



**NEUMANN JÁNOS
INFORMATIKAI KAR**



DIPLOMAMUNKA

**OE-NIK
2022**

Hallgató neve:
Hallgató törzskönyvi száma:

**Szántó Marcell
T/008824/FI12904/N**

Óbudai Egyetem
Neumann János Informatikai Kar
Biomatika és Alkalmazott Mesterséges Intelligencia Intézet

DIPLOMAMUNKA FELADATLAP

Hallgató neve: **Szántó Marcell**
Törzskönyvi száma: T/008824/FI12904/N
Neptun kódja: Q1ISP7

A diplomamunka címe:

Gesztusdetektor fejlesztése IMU alapon
Development of an IMU-based gesture detector

Intézményi konzulensek: Dr. Eigner György, Dr. Kertész Gábor
Külső konzulens:

Beadási határidő: 2022. december 15.

A feladat

A cukorbeteg döntéstámogatás egyik legfontosabb és sok esetben leginkább hiányos komponense a páciensek szempontjából az automata naplózási mechanizmusok. A döntéstámogatás elengedhetetlen részét képezi a páciensek által végzett naplózás – vagyis a fiziológiai és egyéb paraméterek (pl. kézzel mért vércukorszint, menstruáció, gyógyszeresedés, betegségek, ételbevitel, stb.) bevitele a rendszerbe. Ez azonban sok adminisztratív terhet ró a páciensekre, sok esetben pedig a rendszerek korlátozott használatához vezet.

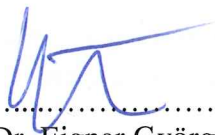
Ezen problémára jó megoldás a naplózást megkönnyítő funkciók építése, mint pl. a hang alapú diktálás, esetleg fotó bevitele ételmennyiség becslésére. Az automata naplózási mechanizmusok további kényelmi funkciót jelentenek, melyek épülhetnek pl. vércukorszintmérő szenzorokra, aktivitásmérő karkötőre, okosórára is.

A diplomamunka keretében a Jelölt feladata lesz a vonatkozó szakirodalom áttekintése és megismerése, saját szoftver, esetleg hardver rendszer fejlesztése az adatgeneráláshoz, valamint automata naplózási funkció fejlesztése, elsősorban az étkezéshez kötődő gesztusok rekogníciójára vonatkozóan.

A diplomamunkának tartalmaznia kell:

- szakirodalom felmérését,
- adatgenerálási módszerek kidolgozása,
- adattisztítási folyamatok kidolgozása,
- étkezéshez kötődő gesztusok felismerhetőségének vizsgálata,
- az eredmények kiértékelését.




.....
Dr. Eigner György
intézetigazgató

A diplomamunka elévülésének határideje: **2024. december 15.**
(OE TVSz 55.§ szerint)

A diplomamunkát beadásra alkalmasnak tartom:

.....
külső konzulens

.....
intézményi konzulens



HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 20. 22. 12. 12.

Hallgató aláírása



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:
Szántó Marcell..... Q1ISP7 MSc/ nappali
Telefon: Levelezési cím (pl: lakcím):
+36/30-7177-083 marci199803@gmail.com
Szakdolgozat / Diplomamunka¹ címe magyarul:
Gesztusdetektor fejlesztése IMU alapon
Szakdolgozat / Diplomamunka² címe angolul:
Development of an IMU-based gesture detector
Intézményi konzulens: Külső konzulens:
Dr. Eigner György, Dr. Kertész Gábor....
Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.09.28	Azonos témát feldolgozó kutatások elemzése	
2.	2021.10.27	Adatgyűjtési technikák áttekintése	
3.	2021.11.24	Adattisztítási és zajszűrési módszerek kiválasztása	
4.	2021.12.07	Alkalmazni kívánt gépi tanulási algoritmusok elemzése és kiválasztása	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4³ tantárgy követelményét teljesítette, beszámolóra / védésre⁴ bocsátható.

Budapest, 2021. 12. 10.


.....

intézményi konzulens

¹ Megfelelő aláhúzendó!

² Megfelelő aláhúzendó!

³ Megfelelő aláhúzendó!

⁴ Megfelelő aláhúzendó!



