

Számítási intelligencia alkalmazási lehetőségei ipar 4.0 környezetben

Írta:

Szűcs Balázs

Témavezető:

DR. BALLAGI ÁRON

Tanszékvezető egyetemi docens

Széchenyi István Egyetem, Automatizálási és Mechatronikai Tanszék



PH.D. DOKTORI ÉRTEKEZÉS

Infrastrukturális Rendszerek Modellezése és Fejlesztése
Multidiszciplináris Műszaki Tudományi Doktori Iskola

Győr
2023

„An investment in knowledge pays the best interest.”

Benjamin Franklin

Köszönetnyilvánítás

Ezután szeretnék köszönetet mondani témavezetőmnek, Dr. Ballagi Áronnak aki szakmailag támogatott a disszertáció elkészítésében, az AUDI HUNGARIA Zrt. munkatársainak és a közvetlen csapatom tagjainak, akik nélkül nem sikerült volna a gyakorlati alkalmazások implementálása. Szeretném megköszönni családom biztatását és szerető feleségem türelmét, kitartását és támogatását ami nélkül e munka nem jöhetett volna létre. Továbbá köszönöm a támogatást rokonaimnak és barátaimnak akik bátorítottak munkám során. Szeretném munkám nagyapám, Dr. Bacsó András emlékének ajánlani, akitől a kíváncsiságot, a tudásvágyat, a világ felé való nyitottságot és a pozitív gondolkodást tanultam.

Kivonat

Kutatásomat az AUDI HUNGARIA Zrt. járműhajtás gyártás területén végeztem. Munkám célja a számítási intelligencia ipari alkalmazási lehetőségeinek felkutatása és bevezetése a gyártásba Ipar 4.0 környezetben.

Munkámat irodalomkutatással kezdtem, melynek során feldolgoztam az ipar 4.0-val, a gépi tanulással és különböző mesterséges intelligencia módszerekkel foglalkozó nemzetközi szakirodalmat és folyóiratcikkeket. Az irodalomkutatás során összefoglaltam a kutatási témámban releváns eljárásokat, módszereket és információkat, melyek alapján azonosítottam a potenciális felhasználási lehetőségeket és kiválasztottam a munka során felhasznált modelleket.

Az adatgyűjtés során szembesültem az életkorban és technológiában heterogén géppark okozta nehézségekkel, ezért egy standardizált adatgyűjtési eljárást és szabályrendszert dolgoztam ki a kihívások leküzdésére. A problémák ismertetése után bemutatom a kidolgozott adatgyűjtési keretrendszert.

Az adatgyűjtés nehézségeinek és a javasolt adatintegrációs keretrendszer bemutatása után rávilágítok a mesterséges intelligencia ipari alkalmazásainak kihívásaira, majd röviden elemezem a kihívásokkal kapcsolatos kérdéseket, azonosítom a szükséges résztvevőket, majd végül javaslatot teszek egy munkamenetre az MI alkalmazások sikeres ipari bevezetésére.

A dolgozatomban prezentálom a gépi tanulási algoritmusok, mint a K-Means és a DBSCAN lehetséges ipari felhasználásait anomális felismerésre, valamint gyakorlati példákon keresztül ismertetem azokat. A következő fejezetben az anomaliafelismerés eredményeire építve bemutatom, hogyan lehet az LSTM és az autoencoder neurális hálózatokat folyamatfelügyeletre használni. A numerikus, mérési adatokkal tanított modellek gyakorlati megvalósításán túl ismertetem a konvolúciós neurális hálózatok ipari alkalmazását nemdeterminisztikus zavarok eliminálására képfeldolgozási alkalmazások esetén. Végül megmutatom, hogyan lehet a természetes nyelvfeldolgozás segítségével hatékonyabbá tenni a karbantartási tevékenységek tervezését és végrehajtását.

Disszertációm az általam bemutatott modellek és metódusok összegzésével zárom, mely során bemutatom a saját eredményeket, végül pedig kitekintést adok a további kutatási és fejlesztési irányokra.

Abstract

I did my research in the field of vehicle drive production at AUDI HUNGARIA Zrt. The purpose of my work is to search for the industrial application possibilities of computational intelligence and introduce them into production in an Industry 4.0 environment.

I started my work with literature research, during which I processed international literature and journal articles dealing with Industry 4.0, machine learning and various artificial intelligence methods. During the literature research, I summarized the procedures, methods and information relevant to my research topic, based on which I identified the potential uses cases and selected the models used in my work.

During the data collection, I faced difficulties caused by a heterogeneous fleet of machines in terms of age and technology, so I developed a standardized data collection procedure and set of rules to overcome the challenges. After describing the problems, I present the developed data integration framework.

After presenting the difficulties of data collection and the proposed data integration framework, I will highlight the challenges of industrial applications of artificial intelligence, then briefly analyze the issues related to the challenges, identify the necessary participants, and finally propose a workflow for the successful industrial introduction of AI applications.

In my thesis, I show the possible industrial uses of machine learning algorithms such as K-Means and DBSCAN for anomaly detection, and I explain them through practical examples. In the next chapter, based on the results of anomaly detection, I show how LSTM and autoencoder neural networks can be used for process monitoring. In addition to the practical implementation of numerical models trained with measurement data, I show the industrial application of convolutional neural networks to eliminate non-deterministic disturbances in image processing applications. Finally, I show how natural language processing can be used to make the planning and execution of maintenance activities more efficient.

I conclude my dissertation with a summary of the models and methods I presented, during which I present my own results, and finally I give an outlook on further research and development directions.

Tartalomjegyzék

1. Bevezetés	9
2. A kutatási téma és annak időszerűsége	9
3. Az alkalmazott módszertan bemutatása	10
4. Előzmények	11
5. Irodalmi áttekintés	11
5.1 Ipar 4.0.....	11
5.2 Kiber-fizikai rendszerek	12
5.3 Dolgok internete (Internet of Things).....	13
5.4 Felhőalapú számítástechnika	14
5.5 Kognitív számítástechnika.....	15
5.6 Mesterséges intelligencia.....	16
5.7 Főkomponens analízis (PCA).....	18
5.8 Lineáris diszkriminancia analízis (LDA)	19
5.9 K-átlag	20
5.10 A DBSCAN	21
5.11 A perceptron	22
5.12 Aktivációs függvények	23
5.13 Mesterséges neurális hálózatok (ANN).....	27
5.14 Visszacsatolt neurális hálózatok (RNN).....	29
5.15 Long-Short Term Memory (LSTM) hálózatok	31
5.16 Konvolúciós neurális hálózatok (CNN)	33
5.17 Autoencoder hálózatok	34
5.18 Felügyelet nélküli tanulás.....	36
5.19 Felügyelt tanulás.....	37
5.20 Hibavisszaterjesztés (backpropagation)	38
5.21 Modellek teljesítményének javítása	38
5.22 Univerzális approximátor tétel	39
5.23 Természetes nyelvi feldolgozás (Natural Language Processing).....	40
5.24 Python.....	43
5.25 TensorFlow	43
5.26 Optimalizációs eljárások	45
5.27 Scikit-learn	46
5.28 A gépi tanulás korlátai.....	46
5.29 Supervisory Control and Data Acquisition (SCADA)	47

6.	Környezet bemutatása	48
6.1	Az Ipar 4.0 az AUDI HUNGARIA Zrt-nél	48
6.2	SCADA rendszerek	48
6.3	Hálózati architektúra	50
6.4	Az Ipar 4.0 és a SCADA szerepe a kutatásomban	51
7.	Az ipari adatgyűjtés kihívásai	52
8.	Saját eredmények.....	54
9.	Kutatásom célja	54
10.	Adatintegrációs keretrendszer	54
10.1.1	Szabványosított névkonvenció	56
10.1.2	Szabványosított üzenetstruktúra.....	57
10.1.3	Adatelosztási réteg	59
10.1.4	Az adatintegrációs keretrendszer gyakorlati megvalósítása.....	59
10.1.5	Az adatintegrációs keretrendszer termelési adatok gyűjtésére.....	65
10.1.6	Számítási intelligencia ipari alkalmazásainak kihívásai	66
10.1.7	Eredmények összegzése	70
I.	Tézis.....	72
11.	Számítási intelligencia ipari felhasználási esetei.....	73
11.1.1	Gyártási anomálfelismerés gépi tanulási módszerekkel.....	74
11.1.2	Eredmények összefoglalása.....	81
II.	Tézis.....	82
11.1.3	Folyamatfelügyelet LTSM és autoencoder hálózatokkal.....	83
11.1.4	Eredmények összefoglalása.....	87
III.	Tézis.....	88
11.1.5	Nemdeterminisztikus zavarok eliminálása ipari képfeldolg. alkalmazásokban. 89	
11.1.6	Eredmények összefoglalása.....	95
IV.	Tézis.....	96
11.1.7	Karbantartási hatékonyság növelése természetes nyelvfeldolgozással	97
11.1.8	Eredmények összefoglalása.....	102
V.	Tézis.....	103
12.	Új tudományos eredmények összefoglalása.....	104
I.	Tézis	105
II.	Tézis	106
III.	Tézis	106
IV.	Tézis	106

V. Tézis	107
13. Konklúzió, további feladatok	108
14. Irodalomjegyzék.....	110
15. Saját publikációk.....	122

1. Bevezetés

2. A kutatási téma és annak időszerűsége

Kutatási témám a számítási intelligencia ipari alkalmazási lehetőségei ipar 4.0 környezetben. A számítási kapacitások és az adatok, adatforrások növekedésével a mesterséges intelligencia módszerek egyre nagyobb hangsúlyt kapnak mindennapi életünkben. A különböző szórakoztatóipari felhasználások, ajánló rendszerek, intelligens asszisztensek, önvezető járművek és intelligens robotok, valamint a szociális média profilalkotási eljárásai és célzott hirdetései is különféle MI módszereket alkalmaznak. Munkám célja ezen mesterséges intelligencia módszerek gyakorlati ipari felhasználási eseteinek felkutatása és alkalmazása.

A XXI. század elején végbement negyedik ipari forradalomnak, azaz az Ipar 4.0-nak köszönhetően robbanásszerű növekedésnek indult az ipari automatizálásban a különböző intelligens szenzorok, kommunikációs megoldások és a dolgok internetének alkalmazása, valamint az automatizálási komponensek hálózatba kapcsolása. Az említett folyamatnak és alkalmazott új technológiáknak hála rengeteg adat áll rendelkezésünkre az ipari folyamatokról, paramétereikről és berendezések állapotáról.

A rendelkezésre álló adatokkal olyan intelligens modellek hozhatóak létre, melyek segítségével a berendezések meghibásodása előre jelezhető, a folyamatfelügyeleti- és minőségbiztosítási rendszerek hatásfoka javítható, ezáltal a berendezések rendelkezésre állása, valamint a gyártási minőség színvonala növelhető. A minőség és a rendelkezésre állás növelésével a gyártási, karbantartási és selejt költségek csökkenthetőek és jelentős megtakarítások érhetőek el a szériagyártásban.

Munkám célja tehát olyan mesterséges intelligencia módszerek felkutatása, kidolgozása és bevezetése a gyártásba, melyekkel jelentős megtakarítások érhetőek el. A költségek csökkentésével és a berendezések, folyamatok hatásfokának növelésével nem csak anyagi megtakarítás és magasabb profit realizálható, hanem szervesen hozzájárulnak a termelő vállalatok fenntartható működéséhez.

Kutatásom során megtapasztaltam az ipari adatgyűjtés nehézségeit, és ezek leküzdésére adatgyűjtési eljárást dolgoztam ki, valamint különböző mesterséges intelligencia módszereket felhasználva olyan modelleket hoztam létre, melyek segítségével a karbantartási költségek és a selejtek aránya csökkenthető, valamint a gyártott termékek minősége növelhető a gyakorlatban.

3. Az alkalmazott módszertan bemutatása

Kutatásom irodalomkutatással kezdtem, melynek során feldolgoztam az ipar 4.0-val, a gépi tanulással és különböző mesterséges intelligencia módszerekkel foglalkozó nemzetközi szakirodalmat és folyóiratcikkeket. Az irodalomkutatás során összefoglaltam a kutatási témámban releváns eljárásokat, módszereket és információkat, melyek alapján azonosítottam a potenciális felhasználási lehetőségeket és kiválasztottam a munkám során felhasznált modelleket.

A következő lépésben adatgyűjtést végeztem a kiválasztott modellek tanításához és az alkalmazási esetek kiértékeléséhez és érvényesítéséhez. Az adatok az AUDI HUNGARIA Zrt. termelési folyamataiból gyűjtöttem. Az adatgyűjtés során szembesültem az életkorban és technológiában heterogén géppark okozta nehézségekkel, ezért egy standardizált adatgyűjtési eljárást és szabályrendszert dolgoztam ki a kihívások leküzdésére. A problémák ismertetése után bemutatom a kidolgozott adatgyűjtési keretrendszert.

A gyűjtött adatok előkészítéséhez, feldolgozásához, valamint a modellek elkészítéséhez Python nyelvet, a scikit gépi tanulási csomagot, valamint a Google által fejlesztett TensorFlow keretrendszert használtam.

A munkám végén bemutatom az általam létrehozott modellek gyakorlati alkalmazásait, majd összegzem az eredményeket. Dolgozatomat egy kitekintéssel zárom, melyben vázolom a további potenciális kutatásokat, fejlesztési irányokat.

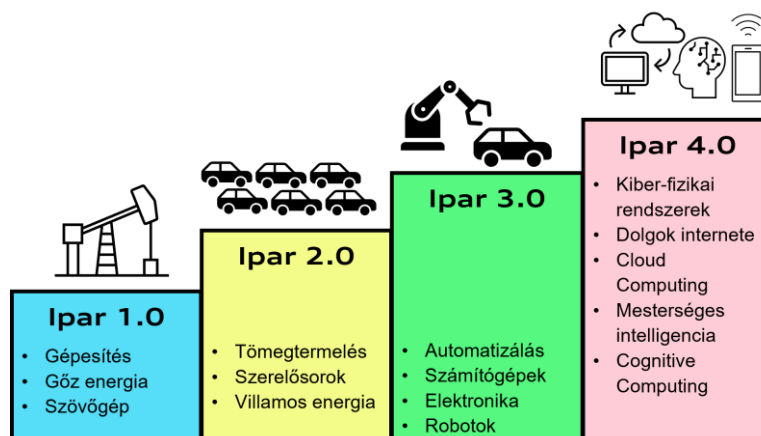
4. Előzmények

5. Irodalmi áttekintés

A következő fejezetekben a disszertáció elkészítése során felhasznált releváns szakirodalom kerül összegzésre és bemutatásra.

5.1 Ipar 4.0

Az Ipar 4.0, másik közismert nevén a negyedik ipari forradalom egy ipari automatizálási trend. A kifejezés 2011-ben jelent meg a először német nyelvterületen, a Hannover Messe-én [1]. Az Ipar 4.0 megoldások célja az adatcsere megvalósítása a különböző termelő berendezések és folyamatok között. Az adatok kiértékelése segítségével a gyártási folyamatok egyszerűsíthetők, a működési hatékonyság növelhető. Fő célkitűzései [2] az adatforrások, szenzorok, rendszerek és emberek összekapcsolása, az adatok transzparenciájának megteremtése, a felhasználók technikai támogatása a döntéshozatalban és a decentralizált, autonóm döntéshozó rendszerek kialakítása. Az Ipar 4.0-át alkalmazó gyárak másik megnevezése az okosgyár (*Smart Factory*), melynek elemei négy csoportba [3] sorolhatók, ezek a kiber-fizikai rendszerek (*Cyber-Physical Systems, CPS*), a dolgok internete (*Internet of Things, IoT*), a felhőalapú számítástechnika (*Cloud Computing*) és a kognitív számítástechnika (*Cognitive Computing*).



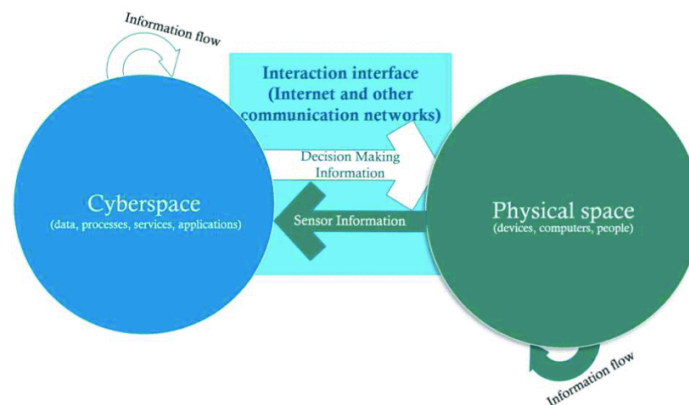
1. ábra: Az ipar fejlődése

5.2 Kiber-fizikai rendszerek

A kiber-fizikai rendszerek (CPS) érzékelőket, berendezéseket, információs rendszereket és számítási erőforrásokat integrál a teljesen értékláncon keresztül, melyek gyakran túlnyúlnak a vállalati vagy üzleti folyamatokon [4]. A CPS rendszerek decentralis, heterogén eszközökből álló, lazán összekapcsolt fizikai és digitális rendszerek összessége, melyek a fizikai folyamatok és rendszerek mért jelei alapján, a számítási kapacitásokat felhasználva befolyásolják a forrás folyamatokat és rendszereket. A szenzorok által mért értékeket a digitális aggregátorok feldolgozzák és az aktuátorok segítségével képesek beavatkozni a forrásfolyamatba. Ennek tipikus példái az intelligens épületek és gyártócellák, robotok és az önvezető járművek.

A kiber-fizikai rendszerek fő rétegei [4]:

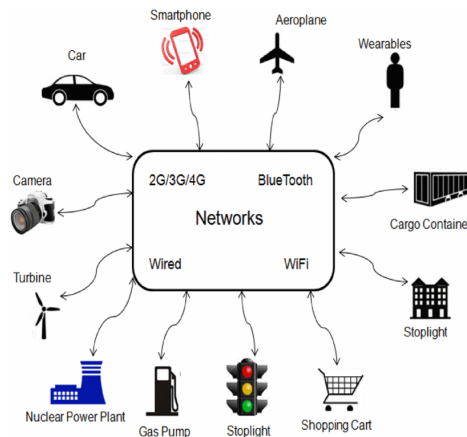
- Fizikai réteg
- Digitális réteg
- Interfész réteg a fizikai és a digitális réteg között
- Interfész réteg a digitális és fizikai réteg és a humán felhasználók között



2. ábra: Kiber-fizikai rendszerek koncepcionális ábrája [4]

5.3 Dolgok internete (Internet of Things)

A dolgok internete (*IoT*) olyan eszközök összesége, amelyek egymással, a fizikai réteg eszközeivel vagy más számítási kapacitásokkal, például felhőalapú szolgáltatásokkal állnak összeköttetésben. Az IoT eszközök kommunikációjuk során szabványos, interoperábilis kommunikációs protokollokat használnak, melyek célja az adatok gyűjtése, megosztása és/vagy befolyásolja az egyes folyamatokat a hatékonyság javítása érdekében [5][6]. Az IoT eszközök elengedhetetlen részei az ipari CPS rendszereknek. Az IoT eszközök az egymás közötti kommunikációhoz elsősorban IP (*Internet Protocol*) alapú kommunikációt használnak, melyhez nincs szükség emberi beavatkozásra [6].

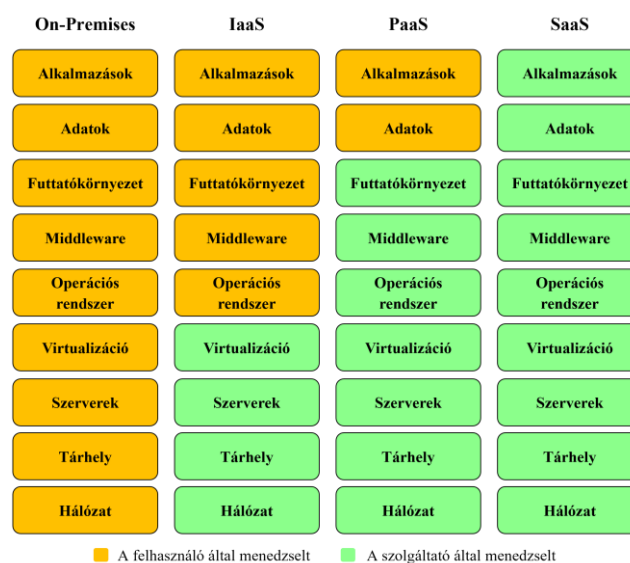


3. ábra: IoT eszközök [7]

Az IoT eszközök közé tartoznak például az okostelefonok, tabletetek, intelligens eszközök és szenzorok, amik az IP alapú kommunikáció mellett más vezeték nélküli technológiákat is alkalmazhatnak, mint például az RFID-t (*Radio-frequency identification*), Bluetooth-t és QR (*Quick Response*) kódokat az egymás közötti adatcserehez.

5.4 Felhőalapú számítástechnika

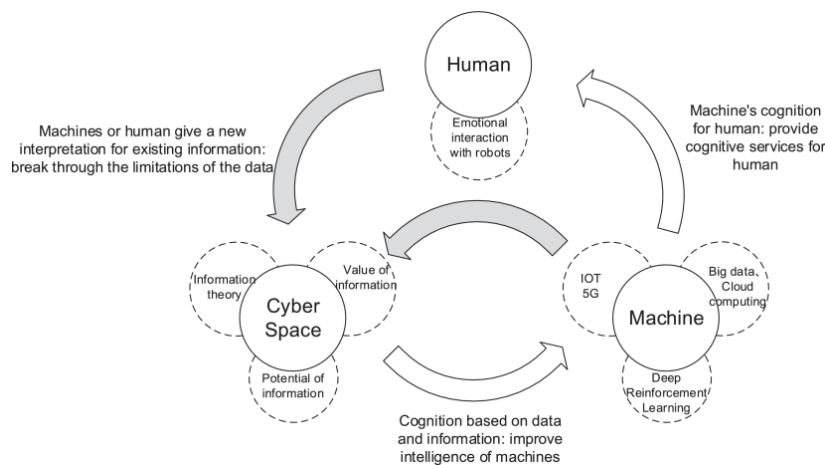
A CPS és IoT eszközök hatalmas mennyiségű adatot állítanak elő, melyek tárolásához és feldolgozásához nagy tároló- és számítási kapacitásra van szükség, részben erre kínál megoldást a felhőalapú számítástechnika vagy másnéven cloud computing [8]. A cloud szolgáltatások fő jellemzői közé tartozik az agilitás, az elaszticitás és a költséghatékonyság. A felhőalapú számítástechnika igény alapú, globálisan és azonnal hozzáférhető (agilitás), automatikusan skálázható (elaszticitás), nagy rendelkezésre állású tárolási és számítási kapacitást tesz elérhetővé, mellyel elkerülhető a lokális (on-site, on-premises) adatbázisok és kapacitások beszerzési és üzemeltetési költségei (költséghatékonyság). A cloud fő szolgáltatási az infrastruktúra szolgáltatás (*Infrastructure as a Service, IaaS*), mely különböző hálózati megoldásokat, virtuális és dedikált hardvereket biztosít az IT feladatok ellátáshoz a felhasználóknak, a platformszolgáltatás (*Platform as a Service, PaaS*), mely menedzselt infrastruktúra szolgáltatást biztosít a különböző felhasználói alkalmazások üzemeltetéséhez úgy, hogy a felhasználónak nem kell az alpinfrastruktúra üzemeltetésével foglalkoznia, illetve a szoftverszolgáltatás (*Software as a Service, SaaS*), mely különböző szolgáltatók által üzemeltett, a felhőben futó és a felhasználók számára elérhető szoftvereket jelent. Az PaaS és a SaaS közé tartoznak a különböző előre elkészített gépi tanulási környezetek és előre tanított mesterséges intelligencia modellek. A piacon különböző privát, valamint mindenki által elérhető publikus felhőszolgáltatások vehetők igénybe, ilyenek például az Amazon Web Services (AWS), a Google Cloud Platform (GCP) vagy a Microsoft Azure [9].



4. ábra: Üzemeltetési feladatok On-Premises és Cloud szolgáltatások esetén

5.5 Kognitív számítástechnika

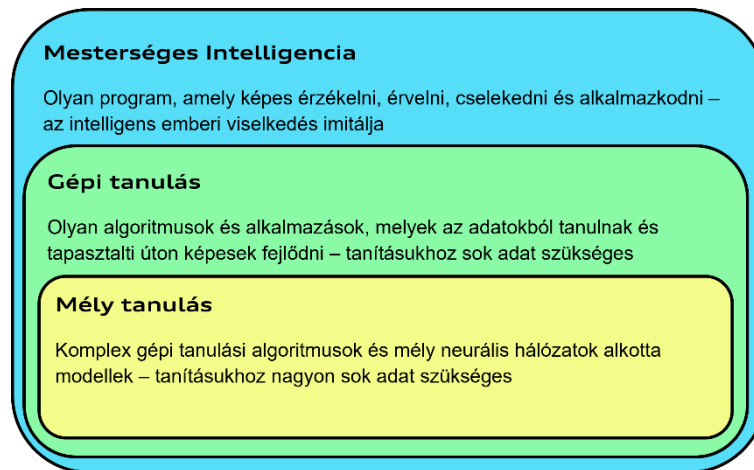
A kognitív számítástechnika (*Cognitive Computing, CC*) a mesterséges intelligencián (*Artificial Intelligence, AI*) és jelfeldolgozáson alapuló technológiai gyűjtőfogalom, melyet a kiber-fizikai rendszerek elterjedése [10], valamint a kognitív informatikai, neurális informatikai és a természetes intelligencia kutatások és technológiai fejlődése [11] hívott életre. A CC olyan technológiákon alapul, mint a gépi tanulás, az érvelés, a természetes nyelvi feldolgozást (*Natural Language Processing, NLP*), a beszédfelismerést, gépi látás, az ember-számítógép interakció, a párbeszédet és a narratív szöveggenerálás [12]. A kognitív rendszerek olyan hardverek és/vagy szoftverek kombinációját jelenti, amely az emberi agy viselkedését és funkcionalitását hivatottak utánozni [13]. Az ilyen rendszerek célja, hogy az adatokból tanult minták és viselkedési formák, valamint emberi és/vagy gépi útmutatás segítségével komplex döntéshozatali folyamatokkal is meg tudjanak birkózni autonóm módon, vagy a felhasználót támogassák a döntéshozatalban [14], ezáltal gyors reakciókat és hatékony munkavégzést tesznek lehetővé.



5. ábra: Emberközpontú kognitív ciklus [14]

5.6 Mesterséges intelligencia

A mesterséges intelligencia (*MI*, *Artificial Intelligence*, *AI*) olyan, gépek által alkalmazott intelligencia, amely az észlelt információkat szintetizálva következtetéseket tud levonni. A kifejezést John McCarthy alkotta meg [15] és 1956-ban megtartotta az első AI konferenciát. Az MI logikai, matematikai, számítástechnikai, pszichológiai-, kognitív- és idegtudományi, valamint evolúciós elméleteket alkalmazó algoritmusok és modellek. Az AI célja olyan intelligens rendszerek létrehozása, amely az emberi intelligenciát utánozva képes komplex és valós problémákat megérteni és/vagy megoldani. A mesterséges intelligenciát két alterületre szokták bontani, ezek a gépi tanulás és a mélytanulás (6. ábra).



6. ábra: A mesterséges intelligencia és alterületei

Gépi tanulás esetén olyan modellekről vagy algoritmusokról beszélünk, melyeknek egy vagy több tanítható, hangolható paramétere van és a tanítás során ezen a paramétereket optimalizáljuk, hogy a hibát csökkentve minél pontosabb megoldást kapjunk az adott problémára. Tipikus alkalmazási esetei [16]:

- Adatosztályozás – az adatokat különböző kategóriákba sorolja
- Regressziós problémák, modellezés – ok-okozati összefüggések feltárása, adatkapcsolatos leképezése és adatok előrejelzése
- Arcdetektálás és felismerés – személyek észlelése és azonosítása
- Vizuális érzékelés – környezeti/képi elemek értelmezése
- Kép- és beszédfeldolgozás – képi és nyelvi információk kinyerése, képek, valamint írott és szóbeli beszéd szintetizálása

- Adaptív rendszerek – adaptív viselkedés, mely során a rendszer a múltbéli tapasztalatok alapján szabályokat képez
- Adatbányászat – (rejtett) adatok kinyerése tanuló algoritmusokkal
- Problémamegoldás – Megoldás keresése adott problémára logika, keresési fák és következtetések alapján
- Anomália felismerés – Anomáliák, feltűnő adatok, folyamatok és események észlelése
- Automatizálás

A mélytanulás a gépi tanulás speciális esete, melyhez többretegű (mély) mesterséges neurális hálózatokat [17] (*Artificial Neural Networks, ANN*) alkalmaznak. A mesterséges neurális hálózatok felépítése az emberi agy felépítését imitálja, az adatokat neuronok és szinapszisok segítségével dolgozza fel. Az ANN modellek tanítása a klasszikus gépi tanulási algoritmusok képest relatív nagy adathalmazokat és számítási kapacitást igényel.

5.7 Főkomponens analízis (PCA)

A főkomponens analízist (*Principal Component Analysis, PCA*) egy többváltozós adatredukciós statisztikai eljárás, amit gyakran használnak a gép tanulásban az adatok dimenziójának csökkentésére. A módszert Karl Pearson alkotott meg 1901-ben [18]. A eljárás lényege, hogy az eredeti adatokat egy új koordináta-rendszerbe transzformálja, amiben változókat a varianciájuk szerint csökkenő sorrendbe rendezik [19]. Az új adatok, amelyeket főkomponenseknek neveznek, egymástól független lineáris kombináció az eredeti változóknak. A PCA alapja az eredeti adatok kovariancia mátrixának spektrális felbontása, hatékonyan kiszámítható az szinguláris érték felbontás (*Singular Value Decomposition, SVD*) algoritmus segítségével [19]. Az SVD segítségével az adatmátrix három mátrixra bontható, amelyekkel kiszámíthatók a főkomponensek. A PCA-nak egyik korlátja, hogy a változók között lineáris kapcsolatok feltételez, valamint hogy a főkomponensek nem mindig könnyen értelmezhetők. Ezen korlátok feloldására a Nemlineáris PCA és a Sűrűségi PCA kínál megoldást [22][23].

A klasszikus főkomponens analízis lépései és a matematikai algoritmus a következő [19][25]:

- 1) Az adatok középre igazítása a változók átlagának levonásával.
- 2) A középre igazított adatok kovarianciamátrixának kiszámítása.
- 3) Az kovarianciamátrix sajátvektorainak és sajátértékeinek meghatározása.
- 4) Az első k sajátvektor kiválasztása, amelyekhez a legnagyobb sajátértékek tartoznak, ahol k a csökkentett adatok kívánt dimenzióinak száma.
- 5) Az adatok projekciója a kiválasztott sajátvektorokra, hogy az új, alacsonyabb dimenziós ábrázolást megkapjuk.

Legyen X egy $n \times p$ méretű mátrix, ahol n az észlelések száma, p pedig a változók száma. Feltételezzük, hogy X középre van igazítva (azaz, minden oszlopának átlaga nulla).

Az első lépés a kovariancia-mátrix C kiszámítása:

$$C = \frac{1}{(n-1)} X^T X \quad (1)$$

ahol X^T a transzponáltja X -nek.

Ezután meghatározzuk a kovariancia-mátrix sajátvektorait és sajátértékeit:

$$Cv = \lambda v \quad (2)$$

ahol λ az sajátérték és v a sajátvektor.

