolásra, amik ezt az adott küszöbértéket meghaladják. Ezzel a módszerrel a keletkező gráf összekapcsoltságát lehet dinamikus módon szabályozni, és egy ritkább, a szemantikai modelleket jobban közelítő struktúra alakítható ki. A modell hátránya, hogy a küszöbérték alapja továbbra is egy távolság mérték, amit minden csúcspárra szükséges kiszámolni.



1. ábra Létrejövő tudásgráfok különböző küszöbértékek mentén
(20 elemű mintavételezett CUI vektortérből)

Az első modell ugyan jól képes megragadni a fogalmak közötti kapcsolatokat, azonban könnyen előfordulhat, hogy egy nem kiegyensúlyozott vektortérből történő konvertálás során jelentősen eltérő fokszámú csúcsok alakulnak ki. Abban az esetben, ha a gráfalapú modell erre érzékeny, akkor az átalakítás megtörténhet a legközelebbi szomszédok módszere segítségével is, ahol a csúcsok a k legközelebbi szomszédukkal kerülnek összekapcsolásra, egységes fokszámot eredményezve. A konkrét szakdolgozati probléma során mindkét modell alkalmazásra kerül, azonban előzetesen azzal a hipotézissel lehet élni, hogy az adott speciális orvosi probléma megoldására gyűjtött hasonló klinikai fogalmak miatt a lokális sűrűségek leképzése (kiegyensúlyozatlan fokszám) előnyösebb lehet a modell szempontjából.

1.4. Gráfalapú neurális hálók

A gráfalapú neurális modellek (GNN) a mélytanulási módszerek egyik speciális, gráf struktúrájú adatokra kifejlesztett fajtája. A modellek létrejöttének egyik fő mozgatórugója a gráf egyedi, kevésbé strukturált formája, amely lehetővé teszi számos természeti és egyéb komplex jelenség leírását és reprezentálást. A mindennapi élet során is számos helyen megjelennek a gráfok, kezdve az egészen elvont molekuláris szinttől a különböző társadalmi hálózatok, közlekedési rendszerek, vagy akár a különböző közösségi oldalak és online vásárlási felületek ajánlási rendszerének szintjéig. Maguk a gráfok olyan adatstruktúrák, melyek különböző objektumok közötti kapcsolatot írnak le. Ebből fakadóan a gráfalapú neurális hálók általános célja ezeknek a kapcsolatoknak a felhasználásával olyan potenciális információtöbblettel rendelkező reprezentáció kialakítása, melyek felhasználják az objektumok jellemzői mellett a gráf strukturális szerkezetét is. Az évek során számos felhasználásai területen kerültek alkalmazásra a GNN neurális hálók, azonban leggyakoribb felhasználásuk a gráfok csúcsainak csoportosítására, csúcsok közötti kapcsolatok előrejelzésére, illetve gráf szintű következtetésekre vonatkozó problémák megoldása. (Vrahatis, 2024)

1.4.1. Gráf reprezentálás általánosságban

Mielőtt részletesen bemutatásra kerülne a GNN algoritmusok logikája, fontosnak tartom a gráf, mint adatstruktúra néhány jellemzőjének kiemelését Velickovic (2023) alapján, melyekre a későbbiekben a szöveg visszahivatkozik. Elsőnek definiálni kell magát a gráfot, mint struktúrát, amit a csúcsok (*vertices*) halmazával, illetve a csúcsokat összekötő élek (*edges*) halmazával lehet megtenni.

$$G = V, E$$
$$E \subseteq V \; x \; V$$

Az adatok reprezentálásában a tabuláris adatstruktúrában megszokott változók, illetve megfigyelések egységek szerepét a gráfban a csúcsok és a csúcsokhoz tartozó vektorok veszik át. A bementi gráf felépíthető úgy is, hogy a csúcsok egy megfigyelési egységet jelentenek, azaz ebben az esetben az egy adott változóhoz tartozó k darab megfigyelési egység egy, az adott csúcshoz tartozó k dimenziós vektorban kerül eltárolásra. A gráf reprezentálhat önmagában is egy megfigyelési egységet, amely esetben a csúcsok a változók szerepét töltik be.

$$u \in V \text{ és } x_u \in \mathbb{R}^k$$

$$X = [x_1, x_2, \ldots, x_{|V|}]^T \in \mathbb{R}^{|V| * k}$$

A bementi adatokat a neurális háló az X adatmátrix (*node feature matrix*) formájában dolgozza fel, amely mátrix a csúcspontokhoz tartozó *k* dimenziós (ami a gráf, mint megfigyelési egység reprezentáció esetén *k* = 1) vektorok (x_u) összefűzésével jön létre. A reprezentációs tanulási feladatokhoz a csúcsok – és ezáltal az adatmátrix – előállítása után az élek halmazának az elállítása is szükséges, ami a szomszédossági mátrix segítségével valósítható meg, melynek $a_{u,v}$ elemei a két csúcs közötti él meglétét jelöli. (Velickovic, 2023)

$$a_{u,v} = \begin{cases} 1 & u, v \in E \\ 0 & u, v \notin E \end{cases} \qquad u, v \in V$$

A gráf adatstruktúrán alapuló neurális hálók esetén kiemelendő még az permutáció-invariancia tulajdonság, hiszen a gráfok olyan speciális tulajdonságú adatszerkezetek, melyek estén ugyanaz a gráfszerkezet különböző módon ábrázolható a csúcsok permutációjával. Ez a tulajdonság az X mátrix különböző módon történő felépítéséhez vezet, azonban az összes különböző felépítésnek ugyanazt a globális gráfot szükséges leírnia, így egy adott gráf bármilyen elrendezésű bemenete esetén a kimenetnek azonosnak kell lennie.

$$f\ PX, PAP^T\ = f\ X, A \qquad (permutáció - invarianca)$$

A gráfalapú neurális hálók tanulási folyamata a hagyományos mélytanulási neuronhálók algoritmusán alapul, annak a kiterjesztését jelenti. A tanulási folyamat egy adott rétegének a célja a gráfok csúcsait leíró vektorok új reprezentációja, mely reprezentációs vektor (vagy beágyazás) kialakításában figyelembevételre kerül az adott csúcs közvetlen szomszédos környezete. A reprezentációs folyamat egy iteratív üzenetküldési lépésen keresztül megy végbe, melynek lényege, hogy a csúcsok a szomszédikkal (N_u) interakcióba lépnek, információt fogadnak és küldenek, majd a kapott információ felhasználása után frissítik saját rejtett állapotukat. (Wu et al., 2019)

$$N_u = \{v \mid\ u, v\ \in E\ vagy\ \ v, u\ \in E\}$$

A beágyazott vektorokat is figyelembe véve:

$$X_{N_u} = \{\{x_v | v \in N_u\}\}$$

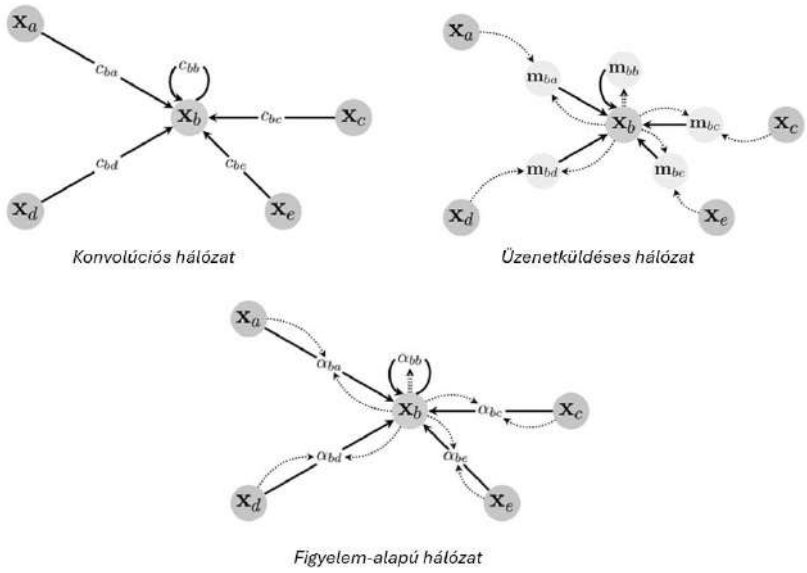
1.4.2. Gráf neurális hálók fajtái

A gráfalapú neurális hálók egy rétegének algoritmusa az előző fejezetben röviden ismertetett általános séma alapján végzi el a tanulást. Első lépésben minden csúcs az éppen aktuális reprezentációját üzenetté alakítja és elküldi (propagálja) a szomszédos csúcsoknak egy előre definiált függvény segítségével. Második lépésben a csúcsok összesítik egy permutáció-invariáns aggregáló függvénnyel (például átlagolás, összeadás vagy maximum kiválasztása) a kapott információkat, majd az így létrejött információval frissítik az éppen aktuális csúcs reprezentációt, létrehozva ezzel egy köztes látens, rejtett állapotot (h'_u).

$$h'_u = \varphi(x_u, X_{N_u})$$

Bronstein és szerzőtársai (2021) alapján a gráfalapú neurális hálókat három alapvető kategóriába lehet besorolni, annak megfelelően, hogy a csúcs saját jellemzői, illetve szomszédos csúcsok információi miként kerülnek feldolgozásra, azaz a frissített csúcs reprezentációt előállító φ függvény milyen transzformációkat végez:

- konvolúciós hálózatok
- üzenetküldésen alapuló hálózatok
- figyelem-alapú hálózatok



2. ábra Gráfalapú neurális hálók

Forrás: Bronstein et al. (2021) alapján saját szerkesztés

A **konvolúciós hálózatok** (Kipf & Welling, 2017) a képfeldolgozásban elterjedt hagyományos konvolúciós neurális hálók alapelvét ültetik át a gráf topológiára, ahol a konvolúciós szűrők egy rögzített terület helyett a gráf csúcsainak a szomszédait dolgozzák fel. A konvolúciós háló a neurális hálók között egy széles-körben elterjedt és számos problémára alkalmazható módszer, ezáltal a különböző további modellek kiértékelése esetén gyakran használt viszonyítási alapként szolgál.

$$h'_u = \phi\left(x_u, \bigoplus_{v \in \mathcal{N}_u} c_{vu}\psi\ x_v\right)$$

Az aggregáló lépés során a szomszédos csúcsok fix súllyal kerülnek összegzésre, amely folyamatban a súly állandó, és a gráf struktúrájától függ. Abban az esetben, ha az aggregáció ($\bigoplus$) összeadással történik, akkor ez megfeleltethető egy helyfüggő lineáris szűrésnek, analógot vonva a képeken használt konvolúciós szűrőkkel, ahol a helyi ablak kontextusában egy adott súlyfüggvény alapján kerülnek aggregálásra a pixelek. (Bronstein, 2021)

Az **üzenetküldés elvén** alapuló gráf neurális hálók az éleken keresztül tanult vektorokat (üzeneteket) juttatnak el a csúcsok között. Ezt az üzenetet egy tanuló függvény (ψ) számítja ki a küldő, illetve fogadó csúcsok jellemzői alapján, lehetővé téve komplex kétoldali interakciók modellezését.

$$h_u = \phi\left(x_u, \bigoplus_{v \in \mathcal{N}_u} \psi\ x_u, x_v\right)$$

Az üzenetküldésen alapuló architektúra a tanulható függvény által képes a releváns információt kiemelni, ezáltal képes kifejező reprezentációk előállítására, azonban ezeknek az üzeneteknek az előállítása erőteljes számítási teljesítményt és memóriát igényel. Ezenfelül számos természetesen módon kialakuló gráfok esetén a két csúcs közötti közvetlen él azonos osztályba tartozást kódol magába (hálózati homofília), mely esetben a konvolúciós aggregáció – a szomszédok közöti hasonlóságot kihasználva – jobb aggregációt és skálázhatóságot biztosít. (Bronstein, 2021)

Az előző bekezdésben kifejtett problémára az arany középutat a figyelem-mechanizmuson alapuló biztosíthatják, amelyek lehetőséget biztosítanak a csúcsok közötti összetett interakciók modellezésére, azonban ezt skalár értékek kiszámításával teszik, ami által jobban skálázhatóbbak, mind az üzenetküldésen alapuló változatok.

1.4.3. Graph Attention Network (GAT)

A Velickovic és szerzőtársai (2018) által javasolt Graph Attention Network (GAT) a gráfalapú mélytanulási hálók egyik speciális architektúrája, aminek fő újítása a szomszédos csúcsok információinak aggregálása során alkalmazott figyelem-mechanizmus. A módszer előnye, hogy amíg a konvolúciós alapon nyugvó hálózat egyedül a fokszám normalizálás során veszi figyelembe egy adott csúcs környezetét, addig a GAT modell nem csak egy adott csúcs fokszámát veszi figyelembe, hanem az adott csúcs jellemzőjének a gráf struktúrában elfoglalt fontosságát is.

A megközelítés lényege, hogy az aggregálás során a szomszédoktól származó információkat nem statikus – például az előző bekezdésben említett fokszámok – vagy egyenlő súlyokkal veszi figyelembe a modell, hanem dinamikus, tanult súlyozást alkalmaz, elősegítve a lokális mintázatok megtanulását, azonban mindvégig figyelembe véve a gráf globális szerkezetét. A GAT alapját jelentő figyelem-mechanizmuson alapuló módszerek az elmúlt években kiemelt figyelmet élveztek, különösen a szekvenciális adatokon alapuló modellek esetén. Népszerűségük abból fakad, hogy a figyelem-mechanizmuson alapuló modell képes a változó méretű bementi adatokból csak a legrelevánsabb információk kiemelésére, ezáltal javítva a döntéshozatal pontosságát és hatékonyságát. A figyelem-mechanizmus kiterjesztése a gráf adatszerkezetre a meglévő módszerek logikáján alapult: a csúcsok rejtett állapotának előállítása a szomszédos csúcsok információnak feldolgozásával, *self-attention*[2] stratégia mentén.

