Multidiszciplináris tudományok, 9. kötet. (2019) 4 sz. pp. 322-332 **https://doi.org/10.35925/j.multi.2019.4.32**

TANULÓ ALGORITMUSOK A FELÜGYELETI RENDSZEREKBEN - ÁTTEKINTÉS

Hornyák Olivér
egyetemi docens, Miskolci Egyetem, Informatikai Intézet
515 Miskolc, Miskolc-Egyetemváros, e-mail: oliver.hornyak@uni-miskolc.hu

Mileff Péter
egyetemi docens, Miskolci Egyetem, Informatikai Intézet
3515 Miskolc, Miskolc-Egyetemváros, e-mail: mileff@iit.uni-miskolc.hu

Nehéz Károly
egyetemi docens, Miskolci Egyetem, Informatikai Intézet
Cím: 3515 Miskolc, Miskolc-Egyetemváros, e-mail: aitnehez@uni-miskolc.hu

Absztrakt

Az Internet of Things (IoT) napjaink egyik legmeghatározóbb fejlődési irányvonala. Az utóbbi pár évben nagy lendülettel a privát és publikus szektorban is gyökeresen átalakítja az üzleti folyamatokat. Az egymással is kommunikáló okos rendszerek, az intelligens gépek és a különféle szenzorok egyre nagyobb hatással vannak a hétköznapi életünkre. Bár tudományos léptékkel mérve az IoT még gyerekcipőben jár, jól látszik, hogy a vállalatok, az ipar rengeteg tőkét fektet a tervezési, gyártási, értékelési és egyéb üzleti folyamatokba való beépítésére. A közeljövőben minden bizonnyal az IoT életünk szerves részévé válik. Jelen cikkben egy rövid áttekintést nyújtunk mindazokról az alapfogalmakról, amelyekre az IoT és annak részei épülnek. Tárgyaljuk, hogyan és milyen feltételek mellett lehetséges egy ilyen rendszert működtetni, milyen szerepe lehet a tanuló algoritmusoknak a keletkezett adathalmaz feldolgozásában és hogyan illeszthető be egy ilyen komplex infrastruktúra a felügyeleti rendszerekbe.

Kulcsszavak: *IoT, integrált rendszerek, tanuló algoritmusok, adatbányászat*

Abstract

The Internet of Things (IoT) is one of the most trending scientific topics. Over the past few years, it has been revolutionizing processes in the private and public sectors. Smart systems that communicate with each other, intelligent machines and various sensors are increasingly affecting our daily lives. Although IoT is still in its infancy on a scientific scale, companies, are investing a lot of capital in integrating IoT into their planning, manufacturing, etc. processes. This article provides an overview on the main mathematical methods for IoT data analysis, on the conditions to operate such a system, the key learning algorithms to process the resulting data set, and how to integrate such a complex infrastructure into management systems.

Keywords: *IOT, integrated systems, learning algorithms, data mining*

1. Bevezetés

Az IoT (Internet of Things / dolgok internete) globális hálózatba ágyazott összekapcsolt eszközökkel foglalkozik. Nevezetesen, az IoT egy vagy több kommunikációs protokollt (például WAMP -Web Application Messaging Protocol [4], MQTT MQ Telemetry Transport (MQTT) [5], és számos további protokollt [1-3]) használó eszközök vezeték nélküli érzékelők hálózata (WSN - Wireless Sensor Network). Az IoT alkalmazások több kontextusban létezhetnek. Jellemző vállalati felhasználási területek: kiskereskedelem, okos közművek és energia, egészségügy, okos város, gyártósorok karbantartása, távfelügyelet, önvezető autók, míg a végfelhasználók esetén: háztartási alkalmazások, hordható/mobil eszközök, csatlakoztatott okos eszközök, személyes egészségvédelem, távérzékelés.

2020-ban körülbelül 17 milliárd olyan eszköz vesz körül minket, amely potenciálisan adatot továbbíthat a felhő rendszerekbe, azaz körülbelül 5 kvintillion ($5x10^{18}$) bájtnyi adat áll rendelkezésre napi átlagban [6]. Informatikai szempontból az IoT számos kihívással jár. Cikkünk néhány ilyen kihívással foglalkozik. Ezek egyik aspektusa, hogy a tanuló algoritmusokban megtestesülő intelligencia felépítéséhez és terjesztéséhez megfelelő informatikai architektúra szükséges:

- biztonságos üzenetküldés,
- biztonságos adattárolás és
- alkalmazástelepítési infrastruktúra.