A sajátvektorokat a sajátértékek csökkenő sorrendjében rendezzük. Kiválasztjuk a legnagyobb k sajátértékkel rendelkező sajátvektorokat, ahol k a csökkentett adatok kívánt dimenzióinak száma. Végül a középre igazított adatokat vetítjük a kiválasztott sajátvektorokra, hogy az új, alacsonyabb dimenziós ábrázolást megkapjuk:

$$Y = XV_k \quad (3)$$

ahol V_k a kiválasztott legnagyobb k sajátvektorokat tartalmazó mátrix, és Y az új $n \times k$ méretű mátrix, ami a csökkentett adatokat reprezentálja. A V_k alsó index azt jelzi, hogy csak a legnagyobb k sajátvektorokat választjuk ki.

5.8 Lineáris diszkriminancia analízis (LDA)

A Lineáris diszkriminancia analízis (LDA) [20] egy olyan statisztikai technika, amit gyakran alkalmaznak dimenziócsökkentésre és osztályozásra a gépi tanulásban és mintaillesztésben. Az LDA célja, hogy olyan lineáris kombinációt találjon a jellemzőkből, ami maximalizálja az eltérést a különböző osztályok között, miközben minimalizálja a varianciát az egyes osztályokon belül. Ellentétben a főkomponens analízissel (PCA), ami kizárólag a variancia maximalizálására összpontosít, az LDA figyelembe veszi az osztálycímkeket is, ami különösen hasznos a felügyelt osztályozási feladatokban.

Az LDA azzal dolgozik, hogy a kiinduló magas dimenziós teret egy alacsonyabb dimenziós térbe képezi le, ahol az osztályok közötti szeparáció optimális. Az elkészült lineáris diszkriminánsokat az osztályok közötti és az osztályokon belüli szórásokat összekombináló mátrix sajátvektoraként számolják ki. Az így kapott lineáris diszkriminánsok új jellemzőkké válnak, amiket az osztályozáshoz lehet használni.

Lényegében az LDA azt célozza, hogy maximalizálja az osztályok közötti variancia és az osztályokon belüli variancia arányát, ezzel növelve az osztályok elkülöníthetőségét. Ez különösen hatékony, amikor az osztályok jól meghatározottak és megkülönböztethetők. Az LDA széles körben használják különböző területeken, beleértve az képfelismerést, orvosi diagnózist és természetes nyelvfeldolgozást is.

A többosztályú lineáris diszkriminancia analízis (Multiclass Linear Discriminant Analysis vagy One-vs-All LDA) [21] egy kibővített változata a szokásos Lineáris Diszkriminancia Analízis (LDA) algoritmusnak, amely képes több osztály kezelésére egy osztályozási

problémában. A sztenderd LDA kétosztályos osztályozási feladatokra van tervezve, azonban sok valós világbeli probléma több mint két osztályt tartalmaz. A többosztályú LDA lehetővé teszi az LDA kiterjesztését több mint két osztály esetére.

5.9 K-átlag

A K-átlag vagy más néven K-Means egy felügyelet nélküli gépi tanulási algoritmus, melyet az adatok csoportosítására, klaszterezésére használnak. Az adatokat azok hasonlósága alapján K klaszterbe sorolja. Az algoritmus iteratív, minden adatpontot a hozzá legközelebbi klaszter középpontjához rendeli, majd a frissíti a klaszterek középpontját, mely során az adott klaszter középpontjához rendelt pontok átlagát veszi figyelembe [26]. Az algoritmus addig fut, amíg a középpontok már nem változnak vagy a futások száma eléri a beállított maximális iterációs számát.

A K-Means algoritmus érzékeny lehet a középpontok kezdeti helyzetére, ami miatt szuboptimális megoldásra juthat. A problémát úgy lehet kezelni, hogy több futtatást végzünk az algoritmussal különböző kezdeti középpontokkal, majd a legkisebb klaszterek közötti négyzetösszegű hibát választjuk [27].

A K-Means algoritmus széles körben használatos különböző területeken, mint például a gépi látás, a természetes nyelvfeldolgozás (NLP) és a bioinformatika. Az algoritmus hatásosságát befolyásolhatja a K értékének (klaszterek számának) megválasztása, valamint az adatok eloszlása és dimenziója [28].

Az optimális klaszterszám meghatározására a könyök-módszer vagy Elbow-method egy gyakran használt eljárás [29]. Az adatpontok és a hozzájuk tartozó klaszter középpontok közötti hiba négyzetösszege (*Error Sum of Squares, SSE*) alapján, valamint a klaszterszám függvényében történő ábrázolásával meghatározható egy könyökpont, amely a klaszterek számának növelése mellett a hibacsökkenés ütemének lelassulását mutatja [30]. A hibagörbe meredeksége egy bizonyos számú csoport bevezetése, azaz a könyökpont után drasztikusan csökken. Ez a könyökpont az ideális klaszterszámot jelenti, amelynek meghatározása az adott problématerülettől függően változhat [31].

A könyök-módszernek vannak korlátai, például a módszer nem működik jól a bonyolult, szabálytalan alakú adathalmazoknál és a könyökpont sem mindig azonosítható [32]. Ezért fontos a csoportképzés értékelésében más klaszterezési metrikák, például a sziluettérték használata az eredmények megbízhatósága érdekében [33].

5.10 A DBSCAN

A DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) egy népszerű, nem felügyelt tanulási algoritmus, amelyet adathalmazokban található csoportok azonosítására használnak [34]. Használata elsősorban bonyolult alakúak és eltérő sűrűségű adathalmazok esetén célravezető [35]. A DBSCAN algoritmus meghatározza a csoportok "magpontjait", majd az egyes magpontokhoz a felhasználó által meghatározott paraméterek alapján hozzárendeli az egyes adatpontokat, így csoportokat képez az adathalmazon belül [36]. A magpontok olyan pontok, melyeknek legalább n (minPts paraméter) számú szomszédos pontjuk van egy adott ε távolságon belül. A csoportok kiterjesztése úgy történik, hogy az algoritmus összekapcsolja a szomszédos pontokat, amelyek szintén magpontok, majd ezt a folyamatot iteratív módon ismétli, amíg nem talál több magpontot [37].

A DBSCAN egyik fő előnye, hogy azonosítani tudja a feltűnő (kilógó, *outlier*), vagy zajos pontokat, amelyeket nem tartozik egyik klaszterhez sem [26]. Ez tulajdonsága lehetővé teszi az anomáliák észlelését, valamint jól teljesít olyan feladatokban estén is, ahol az adatok zajjal terhelték [38]. Az optimális ε távolságparaméter és az n magpontok számának meghatározása a DBSCAN esetében kihívást jelenthet, az algoritmus érzékeny a távolságmérési metrika és a paraméterértékek kiválasztására [39].

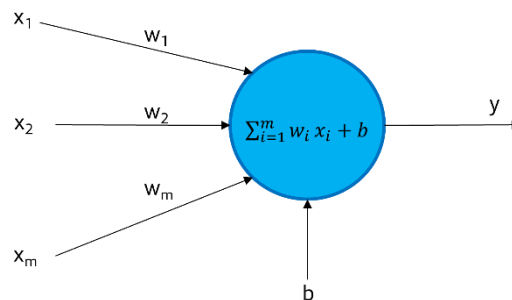
A korlátai ellenére a DBSCAN-t sikeresen alkalmazták számos területen, többek között kép szegmentálásban [40], bioinformatikában [41] és társadalmi hálózatelemzésben [42]. A képessége, hogy kezelje a bonyolult adatstruktúrákat és észlelje az outlier pontokat, népszerű választássá tette a klaszterezési feladatok számos területén.

A DBSCAN algoritmus lépései a következők [34]:

- 1) Véletlenszerűen kiválaszt egy még nem látogatott p pontot.
- 2) Ha p -nek legalább n (minPts) szomszédos pontja van ε sugáron belül, megjelöli magpontként és létrehoz egy új C klasztert p pont körül.
- 3) Ha p -nek a szomszédságában egy nem látogatott q pont van, melynek n (minPts) szomszédja van ε sugáron belül, azt megjelöli magpontként és hozzá adja C klaszterhez.
- 4) Minden p szomszédságában lévő nem látogatott q pont estén, ha q még nem része egyik klaszternek sem, akkor hozzá adja a C klaszterhez.
- 5) Ismétli a 2-4. lépéseket, amíg több pont nem adható hozzá egyetlen csoporthoz sem.

5.11 A perceptron

Az elemi neuron, más néven perceptron, egy lineáris bináris osztályozó, mely a mesterséges neuronhálózatok alapvető építőeleme. A perceptron a biológiai neuron matematikai modellje, melyet Frank Rosenblatt mutatott be 1958-ban [43]. Bináris osztályozási feladatokra lett kifejlesztve, ahol a cél az volt, hogy egy lineáris döntéshatárral két osztályra válassza szét a bemeneti adatokat. A modell (7. ábra) a bemenetek súlyozott összegéből, és az aktiválási függvény segítségével állítja elő a kimenetét.



7. ábra: A perceptron modell [45]

Az aktivációs függvénye általában egy lépcsőfüggvény, amely 1-et ad vissza, ha a súlyozott bemenetek összege meghaladja a küszöbértéket, és 0-t ha nem. A perceptron a tanítás során a bemenetek súlyait iteratívan állítja be, amíg a perceptron kimenete megfelel a kívánt kimenetnek minden tanítópéldánynál. A perceptron működést az alábbi algoritmus írja le:

$$f(x) = \begin{cases} 1 & \text{ha } w \cdot x + b > 0 \\ 0 & \text{egyébként} \end{cases} \quad (4)$$

ahol x a bemeneti vektor, w a valós értékű súlyvektor, $w \cdot x$ a $\sum_{i=1}^m w_i x_i$ skaláris szorzat, m a bemenetek száma és b az előfeszítés vagy zaj (*bias*).

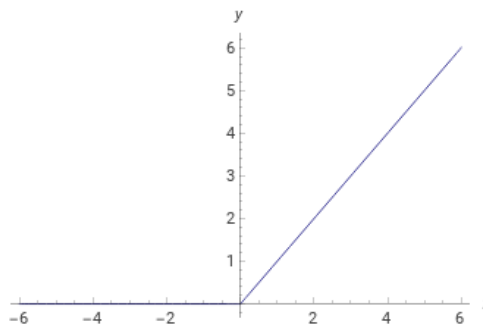
A perceptronnak korlátai vannak a nem lineárisan szeparálható mintázatok tanulásában. Azonban az utat kikövezte a bonyolultabb neurális hálózatok architektúráinak kifejlesztéséhez, például a többrétegű perceptronokhoz [44], amelyek képesek összetettebb mintázatok tanulására.

5.12 Aktivációs függvények

Az aktivációs függvényeket a neurális hálózatokban a nemlinearitás kezelésére használják, és a neurális hálózat rétegeinek kimenetére alkalmazzák. Az aktivációs függvény kiválasztása a konkrét feladattól és az elemezendő adatok jellemzőitől függ. A leggyakrabban alkalmazott aktivációs függvények a ReLU, Leaky ReLU, a Sigmoid, Hiperbolikus tangens és a Softmax.

A ReLU (*Rectified Linear Unit*) egy olyan aktivációs függvény, amely visszaadja a bemeneti értéket, ha az pozitív, 0-t ad vissza egyébként (5. egyenlet). A ReLU könnyen implementálható és segít elkerülni a "eltűnő gradiens", amikor a gradiens nullához közelít, ezáltal lelassul vagy teljesen leáll a tanulási folyamat. Azonban nagy tanítási sebesség esetén előfordulhat a "haldokló ReLU" probléma, amely során a neuronok inaktívvá válhatnak és nem tanulnak tovább [46][47]. Előnye, hogy számítási igénye alacsony a többi aktivációs függvényhez képest, de jól teljesít mély neurális hálózatok esetén is.

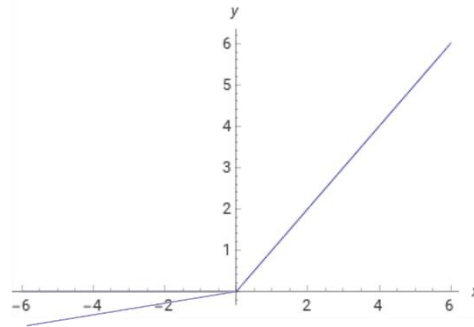
$$R(z) = \begin{cases} z & z > 0 \\ 0 & z \leq 0 \end{cases} \quad (5)$$



8. ábra: A ReLU aktivációs függvény

A Leaky ReLU (6. egyenlet) a ReLU egy változata, amely megoldja a "haldokló ReLU" problémát. A negatív értékeket nem 0-ra állítja, hanem egy kis negatív értékre állítja őket. A Leaky ReLU jobb teljesítményt nyújt a neurális hálózatokban a ReLU-val szemben [47].

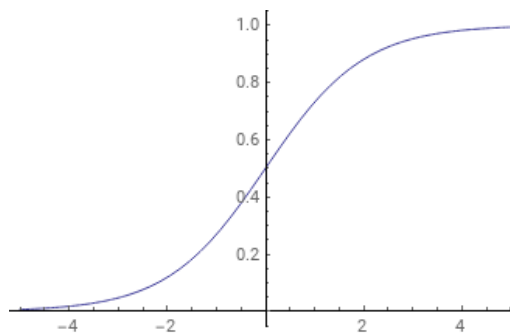
$$R(z) = \begin{cases} z & z > 0 \\ \alpha z & z \leq 0 \end{cases} \quad (6)$$



9. ábra: A Leaky ReLU aktivációs függvény

A szigmoid vagy logisztikus (7. egyenlet) függvény egy olyan aktivációs függvény, amely bármely bemenetet a 0 és 1 közötti értékre képez. Régebben gyakran használták bináris osztályozási problémákhoz, de az újabb neurális hálózatokban más aktivációs függvények váltották fel. Egy probléma a szigmoiddal az, hogy kimenete módosulhat, ha a bemenet túl nagy vagy túl kicsi, ami lassíthatja a tanulást, valamint küzd az „eltűnő gradiens” problémával. [48][49].

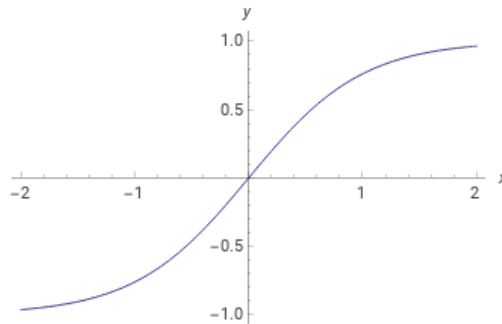
$$S(z) = \frac{1}{1 + e^{-z}} \quad (7)$$



10. ábra: A Sigmoid aktivációs függvény

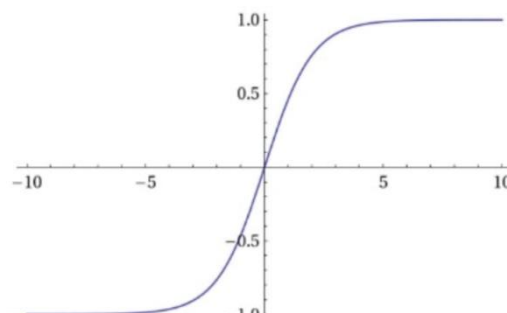
A hiperbolikus tangens (\tanh) függvény (8. egyenlet) hasonló a szigmoidhoz, de a bemenetet -1 és 1 közötti értékre képezi le. A \tanh néhány olyan esetben használható, amikor a neurális hálózat kimenetének a 0 körül kell lennie, például néhány képfelismerési problémánál. A \tanh -nak is lehet "eltűnő gradiens" problémája, és a szigmoidhoz hasonlóan lassabb lehet a tanulás [48][49].

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (8)$$

11. ábra: A \tanh függvény

A softmax (9) egy nemlineáris aktivációs függvény, amely bármely bemeneti értéket 0 és 1 közötti értékre képez le, és biztosítja, hogy az összes kimenet összege 1 legyen. A softmax függvényt általában többosztályos osztályozási problémák esetén használják, mivel az összes lehetséges osztályra kiterjedő valószínűségi eloszlást ad a kimenetén. Mindazonáltal ugyanaz az "eltűnő gradiens" problémától szenvedhet, mint a szigmoid és a tanh, különösen, ha nagy és összetett neurális hálózatokban alkalmazzuk [48].

$$S(z_i) = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}} \quad (9)$$

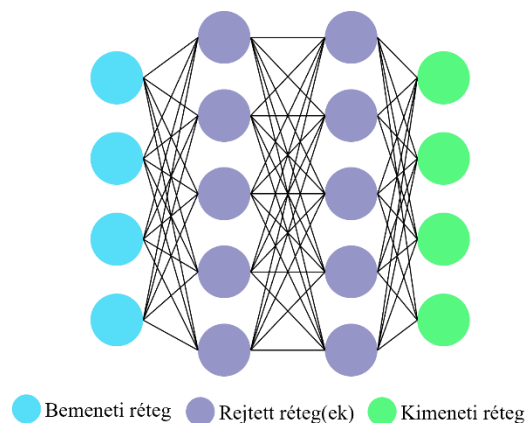


12. ábra: Softmax aktivációs függvény

Az elmúlt években több új aktivációs függvényt vezettek be, amelyek ígéretes eredményeket mutattak, az egyik ilyen függvény a Swish, amely sima és nem monoton függvény, és bizonyos esetekben jobban teljesített, mint a ReLU és más aktivációs függvények [50][51], illetve a GELU (*Gaussian Error Linear Unit*) aktivációs függvény, amely egy simított közelítése a ReLU függvénynek és hatékonyan használható mély tanulási modellekben, különösen természetes nyelvfeldolgozási feladatokhoz [52][53].

5.13 Mesterséges neurális hálózatok (ANN)

A mesterséges neurális hálózatok (*Artificial Neural Network, ANN*) (12. ábra) a gépi tanulás algoritmusainak egy alcsoportja, amelyek biológiai neuronhálózatok szerkezetének és működésének modelljére épülnek [48]. Az ANN több rétegből és összekapcsolt neuronokból áll, amelyek feldolgozzák és továbbítják az információt. Alapvető építőeleme a perceptron, ezért szokás többrétegű perceptronnak (*Multi Layer Perceptron, MLP*) is nevezni. A mesterséges neurális hálózatok egy bemeneti, minimum egy rejtett, és egy kimeneti rétegből állnak. A bemeneti réteg fogadja az adatokat, majd a rejtett réteg(ek) az adatokon matematikai transzformációkat hajtanak végre, mielőtt azokat továbbítják a kimeneti rétegnek. Több rejtett réteggel rendelkező hálózatokat szokás mély neurális hálózatoknak nevezni, innen ered a mélytanulás (*deep learning*) elnevezés is.



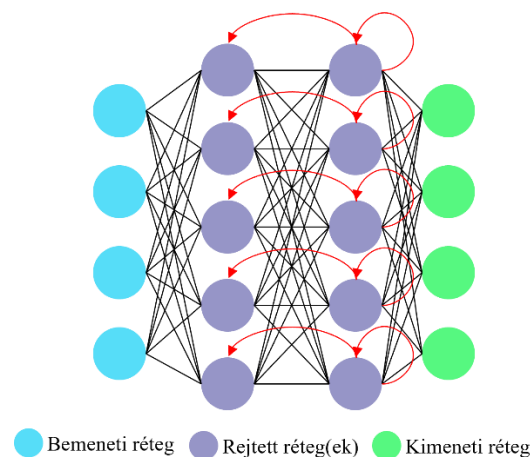
13. ábra: Teljesen összekötött, előrecsatolt mesterséges neurális hálózat

Az ANN-ek különböző típusokba sorolhatók architektúrájuk szerint, például előrecsatolt, visszacsatolt és konvolúciós neurális hálózatokra [48][54]. Az előrecsatolt hálózatok (*Feedforward Neural Network*) a legegyszerűbb típusúak, és egy vagy több rétegből állnak, amelyek az bemeneti adatokat egyetlen irányban dolgozzák fel. A visszacsatolt hálózatok (*Recurrent Neural Network, RNN*) visszacsatolási kapcsolatokkal rendelkeznek, amelyek lehetővé teszik számukra a szekvenciális adatok feldolgozását, például idősorok vagy természetes nyelvi adatok feldolgozását. A konvolúciós neurális hálózatok (*Convolutional Neural Network, CNN*) a kép- és videóadatok feldolgozására specializálódtak, elsősorban a gépi látási alkalmazásokban használják és konvolúciós rétegeket használnak az bementi adatok jellemzőinek kinyerésére.

A sikeres alkalmazások ellenére mesterséges neurális hálózatoknak vannak korlátai és kihívásai. Ezek a hálózatok fekete dobozoként viselkednek, mert nehéz lehet megérteni, hogyan hoznak döntéseket vagy mi alapján jeleznek előre [48][55]. Az értelmezhetőség hiánya problémát jelenthet azokban az alkalmazásokban, ahol az átláthatóság és az elszámoltathatóság fontos, például az egészségügyben és a pénzügyekben. Ezenkívül az ANN-ok sebezhetőek lehetnek az ellenséges támadásokkal szemben, amikor a támadó szándékosan manipulálja a bemeneti adatokat, hogy a hálózat helytelen előrejelzéseket készítsen. Végül az ANN-ek túlillesztést (*overfitting*) szenvedhetnek, amikor a hálózat túlságosan specializálódik a tanítási adatokra, és rosszul teljesít az új adatokon (általánosítási hiba).

5.14 Visszacatolt neurális hálózatok (RNN)

A visszacsatolt vagy rekurrens neurális hálózatok (*Recurrent Neural Network, RNN*) a mesterséges neurális hálózatok olyan típusai, amelyek különösen alkalmasak szekvenciális adatok, például idősorok vagy természetes nyelvi adatok feldolgozására. Ellentétben az előreccatolt neurális hálózatokkal, amelyek csak egy irányba dolgozzák fel a bemeneti adatokat, az RNN-ek képesek fenntartani a korábbi bemenetek „memóriáját”, lehetővé téve számukra, hogy előrejelzéseikbe beépítsék a kontextust és az időbeli függőségeket [56]. A visszacsatolás lehet régek közötti, illetve rétegen belüli is (14. ábra).



14. ábra: Rekurrens neurális hálózat (RNN)

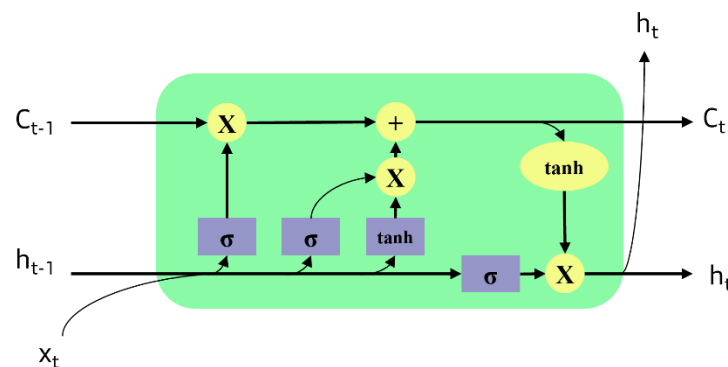
Az RNN magja egy egyszerű számítási egység, az úgynevezett "cella". A cella bemenetként egy bemeneti vektort és egy rejtett állapotvektort vesz fel, frissíti a rejtett állapotot a bemenet és az előző rejtett állapot alapján, és egy kimeneti vektort állít elő [57]. Ily módon a cella kimenete nem csak az aktuális bemenettől függ, hanem a hálózat által feldolgozott összes korábbi bemenettől is.

Az egyik legfontosabb kihívás az RNN-ek betanítása során az eltűnő gradiens problémája, amely akkor fordulhat elő, ha a hálózaton keresztül terjedő gradiens rendkívül kicsivé válnak, és a súlyok nagyon lassan frissítik [58]. A probléma megoldására az alapvető RNN-architektúra számos változatát fejlesztették ki, például a hosszú rövid távú memória (LSTM) hálózatokat [59] és a kapuzott ismétlődő egységeket (GRU) [60], amelyek kapuzási mechanizmusokat tartalmaznak a rejtett elemek szelektív frissítésére, hogy ezáltal enyhítse az eltűnő gradiens problémát.

Az RNN-eket számos alkalmazásban sikeresen alkalmazták, beleértve a beszéd felismerést [61], a nyelvi modellezést [62], a gépi fordítást [63] és a hangulatanalízist [64]. Azonban ismertek bizonyos korlátaik is, például nehézségekbe ütközik a hosszú távú függőségek rögzítése, és hajlamosak elfelejteni a korábbi bemeneteket, ahogy a sorozat hosszabbodik [65].

5.15 Long-Short Term Memory (LSTM) hálózatok

Az LSTM (*Long Short-Term Memory*) hálózatokat Hochreiter és Schmidhuber vezették be 1997-ben [57]. Az LSTM-ek a hagyományos visszatérő neurális hálózatokban jelentkező elhalványodó gradiens problémára adnak megoldást azáltal, hogy lehetővé teszik a modellnek, hogy szelektíven emlékezzen vagy elfelejtse a korábbi bemeneteket egy memóriacella (15. ábra), bemeneti kapu, elfelejtési kapu és kimeneti kapu segítségével [66]. A memóriacella hosszú távú memóriaraként működik, amely fontos információkat tárol, míg a kapuk szabályozzák az információ áramlását a cellába és a cellából.



15. ábra: LSTM cella

A LSTM cella működését leíró képletek a következők:

A bemeneti kapu meghatározza, hogy az adott bemenetből és a korábbi rejtett állapotból milyen információkat kell tárolni a cellában.

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i) \quad (10)$$

ahol σ a szigmoid aktivációs függvény, W_i és b_i a bemeneti kapu súlyai és torzítása (*bias, zaj*), x_t és h_{t-1} pedig a jelenlegi bemenet és a korábbi rejtett állapot.

A törölő kapu határozza meg, hogy milyen információkat kell eldobni a cellában.

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f) \quad (11)$$

ahol W_f és b_f a elfelejtési kapu súlyai és torzítása.

A cella állapot jelenti a cella memóriáját. Az új állapot az előző állapot és az új bemenet kapu szerinti súlyozott kombinációjából áll, az elfelejtési és bemeneti kapuk szerinti súlyokkal.

$$c_t = f_t c_{t-1} + i_t \tanh(W_c[x_t, h_{t-1}] + b_c) \quad (12)$$

ahol W_c és b_c a cellaállapot súlyai és torzítása, \tanh pedig a hiperbolikus tangens aktivációs függvény.

A kimeneti kapu meghatározza, hogy milyen információkat kell kimenetként kiadni a cellás állapotból.

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o) \quad (13)$$

ahol W_o és b_o a kimeneti kapu súlyai és torzítása.

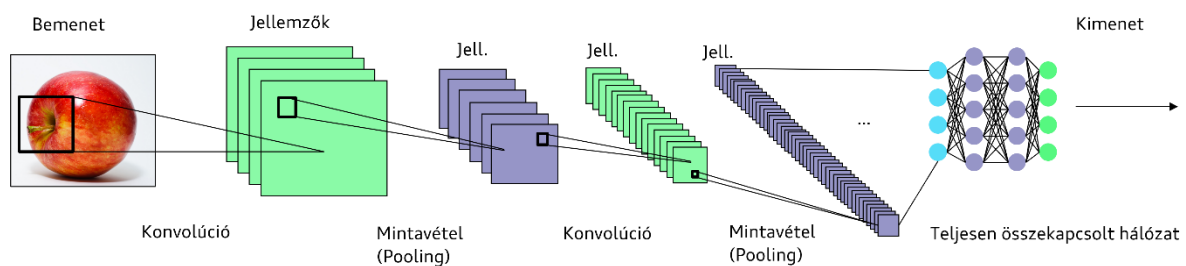
A rejtett állapot a LSTM cella kimenete. Az új rejtett állapot az aktuális cellás állapotból és a kimeneti kapu szerinti súlyozással kapott értékből áll.

$$h_t = o_t \tanh(c_t) \quad (14)$$

Ezek a képletek lehetővé teszik az LSTM cellának, hogy szelektíven tároljon vagy eldobjon információkat, frissítse a memóriáját, és minden időpillanatban új kimenetet adjon ki.

5.16 Konvolúciós neurális hálózatok (CNN)

A konvolúciós neurális hálózatok (CNN-ek) a mély neurális hálózatok egy fajtája, amelyet széles körben használnak kép- és videófeldolgozási feladatokban [54]. Célja, hogy automatikusan kivonja a releváns jellemzőket a bemeneti képekből, és ezek alapján osztályozást, szegmentálást vagy egyéb feladatokat hajtson végre [56]. A CNN-ek több rétegből állnak, ezek a konvolúciós rétegek, a mintavételező (*pooling*) rétegek és a teljesen összekapcsolt rétegek [54]. A konvolúciós rétegek szűrőket alkalmaznak a bemeneti képre, amelyek segítségével kinyerhetők a kép fontos jellemzői [67]. A pooling rétegek mintavételeznek a jellemzőtérképekből, hogy csökkentsék az adatok térbeli felbontását [68]. Végül a teljesen összekapcsolt rétegek a kivont jellemzők alapján végzik el az osztályozást vagy más feladatot.

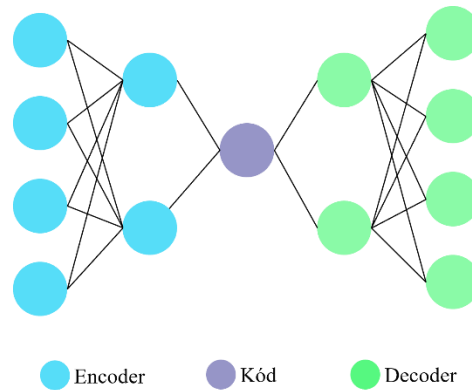


16. ábra: Konvolúciós neurális hálózat sematikus ábrája

A CNN-ek rendkívül hatékonyak a tárgy- és arcfelismerésben, valamint a természetes nyelvi feldolgozásban is [69]. A CNN-ek egyik legfontosabb előnye, hogy képesek megtanulni a bemeneti adatok hierarchikus ábrázolását, ami lehetővé teszi számukra, hogy összetett mintákat is azonosítani tudnak [70]. Nagy adathalmazok kezelésére is képesek, és párhuzamos számítási technikákkal hatékonyan betaníthatók [71]. Az elmúlt években számos előrelépés történt a CNN architektúrákban, például a maradék hálózatok (*Residual Networks*) és figyelemmechanizmusok (*Attention Mechanism*) bevezetése, amelyek tovább javították a teljesítményüket [72]. A CNN-eket más mély tanulási modellekkel, például visszatérő neurális hálózatokkal kombinálva is használják összetettebb feladatok, például videó feliratok elvégzésére [73].

5.17 Autoencoder hálózatok

Az autoencoder neurális hálózatok olyan típusú felügyelet nélküli tanuló neurális hálózatok, amelyek használhatók jellemző (*feature*) tanulásra, adattömörítésre és adatok (újra)generálására. Az autoencoder-nek (17. ábra) van egy encoder része, amely leképezi a bemeneti adatokat egy alacsonyabb dimenziós reprezentációra, és van egy dekóder része, amely az alacsonyabb dimenziós reprezentációt visszaalakítja az eredeti bemeneti adatokra [74]. Az encoder hálózat egy sor rétegből áll, amelyek fokozatosan csökkentik a bemeneti adatok dimenzióját. A dekóder hálózat általában szimmetrikus az encoderrel és olyan rétegekből áll, amelyek növelik az alacsonyabb dimenziós reprezentáció dimenzióját [75].



17. ábra: Autoencoder hálózat

$$\text{Encoder: } \varphi : X \rightarrow F \quad (15)$$

$$\text{Decoder: } \psi : F \rightarrow X \quad (16)$$

$$\varphi, \psi = \arg_{\varphi, \psi} \min \|X - (\psi \circ \varphi)X\|^2 \quad (17)$$

Az autoencoder a tanítás során minimalizálja a rekonstrukciós hibát a bemeneti adatok és a dekóder hálózat kimenete között (15-17). A tanításhoz használt veszteségfüggvény általában a átlagos négyzetes hiba (*Mean Square Error, MSE*) a bemeneti adatok és a rekonstruált adatok között. A tanulási folyamat során visszaterjesztik a hibát a hálózaton, és beállítják a súlyokat a rekonstrukciós hiba minimalizálása érdekében [76].