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:
Szántó Marcell Q1ISP7 MSc/ nappali

Telefon: Levelezési cím (pl: lakcím):
+36/30-7177-083 marci199803@gmail.com

Szakedolgozat / Diplomamunka¹ címe magyarul:

Gesztusdetektor fejlesztése IMU alapon

Szakedolgozat / Diplomamunka² címe angolul:

Development of an IMU-based gesture detector

Intézményi konzulens: Külső konzulens:

Dr. Eigner György, Dr. Kertész Gábor

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2022.02.22	SKlearn 2.0 eredmények értékelése, további adatgyűjtési módszer kidolgozása	h
2.	2022.03.24	Konvolúciós neurális hálózatok lehetőségeinek átbeszélése, stratégia kialakítása	h
3.	2022.04.13	LSTM modell létjogosultságának megvitatása	h
4.	2022.04.27	LSTM modell eredményeinek értékelése, további lépések tárgyalása	h

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakedolgozat I. / Szakedolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4³ tantárgy követelményét teljesítette, beszámolóra / védésre⁴ bocsátható.

Budapest, 2022.05.05

.....
intézményi konzulens

¹ Megfelelő aláhúzendő!

² Megfelelő aláhúzendő!

³ Megfelelő aláhúzendő!

⁴ Megfelelő aláhúzendő!



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:
Szántó Marcell..... Q1ISP7 MSc/ nappali
Telefon: Levelezési cím (pl: lakcím):
+36/30-7177-083 marci199803@gmail.com
Szakdolgozat / Diplomamunka¹ címe magyarul:
Gesztusdetektor fejlesztése IMU alapon
Szakdolgozat / Diplomamunka² címe angolul:
Development of an IMU-based gesture detector
Intézményi konzulens: Külső konzulens:
Dr. Eigner György, Dr. Kertész Gábor....
Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2022.09.06	Kombinált neurális háló létjogosultságának átbeszélése és tervezése	
2.	2022.10.11	Hang alapú gesztusdetekciós eljárás megvitatása	
3.	2022.11.08	Nyomtatott áramkör tervezési folyamat átbeszélése	
4.	2022.11.29	A dolgozat áttekintése, összegzése, javítások megvitatása, munka lezárása	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka⁴ tantárgy követelményét teljesítette, beszámolóra / védésre⁴ bocsátható.

Budapest, 2022. 11. 29.....

.....
intézményi konzulens

¹ Megfelelő aláhúzendő!

² Megfelelő aláhúzendő!

³ Megfelelő aláhúzendő!

⁴ Megfelelő aláhúzendő!

Tartalomjegyzék

1. Bevezetés	4
1.1. A népbetegség – Diabetes Mellitus	4
1.1.1. Epidemiológiai megfigyelések	4
1.1.2. Élettani és kórélettani alapok	5
1.1.3. A diabétesz fajtái	6
1.1.4. A vércukorszintmérés eszközei és jelentősége	8
1.2. Mélytanulás és gépi tanulás	9
1.2.1. Gépi tanulás	10
1.2.2. Mélytanulás	13
1.2.3. TensorFlow Lite 2.0	14
1.3. Az AI és a medicina	15
2. Irodalomkutatás	17
3. Módszertan, tervezett megvalósítás	19
3.1. Adatok	19
3.1.1. Adatgyűjtés	19
3.1.2. Adattisztítás	22
3.1.3. Zajszűrés	23
3.1.4. Adatnormalizáció	26
3.2. Gépi tanulási technikák	26
3.2.1. VGG16	26
3.2.2. DS-CNN	27
3.2.3. LSTM	27
3.2.4. AutoML-SKlearn 1.0 és 2.0	29
3.2.5. Neurális háló kombinációk	31
3.3. Szoftver környezetek	31
3.3.1. Jupyter Notebook App	31
3.3.2. Google Colaboratory	31
3.3.3. Docker	32
3.3.4. Microsoft Azure	33
3.4. Fizikai eszköztervezés	34
3.4.1. IPC szabvány	35
3.4.2. PCB tervezés	37
3.4.3. Alkatrész választás	39
3.4.4. Tesztelés	39
4. Metrikák és értékelési szempontok	41
4.1. AUC és ROC	41

4.2. Konfúziós mátrix	42
4.3. Precision, recall és accuracy	43
4.4. F1 score	44
4.5. Pipeline Profiler	44
5. Eredmények	46
5.1. VGG16	46
5.2. LSTM	46
5.3. AutoML-SKlearn 1.0	47
5.4. AutoML-SKlearn 2.0	48
5.5. DS-CNN	50
5.6. VGG16 + DS-CNN	50
5.7. SKlearn 2.0 + DS-CNN	51
6. Jövőbeni tervek	52
7. Összefoglalás	53
8. Summary	54
Ábrajegyzék	62
Táblázatok jegyzéke	64