Lényeges, hogy nagy adatfolyamok kezelésére ki kell alakítani a helyi eszközök hálózatát egy elosztott számítási környezetben. Képzeljünk el több csomópontot (például kamerák, hőmérséklet- vagy nyomásérzékelők, számlálók), amelyek elérik a mobil hálózatot egy adott területen, például egy gyárban. Ezeket az eszközöket a fizikai jelenségek mérésére és az adatok szerverre továbbítására tervezték. A közelmúltig ezt a klasszikus (pl. egy master / slave modellben) hajtották végre. Alternatív megoldásként az eszközök kommunikáltak társaikkal és kiértékelték a kapott adatokat. Továbbá adatfolyam-csomagokat (helyi puffereket) képeztek, és helyi kiszolgálóként is működhetnek. A döntő kérdés az, hogy ezeknek az eszközöknek előzetesen fel kell-e dolgozniuk az általuk felvett adatokat, mielőtt az adatokat a központi rendszerbe továbbítják.

Az általánosan használt modell szerint a nagy adatot (Big Data) hatalmas számítási erőforrások felhasználásával AI (Artificial Intelligence – mesterséges intelligencia) modellek alakítják át kezelhető adattá (amelyek valószínűleg a felhőben vagy egy „közeli” számítási központban működnek).

1.1. Az adatok jellemzői

[7] az angolul 6 V-vel kezdődő szóval jellemzi az IoT-t:

Volumen (Volume): Az adatmennyiség meghatározó tényező, amelyet figyelembe kell venni. Korábban is kezeltek tömeges / nagyon nagy méretű adatot, azonban a Big Data megjelenésével az IoT eszközökkel generált adatok mennyisége a korábbi ipari gyakorlatokhoz képest jelentősen megnőtt.

Sebesség (Velocity): Az IoT adatainak előállítása és feldolgozása olyan gyors, hogy elérhetők valós időben is. Ezért a hatékony működés és elemzés érdekében olyan eszközökre van szükség, amelyek a valósidejűséget támogatják.

Változatosság (Variety): Általában az adatok különféle formában érkeznek: lehetnek strukturált, félig-strukturált és strukturálatlan adatok. Az IoT az adattípusok széles választékát állíthatja elő: például szöveges, audio, video, szenzorikus, stb. adatokat.

Valódiság (Veracity): A valódiság az adatok minőségre, következetességre és megbízhatóságára utal. Ez az adatgyűjtés a pontosságra vezethető vissza.
Változékonyság (Variability): Ez a tulajdonság az adatáramlás különböző mértékére utal. Az IoT alkalmazások jellegüktől függően különböző adatgeneráló összetevőkkel rendelkeznek, így eltérő mennyiségű adatáramlásokkal rendelkezhetnek. Továbbá elképzelhető, hogy vannak csúcsidők, amikor bizonyos adatforrások megnövekedett adatmennyiséggel dolgoznak.
Érték (Value): Az érték a Big Data hasznos adattá történő átalakításának folyamán jelentkezik. Ezek azok az információk, amelyek versenyelőnyt jelentenek a felhasználóknak. Az adat értéke mind a mögöttes folyamatok / szolgáltatásoktól, mind az adatkezelés módjától függ. Például egy orvosi alkalmazásnak egy érzékelő összes adatát fel kell vennie, míg az időjárás előrejelzés szolgálatnak csak véletlenszerű mintákra van szüksége.
Ezen jellemzők fontossága egy adott alkalmazás esetén különböző lehet. Például ipari környezetben nem mindig az adatok mérete a legfontosabb, máskor a hálózati biztonság és az üzleti érték nagyobb jelentőséggel bírnak.

1.2. Adatforrások

Az IoT különböző forrásaiból származó adatok többnyire nyers adatok, és eredeti formájukban nem mindig alkalmasak elemzésre [8]. Következésképpen rossz stratégia egyszerűen összegyűjteni az összes nyers adatot egy kiszolgálón, és azt remélni, hogy értékes tényadatot találunk benne. Ennek oka nem csak a műszaki korlátokban keresendő, például a számítási erőforrások korlátosságában, hanem az időbeli és térbeli összefüggéseket gyakran nem veszik megfelelően figyelembe.

