Neo4j gráf adatbázis alkalmazása hálózati adatok elemzésére

Application of Neo4j graph database for network data analysis

FERENCZ Katalin[1]PhD hallgató, RIGÓ Ernő[1,2]PhD hallgató, dr. DOMOKOS József[3] egyetemi docens, dr. MOLNÁR László[3]egyetemi adjunktus

[1]Óbudai Egyetem, Alkalmazott Informatika és Alkalmazott Matematika Doktori Iskola, 1034 Budapest, Bécsi út 96/b, tel. +36-1-6665544, ferenczkatalin@yahoo.com
[2]HUN-REN Számítástechnikai és Automatizálási Kutatóintézet (SZTAKI), 1111 Budapest, Kende utca 13-17, tel. +36-1-2796000, rigo.erno@sztaki.hun-ren.hu
[3]Sapientia EMTE Marosvásárhelyi Kar, Villamosmérnöki tanszék, Koronka, 2 szám, tel. +4 0265 206 210, fax +4 0265 206 211, domi@ms.sapientia.ro, laszlo.molnar@ms.sapientia.ro

Abstract

The analysis of network data has become essential with the rise of digitalization and the Internet of Things (IoT). Traditional relational databases face limitations in handling the complexity of modern networks due to their rigid tabular structures and computational constraints. In contrast, graph databases provide a more adaptable approach by representing data relationships through nodes and edges. This article explores how the Neo4j graph database improves data analysis in complex network environments, particularly within IoT systems. By enabling faster anomaly detection, pattern recognition, and the development of predictive models, Neo4j enhances decision-making and forecasting processes. The article highlights the benefits of graph databases, including their ability to execute fast, flexible queries and efficiently manage intricate relationships. Real-world applications of Neo4j are presented, showcasing its effectiveness in deriving insights from network data using various algorithms.

Keywords: network data analysis, graph database, Neo4j, anomaly detection, clustering

Kivonat

A hálózati adatok elemzése a digitalizáció és az Internet of Things (IoT) térnyerésével elengedhetetlenné vált. A hagyományos relációs adatbázisok korlátokba ütköznek a modern hálózatok komplexitásának kezelésében merev táblázatos szerkezetük és számítási korlátjaik miatt. Ezzel szemben a gráf-adatbázisok rugalmasabb megközelítést kínálnak az adatok közötti kapcsolatok csomópontok és élek általi modellezésével. Ez a cikk bemutatja, hogyan javítja a Neo4j gráf-adatbázis a hálózatokban végzett adatelemzést, különösen az IoT rendszerekben. A gyorsabb anomáliaérzékelés, a mintafelismerés és a fejlett prediktív modellek fejlesztése révén a Neo4j hozzájárul a döntéshozatali és előrejelzési folyamatok javításához. A cikk kiemeli a gráf-adatbázisok előnyeit, beleértve a gyors és rugalmas lekérdezések végrehajtását, valamint a bonyolult kapcsolatok hatékony kezelését. Valós példákon keresztül mutatja be a Neo4j alkalmazását, különböző algoritmusok segítségével tárva fel a hálózati adatok feldolgozási lehetőségeit.

Kulcsszavak: hálózati adatelemzés, gráf adatbázisok, Neo4j, anomália detektálás, klaszterezés

1. BEVEZETŐ

A hálózati adatok elemzése egyre fontosabbá válik az IoT (Internet of Things) és a digitalizáció előrehaladtával. A modern hálózatok összetett kapcsolatok és folyamatos adatáramlások révén működnek, ezért az ilyen adatok értelmezése kihívást jelent. A „hagyományos” relációs adatbázisok által használt módszerek nehezen kezelik a hálózati adatok közötti komplex kapcsolatokat, mivel ezek táblákban tárolják az adatokat, és az elemzések megvalósítása során nagy számítási kapacitást igényelhetnek, különösen nagy adatforrások esetén.

A nagy adatkészleteket tartalmazó rendszerek által integrált relációs adatbázisok használata esetében felmerülő problémák kiküszöbölése érdekében az idősorokként tárolt adatok gráf struktúrákká való alakítása egyre inkább értékes megvalósításnak tűnik. Ez annak köszönhető, hogy ennek az adatformának az alkalmazása sok előnyt jelenthet az elemzések és adatfeldolgozási feladatok végrehajtása szempontjából. A gráf adatbázisok, mint például a Neo4j [10], rugalmasabb megközelítést kínálnak, mivel csomópontok

(node-ok) és élek (kapcsolatok) formájában kezelik az adatokat. Ez lehetővé teszi a hálózatokban lévő kapcsolatok gyorsabb és hatékonyabb lekérdezését és elemzését, különösen akkor, ha összetett kapcsolati struktúrák és nagy mennyiségű adat van jelen. Ezzel az új típusú elemzési módszerrel könnyebben észlelhetők az anomáliák, azonosíthatók a mintázatok, és fejlettebb prediktív modellek hozhatók létre, amelyek javítják a döntéshozatalt és az előrejelzést.