[2] *A self-attention mechanizmus egy azonos szekvencia elemeinek kapcsolatait veszi figyelembe*

1.4.3.1. Architektúra és algoritmus

A következőkben Velickovic és szerzőtársai (2018) által közölt tanulmány alapján a GAT alapvető építőelemének számító figyelmem-koefficiens kiszámításának, és az ennek a felhasználásával kialakuló csúcsreprezentáció kialakításának az algoritmusa kerül bemutatásra. Az algoritmus bementi adathalmaza a csúcsok (N), illetve a csúcsokhoz tartozó jellemzők száma (F), aminek az értéke lehet egy (abban az esetben, ha a csúcs egy adott változót képvisel) vagy egy változóhalmaz számossága (ebben az esetben a csúcs egy entitást reprezentál).

$$\boldsymbol{h} = \{\overrightarrow{h_1}, \overrightarrow{h_2}, \dots, \overrightarrow{h_N}\} \text{ , ahol } \overrightarrow{h_i} \in R^F$$

A különálló rétegek a bemeneti vektorhalmazt átalakítják, és kimeneti adathalmazként (h′) visszaadnak egy módosított vektorhalmazt, ahol a csúcsokhoz tartozó változó-vektorhalmaz kardinalitása (F′) potenciálisan eltér a bemeneti halmaztól.

$$\boldsymbol{h}' = \{\overrightarrow{h'_1}, \overrightarrow{h'_2}, \dots, \overrightarrow{h'_N}\} \text{ , ahol } \overrightarrow{h'_i} \in R^{F'}$$

A csúcsokhoz tartozó bemeneti változók magasabb dimenzióba konvertálásához szükséges egy lineáris transzformáció, mely egy tanult súlymátrix alkalmazásával valósul meg, amely az összes csúcsra alkalmazásra kerül.

$$\boldsymbol{W} = R^{F' \times F}$$

$$\boldsymbol{W}_{F' \times F} * h_{i_F} \in R^{F'}$$

A lineáris transzformáció elvégzése után a következő lépés az *attention* koefficiensek kiszámolása, mely a figyelem-mechanizmus lépés során, egy egydimenziós térbe történő leképzéssel történik. A művelet eredményeképpen minden csúcspárhoz kiszámításra kerül egy skalár érték, mely a két csúcs közötti kapcsolat fontosságát reprezentálja.

$$e_{ij} = a(\boldsymbol{W}h_i, \boldsymbol{W}h_j)$$

$$a : R^{F'} \times R^{F'} \to R$$

Az algoritmus hatékonysága érdekében a figyelem koefficiens kiszámítását korlátozni szükséges a csak az egymással kapcsolatban álló csúcsokra (maszkolás) ezzel is megőrizve a gráf szerkezetéből származó strukturális információt. Az egymással kapcsolatban álló csúcsok fogalom a gráf szomszédos csúcsainak definiálásával (N_i) formalizálható, így az e_{ij} koefficiens csak az $i \in G$ csúcs szomszédságában lévő $j \in N_i$ csúcsokra kerül kiszámolásra.

Annak érdekében, hogy a kiszámított skalár értékek a különböző csúcsok között összehasonlíthatók legyenek, a koefficiensek normalizálására van szükség, ami a *softmax* függvény használatával történik meg. A *softmax* függvény normalizálja az *attention* értékeket úgy, hogy azok 1-re összegződjenek, ezzel is biztosítva a relatív fontosságok összehasonlíthatóságát és jobb interpretálhatóságát, valamint csökkentve az instabilitás és túlillesztés kockázatát.

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}} exp(\, e_{ik})}$$

A koefficiensek kiszámításához még tisztázatlan kérdésként fennmaradt a figyelem-mechanizmus által megállapításra kerülő e_{ij} normalizálatlan koefficiensek kiszámítása. A GAT esetén ezeknek az értékeknek a kiszámítása egy egyrétegű, előrecsatoló (*feedforward*), paraméterezett neurális hálóval történik, $a \in R^{2F'}$ tanult súlyvektor-paraméterrel, illetve LeakyReLU aktivációs függvénnyel.

$$e_{ij} \;=\; LeakyReLU(\; a^T[Wh_i \;||\; Wh_j]\;)$$

A LeakyReLU aktivációs függvény a ReLU aktivációs függvény módosított változata, ami a függvény inaktív neuron problémájának kiküszöbölésére céljából jött létre.

$$ReLU(x) \;=\; max(0, x)$$

A ReLU függvény népszerű választás a neurális hálók aktivációs függvényeként, mivel számítási szempontból hatékony, és elősegíti a hálózat ritkított (*sparse*) reprezentációjának kialakulását. Ennek hatására számos neuron inaktívvá válik (azaz 0 értéket vesz fel), ami hozzájárulhat a teljesítmény javításához és a túlillesztés elkerüléséhez. Azonban a ReLU függvénynek a hátránya, hogy a negatív *bias* taggal rendelkező neuronok a folyamat teljes egészében inaktívak maradhatnak, ezzel szignifikánsan csökkentve a neurális háló kapacitását, és ezzel együtt a komplex folyamatok megtanulásának képességét. E probléma kiküszöbölése érdekében került bevezetésre a LeakyReLU függvény, ami egy paraméter formájában a negatív értékek esetén is bevezet egy lineáris meredekséget.

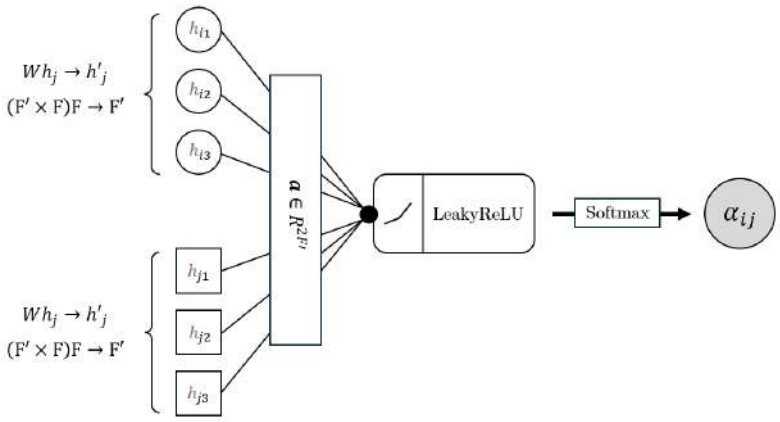
$$LeakyReLU\; x \;= \begin{cases} \alpha x & , ha\; x \;<\; 0 \\ x & , ha\; x \;\geq\; 0 \end{cases}$$

Mindezek fényében a koefficiensek kiszámítása az alábbi képlet formájában írható fel, ahol a LeakyReLU aktivációs függvény α paramétere 0,2.

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\ \mathrm{a}^{\mathrm{T}}[\mathrm{Wh_i}||\mathrm{Wh_j}]\ ))}{\sum_{\mathrm{k}\in\mathcal{N}_i} \exp(\text{LeakyReLU}(\ \mathrm{a}^{\mathrm{T}}[\mathrm{Wh_i}||\mathrm{Wh_j}]\ ))}$$

A fenti képletben a || művelet két mátrix összefűzését, míg a T transzponálást jelöl. Belátható, hogy a fenti képlet egy skalár számot eredményez:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\ \overbrace{\mathrm{a}^{\mathrm{T}}}^{1}[\underbrace{\mathrm{Wh_i}}_{(\mathrm{F}'\times\mathrm{F})\mathrm{F}}||\underbrace{\mathrm{Wh_j}}_{(\mathrm{F}'\times\mathrm{F})\mathrm{F}}]\ ))}{\sum_{\mathrm{k}\in\mathcal{N}_i} \exp(\text{LeakyReLU}(\ \underbrace{\mathrm{a}^{\mathrm{T}}}_{(1\times 2\mathrm{F}')}[\underbrace{\mathrm{Wh_i}}_{(\mathrm{F}'\times\mathrm{F})\mathrm{F}}||\underbrace{\mathrm{Wh_j}}_{(\mathrm{F}'\times\mathrm{F})\mathrm{F}}]\ ))} \quad \underbrace{\qquad}_{(2\mathrm{F}'\times 1)}$$

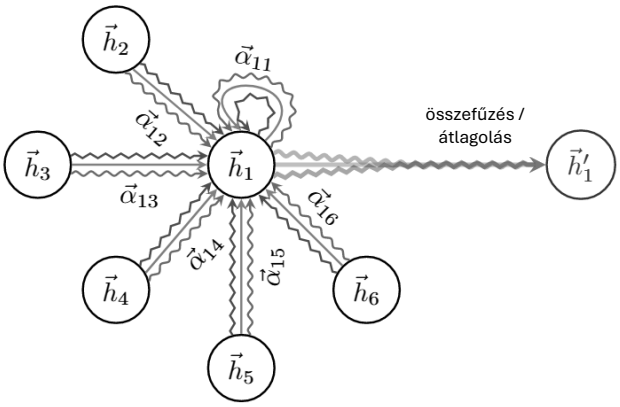


3. ábra Koefficiensek létrejöttétnek folyamata
Forrás: Velickovic (2018) alapján saját szerkesztés

Utolsó lépésként egy adott csúcs módosított reprezentációja a szomszédos csúcsok változóinak és a csúcsokhoz tartozó figyelem-koefficiensek lineáris kombinációjaként kerül előállításra, mely folyamat eredményeképpen egy olyan új csúcsreprezentáció kerül előállításra, ahol a szomszédos csúcsok információi eltérő, a modell által megtanult jelentőségüknek megfelelő súllyal kerülnek feldolgozásra.

$$h'_i = \sigma(\sum_{j\in N_i} \alpha_{ij} \boldsymbol{W} h_j)$$

Velickovic (2018) úgy érvelt, hogy a tanulási folyamat stabilabb működése érdekében a modell kiegészíthető egy párhuzamos figyelem-mechanizmus folyamattal, melyre a *multi-head attention* módszerként hivatkozik. A folyamat során a korábban leírt algoritmus K párhuzamos szálon fut, ahol minden szál különálló tanult paraméterekkel rendelkezik, majd a szálak összegzésre kerülnek.



4. ábra Párhuzamos számítás
Forrás: Velickovic (2018)

Az összegzés a köztes rétegek esetén összefűzéssel történik egészen a kimeneti rétegig, ahol az eredmények átlagolásra kerülnek, majd alkalmazásra kerül az aktiváció.

$$h'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}}\alpha_{ij}^{k}W^{k}\,h_j\right)$$

Velickovic (n.d) arra a megállapításra jutott, hogy a *droupout* technika alkalmazásával a modell robusztussága és általánosíthatósága tovább javítható, különösen alacsony elemszámú tanítóhalmaz esetén. Srivastava és szerzőtársai alapján (2014) a *dropout* egy olyan regularizációs technika, amely a tanítási lépés során véletlenszerűen, $1-p$ valószínűséggel inaktivál bizonyos neuronokat, megakadályozva a neuronok túlzott egymásra épülését, ezzel is csökkentve a túlillesztést. A túlillesztés a neurális hálók esetén különösen alacsony esetszámú tanítóhalmaz esetén jelent problémát, hiszen ebben az esetben a rétegek közötti összetett, nem-lineáris kapcsolatok mintavételezési zaj alapján alakulhatnak ki. A GAT architektúra esetén a technika kiterjed az *attention* koefficiens számítására, az algoritmus a kofficiens kiszámítása során sztochasztikusan mintavételezett csúcsokkal került kapcsolatba.

1.4.4. Graph Attention Network továbbfejlesztése (GATv2)

A GAT architektúra az elmúlt években jelentős népszerűségnek örvendett, és a gráfokon alapuló reprezentációs tanulási feladatok esetén az egyik legelterjedtebb algoritmussá vált. Brody és szerzőtársai (2022) azonban tanulmányukban úgy érveltek, hogy a GAT által eredetileg alkalmazott szomszédos csúcsok információinak aggregálására vonatkozó algoritmus továbbfejleszthető egy olyan változattá, ami az aggregálás során egy rugalmasabb és dinamikusabb figyelem-mechanizmust használ. Munkájuk eredménye a GAT továbbfejlesztésének tekinthető GATv2 architektúra.

A tanulmány úgy érvel, hogy az eredeti GAT algoritmus az *attention* koefficiensek kiszámításakor statikus figyelem-mechanizmust használ. Ez a fogalom arra utal, hogy a szomszéd csúcsok rangsorolásának módját nem befolyásolja az adott csúcs reprezentációja, a gráf összes csúcsa ugyanolyan módon értékeli a folyamat során a szomszédait, ami potenciálisan csökkenti a modell kifejezőképességét. A GATv2 algoritmus erre válaszként bevezette a dinamikus súlyozást, ahol minden csúcs dinamikusan tudja meghatározni a legrelevánsabb szomszédos csúcsokat. Ennek megfelelően az algoritmus dinamikusan, az adott csúcs reprezentációjától függően alaktíva képes a szomszédos csúcsokhoz tartozó súlyok megállapítására. Az algoritmus ezzel eléri azt, hogy a csúcsok nem egy globális, minden csúcs esetén azonos rangsor alapján, hanem csúcsonként eltérő, egyedi szempontok alapján válasszák ki a legfontosabb információkat, amely tulajdonoság különösen előnyös komplex, heterogén gráfok esetén.

A dinamikus súlyozás megvalósítása a műveletek sorrendjének megváltoztatásával történik. Az eredeti GAT algoritmus esetén először minden csúcs reprezentációja átalakításra kerül egy lineáris transzformációval, majd az adott csúcs és szomszédjának módosított reprezentációja összefűzésre kerül, amire egy tanult súlyvektor kerül alkalmazásra.

$$e_{ij} \;=\; LeakyReLU(\; a^T[Wh_i \;||\; Wh_j]\;)$$

A fenti képletben szereplő tanult súlyvektor felbontható két komponensre, így az előző képlet az alábbiak szerint átalakítható:

$$a \;= [a_1 \;||\; a_2] \;\in\; R^{2F\prime} \;\rightarrow\; a_1, a_2 \;\in\; R^{F\prime}$$

$$e_{ij} \;=\; LeakyReLU(\; a_1^T Wh_i \;+\; a_2^T Wh_j\;)$$

A dinamikus súlyozással operáló GATv2 esetén a műveletek sorrendje megváltozik: elsőnek a két reprezentáció kerül összefűzésre, majd egy közös lineáris transzformációs és nemlineáris aktivációs függvény kerül alkalmazásra a létrejött mátrix elemeire, és csak ezután, utolsó lépésként kerül az eredmény beszorzásra a tanult súlyvektorral.

$$e_{ij} = \mathrm{a}^{\mathrm{T}} \, \mathrm{LeakyReLU}(\, \mathrm{W}[\mathrm{h_i} \,||\, \mathrm{h_j}] \,)$$

Intuitív módon belátható, hogy az eredeti képlet módosításában szereplő $\mathrm{a}_1^{\mathrm{T}}\mathrm{Wh_i}$ tag azonos minden j- re, azaz a kiindulási csúcs (vagy jelen esetben *query*) csak egy állandó eltolást ad e_{ij}-hez, a végső koefficiens értékének megállapítása kizárólag a $\mathrm{a}_2^{\mathrm{T}}\mathrm{Wh_j}$ alapján történik. Ebből következik, hogy az adott kiindulási csúcs reprezentációjának nincsen szerepe a szomszéd csúcsok rangsorolásában, ami statikus figyelemként került definiálásra. A GATv2 esetén a lineáris transzformáció a teljes, összefűzött vektorra került alkalmazásra, így a két csúcs információi együttesen, és nem additív módon kerülnek összevezetésre és felhasználásra az aktiváció és súlyvektor alkalmazása során. Ebből következik, hogy a kiindulási csúcs nem csak egy állandó eltolást biztosít, hanem dinamikusan befolyásol. (Brody et al., 2022)

1.4.5. GAT architektúra előnyei és hátrányai

A különböző architektúrák és algoritmusok ismertetése után fontos kiemelni, hogy általánosságban a figyelem-mechanizmuson alapuló gráfalapú neurális hálók milyen előnyökkel és hátrányokkal rendelkeznek, miért jelenthetnek ideális választást a szakdolgozat kutatási kérdésének megválaszolásában, illetve milyen fő különbségekben térnek el a többi gráfalapú neurális hálóktól.