Az autoencoder-eket használják zajos képek és adatok szűrésére, az anomálfelismerésre és a dimenzionalitás-csökkentésre. Az autoencoder használható a generatív modellezéshez is,

amikor mintákat vesznek a látens térben, majd a mintákat átadják a dekóder hálózaton keresztül az új adatok generálásához [77].

Az egyik népszerű autoencoder típus a variációs autoencoder (VAE) [78]. A VAE-k valószínűségi eloszlást tanulnak az alacsonyabb dimenziós reprezentációról, és új adatokat generálnak, mintavételezéssel az eloszlásból. Egy másik autoencoder típus a zajszűrő autoencoder, amely tiszta adatokat állít elő zajos bemeneti adatokból [79].

5.18 Felügyelet nélküli tanulás

Az felügyelet nélküli (*unsupervised learning*) tanulás egy olyan gépi tanulási típus, amelyben az algoritmusokat fel nem címkézett adatokkal tanítjuk annak érdekében, hogy azok felismerjék az adatok alapjául szolgáló mintázatokat és struktúrákat. A felügyelttől eltérően, amely címkézett adatokat igényel a predikciókhoz, az felügyelet nélküli tanulás lehetővé teszi az algoritmusok számára, hogy előzetes tudás nélkül azonosítsák a mintázatokat az adatokban. Az unsupervised learning esetén gyakran használt technikák közé tartoznak a klaszterezés, a dimenziócsökkentés és az anomáliadetektálás [80].

Az unsupervised learning egyik példája a clustering, amely az adatpontok csoportosítását jelenti a hasonlóság alapján. Ez hasznos lehet a természetes csoportosítások felfedezéséhez az adatokban, például a különböző ügyfélszegmensek azonosításához a marketing adathalmazban [81]. Másik gyakori technikája a dimenziócsökkentés, amely a változók számának csökkentését jelenti az adathalmazban, miközben megőrzi a legfontosabb információkat. Ez hasznos lehet nagy dimenziós adatok vizualizálásához vagy zaj csökkentéséhez az adathalmazban [82]. Az anomáliadetektálás egy másik példája az unsupervised learning-nek, amely az adathalmazban szokatlan mintákat vagy kiugró értékeket azonosít. Ez hasznos lehet pénzügyi tranzakciók csalásának felismerésében vagy hálózati forgalom szokatlan viselkedésének azonosításában [83]. A felügyelet nélküli tanulás lehetővé teszi az algoritmusok számára, hogy előzetes tudás vagy felügyelet nélkül tanuljanak, így olyan információkat és trendeket azonosíthatnak, amelyek más módszerekkel nem lennének láthatók [84].

5.19 Felügyelt tanulás

A felügyelt tanulás egy olyan gépi tanulás algoritmus típus, amely címkézett adatokat használ a modellek tanításához, hogy azok megjósolják a bemeneti változók alapján a jövőbeli kimeneteket [82]. Az címkézett adatok olyan bemeneti változókat és az ezekhez tartozó kimeneti változókat tartalmaznak, amelyek alapján a modell le tudja képezni a köztük lévő kapcsolatot. A felügyelt tanulás célja, hogy pontosan előre jelezze az új, a tanulás során nem látott bemeneti változókhoz tartozó kimeneti változót [80].

A felügyelt tanulás két kategóriára osztható: regresszióra és osztályozásra [25]. A regresszióban a modell folytonos kimeneti változót jósol, míg az osztályozásban a modell diszkrét kimeneti változót jósol[85]. A felügyelt tanulás általános alkalmazásai közé tartozik például a képfelismerés, beszéd felismerés és az érzelmi tartalmú elemzés [48].

A felügyelt tanulás folyamata több lépésből áll, beleértve az adat előkészítését, modell kiválasztást, tanítást és a modell értékelését [86]. Az adat előkészítése során a címkézett adatokat meg kell tisztítani és előfeldolgozását kell végezni annak érdekében, hogy az adatok megfelelőek legyen a modell tanításához. A modell kiválasztásának egyik szempontja az algoritmus és a paraméterek megfelelő kiválasztása a teljesítmény optimalizálása érdekében. A modell tanítása magában foglalja a címkézett adatok modellbe történő betáplálását annak érdekében, hogy beállítsa annak paramétereit és javítsa az pontosságát. Végül a modell értékeléséhez a modell tesztelni kell új, a tanítás során nem használt adatokkal annak érdekében, hogy felmérjük annak teljesítményét.

5.20 Hibavisszaterjesztés (backpropagation)

A hibavisszaterjesztés (backpropagation) neurális hálózatok tanításához használt algoritmus, amely az összeköttetések súlyainak frissítésével tanítja a modellt. A működése az előrejelzett kimenet és az igazi kimenet közötti hibát visszaterjeszti a hálózaton keresztül, kiszámolja a hiba gradiensét a súlyokhoz képest. A gradienst ezután felhasználja a súlyok frissítésére úgy, hogy minimalizálja a hibát [76]. A tanítást iteratív módon ismétli, amíg a modell egy elfogadható pontossági szintre nem konvergál. A hibavisszaterjesztés kulcsfontosságú szerepet játszik a mélytanulási modellek sikerében [54].

5.21 Modellek teljesítményének javítása

A gépi tanulási algoritmusok gyakran több technikát igényelnek a teljesítményük javításához és az túltanítás (*overfitting*) elkerüléséhez. Erre a célra a regularizáció, a korai leállítás (*early stopping*), a kiejtés (*dropout*) és a hiperparaméter beállítás a gépi tanulásban gyakran használt technikák.

A regularizáció [48] a veszteségfüggvényhez hozzáad egy büntető tagot. Ez a büntető tag büntetést szab a nagy együtthatókért és arra ösztönzi a modellt, hogy kisebb együtthatókkal dolgozzon. Ilyenek az L1 és L2 regularizáció. Az L1 regularizáció az együtthatók abszolút értékének arányában bünteti a modellt, míg az L2 regularizáció a négyzetes együtthatók arányában büntet. A regularizáció segíthet a modell bonyolultságának csökkentésében és az *overfitting* elkerülésében.

Az *early stopping* [87] technika segítségével az túltanítás elkerülésére a modell tanítása közben leállítjuk a tanítási folyamat úgy, hogy a tanítás során figyelemmel kísérjük a validációs veszteséget és leállítjuk a tanítást, amikor az már tovább nem javul. Ezzel megakadályozható, hogy a modell a zajt is megtanulja és jobb általánosításra lesz képes.

A kiejtés (*dropout*) [88] egy olyan technika, amely segíti az túlillesztés elkerülését azzal, hogy a hálózat egyes egységeit véletlenszerűen eltávolítja, kiejti a tanítás során. Ezáltal a hálózat kénytelen redundáns reprezentációkat tanulni és ellenállóbbá válik a zajos adatokkal szemben. A *dropout* egy olyan regularizációs technika, amelyet más regularizációs módszerekkel, például az L1 és L2 regularizációval együtt lehet alkalmazni.

A hiperparaméter beállítás egy olyan technika, amely segítségével az optimális értékeket állítják be a gépi tanulási algoritmusok hiperparamétereire. A hiperparaméterek olyan

paraméterek, amelyeket kézzel kell beállítani a tanítás során. A hiperparaméterek például a tanulási ráta, a batch méret és a regularizáció mértéke. A megfelelő hiperparaméterek beállítása segíthet javítani a modell teljesítményét és elkerülni az túlillesztést. A hiperparaméterek beállításához használható technikák közé tartozik a rácskeresés, a véletlen keresés és a bayesi optimalizáció [89].

5.22 Univerzális approximátor tétel

Az univerzális approximátor tétel kimondja, hogy egyetlen rejtett rétegű, véges számú neuronból álló neurális háló bármilyen folytonos függvényt képes kívánt pontossággal közelíteni egy zárt halmazon [90]. Ez azt jelenti, hogy a háló képes ábrázolni bármilyen bemenet-kimenet leképezést, ha elegendő számú rejtett egységet használunk [91]. A tétel érvényes többrétegű hálókra is, amennyiben rendelkeznek elegendő számú rejtett neuronokkal [92]. Az univerzális közelítő tétel bizonyítása a sűrű részhalmazok fogalmán alapul és matematikai elemzésre épül [93].

Az univerzális közelítő tétel fontos következményekkel jár a neurális hálók gyakorlati alkalmazására nézve, mivel bizonyítja, hogy a háló képes komplex függvények közelítésére egy viszonylag egyszerű architektúrával [94]. Azonban, ez nem garantálja, hogy a háló optimális megoldásra fog konvergálni, vagy hogy az approximációs hiba gyakorlati szempontból elég kicsi lesz [95].

5.23 Természetes nyelvi feldolgozás (Natural Language Processing)

Természetes nyelvi feldolgozás (*Natural Language Processing, NLP*) a számítástudomány és az mesterséges intelligencia egy olyan területe, amely az emberek és a számítógépek közötti nyelvi kommunikációra összpontosít. Az NLP célja, hogy lehetővé tegye a gépek számára az emberi nyelv megértését, értelmezését és generálását. Számos feladatot foglal magában, például szintaktikai és szemantikai elemzést, információ kinyerést, gépi fordítást, hangfelismerést és érzelmi elemzést. Az NLP technikák statisztikai és gépi tanulási modelleket használnak nagy mennyiségű szöveges adatok feldolgozásához [96]. Az NLP egyik kulcskérdése az emberi nyelv többértelműségének és bonyolultságának kezelése, ideértve a nyelvtani, szintaktikai és szókincsbeli változatosságot is. Az NLP-nek számos gyakorlati alkalmazása van, például szöveg osztályozás, chatbotok, virtuális asszisztensek és nyelvi fordítás [97].

Az NLP az utóbbi években jelentősen fejlődött a nagy adathalmazok, erős számítási erőforrások és mély tanulási technikák elérhetősége miatt [98]. Azonban még mindig sok nyitott kutatási kérdés van a területen, ideértve az NLP modellek pontosságának és robusztusságának javítását, az alacsony erőforrású (az NLP alkalmazások tanításához kevés írásos nyelvi anyaggal rendelkező) nyelvek és dialektusok kezelését és az NLP alkalmazások etikus felhasználását és társadalmi torzítás kizárásának biztosítását. [99].

Az NLP lépései közé tartozik a szöveg előfeldolgozása, a szintaktikai elemzése, a szemantikai elemzés és a szöveggenerálás. Az első szakaszban a szöveget megtisztítják és tokenizálják kisebb egységekre, úgynevezett n-gram-okra. Az n-gramok olyan folytonos n elemet tartalmazó sorozatok (vagy tokenek), amelyek nagyobb szövegsorozatból vannak kiválasztva. Az n-gramok elemei karakterek, szavak vagy akár kifejezések is lehetnek [97]. Például az "A gyors barna róka" mondat 2-gramja (vagy bigramja) az "A gyors", "gyors barna" és "barna róka" [96]. Az n-gramok egyik fő alkalmazása a nyelvi modellezés, ahol az előző n-1 szót használják egy szó valószínűségének megjóslására [101]. Az n-gramok viszont szenvednek az dimenzionalitás átkától, ami azt jelenti, hogy az elképzelhető n-gramok száma exponenciálisan növekszik a tokenizált szöveg méretével. Ezért ritka vagy nem ismert n-gramokkal való találkozás esetén ritkaság és túlillesztési (*overfitting*) problémákhoz vezethet. Ennek enyhítésére simítási technikákat, például Laplace-simítást vagy Good-Turing-simítást lehet használni [101].

A második lépés a szöveg szintaktikai szerkezetének elemzése, amelynek célja egy mondat struktúrájának elemzése és a mondat alkotóelemei közötti grammatikai kapcsolatok

meghatározása [96]. A folyamat magában foglalja a mondat szétbontását összetevő részekre, mint például a főnevek, igék, melléknevek és határozók, és azonosítja, hogy ezek a nyelvtani elemek hogyan viszonyulnak egymáshoz a szerepük és funkciójuk tekintetében a mondaton belül [97]. A szintaxis elemzésnek két fő megközelítése az NLP-ben a szabályalapú és a statisztikai megközelítés [102]. A szabályalapú megközelítés előre meghatározott grammatikai szabályokat és mintákat használ a mondatok elemzéséhez, míg a statisztikai megközelítés gépi tanulási algoritmusokat használ nagy korpuszokból (*szövegtörzsből*), amelyeknek az annotált, felcímkézett mondatokból tanulja meg az automatikus szintaktikai szerkezetek azonosítását [104]. A szintaxis elemzés kimenete általában egy szintaktikai fa vagy gráf, amely ábrázolja a mondat hierarchikus szerkezetét és az alkatrészek közötti kapcsolatokat [105]. Ezek az információk számos NLP-feladathoz használhatók, például gépi fordításhoz, szövegösszefoglaláshoz és érzelemfelismeréshez [106].

A harmadik szakaszban a szöveg jelentése elemzésre kerül, hogy azonosítsuk a szemantikai kapcsolatokat, ideértve a szóértelmezés tisztázását és a nevezetes entitások felismerését. Ez a folyamat magában foglalja a szavak és kifejezések jelentésének értelmezését [97]. Ez egy összetett folyamat, amely különböző alfeladatokat foglal magában, például a részfeladatok címkézését, a nevesített entitások azonosítását, a szemantikai címkézését és az érzelmi analízist [96]. A címkézés az egyes szavak nyelvtani kategóriájának azonosítását jelenti egy mondaton belül [102]. A nevesített entitások azonosítása magában foglalja a nevek, szervezetek és helyszínek stb. azonosítását és kategorizálását [107]. A szemantikai szerepcímkézés a különböző szavak közötti szemantikai kapcsolatok azonosítását jelenti, például a tárgy, alany és állítmány meghatározását [108]. Az érzelmi analízis pedig az adott mondat hangulatának azonosítását jelenti, hogy pozitív, negatív vagy semleges-e [109]. A szemantikai elemzés fő célja, hogy kinyerje a szövegből a jelentést, és olyan formában jelenítse meg, hogy azt a számítógép megérthesse [110]. Ezt a szemantikai reprezentáció létrehozásával érik el, például egy szemantikai gráf vagy tudástár létrehozásával, amely tartalmazza a szövegben szereplő különböző fogalmak közötti kapcsolatokat [111]. Ez a reprezentáció különféle NLP-feladatokhoz használható, például az információkinyeréshez, kérdések megválaszolásához és szövegösszefoglaláshoz [112].

Végül a szöveggeneráló szakaszban a gép a bemeneti adatok és az elemzés során tanult szabályok alapján generál természetes nyelvű szöveget [102]. A folyamat általában a szöveggenerálás feladatának vagy céljának meghatározásával kezdődik, mint például összefoglalás vagy párbeszédképzés. Ezután a modellt egy nagyobb szövegtörzsen tanítják,

deep learning vagy egyéb neurális hálózatok használatával [113]. A modell a bemenet alapján szöveget generál, amelyben lehetnek kulcsszavak, ígéretek vagy más utalások [114]. A generált szöveget koherencia, nyelvtani helyesség és az adott feladatra való relevancia alapján értékelik [115]. Az értékelési folyamat során emberi értékelőket vagy automatizált metrikákat, mint a BLEU pontszámok használhatunk [116]. Végül a modellt finomhangolják és optimalizálják a teljesítmény javítása érdekében [117]. A folyamat több iterációt és állítást is magába foglalhat, amíg el nem érik a kívánt szövegminőséget [118]. Az NLP technikákat számos alkalmazásban használják, ideértve a chatbotokat, a gépi fordítást és az érzelmek elemzését.

5.24 Python

Python egy magas szintű, általános célú, objektumorientált, interpretált programozási nyelv, amelyet egyszerűségével, olvashatóságával és sokoldalúságával teszi népszerűvé. Felhasználási területei többek között a webfejlesztés, adatelemzés és mesterséges intelligencia [119]. A tervezési filozófiája a kód olvashatóságát helyezi előtérbe, és kiemeli a whitespace (*szóköz*) karakter használatát a kódblokkok elhatárolásához, más programozási nyelvekkel ellentétben, kapcsos zárójeleket vagy kulcsszavakat használnak. A Python szintaxisa egyszerű és az angol nyelvre emlékeztető, ezért kezdők számára is könnyen tanulható [120]. A Python könnyen bővíthető, nagy modul- és csomagkönyvtárral, amelyeket könnyen integrálhatunk a programokba, hogy új funkcionalitást adjunk hozzájuk, például a NumPy-t a tudományos számításokhoz vagy a Django-t a webfejlesztéshez [121]. Emellett a Pythonnak egy nagy és aktív közössége van, amely hozzájárul a folyamatosan fejlődő ökoszisztémájához, amely tartalmaz keretrendszereket, könyvtárakat és eszközöket.

A Python egyik legkiemelkedőbb jellemzője a dinamikus típusrendszer, amelyben a változó típusa a futásidőben és nem a fordításidőben kerül megállapításra, ezzel biztosítva a rugalmasságot és a könnyű használatot. A Python objektumorientált, lehetővé téve osztályok és objektumok létrehozását, amelyek segítségével adatokat és funkciókat lehet összefogni [122].

5.25 TensorFlow

A TensorFlow egy nyílt forrású gépi tanulási platform, amelyet a Google fejlesztett ki [123]. Segítségével felhasználók képesek gépi tanulási modellek létrehozására és telepítésére különböző alkalmazásokban, például természetes nyelvfeldolgozásban, gépi látásban és prediktív analitikában [124]. A TensorFlow adatáram-gráfokat használ a gépi tanulási modell által végrehajtott számítások reprezentálására, ami lehetővé teszi a modell viselkedésének könnyű vizualizálását és megértését [125]. A platform magas szintű API-t (*Application Programming Interface*) tartalmaz az neurális hálózatok építéséhez (*Keras*), valamint alacsony szintű műveleteket is támogat, amelyek lehetővé teszik a fejlesztők számára a modellek testreszabását [126]. A TensorFlow továbbá eszközöket biztosít a modellek tanításához és validálásához, beleértve az automatikus differenciálást, amely megkönnyíti a gradiens számítását a modellek paramétereinek optimalizálásához [127]. A TensorFlow emellett támogatja a megosztott számítást, amely lehetővé teszi a felhasználók számára a modellek

skálázását nagy adatkészletek és összetett számítások kezeléséhez [128]. Összességében a TensorFlow egy erőteljes eszköz a gépi tanulás kutatói és gyakorlóinak számára, széles körű lehetőségekkel és alkalmazásokkal.

5.26 Optimalizációs eljárások

Az optimalizációs algoritmusok kulcsfontosságú szerepet játszanak a gépi tanulásban a modellek tanítása során a hiba vagy költségfüggvény minimalizálása révén. A gépi tanulás kontextusában az oktatási ráta vagy tanítási sebesség egy hiperparaméter [48], amely a modell tanítása során annak paramétereinek optimalizálása lépésközét szabályozza. Ez határozza meg, hogy mennyire állítják be a modell paramétereit a becsült hiba vagy a veszteségfüggvény gradienseinek változása alapján. A túl nagy oktatási ráta esetén a modell átlépheti a legjobb megoldást, míg egy alacsony oktatási ráta lassabb konvergenciát vagy akár a konvergencia hiányát is okozhatja. Az oktatási rátát általában manuálisan állítják be a szakember által, vagy automatikusan hangolják a rácskeresés, véletlenszerű keresés vagy Bayesi optimalizáció használatával. A gradiens csökkentés (GD) egy széles körben használt optimalizációs algoritmus, amely iteratívan frissíti a modellparamétereket a költségfüggvény negatív gradienseinek irányában. A sztochasztikus gradiens csökkentés (SGD) egy változata a GD-nek, amely véletlenszerűen választ ki egy részhalmazt az edzési adatokból (mini-adathalmaz) a gradiens kiszámításához, ami csökkenti a számítási terhet és javítja a konvergencia sebességét [129]. Az Adagrad egy másik optimalizációs algoritmus, amely a paraméterváltoztatások gyakorisága alapján módosítja a tanítási sebességet (*learning rate*) úgy, hogy a tanítási sebességet a gyakran frissített paraméterek esetében csökkenti és a ritkán frissített paraméterek esetében növeli [130]. Az Adam ötvözi az SGD és az Adagrad előnyeit, mivel mind az első, mind a második gradiens pillanatnyi értékeire alapozva állítja be a tanítási sebességet [131]. Az RMSprop az Adagrad egy változata, amely az elmúlt gradiens exponenciálisan csökkenő átlagát használja az tanítási sebesség adaptálásához [132]. Az AdaDelta egy másik adaptív tanulási rátájú optimalizációs algoritmus, amely az elmúlt gradiens mozgó átlagát és frissítéseit használja a learning rate adaptálásához [133]. Végül a Nadam a Nesterov-gyorsított gradiens (NAG) és az Adam kombinációja, amely NAG-t használ a gradiens frissítéséhez és Adam-et az tanítási sebesség frissítéséhez [134].

5.27 Scikit-learn

A scikit-learn egy széles körben használt, nyílt forráskódú gépi tanulási könyvtár Pythonhoz [135]. Egy egységes felületen keresztül kínál hatékony eszközöket osztályozáshoz, regresszióhoz, csoportosításhoz és dimenziócsökkentéshez [136]. A scikit-learn számos algoritmust és módszert tartalmaz az adatok előfeldolgozásához, jellemzők kinyeréséhez, modell kiválasztásához és értékeléséhez, mint például a k-legközelebbi szomszéd, döntési fák, tartóvektor-gépek (*Support Vector Machine, SVM*), véletlen erdők, gradiens fokozás és neurális hálózatok [137]. Emellett a scikit-learn eszközöket kínál keresztvalidációhoz, rácskereséshez, jellemzők kiválasztásához és gépi tanulási pipeline-ok építéséhez a munkafolyamatok automatizálására [138]. A könyvtár jól dokumentált, tiszta példákkal és útmutatókkal, így hozzáférhető a különböző szintű felhasználók számára [139].

5.28 A gépi tanulás korlátai

A gépi tanulásnak számos korlátja van. Többek között erősen támaszkodik az adatok minőségére és mennyiségére [140]. Probléma még a transzparencia, a gépi tanulás modellek átláthatatlansága miatt nehéz megérteni, hogyan jutnak el döntéseikhez [141]. A torzítás szintén okozhat problémát és elősegítheti a társadalmi diszkriminációt [142]. Az általánosítás szintén problémás lehet, a modellek nem képesek absztrakt gondolkodásra vagy általánosításra a tanulási adatokon kívül [143]. A gépi tanulás jelentős számítási erőforrást igényel, ami sok alkalmazás számára elérhetetlen [144]. Ezek a korlátok rámutatnak arra, hogy a kutatóknak és a felhasználóknak ügyelniük kell a gépi tanulás megfelelő alkalmazására.

5.29 Supervisory Control and Data Acquisition (SCADA)

A Supervisory Control and Data Acquisition (SCADA) rendszerek összetett rendszerek, amelyek ipari folyamatok és infrastruktúrák ellenőrzésére és vezérlésére használnak. A SCADA rendszerek hardver- és szoftverkomponensekből állnak, amelyek adatokat gyűjtenek szenzorokból és más eszközökből, majd valós időben elemzik és megjelenítik az adatokat az emberi operátoroknak. A rendszert programozhatják arra is, hogy az adatok alapján automatikusan állítsa be a folyamatokat [145]. A SCADA rendszerek hierarchikus felépítésűek és négy szintre oszthatók [146]:

0. Terepi szint: a fizikai folyamatot jelenti, beleértve az érzékelőket, az aktuátorokat és egyéb eszközöket, amelyek adatokat gyűjtenek és ellenőrzik azokat.
1. Vezérlő szint: a programozható logikai vezérlőket (PLC) és egyéb eszközöket tartalmazza, amelyek a fizikai folyamatot figyelik és vezérlik.
2. Felügyeleti szint: maga a SCADA rendszer, amely adatokat gyűjt és feldolgoz az Szint 1 eszközeiből, valamint grafikus felhasználói felületet biztosít az operátoroknak a folyamat figyeléséhez és vezérléséhez.
3. Termelésirányítási szint: a vállalati rendszerekre vonatkozik, amelyek magasabb szintű irányítási és döntéshozási képességeket biztosítanak.

A alsó szintek, mint a 0. és az 1. szint 1 az adatgyűjtésért és az adatok vezérléséért felelnek a fizikai folyamatból, míg a 2. és 3. szint a magasabb szintű irányítási és döntéshozási képességeket biztosítanak [147]. A SCADA rendszer különböző szintekre osztása a biztonsági szempontból is fontos, mivel lehetővé teszi a biztonsági intézkedések megvalósítását minden szinten annak érdekében, hogy megakadályozzák az illetéktelen hozzáférést a rendszerhez [148].

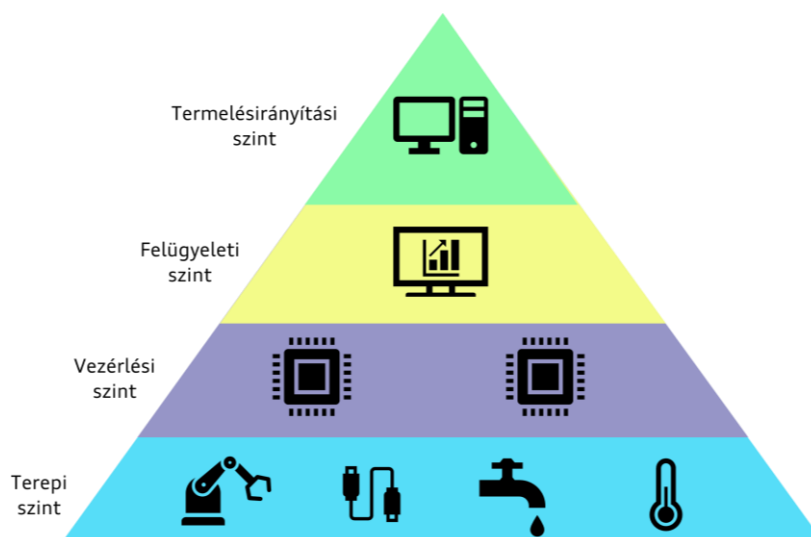
6. Környezet bemutatása

6.1 Az Ipar 4.0 az AUDI HUNGARIA Zrt-nél

Kutatásom a győri AUDI HUNGARIA Zrt-nél, a járműhajtás gyártás területén végeztem. A győri Audi 1993 óta foglalkozik motorgyártással, a napi több mint 9000 legyártott erőforrással a világ legnagyobb motorgyára. A motor összeszerelő sorok mellett a mechanikus megmunkáló sorokkal a teljes gyártási mélységet lefedi, mint például a főtengely, hengerfej, forgattyúház vagy hajtókar mechanikus megmunkálás. Az Otto és a Diesel erőforrások mellett elektromos járműhajtásokat is gyártanak. A széles termékpaletta, illetve a különböző életkorú és eltérő technológiájú gyártósorok, berendezések rendkívül heterogén környezetet eredményeznek. A fenntartható és hatékony gyártás érdekében elengedhetetlen az Ipar 4.0 eszköztárának alkalmazása a termelésben.

6.2 SCADA rendszerek

A különböző törvényi előírások, illetve minőségbiztosítási és termelésirányítási szempontok miatt széleskörű adatgyűjtésre van szükség. Az adatgyűjtését SCADA rendszerek (3.27 fejezet) segítségével végzik. Kutatásom ezen adatok gyűjtésére és azok számítási intelligenciával való feldolgozására koncentrálok.



18. ábra: SCADA rendszer hierarchiája [146].

A terepi szinten található érzékelők, aktuátorok, szelepek és egyéb vezérléstechnikai- és mérőeszközök végzik az adatok gyűjtését, illetve szükség esetén beavatkoznak a folyamatokba. Ezek az eszközök közvetlenül vagy decentralizált periféria modulokon keresztül vannak összekötve a vezérlési szinten található programozható logikai vezérlőkkel (*Programmable Logic Controller, PLC*), numerikus vezérlőegységekkel (*Numerical Control Unit, NCU*) és egyéb CAM (*Computer-aided Manufacturing*), mérő vagy vezérlőegységekkel. A decentralizált perifériák különböző ipari szabványokkal kommunikálnak a vezérlőegységekkel, mint AS-i BUS, CANBUS, Profibus, Profinet vagy egyéb ipari ethernet alapú megoldások.

A vezérlési szinten található intelligens eszközök végzik a mért jelek és adatok feldolgozását és kiértékelését, illetve szükség esetén az aktoroknak kiadott jelekkel beavatkoznak a folyamatokba. A legelterjedtebb eszközök a Siemens S7-300-as és S7-1500-as családba tartozó PLC-k, valamint a 840D Power Line és 840D Solution Line NCU-k, de megtalálhatóak egyéb gyártók termékei is az AUDI HUNGARIA-nál, mint a FANUC, Allen Bradley, Beckhoff vagy Phoenix Contact termékei. Ehhez a szinthez tartoznak a különböző ipari számítógépek, valamint az in-process (folyamat közbeni) és post-process (folyamatvégi) mérőeszközök vezérlőegységei is, amik a gyártott termékek minőségét felügyelik, illetve a mért értékek alapján be is avatkozhatnak a gyártási folyamatokba. Az ipari szabványoknak köszönhetően, mint a már említett Profibus vagy Profinet, a különböző gyártók termékei képesek kommunikálni egymással.

A felügyeleti szinthez tartoznak a megmunkáló- és mérőberendezések számítógépei, amelyek a mért adatokat megjelenítik az emberi operátorok számára, illetve az operátorok ezeken az eszközökön képesek beavatkozni a berendezések működésébe és a folyamatokba, ezzel megvalósítva az úgynevezett HMI (*Human-Machine Interface, ember-gép interfész*) funkciót is. Az adatok kijelzése, a folyamatfelügyelet és kezelői funkciók mellett ez a réteg felelős az adatok továbbításáért is a termelésirányítási és adatgyűjtési szintnek. Ezek az eszközökön futnak az olyan adatgyűjtő ügynökök is, amik közvetlenül képesek kommunikálni a vezérlési szint eszközeivel, illetve a egyéb felügyeleti és termelésirányítási rendszerekkel. Ezek az ágensek az adatgyűjtés és továbbítás mellett gyakran fordítási feladatokat is ellátnak a különböző kommunikációs protokollok között.

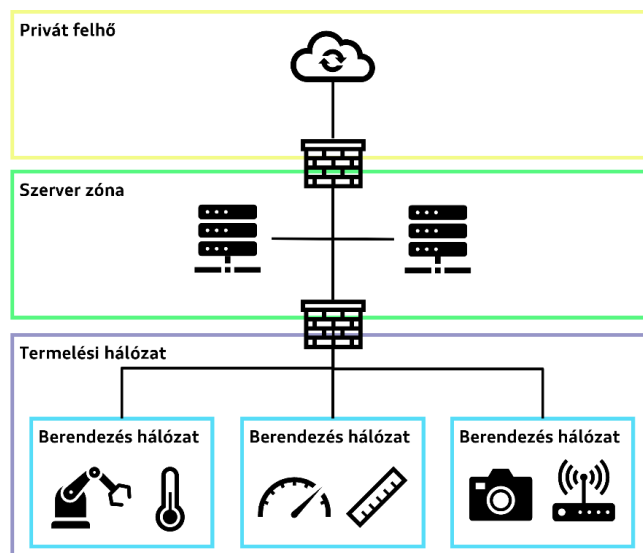
A termelésirányítási szinthez tartoznak a vállaltirányítási, minőségbiztosítási, munkadarab visszakövető és termelési adatelemző rendszerek. A gyártott darabszámokra, készletekre valamint minőségi jellemzőkre vonatkozó adatok és kiértékelések ezekben a rendszerekben találhatóak. A termelési adatok mellett fontos szerepük van a berendezések állapotára és

meghibásodásaira vonatkozó adatoknak és feljegyzéseknek, melyek alapján a karbantartások tervezhetőek, illetve a minőségi problémák és a berendezések meghibásodásai közötti összefüggések feltárhatóak.