2. GRÁFADATBÁZISOK ALKALMAZÁSA ADATELEMZÉSHEZ

A gráfadatbázisok alkalmazása az adatelemzés területén egyre nagyobb népszerűségre tesz szert, különösen a komplex kapcsolati hálózatok elemzésekor, ahol a hagyományos adatbázisok hatékonysága korlátokba ütközik. A hagyományos relációs adatbázisok táblaalapú szerkezetük miatt nehezen kezelik a bonyolult összefüggéseket és az adatpontok közötti kapcsolati hálókat. Az ilyen struktúrák elemzése komoly számítási kapacitást és bonyolult lekérdezéseket igényel, amelyek gyakran lassítják az adatok feldolgozását [9].

A gráfadatbázisok, mint a Neo4j, képesek intuitív módon modellezni az adatpontok közötti kapcsolatokat a csomópontok és élek segítségével. Ez a módszer különösen hasznos hálózati adatok, például IoT rendszerek elemzésekor, ahol az eszközök közötti kapcsolatok komplex hálózatot alkotnak. Az ilyen rendszerek esetében az anomáliák észlelése, a mintázatok azonosítása, valamint a prediktív modellek létrehozása kiemelt fontosságú [6].

A gráfstruktúra előnyei közé tartozik a gyors és rugalmas adatlekérdezés, amely nemcsak egyszerűsíti a kapcsolati hálók vizsgálatát, hanem segíti az olyan feladatok megoldását is, mint az útkeresés és a közösségdetektálás. Ezen felül a Neo4j és más gráf adatbázisok támogatják a fejlett elemzési algoritmusok használatát, amelyek lehetővé teszik a komplex rendszerek viselkedésének mélyebb megértését.

Az adatelemzések kiemelkedően fontosak a modern gazdaságban és technológiai környezetben, mivel az adatokból nyert információk alapvetően segítik a döntéshozatalt, optimalizálják a működési folyamatokat, és előrejelzéseket biztosítanak a jövőbeli trendekről. Az adatok közötti rejtett kapcsolatok feltárása különösen lényeges olyan területeken, mint az üzleti intelligencia, hálózatbiztonság vagy akár az IoT rendszerek elemzése. A gráfadatbázisok előnyei közé tartozik, hogy hatékonyan kezelik az összetett kapcsolatokat és gyorsan lehetővé teszik az olyan műveletek elvégzését, mint az útvonalkeresés, anomália detektálás és közösség detektálás. Mivel a gráf adatbázisok intuitív struktúrát kínálnak a kapcsolatok és hálózatok modellezésére, ezek elősegítik a komplex adathalmazok vizuális elemzését és jobban kihasználják a csomópontok közötti kölcsönhatásokat, mint a hagyományos adatbázisok.

3. SZAKIRODALOM TANULMÁNYOZÁSA

A Victor Chang és társai által készített tudományos publikáció [3] a Neo4j gráf adatbázisok komplex adathalmazok elemzésére való alkalmazására összpontosít. Feltárja, hogyan hasznosítható a Neo4j az adatelemzésben, különösen olyan környezetekben, ahol a hagyományos relációs adatbázisok nem hatékonyak a nagy léptékű, egymással összekapcsolt adatok kezelésében. A cikk kiemeli a különféle gráfalgoritmusok használatát többek között olyan feladatokhoz, mint a közösségészlelés, a hasonlóság számítása, a közönségközpontúság és az oldalértékelés (PageRank). Ezek az algoritmusok segítenek a hálózati adatokban rejlő kapcsolatok és minták feltárásában, így a Neo4j különösen hasznos a különféle hálózati adatok elemzéséhez.

A Mohamad és Eka által készített tudományos publikáció [1] egy kutatási gráf-adatbázis fejlesztését tárgyalja a Neo4j használatával, amely lehetővé teszi az egyetemek számára a hatékony többperspektívás adatelemzést. Mivel a hagyományos relációs adatbázisok nehezen kezelik az egymással összekapcsolt kutatási adatokat tartalmazó összetett lekérdezéseket, a szerzők javasoltak egy gráf adatbázis-modellt, mely ezt a korlátot feloldja azáltal, hogy tudományos publikációkkal, kutatókkal, kapcsolódási pontokkal és kutatási projektekkel kapcsolatos adatokat jelenít meg. A tanulmány azt mutatja, hogy a Neo4j felülmúlja a relációs adatbázisokat a többdimenziós elemzés lekérdezéseinek végrehajtásában.

A López és De La Cruz által készített tanulmány [7] a Neo4j gráfadatbázist a hagyományos relációs adatbázisokhoz (RDBMS) képest értékeli. Az áttekintés kiemeli a Neo4j előnyeit az összetett kapcsolatok és a nagyméretű adatok kezelésében, különösen a közösségi hálózatokban és a webes alkalmazásokban. A szerzők különböző országokban végzett benchmarking kísérleteket elemeznek, összehasonlítva a Neo4j-t olyan relációs adatbázisokkal, mint a MySQL és a PostgreSQL. A legfontosabb eredmények azt sugallják, hogy a Neo4j gyorsabb lekérdezési választ ad, és bizonyos alkalmazásokhoz jobban méretezhető.