A GAT neurális hálók megkérdőjelezhetetlen előnye az előző bekezdésekben részletesen kifejtett figyelem-mechanizmus, melynek bevezetésével a modell a csúcsok információnak aggregálásakor súlyozottan veszi figyelembe a szomszédos csúcsoktól származó az üzeneteket a rejtett állapot reprezentációjának kialakításához. Ez a tulajdonság a szakdolgozat kutatási kérdése szempontjából különösen ideális, hiszen a szóbeágyazások révén létrejött tudásgráfok alapján a GAT képest felismerni azokat a klinikai paramétereket, melyek a kórkép kialakulása szempontjából szorosabb összefüggésben állnak. A modell ezenfelül javítja az interpretálhatóságot is, hiszen az *attention* koefficiensek vizsgálatával megállapítható, hogy mely csúcsok közötti kapcsolatok járulnak hozzá leginkább a modell döntéséhez.

Ezek a jellmezők lehetőséget biztosít a neurális hálók feketedoboz tulajdonságának enyhítésére, ami különösen az orvostudományi esetekben rendkívül előnyös tulajdonság, hiszen ezen a területen az ok-okozati összefüggések feltárása magas prioritást élveznek. Szintén előnyként kiemelendő a GAT azon tulajdonsága, hogy a lineáris transzformáció által képes az eredeti bemenő változó értékeit egy magasabb dimenziós jellemző vektorrá alakítani, így a szóbeágyazásokból származó részletes szemantikai információkat kombinálva a gráf szerkezettel egy összetett, többdimenziós jellemzőtér létrehozására van lehetőség, ami jelentősen javíthatja a diagnosztikai mintázatok felismerését. (Velickovic, n.d)

Velickovic (n.d) alapján az architektúra a felépítéséből adódóan a GAT számos egyéb technikai jellegű előnnyel is rendelkezik:

- alacsony számítási teljesítmény a párhuzamos folyamatoknak, valamint a paraméterek számának a gráf méretétől való függetlenségének köszönhetően
- a mátrix szorzások és a ritka mátrixszerkezetek miatt az algoritmus alacsony tárhelyigénnyel rendelkezik
- induktív tanulási feladatokra alkalmas algoritmus a globális helyett lokális, élek menti számításnak köszönhetően

Az előnyök mellett, mint minden statisztikai modell, a GAT is rendelkezik limitációkkal, amelyeket az adott problémára való alkalmazhatóság során figyelembe kell venni. Az egyik hátránya a modellnek a korábban már említett túlillesztés kockázata. A figyelem-mechanizmusból fakadóan a neurális háló nagy számú paramméterrel képes alkalmazkodni a tanulási folyamat során, mely nagy szabadság alacsony elemszámú vagy zajjal terhelt adat esetén a túlillesztés kockázatát növeli. A túlillesztés csökkenthető erősebb regularizáció vagy *dropout* módszer alkalmazásával, illetve a rétegek és neuronok számának csökkentésével. (Vrahatis et al., 2024)

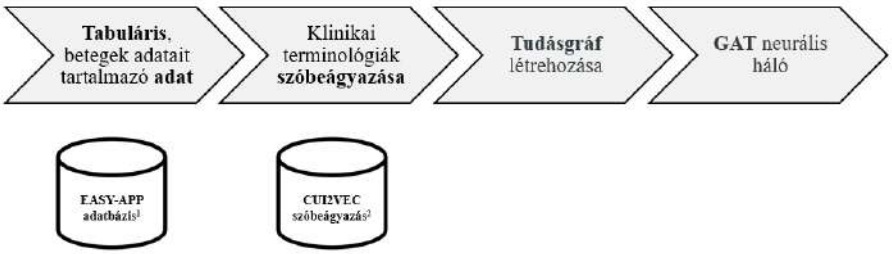
Ezenfelül a modell előnyeként említett interpretációs tulajdonsága esetén fontos szem előtt tartani, hogy a figyelem-mechanizmus alapvetően lokális interpretálhatóságot biztosít, azaz alkalmas az alapvető csúcsok közötti kapcsolatok feltérképezésére, de nem oldja fel teljesen a neurális hálók feketedoboz tulajdonságát. Ennek az az oka, hogy a végső előrejelzést számos súly és csúcs-jellemző együttes hatása adja, és sok esetben a különböző transzformációk miatt a rejtett rétegekben már nem valósítható meg a közvetlen interpretáció, hiszen a látens rétegek neuronjai a kiindulási csúcs-reprezentáció egy már transzformált változatával dolgoznak, melyekhez nem kapcsolható közvetlen és intuitív definíció.

Ezenfelül a GAT esetén egyéb, technikai jellegű korlát is felmerül. Ugyan a párhuzamos folyamatok miatt a számítási-teljesítmény igény optimalizálásra került, még így is egy GAT réteg nagy konstans szorzóval dolgozik a többi alapvető gráf neurális hálókhoz képest. Ennek az oka, hogy mindegy egyes élre külön-külön szükséges egy koefficiens kiszámítása, mely probléma tovább növekszik, ha egy sűrű szerkezetű gráfon történik a tanulás, ahol az élek száma kvadratikusan változik. Nagy fokszámú csúcsoknál a *softmax* normalizáció numerikus instabilitást okozhat, illetve a gradiens sok és nem számottevő súly esetén eltűnhet, ami a *vanishing gradient* jelenséghez vezethet. (Vrahatis et al., 2024)

2. Saját kutatás

A szakdolgozat gyakorlati célja az előző fejezetben bemutatott GAT módszer valós példán történő alkalmazása, és az így kapott eredmények kontextusba foglalása. Az elemzés a már korábban bemutatott hasnyálmirigy-gyulladás gyanújával beutalt páciensek adatait tartalmazó adatbázison került elvégzésre, amely adathalmazon egy súlyosságra becslést adó, bináris klasszifikációt elvégző gráfalapú neurális háló került illesztésre. Az eredmények kontextusba helyezése érdekében a neurális hálón alapuló modell mellett három gépi tanulási algoritmuson alapuló modell is illesztésre került az adatbázison, melyek viszonyítási és összehasonlítási alapként szolgálnak a neurális háló eredményeinek értelmezésekor.

A fejezet során az elemzés különböző lépései kerülnek részletesen bemutatásra azzal a céllal, hogy a lépések világosan dokumentáltak és akár reprodukálhatóak legyenek. A dolgozat célja az is, hogy a végső modell kialakításához vezető döntési pontok részletesen bemutatásra és feltüntetésre kerüljenek, ezzel is csökkentve a neurális hálókra jellemző feketedoboz jelenséget. A saját kutatás a tabuláris adatokon történő adattisztítás és előfeldolgozás lépéseiből, az adatok gráf struktúrába ágyazásából, a benchmark modellek elkészítéséből, és végül a gráfalapú neurális háló alkalmazásából állt. A felhasznált adatbázisokat az elméleti háttér szekcióban bemutatott betegek adatait tartalmazó EASY-APP, és a szóbeágyazási vektortérmodellt alkotó CUI2VEC adattömegek jelentették.



5. ábra Az elemzés lépései

Az elemzés teljes egészét Python programnyelven végeztem el, annak is a 3.13.0 verziójával. A Python egy nyílt forráskódú programozási nyelv, melynek adatelemzési szempontból kiemelt előnye a szabadon elérhető, nagy számú kiegészítő könyvtár.

Az elemzés során az alapcsomag mellett az adattisztítás és adatfeldolgozás lépéseihez a pandas 2.2.3, valamint NumPy 2.1.3 verzióját, míg az előfeldolgozáshoz, illetve a gépi tanuláson alapuló modellekhez a Scikit-learn 1.6.1 verzióját használtam. A gráfalapú neurális háló implementálását a Pytorch könyvtár geometrikus modelleket tartalmazó Pytorch-geometric könyvtár 2.6.1 verziójával végeztem el, míg a gráf felépítése a NetworkX 3.4.1 verziójával történt. A különböző vizualizációkat a Matplotlib 3.9.2 és Seaborn 0.13.2 verziójával készítettem el.

2.1. Adatok előfeldolgozása és tisztítása

Első lépésként az EASY-APP adatbázis, illetve a CUI2VEC vektortérmodell összekapcsolására volt szükség annak érdekében, hogy az adatbázisban található fogalmak elhelyezésre kerüljenek a CUI2VEC vektortérben. Az adatbázisban a magyarázó változók szerepét betöltő 46 különböző klinikai paraméterek közül 9 darab kizárásra került az alábbi indokok miatt:

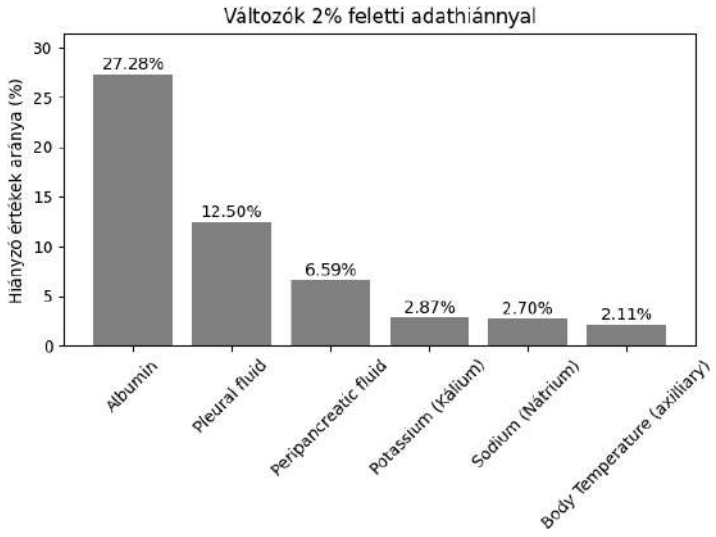
- **Mortality:** a súlyosság kategóriában implicit szerepel
- **Charleson:** nem szerepel a CUI2VEC vektortérben, a többi változóval is kifejezhető
- **Abdominal guarding:** nem szerepel a CUI2VEC térben, többi változóval kifejezhető
- **Peripancreatic fluid:** nem szerepel a CUI2VEC vektortérben
- **Idiopathic:** hiányos definíció, nem beazonosítható fogalom
- **Diet-induced:** hiányos definíció, nem beazonosítható fogalom
- **Drug-induced:** hiányos definíció, nem beazonosítható fogalom
- **Tumor:** nem egyértelmű definíció
- **Other:** hiányos definíció, nem beazonosítható fogalom

A súlyosság magyarázott változó a kutatási kérdés definíciója szekcióban leírtak alapján bináris változóvá került átalakításra, az enyhe lefolyás (eredeti 1) átkategorizálásra került nem súlyos (0) kimenetté, míg a közepes lefolyás (eredeti 2) és súlyos lefolyás (eredeti 3) összevonásra került, kialakítva ezzel a súlyos (1) kategóriát. Ezzel a kategorizálással az alábbi, enyhén kiegyensúlyozatlan eloszlás alakult ki:

Kimenet (súlyosság)	Esetszám
0 (nem súlyos)	878 beteg
1 (súlyos)	306 beteg

2. táblázat Kimeneti változó (súlyosság) eloszlása az adatbázisban

Az adatokbázisban a változók folytonos, illetve kategoriális mérési szintűek. A kategoriális változók mind bináris kimenetelűek, így dummy változók létrehozására nem volt szükség, az adatokat ebben a formátumban képes feldolgozni a gráfalapú neurális háló. Az adatbázisban az adathiány a legtöbb változó esetén alacsony, a 6. ábra szemlélteti azokat a változókat, amelyek 2% feletti adathiánnyal rendelkeznek. Az ábráról leolvasható, hogy 3 változó esetén 5% feletti az adathiány, valamint az *Albumin* esetén kiemelkedően magas, 25% feletti. Ennek következtében az *Albumin* változó sem került végül bevonásra a modellbe, ami végül így 36 darab magyarázó változót eredményezett.



6. ábra Az adatbázis változóinak 2% adathiány feletti részhalmaza (saját szerkesztés)

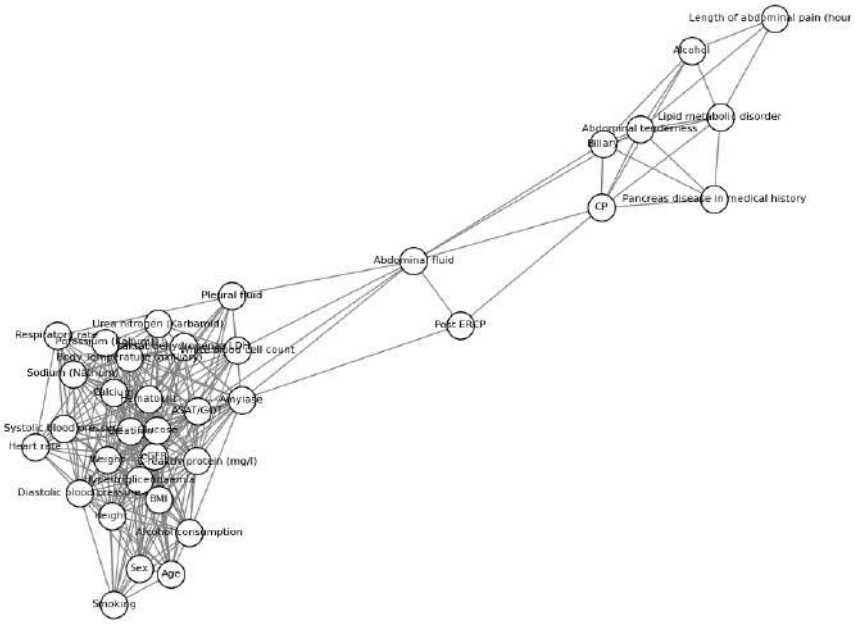
Az adathiányok kezelése imputálással történt, a Kui és szerzőtársai (2022) által is alkalmazott legközelebbi szomszédok módszeren alapuló KNN imputáló algoritmus segítségével. Az algoritmus az adott beteg hiányzó értékeinek becslésére a beteghez 10 leginkább hasonló (euklideszi távolság alapján) beteg értékeinek átlagát használja fel, ezáltal képes az adatok belső struktúrája alapján pótolni az értéket, többváltozós rendszerben. Az algoritmus a bináris kategorikus változókat szintén az átlaggal helyettesíti, ami nem bináris eredményhez vezetett az adathiánnyal érintett esetekben, így ez utólag korrigálásra került az eredmények [0,1] kategóriákká konvertálásával. A *nem* az adatbázisban szöveges kategorikus változóként szerepelt, ami ezáltal lecserélésre került bináris változóra, ahol a női kategória jelentette a referenciakategóriát. Az imputált adatokon utolsó lépésként standardizálás is alkalmazásra került.