Másodszor, az IoT környezetben bekövetkezett apró, nem észlelt változások alkalmanként rendkívüli jelentőséggel bírhatnak. Az intelligencia elterjedése a hálózatban kiemelt jelentőségű ennek a vizsgálatában. A megfelelő modell megalkotására van szükség az adatok értékeléséhez, tudomásul kell venni, hogy a nem észlelt helyi zajforrások az adatot szennyezhetik. És ismert, hogy az IoT adatbázis- és feldolgozási erőforrásigénye hatalmas, így a hardver kiválasztásakor erre figyelemmel kell lenni.

A hatékony tárgyak internete informatikai környezet felállításához a statikus nagy adatmennyiségen kívül foglalkozni kell az internettel/hálózattal kapcsolatos technikai kérdésekkel [7], amelyek közé tartozik a vezetékes és vezeték nélküli adatátvitel, útvonalak kialakítás, lokalizáció, klaszterek kialakítása, biztonság és rendelkezésre állás.

- **Nagyobb adatfolyam**: Rengeteg adatgyűjtő eszköz van elosztva és telepítve az IoT alkalmazásokhoz, és ezek folyamatosan generálnak adatfolyamokat. Ez óriási mennyiségű folyamatos adathoz vezet.
- **Heterogenitás**: A különböző IoT eszközök különböző információkat gyűjtenek, ami az adataik természetes heterogenitását eredményezi.
- **Idő és tér korreláció**: A legtöbb tárgyak internete alkalmazásában az érzékelő eszközöket egy adott helyhez csatolják, és így minden egyes adatelem hely- és időbélyeggel van ellátva.
- **Magas zajszintű adatok**: Az IoT eszközök érzékelői hibákat és zajt is gyűjtenek a gyakorlatban (hibákat okozhat továbbá az átviteli csatorna rossz minősége is).

Az IoT alkalmazásokba intelligens algoritmusokat kell beágyazni és futtatni [8]. A feladat elvégzéséhez a közelmúltban félig felügyelt tanulási algoritmusokat alkalmaznak a hiányzó minták kiegyensúlyozására vagy akár további minták mesterséges létrehozására. Ezek a technikák egyrészt modellezhetik a kis mennyiségű címkézett adatot nagy mennyiségű, nem címkézett adattal

(transzduktív tanulás), másrészt kis mennyiségű címkézetlen adatot nagy mennyiségű címkézett adatba helyezhetnek (induktív tanulás).

1.3. Adat analitika és gépi tanulás

Az IoT-adat nemcsak Big Data, hanem egyéb forrásból származó megfigyeléseket is tartalmazhat (pl.: audio, vizuális és numerikus), és paraméterei egy adott kontextusban időben változhatnak. Ezek a tulajdonságok megnehezítik az adatelemzés / mesterséges intelligencia adaptálását az IoT-hez [8]. Az IoT-nek olyan intelligens algoritmusokra van szüksége, amelyek rugalmasak a különféle források és változó tulajdonságok bevonásával, valamint adaptíven parametrizálhatók. A gyakorlatban figyelembe kell vennünk a korábban már megtanult kategóriákat és kapcsolatokat az új megfigyelések értelmezése érdekében. A hatékony tanulási stratégia szempontjából elengedhetetlen a már megtanult osztályok és kapcsolatok átadása. Az emberek ezt a stratégiát arra használják, hogy új viselkedést tanuljanak hasonló tanult viselkedés alapján, vagy emlékezzenek az új objektumok tulajdonságaira, korábban ismert osztályok alapján.

1.4. Infrastruktúra

A különböző eszközökre elosztott intelligens alkalmazások készítése kifinomultabb architektúrát igényel, mint az egyszerű monolitikus szoftverek architektúrája. Így beszélhetünk a mikroszolgáltatásokról, valamint az infrastruktúra által kezelt verzió kezelésről és az üzenetküldésről.

1.4.1. Mikroszolgáltatások: biztonság, stabilitás és adatkezelés

A telekommunikációs protokollokat és az üzenetküldési technológiákat egyrészt a különféle alkotóelemei közötti kompatibilitási kérdések motiválják. Másrészt a biztonsági és stabilitási kérdések is nagyon fontosak, például a behatolás megelőzése vagy a hálózati leállása. Ebben a cikkben az alkalmazási kört az ipari alkalmazásokra korlátozzuk, az alábbiakban áttekintjük az üzenetküldési technológiákat és az azokban felmerülő kérdéseket:

1.4.2. OPC szerver

Az OPC (Open Platform Communications) 1996-ban került bevezetésre az ipari automatizálás egyesítő protokolljaként. A módszer fő gondolata az, hogy például egy szerelő sor sok különféle vezérlő egységgel működhet, pl. PLC-kkel (programozható logikai vezérlők), amelyeknek van saját üzenetküldési protokollja. Az OPC-kiszolgáló közvetlenül adatokat gyűjt az összes PLC-ről és HMI-ből (ember - gép interfészekről) egy gyári környezetben egy kiszolgáló-kliens modell segítségével. A PLC-k hagyományosan különféle Modbus protokollokat és ethernet hálózatokat használnak. Tekintettel a kompatibilitási problémákra, az OPC alternatívája egy adott gyártósorra épített egyedi szoftver, amely időigényes, drága és rugalmatlan. Ez indokolja az egységes protokoll használatát.