A Soad Almabdy által készített cikk [2] a relációs adatbázisok és a gráfadatbázisok teljesítményét és funkcióit hasonlítja össze a közösségi hálózatok adatainak kezeléséhez. A tanulmány arra a következtetésre jutott, hogy a gráfadatbázisok jobban megfelelnek a közösségi hálózatok kezelésére és elemzésére, mint a hagyományos relációs adatbázisok, és előnyöket kínálnak a tárolási rugalmasság, a lekérdezési teljesítmény és a valós idejű adatfeldolgozás terén.

A José Guia és társai által készített „Graph Databases: Neo4j Analysis" című tanulmány [5] a gráfadatbázisok, különösen a Neo4j növekvő relevanciájára összpontosít, az adatok növekvő összetettsége és összekapcsolhatósága miatt, különösen olyan területeken, mint a hálózatok.

Mindenzen dolgozatok azt tárgyalják és jelzik előre, hogy a gráfadatbázisok nagyon jól alkalmazhatóak az ipari IoT hálózati adatok elemzésére, sőt komplex és nagyméretű adatmennyiséget előnyösebb gráfadatbázisokkal feldolgozni, mint a hagyományos relációs adatbázisrendszerekkel.

4. GRÁF ADATBÁZISOK, NOSQL ÉS SQL ADATBÁZISOK ÖSSZEHASONLÍTÁSA

Az adatok hatékony tárolása és elemzése kulcsfontosságú a modern technológiai környezetben, különösen a nagy és komplex adathalmazok kezelésében. Az adatbázisok típusai közötti különbségek jelentős hatással vannak arra, hogy hogyan lehet feldolgozni az adatokat. Az SQL (Structured Query Language), NoSQL (Not only Structured Query Language) és gráfadatbázisok mind különböző megközelítéseket kínálnak az adatok tárolására és kezelésére. Ebben az összehasonlításban áttekintjük mindhárom típus alapvető tulajdonságait, valamint azok előnyeit és hátrányait a nagy és összetett adatkészletek elemzése szempontjából.

Az SQL adatbázisok (relációs adatbázisok) táblákban tárolják az adatokat, szigorúan strukturált sémával. Az adatok között meghatározott kapcsolatok vannak, és lekérdezéseikhez az SQL nyelvet használják. Ezek az adatbázisok rendkívül hatékonyak jól strukturált, rendezett adatok kezelésében. Az SQL adatbázisok megbízhatóan kezelik a strukturált adatokat, és lehetővé teszik az adatok közötti kapcsolatok könnyű lekérdezését [7]. Nagy mennyiségű adat tárolása is megvalósítható ezekben az adatbázisokban. Ugyanakkor hátrányuk, hogy kevésbé alkalmasak a bonyolult kapcsolati hálók és hierarchiák kezelésére. Továbbá, kevésbé rugalmasak, ha gyorsan változó vagy különböző típusú adatokat kell tárolni és feldolgozni [2].

A NoSQL adatbázisok nem relációs adatbázisok, amelyek lehetővé teszik a strukturálatlan és gyorsan változó adatok tárolását. Rugalmas sémával rendelkeznek, és sokféle adatformát képesek kezelni, beleértve a dokumentumokat, kulcs-érték párokat, oszlopokat és gráfokat. [7].

A gráfadatbázisok csomópontok és élek segítségével ábrázolják az adatokat, ami ideális megoldás a komplex kapcsolatok, hálózatok kezelésére. Különösen hasznosak az olyan alkalmazási területeken, mint a közösségi hálózatok, az IoT rendszerek és a hálózati elemzések A gráfadatbázisok gyors és hatékony lekérdezéseket tesznek lehetővé, különösen a komplex kapcsolatokkal és hálózatokkal rendelkező adathalmazok esetén. Ezek az adatbázisok könnyen kezelik a dinamikusan változó adatokat és kapcsolatokat, ami ideálissá teszi őket a nagy méretű és bonyolult hálózatok elemzésében [4]. Ugyanakkor hátrányuk, hogy speciális algoritmusokat igényelnek, amelyek alkalmazásához speciális tudás szükséges. Emellett jelenleg kevésbé elterjedtek, mint az SQL vagy NoSQL rendszerek, így kevesebb támogatást nyújtanak felhasználóik számára [9].