A következő lépés a már teljes adatbázis fogalmainak CUI2VEC vektortérben történő megfeleltetése, és a fogalmakhoz tartozó 500 dimenziójú vektorok kinyerése. A fogalmak és a CUI kódok összepárosítására az Amerikai Egyesült Államok Orvostudományi Nemzeti Könyvtára által publikált egységes orvosi nyelvrendszer (UMLS) adatbázisát használtam (NIH, n.d.), mely adatbázis ingyenesen elérhető kutatási célra szóló licensz igénylésével. A fogalmak összepárosításakor igyekeztem a legpontosabb egyezéssel előállítani a megfeleltetést, így a laboreredmények kapcsán a mérési fogalmakat, a több változó esetén pedig az általános fogalmakat használtam a párosításhoz.

2.2. Tudásgráf kialakítása

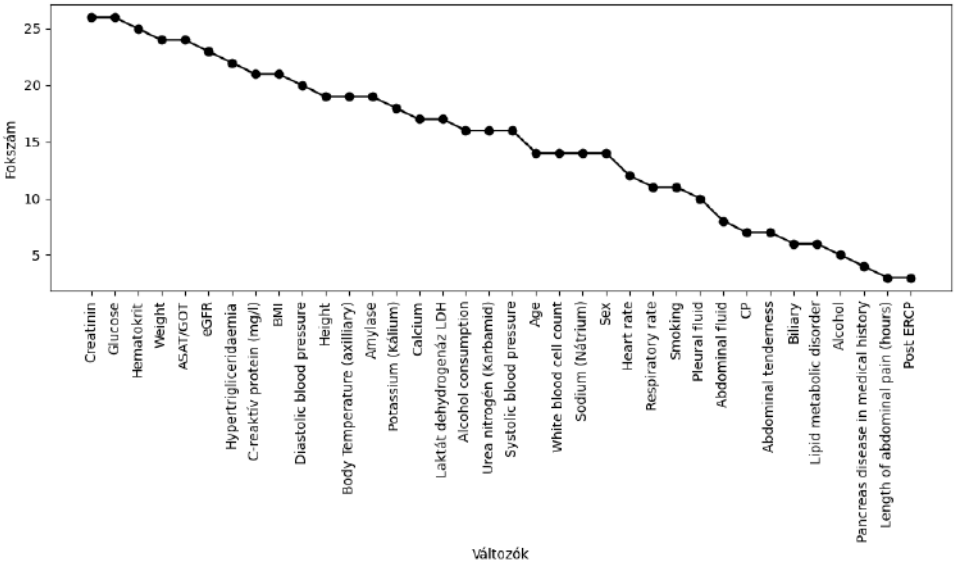
A fogalmak vektortérben történő elhelyezése után egy 36 elemből álló, 500 dimenziós tér alakult ki, mely magas dimenziós vektortér gráffá történő konvertálása az utolsó szükséges lépés volt a keretrendszer kialakításához. Az elméleti háttér szekcióban két tudásgráf kialakítási módszer került bemutatásra, melyek közül a végleges modell a küszöbértéken alapuló gráf kialakítás módszert használja. E mögött a döntés mögött az az elgondolás áll, hogy a küszöbérték módszer biztosítja a szemantikai értelemben vett pontosabb hasonlósági leképezést, mivel a legközelebbi szomszédok módszere minden esetben, még relatíve nagy távolság esetén is létrehoz egy adott számú élet, ami könnyen zajos kapcsolatokat eredményezhet. Ezenfelül a probléma orvosi természetéből adódóan egyes fogalmak általánosabbak, gyakrabban járnak együtt egyéb fogalmakkal, míg vannak egyedi, kevésbé összekapcsolt fogalmak, melyek esetén érdemes lehet megőrizni az elszigeteltséget. A döntés mellet szólt az is, hogy egy küszöbértékkel létrehozott gráfban – mint ahogy az a lenti, 7. ábrán is látszódik – könnyebben kialakulnak klaszterek, így a modell elméletben könnyebben képes felismerni az összetartozó fogalmakat, ami a predikciós képesség növekedéséhez vezethet.

A modell esetén egy gráf egy adott beteg adatait tartalmazza, ahol a csúcsok a betegre vonatkozó megfigyeléseket/változókat jelképezik, a csúcs értéke pedig az adott beteg, adott változóra vonatkozó értéke.



7. ábra Létrejött gráf (0.28 küszöbértékkel, saját szerkesztés)

A végleges gráfban a vektorok között távolság mértékének a magas-dimenziós vektorterekben jobban alkalmazható koszinusz távolság került használatra, így a csúcsok egy bizonyos koszinusz távolság felett kerültek összekötésre. A választott küszöbérték 0.28-ra esett, amely érték megválasztásával egy közepes mértékben összekötött, összefüggő gráf alakult ki, 0.43-as sűrűség értékkel. A küszöbértéken alapuló módszer egy viszonylag kiegyensúlyozatlan fokszámeloszlást eredményezett, azonban ez a fent leírtak alapján nem jelent problémát.



8. ábra Fokszámeloszlás változók szerint (saját szerkesztés)

2.3. Elemzés

A fejezet fő célja, hogy bemutassa az adott problémára vonatkozó legoptimálisabb neurális hálózat paraméterezést, és a mélytanulási háló eredményeit összehasonlítsa a hagyományos, gépi tanulási alapokon nyugvó modellekkel. A fejezet első részében bemutatásra kerül az a világos szempontrendszer, amely alapján különböző modellek összehasonlíthatóak, majd illesztésre kerülnek a gépi tanuláson alapuló módszerek, melyek a probléma klasszifikációs mivolta miatt a klasszikus logisztikus regresszió, valamint az összetettebb random forest, illetve XGBoost módszerek. A fejezetet a GAT modell részletes leírása, illetve az eredmények összefoglalása zárja.

2.3.1. Elemzés keretrendszere

Annak érdekében, hogy a különböző modellek összehasonlíthatóak legyenek, egy egységes értékelési keretrendszer került felállításra, melynek elemei a gépi tanulási alapokon nyugvó modelleken, illetve a neurális hálón is alkalmazhatóak. Mivel a kutatási kérdés egy bináris klasszifikációs probléma, orvosi környezetben, ezért a klasszifikációs problémákra alkalmazható különböző teljesítmény-mutatók mellett a modellek interpretálhatóságára is kiemelt fontosságú. A szakdolgozat azonban elsősorban a vizsgált problémát statisztikai oldalról közelíti meg, így a modellek mélyebb, változó szintig lehatoló elemzése kizárólag csak statisztikai összehasonlítási és ábrázolási céllal történik meg, az eredmények klinikai interpretációja nem része a szakdolgozatnak.

Az összes modell a 2.1 fejezetben bemutatott, csökkentett változótartalmú adatbázison került futtatásra. Elsőnek az adatbázis tanító, illetve teszt adathalmazra lett bontva, 75-25 arányban. Ez a folyamat egyszer történt meg, fix kiindulási ponttal (*seed*), így az összes modell ugyanazon a tanító részhalmazon került tanításra, illetve teszt részhalmazon kiértékelésre. Az összes modell esetén a kimeneti érték egy valószínűséggé konvertálható mérték, ahol a két kategória közötti küszöbértéknek 0.5 lett választva. A különböző modellek egyedi paramétereinek finomhangolását a különálló fejezetek részletezik, ebben a szekcióban kizárólag a minden modellre egységesen alkalmazott, teszt adathalmazon történő kiértékelés módszertana kerül bemutatásra.

A bináris klasszifikációs módszerek teljesítményének elemzésére a legyakrabban használt kiindulási módszer a **tévesztési (*confusion*) mátrix** alkalmazása, ami összehasonlítja a modell által képzett besorolásokat a valós állapotokkal, az alábbi kategóriákat megkülönböztetve:

- **valós pozitív (TP):** a modell súlyos lefolyást jelez előre, és a beteg esetén valóban súlyos volt a lefolyás
- **valós negatív (TN):** a modell nem súlyos lefolyást jelez előre, és a beteg esetén valóban nem súlyos volt a lefolyás
- **hamis pozitív (FP):** a modell súlyos lefolyást jelez előre, de a beteg esetén valójában nem volt súlyos a lefolyás
- **hamis negatív (FN):** a modell nem súlyos lefolyást jelez előre, de a beteg esetén valójában súlyos volt a lefolyás

Az egészségügyi előrejelzési modellek és a tesztek esetén a különböző kategóriák eltérő fontosságot kaphatnak, azonban a jelenlegi kutatási probléma esetén megállapítható, hogy a hamis negatív arányra (téves negatív döntés, statisztikai értelemben *Type II* hiba) kiemelt figyelmet kell fordítani, hiszen a páciens hibás negatív diagnózisa potenciálisan életveszélyes állapothoz vezethet abban az esetben, ha a szükséges kezelést a beteg emiatt később kapja meg. Természetesen a hamis pozitív arány (*Type I* hiba) sem elhanyagolható, hiszen ez a beteg számára jelentős kellemetlenséget és ijedelmet tud okozni, azonban az optimális egyensúly megtalálásakor a mérleg nyelve kevésbé ebbe az irányba dől.

A tévesztési mátrix átfogó képet tud adni a klasszifikációs modell teljesítményéről, azonban gyakran hasznos a teljesítményt egy értékkel kifejezni, amely érték alapján a modell teljesítménye finomhangolható is. A legáltalánosabb ilyen mérték a **pontosság** (*accuracy*), ami összeveti a helyesen besorolt eseteket az összes esettel:

$$\text{Pontosság} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Az előző bekezdés alapján kiemelten fontos az a tény is, hogy mennyire megbízhatóan képes a modell a valós pozitív esetek azonosítására, így vizsgálandó az **érzékenység** (*recall*) mérőszám is, ami a modell által helyesen pozitívnak minősített esetek és az összes mintában szereplő pozitív esetek arányát vizsgálja. A teljesítmény optimalizálása azonban nem szabad, hogy teljes mértékben az érzékenységen alapuljon, hiszen az könnyen a hamis pozitívok arányának a növekedésével járhat együtt.

Ennek következményeként a **precizitás** (*precision*) mérték is vizsgálandó, ami azt vizsgálja, hogy a modell által súlyosnak minősített esetek közül mennyi a valóban súlyos, bevonva ezzel a hamis pozitív eseteket is.

$$\text{Érzékenység} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Precizitás} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Az egészségügyi kutatásokban gyakran kihívást jelent a kiegyensúlyozatlan eloszlás, hiszen számos probléma esetén a negatív – vagyis betegségnek vagy kezelésnek ki nem tett – esetek jelentősen felül tudják múlni a pozitív kimenetelek számát. A jelenlegi adatbázisban is tapasztalható egy enyhén kiegyensúlyozatlan eloszlás (878 negatív, 306 pozitív eset), így a fő teljesítmény mértéknek érdemes egy olyan mérték választani, mely figyelembe veszi a kiegyensúlyozatlan eloszlást.

Kiegyensúlyozatlan adatok esetén a sima pontosság mérték jelentős mértékben félrevezető eredményt tud adni, hiszen a képlet alapján a modell akkor is jó eredményt tud elérni, ha az összes eset vonatkozásában egyedül csak a többségben lévő osztályt jelzi előre. Ennek a kiküszöbölésére a modell tesztadatokon vett teljesítményének megállapítására a kiegyensúlyozott pontosság került alkalmazásra, elkerülve ezzel a nagyobb létszámú csoport torzító hatását. A mérték a pozitív esetek helyes felismerésének arányának (érzékenység) és a negatív esetek helyes felismerésének arányának (specifikusság) az átlagát veszi, így a két osztályra vonatkozó teljesítményt külön-külön, azonos súllyal veszi figyelembe.

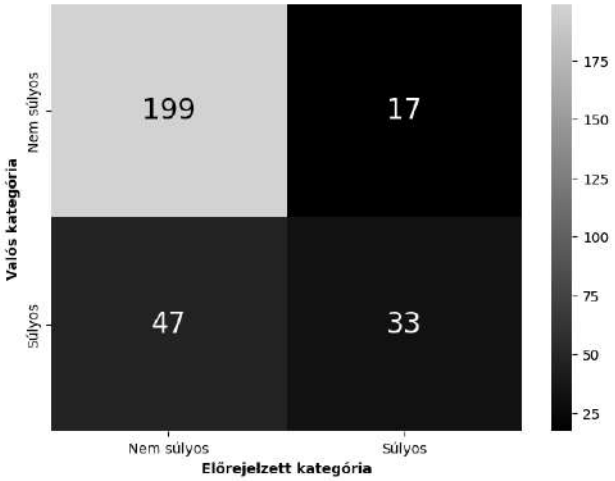
$$\text{Kiegyensúlyozott pontosság} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}}\right)$$

A különböző benchmark modellek illesztése esetén – a fix keretrendszerben történő összehasonlíthatóság érdekében – nem történt külön-külön változószelekciós eljárás, az összes modell ugyanazon a változókészleten lett tanítva, szükség szerint regularizációval. E döntés mögött a fő indok a GAT modell architektúrájában rejlik, mely a csúcsok közötti figyelem-koefficiensek hozzárendelésével elvégez egy bizonyos szintű változó (figyelem) szelekciót. Ennek következtében a GAT modell esetén az összes változóra szükséges volt ahhoz, hogy a dolgozat megmutassa azt, hogy a figyelem-mechanizmus milyen mértékben képes a releváns kapcsolatok feltérképezésére.

2.3.2. Benchmark modellek – logisztikus regresszió

Az első benchmark modell a bináris klasszifikációs kérdések esetén leggyakrabban alkalmazott logisztikus regresszió, ami a súlyos kategória bekövetkezési valószínűségét modellezi. A modell a tanító adathalmazon került illesztésre, ahol a kiegyensúlyozott pontosság 71,9%-ra adódott, míg a teszthalmazon ez az érték 66,69% lett.

A modell klasszifikációs teljesítményét az alábbi tévesztési mátrix foglalja össze:



9. ábra Tévesztési mátrix a teszt halmazon, logisztikus regresszió

A különböző modell-teljesítményt leíró mértékek az alábbiak szerint alakultak:

Általános pontosság	Érzékenység	Precizitás	Specifikusság	**Kiegyensúlyozott pontosság**
78,38%	41,25%	66,05%	92,13%	**66,69%**

3. táblázat Teljesítmény-mértékek teszt halmazon, logisztikus regresszió

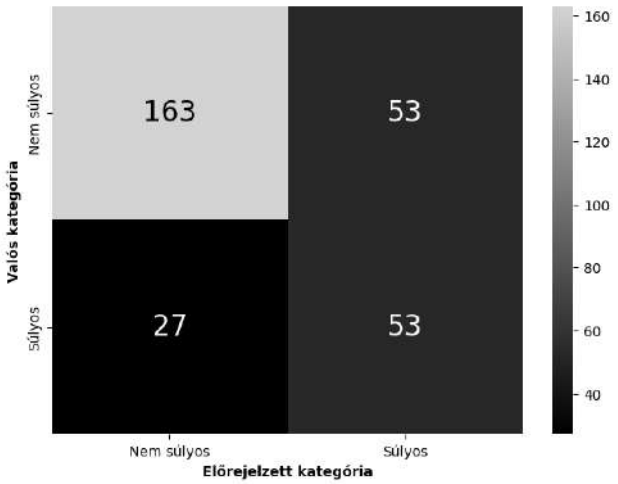
A logisztikus regresszión alapuló modell a célként kitűzött érzékenység, illetve kiegyensúlyozott pontosság tekintetében gyengén teljesített, a súlyos esetek mindössze 42%-át ismerte fel, ami különösen gyenge teljesítmény. A modell előnye, hogy a nem súlyos esetek 92%-át felismerte, ami alacsony hamis pozitív arányhoz vezetett. Ebből következik, hogy a kiegyensúlyozatlan osztályeloszlás okán használt kiegyensúlyozott pontosság 66,7%-ra tehető, azonban az alacsony érzékenység miatt a modellnek a konkrét problémára történő alkalmazása nem ajánlott.