A SCADA meglévő megoldásaira és annak kibővítésére épít az Ipar 4.0 az AUDI HUNGARIA-nál. Egyre nagyobb körben kerülnek bevezetésre IoT eszközök, mikroszámítógépek, RFID eszközök, önjáró logisztikai járművek és mesterséges intelligencia megoldások. A gyűjtött adatok körének nagymértékű bővülése, valamint az adatok feldolgozáshoz szükséges kapacitás igénye miatt a felhőalapú megoldások és szolgáltatások is alkalmazásra kerülnek a termelésben.

6.3 Hálózati architektúra

Az adatok kinyeréséhez és továbbításán elengedhetetlen a megfelelő hálózati architektúra. A különböző hálózati komponensen különböző hálózati zónákban találhatóak (19. ábra). Ezek egyrészt a hálózati biztonságot hivatottak szolgálni, másrészt a hálózat karbantarthatóságát segítik. A termelési, a szerver és a felhő zónát tűzfalak határolják.



19. ábra: A hálózati architektúra sematikus ábrázolása

Az egyes berendezések, gyártó- és robotcellák saját berendezés hálózattal rendelkeznek, melyeknek saját IP (*Internet Protocol*) cím tartományuk és alhálózatuk van. Ezekben a hálózatokban találhatóak SCADA szint terepi, a berendezés hálózatra kötött eszközei, mint a

szenzorok, kamerák és a vezérlő szint eszközei, mint a PLC-k és egyéb vezérlőeszközök. Az adatgyűjtés során az ebben a zónában keletkező adatokat kell eljuttatni a célrendszerekhez. A terepi eszközök más ipari szabvánnyal is csatlakozhatnak a berendezés hálózathoz, de a zónából való adattovábbításhoz TCP/IP kapcsolatra van szükség. A külön alhálózatok segítségével elkerülhetőek az IP cím ütközések az egyes berendezések és hálózati eszközeik között. A berendezés hálózathoz a hálózati átjáró egy menedzselt switch vagy router, de bizonyos esetekben a SCADA szerinti felügyeleti szint eszközei is lehetnek, mint például ipari számítógépek vagy PC alapú HMI eszközök.

A termelési hálózat az egyes gyártósorokat és az azokon található eszközöket foglalja magába. A hálózati szeparálhatósága miatt szintén saját címtartománnyal és alhálózattal rendelkezik, átjárás a berendezés hálózat vagy a szerver zóna irányába csak olyan menedzselt hálózati átjárókon és routereken keresztül lehetséges, melyek legalább két konfigurálható porttal rendelkeznek és az egyes portok a csatlakoztatni kívánt hálózatnak megfelelően vannak konfigurálva. Ebben a zónában találhatóak a SCADA felügyeleti szintjéhez tartozó számítógépek és vezérlések.

A szerver és felhő zónában a fölérendelt adatfeldolgozó alkalmazások és adatbázisok találhatóak, illetve itt helyezkednek el a különféle alkalmazások. Ezek a szinteken helyezkednek el a termelésirányítási SCADA rendszerek.

6.4 Az Ipar 4.0 és a SCADA szerepe a kutatásomban

Kutatásom az előző fejezetekben bemutatott technológiai környezetben végeztem. Az adatok kinyerésénél és feldolgozásánál fontos szempont volt, hogy az ipari környezetben is jól alkalmazható, az Ipar 4.0 és SCADA nyújtotta lehetőségeket kihasználó megoldásokat készítsék. Az adatgyűjtési eljárásoknál elvárás volt, hogy azok jól illeszkedjenek a vállalati hálózati architektúrába és skálázható módon kerüljenek implementálásra. A terepi szintől származó adatok összegyűjtéséhez és értelmezéséhez elengedhetetlen a terepi adatgyűjtő- és mérőberendezések, vezérlések, PLC-k és fölérendelt rendszerek ismerete. A disszertáció további részében az ismertetett architektúrára épülő adatgyűjtő és adatfeldolgozó alkalmazások kerülnek bemutatásra.

7. Az ipari adatgyűjtés kihívásai

Vállalati környezetben a berendezések kora rendkívül heterogén, illetve a termékek életciklusa elérheti a 10-15 évet is, és ez gyakran a gyártósor életkorát is jelenti. A géppark a legkorszerűbb berendezések mellett olyan örökölt (*legacy*) rendszereket is tartalmaz, amelyek nem rendelkeznek a nagy volumenű adatgyűjtéshez szükséges kommunikációs képességekkel. Az egyik kihívás az adatok kinyerése a régebbi rendszerekből és az adatok megfelelő formába konvertálása.

Az adatgyűjtési módszerek közé tartozik a közvetlen kapcsolat a szenzorok és az adatgyűjtő rendszerek között, valamint vezetékes vagy vezeték nélküli kommunikációs protollok használata [149]. Az adatokat *edge computing* technológiákkal is lehet gyűjteni és feldolgozni, ezek az eszközök a hálózat szélén, forráshoz közel dolgozzák fel az adatokat [150]. Nagymennyiségű adatok tárolására és kezelésére az ipari adatgyűjtési módszerek adatbázisokat és az adattavak is használnak [151]. Adatgyűjtésre az Apache Kafka [152] és a RabbitMQ [153] üzenetközvetítők vagy brókerek a magas áteresztőképességük és alacsony késleltetésük miatt gyakran használatosak. Ezek az üzenetközvetítők olyan funkciókat biztosítanak, mint az üzenetek megőrzése, nagy hibatűrő képesség és skálázhatóság, amelyek fontosak a nagy adatmennyiségek kezelésekor. Az üzenetközvetítők használatával az ipari adatgyűjtésben a szervezetek képessé válhatnak az ipari folyamatok valós idejű monitorozására és optimalizálására [154].

Az adatforrások számának növekedésével az adatok és forrásaik azonosítása kihívást jelent. A géppark kora is hatással van a gépek azonosítására. Egyrészt a gépek elnevezésének rossz változáskövetése miatt a fizikai azonosítók és a gépek elnevezése az informatikai rendszerben eltérhet, másrészt egy adott gép azonosítója eltérő és egymásnak nem egyértelműen megfeleltethető a különböző informatikai rendszerekben. Ezek a tényezők nagyban befolyásolják a kapcsolódó adatforrások és adatkészletek azonosítását és összerendelését. Ezen problémák kiküszöbölésére az ipari berendezések azonosítására gyakran használnak elnevezési konvenciókat, melyek szabványos megközelítést biztosítanak a berendezések elnevezésére és azonosítására egy szervezeten belül. Az egyik gyakori módszer egy kód vagy sorszámozási rendszer használata, amely az eszköztípus, -funkció és a hely azonosítására szolgál [155]. Egy másik módszer az olyan akronímák vagy rövidítések használata, amelyek a szervezetben dolgozók számára valamiféle jelentéssel bírnak [156]. Gyakori módszer még a leíró elnevezések használata, amely az eszköz funkcióján alapulnak [157]. A szabványosított és egyedi elnevezések biztosítása érdekében az előző módszerek kombinációja is alkalmazható

[158]. Az elnevezési konvenció módszerének kiválasztása az adott szervezet és az iparág konkrét igényeitől függ.

Az adatgyűjtés másik lényeges szempontja az informatikai rendszerek interfészeinek kezelhetősége és a hálózati teljesítmény. Különböző típusú adatkészleteket tárolnak különböző informatikai rendszerekben, mint például alkatrészkövető rendszerek, gép- és folyamatadat-rendszerek és minőségi adatbázisok. A különböző adatfeldolgozási felhasználási esetek részben vagy teljesen azonos adatforrásokat használnak. Ha minden rendszer és használati eset között külön interfészt használunk, akkor a kapcsolatok száma gyorsan növekedhet az (18) egyenlet szerint.

$$n \frac{(n - 1)}{2} \quad (18)$$

ahol n a csomópontok száma.

A különálló adatfeldolgozó alkalmazások és a különböző informatikai rendszerek közötti pont-pont interfészek elkerülése érdekében az adatkinyerést és -elosztást egységesíteni kell. A következő részben ismertetem az általam kidolgozott adatintegrációs keretrendszer architektúráját és szabályait [S1].

8. Saját eredmények

9. Kutatásom célja

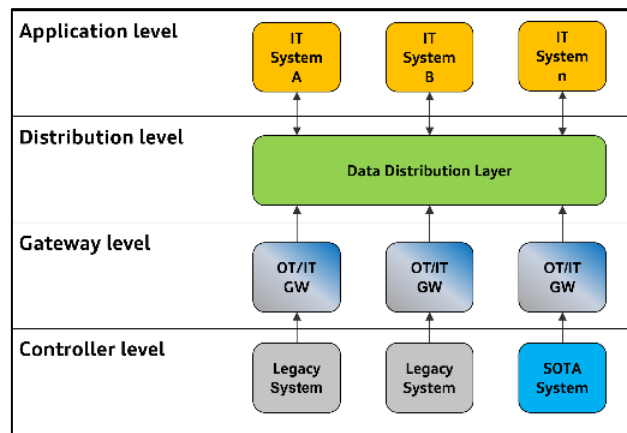
Kutatásom célja az irodalomkutatásban bemutatott kutatásokra, eredményekre és technológiákra építve olyan intelligens rendszerek létrehozása, melyek hozzájárulnak a kibernetikai és kognitív rendszerek fejlődéséhez, valamint az ipari felhasználású mesterséges intelligencia alkalmazások elterjedéséhez.

Az általam kutatott és fejlesztett alkalmazások gyakorlati célja a gyártási költségek csökkentése és a gyártási minőség növelése. A kidolgozott adatgyűjtési eljárások, valamint a gyakorlati ipari alkalmazások segítségével az emberek által nem, vagy csak nehezen felismerhető anomáliák és összefüggések automatizált felismerésével, a hamis pozitív hibák csökkentésével, illetve a karbantartási feljegyzések és gépállapotok elemzésével olyan autonóm rendszerek kialakításán dolgozom, melyekkel a fent említett célok elérhetőek. A minőségi és gazdasági hasznon túl, a döntéshozatali reakcióidők is gyorsíthatók, valamint a géphibák és a selejtarány minimalizálásával az energiafelhasználás és a környezeti terhelés is csökkenthető, ezáltal a kutatott alkalmazások hozzájárulnak a fenntarthatóbb, környezetkímélőbb gyártáshoz.

10. Adatintegrációs keretrendszer

A 7. fejezetben bemutatott, heterogén ipari környezet által okozott adatgyűjtési nehézségekre válaszul, valamint a számítási intelligencia ipari alkalmazásainak támogatására egy adatintegrációs keretrendszert dolgoztam ki, melyhez kapcsolódó első tézisem a fejezet végén mutatok be. Az adatintegrációs keretrendszer [S1] olyan architektúrais összetevők és szabályok összessége, amelyek megoldást javasolnak az adatok gyűjtésére, azonosítására és terjesztésére ipari környezetben. Az architektúra fő részei (20. ábra):

- Vezérlőszint
- Átjáró szint
- Elosztási szint
- Alkalmazási szint



20. ábra: Az adatintegrációs keretrendszer szintjei [S1]

A komponensek kommunikációja lehet kétirányú, de ez nem kötelező.

A vezérlőszint olyan forrásrendszereket tartalmaz, mint a PLC-k, NCU-k, IoT képes eszközök és egyéb automatizálási hardverek. A legkorszerűbb komponensek képesek magasabb szintű protokollokat alkalmazni és eseményvezérelt kommunikációra, például MQTT kapcsolatra, de bizonyos esetekben az adatok más protokollokkal való kinyerése és az üzenetek átalakítása szükséges.

Az átjáró szint funkciója az üzemeltetéstechnológiai (*OT*) hálózat fizikai elkülönítése a vállalati IT hálózattól. A biztonságon kívül ennek a szintnek egyéb feladatai is lehetnek, mint például a bejövő üzenetek pufferelése hálózati hiba esetén, vagy a régebbi eszközök lekérdezése, adatgyűjtő ügynökök hosztolása és a protokollok fordítása.

A disztribúciós réteg célja az üzenetek továbbítása a célrendszerekhez. Ennek az architektúrának az egyik előnye, hogy a forrásrendszernek nem kell ismernie a vevőt, csak elküldi az üzenetet a terjesztési szolgáltatásnak, és továbbítja az üzenetet a kijelölt rendszernek, amely előfizetett az adatforrásra.

Az alkalmazási szint tartalmazza a örökölt informatikai rendszereket és egyéb felhasználási eseteket, amelyek a terepi eszközökből és a kapcsolódó informatikai rendszerekből származó adatokat dolgozzák fel. Ezek a komponensek a disztribúciós rétegen keresztül kommunikálnak, így elkerülhetők a pont-pont interfészek a rendszerek között, és az adatok más rendszerek számára is felhasználhatók, nem szükséges az adatokat különálló interfészekon keresztül duplikálni.

Az architektúra önmagában nem tudja garantálni a megbízható és kezelhető üzenetáramlást a komponensek között, további szabályokra van szükség a kommunikáció kezeléséhez. Ezeket a szabályokat az adatkezelés határozza meg, és a következők:

- Az adatforrások szabványosított elnevezési konvenciója
- Szabványosított üzenetstruktúrák
- Különálló csatornák az előre meghatározott üzenettípusokhoz

10.1.1 Szabványosított névkonvenció

Az adatok forrását egyértelműen meg kell határozni [159][160] annak érdekében, hogy az információ a megfelelő adatfeldolgozóhoz kerüljön, illetve a más IT rendszerekben található, kapcsolódó adatokat egyértelműen össze lehessen rendelni. Az adatforrások és adatok egyértelmű azonosíthatósága érdekében bevezettük a forráskomponensek szabványos elnevezési konvencióját [S1]. Az egyes hálózati komponensen jelölése nem csak a forrást azonosítja, hanem további információkat is tartalmaz, például az egység helyét, hierarchiáját és típusát. Az elnevezési konvenció más eszközök azonosítására is alkalmas, mint épületek, csarnokok, létesítményi berendezések, logisztikai járművek, tárolók. Az elnevezési konvencióra példák az 1. táblázatban láthatók.

1. táblázat: Az egységes névkonvenció példái [S1]

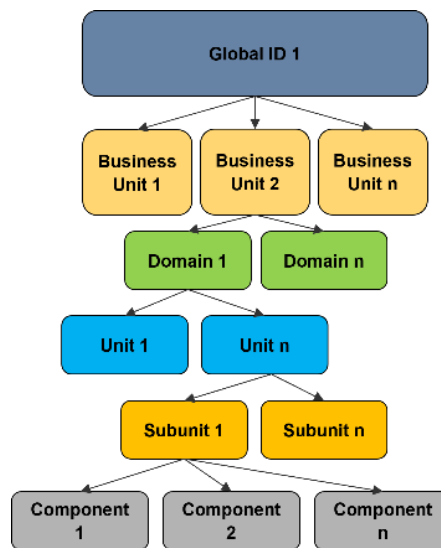
<i>Globális azonosító</i>	<i>Üzleti terület azonosító</i>	<i>Tartomány</i>	<i>Egység</i>	<i>Alegység</i>	<i>Komponens</i>
01	P	TartományA	MG0012	MA001	MS01
02	L	TartományB	TU0123		
01	F	TartományC	AT0200	CP012	SS01

A kódolás hat, fastruktúrába rendezett szintből áll (2. ábra), amelyek mindegyike a hierarchia egy külön szintjét azonosítja. Az egyes szintek használata nem kötelező, de a jelölésnek a fentről lefelé irányuló struktúrát kell követnie, kezdve a globális azonosítóval. Az 1. táblázat a szabványos elnevezési konvenció felépítését és három példát mutatja be (fentről lefelé, balról jobbra olvasva):

- Az 1. gép (MA001) 1. mérőrendszere (MS01) a 12. gépcsoportban (MG0012), az A tartományban, termelési terület (P), Gyár: 01.
- 123-as vontatóegysége (TU0123), B tartomány, Logisztikai terület (L), Gyár: 02.

- 1. sebességérzékelője (SS01) a 12-es számú kompresszornak (CP012) a 200-as légtechnikai berendezésben (AT0200) a C tartományon belül, Létesítménykezelési terület (F), Gyár: 01.

A komponensek jelölését szabványosítottam és egy kódkönyvtárban leírtam. A szakaszok határoló karakter tetszőleges, függ a használati esettől vagy az adatokat feldolgozó rendszertől, a leggyakrabban használt karakterek a kötőjel (-), a törtvonal (/) és a kettőspont (:). Az standardizált elnevezési konvenció fastruktúrája a 21. ábrán látható.



21. ábra: A szabványosított elnevezési konvenció fastruktúrája [S1]

10.1.2 Szabványosított üzenetstruktúra

A szabványos elnevezési konvenció mellett erősen ajánlott egy szabványos üzenetstruktúra [S1]. Az üzenetek egységes szerkezete függetleníti az adatgyűjtést és az üzenetprotokollokat, lehetővé teszi az átviteli protokoll cseréjét az adatáramlás megzavarása nélkül, valamint megkönnyíti a gyűjtött adatok kezelését.

Az adatelosztási keretrendszerhez való csatlakozáshoz a résztvevő rendszernek csak a szabványos üzenetstruktúrát kell használnia. Ha a forrásrendszer megfelel az üzenetstruktúra követelményeinek, az adatgyűjtés technológiája és az átviteli protokoll tetszőleges lehet, amennyiben azt az elosztási réteg kezelni tudja.

A szabványos üzenetstruktúra a JavaScript Object Notation [161] (*JSON*) formátumon alapul. A JSON egy könnyű, önleíró szöveges objektum. A szöveges formátum lehetővé teszi

az adatok programozási nyelvtől független értelmezését, ezért számítógépek vagy programok közötti adattárolásra, illetve adatküldésre szolgál.

Az összes üzenetnek kötelező tartalmaznia a forrás azonosítóját, az üzenet időbélyegét, az üzenet verziója és az elveszett üzenetek számlálóját (hányszor próbálta már kézbesíteni az üzenetet a forrásrendszer). Az összefüggő adatkészletek, pl. energiaadatok egy adott berendezésről, előre meghatározott csatornákhöz vagy témákhoz vannak hozzárendelve, így az összes többi tartalom az üzenet típusától függ. Példa a 22. ábrán látható JSON-üzenetre.

```
{
  "SourceID": "01/P/DomainA/MG0012/MA001",
  "TopicRelatedData1": "Data1",
  "TopicRelatedData2": "Data2",
  "TopicRelatedData3": 123,
  "OtherElements": ["..."],
  "MessageTimeStamp": "2022-12-11T22:54:00.000+01:00",
  "MessageVerion": "Example_01.1",
  "LostMessages": "0"
}
```

22. ábra: Szabványosított JSON üzenet [S1]

Szintén jó gyakorlat az üzenetek témaköreinek olyan rendszerezése, amely a szabványos elnevezési konvenciót használja. Ezzel a módszerrel az elnevezési konvenció minden hierarchiaszintje és minden üzenettípusa elérhető. Több elem kiválasztása helyettesítő karakterekkel (#) lehetséges. A 2. táblázat a különböző témajelöléseket és azok jelentését mutatja be.

2. táblázat: Szabványosított névkonvenció az üzenetek megnevezésében

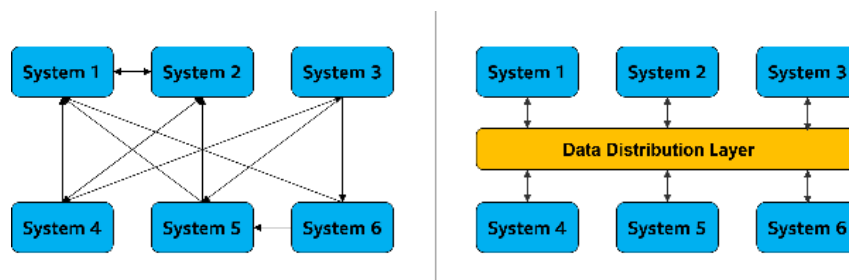
<i>Témakör</i>	<i>Jelentés</i>
01/P/#	Az 1-es gyár termelési területének összes üzenete
01/P/TartományA/##/Energia	Az 1-es gyár termelési területén az A tartományon belüli összes egység és alegység energia üzenete

10.1.3 Adatelosztási réteg

Az összegyűjtött adatok forrás és cél közötti átviteléhez adatelosztási rétegre van szükség [S1]. Az ipari felhasználási esetek skálázható, lazán csatolt és dinamikus hálózati topológiát igényelnek, így a publikáló-feliratkozó (pub-sub) üzenetküldési technológia [162][163] használata célszerű.

Pub-sub üzenetküldésnél a publikáló félnek (forrásnak) nem kell tudnia, hogy ki az előfizető (célállomás), csak közzé kell tennie az üzeneteket az adatelosztási rétegbe a megfelelő témában, majd a szolgáltatás továbbítja az üzeneteket a megfelelő előfizetőknek, akik feliratkoztak az adott témakörre. Ez a tulajdonság biztosítja a pub-alrendszerek laza csatolását és skálázhatóságát. A témakörök az egyes adattípusok logikai csatornáit, az előfizető megkapja az összes adatot, amely a feliratkozott témakörökben publikálásra kerül. A résztvevők egyszerre lehetnek publikálók és előfizetők, illetve tetszés szerint csak az egyikük. Az üzenetküldési műveletek nem korlátozódnak egy témára, hanem a közzététel és az előfizetés is különböző témákban történhet.

Ennek az architektúrának a fő előnye, hogy a kommunikáló rendszerek száma további pont-pont interfészek bevezetése nélkül nagymértékben skálázható, így az architektúra átlátható és kezelhető marad. A 23. ábra a hagyományos interfészeket (balra) és a pub-sub adatelosztási réteggel való interfészeket mutatja.



23. ábra: : Hagyományos interfészek (balra) és interfész az adatelosztási réteggel (jobbra)

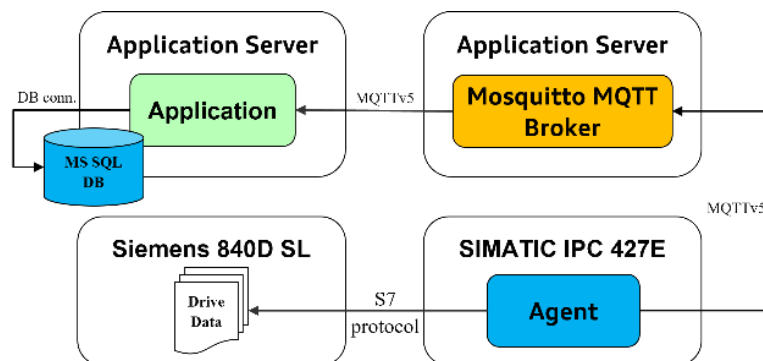
10.1.4 Az adatintegrációs keretrendszer gyakorlati megvalósítása

Ebben a részben az előzőekben ismertetett adatgyűjtési módszer gyakorlati megvalósítását mutatom be [S1]. A kísérleti összeállításban összegyűjtöttem az aktuálisan dolgozó megmunkáló szerszám számát, a szerszám kívánt és hátralévő éltartamát, a Z-tengely pozícióját, egy 3 tengelyes esztergagép főorsó előtolását és áramát.

A keretrendszer a legkorszerűbb, és a régi informatikai rendszerekkel is használható. A rendszerek egy része natív módon képes kommunikálni az elosztási réteggel, de vannak esetek, amikor protokollfordításra és interfészekre van szükség, így nem kerülhető el az adatgyűjtő ügynökök használata.

Vannak más esetek is, amikor a forrás informatikai rendszer nem tudja a szabványos elnevezési konvenciónak megfelelően megadni a forrásazonosítót, ebben az esetben a forrásazonosító fordítása ügynökök használatát is igényli.

Az alábbi gyakorlati megvalósítás a keretrendszer használatát mutatja be örökölt, elavult OT rendszerek esetén, de örökölt IT rendszerek esetén a keretrendszer felépítése a forrásrendszer kivételével megegyezik. Ha a forrás informatikai rendszer képes kommunikálni az elosztási réteggel és a szabványos elnevezési konvenciónak megfelelően a forrásazonosítót is biztosítani tudja, az ügynökök használata elkerülhető. A gyakorlati megvalósítás (24. ábra) architektúrája a 10. fejezetben leírt elemeket tartalmazza.



24. ábra: Az adatintegrációs keretrendszer gyakorlati megvalósításának architektúrája

Az adatforrás egy Siemens 840D Solution Line NCU-t integrált S7-300 PLC-vel. Ez az eszköz Profineten keresztül képes kommunikálni az ipari Etherneten.

Az OT/IT Gateway-hez egy SIMATIC IPC 427E ipari PC-t választottam Ubuntu 20.04.5 LTS operációs rendszerrel, amely az adatgyűjtő ügynök hosztolása mellett az OT és az IT rendszerek fizikai szétválasztásáért is felelős. Az OT/IT-átjáró több ügynököt is hosztolhat különböző tárolókban, például Docker- vagy LXC-tárolókban, illetve az OT/IT átjáró lehet távoli szerver is. Az egyszerűség kedvéért ebben a munkában egy fizikai hardvert és egy

ügynököt használtam. Az integrációban résztvevő komponensek a jelenlegi standard környezet részét képezik a vállalatnál, használatuk az IT biztonsági részleg által előírt és támogatott.

Az adatgyűjtő ügynök egy C# nyelven írt .NET alkalmazás, a kommunikációhoz az S7 protokollt megvalósító Sharp7 könyvtárat használtam. Az elosztási réteghez egy Eclipse Mosquitto MQTT brókert [164] használtam, amely egy Windows alkalmazásszerveren található. Az adatok tárolására egy egyszerű .NET middleware-t hoztam létre, amely előfizet az adott témákra, és az adatokat egy Microsoft SQL Server 2019-ben tárolja. A middleware külön Windows szerveren fut. A gyűjtött adatok a 3. táblázatban láthatók.

3. táblázat: A gyűjtött adatok [S1]

<i>Adat</i>	<i>Változó</i>	<i>Paraméter</i>	<i>Gépi adat</i>	<i>Formátum</i>
Aktuális szerszám száma (ToolNo)	/Channel/State/actTNumber	-	\$P_TOOLNO	UWord
Kívánt éltartam	/Tool/Supervision/data[x,y]	ToolNo, 6	\$TC_MOP13	Double
Hátralévő éltartam	/Tool/Supervision/data[x,y]	ToolNo, 4	\$TC_MOP4	Double
Főorsó előtolás	/DriveHsa/State/actualSpeed	-	\$MD_1701	Float
Főorsó áram	/DriveHsa/State/actualCurrent	-	\$MD_1708	Float
Z-Tengely pozíció	/Nck/MachineAxis/measPos1[axis]	3	-	Double

A mintavételi sebesség a kommunikáció sebességének függvénye, amely a numerikus vezérlőegység hardver típusától és a felhasználói programtól függ, de nem lehet gyorsabb, mint a PLC legkisebb elméleti ciklusideje. Ebben az esetben az adatgyűjtő mintavételi sebessége 500 ezredmásodperc. Az ügynök elküldi az összegyűjtött adatokat az MQTT brókernek, a témakörök, ahová az ügynök közzéteszi az üzeneteket, a „01/P/TestDomian/MG0001/MA001/...” témakörön belüli altémák, nevezetesen a „DesiredWorkpieceCount”, „RemainingWorkpieceCount”, „MainSpindleFeed”, „MainSpindleCurrent” és „ZAxisPos”.

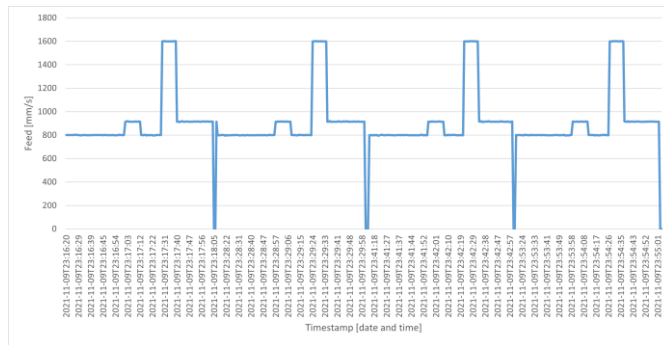
A Eclipse Mosquitto MQTT bróker beállításának egyszerűsége érdekében csak a bróker minimálisan szükséges beállításait használtuk. A szabványos portokat használtuk, 1883 a nem biztonságos, 8883 a biztonságos kapcsolathoz a Transport Layer Security-vel (*TLS*). Ahhoz, hogy a kliens csatlakozhasson a brókerhez, létrehoztunk egy felhasználónevet és egy jelszót, valamint a hozzáférés-vezérlési listában (*Access Control List, ACL*) meghatároztuk, hogy mely témakörökhöz férhet hozzá a kliens. A bróker *retain* (megőrzési) módban működik, ami azt jelenti, hogy ha egy új kliens feliratkozik egy témakörre, akkor a bróker az adott témában utoljára kapott üzenetet elküldi a kliensnek. A szolgáltatások minősége (*Quality of Service,*

QoS) QoS 0-ra van állítva, ami azt jelenti, „*fire and forget*” mechanizmust jelent, a bróker pontosan egyszer küldi el az üzeneteket a klienseknek, anélkül, hogy az üzenet megérkezését meg kellene erősítenie a kliensnek. Ez a beállítás lehetővé teszi a legalacsonyabb késleltetésű kommunikációt. QoS 2 és QoS 3 is elérhető, QoS 2 esetén az üzenet kézbesítése legalább egyszer megtörténik, a QoS 3 esetén a bróker pontosan egyszer küldi el az üzenetet, viszont kézfogósos (*handshake*) mechanizmust igényel. Az átláthatóság és a hibakeresés elősegítése érdekében az eseménynaplózás is bekapcsolásra került brókeren.

A middleware egy C# nyelven írt .NET alkalmazás. Az MQTT kapcsolathoz az MQTTnet könyvtárat használtam. A middleware előfizet a megfelelő témakörökre, és a kapott adatokat egy Microsoft SQL adatbázisba írja. Az alkalmazás az Entity Framework és a Data Transfer Objects (*DTO*) segítségével képezi le az adatbázistáblákat.

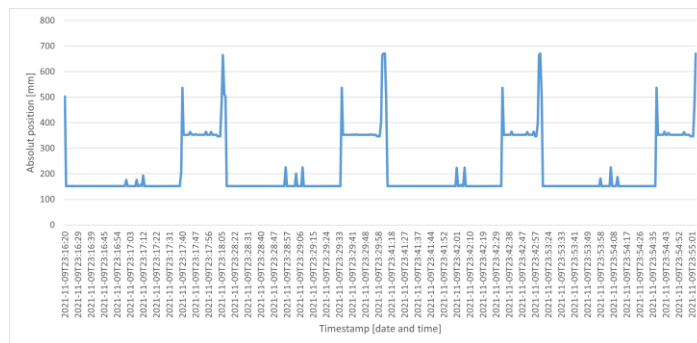
Az adatbázis „*code first*”, ami azt jelenti, hogy az adatbázistáblák az middleware osztályai alapján jönnek létre, ez a technológia és az Entity Framework és a LINQ lehetővé teszi az adatokhoz való erősen típusos hozzáférést. A „*code first*” adatbázisok egyik előnye, hogy adatbázis váltáskor az adatbázis dialektusának megadásával az adatbázis táblák és kapcsolataik a migráció során automatikusan létrehozhatók. Az erősen típusos hozzáférés, illetve az osztályok segítségével a fejlesztőknek nem kell a teljes adatbázis ismerniük a fejlesztés során, mivel kellő információval rendelkeznek arról. A LINQ segítségével az adatok könnyen elérhetőek és a kódból manipulálhatóak. Ebben az esetben az alkalmazás manipuláció nélkül tárolja az adatokat. A különböző témákból származó adatok az SQL-adatbázis különböző tábláiban kerültek tárolásra. Az adatbázis tábla oszlopai a következők: ID (növekményes azonosító elsődleges kulcsként), SourceID (szabványos elnevezési konvención alapuló ügyfél-azonosító), TimeStamp (az MQTT üzenetből származó adatok időbélyege) és maga az érték.