5. A NEO4J GRÁF ADATBÁZIS ELEMZÉSI LEHETŐSÉGEI

A Neo4j gráfadatbázisok kiemelkedő lehetőségeket nyújtanak a nagy és komplex adatkészletek elemzésében, különösen a kapcsolati adatok vizsgálata során. A gráfstruktúrák természetesen tükrözik a valós élet hálózatait, ami lehetővé teszi a hatékony adatelemzést számos területen, például pénzügyekben, biológiában, logisztikában és különféle hálózatok elemzésében. A Neo4j előnyei a következő kulcsfontosságú területeken jelentkeznek [11] [3]:

- Kapcsolati elemzés: A gráfadatbázis legnagyobb erőssége a kapcsolatok kezelésében rejlik. Ahelyett, hogy táblák közötti kapcsolódási pontokat keresnénk (mint a relációs adatbázisokban), a Neo4j természetes módon tárolja és elemzi az entitások közötti kapcsolatokat, így nagymértékben csökkenthető a lekérdezések komplexitása és futási ideje.

- Gráf algoritmusok alkalmazása: A Neo4j széles körben elérhető algoritmusokat kínál a gráfok elemzéséhez, például a legközelebbi szomszéd, központiság, közvetettség és közösség felismerés. Ezek az algoritmusok lehetővé teszik a mélyebb betekintést komplex hálózati struktúrákba, és feltárhatják a rejtett összefüggéseket az adatok között. Továbbá olyan algoritmusokat is biztosít a felhasználó számára, melyek által lehetővé válik az anomália detektálás és a predikciók készítése is.
- Adatfeltárás nagy adatkészletekben: A Neo4j skálázhatóságának köszönhetően képes hatékonyan kezelni és elemezni akár több milliárd csúcsot és élhálózatot is. Ez lehetővé teszi a felhasználó számára, hogy nagy mennyiségű adatból kapcsolati összefüggéseket azonosítsanak anélkül, hogy az adatok feldolgozása túlzott erőforrásokat igényelne.
- Valós idejű adatelemzés: A valós idejű elemzési képességek lehetővé teszik a felhasználók számára, hogy dinamikus hálózatokkal dolgozzanak, és azonnal reagáljanak az új adatokra vagy változásokra. Ez különösen hasznos olyan alkalmazásoknál, mint a hálózati támadások detektálása vagy az ajánlórendszerek készítése, amelyek gyors és pontos válaszokat igényelnek.
- Kódolási egyszerűség: A Neo4j Cypher lekérdezőnyelve kifejezetten a gráfokkal való munkavégzésre lett optimalizálva, ezáltal jelentősen csökkenti a bonyolult SQL lekérdezések igényét, miközben könnyen olvasható és hatékony kódot eredményez.

A fentiek alapján a Neo4j kiváló választás lehet olyan helyzetekben, ahol a kapcsolatok, hálózatok és ezek hatékony elemzése kritikus fontosságú nagy és komplex adatkészletek kezelésénél. A rendszer erőssége a kapcsolatok közvetlen tárolása és elemzése, amely jelentősen leegyszerűsíti a lekérdezéseket. Ezenkívül a beépített gráf algoritmusok segítenek az összefüggések feltárásában és előrejelzések készítésében. A valós idejű elemzési képességek és a skálázhatóság nagy adatmennyiségekkel is megkönnyítik a komplex hálózati struktúrák kezelését.

6. HÁLÓZATI ADATELEMZÉS GRÁF ADATBÁZISOK SEGÍTSÉGÉVEL

A Neo4j gráfadatbázisok alkalmazási lehetőségeinek feltérképezése érdekében egy hálózati adatkészlet adatain teszteltem a gráfstruktúra nyújtotta lehetőségeket. Ez az adatkészlet egy hálózati kommunikációs rendszer nyilvános interfészeiről gyűjtött hálózati áramlási és sebezhetőségi adatokból áll, valamint több hetes időszak adatsorát tartalmazza.

A következőkben a felhasznált adathalmaz fontosabb információit ismertetjük. Az hálózati adatok 5 csúcsra vannak osztva, valamint meghatározottak a csomópontok közötti kapcsolatok típusai. Az adott csomópontoknak és a kapcsolatoknak, mint tulajdonság adtam meg a hálózati adatgyűjtés során gyűjtött információkat. Az adatkészletünk gráfstruktúrájának a csomópontok és kapcsolatok szerinti felépítése a következő:

Csomópontok:

- VulnService: Ez a csomópont az IP címek alapján kerül meghatározásra, és olyan fontos attribútumokat tartalmaz, mint az IP cím, az utolsó látott státusz, és a hálózat neve.
- VulnPort: Ezt a csomópontot a port számok alapján határoztuk meg, melyekhez tartozik az IP cím, az utolsó látott státusz, a szolgáltatás neve, a hálózat neve, a port száma és a használt protokoll.
- VulnServiceGroup: Egy mesterségesen létrehozott csoport, ami az IP címek első három szegmensét használja, és az utolsó szegmensét eltávolítja. Ezáltal egy csoportot hoz létre azon IP címekből, amelyek az első három szegmens alapján azonosak.
- NVDCVE: Ezt a csomópontot a National Vulnerability Database (NVD) [12] által kiadott CVE (Common Vulnerabilities and Exposures) azonosítók alapján határozták meg [hivatkozás a honlapra], amelyek fontos tulajdonságai közé tartozik az azonosító és a súlyosság.
- ServiceType: Ez a csomópont a szolgáltatások típusát határozza meg.