2.3.3. Benchmark modellek – random forest

A logisztikus regresszió mellett a klasszifikációs problémákra általánosságba jó eredménnyel alkalmazható döntési fákon alapuló modellek is illesztésre kerültek, melyek közül az első modell a random forest algoritmus volt. A random forest a több, egyszerű döntési fákat és *bootstrap* mintavételezést egyesítő *bagging* módszerek egyik alesete, ahol a modellben szereplő fák csak az eredeti magyarázó változókat tartalmazó halmaz egy véletlenszerűen kiválasztott részhalmazán kerülnek illesztésre. (James et al., 2023)

A modell az alapparaméterek használatával jelentős mértékben túltanult a vizsgált adatbázison, tökéletes klasszifikációs pontossággot elérve a tanítóhalmazon, míg mindössze 60%-os kiegyensúlyozott pontosságot elérve a teszthalmazon. A túltanulás elkerülése érdekében a paraméterek hangolására volt szükség, ami keresztvalidációval, a kiegyensúlyozott pontosság metrika alapján történt. A túlillesztés szempontjából kritikus paraméterek a fáknak a maximális mélysége, a végső levek minimális elemszáma, illetve a belső csomópontok minimális, szétválasztáshoz szükséges elemszáma. A túlillesztés elkerülése érdekében a fa vissza lett vágva 5 mélységig, illetve a robosztus és általánosítható eredmény érdekében a minimális végső levél elemszáma megnövelésre került 30-ra, valamint minimum 50 elemszám volt szükséges egy vágáshoz. A modell 250 fával dolgozott, Gini kritérium alapján, az összes magyarázó változó négyzetgyökének számosságával megegyező véletlen változóhalmazon. Az így létrejött modell tanítóhalmazon vett kiegyensúlyozott pontossága 81%, míg teszthalmazon vett eredménye 70,85% lett, ami egy enyhén minimális, de elfogadható túltanulást jelent.

A modell klasszifikációs teljesítményét az alábbi tévesztési mátrix foglalja össze:



10. ábra Tévesztési mátrix a teszt halmazon, random forest

A különböző modell-teljesítményt leíró mértékek az alábbiak szerint alakultak:

Általános pontosság	Érzékenység	Precizitás	Specifikusság	**Kiegyensúlyozott pontosság**
72,97%	66,25%	50,00%	75,46%	**70,85%**

4. táblázat Teljesítmény-mértékek teszt halmazon, random forest

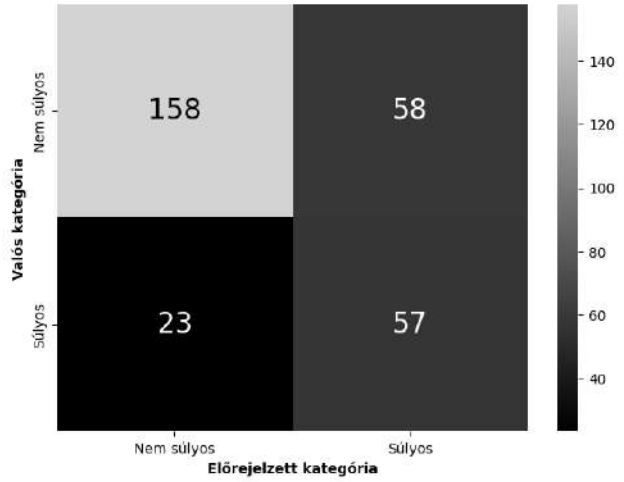
A random forest modell esetén az érzékenység jelentős mértékben nőtt a logisztikus regresszióhoz képest, azonban ez a növekedés a hamis pozitív esetek növekedésével járt együtt. Mivel a specifikusság míg így is elfogadható nagyságú, ezért kimondható, hogy a modell a probléma klinikai mivoltából kiindulva jobban teljesített, mint a logisztikus regresszió, amit a kiegyensúlyozott pontosság 70% körüli értéke is alátámaszt. Azonban a precizitás alacsony értéke arra utal, hogy sok a hamis pozitív érték is, ami túlkezeléshez és felesleges vizsgálatokhoz vezethet.

2.3.4. Benchmark modellek – XGBoost

Az utolsó benchmark modell a *boosting* modellek családjába tartozó XGBoost algoritmus, ahol a döntési fák szekvenciálisan, a korábbi fából kinyert információ segítségével, azoknak a reziduálisain kerülnek illesztésre. Ebből következik, hogy a fák nem egy bootstrap mintán kerülnek illesztésre, hanem mindig egy módosított mintán. A modell fontos tulajdonsága – a szekvenciális felépítésből adódóan – a λ tanulási paraméter, melynek alacsonyan tartásával elkerülhető a túlillesztés. (James et al., 2023)

A XGBoost esetén kiemelten fontos paraméterek finomhangolása. Az adatbázison az alapbeállítások itt is jelentős túlillesztést eredményeztek, így ennek a csökkentése jelentette az optimalizálás gerincét. A túltanulás elkerülése érdekében a tanulási paraméter csökkentésre került 0.005-re, míg a fák száma ennek megfelelően 750-re lett növelve, illetve a részminta halmaz aránya 70%-ra került beállításra. A fák mélysége az alulillesztés elkerülése érdekében 3-ra, míg a levelenkéni minimum mintaszám 20 elemre került beállításra. Az így létrejött modell tanítóhalmazon vett kiegyensúlyozott pontossága 79,9%, teszthalmazon vett teljesítménye pedig 72,2% lett, ami egy minimális, de elfogadható túltanulást jelent.

A modell klasszifikációs teljesítményét az alábbi tévesztési mátrix foglalja össze:



11. ábra Tévesztési mátrix a teszt halmazon, XGBoost

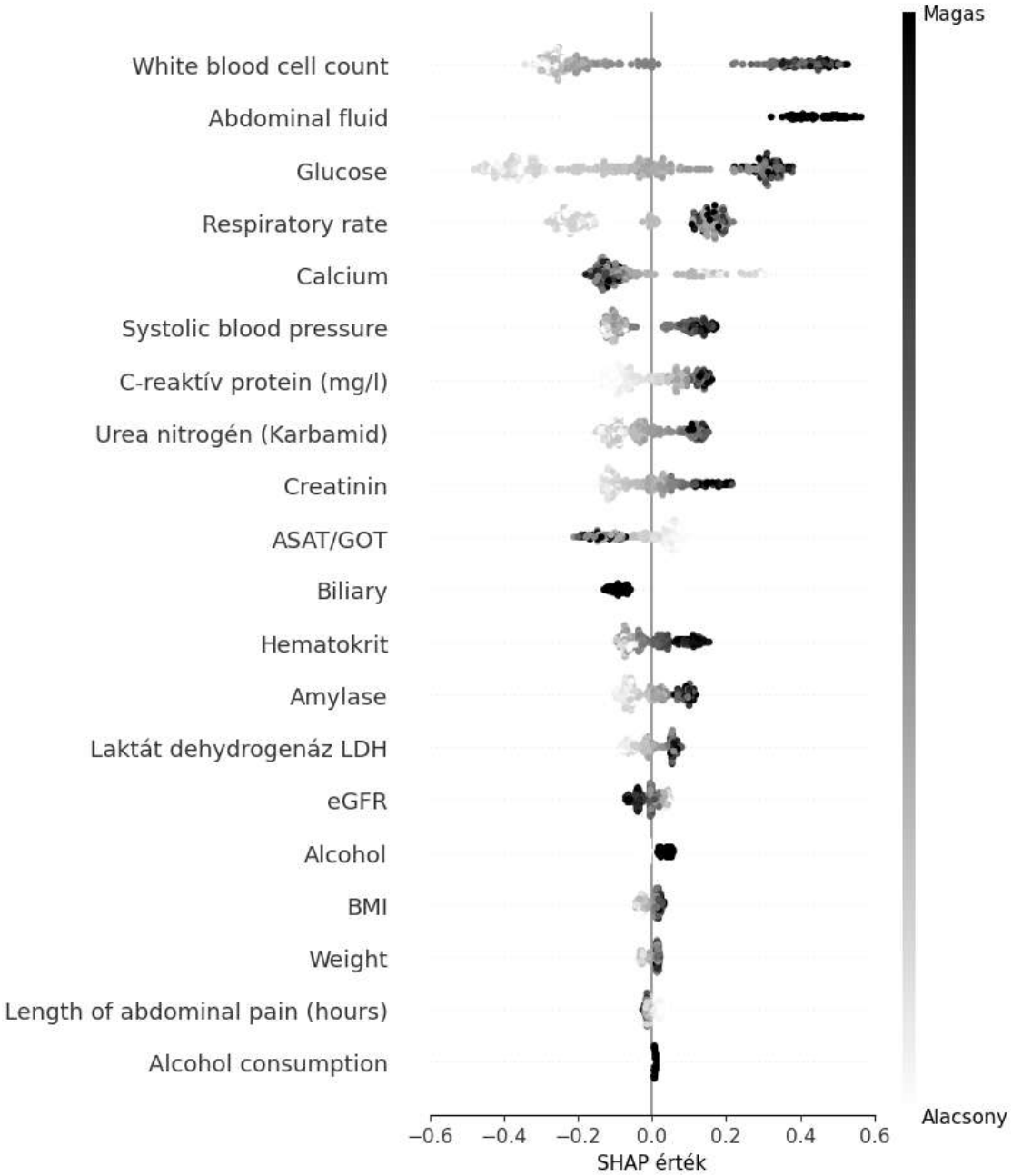
A különböző modell-teljesítményt leíró mértékek az alábbiak szerint alakultak:

Általános pontosság	Érzékenység	Precizitás	Specifikusság	**Kiegyensúlyozott pontosság**
72,64%	71,25%	49,57%	73,15%	**72,20%**

5. táblázat Teljesítmény-mértékek teszt halmazon, XGBoost

A három benchmark modell közül az XGBoost érte el a legjobb eredményt, legmagasabb érzékenységgel, illetve legmagasabb, 72% feletti kiegyensúlyozott pontossággal. A precizitás továbbra is alacsonynak bizonyult, azonban a feltevésnél kikötésre került, hogy a modell optimalizálása az érzékenységet és specifikusságot magába foglaló kiegyensúlyozott pontosságra fog történi, azzal a motivációval, hogy a valós pozitív esetek minél magasabb százalékú azonosítása jelenti a prioritást.

Mivel az XGBoost érte el a három benchmark modell közül a legjobb teljesítményt, így itt vizsgálatra került a változók fontossága a SHAP értékek alapján. A 12. ábrán látható a modell húsz, döntéshozatal szempontjából legfontosabb változója. Az ábráról az is leolvasható, hogy a változók értékeinek iránya milyen irányban befolyásolják a súlyossági előrejelzést, összehasonlítási alapot jelentve a GAT modell értékeléséhez.



12. ábra SHAP értékek, XGBoost

2.3.5. GAT neurális háló

A neurális hálók esetén számos, egymásra jelentős mértékben épülő paraméter áll rendelkezésre, melyeknek az adott problémára legmegfelelőbb teljesítményt nyújtó finomhangolása jelentős kihívást jelent. Ennek fényében a fejezetben részletesen bemutatásra kerül az, hogy a GAT neurális háló különböző paraméterei végül milyen beállítások mellett biztosították a legjobb teljesítményt, valamint milyen gondolatmenet, részeredmények, és az azokból következő döntések vezettek el a végleges modell felépítéséhez. Az optimalizálás vezérfonalát továbbra is a kiegyensúlyozott pontosság, illetve a túltanulás elkerülése jelentette. A modell tanítására és kiértékelésére használt részhalmazok megegyeztek a benchmark modellek esetén használt részhalmazokkal.

Az első eldöntendő kérdés a tanítóhalmaz batch méretének a meghatározása volt, mely méret megadja azt, hogy az algoritmus hány különböző beteghez tartozó gráfot használ fel egy tanítási lépés során. A végleges modell tanítási lépésenként egyszerre 48 – lépésenként különböző – beteg gráfját dolgozta fel, amiből a modell 48 bináris előrejelzést adott vissza, melyek alapján számolódott a veszteség, és történt meg a visszaterjesztés. Ez a batch méret a kipróbált változtatok közül az alacsonyabbak közé tartozott, ami zajosabb gradienshez és lassabb konvergenciához vezetett. Azonban az alacsonyabb batch méret által keltett zaj miatti nagyobb variancia hozzájárult ahhoz, hogy a gradiens ne szoruljon be egy lokális minimumba, illetve ne tanuljon túl a modell. A kipróbált batch méretek közül a magasabb értékek jellemzően túltanuláshoz vezettek, a modell túl biztosan haladt egy irányba, ami a teszt pontosság egy idő után történő stagnálásához, és a tanítóhalmazon vett pontosság folyamatos növekedéséhez vezetett.

Az optimalizálás a keresztentrópia veszteségfüggvény alapján történt, mely alkalmas a logit-alapú kimenettel rendelkező modellek kezelésére. A Pytorch csomagban elérhető keresztentrópia függvény automatikusan alkalmazza *softmax* aktivációt a kimeneti logitokra, így azokat az adott osztályba esés valószínűségévé alakítja, amely közvetlenül felhasználható a keresztentrópia klasszikus képletében. A keresztentrópia előnye, hogy érzékeny a hibás, de magabiztos előrejelzésekre, amely tulajdonság elősegíti a modell stabil konvergenciáját. Emellett a függvény könnyen bővíthető egy súlyozást elvégző taggal, mely alkalmazásával a kiegyensúlyozatlan eloszlásból fakadó torzítás ellensúlyozható. Mivel a használt klinikai adatbázis osztályeloszlása enyhén kiegyensúlyozatlan volt, így súlyozott keresztentrópia került alkalmazásra, ezzel is segítve a súlyos esetek megfelelő reprezentációját a tanulási folyamat során.

$$Veszteség \; = \; - \left[w_1 \; y \; log(p) \; + \; w_0 \; (1 \; - \; y) \; log(1 \; - \; p) \right]$$

Az előző két bekezdés összekötése a veszteségfüggvényt minimalizáló optimalizációs algoritmus kiválasztása. Mivel a tanítási folyamat kisebb elemszámú batchek alkalmazásával történt, így a veszteségfüggvény aktuális értéke és annak gradiense minden lépésben eltérő, kisebb elemszámú mintahalmazon alapult. Ez a tulajdonság zajos becslést eredményez a valódi – teljes adathalmazra vonatkozó – gradiensről, amit figyelembe kellet venni az optimalizálási algoritmus kiválasztásánál.