1.4.3. OPC UA szerver

A fent említett OPC protokoll volt a telekommunikációs szabvány az üzemi automatizálásában 2006-ig, amikor az OPC UA (Open Platform Communications Unified Architecture) új verzióját jelent meg. Az OPC UA az OPC képességeit tovább bővíti: az új szabvány már platformfüggetlen, míg az OPC kizárólag Windows operációs rendszeren futott, és számos nyílt forráskódú OPC UA csomag érhető el C, Python, JavaScript stb. technológiákkal külső fejlesztőknek.

Az OPC UA szerver felépítése egyszerű. Van egy "gyökércsomópont", amely megnyitja a kommunikációs csatornát (amelyet tűzfalak védhetnek), és a címteret, amelyben minden csomópont egy objektum (objektum-orientált programozási értelemben), megfelelő funkciókkal és attribútumokkal (statikus és dinamikus adatok). A gyökércsomópont biztonságos protokollokkal, például MQTT és / vagy WAMP segítségével csatlakoztatható a külvilághoz.

Napjainkban az OPC UA az Industry 4.0 kapuja. A teljes automatizáltság eléréséhez az OPC UA technológiának be kell épülnie nemcsak a vezérlőberendezések új generációiba (PLC-k, HMI-k stb.), hanem minden olyan eszközbe is, amely ipari IoT hálózatokhoz kapcsolódik. Ez egyszerűsíti a szoftveres adatfeldolgozást. Kezelhetőbb, olcsóbb, az Ipar 4.0 igényeinek jobban megfelelő megoldást nyújt.

Az alkalmazást felhőben vagy távoli szervereken is elhelyezhetik. Következésképpen a helyi IoT hálózatból a „külvilágba" vagy a „külvilágból" való információáramlást és az alkalmazások frissítését kezelni kell.

2. Rendszertervezés

Az IoT megkönnyíti és elérhetővé teszi a folyamatos állapot-alapú megfigyelést és a prediktív beavatkozást. Ezen a területen az alábbi négy fő trendet figyelhetjük meg:

- Vezeték nélküli kapcsolatok dominánssá válása,
- Olcsó érzékelők megjelenése a piacon,
- Felhő alapú számítástechnika térnyerése,
- Modern analitika, vagy mesterséges intelligencia módszerek megjelenése (AI).

Ezek a technológiák elszigetelve, önmagukban is értékesek. Egyetlen megoldásként kombinálva képesek átalakítani az ipari világot. Az IoT architektúra négy fő részből áll: végfelhasználói eszközök, hálózat, eszközkezelő szolgáltatás, adatkezelő szolgáltatás.

2.1. Végfelhasználói eszközök (edge devices)

Ezek az eszközök az IoT hálózat „végén", az adatforrás közelében találhatók. Ezek egy feldolgozó egységből (ARM architektúrájú vagy hagyományos / miniszámítógép), valamint egy vagy több megfigyelő végpontból (szenzorok, kamerák, mikrofonok stb.) állnak.

Két alapvető modell létezik az IoT feladatokban: az adatok valós időben történő streaming-modellje és az adatkezelés és számítás fordított modellje. Minél közelebb kerülünk az adathoz, annál kisebb lesz a számítási erőforrásigény. Ezek az eszközök csak kis mennyiségű előfeldolgozást tudnak elvégezni, ezért okosan kell megválasztanunk módszereinket.