Absztrakt

A társadalmunk jelentős százalékát érintő diabetes mellitus a patofiziológiás tünetek és szövődmények okozta komplikációk megnehezítik a benne szenvedők életét. Magyarországon 1980 óta több mint 314 millióval nőtt az érintettek száma. Egy 2022-es felmérés szerint hazánkban a prediabéteszes és diabéteszes esetek száma eléri az 1,1 milliót. Az ultrafeldolgozott élelmiszerek, mozgásszegény, egészségtelen életmód és a genetikai hajlam gyakran vezethet cukorbetegség kialakulásához. A betegség az inzulin nevű hormonhoz köthető és annak hiányában vagy a sejtek inzulinérzékenységének nagyfokú romlásában, tehát az inzulinhatás elmaradásában és adott esetben az inzulinelválasztás hiányában nyilvánul meg. A vércukorszintet több élettani folyamat mellett a szénhidrátbevitel befolyásolja a legjobban. Az inzulinterápia, mint szabályozási folyamat a vércukorszint megadott határértékek között tartását hivatott elvégezni. Ebből a rendszerből az ellenőrző jelet a vércukorszint mérésével kaphatjuk meg, beavatkozni pedig inzulin vagy antidiabetikum adásával tudunk. A jelenleg forgalomban lévő inzulinpumpák nem alkalmasak vércukorszint mérésre, a páciensnek kell elvégeznie napi több alkalommal. A gyakran alkalmazott alapbólus terápia hatékonyabbá tételével javítható a diabetes mellitusban szenvedő emberek életminősége és a szénhidrátbevitelük automatikus naplózásával csökkenthető a rájuk nehezedő adminisztratív terhelést. Munkám célja olyan hardver és szoftver készítése, amely alkalmas a fent említettek elvégzésére, elhanyagolható hibázási valószínűséggel. Korábbi tevékenységem során több eszközt is teszteltem, illetve saját adatgyűjtőt készítettem, amivel az étkezés mozgulatsorát tudtam azonosítani. Projektemben a munkám során kipróbált algoritmusok megbízhatóságát is megpróbáltam növeltem olyan módon, hogy a gyorsulás és elfordulásmérő által szolgáltatott adatokat kiegészítettem mikrofonnal rögzített hangokkal. Ezek a minták az étkezés leggyakoribb ismertetőjegyét tartalmazzák, amik az evőeszköz és tányér összeütközésekor keletkeznek. A két különböző bemenetet feldolgozó neurális háló kombinációja jobb hatékonyságot eredményezett. Egy DS-CNN modell segítségével teszteket végeztem az adathalmaz tanulhatóságát illetően. Munkám részét képezte még egy IPC szabványnak megfelelő nyomtatott áramkör elkészítése, ami az egészség-gügyi eszközök esetében nélkülözhetetlen sztenderd.