Az összegyűjtött adatok egy 3 tengelyes esztergagép megmunkáló szerszámának száma, a szerszám kívánt és hátralévő éltartama, a Z-tengely helyzete, valamint a főorsó előtolása és árama. A következő diagramok a T6013 szerszám fent említett adatkészletét mutatják be négy egymást követő megmunkálás során [S2]. A főorsó előtolása a 25. ábrán látható. A diagram X-tengelye az adatok időbélyegét mutatja dátum és idő formátumban, az Y-tengely pedig a főorsó előtolását mm/s-ban.



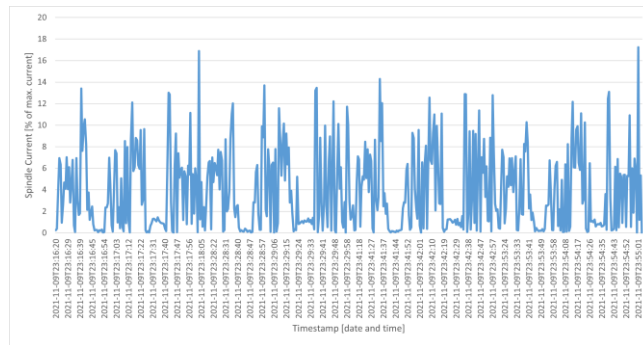
25. ábra: Főorsó előtolás (X-tengely: dátum és idő, Y-tengely: előtolás [mm/s]) [S2]

A Z-tengely abszolút pozíciója a 26. ábrán látható. A diagram X-tengelye az adatok időbélyegeit ábrázolja dátum és idő formátumban, az Y-tengely pedig az abszolút pozíciót milliméterben.



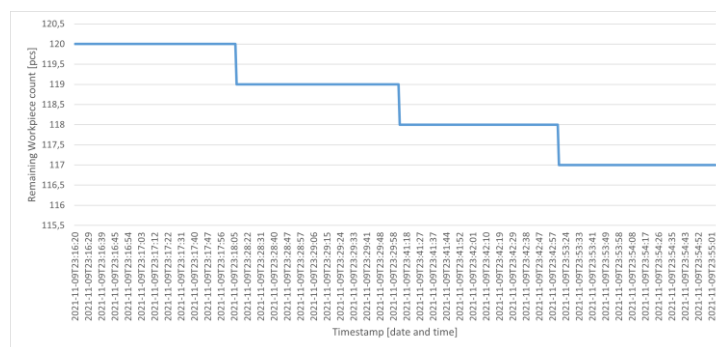
26. ábra: Z-tengely helyzete (X-tengely: dátum és idő, Y-tengely: abszolút pozíció [mm]) [S2]

A főorsó áramát a maximális hajtásáram százalékában a 27. ábra mutatja. A diagram X-tengelye az adatok időbélyegét mutatja dátum és idő formátumban, az Y tengely pedig az orsó áramát a maximum hajtás áram százalékában.



27. ábra: A főorsó árama a hajtás maximális áramának százalékában (X tengely: dátum és idő, Y tengely: a maximális áram %-a) [S2]

A hátralévő munkadarabok száma a 28. ábrán látható. A diagram X-tengelye az adatok időbélyegeit mutatja dátum és idő formátumban, az Y-tengely pedig az adott szerszámhoz tartozó megmunkálható munkadarabok fennmaradó számát.



28. ábra: Fennmaradó munkadarabok száma (X-tengely: dátum és idő, Y-tengely: maradék munkadarabok száma [db]) [S2]

A kívánt munkadarabszám minden szerszámnál egy specifikus, állandó érték, ezért nincs ábrázolva.

10.1.5 Az adatintegrációs keretrendszer termelési adatok gyűjtésére

Az adatintegrációs keretrendszer gyakorlati megvalósítására alapozva az adatgyűjtési eljárás bevezetésre került az AUDI HUNGARIA Zrt. 3. számú főtengely gyártósorára, ahol a feladata a gépállapotok és üzenetek, valamint egyéb termeléssel kapcsolatos adatok gyűjtése. A terület amin bevezettük a megoldást 36 megmunkáló berendezést tartalmaz, amelyek mindegyike az előző szakaszban bemutatott ügynökökön keresztül csatlakozik az adat elosztási réteghez. Az gyűjtött adatok a gépek állapota, gépinformációk, például alkatrész számlálók és ciklusidőmérések, munkadarab mozgások, az operátorok azonosító információi, hibaüzenetek, a berendezések energiafogyasztása, a megmunkálási előtolás override (*felülírási*) értéke és az ügynök MQTT állapotával kapcsolatos jelentések. Az ügynökök által közzétett témák a szabványosított elnevezési konvencióra épülnek. A témák alapja a gépazonosító a gyár hierarchikus szerkezetében, az üzleti egységben, a termelési területen és a termelési vonalon belül, amely követi a mintát: „Gyár azonosító / Üzleti egység / Tartomány / Gyártósor / Gép azonosító / Témakör”. Az egyes témakörökhöz tartozó átlagos napi üzenetszám és az egyes üzenetek mérete a 4. táblázatban láthatók.

4. táblázat: A gyűjtött termelési adatok témaköre, átlagos napi száma és mérete

<i>Témakör (Topic)</i>	<i>Napi átlagos üzenetszám (db)</i>	<i>Átlagos üzenet méret (Byte)</i>
MqttState	2	128
PartMovement	10010	234
Energy	42657	147
MachineState	15504	131
MachineInfo	10010	240
Messages	37053	93
Operator	2	188
Override	251	130

A 36 berendezés naponta átlagosan 115 000 üzenetet küld a brókernek melyek átlagos napi amit a middleware egy SQL-adatbázisban ír a további elemzési és megjelenítési feladatokhoz.

10.1.6 Számítási intelligencia ipari alkalmazásainak kihívásai

A sorozatgyártó üzemek nagymértékben automatizáltak. Az Ipar 4.0 és az IoT eszközök elterjedésével nagy volumenű adat áll rendelkezésre a modellek betanításához. A szériagyártó üzemekben a folyamatok szigorúan szabványosítottak és erősen specializáltak, viszont ez már nem mondható el az adatgyűjtésről, adatelemzésről és adatértelmezésről. Az adatgyűjtés heterogenitása mellett az ipari és vállalati rendszerek sokfélesége, ezen rendszerek összekapcsolhatósága, valamint a strukturált és szabványosított adattárolás hiánya tovább nehezíti a modellalkotást. Az adatok és a jellemzők kiválasztása a sikeres modellalkotás kulcsfontosságú lépései, nagyon fontos szerepe van a folyamat- vagy szakismereteknek és az adatelemzés funkcionális megközelítésének. Az adatintegrációs keretrendszerrel gyűjtött adatokat felhasználva modelleket készítettem, melyeken keresztül a mesterséges intelligencia ipari alkalmazásainak kihívásait, a modellalkotással kapcsolatos kérdéseket és nehézségeket, valamint annak multidiszciplináris jellegére vonatkozó kihívásokat, a modellalkotás klasszikus munkafolyamatait és azok kihívásait az ipari alkalmazások esetén [S3] a következő részben mutatom be.

Az adatok előkészítése az adatelemzés és a modellfejlesztés legelső lépése. Az adat-előkészítés célja a folyamatadatok áttekintése, majd a leginkább megfelelő minták és jellemzők kiválasztása a modellezéshez. A fő feladatok az adatok kinyerése az adatbázisokból és egyéb rendszerekből vagy folyamatokból, az adatstruktúra elemzése és az adatok kiválasztása a modellek felépítéséhez és betanításához. Az adatok és a jellemzők kiválasztása az előzetes szakmai ismertek és a megoldandó probléma alapján történik. Az adatok előkészítése és kiválasztása az adatelemzés és a modellalkotás elengedhetetlen lépései. A gyártásban számos adatgyűjtési rendszer létezik, mindegyik más-más céllal. Az alkatrészkövető rendszer nyomon követi a legyártott alkatrészeket és azok állapotát, egy beépített felügyeleti rendszer méri és tárolja a folyamatjeleket és a mért értékeket, a vállalatirányítási rendszer (*ERP*) pedig további információkat tárol, például pénzügyi vagy beszállítói adatokat. Vannak adatok, amelyek csak az egyik vagy akár több rendszerben is tárolásra, kiszámításra kerülnek. Melyik lesz a vezető rendszer, amely biztosítja az „alapigazságot”? Az adatábrázolás minden rendszerben eltérő lehet. Honnan tudjuk, hogy milyen adatformátumot használjunk? Hogyan konvertáljuk az adatokat a rendszerek között? Ha egy mérőrendszer csak a mért értékeket tárolja és a mérés időbélyegét és az alkatrészazonosítót csak az alkatrészkövető rendszer a redundáns adattárolás megelőzése érdekében, hogyan kapcsoljuk össze a két rendszerben lévő adatokat? Melyik rendszerben melyik időbélyeget használják, a mérés vagy az adatgyűjtés időbélyegét? Hogyan

szinkronizáljuk őket? Ha az összekapcsolandó rendszerekben eltérő az adatforrás azonosítója vagy neve, hogyan kapcsoljuk össze a megfelelő adatokat? Milyen technológiát kell használni az adatgyűjtéshez? Melyek az összegyűjtött adatok legfontosabb jellemzői? Ez csak néhány a valós ipari alkalmazásokban felmerülő adat-előkészítés lehetséges kihívásai közül, amelyekkel foglalkozni kell. Ezeket a kérdéseket nem lehet megválaszolni a rendszer architektúra, az adatgyűjtés, az infrastruktúra és a folyamat pontos ismerete nélkül.

Az adatok minőségének javítása érdekében elengedhetetlen az adatok előfeldolgozása, és a modellek hatékonyabbá tételéhez adatátalakításra is szükség lehet (*feature engineering*) [165]. Ez a folyamat növeli a modell teljesítményét és pontosságát. Ebben a lépésben az időbeli inkonzisztenciát ki kell küszöbölni, a kiugró értékeket és a durva hibákat el kell távolítani, a hiányzó értékeket pótolni kell, ilyen tevékenységek például a minták törlése, a hiányzó értékek becslése és az adatok közötti skálázás különbségének figyelem bevétele (a skálázás vagy a normalizálás lépése) a pontosság növelése és a tanítási idő csökkentése érdekében. Ebben a lépésben megtörténik az adatkészlet felosztása a betanítási, tesztelési és érvényesítési adatkészletekre, amelyeket a betanításhoz (*training*) és a modell ellenőrzéséhez (*testing*) és a teljesítményének értékeléshez használnak (*validation*) [166]. A megfelelő teljesítmény elérése érdekében foglalkoznunk kell tehát az inkonzisztenciával, a hiányzó értékekkel, a kiugró értékekkel, vagy törölnünk kell a sérült mintákat [167]. De ezek a minták valóban sérültek, vagy csak egzotikus értékek a folyamatban? Hogyan pótoljuk a hiányzó értékeket? A minták átlagértékének felhasználása megfelelő? Ha vannak címkék, valóban megfelelőek a kategóriák? Az adatok valósak és jól leírják a folyamatot? E kérdések megválaszolásához elengedhetetlen a fizikai folyamat és a gyártástechnológia ismerete.

Az adatkiválasztás és előfeldolgozás után következik a megfelelő gépi tanulási algoritmus kiválasztása a modell felépítéséhez. Az adatjellemzők részletes elemzése alapján a modell architektúrája és a modell felépítése jól meghatározható. A modell architektúrájának kiválasztása után a modell paraméterei a betanítási adatkészleten való tanítással és teszteléssel finomhangolhatóak úgy, hogy a tanítás során optimalizáljuk a modell veszteségfüggvényét. A legjobban teljesítő modell kiválasztásához először meg kell határoznunk a modell kívánt viselkedését. Az architektúra, az algoritmus, a veszteségfüggvény és az optimalizálási eljárás nagyon eltérő lehet a különböző feladatoknál. Elengedhetetlen a probléma keretbe foglalása. Regresszió vagy osztályozási problémával állunk szemben? Van-e szezonális az adatokban vagy sem? Ez egy idősor vagy diszkrét értékekkel van dolgunk? Elengedő egy önálló modell, esetleg egy kaszkád modellre vagy egy több modelles architektúrára van szükség, különböző

architektúrákkal, amelyek megfelelnek a releváns jellemzőknek? Az ipari folyamatok jól szabványosítottak és ellenőrzöttek, így a termék minősége erősen kiegyensúlyozatlan a pozitív értelemben: a hibás és a selejt termékek aránya relatív alacsony a megfelelő minőségű termékkel szemben. Hogyan lehet betanítani egy osztályozási modellt, ha a bemeneti adatok erősen kiegyensúlyozatlanok? Milyen képzési stratégiát válasszunk? Mekkora adatsomagokkal (*batch size*) és hány cikluson keresztül (*epochs*) tanítsuk a modellt?

A tanítási fázis után a modell teljesítményét validálni kell. A leggyakrabban használt ellenőrzési és teljesítményértékelési módszerek a kereszt-validáció, a modellstabilitás-elemzés, a modell robusztusságának elemzése és a paraméterváltozás érzékenységének elemzése. A modell kiértékeléshez külön tesztelési és validációs adatkészlet szükséges. A modell éles üzembe helyezése előtt elengedhetetlen a validálási folyamat. De ha a bemeneti adatok erősen kiegyensúlyozatlanok, a tanítás és a kiértékelés is hosszú időt vehet igénybe. A gyártás során egy ritka esemény megjelenése napok és évek között változhat. Ha a modell tévesen osztályozza vagy nem tudja észlelni ezeket a ritka eseményeket, akkor a képzési adatok rosszak vagy a modell nem megfelelő? Fel kell-e ismernünk azokat az eseményeket, amelyek néhány évente előfordulnak, vagy sem? Gyakran nincs idő drága és jól előkészített gyártási vagy folyamat tesztek elvégzésére. El kell döntenünk, hogy mely funkciók és események érik meg az időt és a költségeket.

A gépi tanulás ipari alkalmazása nem csak gépi tanulási probléma, hanem összetett, multidiszciplináris mérnöki feladat. Az ML módszerek sikeres ipari alkalmazásához nem csak a szakterület ismerete (*domain knowledge*) [165], hanem az ipari folyamatok, a termékek és jellemzőik, a minőségi jellemzők, az adatgyűjtési módszerek, az automatizálási technológiák és az informatikai architektúra ismerete elengedhetetlen.

Az MI célja a gyártásban, hogy támogassa a döntéshozatalt és a jó minőségű termékek hatékony előállítását, ezért a módszerek alkalmazásait a műhelyben (*shopfloor*), a gyártás közelében kell elhelyezni, ahol a valós érték termelődik, és ahol az eredmények értékelésére és validálására a szaktudás jelen van, valamint ahol a gyors beavatkozással megtakarítások érhetőek el.

A fent említett tapasztalatok alapján a gépi tanulás sikeres ipari alkalmazásához a következő lépésekből álló munkafolyamatot dolgoztam ki [S3]:

1. A probléma megfogalmazása technológiai megközelítéssel, domain knowledge segítségével [165]: meghatározzuk, hogy a modell milyen eseményt, eseteket észlel, illetve

minősít, majd a termék és a gyártási folyamat alapján kiválasztjuk a releváns technológiai paramétereket, jellemzőket. (Ellentétben a klasszikus BigData módszerekkel, ahol minden adatot összegyűjtünk, és igyekszünk mintát találni az adatokban.) A megfelelő paraméterek kiválasztásához konzultálni kell a konstruktőrrel, a folyamat-, automatizálási és minőségbiztosítási mérnökökkel.

2. Forrásrendszerek és adatformátumok azonosítása, adattérképek létrehozása: azonosítjuk, hogy melyik rendszerekből mely adatok nyerhetők ki, az adatok azonosítása és adatszótár készítése. Adatgyűjtési technológia meghatározása a rendszerekből történő adatgyűjtéshez. Szabványos elnevezési konvenció alkalmazása az adatok elnevezésére a különböző rendszerekben. A lépések végrehajtásához az automatizálási és informatikai mérnökökkel, valamint az alkalmazásgazdákkal szükséges együttműködni.

3. Modellkészítés, tanítás, értékelés [166]: a gyártás igényei alapján elkészítjük a modellt az előre meghatározott feladathoz, majd a valós folyamatadatokon teszteljük és validáljuk a modell teljesítményét. Ebben a folyamatban az adattudós, termék-, folyamat- és automatizálási mérnökök vesznek részt.

4. A modell üzembe helyezése és üzemeltetése a termelésben: a modell sikeres kiértékelése után a modell üzembe helyezése a termelésben. Teljes életciklus menedzsment (*Life Cycle Magement, LCM*) szükséges a további fejlesztéshez és szükség esetén finomhangoláshoz. Az LCM folyamatok végrehajtásához szükséges a kapcsolattartás a termék-, folyamat-, automatizálási és minőségbiztosítási mérnökökkel, valamint a folyamat- és alkalmazásgazdákkal egyaránt.

10.1.7 Eredmények összegzése

A 10. fejezetben bemutatam az általam kidolgozott adatintegrációs keretrendszert, mely a heterogén ipari környezetben tapasztalt gyakorlati adatgyűjtési- és elosztás problémákra nyújt megoldást. A keretrendszer segítségével a különböző életkorú és képességű forrásrendszerekből az adatok egységesített formában gyűjthetők és továbbíthatók. A szabványosított névkonvenció segítségével a forrásrendszerek elnevezése standardizálható és az adatok egyértelműen beazonosíthatóvá válnak. Az adatintegrációs réteg bevezetésével elkerülhető az újabb pont-pont interfészek bevezetése, ezáltal a hálózati architektúra egyszerűsíthető, valamint a már az adatintegrációs rétegen közzétett adatokat felhasználni kívánó újabb felhasználási esetek bekötése egyszerűbbé vált.

A bemutatott adatgyűjtő keretrendszert az OT/IT rendszerekből történő adatgyűjtésre és az interfész dzsungel megelőzésére szolgál, ami egyszerűsíti a vállalati hálózat architektúráját és az új adatfeldolgozási felhasználási eseteket gyors integrálását teszi lehetővé. A keretrendszer lehetővé teszi olyan régebbi OT és informatikai rendszerekből származó adatok gyűjtését, amelyek nem képesek a legkorszerűbb kommunikációs protokollokat használni, illetve nem teljesítik az adatkezelési követelményeket.

A javasolt elemekkel, így a szabványosított elnevezési konvencióval, az adatgyűjtő ügynök és az adatelosztó réteg használatával egyszerűsíthető a kapcsolódó adatok összekapcsolása, és kiküszöbölhetők a rosszul menedzselte rendszerből és a változáskezelés hiányából adódó nehézségek. A szabványos elnevezési konvenció a vagyonkezelés részeként is használható.

Az adatelosztási réteg bevezetésével elkerülhetők a pont-pont interfészek, így a hálózatkezelés és üzemeltetés is egyszerűbbé válik. Az adatelosztási réteg az adathozzáférési szabályzatok, a felhasználókezelés és a naplózás révén átláthatóságot és nyomon követhetőséget is biztosít. Konkrét felhasználók vagy kliensek csak azokhoz a témákhoz férhetnek hozzá, amelyek a bróker hozzáférés-vezérlési listájában engedélyezettek, írási és olvasási jogosultságok is beállíthatók, valamint a kliens csatlakozási kísérletei is naplózásra kerülnek. Ezek a funkciók lehetővé teszik az IT biztonsági szabályoknak való megfelelést is.

A keretrendszer lehetővé teszi a régebbi rendszerekből történő adatgyűjtést, így heterogén rendszerekből szabványosított módon gyűjthetők az üzemi és folyamatadatok. A szabványos üzenetstruktúra megkönnyíti az adatkezelést és -tárolást, az újonnan csatlakozó klienseknek csak a szabványos elnevezési konvenció és üzenetstruktúra követelményeinek kell megfelelniük ahhoz, hogy adatokat küldjenek a brókernek. Ez a funkció lehetővé teszi az

adattárolást az adatforrások további testreszabása nélkül. A szabványos üzenetstruktúra meghatározza az adatok témáját is. Ez a tulajdonság lehetővé teszi, hogy az ügyfelek csak arra a témakörre iratkozzanak fel, amire valóban szükségük van. Ez a funkció kiküszöböli az adattavakra való igényt is, minden használati esetnek csak azt az adatot kell összegyűjtenie, amire valóban szüksége van.

Abban az esetben, ha új felhasználási esetnek szüksége van az üzenetközvetítőn elérhető adatokhoz, új felhasználót kell létrehozni a közvetítőn, és az új ügyfél által igényelt témában adott hozzáférési jogosultság után feliratkozhat a témára. és megkezdheti az adatgyűjtést a brókertől. Ez a funkció lehetővé teszi az új adatfeldolgozási felhasználási esetek, például gépi tanulási modellek, mesterséges intelligencia (AI) alapú adatfeldolgozók és prediktív rendszerek gyors bekapcsolását.

Az adatintegrációs keretrendszer jó kiindulási alapot biztosít az ipari mesterséges intelligens alkalmazásokhoz a folyamat- és gépadatok adatgyűjtésének, kezelésének és elosztásának leegyszerűsítésével, az új adatgyűjtések pedig könnyen bevezethetők az adatelosztási rétegbe.

Az elméleti alapokon túl bemutattam, hogyan használható az általam kidolgozott metodika NC adatok gyűjtésére Siemens 840D SL vezérlések esetén, valamint az adatgyűjtés kiterjeszttem egy 36 megmunkáló berendezésből álló gyártósorra, ahol termelési és diagnosztikai adatok gyűjtünk.

Az adatintegrációs keretrendszerrel gyűjtött adatokat felhasználva modelleket készítettem, melyeken keresztül a mesterséges intelligencia ipari alkalmazásainak kihívásait, a modellalkotással kapcsolatos kérdéseket és nehézségeket, valamint annak multidiszciplináris jellegére vonatkozó kihívásokat a disszertációm 10.1.6-os fejezetében mutattam be. Ismertettem az ipari folyamatok, a termék és gyártástechnológia, a hálózati architektúra és az IT rendszerek ismeretének és a *domain knowledge* fontosságát. Azonosítottam a mesterséges intelligencia sikeres ipari alkalmazásához szükséges résztvevőket és stakeholder-eket, valamint javaslatot tettem a gyakorlati alkalmazások elkészítésének egy lehetséges munkamenetére.

Az MI sikeres ipari alkalmazásának munkamenete a következő:

- 1. A probléma megfogalmazása technológiai megközelítéssel, domain knowledge segítségével.** Stakeholder-ek: konstruktőr, a folyamat-, automatizálási és minőségbiztosítási mérnökök.

- 2. Forrásrendszerek és adatformátumok azonosítása, adattérképek létrehozása.** Stakeholder-ek: automatizálási és IT mérnökökkel, alkalmazásgazdák.
- 3. Modellkészítés, tanítás, értékelés.** Stakeholder-ek: adattudós, termék-, folyamat- és automatizálási mérnökök.
- 4. A modell üzembe helyezése és üzemeltetése a termelésben.** Stakeholder-ek: termék-, folyamat-, automatizálási és minőségbiztosítási mérnökök, folyamat és alkalmazásgazdák.

I. Tézis

Kidolgoztam és a gyakorlatban megvalósítottam egy újszerű adatintegrációs keretrendszert, mely hardver- és szoftverkomponensek, valamint névkonvenciók és szabályok segítségével heterogén ipari környezetben is standardizált és skálázható módon valósítja meg az adatgyűjtést. A hálózati entitások egyértelmű és standardizált azonosításával, az üzenetstruktúra szabványosításával, a hálózati architektúra egyszerűsítésével és az új adatfeldolgozó felhasználási esetek gyors integrálásának lehetőségével. Az adatintegrációs keretrendszer bevezetése növeli a transzparenciát és megkönnyíti a hálózat menedzselhetőségét, valamint lehetőséget teremt a mesterséges intelligencia módszerek alapjául szolgáló nagy volumenű ipari adatgyűjtéshez.

Munkám során azonosítottam a mesterséges intelligencia ipari alkalmazásainak kihívásait és nehézségeit, valamint bemutattam annak multidiszciplináris jellegét és rávilágítottam a *domain knowledge* fontosságára. Meghatároztam az MI sikeres ipari implementációjának stakeholder-eit és javaslatot tettem a sikeres ipari alkalmazás munkamenetére.

A tézisemhez tartozó publikációim a következők: [S1] [S2] [S3]

11. Számítási intelligencia ipari felhasználási esetei

A gépi tanulási módszerek ipari alkalmazásai ígéretes eredményeket hoztak az intelligens gyártás és minőségbiztosítás területén. A mesterséges intelligencia potenciális ipari felhasználási esetei a mintafelismerés, osztályozás, anomália- vagy hibaészlelés és hiba-előrejelzés a prediktív karbantartáshoz. A következő fejezetekben bemutatom a munkám során vizsgált problémákat, a kutatásom eredményeit, valamint a számítási intelligencia gyakorlati ipari alkalmazásait, eredményeit és a hozzájuk kapcsolódó téziseim.

11.1.1 Gyártási anomálfelismerés gépi tanulási módszerekkel

Az AUDI HUNGARIA Zrt-nél a belső égésű motorok és különböző alkotó elemeinek gyártása, valamint összeszerelése során rengeteg mérési adat keletkezik. A mérési eredmények kiértékelése és ellenőrzése elengedhetetlen a szigorú gyártási követelmények és a minőségbiztosítási előírások betartásához. Az autógyártásban a sorozatgyártás természete miatt, valamint a törvényi előírásoknak köszönhetően több évre visszamenőleg, hatalmas mennyiségben állnak rendelkezésre a különféle mérési adatok és feljegyzések.

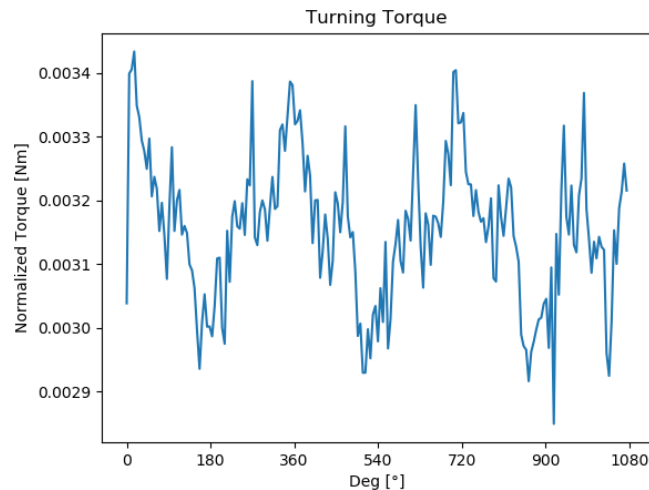
A munka során a gyártási folyamatok közben keletkezett nagy adathalmazra épülő, az iparban lehetségesen használható módszerek felkutatására és gyakorlati alkalmazására összpontosítottam [S4], mint a K-átlag, főkomponens analízis (PCA), lineáris diszkriminancia-analízis (LDA) [168], DBSCAN és az LSTM neurális hálózatok, különös tekintettel a főtengelyek átforgatási nyomatékok és a csapágykárral járó motorhibák közötti lehetséges összefüggések és egyéb anomáliák feltárására.

Az adatok előfeldolgozására és a gépi tanulási modellek elkészítésére a Python programozási nyelvet választottam. Az adatok formázására, előfeldolgozására és megjelenítésére a numpy, a pandas és a matplotlib.pyplot csomagokat, illetve a gépi tanuláshoz és a klaszterezéshez a scikit-learn modult és a TensorFlow keretrendszert használtam.

Munkám során egy motorszerelde terepi méréseit és egyéb reklamációs, valamint minőségbiztosítási feljegyzéseit elemeztem és értékeltem ki a gépi tanulási módszerek segítségével.

Az adatok forrása egy főtengely átforgatási nyomaték mérő berendezés volt. A gyártási folyamat egy szakaszában a motorblokkba beépítésre kerülnek a főtengelycsapágyak, a főtengely, a hatókarok, hajtókarcsapágyak, dugattyúk és egyéb alkatrészeik, tömítései. A szerelési folyamat végeztével egy automata mérőberendezés minden egyes motor esetén megméri az elkészült motorokban a főtengelyek átforgatási nyomatékát úgy, hogy arra kapcsolódva megforgatja azt. A mérés során az átforgatásra használt villamos motor áramfelvételéből kiszámítja az egyes szöghelyzetekhez tartozó nyomatékértékeket. A mérés közben három teljes körbefordulás során, 5,4 fokonként, összesen kétszáz ponton kerülnek az adatok tárolásra. A mérőberendezés csak azt vizsgálja, hogy a nyomaték a mérés során egy minimum-maximum toleranciaablakon belül van-e, a görbe lefutásából, jellegéből nem von le következtetést. A szöghelyzet-nyomaték párokon túl a rendszer tárolja a motor egyedi azonosítószámát, a mérés időpontját, a mérőprogram számát, a mérésre vonatkozó diagnosztikai adatokat és a mérés kimenetelének eredményét.

Az elemzéshez egy adott motortípushoz tartozó 2016-os mérési eredményeket használtam fel. Az előfeldolgozás után összesen 141 351 mérés került kiértékelésre. Egy tipikus átforgatási nyomaték mérés eredménye az 29. ábrán látható.



29. ábra: Tipikus átforgatási nyomaték görbe

Az adatok előfeldolgozása során szükség volt az ismétlődő adatok, valamint a hiányos és hibás bejegyzések eltávolítására [167]. Az adatok vizsgálata során csak a „jó” eredményű mérésekkel folytattam a munkát, tehát a mérések azon halmazát vizsgáltam, mely tartalmazza ellenőrzésen potenciálisan túljutó hibás motorokat.

Az előfeldolgozás során az adatok normalizálásra [169][170] is kerültek. A 200 dimenziós bemeneti vektorokat egyenként egységnyi normájúvá skáláztam a scikit-learn és az L2-norma (19) segítségével.

$$\|x\| = \sqrt{\sum_{i=0}^n x_i^2} \quad (19)$$

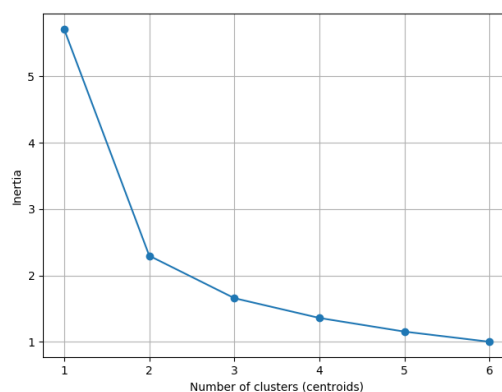
Az adatokban fellelhető mintázatok felderítésére a K-átlag++ algoritmust használtam. A K-átlag++ úgy osztályozza a mintákat, hogy azokat C különálló csoportba sorolja úgy, hogy a klaszteren belüli távolságok négyzetösszegét (20), az úgynevezett tehetetlenséget vagy másnéven inerciát minimalizálja.

$$\sum_{i=0}^n \min_{\mu_j \in C} (\|x_i - \mu_j\|^2) \quad (20)$$

Minden klasztert az átlagos μ_j jellemezi, melyet centroidnak hívnak. Az inercia egy nem normalizált mérték: csak azt tudjuk, hogy az alacsonyabb értékek jobbák és a nulla az optimális. Ezt kihasználva, az könyökmódszer [29] segítségével határoztam meg az adatokban fellelhető és elkülöníthető csoportok ideális számát. Iteratív módon növeltem a feltételezett csoportközéppontok számát, majd minden egyes lépésben az algoritmus kiszámolta a teljes mintára vonatkozó inerciát. Addig növeltem a csoportok számát, míg az inercia nem teljesítette azt a kritériumot, hogy egy újabb csoport hozzáadása esetén számottevően már nem csökken az inercia értéke.