A tipikus számítási feladatok öt összetevője:

1. **Komplex eseményfeldolgozás**: A komplex eseményfeldolgozó szoftvereket és szolgáltatásokat évtizedek óta használják számos területen. A rendszer több érzékelőből vagy jelből vesz adatokat, majd az adatokra specifikus minták kialakulásakor reagál. Egy megfelelő platformban modelleket és mintaillesztést fejlesztenek ki a felhőben, majd a végfelhasználói eszközökre telepítik. A rendszerek építéséhez és futtatásához az egyik általános nyílt forráskódú technológia az Apache Storm.
2. **Gépi tanulás mesterséges intelligencia**: A gépi tanulás alapvető gondolata az, hogy lehetővé tegye a gép számára az adatok jelentőségének megtanulását. Ezeket a modelleket ismert algoritmusok segítségével lehet felépíteni, míg a fejlettebb gépi tanulási eszközök a számítógép képzésére koncentrálnak, hogy azok megtanulják a mintákat és az anomáliákat, hogy ezután saját magának

tanulhasson. A legtöbb IoT eszköz támogatja a gépi tanulási modelleket az eszközön belül a pl. a TensorFlow vagy más technológia használatával.
3. **Alkalmazások**. Néhány nagyobb teljesítőképességű IoT lehetővé tette az alkalmazások közvetlenül a végfelhasználói eszközön történő futtatását. Az alkalmazások az IoT szélső eszközeinek fontos elemévé váltak, mivel az adatok, a döntési folyamatok és a riasztó / figyelő rendszerek közvetlenül a primer adatokkal futnak és teljesen lokálisak.
4. **Offline adatok**. Számos IoT-alkalmazás a hálózati kapcsolat ingadozásától szenved. Ezekben az esetekben sok IoT eszköz lokális tárolási lehetőségeket biztosít az adatok ideiglenes tárolására, amíg a kapcsolat helyre nem áll. Az IoT végfelhasználói eszköz a felhőhöz való kapcsolódás nélkül is képes lehet döntéseket hozni.
5. **Adatkezelés**. Az IoT fontos eleme az adatkezelés: annak ismerete, mely adatokat kell megőrizni, és melyeket kell eleldobni, mivel kevés üzleti értéke van. Ezenkívül az adatkezelés aggregálást is biztosíthat, ami csökkenti az elküldött adatok mennyiségét, redukálja a hálózat terhelését.
Sok esetben az IoT eszköznek szüksége lehet korábbi, vagy nem szabványos protokoll támogatására. Ezekben az esetekben az IoT eszköz olyan szolgáltatásokat nyújthat, amelyek segítik a gyűjtött, lefordított és a felhőbe továbbított adatok kezelését.

2.2. A hálózat

A hálózat kapcsolatokból és csomópontokból áll. Az ilyen kapcsolatok és csomópontok túlságosan heterogének lehetnek, mivel az internetes hálózat nagyon dinamikus. Ez azt jelenti, hogy új eszközök csatlakozhatnak vagy kiléphetnek a hálózatból, vagy ezek az eszközök új protokollt igényelhetnek. Ezt a bonyolultságot rugalmas IoT architektúrával kell kezelni.

2.3. Eszközkezelő szolgáltatás

Az IoT-hálózat heterogén erőforrásokat és feladatokat tartalmaz. Az eszköz feladatkezelése azt jelenti, hogy telepítünk és futtatunk egy megfelelő alkalmazást az eszközön, erőforrásokat párosítunk annak funkcióihoz, és megőrizzük az eszköz metaadatait az eszköz állapotának fenntartása érdekében.

2.4. Adatkezelő szolgáltatás

A hálózaton és eszközkezelő szolgáltatáson kívül szükség van adatfolyam (streaming) és tárolási modellre. Ezen felül megfelelő adatfeldolgozási szűrésre van szükségünk, például az SQL lekérdezésekre.