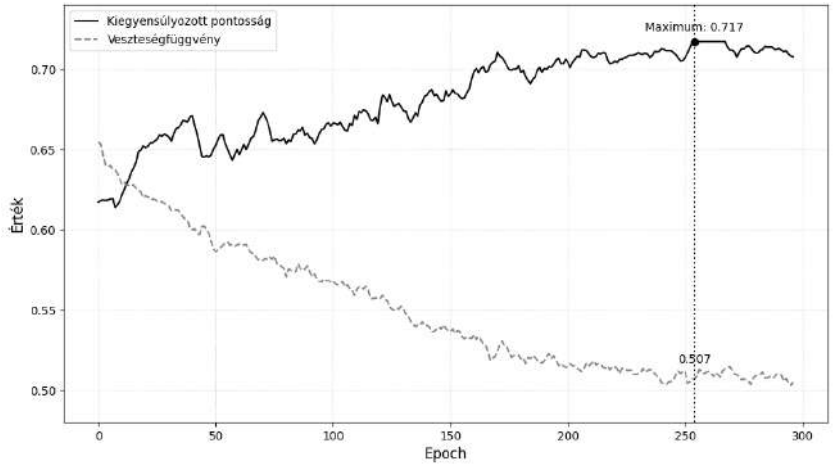
A modell tanítása során a sztochasztikus gradiens (SGD), illetve az adaptív, momentumom alapuló (Adam) módszerek kerültek alkalmazásra, azonban a végső választás az alacsony batch elemszám általi zajosabb adatokon jobb konvergálást biztosító Adam algoritmusra esett. A Kingma és Ba által (2017) bemutatott Adam algoritmus előnye, hogy a tanulási ráta (α) minden paraméterre adaptívan változik az első, illetve második momentum alkalmazásával. Az első momentum egy exponenciális mozgóátlagot (m) reprezentál, amely – figyelembe véve a korábbi lépéseket – kijelöli az irányt, míg a második momentum a gradiens négyzeteinek mozgóátlagát (v) képviseli, mely becslés ad a bizonytalanságra, ezzel skálázva és adaptívvá téve az optimalizációs folyamatot. Mindkét becslés esetén a mozgóátlagolás első elemei miatt szükséges a paraméterek korrigálása:

$$\theta_t \; = \; \theta_{t-1} \; - \; \alpha \; \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}} + \epsilon}$$

A tanulási ráta vonatkozásában a tapasztalati úton kialakuló leginkább megfelelő értéknek a 0.003 bizonyult. A kipróbált értékek a [0,01 – 0,001] intervallumból kerültek ki, ahol a 0.01 esetén a modell nem volt képes stabil konvergenciára. Ezzel szemben a 0.001 túl lassú tanulást eredményezett, és a validációs teljesítmény sem javult tovább, a modell egy nem optimális konfigurációban stabilizálódott. A középutat a 0.005 körüli értékek jelentették, azonban a 0.005 esetén még mindig hajlamos volt a modell túllépni az optimális irányon. Ezenfelül a tanítás során a fix tanulási paraméterek alkalmazásakor problémát jelentett az, hogy a tanulási folyamat végén a validációs pontosság jelentős fluktuációt mutatott, nem alakult ki stabil konvergencia.

E probléma áthidalására bevezetésre került egy tanulási rátát szabályozó időzítő, ami a koszinuszgörbe alakját követve az idő múlásával fokozatosan csökkentette a globális tanulási rátát. A módszer előnye, hogy a tanulási folyamat során kontroláltan csökken a ráta, ami elősegíti a végső minimum érték körüli stabilizálódást.

Az időzítő az alapértelmezetten is már adaptív Adam algoritmussal is hatékonyan működött együtt, hiszen az algoritmus a globálisan csökkenő tanulási ráta felhasználása mellett a különálló paraméterek tényleges lépésmértékeit a saját belső állapotuk alapján módosította. A módszernek létezik egy adott intervallumonként a tanulási rátát az eredeti értékre visszaállító alfajtája is, melynek a célja a lokális minimumhelyek elkerülése, azonban a vizsgált GAT modell esetén a kialakult gráfstruktúrán ez nem működött jól, a tanulási folyamat inkább finomítást igényelt. Ebből következően végső tanulási rátának 0.003 lett megválasztva, amit az időzítő lassan, 260 lépésben (*epoch*) csökkentett egészen 0.0001-ig, mely beállítás a futtatások alapján a legstabilabb tanulást és konvergenciát biztosította.



13. ábra Kiegyensúlyozott pontosság és veszteségfüggvény értékének alakulása

A GAT neurális háló architektúráját az alkalmazott rétegek száma, a rejtett reprezentációk dimenziója, valamint a figyelem-mechanizmus során alkalmazott párhuzamos fejek (*multi-head attention)* száma határozza meg, mely tulajdonságok befolyásolják a modell kapacitását és általánosító képességét, így finomhangolásuk kiemelten fontos az alulilleszkedés, illetve a túltanulás közötti optimális állapot elérésében. A modell összetettségét befolyásoló rétegek száma vonatkozásában az adott problémára 2 és 4 közötti rétegszámok lettek kipróbálva. A kétszintű modell ugyan gyors konvergenciát biztosított, azonban a validációs halmazon mért kiegyensúlyozott pontosság (az eddig, illetve lent részletezett paraméterek változatlansága mellett) alacsonynak bizonyult, 0.65 körüli értékkel, amiből arra lehetett következtetni, hogy a modell nem rendelkezett kellő komplexitással a mintázatok felimeréséhez és megtanulásához. A négyrétegű modell sem teljesített jobban, itt feltételezhetően az egyes csúcsok rejtett reprezentációi esetén felmerült a gráfalapú neurális hálókra jellemző túlsimítás (*over-smoothing*), amit a 7. ábrán is látható összefüggő komponensek okozhattak.

Ez azt eredményezte, hogy a negyedik lépés során már az aggregált információk túlságosan hasonló forrásból származhattak – ezzel hasonlóvá téve a csúcsok reprezentációit – ami ahhoz vezetett, hogy a modell nem volt képes kellőképpen differenciálni az osztályok között. Az optimális választást a középút jelentette, három réteg kellően összetettnek bizonyult a mintázatok megtanulásához, azonban még nem lépett fel információvesztés a túlsimításból fakadó esetleges túlzott aggregálás miatt, illetve nem lépett fel számottevő túltanulás sem.

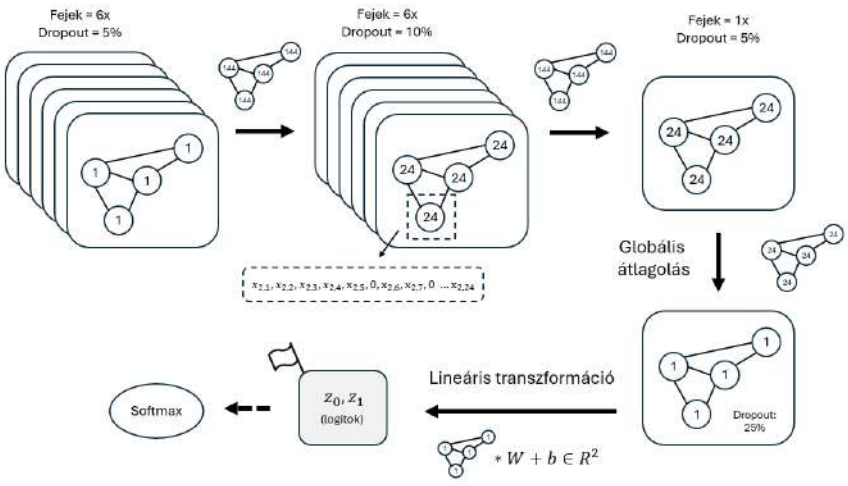
Az előzőekben részletezett három GAT réteg a gráf csúcsaira számítanak ki rejtett reprezentációkat, azonban a klasszifikációhoz gráfszintű reprezentációra van szükség. Ennek előállítását a modell egy globális átlagolással végzi, amely minden gráfra külön-külön kiszámítja a benne található csúcsok rejtett vektorainak átlagát. Az így kapott R^{24} dimenziós vektor minden gráfot egyetlen reprezentációként ír le, melyet egy lineáris transzformációs réteget bevonásával a modell 2 elemű, bináris klasszifikációra alkalmas kimenetté alakít át.

Az utolsó finomhangolható paraméterpár a rejtett reprezentációk dimenziója, illetve az ezzel a paraméterrel szorosan összefüggő párhuzamos fejek száma. A rejtett dimenziószám meghatározása is empirikus úton történt, ahol a futtatások során a magasabb dimenziószámok esetén (32 és afölött) túltanulás lépett fel, a tanítóhalmazon elért pontosság közel 100%-ra emelkedett, miközben a teszt halmazon mért kiegyensúlyozott pontosság 0.64–0.66 körül stagnált. Ebből az eredményből az következett, hogy alacsonyabb dimenzió választására van szükség, így elsőnek a dimenziószám lecsökkentésre került 16-ra. Ezen a szinten túltanulás már nem lépett fel, azonban a validációs halmazon elért teljesítmény is korlátozott volt, így szükséges volt a kapacitás további növelése. A középutat a 24 dimenziós reprezentáció jelentette, amely dimenzióméret már elegendő tanulási kapacitást biztosított az összefüggések megragadásához, túltanulásra mutató tendencia nélkül.

A GAT modellben alkalmazott párhuzamos fejek módszer célja, hogy stabilizálja a tanulási folyamatot, és több nézőpontból térképezze fel a csúcsok környezetét, amit a modell a fejenként különböző koefficiensek és transzformációs súlymátrixokat tanulásával ér el. A párhuzamos fejek által képzett reprezentációk az első és második rétegben összefűzésre kerülnek, így a modell teljesítményét jelentős mértékben befolyásolja az alkalmazott fejek száma.

A fejek száma a [4-10] intervallum értékei között kerültek kipróbálásra, ahol a legoptimálisabb választást a 6 fej alkalmazása jelentette. A hat fej még nem növelte meg olyan mértékben a paraméterek számát, hogy az túltanulást vagy zajt okozott volna, azonban kellően széleskörű kitekintést nyújtott, ezzel is hozzájárulva a folyamat stabilitásához, illetve a mintázatok magabiztosabb felismeréséhez.

Az előző két gondolati egység szintetizálást, illetve az egész modell végső hangolását a regularizációs paraméterek jelentették. Mivel az alacsonyabb batch méret már egy enyhe implicit regularizációt jelent, így egy erősebbre megválasztott regularizáció könnyen alultanuláshoz vezethetett volna, így a paraméterek finomhangolása során ezt a tényezőt figyelembe kellett venni. A végső modellben két különböző *dropout* technikán alapuló regularizáció került beállításra. Az első módszer a rétegekben fejtette ki hatását, a csúcsokat képviselő jellemzővektor bizonyos százalékú komponenseinek véletlenszerű eldobásával. Ez a regularizációs technika különösen erőteljes hatást tud kiváltani az első rétegben, ahol a bemeneti csúcsjellemző mindössze egy elemű (adott változóra vonatkozó érték), így itt egyfajta csúcsmaszkolást valósít meg. Ennek megfelelően az első és utolsó rétegben egy enyhe 5%-os, míg a második rétegben 10%-os *dropout* került beállításra. Ezenfelül a globális átlagolás után kialakult gráfszintű reprezentációt jelentő 24 elemű vektorra is alkalmazásra került a véletlenszerű eldobást megvalósító regularizáció, mely a komponensek 25%-át távolította el. Ez a folyamat azzal az előnnyel jár, hogy a végső osztályozó nem tanulhatja meg egy adott komponens kiemelten történő kezelését, hanem arra kényszerül, hogy a döntést több dimenzió alapján hozza meg, ezzel megakadályozva a túltanulást, és növelve az általánosíthatóságot.

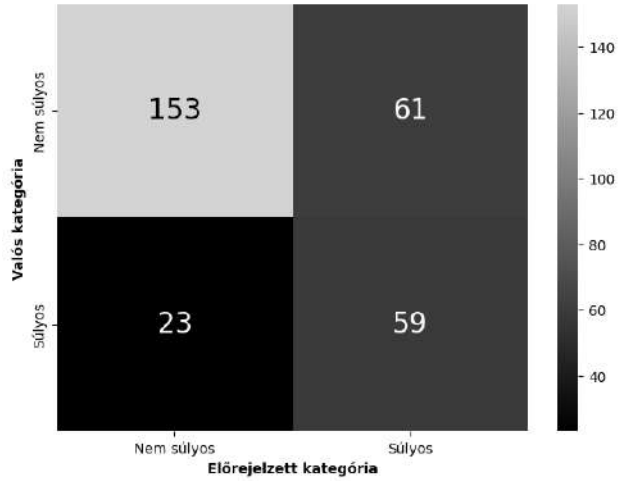


14. ábra Végleges GAT architektúra (saját szerkesztés)
/a csúcsok a jellemzővektor komponensszámát jelzik/

A modellezés során a GAT eredeti (Velickovic, 2018), illetve v2 (Brody, 2022) verziója is párhuzamosan feltanításra került, azonban a vizsgált adatbázison jelentős különbséget nem mutatott a két modell, így az eredeti GAT verzió került alkalmazásra.

A végső modell architektúrája így 3 GAT réteggel és két kiegészítő, összegző és lineáris transzformációt elvégző réteggel, 24 dimenziós rejtett reprezentációval és 6 párhuzamos fejjel rendelkezett. A túltanulás érdekében regularizáció került bevezetésre, az optimalizáció pedig az Adam algoritmus alapján történt, keresztentrópia veszteségfüggvény alapján, ütemezetten csökkentő, 0.003-as tanulási rátával.