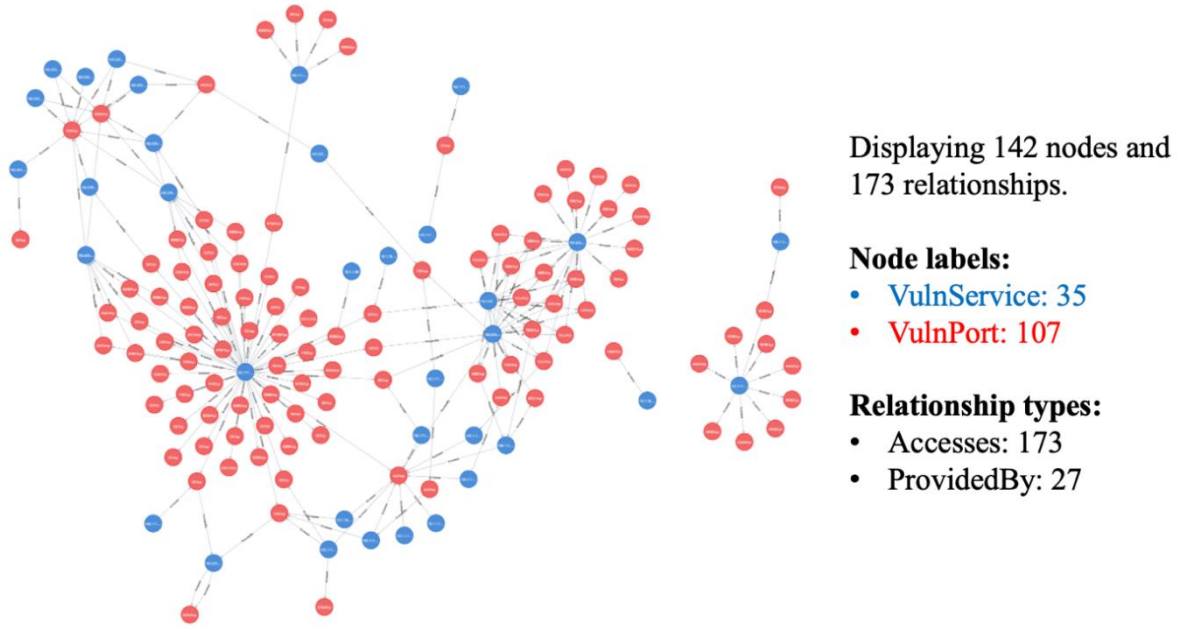
Kapcsolatok:

- Accesses: A kapcsolat a VulnService csomópontból a VulnPort csomópont felé irányul, jelentése, hogy a VulnService hozzáfér a VulnPort által kezelt információkhoz.
- ProvidedBy: Ez a kapcsolat a VulnPort csomópontból a VulnService csomópont felé mutat, ami azt jelenti, hogy a VulnPort nyújt szolgáltatásokat vagy funkciókat a VulnService számára.
- PartOfGroup: A VulnService csomópont a VulnServiceGroup csomópont része, ezáltal a kapcsolat a VulnService-ből indul ki és a csoportot jelölő VulnServiceGroup felé mutat.
- AccessesGroup: Ez a kapcsolat a VulnServiceGroup csomópontok között áll fenn, jelölve, hogy egyes csoportok hogyan férhetnek hozzá egymás erőforrásaihoz vagy információihoz.

- ImpactedBy: A kapcsolat a VulnPort csomópontból indul ki és az NVDCVE csomópont felé irányul, ami azt jelzi, hogy a VulnPort működését az NVDCVE által nyilvántartott sebezhetőségek befolyásolják.
- IsOfType: Ez a kapcsolat a VulnPort csomópont és a ServiceType csomópont között fennáll, azt jelölve, hogy a VulnPort milyen típusú szolgáltatásokat nyújt vagy kezel.

Az adatkészlet lekérdezéséhez a beépített cypher lekérdező nyelvet lehet használni, mely által az egyszerűbb statisztikai lekérdezésektől kiindulva a komplexebb gráfelemző algoritmusok, klaszterezések végrehajtása is lehetővé válik. A cypher lekérdező nyelv a Neo4j gráf adatbázis kezelő rendszer számára van kifejlesztve és célja, hogy intuitív és ember által könnyen olvasható módon tegye lehetővé a gráfadatok lekérdezését, valamint manipulálását. A nyelv a mintaillesztésen alapul, amely lehetővé teszi a felhasználók számára, hogy egyszerűen írjanak le mintákat a gráfokban lévő csomópontok és élek közötti kapcsolatokra, így támogatja hatékonyan az összetett lekérdezéseket és adatmanipulációs műveleteket.