Nagyon magas dimenziójú terekben viszont az euklideszi távolságok felfújódnak, ez az úgynevezett „dimenzionalitás átka”. Ennek leküzdésére egy dimenziócsökkentő algoritmust, a PCA-t [19] használtam. A 200 dimenziós vektorokból 2 dimenziós vektorokat készítettem úgy, hogy a két legnagyobb varianciájú főkomponenst tartottam meg. A PCA által dimenziócsökkentett adatokat a K-átlag++ algoritmus segítségével csoportosítottam.

A klaszterek számának meghatározásához az előzőekben említett Elbow-method eljárást alkalmaztam. A K-átlag++ algoritmus minden iterációs lépésben növelte a centroidok számát, amivel a (20) kifejezést igyekszik minimalizálni. A kifejezés értékének alakulása a klaszterszám függvényében 30. ábrán, értékei pedig az 5. táblázatban láthatóak.

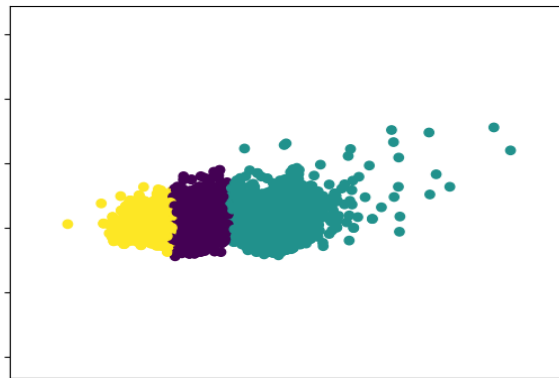


30. ábra: Inercia alakulása a klaszterek számának függvényében

5. táblázat: Az inercia értéke a centroidok számának függvényében

<i>Centroidok száma</i>	1	2	3	4	5	6
<i>Inercia</i>	5.7099	2.2981	1.6593	1.3609	1.1554	1.0023

Az 4. táblázatból és 31. ábráról is leolvasható, hogy a harmadik centroid hozzáadása után a görbe meredeksége csökkenő tendenciát mutat. Az inercia egy mértékegység nélküli metrika, de az elmondható róla, hogy a nullához közeli értékek jobbák. Ellenben fontos megjegyezni, hogy nem minden esetben célravezető a nulla körüli tehetetlenség elérése. Ha a minta nagy szórással rendelkezik (mint esetünkben), további centroidok hozzáadásával az inercia csökkenthető, ám a valóságban az történik, hogy minden egyes kilógó, szélsőséges egyedhez egy saját csoportot illesztünk [S4]. A PCA által dimenziócsökkentett mintákhoz a K-átlag által hozzárendelt csoportok a 31. ábrán láthatók.



31. ábra: A PCA-val 2D-re csökkentett adatok és a K-átlag++ csoportosítás eredménye

Az x tengelyen az első főkomponens, az y tengelyen pedig a második főkomponens került ábrázolásra, a koordináták dimenzió nélküli értékek. A kétdimenziós ábra alapján nem egyértelmű, hogy a csoportok egyértelműen elkülöníthetők, ezért a radikálisabb szétválasztásához az LDA [168] eljárást alkalmaztam.

A PCA egyik alternatívája a lineáris diszkriminancia-analízis (LDA). Az LDA a független változók olyan lineáris kombinációját képes megtalálni, amely a függő változó alapján kialakított csoportokat a lehető legjobban megkülönbözteti. Az LDA első lépése a diszkrimináló függvény(ek) kiszámítása. A függvények száma úgy számítható ki, hogy a függő

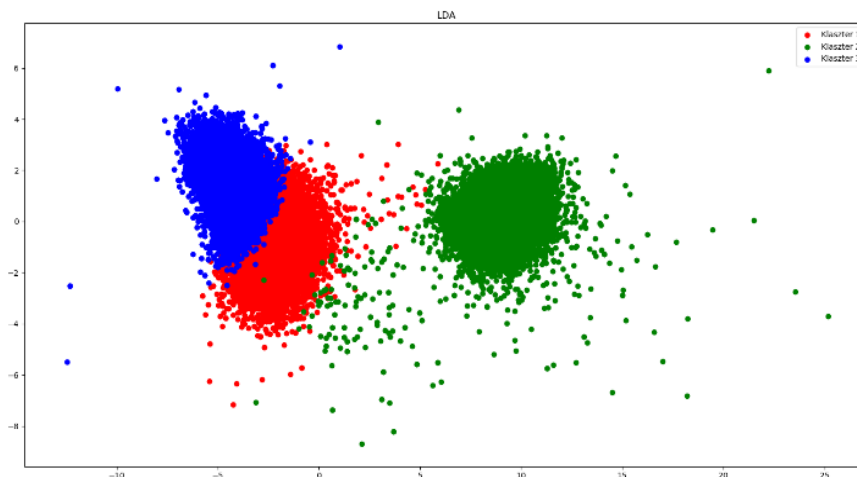
változó lehetséges kimeneteleinek száma -1 és a független változók számának a minimumát vesszük. Az általános diszkrimináló függvényt a (21) egyenlet írja le.

$$D_j = d_{0j} + d_{1j}x_1 + d_{2j}x_2 + \dots + d_{kj}x_k \quad (21)$$

ahol j az adott függvény sorszáma, az x_i -k a mért független változók, d_0 konstans, a d_{ij} az x_i mért változók j -edik diszkrimináló függvényéhez tartozó együtthatója.

A függvény akkor optimális, ha a függő változó által meghatározott csoportok közötti külső négyzetösszeg és a csoportokon belüli négyzetösszeg hányadosa maximális. Az algoritmus hátránya, hogy szükséges megadni, hány csoportba kívánjuk az adataikat sorolni, ezért alkalmazása előtt szükséges felderíteni a mintában rejlő csoportokat, például a K-átlag és a könyök módszer segítségével.

Az LDA a diszkriminálásnak köszönhetően a térben jobban elkülöníti az egyes pontokat, ezért az adatok ábrázolásánál az alkalmazása célszerűbb. Segítségével pontosabban sikerült feltárni az egyes csoportokban rejlő szélsőséges mintákat, melyek a tovább vizsgáltam. Az LDA és a K-átlag++ eredménye a 32. ábrán látható.

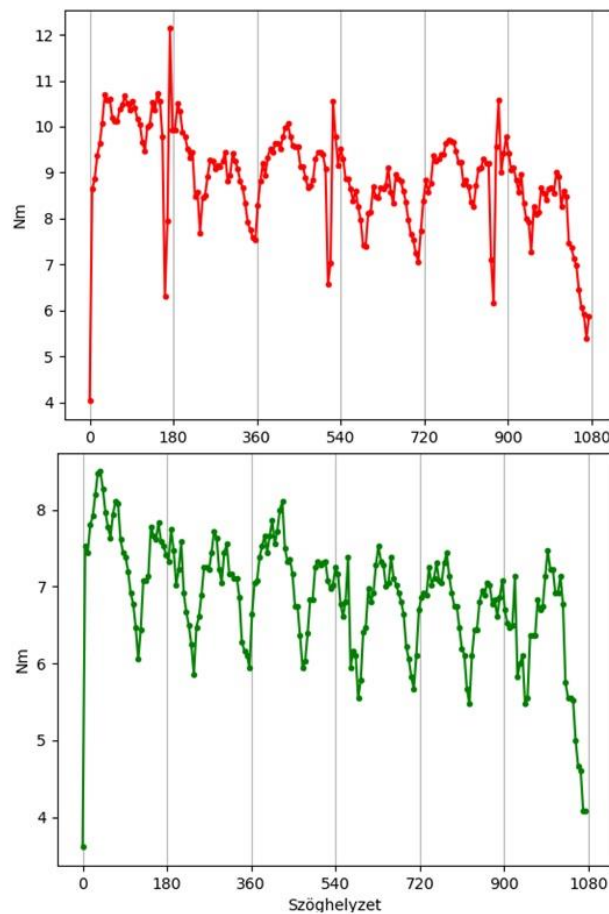


32. ábra: Az 2D LDA és K-átlag++ csoportosítás eredménye

Az LDA eredményének ábrázolása után az egyes csoportok szórása még szembetűnőbb. A csoportok vizsgálata során megállapítottam, hogy az adott motortípuson belüli három különálló

csoport az azonos, de különböző beszállítók által gyártott, úgynevezett helyettesítő alkatrészeknek köszönhető. Ezen alkatrészek paraméterei minimálisan eltérőek lehetnek. A szélsőséges egyedek vizsgálata során ábrázoltam az eredeti, dimenziócsökkentés előtti mérési értékeket, majd ezeket összevettem a csoportok közepéből származó, átlagos mintákkal.

Az ábrázolás során feltűnt, hogy míg az átlagos mérések nyomatékgörbéjének lefutása a kezdeti megindítási nyomatéktól eltekintve egyenletes, periodikusan csökkenő jellegű, addig a szélsőséges egyedekhez tartozó mérések esetén az adott példány átlagos mérési értékeihez képest extrém nagy, impulzusszerű ugrásokat, kilengéseket tartalmaz a mérési görbe. Az átlagos és a feltűnő egyedek méréseinek összehasonlítása a 33. ábrán látható.

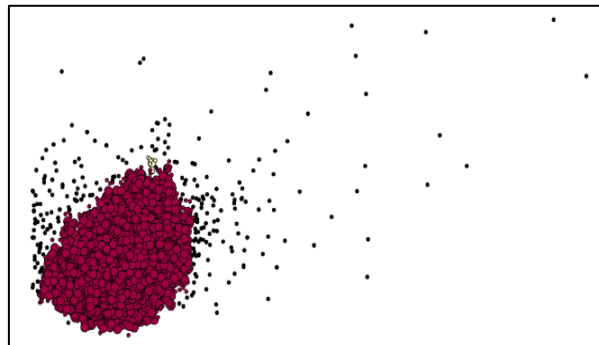


33. ábra: Egy feltűnő (felső, piros) és egy átlagos (alsó, zöld) nyomatékgörbe összehasonlítása

A feltűnő minták esetében megvizsgáltam az utómunka jegyzőkönyveket, valamint a további vizsgálatok során kiderül, hogy a következő munkalépésben végrehajtott főtengely axiális irányú játékanak mérésekor az esetek nagy részében túrésen kívüli eredménnyel zárult.

A rossz főtengely axiális irányú játékkal rendelkező motorokat minden esetben javításra kerülnek, mely során az esetleges eltéréseket, észlelt hibákat jegyzőkönyvbe veszik. A jegyzőkönyvek elemzése során felfedeztem, hogy a jellegzetes nyomatékgörbével és nem megfelelő főtengely axiális irányú játékkal rendelkező motorok egy részében idegen anyagot, forgácsot vagy egyéb szennyeződést találtak a főtengely, illetve a hajtókarcsapágókban vagy csapágyhelyeken. Ezeknek a motoroknak az azonosítására további módszereket dolgoztam ki.

A feltűnő egyedek manuális szelektálása rendkívül időigényes és aprólékos feladat lenne, ezért szükséges a folyamat automatizálása, melyhez a DBSCAN algoritmust használtam. Az ϵ és minPts paraméterek megválasztása tapasztalati úton történt, majd az egyes csoportokat külön-külön elemezve szétválasztottam az átlagos és a jó méréseket. Az eljárás során a DBSCAN azt a tulajdonságát használtam fel, hogy a ritkábban, a sokaságtól távol elhelyezkedő mintákat egységesen zajként osztályozza, így tehát két csoportra tudtam bontani a mérési eredményeket. Az adatokra alkalmazott DBSCAN algoritmus eredményének szemléltetése a 34. ábrán látható.



34. ábra: A 2D adatokra alkalmazott DBSCAN eredménye, az átlagos minták pirossal, a feltűnő minták feketével ábrázolva

A minták osztályozásával és a csoportosított minták felhasználásával olyan mesterséges intelligencia modellek taníthatók, melyek célja az idegen anyagot tartalmazó főtengely- és/vagy hajtókarcsapágók kiszűrése. Az előzőekben bemutatott munkamenet általánosan kiterjeszthető bármilyen beépített mérőrendszerrel rendelkező ipari folyamatra.

11.1.2 Eredmények összefoglalása

A 11.1.1 fejezetben bemutattam a gépi tanulási algoritmusok alkalmazási lehetőségeit anomáliafelismerésre. A kutatáshoz egy motorszereldei automata állomás a főtengely átforgatási nyomatékának méréseit használtam fel és a cél az átforgatási nyomaték mérések és csapágykárral járó motorhibák közötti összefüggések feltárása volt. A kutatásban csak a jónak értékelt mérési eredményeket használtam fel.

Az adatok gépi tanulással történő elemzéséhez az elsődleges feladat az adatok szelektálása és rendezése volt. Ezután az adatokat normalizáltam és a dimenziócsökkentettem, melyek következtében a mesterséges intelligencia algoritmusok tanítási sebessége gyorsítható. A 200 dimenziós bemeneti vektorokat egyenként egységnyi normájúvá skáláztam a scikit-learn és az L2-norma segítségével.

Az adatok dimenziójának csökkentésére a PCA és LDA algoritmusokat használtam. Az adatok 2D ábrázolásánál az LDA jobban teljesít, mivel jobban diszkriminálja az adatokban rejlő egyes csoportokat. A K-átlag++ és a DBSCAN klaszterezési eljárásnál is az LDA által dimenziócsökkentett adatokat használtam fel.

Az adatok szűrése és előfeldolgozása után a mintát a két különböző klaszterező eljárással vizsgálva meghatároztam a mintában fellelhető, az adott motortípuson belüli különböző csoportok számát. Megállapítottam, hogy a csoportjaiktól távol eső, szélsőséges egyedek mérési görbéi eltérést mutatnak az átlagostól. Az feltűnő egyedeket azonosítottam és a nyomatékgörbéket a reklamációs és utómunka jegyzőkönyvekkel, valamint egyéb mérésekkel összehasonlítva megállapítottam, hogy az esetek nagy részében a főtengely axiális irányú játékanak mérése nem lett megfelelő, illetve a motorok utómunkája során a főtengely, vagy a hajtókarcsapágyakban idegen anyagot találtak.

A 11.1.1 fejezetben bemutatott munkamenet és gépi tanulási algoritmusok segítségével az ipari gyártási folyamatokban az anomáliák és nem ismert összefüggések feltárhatók, ezáltal célzott intézkedések hozhatók a minőség javítására és a selejtes termékek számának csökkentésére. A bemutatott munkamenet a megfelelő *domain knowledge* segítségével egyéb mért, szerelési és megmunkálási folyamatra is kiterjeszhető.

II. Tézis

Gépi tanulási algoritmusok segítségével feltártam a főtengely átforgatási nyomatékának mérési eredményei és a csapágykárokkal való motorkiesések közötti összefüggéseket. A PCA és LDA dimenziócsökkentő algoritmusok, valamint a K-átlag és a DBSCAN eljárások segítségével azonosítottam az anomáliát tartalmazó méréseket, melyekhez tartozó motorok további vizsgálata során felismertem, hogy az anomáliát tartalmazó mérésekhez tartozó belső égésű motorok főtengelyének axiális irányú játéka túrésen kívül esik, valamint az utómunka során idegenanyagot találtak a csapágyhelyeken. Az összefüggések feltárására bemutattam az általam kidolgozott, mért szerelési és megmunkálási folyamatokra általánosan kiterjeszthető gépi tanulási algoritmusok alkalmazásának munkamenetét és annak gyakorlati alkalmazhatóságát.

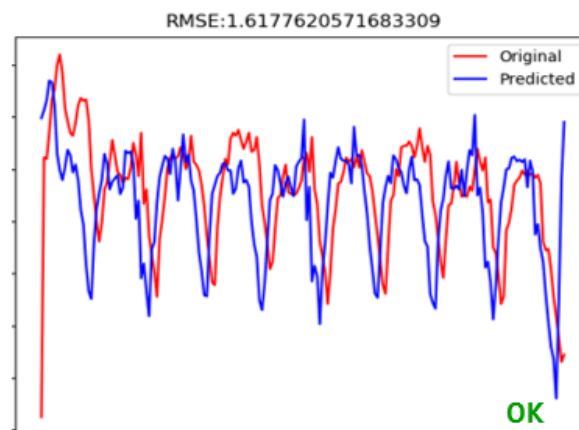
A tézisemben foglalt eredményeket az alábbi publikációmban ismertettem: [S4]

11.1.3 Folyamatfelügyelet LSTM és autoencoder hálózatokkal

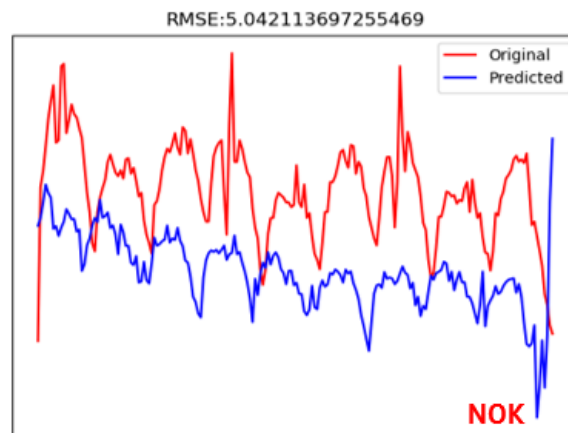
A historikus adatok elemzése mellett kiemelt fontosságú az ipari folyamatok kvázi valós idejű, termelésközeli felügyelete. Munkám célja a szélsőséges minták detektálása mellett olyan modellek létrehozása, amelyek a mérések végrehajtása vagy akár a folyamat közben képes érzékelni az anomáliákat a folyamatokban.

Első lépésként a 11.1.1-ös fejezetben bemutatott főtengely átforgatási nyomaték probléma felismerésére kerestem megoldást. A hibás minták felismerése és kiszűrése lehetővé tette az olyan neurális hálózat alapú modellek tanítását, amelyek az ideális kimeneti értéktől való eltérést felismerik és jelzik az automatizálási berendezés felé. Az előző fejezetben ismertetett osztályozás alapján a teljes minta csupán 2%-át teszik ki a DBSCAN algoritmus által rossznak ítélt mérések, ezért a 5.15 fejezetben bemutatott LSTM [57] idősor előrejelzés segítségével azonosítottam az anomáliát tartalmazó mintákat [S4]. Az adatok a 11.1.1 fejezetben leírt előfeldolgozáson, szelektáláson és normalizáláson esettek át a modell tanítása előtt.

A többrétegű LSTM neurális hálózat tanításakor csak a jó mérési eredményeket használtam fel. Az eredeti mintákból öt időlépéses eltolással új tanítómintákat hoztam létre. Az így kapott $5 * 200$ dimenziós mintákkal feltanított hálózat kellően nagy pontossággal tudja rekonstruálni a bemenetére érkező jel időbeli eltolóját a kimenetén. Mivel a hálózat anomáliát tartalmazó mérésekkel nem találkozott a tanítás során, ezért a bemenetére érkező hibás méréseket csak nagy hibával tudja rekonstruálni a kimenetén. A be- és kimeneti jeleket egymáshoz igazítva szembetűnő az eltérés. A potenciálisan idegen anyagot tartalmazó motorok azonosításához így elegendő kiszámolni a rekonstrukciós hibát, például az átlagos négyzetes hibát. A LSTM rekonstrukció eredménye, illetve a számított átlagos négyzetes hiba (RMSE) a 35., illetve a 36. ábrán látható.



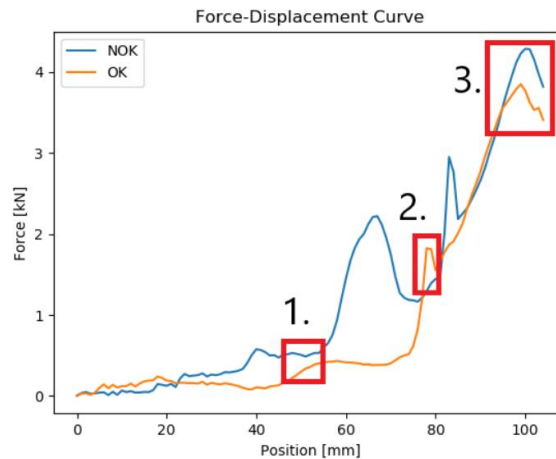
35. ábra: A jó mérési (piros) görbe és rekonstruáltja (kék), RMSE: 1.6178 [S4]



36. ábra: A rossz mérési (piros) görbe és rekonstruáltja (kék), RMSE: 5.0421 [S4]

Az LSTM hálózatok egy lehetséges alkalmazási lehetősége az idősor jellegű adatok előrejelzésére és osztályozására. Azokban az esetekben, amikor viszonylag kevés minta áll rendelkezésre egy adott csoportból, a fent bemutatott módszer segítségével könnyen osztályozhatjuk az egyes méréseket. A potenciálisan gyanús mérések korai észlelése segít az idegen anyag okozta csapágykáros motorkiesések elkerülésében, így minőségi javulás és költségcsökkenés érhető el. A módszer kiterjeszhető bármilyen egyén idősor jellegű adat osztályozására és felügyeletére, mint például csavarozási folyamatok nyomaték és gradiens felügyelete, valamint bepréseli folyamatok út-erő felügyeleti rendszereinek mérési eredményeire. A modell hátránya, hogy a tanítási minták előkészítése időigényes, valamint a modell architektúra is meglehetősen bonyolult.

Egy másik módszer az anomáliák detektálásra a 5.13 fejezetben említett autoencoder [74] hálózatok használata. A módszer hatékonyságát egy főtengely tömítés bepréselésének példáján mutatom be [S5]. Az illesztési és összeszerelési folyamatokat, mint préseles, szegecseles, összeragasztás és a tömítések bepréselése gyakran út-erő, csavarozás esetén szögelfordulást-nyomaték felügyelettel ellenőrzik. Az ilyen típusú felügyeleti rendszerek úgy működnek, hogy az illesztési folyamat során a mérési görbének bizonyos értékeket el kell érnie az út függvényében, vagy épp ellenkezőleg egy határérték alatt kell maradnia a funkcionálisan megbízható csatlakozás elérése érdekében. Az ilyen típusú felügyeletelnél burkológörbét vagy ablak felügyeleti ablakokat használnak. A burkológörbék és a felügyeleti ablakok jól előkészített gyártási tesztek tapasztalati értékei alapján kerülnek beállításra. Ablakokkal felügyelt út-erő mérés a 37. ábrán látható.



37. ábra: Tömítőgyűrű bepréselés út-erő felügyelete ablakokkal [S5]

A felügyeleti ablakok és a hozzájuk tartozó szabályok előre definiálva vannak, a görbéknek mindegyik ablakba be kell lépnie egy előre meghatározott irányból, illetve bizonyos oldalait nem érintheti az ablaknak. Mint a 37. ábrán látható a kék görbe nem sérti az ablakok szabályait, de a folyamat szemmel láthatóan eltér az OK bepréseléstől.

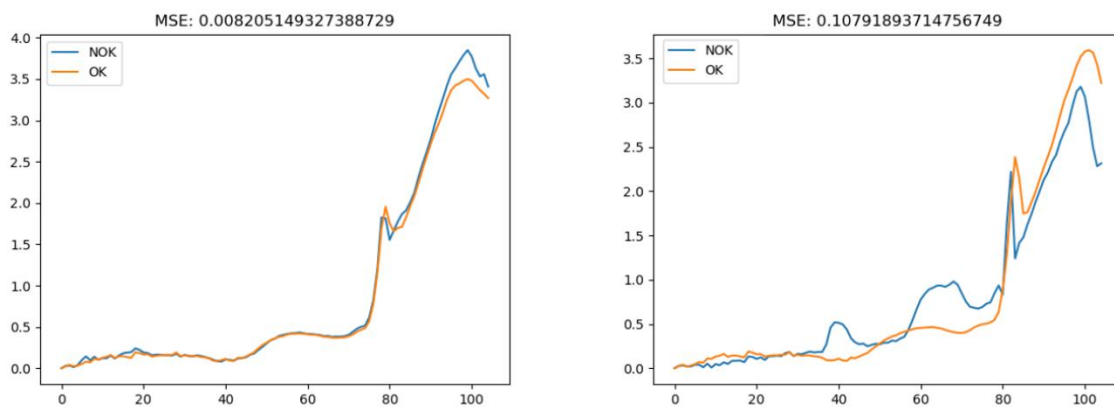
Az ilyen típusú folyamatfelügyelettel az a probléma, hogy nem képes kezelni az ablakokon kívüli váratlan anomáliákat és a burkológörbén belüli zavarokat. A mérések továbbra is OK kategóriába sorolhatók ha a görbék nem sértik a burkológörbét vagy a felügyeleti ablakok előre meghatározott szabályait, még akkor sem, ha a felügyelt szakaszok szemmel láthatóan anomáliákat tartalmaznak. Az autoencoder alkalmazásával a célunk az összes anomália észlelése anélkül, hogy újabb ablakokat kellene meghatározni.

A modell tanítására használt bemeneti adatok mérésenként 105 pontot tartalmaznak, mindegyik pont egy út-erő értékpár. Az erő értéke 0 és 4 kN közötti értékeket vehet fel, ezért a tanítás és a konvergencia gyorsítása érdekében a bemeneti adatokat a 0 és 1 közötti tartomány skáláztam [171]. Összesen 26365 mérést használtam a modell betanításához.

A modell elkészítése során a közönséges autoencoder architektúrát használtam. A hálózat encoder részében, a bemeneti réteg 105 csomópontból áll, az első rejtett réteg 50 csomópontot, a második rejtett réteg pedig 20 neuront tartalmaz. A modell decoder része az encoder rész inverze, rendre 20, 50 és 105 neuronból áll. Az neuronok aktivációjára a ReLU [46][47] aktivációs függvényt használtam (3.11 fejezet), optimalizációs eljárásához az Adam [131]

algoritmust alkalmaztam (3.24 fejezet), a modell veszteségfüggvényét pedig az átlagos négyzetes hiba segítségével számítottam ki a tanítás során. A modellt 32 mintát tartalmazó batchekkel tanítottam, melyek csak OK méréseket tartalmaztak. 15 epoch után a modell megtanulta rekonstruálni a bemeneti adatokat. A modellt elkészítéséhez szintén a TensorFlow és a scikit-learn csomagokat, valamint Python-t használtam.

A modellt, hasonlóan az LSTM alapú megoldáshoz csak OK mérésekkel tanítottam, így a modell a bemeneti adatokat relatív kis hibával rekonstruálja a kimenten. Ha a modell anomáliát tartalmazó mintával találkozik a bemenetén, akkor a rekonstrukciós hiba egy vagy két nagyságrenddel nagyobb lesz, mint az OK mérések esetén. A bemeneti minta és a rekonstrukciója a 38. ábrán látható.



38. ábra: Mérések rekonstrukciója, kék: eredeti görbe, narancs: rekonstruált görbe. A rekonstrukciós hiba viszonylag kicsi az OK esetben (balra), és egy-két nagyságrenddel nagyobb NOK minták esetén (jobbra) [S5]

A méréseket a rekonstrukciós hiba alapján osztályoztam. Ezzel a módszerrel sikerült a felügyeletet kiterjeszteni a teljes mérési görbére, így az képes észlelni a felügyeleti ablakokon kívüli anomáliákat. Az eredmények osztályozását a rekonstrukciós hiba nagysága alapján, akár egyszerű gépi tanulási algoritmussal is végre lehet hajtani.

11.1.4 Eredmények összefoglalása

A 11.1.3 fejezetben ismertettem a klasszikus folyamatfelügyeleti módszerek, mint a burkológörbe és a felügyeleti ablakok hiányosságát és javaslatot tettem az LSTM és az autoencoder hálózatok használatára a folyamatfelügyelet javítására. Az LSTM hálózatok alkalmazását a 11.1.1 fejezetben bemutatott főtengely átforgatási nyomaték mérés felügyeletére, az autoencoder hálózatokat pedig egy főtengely tömítés bepréseli folyamat felügyeletére gyakorlati példáján keresztül vizsgáltam.

A bemeneti adatok megfelelő előfeldolgozása és skálázása után, a feltárt jelenség ismeretében megtanítottam egy LSTM neurális hálózatot a jó mérési görbék pontos rekonstruálására. A hálózat a rossz mérési görbéket viszont nagy hibával rekonstruálja, így a rekonstrukciós hiba kiszámításának a segítségével a hibás mérések valós időben, automatizált módon detektálhatóvá váltak. A feltárt összefüggések és módszerek segítségével tehát az idegen anyag jelenléte a belső égésű motorok főtengely- és hajtókarcsapágyaiban előre jelezhető, így jelentős minőségjavulás és költségcsökkentés érhető el.

Az autoencoder hálózatok esetén az univerzális approximátor tételt kihasználva, az adatok előfeldolgozása után megtanítottam a modellnek a jó mérési görbék rekonstruálását. Az autoencoder hálózatok tanítására csak jó minőségű mintákat használtam, így azok a bemenetükre kapcsolt jó mérési eredmények relatív kicsi hibával képesek rekonstruálni a kimenetükön. A rekonstrukciós hiba mérésére az eredeti és a kimeneti görbék közötti átlagos négyzetes hiba mértékét használtam. Hibás mérések esetén a rekonstrukciós hiba nagyságrendekkel nagyobb lesz, így az anomáliát tartalmazó görbék automatikusan felismerhetők.

A bemutatott LSTM és autoencoder alapú módszerek egyszerű, de hatékony folyamatfelügyeletet tesznek lehetővé. A modellek képesek észlelni az előre nem látható hibákat és anomáliákat az olyan folyamatokban, ahol a mérési görbék jellege jól ismert. A bemenet és a kimenet közötti rekonstrukciós hiba figyelésével az anomáliát tartalmazó mérések és görbék detektálhatók. Ezekkel az eljárásokkal a felügyeleti ablakok és burkológörbék kiegészíthetők vagy akár teljes mértékben el is hagyhatók, így a teljes mérési görbékre kiterjedő folyamatfelügyelet érhető el. Az autoencoder alapú modell kisebb rekonstrukciós hibával képes dolgozni, mint az LSTM alapú verzió, azaz a bemenete és kimenete közötti átlagos négyzetes hiba mértéke kisebb, az így kisebb anomáliák észlelésére is alkalmas. A módszerek

kiterjeszhető bármilyen idősor jellegű adatot előállító ipari folyamatra, mint a bepréselések, csavarozások és egyéb illesztési eljárások.

III. Tézis

Kidolgoztam és bemutattam az LSTM és autoencoder mesterséges neurális hálózatok ipari alkalmazásának munkamentetét ipari környezetben újszerű folyamatfelügyeltre és gyakorlati példákon bemutattam azt. Megmutattam, hogy a modellek kimeneti rekonstrukciós hibájának mérésével a teljes mérési görbe felügyelhetővé válik, így a felügyeleti ablakok és burkológörbék az ismertett módszerekkel kiegészíthetők vagy azok akár el is hagyhatók. A bemutatott módszerek általános kiterjeszhető bármilyen idősor jellegű mérési adatra és a hozzá tartozó folyamat felügyeletére.

A tézisemhez tartozó publikációim a következők: [S4] [S5]

11.1.5 Nemdeterminisztikus zavarok eliminálása ipari képfeldolgozó alkalmazásokban

Napjainkban a magas minőségi elvárások szükségessé teszik az in-line, automatikus mérő- és felügyeleti rendszerek alkalmazását. Az iparban az előző fejezetben ismertetett mérési adatfeldolgozás mellett egyre nagyobb szerepe van a képfeldolgozási alkalmazásoknak is, melyek különböző jelenlét- és pozícióellenőrzések mellett mérési feladatok is elláthatnak. Tipikus feladatok közé tartozik a tárgyfelismerés vagy az olyan fejlettebb funkciók alkalmazása, mint a távolság- és/vagy szög mérés, helyzet- vagy színérzékelés, felületi szkennelés és mérés. Ezek a módszerek könnyen használhatók, ha a vizsgált képek nagyon hasonlítanak a rendszer beállításához és finomhangolásához használt képekhez. De ha a képek szokatlan részeket vagy külső, nem determinisztikus zavarokat tartalmaznak, ezen felügyeleti rendszerek hamis pozitív arány növekedhet. Az említett problémák egy lehetséges megoldása a konvolúciós neurális hálózatok alkalmazása [S6].