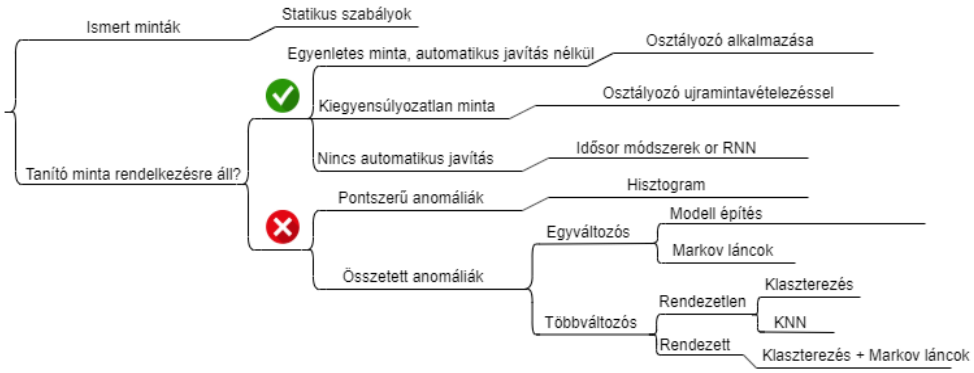
3. Az adatok kezelése

Az alábbiakban leírjuk a fő kihívásokat:
Adatmennyiség: optimalizált tárolási infrastruktúrára van szükségünk a Big Data típusú adatok számára.
Időérzékenység: valós idejű / kötegelt feldolgozás: A bejövő adatokat valós időben kell eltárolni. Alternatíva a kötegelt feldolgozás, amely több időt és nagyobb erőforrásokat igényel.
Heterogenitás: Előfordul, hogy az adatszerkezet nem egységes.
Kódolás / Dekódolás: Az adatot a forrásnál kódolni kell, a tárolóban dekódolni szükséges.

Adatfolyam-vezérlés: A bonyolultság miatt nyomon kell követnünk az összes adat átalakítást. Ennek többféle módja van, mint például az SQL kód vagy a grafikus pipe reprezentáció. A metaadat-kezelés alapvető fontosságú a hálózati állapot és az esetleges adatfolyam-optimalizálás kezelésében. Figyelemmel kell kísérni az adatforrások, például a gépek, a gyári környezet, az eszközök adatainak tulajdonságait is.
Adatminőség, átalakítás a használhatóság érdekében: A tárolóban hiányzó adatokkal és az adatforrással is kell foglalkoznunk; a minőségmenedzsmentnek automatizáltnak és átláthatónak kell lennie. Nagy historikus adattár létrehozása azt jelenti, hogy pontosan nyomon kell követni az idősorokat. Az adatok auditálhatósága a folyamat kritikus része, mivel az adatoknak gyakran üzleti értékei vannak.
A tárolási architektúra kiválasztása: Ez önmagában egy bonyolult feladat. Az első kérdés: melyik adatmodellt kell használni? A második kérdés az, hogy az SQL / NoSQL modellt jobb választani [9] és milyen tárolási infrastruktúrát célszerű használni. A megfelelő tárolási technológia kiválasztása a hosszú távú adatkezelési megoldáshoz nem könnyű feladat, mivel számos technikai és fogalmi döntést igényel. Vannak különféle idősorozatokat tartalmazó adatbázisok, például az effxDB vagy PostgreSQL időalapú DB kiterjesztéssel, vagy választhatunk fürttárolási (cluster) megoldást, például ilyen a HDFS, GlusterFS vagy Cassandra [10]. A „legjobb adattárolási módszer” megtalálása feladatfüggő, a fenti megoldások mindegyikének megvannak az előnyei és hátrányai.
Az adatkezelési folyamatok menedzselése: Az adatok tárolása után megfelelő eszközökre van szükségük az adatok átalakításához és elemzéséhez. Ez lehet klasszikus elemzés vagy komoly gépi tanulási modell. Az ilyen szolgáltatást nyújtó eszközöket a tárolási technológián felül lokálisan vagy a felhőben is üzemeltetni és kezelni kell.

3.1. Ritka események és rendellenességek

Határozzuk meg, mit értünk a **ritka esemény** és **rendellenesség fogalma** alatt. *Ritka esemény* egy marginális előfordulású időbeli esemény, azaz olyan esemény, amely túl kevés alkalommal fordul elő egy adott átlagos időrend (norma) vonatkozásában. A *rendellenesség/anomália* egy olyan ritka esemény, amelynek más jellemzői vannak, mint a normális eloszlású eseményeknek, vagyis anomáliát egy olyan folyamat generál, amely viselkedése eltér a normál(is)nak tulajdonítható folyamatoktól. Egy anomáliát gyakran az outlier analógjának tekintik. Az outlier (külső érték/ kívül eső érték) egy olyan adatpont, amely tulajdonságai alapján az adatkészlet természetes állapotához képest eltérést mutat.