Az adatkészlet megismeréséhez készített egyszerűbb cypher lekérdezések által olyan kérdésekre lehet választ találni, mint hogy pontosan hány VulnService csomópont van az adatkészletben, vagy számolja meg egy adott kapcsolattípus alapján az érintett kapcsolatokat és csomópontokat (1. ábra).



1. ábra: Az Accesses kapcsolattípus alapján kirajzolt gráf

De specifikált IP címek alapján ki lehet listázni tulajdonságokat és különböző információkat a csomópontokról. Továbbá értékes jellemzője a gráfadatbázisoknak, hogy megadott specifikációk alapján rész-gráfokat lehet ábrázolni a teljes adatkészlet alapján, vagy különböző kisebb csoportokat is meg lehet jeleníteni jól specifikált paraméterek alapján.

A Neo4j adatbáziskezelő rendszer számos eszközt és funkciót kínál a gráfok elemzésére a beépített Graph Data Science Library által, mely több gráf elemző algoritmust is tartalmaz, melyek közül néhányat mi is megvizsgáltunk. A Neo4j adatbázisunkban több algoritmust is teszteltünk az adatkészleten, az alábbiakban részletesen ismertetjük ezeket:

Útkeresés: A gráfban komplex utak keresése történt, például két különböző VulnService csomópont közötti legrövidebb út meghatározása, valamint minden lehetséges útvonal felderítése ezen csomópontok között. Ez a módszer segít azonosítani az összeköttetéseket és a távolságokat a hálózati elemek között.

Központisági mutatók: Az algoritmusok között szerepelt a központisági mutatók (degree centrality) kiszámítása, ami az egyes csomópontok fontosságát méri a hozzá tartozó kapcsolatok száma alapján (1. táblázat). Ez a módszer kulcsfontosságú a nagy forgalmat lebonyolító vagy stratégiai fontosságú csomópontok azonosítására.

VulnService és VulnPort csomópontok fokszám központiságának kiszámítása (részleges eredmény) 1. táblázat

Név/IP cím	Fokszám Központiság
“x.x.2.4”	9
“x.x.87.91”	6
“x.x.251.1”	5

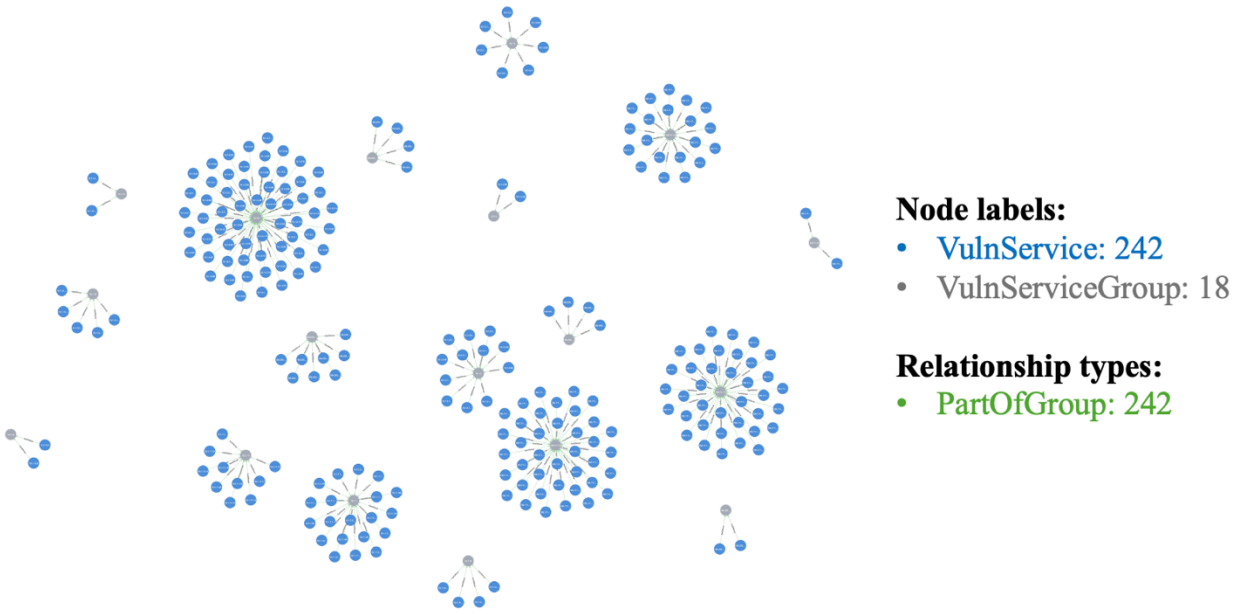
Közösségdetektálás: Több közösségdetektáló algoritmust is alkalmaztunk, mint például a Label Propagation, Louvain, és PageRank (2. táblázat). Ezek segítségével azonosíthatók azok a csoportok, amelyek intenzív kommunikációt folytatnak, vagy erősen összetartoznak. Az eredmények között szerepel a közösségek struktúrájának feltérképezése és a csomópontok közötti erősebb és gyengébb kapcsolatok kiemelése.