Az így kialakult modell klasszifikációs teljesítményét az alábbi tévesztési mátrix foglalja össze:



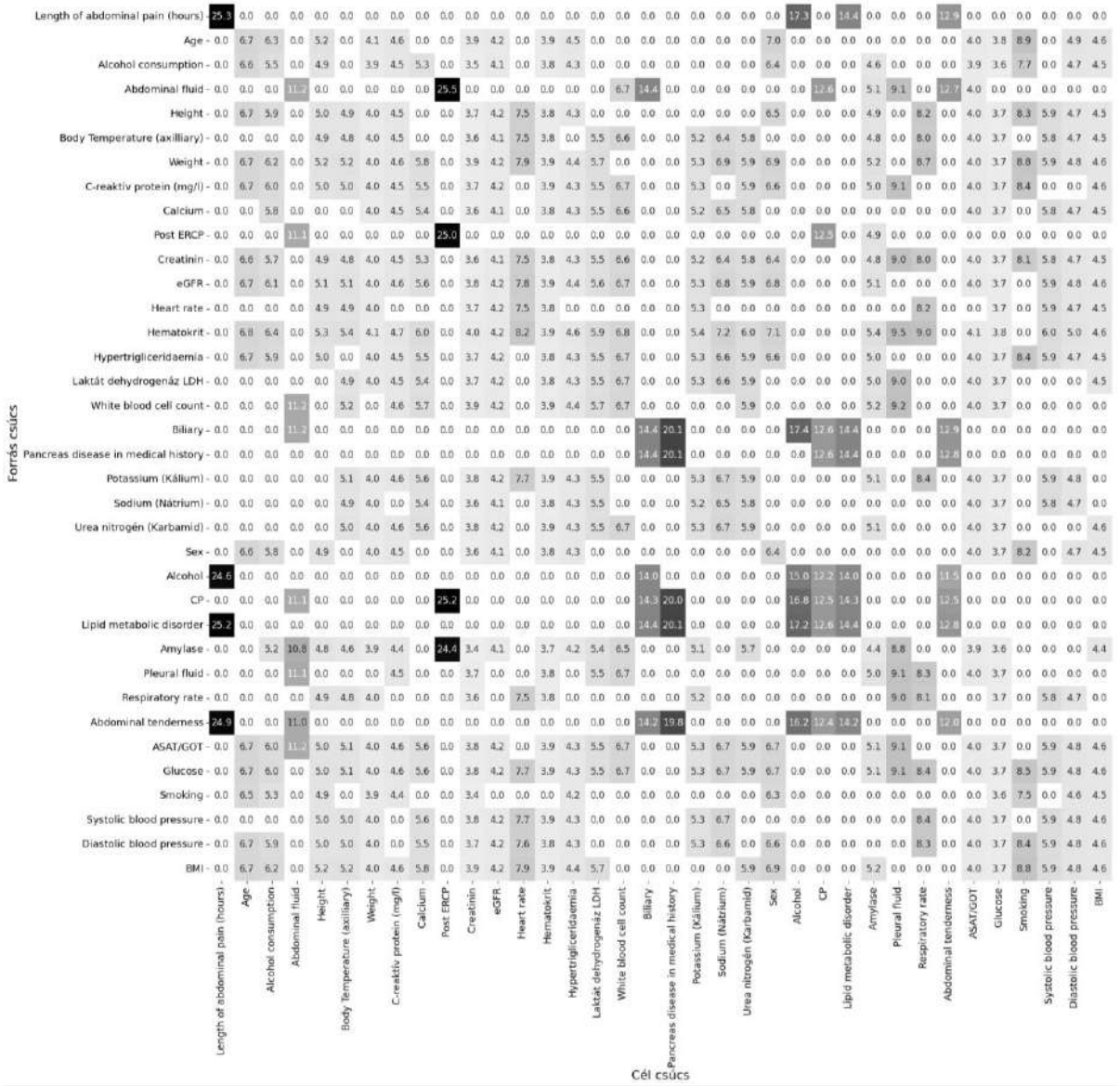
15. ábra Tévesztési mátrix a teszt halmazon, XGBoost

Az eredmények részletes elemzést és összevetését a következő fejezet tartalmazza, a különböző modell-teljesítményt leíró mértékek az alábbiak szerint alakultak:

Általános pontosság	Érzékenység	Precizitás	Specifikusság	**Kiegyensúlyozott pontosság**
71,60%	72,00%	49,20%	71,50%	**71,70%**

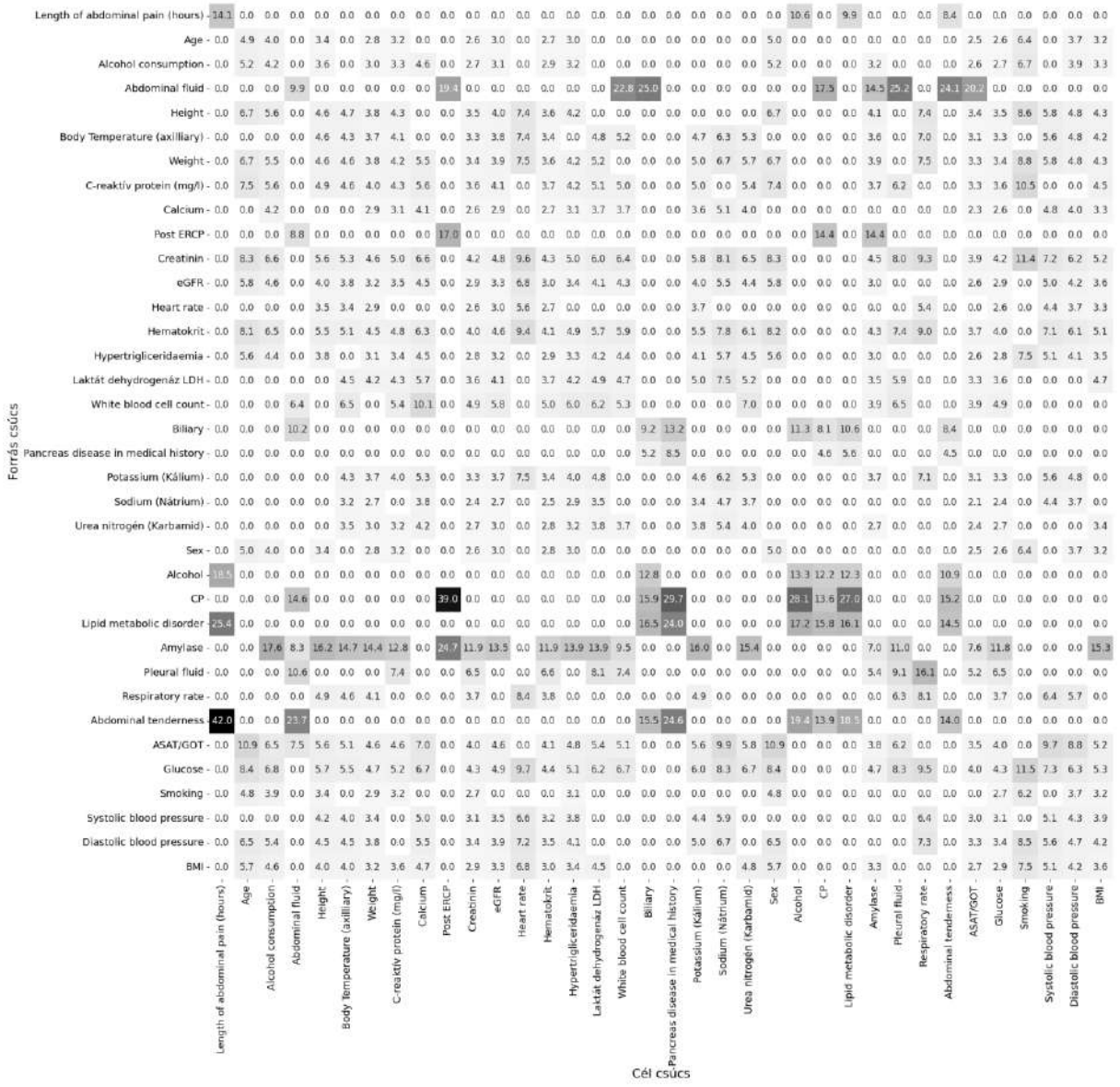
6. táblázat Teljesítmény-mértékek teszt halmazon, XGBoost

A GAT neurális háló egyik jelentős előnye a tanult figyelem-mechanizmus, mely a magyarázóerő növelése mellett az interpretációt is segíti. A 16. és 17. ábra a modell figyelemmátrixát ábrázolja, ahol a csúcsok közötti tanult koefficiens értékek vannak feltüntetve. A 16. ábrán a feltanított modell egyik teszt adaton kiértékelt első réteg, első feje által megállapított koefficiensei láthatók még a tanulási fázisban. Az ábrázolás előnye, hogy a mátrixból leolvasható az a tény, hogy a tanulási folyamat elején a modell mely kapcsolatoknak tulajdonított nagyobb figyelmet, amit össze lehet hasonlítani későbbi állapotokkal.



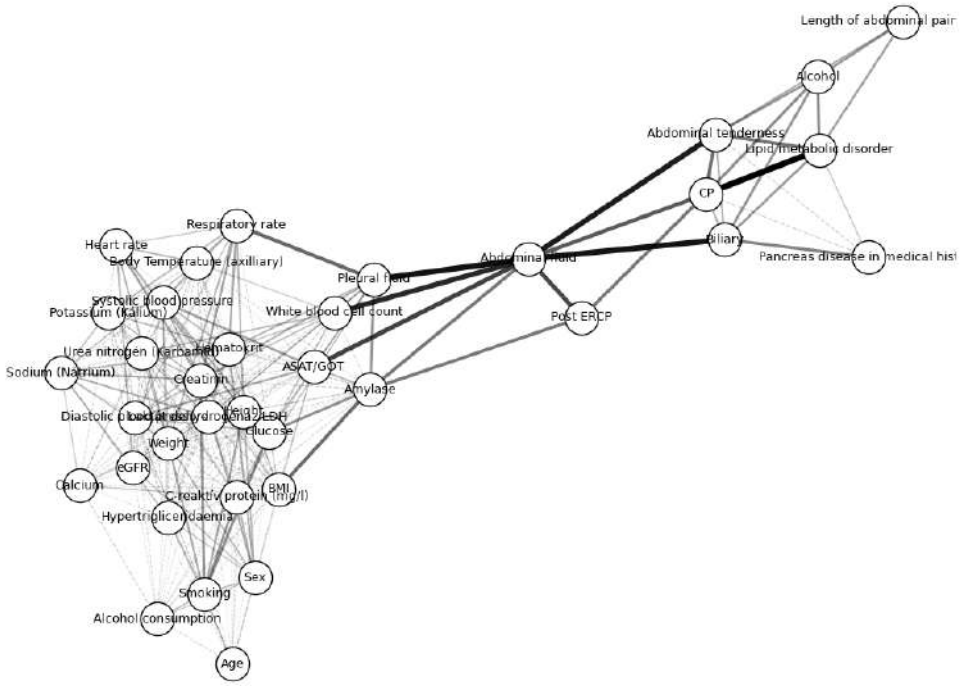
16. ábra Figyelemmátrix, 1. réteg, 1. fej, egy beteg (saját szerkesztés)
/ az értékek a könnyebb interpretálhatóság érdekében 100-as szorzóval lettek ellátva /

A 17. ábra már a végleges modell, összes betegen kiértékelt és átlagolt, utolsó réteg figyelem-értékeit tartalmazza, amely tanulmányozásakor látszódik, hogy a modell már sokkal kevésbé szór a különböző kapcsolatok fontosságának megítélésében, kialakultak az eredmény szempontjából domináns kapcsolatok. Fontos azonban megjegyezni a konkrét értékek interpretálhatósága szempontjából, hogy az utolsó rétegben szereplő csúcsok jellemzői már jelentősen eltérő, absztraktabb reprezentációi a bemeneti értékeknek, azonban a változók közötti kapcsolatok szemléltetésére alkalmas a lenti mátrix is.



17. ábra Figyelemmátrix, 3. réteg, 1. fej, összes teszt átlaga (saját szerkesztés)
/ az értékek a könnyebb interpretálhatóság érdekében 100-as szorzóval lettek ellátva /

A kialakult modell előrejelző képessége mellett a figyelemmátrix is jelentős értéket képvisel, hiszen az adatok alapján kialakuló kapcsolatfontosságokat klinikai szempontból is meg lehet vizsgálni, valamint alkalmas lehet akár – komplexebb modellek esetén – potenciális új változók közötti kapcsolatok felderítésére. A végső eredmények az eredeti gráf formájában is vizualizálhatóak, így a változók CUI vektortér alapján kialakult pozíciója is bevonható a vizsgálatba.



18. ábra Figyelem gráf, fontosság alapján (saját szerkesztés)

2.3.6. Eredmények összefoglalása és limitációk

A hasnyálmirigy-súlyosság előrejelzésére alkalmas bináris klasszifikációs problémára javasolt három benchmark modell és GAT neurális háló egy egységes keretrendszerben került kiértékelésre, ahol az általános pontosság mellett vizsgálatra került az érzékenység, illetve precizitás is, valamint a kiegyensúlyozatlan osztályeloszlásból fakadóan a kiegyensúlyozott pontosság is.

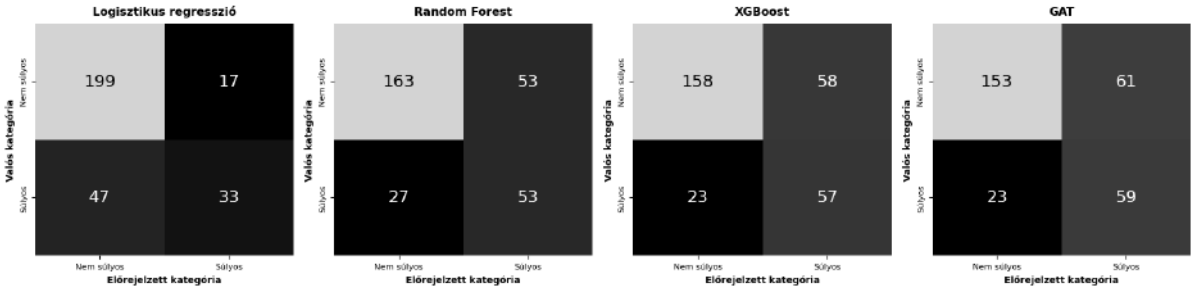
	Általános pontosság	Érzékenység	Precizitás	Specifikusság	**Kiegyensúlyozott pontosság**
Logisztikus regresszió	78,38%	41,25%	66,05%	92,13%	**66,69%**
Random forest	72,97%	66,25%	50,00%	75,46%	**70,85%**
XGBoost	72,64%	71,25%	49,57%	73,15%	**72,20%**
GAT	71,60%	72,00%	49,20%	71,50%	**71,70%**

7. táblázat Eredmények összefoglalása

A **logisztikus regresszió** a többi modell közül a legmagasabb általános pontosságot, illetve specifikusságot érte el, ugyanakkor a kiegyensúlyozatlan osztályeloszlás miatt a pusztán erre a két értékre támaszkodó modellértékelés félrevezető képet nyújt a modell teljesítményéről. A modell ugyan magas specifikusságot ért el, vagyis az enyhe kimeneteleket magas pontossággal ismerte fel, azonban ez a magabiztosság az érzékenység kárára alakult ki, hiszen a súlyos esetek közel 60%-át nem volt képes a modell azonosítani. E mögött a jelenség mögött az állhat, hogy a modell a lineáris természetéből fakadóan nem képes megragadni a vizsgált, klinikai eredetű változókra jellemző komplex, nemlineáris kapcsolatokat, így a jobb eredmény érdekében vagy kernelizálni szükséges a modellt, vagy egyéb, faalapú struktúrán alapuló gépi tanulási modellt kell alkalmazni.