1. ábra. *Anomáliák detektálásának általános folyamata.[12]*

A ritka események gyakran katasztrófákra (vagy összeomlásra) utalnak, így egy ritka esemény bekövetkezése gyakran hatalmas költséggel jár, viszont nagyon nehéz egy ilyen esemény bekövetkezésének előre jelzése. A klasszikus módszerek nem működnek jól kiegyensúlyozatlan adathalmazokon, ahol a betanító mintákban a normális adatok száma sokkal nagyobb, mint a rendellenesek száma (pl. 1 millió adatból 2-3 darab jelöl rendellenes viselkedést). A kiegyensúlyozatlan adatok kezelésének egyik megoldása a rendellenes adatok számának mesterséges növelése és a normál adatok számának csökkentése. A másik lehetséges megoldás az eltérő súlyozás alkalmazása. A következő táblázat a rendellenességek észlelésének algoritmusainak átfogó listáját [11] tartalmazza:

1. táblázat. *Gépi tanulási folyamat outlier felismerésére*

	Gépi tanuló algorit-mus	**Hiányzó adat becslése**	**Elosztott / központosított**	**Komplexitás**	**Cél(ok)**
Bayesian Belief Networks	Bayesian	Igen	Elosztott	Alacsony	Outlier felismerés
Outlier Detection k-NN módszerrel	k-NN	Igen	Elosztott	Közepes	Elosztott outlier felismerés
Detecting Selective Forwarding Attacks Using SVM	SVM	Nem	Központosított	Közepes	Fekete lyuk felismerése
Distributed Outlier Detection SVMs módszerrel			Elosztott	Alacsony	Outlier felismerés
Online Outlier Detection			Központosított	Közepes	Online outlier felismerés
Intrusion Detection System			Központosított	Magas	Behatolás felismerés SVM algoritmussal
Linear Outlier Detection			Elosztott	Közepes	Adaptív outlier felismerés
SOM módszeren alapuló elemzés	SOM	Nem	Elosztott	Közepes	Anomáliaszerű viselkedés felismerése

Magyarázat:

- Hiányzó adat becslése: A valós példákban az adatminőség kérdéses. A hiányzó adatok egy automatizált folyamat részei, és nem mindig garantálhatjuk az adatok minőségét működés közben.
- Elosztott / központosított: Az előrejelzés eredménye eloszthető vagy központosítható.
- Komplexitás: azt jelzi, hogy milyen nehéz a probléma számítási szempontból
- A mechanizmus: az outlier felismerése adaptív vagy elosztott is lehet.
- Fekete lyuk
- Behatolás

3.2. Módszerek rendellenességek észlelésére

Az alábbiakban felvázolunk néhány klasszikus rendellenesség-észlelési módszert:

Elosztott "kiugró érték" keresés Bayes-féle hálókkal

Bayes féle háló alkalmas arra, hogy egy mintáról eldöntse, hogy egy adott osztályba tartozik-e vagy sem. Ha egy megfigyelés értéke kiugró (vagy sem), könnyebben eldönthető, ha több érzékelő együttes mérését is figyelembe vesszük és több háló együttes korrelációját vesszük figyelembe. [13]

Kiugró értékek keresése k-NN algoritmussal

Ez a szomszédság alapú osztályozási módszer egyfajta "példákon alapuló" vagy más szóval "nem általánosító" tanulási módszer. Nem épít általános következtető modellt, hanem csupán csak tárolja a tanítási mintákat. A kérdéses minta ahhoz az adatosztály tartozik, amely a leginkább képviseli a pont legközelebbi szomszédait, így azok az értékek lesznek kiugróak, amelyek egyik korábbi osztályhoz sem tartoznak. [14][18]

Elosztott "kiugró érték" keresés SVM-el

Kiugró értékek keresésére alkalmasnak bizonyult a Support Vector Machine (SVM) módszer is. Ennek egyik módosítása az un. "egyosztályos SVM" (one class SVM), ami alkalmas annak eldöntésére, hogy egy új megfigyelés ugyanabba az eloszlásba tartozik-e, mint a már meglévő megfigyeléseink. [15]

Deep learning alapú anomália keresés

A "sok rétegű" neurális hálózatok is alkalmasak anomália keresésre. Zhang és társai [16] egy nem felügyelt típusú többváltozós idősorok analízisét mutatják be.

Fuzzy módszerek

[17] Inkrementálisan felépített fuzzy modellekre mutat példát.

4. Összefoglalás

A cikkben rövid összefoglalást mutattunk be az IoT rendszerek informatikai, infrastrukturális és adatbányászati, adatfeldolgozási vonatkozásairól. Érzékelők, szenzorok ipari alkalmazása már évtizedek óta ismert, de ezek tömeges elterjedése, valamint az anomáliák és ritka események felismerésének igénye számos új kihívást teremt a kutatóknak.

5. Köszönetnyilvánítás

A cikkünkben ismertetett kutatómunka az Európai Unió és a magyar állam támogatásával, az Európai Regionális Fejlesztési Alap társfinanszírozásával, a GINOP-2.3.4-15-2016-00004 projekt keretében valósult meg, a felsőoktatás és az ipar együttműködésének elősegítése céljából.