A Page Rank algoritmus részleges eredménye 2. táblázat

Név/IP cím	Pontszám
“x.x.2.44”	1.245284994
“x.x.251.6”	0.538105603
“x.x.251.168”	0.473421336

Anomália detektálás: Kiegészítő eszközök segítségével váratlan hálózati forgalmi minták, szokatlan port használatok, és a hálózati kommunikáció szokatlan időpontjai azonosíthatók. A módszer segít felismerni a túlzottan gyakori kapcsolatokat, a szokatlanul nagy adatforgalmat, és a ritka kapcsolatokat tartalmazó csomópontokat.

Mintadetektálás: Ez a módszer lehetővé teszi előre definiált minták keresését a gráfban (2. ábra), különösen hasznos lehet biztonsági vizsgálatokban. Alkalmazása során különböző támadási minták azonosítására is sor került.

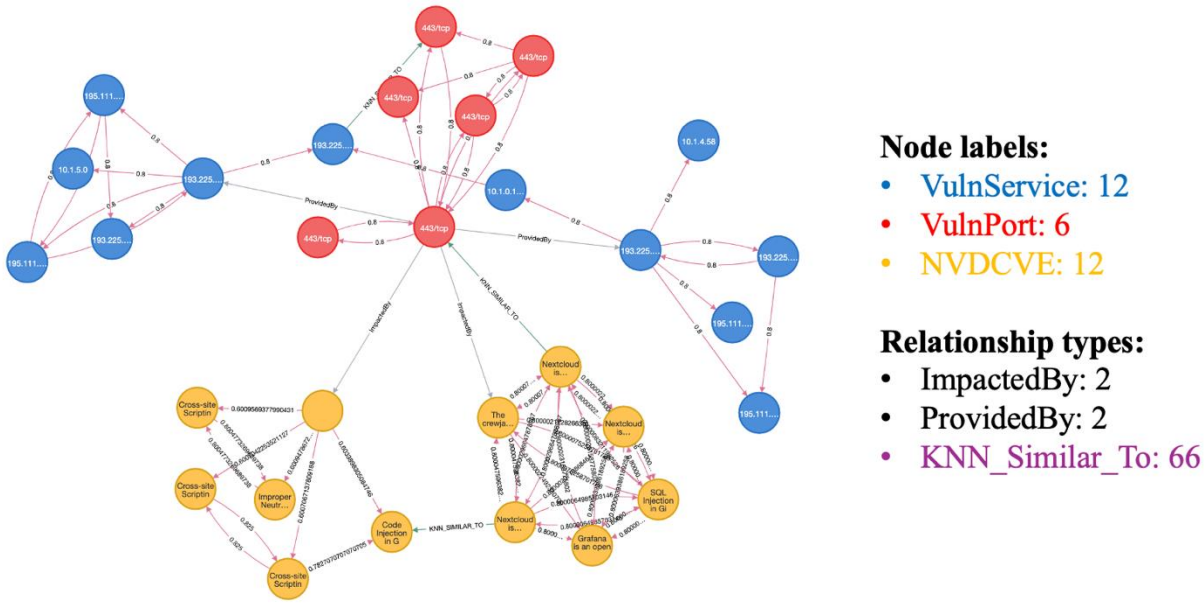


2. ábra - A hierarchikus mintadetektálás eredménye

Ezen algoritmusok alkalmazása átfogó képet adott a hálózati struktúránkról, segítve a biztonsági kockázatok jobb megértését és kezelését.

Az adathalmaz további tanulmányozása érdekében klaszterezési eljárásokat is alkalmaztunk, hogy a hasonló tulajdonságokkal rendelkező csomópontokat csoportosítani tudjuk vagy a kapcsolatok alapján tudjunk kialakítani új csoportokat. Ennek megvalósításához a K-NN (K-Nearest Neighbors) és Lovain-algoritmusokat finomhangoltuk a mi adatkészletünkre.

A K-NN algoritmus egy gépi tanulási módszer, amely csomópontok vagy adatpontok osztályozását és szegmentációját végzi a "legközelebbi" szomszédok alapján. Az algoritmus főként a csomópontok közötti hasonlóságok azonosítására szolgál, lehetővé téve a leginkább hasonló vagy eltérő csomópontok azonosítását, így használható anomália detektálásra is. Az előnyei között szerepel a nem-lineáris kapcsolatok felismerése, ami sokrétű alkalmazást tesz lehetővé, beleértve az IP címek és portok közötti összetett kapcsolatok elemzését is. A 3. ábrán a KNN algoritmus futtatásának egy részeredménye látható, az első 25 kapcsolatra limitálva, amely bemutatja, hogy mely az az 5 legközelebbi szomszéd minden egyes csomópont esetében, melyek megfelelnek az paraméter listában meghatározott tulajdonságoknak.