A **random forest és XGBoost** módszerek alkalmasabbak a nemlineáris összefüggések pontosabb modellezésére, így a logisztikus regresszió után ezek a módszerek kerültek illesztésre. Mindkét modell esetében jelentős javulás volt megfigyelhető az érzékenység és a kiegyensúlyozott pontosság tekintetében, így a súlyos esetek felismerése lényegesen eredményesebbé vált. A modellek előnye, hogy a paramétereik dinamikusabban hangolhatóak, így lehetőség volt kimondottan a kiegyensúlyozott egyensúly mértéken alapuló finomhangolásra, amelyből fakadóan a kiegyensúlyozott pontosságot alkotó teljesítmény-

mértékek javultak, azonban ez a javulás a precizitás, azaz a hamis pozitívok arányának a növekedésével járt. A random forest modell általánosságban véve egy egyszerűbb és ezáltal robusztusabb, túltanulásra kevésbé hajlamos modell, míg az XGBoost algoritmus az egyik leghatékonyabb osztályozó algoritmus, ami a gradiens-alapú optimalizálásnak és különböző regularizációs és mintavételezési technikáknak köszönhetően képes komplex nemlineáris összefüggések felderítésére, valamint megfelelő finomhangolással a túltanulás elkerülésére.



19. ábra Eredmények összehasonlítása (tévesztési mátrix)

A **GAT neurális háló** teljesítménye közel azonos volt az XGBoost teljesítményével, ami arra utal, hogy a gráfalapú struktúrán alapuló neurális háló képes hasonló prediktív teljesítményt elérni, mint a legelterjedtebb gépi tanulási módszer. A végső kiegyensúlyozott pontosság 72% körülire adódott, ami kiegyensúlyozott mértékű érzékenység, illetve specifikusság mutatóból adódott. Ez az eredmény arra utal, hogy a modell mindkét osztályt – súlyos, illetve nem súlyos – hasonló pontossággal képes osztályozni. A precizitás a GAT modell esetén is alacsony, amiből az következik, hogy a modell alkalmazása esetén is magas a hamis pozitív arány. A magas arány potenciálisan klinikai túlkezeléshez is vezethet, azonban az elemzés elején kikötésre került, hogy az optimalizálás a kiegyensúlyozott pontosság, illetve érzékenység mentén fog történni. Ennek klinikai indoka, hogy a súlyos hasnyálmirigy-gyulladás korai felismerése döntő fontosságú a beteg ellátása szempontjából, míg a tévesen súlyosnak ítélt esetek többnyire csak fokozott megfigyelést vagy további vizsgálatokat vonnak maguk után. Fontos szempont még az is, hogy a klinikai ellátásban a statisztikai modellek jelenleg nem önálló diagnosztikai eszközként, hanem sokkal inkább döntéstámogató rendszerként funkcionálnak, így előnyösebb, ha a modell a nehezebben megállapítható, potenciálisan súlyos esetek esetén képes magasabb megbízhatóságra.

Az eredmények összehasonlítása kapcsán érdemes kitérni a GAT modell alkalmazhatóságának módszertani, illetve gyakorlati feltételeire, hiszen a tabuláris adatokon történő alkalmazásuk jelentősen komplexebb, mint a bemutatott benchmark gépi tanulási modellek esetén. Az alkalmazás előfeltétele, hogy a prediktív probléma olyan természetű legyen, ahol a különböző változók közötti kapcsolati információk is relevánsak lehetnek a tanulás szempontjából, azaz a jellemzők valós, többletinformációt nyújtó gráfalapú kapcsolati szerkezetbe rendezhetőek. Ezenfelül szükséges egy olyan metrikának a meghatározása is, amely alapján a gráf élei kialakíthatóak, és egy olyan küszöbérték, mely alkalmazásával a gráf összetettsége szabályozható.

A modell alkalmazása során különösen fontos a bemeneti jellemzők megfelelő skálázása, mivel a figyelem-mechanizmus érzékeny a változók nagyságrendjére. Eltérő skálájú változók esetén a nagyobb értékek dominálhatják a súlyokat, ami tanulási instabilitáshoz és pontatlansághoz vezethet. Emellett a hiányzó értékek kezelése is kiemelt jelentőségű, mivel a GAT rétegeiben az aggregáció a szomszédos csúcsokon keresztül történik, így a hiányzó értékek numerikus hibát vagy a reprezentáció torzulását okozhatják.

A gráfalapú neurális hálók több korláttal is rendelkeznek. A modellek esetén a gráf struktúrája jelentősen befolyásolja a teljesítményt, különösen abban az esetben, ha a jellemzők – mint például a CUI kódok – közötti kapcsolatok nem elég informatívak, vagy nem jól definiáltak. Ezenfelül a gráf sűrűségét befolyásoló küszöbérték megválasztására sincsen elfogadott érték, ez is egyfajta finomhangolható paraméterként fogható fel, amely empirikus módon állapítható meg. Ezenfelül a GAT modellre is érvényes a neurális hálók ellen gyakran felemlegetett feketedoboz jelenség. Habár a figyelemkoefficiensek valamiféle betekintést nyújtanak a tanulás folyamatába, ezek nem mindig korrelálnak a jellemzők valódi fontosságával, a rétegek után kialakuló csúcsreprezentációk pedig absztrakt formájúak, így az eredeti változókhoz való hozzárendelésük nehézkes. Az előző bekezdésben említettek alapján a GAT modell ezenfelül érzékeny az adatminőségre is, így kifinomultabb előfeldolgozási lépések alkalmazása szükséges. Mindezek mellett a számos hiperparaméter hangolása is erősen befolyásolja a modell teljesítményét, melyek hangolása nem triviális, és kis adatméret esetén könnyen túltanuláshoz is vezethet egy nem megfelelően paraméterezett modell.

3. Összefoglalás

Szakdolgozatom során a gráfalapú, figyelem-mechanizmuson alapuló neurális hálók alkalmazását vizsgáltam heveny hasnyálmirigy gyulladás súlyosságának előrejelzésére, szóbeágyazási, illetve tudásgráfon alapuló keretrendszer alkalmazásával.

A téma aktualitását az elmúlt évtizedben különös figyelemnek örvendő neurális hálón alapuló diagnosztikai eljárások jelentették, melyek klinikai kontextusban történő alkalmazása kiemelkedő eredményt ért el az összetett mintázatokon alapuló diagnosztikai előrejelzések és betegséglefolyások modellezésében. Ebből fakadóan arra a kérdésre kerestem a választ, hogy a gráfszerkezeten alapuló *Graph Attention Network* mélytanulási modell milyen teljesítménnyel képes a hasnyálmirigy-gyulladás súlyosságát előrejelző bináris klasszifikációra, és ezek az eredmények hogyan viszonyulnak a klasszikus gépi tanulási algoritmusok eredményeihez.

Az elemzés elvégzése előtt pontosan definiáltam a kutatási kérdést, és általánosan bemutattam a hasnyálmirigy-gyulladás etiológiai okait és regionális mintázatait, ahol arra a megállapításra jutottam, hogy a kelet-európai régióban a betegség különösen nagy terhet jelent az ellátórendszer számára. A probléma kontextusba helyezése érdekében bemutattam a korábbi előrejelzési módszerek kialakítására vonatkozó kutatásokat, valamint a problémafelvetést és adatbázist nyújtó EASY-APP modellt, mely kutatás a tabuláris jellegű adattömeg okán elvetette a neurális hálók alkalmazásának lehetőségét. Ebből fakadóan egy olyan keretrendszert vázoltam fel, ami az adatbázis változóhalmazát egy magas-dimenziós szóbeágyazási modellen alapuló vektortérbe helyezi el, majd ebből a vektortérből egy tudásgráfot alakít ki, amely ezáltal képes egy tabuláris adatbázist átalakítani a neurális háló bemeneteként használható gráf struktúrába.

A *Graph Attention Network (GAT)* módszertanának részletes bemutatása előtt a módszer kontextusba helyezése érdekében kitértem a gráfalapú neurális hálózatok általános felépítésére és fajtáira, illetve a gráfokra vonatkozó alapvető fogalmakra. A fogalmak tisztázása után részletesen bemutattam a GAT háló középpontjában álló csúcsok közötti figyelem koefficiens meghatározásának algoritmusát, illetve az architektúra általános felépítését. Az eredeti GAT neurális háló mellett kitértem a dinamikus súlyozást használó továbbfejlesztett GATv2 variánsra is, bemutatva a két modell közötti módszertani különbséget. Az architektúrák bemutatása után részletesen kitértem az alkalmazhatóság feltételeire, illetve a modell előnyeire és hátrányaira.

Az elméleti áttekintés után a módszer alkalmazására tértem át, ahol részletesen kifejtettem az adott kutatási problémára vonatkozó optimális felépítést, kitérve az adatbázis változóinak CUI vektortérben történő megfeleltetésére, az optimális küszöbértékre, illetve a véglegesen kialakuló gráf szerkezetére. Az előfeldolgozási lépések során bemutattam az adathiányok kezelését, illetve az alkalmazott imputálás lépéseit, ahol arra a megállapításra jutottam, hogy a kimeneti súlyosság enyhén kiegyensúlyozatlan eloszlást mutat, így az értékelési keretrendszer ennek megfelelően került kialakításra.

A neurális háló eredményeinek kontextusba helyezése érdekében az adatbázison gépi tanulási modelleket is illesztettem, így a GAT eredményeit a logisztikus regresszió, random forest, illetve XGBoost algoritmusokkal hasonlítottam össze. Kiemelten fontosnak tartottam egy minden modellre alkalmazható egységes értékelési keretrendszer felépítését, így minden esetben a tévesztési mátrix különböző elemeiből származtatható pontosságot, érzékenységet és precizitást vizsgáltam. Mivel az egészségügyi problémákra gyakran jellemző kiegyensúlyozatlan eloszlás a vizsgált adatbázisban is jelen volt, ezért az optimalizálás a kiegyensúlyozott pontosság mentén történt az összes modell esetén.

A dolgozat célként tűzte ki a neurális hálókra jellemző feketedoboz jelenség csökkentését, így részletesen kitértem az adott probléma vonatkozásában legoptimálisabbnak bizonyuló paraméterbeállításokra, illetve az ezekhez a beállításokhoz vezető gondolatmenetre és döntési pontokra. A GAT eredményeinek interpretálhatóságát különböző vizualizációkkal segítettem, azonban a dolgozatnak nem volt célja az eredmények klinikai értelmezése.

Az adatbázison történő alkalmazás rávilágított arra, hogy a GAT neurális háló képes elérni a jelenleg legnépszerűbbnek számító XGBoost gépi tanulási algoritmus teljesítményét, így megalapozottnak tekinthető az alkalmazása a kutatási probléma vonatkozásában. Ezenfelül a dolgozatban bemutatott, szóbeágyazási és tudásgráfon alapuló keretrendszer egy olyan rugalmas és könnyen adaptálható megközelítést képvisel, amely nemcsak a heveny hasnyálmirigy-gyulladás súlyosságának előrejelzésére alkalmas, hanem más, hasonló jellegű klasszifikációs problémák esetén is alkalmazható. Mindezek alapján összegzésként elmondható, hogy a dolgozatban részletesen ismertetett GAT modell és keretrendszer együttesen egy ígéretes és széles körben alkalmazható alternatívát jelentenek az elterjedt klasszikus gépi tanulási megközelítésekkel szemben.

Felhasznált irodalom

Amerikai Egyesült Államok Orvostudományi Nemzetközi Könyvtára. (2023, február 9.) *Acute Pancreatitis.* https://www.ncbi.nlm.nih.gov/books/NBK482468/

Banks, P. A., Bollen, T. L., Dervenis, C., Gooszen, H. G., Johnson, C. D., Sarr, M. G., Tsiotos, G. G., & Vege, S. S. (2013). Classification of acute pancreatitis—2012: Revision of the Atlanta classification and definitions by international consensus. *Gut*, *62*(1), 102–111. https://doi.org/10.1136/gutjnl-2012-302779

Beam, A. L., Kompa, B., Schmaltz, A., Fried, I., Weber, G., Palmer, N. P., Shi, X., Cai, T., & Kohane, I. S. (2019). *Clinical Concept Embeddings Learned from Massive Sources of Multimodal Medical Data*. https://doi.org/10.48550/arXiv.1804.01486

Brody, S., Alon, U., & Yahav, E. (2022). *How Attentive are Graph Attention Networks?*. https://doi.org/10.48550/arXiv.2105.14491

Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. (2021). *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. https://doi.org/10.48550/ARXIV.2104.13478

Harris, Z. S. (1954). Distributional Structure. WORD, *10*(2–3), 146–162. https://doi.org/10.1080/00437956.1954.11659520

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An Introduction to Statistical Learning: With Applications in Python*. Springer International Publishing. https://doi.org/10.1007/978-3-031-38747-0

Kingma, D. P., & Ba, J. (2017). *Adam: A Method for Stochastic Optimization*. https://doi.org/10.48550/arXiv.1412.6980

Kipf, T. N., & Welling, M. (2017). *Semi-Supervised Classification with Graph Convolutional Networks*. https://doi.org/10.48550/arXiv.1609.02907

Kui, B., Pintér, J., Molontay, R., Nagy, M., Farkas, N., Gede, N., Vincze, Á., Bajor, J., Gódi, S., Czimmer, J., Szabó, I., Illés, A., Sarlós, P., Hágendorn, R., Pár, G., Papp, M., Vitális, Z., Kovács, G., Fehér, E., … Hungarian Pancreatic Study Group. (2022). EASY-APP: An artificial intelligence model and application for early and easy prediction of severity in acute pancreatitis. *Clinical and Translational Medicine*, *12*(6), e842. https://doi.org/10.1002/ctm2.842

Li, C.-L., Jiang, M., Pan, C.-Q., Li, J., & Xu, L.-G. (2021). The global, regional, and national burden of acute pancreatitis in 204 countries and territories, 1990-2019. *BMC Gastroenterology*, *21*(1), 332. https://doi.org/10.1186/s12876-021-01906-2

Longa, A. (2021, március 5.). *Graph attention networks (GAT).* https://antoniolonga.github.io/Pytorch_geometric_tutorials/posts/post3.html

Roberts, S. E., Morrison-Rees, S., John, A., Williams, J. G., Brown, T. H., & Samuel, D. G. (2017). The incidence and aetiology of acute pancreatitis across Europe. *Pancreatology*, *17*(2), 155–165. https://doi.org/10.1016/j.pan.2017.01.005

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, *15*(1), 1929–1958.

U.S. National Library of Medicine (NIH). (n.d.). *UMLS Terminology Services*. Letöltve 2025. március 08. https://uts.nlm.nih.gov/uts/umls/home

Veličković, P. (2023). Everything is connected: Graph neural networks. *Current Opinion in Structural Biology*, *79*, 102538. https://doi.org/10.1016/j.sbi.2023.102538

Veličković, P. (n.d.). *Graph Attention Networks.* Letöltve 2025. március 03. https://petar-v.com/GAT/

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). *Graph Attention Networks*. https://doi.org/10.48550/arXiv.1710.10903

Vrahatis, A. G., Lazaros, K., & Kotsiantis, S. (2024). Graph Attention Networks: A Comprehensive Review of Methods and Applications. *Future Internet*, *16*(9), 318. https://doi.org/10.3390/fi16090318

Winge, E. (2018). *Word embedding models as graphs: Conversion and evaluation*, University of Oslo https://www.duo.uio.no/bitstream/handle/10852/66494/1/Winge-thesis.pdf

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2019). *A Comprehensive Survey on Graph Neural Networks*. https://doi.org/10.48550/ARXIV.1901.00596