A modell fejlesztéséhez az AUDI HUNGARIA Zrt. egy motorszereldejéből származó adatokat használtam fel. A képfeldolgozás során egy automata állomás ellenőrzi a szerelősoron lévő belső égésű motor elektromos csatlakozóit. A jelenlegi képfeldolgozó alkalmazás ellenőrzi, hogy a vizsgált csatlakozó jelen van, teljesen csatlakoztatva és reteszelve van-e vagy sem. Az ellenőrzött elektromos csatlakozó a 39. ábrán látható, az alkatrészszám szándékosan került kitakarásra.



39. ábra: Az ellenőrzött elektromos csatlakozó

A jelenlét és a csatlakoztatás ellenőrzésére az állomás kiszámítja a műanyag címke felső szélének és a csatlakozó zárjának távolságát (40. ábra). Amint látható, a csatlakozó teljesen

rögzítve és reteszelve van, de bizonyos esetekben a képfeldolgozó rendszer nem tudja észlelni ezeket az éleket.



40. ábra: A műanyag címke felső széle és a csatlakozózár közötti távolság mérése

A 41. ábrán a tipikus hamis pozitív hibák láthatók. Az emberi szem számára egyértelmű, hogy a csatlakozók rögzítve és reteszelve vannak, de a képfeldolgozó rendszer algoritmusai nem tudja ráhelyezni a mérőablakot a képekre, így az ellenőrzés eredménye NOK (hibás) lesz.



41. ábra: Tipikus hamis pozitív esetek

A problémát az okozza, hogy egyes kábelek nem megfelelően vannak rögzítve, ezért azok eltakarják a kiértékeléshez használt éleket. Az ellenőrzött területen lévő árnyékok, a megvilágítás vagy az anyag színének változása ugyanazt a hatást okozhatja, mint a belógó kábelek. A konvolúciós hálózatok alkalmazásának célja az, hogy csökkentsük ezeket a hibákat

a jól csatlakoztatott csatlakozó magas szintű reprezentációinak megtanításával, ezáltal a nemdeterminisztikus, a képfeldolgozást hátráltató hibákat elimináljuk.

A modell elkészítéséhez és validáláshoz Python-t, valamint a TensorFlow keretrendszert és scikit-learn csomagokat alkalmaztam. A modell tanításához használt adatkészlet 22755 képet tartalmaz. Az OK – NOK képek eredeti arányát az 6. táblázat mutatja. Ezek a képek harminc nap gyártásából származnak, amiket az automata ellenőrző berendezés gyűjtött.

6. táblázat: Az OK - NOK képek, illetve összesített darabszámuk

<i>OK</i>	<i>NOK</i>	<i>Összesen</i>
21844	911	22755

A konvolúciós neurális hálózatok tanítása nagy számítási kapacitást igényel, valamint a tanítási idő sem elhanyagolható, ezért célszerű a bemeneti adatokat a lehető legkisebb méretűre alakítani. A sorozatgyártásban sok környezeti változó rögzíthető, ezt a tulajdonságot sikerült kihasználni a munka során. Az ellenőrzött tételek készlete két-három típusra korlátozódik, valamint a motorok pozíciója a gyártósoron, így a csatlakozó helyzete is nagyon korlátozott intervallumban változik csak a képen belül. Ez lehetővé tette a vizsgált terület (*region of interest, ROI*), ezáltal a kép méretének csökkentését. Az előfeldolgozás során, a modell bemenetén, a nulladik lépésben a dimenzió méretének csökkentése és az emberi ellenőrzés felgyorsítása érdekében a képeket levágtuk és elforgattuk. Az eredeti és az előfeldolgozott kép a 42. ábrán, paramétereik a 7. táblázatban láthatók.



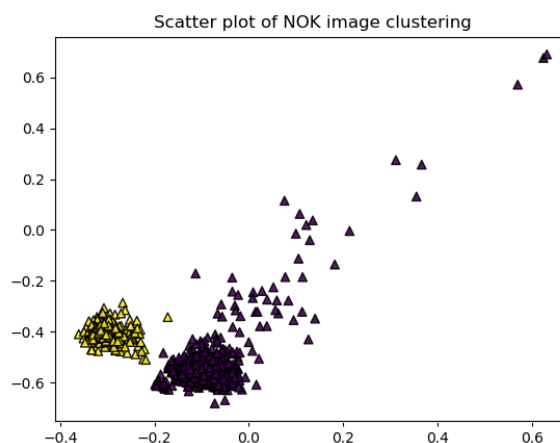
42. ábra: Az eredeti (bal) és az előfeldolgozott (jobb) kép

7. táblázat: Az eredeti és az előfeldolgozott képek paramétereit, a Dimenzió a neurális hálózat bemeneti rétegének csomópontjait jelenti (Szélesség x Magasság)

<i>Paraméter</i>	<i>Eredeti</i>	<i>Előfeldolgozott</i>
Szélesség	1200 px	510 px
Magasság	1000 px	540 px
Méret	70 kB	30 kB
Dimenzió	1 200 000	275 400

A hamis pozitív és hamis negatív eredményeket tartalmazó képek valódi számának meghatározásához felül kellett vizsgálnom az egyes kategóriákat. A folyamat felgyorsítása érdekében az már ismertetett a K-átlag++ [26] algoritmust használtam. Az algoritmus a bemeneti adatokat úgy csoportosítja, hogy a mintákat n darab, egyenlő szórással rendelkező csoportba sorolja, a már ismertetett (19) egyenlet szerint minimalizálva az inerciát.

Ebben az esetben a távolság metrikának a minták és a hozzájuk rendelt csoport centroidjának L2-távolságát használtam. A centroidok helyzetét az algoritmus a minták átlagos pixelértékeiből számította ki. Feltételeztem, hogy egy egyértelmű zavaró tényező, például a csatlakozót eltakaró kábel jelenléte esetén a minta átlagos pixelértéke nagymértékben eltér a zavarmentes mintákétól. A nem egyértelmű esetekben, ahol a képek csak kisebb zavarokat tartalmaztak, mint például részlegesen rögzített csatlakozó retesz, a felülvizsgálatot manuálisan végeztem el. A NOK képek klaszterezésének eredménye az 43. ábrán látható. A diagram tengelyei dimenzió nélküliek, a képek térbeli 2D reprezentációja a pixelértékekből főkomponens-analízissel (PCA) jött létre [19].



43. ábra: A NOK képek 2D szórásdiagramja, a sárgával jelölt minták nem tartalmaz zavart, a lilával jelölt minták igen. Az alacsonyabb sűrűségű régiókban elhelyezkedő képeken egyáltalán nem volt csatlakozó

A csoportosítás után ismét manuálisan ellenőriztem a zavaró tényezőket tartalmazó kategóriát. Az előfeldolgozás eredményeit a 8. táblázatban láthatjuk.

8. táblázat: Képek aránya az előfeldolgozás után

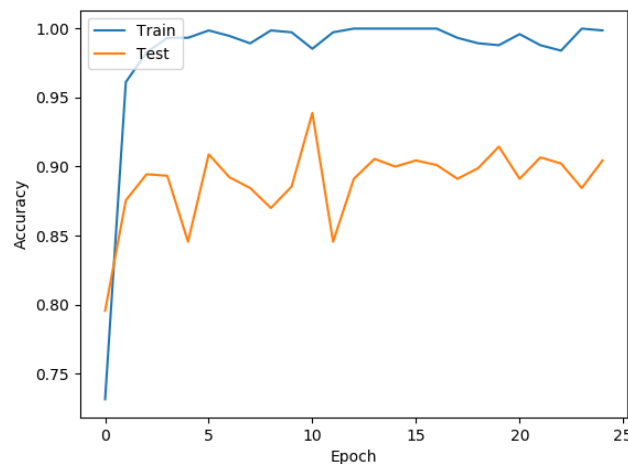
<i>OK</i>	<i>NOK</i>	<i>Nincs csatlakozó</i>	<i>Összesen</i>
22582	81	92	22755

Mint látható, korábban 738 kép volt NOK kategóriába sorolva, pedig valóban rendben voltak. 92 esetben egyáltalán nem volt csatlakozó a képeken, ezt a motorszállító paletta rossz pozíciója okozta. Ha összeadjuk a NOK és a hiányzó csatlakozó kategóriákat, akkor 173 valódi NOK képünk lesz, ami az összes kép 0,76 százaléka. Elméletileg ez azt jelenti, hogy ha egy hálózatot betanítunk ezzel az adatkészlettel és elérjük a 98 százalékos pontosságot, a hálózat minden NOK képet OK-nak minősíthetne és még mindig 98 százalékos a pontossága lenne. A probléma megelőzése érdekében több NOK mintára van szükség a képzéshez, erre az adatkiegészítés (*image augmentation*) eljárást alkalmaztam [172].

A TensorFlow beépített *ImageDataGenerator* osztályával, a CNN modell tanítása során valós időben lehet generálni új képeket, a megadott forrásképek véletlenszerűen függőlegesen és vízszintesen eltolásával, forgatásával, tükrözésével, átméretezésével, kicsinyítésével vagy nagyításával valamint a képek világosításával vagy sötétítésével. A kép generátor forrásaként 173 NOK és 173 véletlenszerűen kiválasztott OK képet használtam. A képgenerátor minden egyes tanítási epoch során, minden lépésében 32 darabos kötegekben generálta a modell bemeneti képeit. Ebben az esetben 25 epoch és 25 lépését használtam a tanításhoz, így összesen 20 000 képet használtam fel. Ezzel a módszerrel csökkenthető az OK-NOK arány torzítása és növelhető a tanítási sebesség. A képek 80%-a a hálózat betanítására használtam, míg a maradék 20%-ot validációs adatkészletnek.

Az architektúra hasonló a LeNet-4 modell architektúrához, amely egy viszonylag egyszerű, de pontos modell [173]. A modell paramétereit a bemeneti képparaméterekhez illesztettem. Az architektúra a bemeneti réteg után két konvolúciós réteget tartalmaz ReLU aktivációs függvényvel, a harmadik réteg egy MaxPooling réteg, amelyet egy Flatten réteg követ, amely a hálózat konvolúciós részét köti össze a hálózat kimenetén lévő teljesen összekötött neurális hálózattal. A hálózat teljesen összekötött része három rétegből áll, a kimeneti réteg pedig egy

csomópontot tartalmaz softmax aktivációs függvénnyel. A jobb általánosító képesség érdekében 20%-os neuron kiejtést (*dropout*) [88] alkalmaztam a teljesen összekapcsolt rétegek között. Több dropout értéket teszteltem, de a szakirodalomban széles körben alkalmazott 20%-s érték teljesített a legjobban. Az optimalizálás eljárás során az Adam [131] algoritmust használtam, a veszteségfüggvény az átlagos négyzetes hiba, az optimalizálás metrikája pedig a modell pontossága volt. Az eredmények érvényesítéséhez egy másik képgenerátort konfiguráltam, 25 epoch-nyi tanítás után a modell pontossága 90% körül alakult (44. ábra).



44. ábra: A CNN modell pontosságának alakulása a tanítás során

A betanítási fázist követően az eredeti képfeldolgozó algoritmus által korábban NOK-nak minősített 911 képet újra osztályoztam a CNN modellel. A téves besorolási arány javulása a 9. táblázatban látható.

9. táblázat: A téves osztályozás arányának javulása

<i>Összes minta</i>	<i>Valódi NOK</i>	<i>Téves osztályozás</i>	
		<i>Eredeti rendszer</i>	<i>CNN modell</i>
911	173	738 (81,01%)	23 (2,52%)

11.1.6 Eredmények összefoglalása

A 11.1.5 fejezetben bemutattam az ipari képfeldolgozó alkalmazások nemdeterminisztikus zavarainak elhárítását konvolúciós neurális hálózatok segítségével. A modell fejlesztéséhez az AUDI HUNGARIA Zrt. egy motorszereldéjéből származó adatokat használtam fel. A képfeldolgozás során egy automata állomás ellenőrzi a szerelősoron lévő belső égésű motor elektromos csatlakozóit.

A problémát az olyan nemdeterminisztikus zavarok okozzák, mint az ellenőrzött területen lévő árnyékok, a megvilágítás vagy az anyag színének változása és a belógó kábelek. A konvolúciós hálózatok alkalmazásának célja az, hogy csökkentsük ezeket a hibákat a jól csatlakoztatott csatlakozó magas szintű reprezentációinak megtanításával, ezáltal a nemdeterminisztikus, a képfeldolgozást hátráltató hibákat elimináljuk.

Az előfeldolgozás során a bemeneti képek dimenziójának csökkentése és az emberi ellenőrzés felgyorsítása érdekében a képeket megvágtam és elforgattam. Az OK és NOK képek valódi számának meghatározásához, beleértve a hamis pozitív (hamis NOK) és hamis negatív eredményeket (hamis OK) is, a kategóriák felülvizsgálatát kellett elvégezni. A folyamat felgyorsítása érdekében a K-means++ alkalmazást használtam. A klaszterezés után ismét manuálisan ellenőriztem a zavaró tényezővel rendelkező csoportot.

A NOK képek valós számának meghatározása után a TensorFlow mintageneráló osztályát használtam a tanítás során az OK-NOK minták egyensúlyba hozásához. 20000 mintát generáltam a tanítás során, melynek 80%-át a tanításra, a maradék 20%-át pedig a validációra használtam.

A tanítási fázist követően az eredeti gépi látási algoritmus által korábban NOK-nak minősített 911 képet osztályoztam a modellel, ezek közül 173 darab kép volt valóban NOK. Az eredeti képfeldolgozó alkalmazás 738 képet osztályozott hibásan NOK-nak, a konvolúciós neurális hálózat alapú modell viszont csak 23 darab képet osztályozott rosszul, amivel 97,46%-os javulást értem el az eredeti alkalmasához képest.

A modell jóval egyszerűbb, mint a legmodernebb (VGG [174], CapsNet [175]) modellek, de speciális feladatokra ígéretes eredményekre képes. Ennek egyik oka, hogy az általános mesterséges intelligenciával szemben, ahol a cél a lehető legtöbb objektum megkülönböztetése és osztályozása, az ipari képfeldolgozás célja és hatóköre jól meghatározott és kevés esetre korlátozódik.

A gépi tanulás alkalmazásával a hagyományos ipari képfeldolgozó rendszerek teljesítménye hatékonyan növelhető. Ha a modellt az eredeti rendszer kiterjesztéseként használjuk, a hamis pozitív hibák arányára, így az emberi beavatkozás mértéke csökkenthető és a termelékenység nő. A fent bemutatott eljárások és modell kiterjeszhető egyéb ipari képfeldolgozási feladatra is.

IV. Tézis

Kidolgoztam és bemutattam a klasszikus képfeldolgozási alkalmazásokat zavaró nemdeterminisztikus hibák eliminálásának ipari környezetben újszerű módszerét konvolúciós neurális hálózatok alkalmazásával a hamis pozitív hibák számának csökkentésére. A módszer segítségével azokban az esetekben is érzékelhető az alkatrészek és komponensek megfelelő rögzítése vagy jelenléte, melyekben a klasszikus, pixelek vagy élek közötti távolságmérésen alapuló módszerek segítségével a képek nem megbízhatóan vagy egyáltalán nem alkalmazhatóak a megvilágítás változásai vagy a kiértékelendő területet részben kitakaró zavarok miatt. A gyakorlati alkalmazással 97,46%-kal sikerült csökkentenem a hamis pozitív hibák számát. A bemutatott munkamenet és modell kiterjeszhető más ipari képfeldolgozó alkalmazásokra is.

A tézisemben foglalt eredményeket az alábbi publikációmban mutattam be: [S6]

11.1.7 Karbantartási hatékonyság növelése természetes nyelvfeldolgozással

A természetes nyelvi feldolgozás (NLP) a nyelvi tudományok és a mesterséges intelligencia találkozása. Az NLP célja, hogy a természetes nyelvet emberileg értelmezhető formából számítógéppel értelmezhető formává alakítsa. Az NLP módszerek lehetővé teszik a számítógépek számára a természetes, emberi nyelv megértését. Az NLP-t gépi fordításra, beszédfelismerésre, nyelv- és szövegenerálásra, kérdések megválaszolására és automatizált chatbotokban használják. Az intelligens asszisztensek és chatbot-ok terjedésével, mint a ChatGPT [176] egyre nagyobb figyelemét kapnak a nyelvi modellek. Az NLP népszerű az orvostudományban, az egészségügyben, a közösségi médiában, a reklámparban, az emberi erőforrások területén és más szolgáltatásokban is, de az ipari felhasználásban rejlő lehetőségek egyelőre kihasználatlanok.

A mérési- és folyamatadatok, valamint a képfeldolgozó rendszerek által előállított adatok mellett az ipari folyamatok emberi résztvevői is nagymennyiségű szöveges adatot állítanak elő karbantartási feljegyzések, jegyzőkönyvek és egyéb szöveges dokumentumok, beszámolók formájában. A legtöbb termelő vállalat nyilvántartja a karbantartási tevékenységeket, az alkatrészcsereket és a gépek felújítását. Ennek a tevékenységnek a célja, hogy statisztikát készítsen a gépek állásidejéből, elemezze a karbantartási időket és időközöket, a hibaokokat, a karbantartási költségeket ezáltal javítsa a karbantartási folyamatot és nyomon kövesse a minőséggel kapcsolatos beavatkozásokat. A karbantartási feljegyzéseket, hibaleírásokat különböző emberek állítják elő, mely következtében az azonos hibák hibaleírása rendkívül szélesen skálán mozog. Még alacsony számú berendezés mellett is, az egyedi megfogalmazások és az idővel növekvő rekordmennyiség megnehezíti a hibák közötti minták és összefüggések észlelését. Ezen problémák megoldásaként hoztam létre egy NLP modellt a karbantartási feljegyzések elemzésére [S7].

Az AUDI HUNGARIA Zrt. egy mechanikus megmunkálás gyártósorának karbantartási feljegyzéseit vizsgáltam meg a 2019-es időszakban. A karbantartási napló 343 rekordot tartalmaz 79 gépről. A rekordok klaszterezésére és a hasonlóságok kimutatására felügyelet nélküli NLP klaszterezést alkalmaztam. A hibaleírások címkézettek (hiba jellege és hibaosztálya), az adatkészlet jó kiindulópont lenne az osztályozási modellek betanításához, de a minták száma túl kevés, a modell nem általánosítana jól.

Pythont használtam a szöveg előfeldolgozásához és az NLP modell elkészítéséhez. A modell felépítéséhez a scikit-learn [135], NLTK[102], spaCy [177] könyvtárakat használtam [S7].

Első lépésként a természetes nyelvi szöveget számítógéppel értelmezhetővé alakításához, a szöveg előfeldolgozását végeztem el. Az előfeldolgozás során a szöveges dokumentum kisebb részekre vagy mondatokra szegmentálható, a releváns információt nem tartalmazó kifejezések, például írásjeleket és stopszavakat el kell távolítani. A reguláris kifejezésekkel (*regex*) eltávolítottam az írásjeleket, majd az olyan stopszavakat, mint a „javítva”, „csere”, „csere után”, stb. Ezek a kifejezések további információk (pl.: „Munkadarab pozicionáló tű törése. Csere után OK.”) azt jelentik, hogy a javítás megtörtént, de a hiba jellegéről nem ad további információt. Az előfeldolgozási lépések részeként a szöveget kisbetűsre alakítottuk, mivel ez csökkentheti a tokenek számát. Így a tokenizálás során ugyanazok a szavakat vagy n-grammokat, amelyek kis- vagy nagybetűvel kezdődnek vagy teljes mértékben abból állnak („gép”, „GÉP”) , egyként kifejezésként kezeljük. Az előfeldolgozási további, a tokenizálást megelőző lépései is az információ kinyerésére törekednek, ilyenek a lemmatizálás, a szótövesítés (*stemming*) és a beszédrészek (*Part of Speech, POS*) címkézése. A lemmatizálás az inflexiók végződéseket távolítja el, és a szó alapszótári alakját adja vissza. A szótövesítés a ragozott vagy származtatott szavak szótöveit adja vissza. A POS-címkézés minden szóhoz meghatározza a beszédrész-címkéket mint ige, főnév, melléknév. A fent említett technikákat szóbeágyazásnak is nevezik, amely olyan módszerek gyűjtőneve, ahol a szókincsből származó szavakat vagy kifejezéseket valós számok vektoraira képezzük le.

A tokenizálás során az ember által olvasható karaktereket, szavakat vagy mondatokat számértékké kell konvertálni. Az eljárás során minden mondathoz, szóhoz vagy n-gramhoz számértéket társítunk, és hozzáadjuk a listához vagy a szótárhoz (*bag-of-words, BOW*). Az n-gram egy adott szöveg n elemének összefüggő sorozata, amely szavakból, szóalapú karakterekből és karakterekből is létrehozható. A szöveg számokká alakításához, súlyozó vagy vektorizáló függvények szükségesek, mint például a kifejezés gyakorisága (*term frequency, TF*), az inverz dokumentum gyakorisága (*inverse document frequency, IDF*) vagy a e két módszer kombinációja, a gyakran használt kifejezés gyakoriság-inverz dokumentumgyakoriság (*TF-IDF*) [102]. A tokenizáláshoz n-grammokat használtunk: minden szót unigramokra, bigramokra és trigrammokra bontottam, majd minden dokumentumban megszámláltuk a kifejezések gyakoriságát (TF). Hosszabb n-grammok, például trigrammok vagy négygrammok akkor ajánlottak, ha olyan nyelvekkel foglalkozunk, amelyek digráfokat (kettős betűket)

tartalmazzak mint a magyar. A TF kifejezés gyakorisága, amelyet $TF(t, d)$ -ként is jelölünk, megszámlolja, hogy a t kifejezés hányszor fordul elő minden d dokumentumban (22).

$$TF(t, d) = \frac{f_{t,d}}{\sum_{t \in d} f_{t,d}} \quad (22)$$

A tokenizálás végeztével a tokenek megszámlálásával egy szótárat hozunk létre, egy ritka mátrixot, amely tartalmazza az n -gramos kifejezések gyakoriságát. A dokumentumok ilyen numerikus ábrázolása lehetővé teszi a gépi tanulási algoritmusokba való betáplálásukat.

A karbantartási feljegyzések klaszterezéséhez a DBSCAN [34] algoritmust használtam. A DBSCAN távolságmérés alapján csoportosítja az egymáshoz közel eső pontokat. A ritka vektorok közötti távolság mérésére koszinusz hasonlóságot (23) használtunk. A koszinusz hasonlóságot két normalizált vektor közötti szög mérésére alkalmaztam, így kiszámítjuk a vektorok hasonlóságát, ez azt mutatja meg, hogy mennyire hasonló két vektor orientációja, függetlenül nagyságuktól.

$$S_c(A, B) := \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (23)$$

ahol A_i és B_i az i -edik komponense az A és B vektoroknak.

A csoportosítás eredményeként a 343 karbantartási bejegyzésből 169 rekordot került külön csoportokba [S7]. Ez azt jelenti, hogy a hibák 49,27%-a visszatérő. A többi 174 rekord egyedi bejegyzés. Az ismétlődő hibák elemzésével a karbantartási stratégiai feladatok javíthatók, vagy a súlyos hiba elkerülhető. A fenti példa azt mutatja, hogy a legegyszerűbb NLP elemzés növelheti a karbantartási hatékonyságot.

A manuálisan rögzített karbantartási feljegyzéseken kívül vannak automatikusan rögzített hibaüzenetek is. Egy közepesen összetett berendezés több ezer hibaüzenetet képes kijelezni. Egyes esetekben azonban a gépkezelő nem tudja, hol van a hiba pontos helye, vagy hogy ki a megfelelő, szakértő személy az értesítésre. Ezeket a problémákat az nevesített entitások felismeréssel (*Named Entity Recognition, NER*) lehet megoldani. A NER azonosítja az entitásokat egy dokumentumban, szöveges korpuszban vagy hibaüzenetben. Ezek az entitások

lehetnek személyek, helyek vagy akár gépalkatrészek vagy egyéb komponensek. A hiba forrásának ismeretében a gép tervrajzán megjeleníthető a hiba helye, automatikusan megjeleníthető a komponens dokumentációja és automatikusan értesíthető a megfelelő személy.

A NER tanításához 4533 automatikusan rögzített gépi hibaüzenetet elemeztem. Az üzenetek ipari mosóberendezésekhez, eszterga-, köszörű- és marógépekhez tartoznak. Az elemzés felgyorsítása érdekében a manuális karbantartási feljegyzések esetén alkalmazott technikákat használtam.

Az előfeldolgozás lépései után a hibaüzeneteket 2891 egyedi szóra bontottam, majd csoportosítottam őket. Mivel a hibaüzenetek részben ismétlődő részeket tartalmaznak, ami azt jelenti, hogy a komponensek nevei ismétlődnek, a TF-IDF vektorizálás nem volt hatékony, mert a TF-IDF súlyozás kiszűri a gyakori kifejezéseket, ezért itt is TF vektorizálást alkalmaztam. Végül az entitások klaszterezéséhez újfent a DBSCAN-t használtam.

A csoportosítás után 2192 entitás sikerült automatikusan klaszterezni. A maradék 699 kifejezést kézzel kellett kategorizálni és felcímkézni. 70 kategóriát azonosítottam, ami mindegyik egy külön entitáshoz tartozik. A 10. táblázatban látható modell teljesítménye.

10. táblázat: A modell validálás utáni teljesítménye

<i>Precision</i>	<i>Recall</i>	<i>F-Score</i>
94,67%	92,99%	93,82%

A precision, a recall és az F-score általánosan használt teljesítménymutatók a klasszifikációs modellek hatékonyságának értékelésére.

- A precision azt méri, hogy a modell pontosan hány pozitív példányt osztályozott helyesen. Más szóval, a valós pozitív példányok arányát számolja ki a pozitívra osztályozott példányok és a hamis pozitív példányok összegéhez képest. Egy magas precision érték azt jelzi, hogy a modell pontosan osztályozza a pozitív példányokat [178].
- A recall azt méri, hogy a model mennyire képes felismerni az összes valós pozitív példányt az adatok között. A valós pozitív példányok arányát számolja ki a valós pozitív példányok és a hamis negatív példányok összegéhez képest. Egy magas

recall érték azt jelzi, hogy a modell nagy arányban ismeri fel a pozitív példányokat [179].

- Az F-score a precision és a recall harmonikus közepéből számítandó és egyetlen pontszámmal mutatja meg mindkét metrika egyensúlyát [180].

$$F_{Score} = 2 \frac{(Precision * Recall)}{(Precision + Recall)} \quad (25)$$

Az F-score skálája 0 és 1 között mozog, az 1 az ideális érték [179][180].

Általában egy jó klasszifikációs modellnek magas precision és recall értékeket kell mutatnia. Azonban gyakran felmerül az egyensúlyozás problémája a két metrika között, mivel az egyik javítása általában a másik rovására megy. Az F-score lehetővé teszi ezen két metrika egyensúlyának ellenőrzését, és átfogóbb képet ad a modell teljesítményéről.

11.1.8 Eredmények összefoglalása

Az AUDI HUNGARIA Zrt. egy mechanikus megmunkálás gyártósor karbantartási feljegyzéseit vizsgáltam meg a 2019-es időszakban, majd az automatikusan rögzített gépi hibaüzenetet elemeztem. A karbantartási napló 343 rekordot tartalmaz 79 gépről, az automatikusan rögzített hibaüzenetek pedig további 4533 bejegyzést tartalmaznak.

Első lépésként a természetes nyelvi szöveget számítógéppel értelmezhetővé alakításhoz, a szöveg előfeldolgozását végeztem el. Az előfeldolgozás során a szöveget kisbetűssé konvertáltam, eltávolítottam az írásjeleket, a stop szavakat, majd a lemmatizáció és a szótövesítés után n-gramokra bontottam a szövegkorpuszt.

A karbantartási feljegyzések klaszterezéséhez a DBSCAN algoritmust használtam. A DBSCAN távolságmérés alapján csoportosítja az egymáshoz közel eső pontokat. A ritka vektorok közötti távolság mérésére koszinusz hasonlóságot használtunk. A csoportosítás eredményeként a 343 karbantartási bejegyzésből 169 rekordot került külön csoportokba, mely alapján a hibák 49,27%-a visszatérő.

A karbantartási naplók és hibaüzenetek NLP elemzése ígéretes eredményt mutat. A bemutatott módszerek jól alkalmazhatók a karbantartási folyamatok és a karbantartási stratégiák felgyorsítására és javítására. A korábban említett technikák gyors eredményeket hoznak, különösen akkor, ha nagyméretű, hosszú időre visszanyúló karbantartási naplók állnak rendelkezésre.

Az automatikusan rögzített hibaüzenetek elemzéséhez nevesített entitás felismerés tanításához alkalmaztam, mely tanításához 4533 hibaüzenetet használtam. Az előfeldolgozás után a hibaüzeneteket 2891 egyedi szóra bontottam, majd csoportosítottam őket. Mivel a hibaüzenetek részben ismétlődő részeket tartalmaznak, ami azt jelenti, hogy a komponensek nevei ismétlődnek, a TF-IDF vektorizálás nem volt hatékony, mert a TF-IDF súlyozás kiszűri a gyakori kifejezéseket, ezért itt is TF vektorizálást alkalmaztam. A csoportosítás után 2192 entitás sikerült automatikusan klaszterezni. A maradék 699 kifejezést kézzel kellett kategorizálni és felcímkézni. 70 kategóriát azonosítottam, ami mindegyik egy külön entitáshoz tartozik.

A modell a 9. táblázatban látható 94,67% Precision, 92,99% Recall és 93,82%-os F-Score értékkel rendelkezik. A gépelemek automatikus felismerésének köszönhetően a megfelelő karbantartók automatikusan értesíthetők, így a karbantartási hatékonyság növelhető.

A karbantartási és ipari folyamatok terminológiája eltér a hétköznapi nyelvtől, de megfelelő előfeldolgozással, a releváns stopszavak azonosításával és a megfelelő súlyozási függvényekkel a karbantartási naplók és hibaüzenetek feldolgozása ugyanolyan egyszerű, mint a hétköznapi szövegek feldolgozása.

A gépi tanulási módszerek, valamint a természetes nyelvi feldolgozás nagy lehetőségeket rejt magában az ipari szektorban. Ezen algoritmusok alkalmazásával javítható a termékek minősége, a gyártási és szervezési folyamatok, valamint a karbantartási stratégiák, ezáltal a termelő berendezések teljes berendezés-hatékonysága (*Over Equipment Effectiveness, OEE*).

V. Tézis

Karbantartási feljegyzések és automatizált gépi hibaüzenetek elemzését végeztem természetes nyelvi feldolgozó alkalmazással. Az ipari környezetben újszerű algoritmusok segítségével megmutattam, hogy a kézzel rögzített feljegyzések 49,27% visszatérő hibákra utal, ami hibák szubjektív leírása miatt automatizáltan nem, vagy csak nagyon nehezen kimutatható. Az eredmények alapján a ciklikus karbantartási tevékenységek optimalizálása szükséges. A gépi hibajelentésekben alkalmazott nevesített entitás felismerés segítségével azonosítottam a hibás komponenseket, így emberi beavatkozás nélkül a megfelelő személyzet értesíthető, ezáltal a reakcióidők és a berendezések teljes berendezés-hatékonysága (OEE) javítható.