Irodalom

[1] Kayal, Paridhika; Perros, Harry. A comparison of IoT application layer protocols through a smart parking implementation. In: 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN). IEEE, 2017. p. 331-336. **https://doi.org/10.1109/ICIN.2017.7899436**

[2] Al-Sarawi, Shadi, et al. Internet of Things (IoT) communication protocols. In: 2017 8th International conference on information technology (ICIT). IEEE, 2017. p. 685-690. **https://doi.org/10.1109/ICITECH.2017.8079928**

[3] Fysarakis, Konstantinos, et al. Which iot protocol? comparing standardized approaches over a common m2m application. In: 2016 IEEE Global Communications Conference (GLOBECOM). IEEE, 2016. p. 1-7. **https://doi.org/10.1109/GLOCOM.2016.7842383**

[4] WAMP: https://wamp-proto.org/, 2020

[5] MQTT: https://mqtt.org/, 2020

[6] Ankit Patel:Top 5 Surprising Facts Everyone Should Read About IOT, 2018, http://customerthink.com/top-5-surprising-facts-everyone-should-read-about-iot/

[7] Mohammadi, Mehdi, et al. Deep learning for IoT big data and streaming analytics: A survey. IEEE Communications Surveys & Tutorials, 2018, 20.4: 2923-2960. **https://doi.org/10.1109/COMST.2018.2844341**

[8] Mahdavinejad, Mohammad Saeid, et al. Machine learning for Internet of Things data analysis: A survey. Digital Communications and Networks, 2018, 4.3: 161-175. **https://doi.org/10.1016/j.dcan.2017.10.002**

[9] Fatima, Haleemunnisa; Wasnik, Kumud. Comparison of SQL, NoSQL and NewSQL databases for internet of things. In: 2016 IEEE Bombay Section Symposium (IBSS). IEEE, 2016. p. 1-6. **https://doi.org/10.1109/IBSS.2016.7940198**

[10] Díaz, Manuel; Martín, Cristian; Rubio, Bartolomé. State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. Journal of Network and Computer applications, 2016, 67: 99-117. **https://doi.org/10.1016/j.jnca.2016.01.010**

[11] Alsheikh, Mohammad Abu, et al. Machine learning in wireless sensor networks: Algorithms, strategies, and applications. IEEE Communications Surveys & Tutorials, 2014, 16.4: 1996-2018. **https://doi.org/10.1109/COMST.2014.2320099**

[12] Perera, S.: Introduction to Anomaly Detection: Concepts and Techniques, 2015. https://iwringer.wordpress.com/2015/11/17/anomaly-detection-concepts-and-techniques/

[13] Janakiram, D., Reddy, V. A., & Kumar, A. P. (2006, January). Outlier detection in wireless sensor networks using Bayesian belief networks. In 2006 1st International Conference on Communication Systems Software & Middleware (pp. 1-6). IEEE. **https://doi.org/10.1109/COMSWA.2006.1665221**

[14] Yang, P., & Huang, B. . KNN based outlier detection algorithm in large dataset. In 2008 International Workshop on Education Technology and Training & 2008 International Workshop on Geoscience and Remote Sensing (Vol. 1, pp. 611-613). IEEE. **https://doi.org/10.1109/ETTandGRS.2008.306**

[15] Shahid, Nauman; Naqvi, Ijaz Haider; Qaisar, Saad Bin. One-class support vector machines: analysis of outlier detection for wireless sensor networks in harsh environments. Artificial Intelligence Review, 2015, 43.4: 515-563. **https://doi.org/10.1007/s10462-013-9395-x**

[16] Zhang, Chuxu, et al. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2019. p. 1409-1416. **https://doi.org/10.1609/aaai.v33i01.33011409**

[17] D. Vincze, Sz. Kovács: Incremental rule base creation with fuzzy rule interpolation-based Q-learning, Studies in Computational Intelligence, Volume 313, pp. 191-203., 13 p. (2010) **https://doi.org/10.1007/978-3-642-15220-7_16**

[18] Szabó, N. P., Nehéz, K., Hornyák, O., Piller, I., Deák, Cs., Hanzelik, P. P., Kutasi, Cs., Ott, K.: Cluster analysis of core measurements using heterogeneous data sources: An application to complex Miocene reservoirs. Journal of Petroleum Science and Engineering pp. 575-585, 2019 **https://doi.org/10.1016/j.petrol.2019.03.067**