3. ábra - A KNN algoritmus futtatásának részeredménye

A Louvain algoritmus a gráfokon belüli közösségi struktúrák azonosítására szolgál, mely során az algoritmus iteratív módon csoportokat formál a szorosabban kapcsolódó csomópontokból. Az iterációk eredményeként létrejött közösségek közösségi azonosítók formájában segítenek a csomópontok csoportosításában, így könnyebbé téve a gráfban lévő kapcsolati mintázatok értelmezését. A 3. táblázatban látható a Louvain algoritmus egy részeredménye, ahol az IP címek és azok a communityId-k láthatóak, ahova az algoritmus sorolta az adott IP címet.

A Louvain algoritmus futtatásának részeredménye 3. táblázat

Név/IP cím	communityId
“x.x.87.91”	201
“x.x.1.193”	12
“x.x.250.132”	7

7. KÖVETKEZTETÉSEK

A Neo4j gráf adatbázis alkalmazása lehetőséget nyújt a hálózati adatok mélyebb elemzésére és értelmezésére, különösen az IoT rendszerek hálózataiban és digitalizált környezetekben. A gráf struktúra előnyei a következőképpen összegezhetők:

- Komplex kapcsolatok kezelése: a Neo4j rugalmasabb adatkezelést tesz lehetővé, amely különösen fontos az összetett hálózatok és nagy adatmennyiségek esetén, ahol a relációs adatbázisok hatékonysága korlátokba ütközik.

- Anomáliák és minták azonosítása: a gráf adatbázisok elősegítik az anomáliák és minták gyorsabb észlelését, amelyek kulcsfontosságúak a hálózati biztonság és a prediktív modellek kialakításában.
- Fejlett analitikai algoritmusok: a Neo4j széles körben támogatja a gráf algoritmusokat, mint a közösségdetektálás vagy a központiság számítása, amelyek mélyebb betekintést nyújtanak a hálózati struktúrákba és elősegítik a rejtett kapcsolatok feltárását.
- Valós idejű elemzés és skálázhatóság: a Neo4j képes kezelni a valós idejű adatfolyamokat és nagy méretű adatkészleteket, ami ideális választássá teszi a dinamikusan változó környezetekben.

Összegzésül, a Neo4j alkalmazása növeli a hálózati adatok elemzésének hatékonyságát, és lehetővé teszi a bonyolult kapcsolati hálózatok kezelését, javítva ezzel a döntéshozatalt és az előrejelzést a modern technológiai környezetben.

KÖSZÖNETNYILVÁNÍTÁS

Jelen kutatás az NKFIH-3568-1/2022 számú projekt része, amely a Kulturális és Innovációs Minisztérium által a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból, a 2022-2.1.1-NL támogatási keretből finanszírozott támogatással valósult meg. A számítási erőforrásokat a HUN-REN Cloud biztosította [8][~~6~~].

IRODALMI HIVATKOZÁSOK

[1] Afandi, M.I. and Wahyuni, E.D., 2020, October. University Research Graph Database For Efficient Multi-Perspective Data Analysis Using Neo4j. In *2020 6th Information Technology International Seminar (ITIS)* (pp. 286-290). IEEE.

[2] Almabdy, S., 2018, April. Comparative analysis of relational and graph databases for social networks. In *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)* (pp. 1-4). IEEE.

[3] Chang, V., Songala, Y.K., Xu, Q.A. and Liu, B.S.C., 2022, April. Scientific Data Analysis using Neo4j. In *FEMIB* (pp. 75-84).

[4] Fernandes, D. and Bernardino, J., 2018. Graph Databases Comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB. *Data*, 10, p.0006910203730380.

[5] Guia, J., Soares, V.G. and Bernardino, J., 2017, April. Graph Databases: Neo4j Analysis. In *ICEIS (1)* (pp. 351-356).

[6] Lal, M., 2015. Neo4j graph data modeling. Packt Publishing Ltd.

[7] López, F.M.S. and De La Cruz, E.G.S., 2015. Literature review about Neo4j graph database as a feasible alternative for replacing RDBMS. *Industrial Data*, *18*(2), pp.135-139.

[8] M. Héder et al., “The Past, Present and Future of the ELKH Cloud,” InfTars, vol. 22, no. 2, p. 128, Aug. 2022, doi: 10.22503/inftars.XXII.2022.2.8.

[9] Macak, M., Stovcik, M. and Buhnova, B., 2020, May. The Suitability of Graph Databases for Big Data Analysis: A Benchmark. In *IoTBDS* (pp. 213-220).

[10] Neo4j honlap: https://neo4j.com/

[11] Needham, M. and Hodler, A.E., 2019. Graph algorithms: practical examples in Apache Spark and Neo4j. O'Reilly Media.

[12] NVD honlap: https://nvd.nist.gov/