A tézisemben foglalt eredményeket az alábbi publikációmban mutattam be: [S7]

12. Új tudományos eredmények összefoglalása

Disszertációmban a számítási intelligencia ipari alkalmazásait kutattam Ipar 4.0 környezetben. Az elméleti összefoglalóban bemutatott technológiákra és eredményekre építve céloom a kognitív számítástechnika ipari alkalmazásainak bemutatása és gyakorlati megvalósítása, mely segítségével a reakcióidők és a selejt termékek aránya csökkenthető, a gyártási minőség növelhető, ezáltal a karbantartási és gyártási költségek optimalizálhatók, hozzájárulva az energiatakarékos és fentartható gyártáshoz. A források alapján elkészítettem az általam alkalmazott adatgyűjtési eljárásokat és modelleket, valamint megmutattam azok eredményeit valós ipari környezetben.

Az ipari adatgyűjtés gyakorlati kihívásaira reagálva létrehoztam egy skálázható és standardizált adatintegrációs keretrendszert, mely a technológiai és architektúrais elemek mellett különböző szabályokat és konvenciókat alkalmaz és elengedhetetlen, úgynevezett *enabler* a nagy volumenű ipari adatgyűjtés menedzselhető megvalósításához. Az adatok standardizált gyűjtése lehetőséget teremt az ipari adatok mesterséges intelligencia alkalmazásával történő feldolgozáshoz.

Dolgozatomban megmutattam a mesterséges intelligencia ipari alkalmazásainak kihívásait és kérdéseit, valamint azonosítottam az MI sikeres alkalmazásához szükséges lépéseket és résztvevőket, akiket be kell vonni az AI megoldások sikeres implementációjához.

Gépi tanulási algoritmusok segítségével feltártam a mért adatok és a hibás termékek közötti összefüggéseket, illetve megmutattam a klaszterező algoritmusok alkalmazásának egy általánosan kiterjeszhető munkamenetét, amivel az ipari folyamatokban jelentkező anomáliákat lehet detektálni.

Az anomáliák detektálásának képességet felhasználva különböző ipari folyamatok mérési adatait osztályoztam, majd az így kapott OK – NOK csoportok segítségével LSTM és autoencoder alapú mesterséges neurális hálózatokat alkalmaztam folyamatfelügyeletre. Megmutattam, hogy az említett hálózatok rekonstrukciós hibájának mérésével biztosabb és a teljes mérési görbére kiterjedő, idősor jellegű adatok felügyeletére általánosan kiterjeszhetően alkalmazhatóak az általam alkalmazott megoldások.

Konvolúciós neurális hálózatok alkalmazásával megmutattam, hogy a klasszikus képfeldolgozó algoritmus megbízhatóságát csökkentő nemdeterminisztikus zavarok

eliminálhatók, így a hamis pozitív hibák száma és az emberi beavatkozás szükségessége csökkenthető.

Végül megmutattam, hogy a mérési és képi adatokon felül, az emberek által generált szöveges adatok, mint a karbantartási feljegyzések, valamint az automata gépi üzenetek természetes nyelvfeldolgozással történő elemzése segítségével a visszatérő hibák felismerhetők, a gyakran meghibásodó alkatrészek azonosíthatók, ezáltal a karbantartási tevékenységek optimalizálhatók és a reakcióidők csökkenthetők.

A következőben összefoglalom a disszertációmban megfogalmazott téziseket és a tézisek bizonyításaként felhasznált tudományos publikációm:

I. Tézis

Kidolgoztam és a gyakorlatban megvalósítottam egy újszerű adatintegrációs keretrendszert, mely hardver- és szoftverkomponensek, valamint névkonvenciók és szabályok segítségével heterogén ipari környezetben is standardizált és skálázható módon valósítja meg az adatgyűjtést. A hálózati entitások egyértelmű és standardizált azonosításával, az üzenetstruktúra szabványosításával, a hálózati architektúra egyszerűsítésével és az új adatfeldolgozó felhasználási esetek gyors integrálásának lehetőségével. Az adatintegrációs keretrendszer bevezetése növeli a transzparenciát és megkönnyíti a hálózat menedzselhetőségét, valamint lehetőséget teremt a mesterséges intelligencia módszerek alapjául szolgáló nagy volumenű ipari adatgyűjtéshez.

Munkám során azonosítottam a mesterséges intelligencia ipari alkalmazásainak kihívásait és nehézségeit, valamint bemutattam annak multidiszciplináris jellegét és rávilágítottam a *domain knowledge* fontosságára. Meghatároztam az MI sikeres ipari implementációjának stakeholder-eit és javaslatot tettem a sikeres ipari alkalmazás munkamenetére.

A tézisemben foglalt eredményeket az alábbi publikációmban mutattam be:

A tézisemhez tartozó publikációim a következők: [S1] [S2] [S3]

II. Tézis

Gépi tanulási algoritmusok segítségével feltártam a főtengely átforgatási nyomatékának mérési eredményei és a csapágykárokkal való motorkiesések közötti összefüggéseket. A PCA és LDA dimenziócsökkentő algoritmusok, valamint a K-átlag és a DBSCAN eljárások segítségével azonosítottam az anomáliát tartalmazó méréseket, melyekhez tartozó motorok további vizsgálata során felismertem, hogy az anomáliát tartalmazó mérésekhez tartozó belső égésű motorok főtengelyének axiális irányú játéka túrésen kívül esik, valamint az utómunka során idegenanyagot találtak a csapágyhelyeken. Az összefüggések feltárására bemutattam az általam kidolgozott, mért szerelési és megmunkálási folyamatokra általánosan kiterjeszthető gépi tanulási algoritmusok alkalmazásának munkamenetét és annak gyakorlati alkalmazhatóságát.

A tézisemben foglalt eredményeket az alábbi publikációmban mutattam be: [S4]

III. Tézis

Kidolgoztam és bemutattam az LSTM és autoencoder mesterséges neurális hálózatok ipari alkalmazásának munkamenetét ipari környezetben újszerű folyamatfelügyeltre és gyakorlati példákon bemutattam azt. Megmutattam, hogy a modellek kimeneti rekonstrukciós hibájának mérésével a teljes mérési görbe felügyelhetővé válik, így a felügyeleti ablakok és burkológörbék az ismertetett módszerekkel kiegészíthetők vagy azok akár el is hagyhatók. A bemutatott módszerek általánosan kiterjeszthetők bármilyen idősor jellegű mérési adatra és a hozzá tartozó folyamat felügyeletére.

A tézisemhez tartozó publikációim a következők: [S4] [S5]

IV. Tézis

Kidolgoztam és bemutattam a klasszikus képfeldolgozási alkalmazásokat zavaró nemdeterminisztikus hibák eliminálásának ipari környezetben újszerű módszerét konvolúciós neurális hálózatok alkalmazásával a hamis pozitív hibák számának csökkentésére. A módszer segítségével azokban az esetekben is érzékelhető az alkatrészek és komponensek megfelelő rögzítése vagy jelenléte, melyekben a klasszikus, pixelek vagy élek közötti távolságmérésen

alapuló módszerek segítségével a képek nem megbízhatóan vagy egyáltalán nem alkalmazhatóak a megvilágítás változásai vagy a kiértékelendő területet részben kitakaró zavarok miatt. A gyakorlati alkalmazással 97,46%-kal sikerült csökkentenem a hamis pozitív hibák számát. A bemutatott munkamenet és modell kiterjeszhető más ipari képfeldolgozó alkalmazásokra is.

A tézisemben foglalt eredményeket az alábbi publikációmban mutattam be: [S6]

V. Tézis

Karbantartási feljegyzések és automatizált gépi hibaüzenetek elemzését végeztem természetes nyelvi feldolgozó alkalmazással. Az ipari környezetben újszerű algoritmusok segítségével megmutattam, hogy a kézzel rögzített feljegyzések 49,27% visszatérő hibákra utal, ami hibák szubjektív leírása miatt automatizáltan nem, vagy csak nagyon nehezen kimutatható. Az eredmények alapján a ciklikus karbantartási tevékenységek optimalizálása szükséges. A gépi hibajelentésekben alkalmazott nevesített entitás felismerés segítségével azonosítottam a hibás komponenseket, így emberi beavatkozás nélkül a megfelelő személyzet értesíthető, ezáltal a reakcióidők és a berendezések teljes berendezés-hatékonysága (OEE) javítható.

A tézisemben foglalt eredményeket az alábbi publikációmban mutattam be: [S7]

13. Konklúzió, további feladatok

Napjainkban egyre nagyobb hangsúly kerül a különböző gépi tanulási algoritmusokra és számítási intelligencia megoldásokra épülő eszközökre és szolgáltatásokra. Az internet széleskörű elterjedésének, a közösségi platformok népszerűségének és az egyéb online szolgáltatásoknak hála rengeteg adat áll a rendelkezésünkre, melyek a mesterséges intelligencia algoritmusok fejlesztéséhez és tanításához elengedhetetlen. Ennek következtében elsősorban az online világban, valamint a szórakoztató és fogyasztói elektronikai eszközökben terjednek ezek a megoldások a leggyorsabban. Az Ipar 4.0 terjedésének köszönhetően azonban egyre nagyobb lehetőségek tárulnak fel a mesterséges intelligencia ipari felhasználására is. A terepi szinttől a vállalatirányítási rendszerekig egyre több és több eszköz kapcsolódik vállalati hálózatokhoz. Az IoT eszközök, különböző mérőberendezések és szenzorok a nap huszonnégy órájában generálják az adatokat, és az így létrejövő adathalmaz kiváló alapot kínál az MI módszerek ipari alkalmazásaira.

Munkámban bemutattam a számítási intelligencia ipari alkalmazásainak lehetőségeit és gyakorlati alkalmazásait. Az anomáлияfelismerésre, valamint az adatokban rejlő rejtett összefüggések feltárására kiváló lehetőséget biztosítanak a különböző gépi tanulás és mélytanulási megoldások. Ezen eszközök olyan lehetőségeket tárnak a felhasználók elé, melyekkel a döntési folyamatok egyszerűsíthetők és az azokhoz szükséges idők csökkenthetők, amivel jelentős anyagi és időbeli megtakarítások érhetők el. A selejtarány csökkentésével a minőség és az anyagi, energetikai ráfordítások csökkenthetők, valamint a berendezések rendelkezésre állásának növelésével a gyártási költségek optimalizálhatók és fenntarthatóbb, környezetkímélőbb termelés valósítható meg. A hamis pozitív hibák arányának csökkentésével, valamint a karbantartási feljegyzések elemzésével a ráfordított emberi kapacitások igénye redukálható, a repetitív feladatok aránya csökkenthető, ezáltal több kapacitás jut a magasabb hozzáadott értékű munkák elvégzésére.

Noha a bemutatott metódusok és technikák általánosan kiterjeszthetőek a disszertációban említett egyéb ipari feladatokra, fontos megjegyezni, hogy minden mesterséges intelligencia probléma más és más. Nincs olyan általános célú modell, amely minden hasonló jellegű probléma esetén ugyanolyan jól teljesít, ezért a felvázolt alkalmazások további kutatása, fejlesztése ajánlott. A bemutatott technológiák ugyan jó kiindulási alapot jelentenek a hasonló problémák megoldásához, viszont az egyes felhasználási eseteknél szükséges lehet az előfeldolgozási eljárás, a metódusok és modellek testre szabására. A modellek tanításához szükséges számítási kapacitások növekedése miatt fontos a felhőalapú megoldások ipari

bevezetésének vizsgálata. Továbbá fontos aspektus a szériaérett megoldások felkutatása és gyakorlati bevezetése, ezért a kutatás folytatása ajánlott.

14. Irodalomjegyzék

- [1] Schwab, K. (2017). *The Fourth Industrial Revolution*. Crown Business.
- [2] Hermann, M., Pentek, T., & Otto, B. (2016). Design principles for Industrie 4.0 scenarios. In 2016 49th Hawaii International Conference on System Sciences (HICSS) (pp. 3928-3937). IEEE. doi: 10.1109/HICSS.2016.488
- [3] Erboz, G. (2017). How to Define Industry 4.0: The Main Pillars of Industry 4.0. In *Managerial Trends in the Development of Enterprises in Globalization Era* (pp. 761-767). Nitra.
- [4] Plakhotnikov, D. P., & E. E. (2021). Design and Analysis of Cyber-Physical Systems. In 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus) (pp. 589-593). IEEE. doi: 10.1109/ElConRus51938.2021.9396364
- [5] Samuel, A., & Sipes, C. (2019). Making Internet of Things Real. *IEEE Internet of Things Magazine*, 2(1), 10-12. doi: 10.1109/IOTM.2019.1907777
- [6] Want, R., Schilit, B. N., & Jenson, S. (2015). Enabling the Internet of Things. *IEEE Computer*, 48(1), 28-35.
- [7] Singh, S., & Singh, N. (2015). Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 1577-1581). IEEE. doi: 10.1109/ICGCIoT.2015.7380718
- [8] Linthicum, D. S. (2017). Connecting Fog and Cloud Computing. *IEEE Cloud Computing*, 4(2), 18-20. doi: 10.1109/MCC.2017.37
- [9] Muhammed, A. S., & Ucuş, D. (2020). Comparison of the IoT Platform Vendors, Microsoft Azure, Amazon Web Services, and Google Cloud, from Users' Perspectives. In 2020 8th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-4). IEEE. doi: 10.1109/ISDFS49300.2020.9116254
- [10] Althobaiti, M. M. et al. (2021). An intelligent cognitive computing based intrusion detection for industrial cyber-physical systems. *Measurement*, 186, 110145. doi: 10.1016/j.measurement.2021.110145
- [11] Wang, Y. (2006). Cognitive Informatics: Towards Future Generation Computers that Think and Feel. In 2006 5th IEEE International Conference on Cognitive Informatics (pp. 3-7). IEEE. doi: 10.1109/COGINF.2006.365666

- [12] Kelly III, J. E. (2015). Computing, cognition and the future of knowing. IBM Corporation.
- [13] Wang, Y. (2009). Cognitive Computing and machinable thought. In 2009 8th IEEE International Conference on Cognitive Inform
- [14] Chen, M., Herrera, F., & Hwang, K. (2018). Cognitive Computing: Architecture, Technologies and Intelligent Applications. IEEE Access, 6, 19774-19783. <https://doi.org/10.1109/ACCESS.2018.2791469>
- [15] McCarthy, J. et al. (1955). A proposal for the Dartmouth summer research project on artificial intelligence. Retrieved from <http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html>
- [16] Nayak, A., & Dutta, K. (2017). Impacts of machine learning and artificial intelligence on mankind. In International Conference on Intelligent Computing and Control (I2C2) (pp. 1-3). Coimbatore, India. <https://doi.org/10.1109/I2C2.2017.8321908>
- [17] Zupan, J. (1994). Introduction to Artificial Neural Network (ANN) Methods: What They Are and How to Use Them
- [18] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2(11), 559-572. <https://doi.org/10.1080/14786440109462720>
- [19] Jolliffe, I. (2002). Principal Component Analysis. 2nd Edition. Springer.
- [20] Fisher, R. A. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7(2), 179-188. doi:10.1111/j.1469-1809.1936.tb02137.x
- [21] Rao, R. C. (1948). "The utilization of multiple measurements in problems of biological classification". *Journal of the Royal Statistical Society, Series B*. 10 (2): 159–203. JSTOR 2983775.
- [22] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
- [23] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (Vol. 1, pp. 281-297).
- [24] Lloyd, S. P. (1957). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), 129-137.
- [25] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

- [26] Arthur, D., & Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (pp. 1027-1035).
- [27] Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), 129-137. doi: 10.1109/TIT.1982.1056489
- [28] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3), 264-323. doi: 10.1145/331499.331504
- [29] Lloyd, S. (1982). Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2), 129-137.
- [30] Ketchen Jr, D. J., & Shook, C. L. (1996). The application of cluster analysis in strategic management research: An analysis and critique. *Strategic management journal*, 17(6), 441-458.
- [31] Xu, X., & Wunsch II, D. (2005). *Clustering*. John Wiley & Sons.
- [32] Dang, X. H., & Bailey, J. (2012). Clustering techniques for biological data. *Computational and structural biotechnology journal*, 1(1), e201204005.
- [33] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53-65.
- [34] Ester, M. et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96) (pp. 226-231).
- [35] Ankerst, M. et al. (1999). OPTICS: Ordering points to identify the clustering structure. *ACM SIGMOD Record*, 28(2), 49-60.
- [36] Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall.
- [37] Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. In Proceedings of the ACM SIGMOD Conference on Management of Data (pp. 103-114).
- [38] Suh, S., Kang, U., & Lee, J. (2011). A density-based clustering approach for outlier detection. *Expert Systems with Applications*, 38(5), 5275-5284.
- [39] Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. *International Conference on Database Theory*, 420-434.
- [40] Mirzaei, A., & Soltanian-Zadeh, H. (2012). A new method for MRI brain image segmentation based on K-means algorithm and fuzzy logic. *Journal of medical signals and sensors*, 2(3), 149.

- [41] Liu, Y., & Schmidt, B. (2008). A framework for clustering heterogeneous gene expression data. *Journal of Biomedical Informatics*, 41(5), 849-859.
- [42] Zhang, J., & Chen, K. (2012). Efficient clustering
- [43] Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65(6), 386-408. doi: 10.1037/h0042519
- [44] Haykin, S. (2009). *Neural Networks and Learning Machines* (3rd ed.). Pearson Education.
- [45] Minsky, M. L., & Papert, S. A. (1969). *Perceptrons: An introduction to computational geometry*. MIT Press
- [46] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807-814).
- [47] Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the international conference on machine learning (ICML)* (Vol. 30, No. 1, p. 3).
- [48] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- [49] Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249-256).
- [50] Ramachandran, P., Zoph, B., & Le, Q. V. (2018). Searching for activation functions. arXiv preprint arXiv:1710.05941.
- [51] Elfving, S., Uchibe, E., & Doya, K. (2018). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107, 3-11.
- [52] Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (GELUs). arXiv preprint arXiv:1606.08415.
- [53] Balduzzi, D., Frean, M., & Leary, L. (2020). The shattered gradients problem: If resnets are the answer, then what is the question? arXiv preprint arXiv:2003.02389.
- [54] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436-444.
- [55] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.
- [56] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

- [57] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- [58] Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2), 157-166.
- [59] Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural computation*, 12(10), 2451-2471.
- [60] Cho, K. et al. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724-1734).
- [61] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6), 602-610.
- [62] Mikolov, T. et al. (2010). Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.
- [63] Sutskever, Ilya & Vinyals, Oriol & Le, Quoc. (2014). Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems*. 4.
- [64] Tang, D., Qin, B., & Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1422-1432).
- [65] Bengio, Y., & LeCun, Y. (2007). Scaling learning algorithms towards AI. *Large scale kernel machines*, 34(5).
- [66] Greff, K. et al. (2017). LSTM: A search space odyssey. *IEEE transactions on neural*
- [67] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
- [68] Sermanet, P., & LeCun, Y. (2015). Convolutional neural networks applied to house numbers digit classification. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)* (pp. 3288-3291).
- [69] Zhang, L., Lin, L., Liang, X., & He, K. (2021). Deep learning for computer vision: A brief review. *IEEE Transactions on Multimedia*, 23(11), 2733-2753.
- [70] Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1), 1-127.
- [71] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

- [72] He, K. et al. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- [73] Yao, L. et al. (2015). Describing videos by exploiting temporal structure. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4507-4515).
- [74] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504-507.
- [75] Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19, 153.
- [76] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536.
- [77] Goodfellow, I. et al. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).
- [78] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.
- [79] Vincent, P. et al. (2008). Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning (pp. 1096-1103).
- [80] Alpaydin, E. (2010). *Introduction to Machine Learning* (2nd ed.). MIT Press.
- [81] Jain, A. K., & Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc.
- [82] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning* (1st ed.). Springer.
- [83] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly Detection: A Survey. *ACM Computing Surveys*, 41(3), 15:1-15:58.
- [84] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.
- [85] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: with applications in R*. Springer.
- [86] Murty, M. N. (2011). *Machine learning: An artificial intelligence approach* (Vol. 1). Springer.
- [87] Prechelt, L. (1998). Early stopping-but when? In *Neural Networks: Tricks of the Trade* (pp. 55-69). Springer Berlin Heidelberg.
- [88] N .Srivastava et.al (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 15. 1929-1958.
- [89] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb), 281-305.

- [90] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4), 303-314.
- [91] Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366.
- [92] Leshno, M., Lin, V. Y., Pinkus, A., & Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6), 861-867.
- [93] Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3), 930-945.
- [94] Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2), 251-257.
- [95] Zhang, T. (1998). A universal approximation theorem for neural networks with a single hidden layer. *IEEE Transactions on Neural Networks*, 9(6), 1330-1333
- [96] Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing*. Pearson.
- [97] Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT Press.
- [98] Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *arXiv preprint arXiv:1708.02709*.
- [99] Bender, E. M., & Friedman, B. (2018). Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6, 587-604.
- [100] Liu, B. (2012). *Sentiment analysis and opinion mining*. Morgan & Claypool Publishers.
- [101] Chen, S. F. (1998). Statistical language modeling using the CMU-Cambridge toolkit. *Proceedings of the Sixth Workshop on Very Large Corpora*.
- [102] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media, Inc.
- [103] Charniak, E. (1997). Statistical Parsing with a Context-Free Grammar and Word Statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)* (pp. 598-603).
- [104] Collins, M. (1999). Head-Driven Statistical Models for Natural Language Parsing. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (pp. 404-411).

- [105] McAllester, D. A. (1991). The Complexity of Parsing with a Single Combinatory Rule. In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (pp. 238-245).
- [106] Socher, R. et al. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1631-1642).
- [107] Borthwick, A. (1999). A maximum-entropy-inspired parser. Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, 403-410.
- [108] Gildea, D. (2008). Dependency parsing. Synthesis Lectures on Human Language Technologies, 1(1), 1-127.
- [109] Hutto, C. J., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. Eighth International AAI Conference on Weblogs and Social Media.
- [110] Sowa, J. F. (2014). Conceptual structures: Information processing in mind and machine. Springer.
- [111] Liakata, M., Saha, S., & Dobnik, S. (2012). Learning robust models of natural language semantics. Journal of Computer Science and Technology, 27(1), 77-89.
- [112] Grishman, R. (1997). Information extraction: Techniques and challenges. In IJCAI-97 workshop on new approaches to learning for natural language processing.
- [113] Goldberg, Y. (2016). A primer on neural network models for natural language processing. Journal of Artificial Intelligence Research, 57, 345-420.
- [114] Li, J. et al. (2019). Adapting sequence-to-sequence models for text normalization in social media. arXiv preprint arXiv:1908.01148.
- [115] Liu, C. et al. (2016). How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. arXiv preprint arXiv:1603.08023.
- [116] Papineni, K. et al. (2002). BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics (pp. 311-318).
- [117] Gehring, J. et al. (2017). Convolutional sequence to sequence learning. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 1243-1252).
- [118] Radford, A. et al. (2019). Language models are unsupervised multitask learners. OpenAI Blog

- [119] Van Rossum, G. (1995). Python tutorial. Centrum voor Wiskunde en Informatica Amsterdam.
- [120] Guttag, J. V. (2016). Introduction to Computation and Programming Using Python: With Application to Understanding Data (2nd ed.). MIT Press.
- [121] Lutz, M. (2013). Learning Python (5th ed.). O'Reilly Media, Inc.
- [122] McKinney, W. (2018). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. O'Reilly Media, Inc.
- [123] Abadi, M. et al. (2016). TensorFlow: A System for Large-Scale Machine Learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16) (pp. 265-283).
- [124] "TensorFlow." (2023. február 19.) Google, <https://www.tensorflow.org/>.
- [125] Azzolini, D., & Migliorati, P. (2021). Machine learning workflows with TensorFlow 2.0: from experimentation to production. Springer.
- [126] Gulli, A., & Pal, S. (2017). Deep learning with TensorFlow. Packt Publishing Ltd.
- [127] Baydin, A. G. et al. (2018). Automatic differentiation in machine learning: a survey. Journal of Machine Learning Research, 18(153), 1-43.
- [128] Goyal, N. et al. (2017). Accurate, large minibatch SGD: training imagenet in 1 hour. arXiv preprint arXiv:1706.02677.
- [129] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010 (pp. 177-186). Springer.
- [130] Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul), 2121-2159.
- [131] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- [132] Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 4(2), 26-31.
- [133] Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. arXiv preprint arXiv:1212.5701.
- [134] Dozat, T. (2016). Incorporating Nesterov momentum into Adam. ICLR 2016 Workshop.
- [135] Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

- [136] Varoquaux, G. et al. (2015). Scikit-learn: Machine Learning without Learning the Machinery. In *Journal of Machine Learning Research*, 16, 2825-2830
- [137] Raschka, S., & Mirjalili, V. (2021). *Python Machine Learning*, 3rd Edition. Packt Publishing.
- [138] Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, Inc.
- [139] Scikit-learn Documentation. (2023. február 19.). Retrieved from <https://scikit-learn.org/stable/documentation.html>.
- [140] Domingos, P. (2015). *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books.
- [141] Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.
- [142] Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, 81-91.
- [143] Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40.
- [144] Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260.
- [145] Alsmadi, I., Al-Ali, A. R., & AL-Rawashdeh, A. (2017). SCADA systems: An overview. *International Journal of Computer Science and Information Security*, 15(4), 48-58.
- [146] Patel, S. (2013). SCADA system: architecture, types and applications. *International Journal of Emerging Technology and Advanced Engineering*, 3(8), 535-539.
- [147] Chacko, A., & Babu, G. R. (2015). SCADA system: a brief overview. *International Journal of Science and Research*, 4(1), 743-746.
- [148] Wang, W., & Cui, J. (2019). Industrial cyber-physical system security: A tutorial overview. *IEEE Access*, 7, 105180-105191.
- [149] Verma, N., Jain, M., & Agrawal, R. (2018). Wireless sensor networks for industrial automation: Challenges and solutions. *Journal of Sensor and Actuator Networks*, 7(3), 33. doi: 10.3390/jsan7030033
- [150] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637-646. doi: 10.1109/JIOT.2016.2579198

- [151] Oussous, A., Benjelloun, F.-Z., Lahcen, A. A., & Belfkih, S. (2018). Big data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*, 30(4), 431-448. doi: 10.1016/j.jksuci.2017.08.001
- [152] Apache Kafka. (2021). Retrieved from <https://kafka.apache.org/> [cited 2023-01-22]
- [153] RabbitMQ. (2021). Retrieved from <https://www.rabbitmq.com/> [cited 2023-01-22]
- [154] Choo, K. K. R. (2020). An overview of Industry 4.0: Definition, components, and technologies. *Journal of Reviews on Global Economics*, 9, 126-137. doi: 10.6000/1929-7092.2020.09.09
- [155] Li, Z., Li, B., & Li, J. (2019). Research on equipment coding and naming system for power plants based on the characteristics of the equipment. *IOP Conference Series: Materials Science and Engineering*, 664(1), 012138.
- [156] Das, A., & Akbar, R. (2019). A comprehensive naming convention for industrial components. In *Advances in Manufacturing and Mechanical Engineering* (pp. 281-289). Springer, Singapore.
- [157] Gómez, J. A., & Santana, R. (2020). Development of a descriptive naming convention for industrial equipment. *Industrial Management & Data Systems*.
- [158] Chung, T. W., Chen, C. M., & Yeh, C. T. (2015). Development of a rule-based naming convention for equipment in a semiconductor manufacturing fab. *IEEE Transactions on Semiconductor Manufacturing*, 28(3), 307-314.
- [159] Berners-Lee, Tim, (2005). RFC 3986: Uniform Resource Identifier (URI): Generic Syntax doi:10.17487/RFC3986
- [160] Libes, D. (1990). Choosing a name for your computer, FYI 5, RFC 1178, DOI 10.17487/RFC1178, <https://www.rfc-editor.org/info/rfc1178>
- [161] Bray, T., Ed. (2017). The JavaScript Object Notation (JSON) Data Interchange Format, STD 90, RFC 8259 <https://www.rfc-editor.org/info/std90>
- [162] K. Birman, T. Joseph (1987). Exploiting virtual synchrony in distributed systems. In *Proceedings of the eleventh ACM Symposium on Operating systems principles (SOSP '87)*. Association for Computing Machinery, New York, NY, USA, 123–138. <https://doi.org/10.1145/41457.37515>
- [163] Yusuf, S. (2004) Survey of publish subscribe communication system, *Advanced Internet Application and System Design*
- [164] Light, R. A. (2017). Mosquitto: server and client implementation of the MQTT protocol. *The Journal of Open Source Software*, 2. doi:10.21105/joss.00265

- [165] Kuhn, M., Johnson, K. (2019). Feature Engineering and Selection. 10.1201/9781315108230.
- [166] Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.
- [167] Wickham, H. (2014). Tidy Data. *Journal of Statistical Software*, 59(10). <https://www.jstatsoft.org/article/view/v059i10/v59i10.pdf>
- [168] McLachlan, G. J. (2004). *Discriminant Analysis and Statistical Pattern Recognition*. Wiley Interscience. ISBN: 0-471-69115-1. MR1190469.
- [169] Vafaei, N., Ribeiro, R., & Camarinha-Matos, L. (2015). Importance of Data Normalization in Decision Making: case study with TOPSIS method.
- [170] Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data Preprocessing for Supervised Learning. *International Journal of Computer Science*, 1(1), ISSN 1306-4428.
- [171] Xu, S. et al. (2015). Data cleaning in the process industries. *Reviews in Chemical Engineering*, 31(5), 453-490.
- [172] L. Perez, J. Wang (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning.
- [173] Y. Lecun et al. (1995). Comparison of learning algorithms for handwritten digit recognition. *International Conference on Artificial Neural Networks*.
- [174] K. Simonyan, A. Zisserman (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.
- [175] S. Sabour, N. Frosst, G. Hinton, (2017). Dynamic Routing Between Capsules.
- [176] Radford, A. et al. (2018). Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf
- [177] Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.
- [178] Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1), 37-63.
- [179] Boughorbel, S., Jarray, F., & El-Anbari, M. (2017). Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PloS one*, 12(6), e0177678.
- [180] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427-437.

15. Saját publikációk

- [S1] Szűcs, B. (2023). Data Integration Framework to Collect Data from OT/IT Systems. *Acta Technica Jaurinensis*, 16(2), 34–41. <https://doi.org/10.14513/actatechjaur.00689>
- [S2] Szűcs, B., (2021). Machine learning based optimization of tool replacement strategy in machine tools 10th Jubilee Interdisciplinary Doctoral Conference 12-13th of November 2021 Book of Abstracts ISBN 978-963-429-820-5 pp. 102.
- [S3] Szűcs, B., Ballagi, Á. (2020). Challenges of the application of machine learning in the serial production. *Proceedings of the Engineering Symposium at Bánki*. pp. 19-22. ISBN 978-963-449-225-2
- [S4] Szűcs, B. (2018) Machine Learning Applications for Fault Detection in Internal Combustion Engine Assembly. *Abstracts Book of 10th International Engineering Symposium at Bánki*. pp. 82-83., ISBN: 978-963-449-111-8
- [S5] Szűcs, B., Ballagi, Á. (2020). An Industrial Application of Autoencoders for Force-Displacement Measurement Monitoring. *The 12th Conference of PhD Student in Computer Science*, pp. 28-31.
- [S6] Szűcs, B., Ballagi, Á. (2019). Reducing Pseudo-error Rate of Industrial Machine Vision Systems with Machine Learning Methods. *Acta Technica Jaurinensis*, 12(4), pp. 294–305. <https://doi.org/10.14513/actatechjaur.v12.n4.511>
- [S7] Szűcs, B., Ballagi, Á. (2019) Artificial Intelligence in Maintenance: The Industrial Application of Natural Language Processing. *Proceedings of the Engineering Symposium at Bánki* pp. 61-66. ISBN 978-963-449-189-7