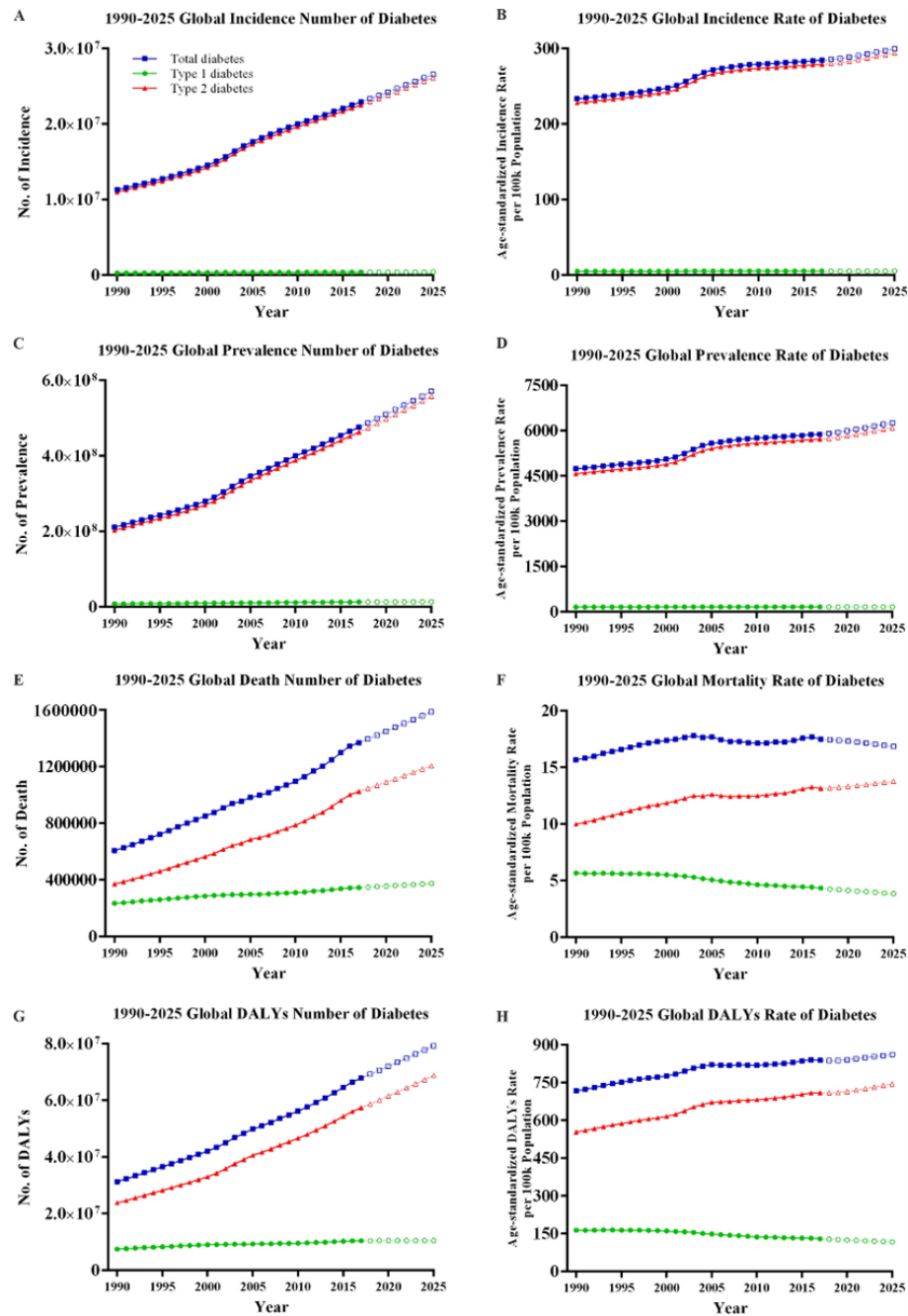
1. Bevezetés

A genetikai hajlam, illetve a modern életforma miatti elhanyagolt testmozgás, az egészségtelen táplálkozás gyakran olyan folyamatokat indít be a szervezetünkben, amelynek következtében prediabéteszes állapot, hosszabb távon pedig cukorbetegség alakulhat ki. Napjaink népbetegségeként is emlegetik a cukorbetegséget. Világszerte több millió ember életét nehezíti meg. A kór patofiziológias tünetei mellett a vércukorszintmérésre való odafigyelés, az inzulinterápia, illetve a szénhidrátbevitel naplózása is terhet jelent a benne szenvedők számára. Munkám során olyan eszköz készítését tűztem ki célul, amely segít levenni az adminisztratív terhelést a diabéteszes emberek válláról és segít javítani az inzulinpumpás kezelést.

1.1. A népbetegség – Diabetes Mellitus

1.1.1. Epidemiológiai megfigyelések

Világszerte a cukorbetegség száma az utóbbi 40 évben több mint 320 millióval nőtt [1]. Egy 2022-es felmérés adatai alapján jelenleg Magyarországon a diabéteszes és prediabéteszes állapotban lévők száma eléri az 1,1 millió főt [2]. Előrejelzések alapján ez a szám sajnos tovább fog növekedni. A kór kialakulásának okai között feltűnik a mozgásszegény életmód, a túlzott cukorbevitel, a feldolgozott és a magas glikémiás indexű ételek nagy mértékű fogyasztása, illetve a genetikai hajlam sem elhanyagolható.

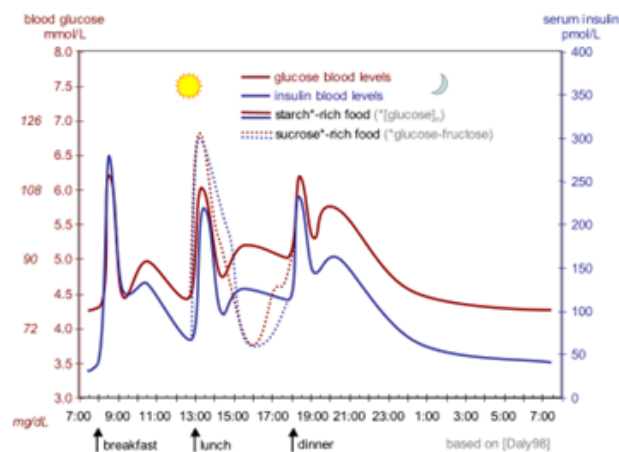


1-1. ábra. A diabéteszesek számának változása az idő függvényében [3].

1.1.2. Élettani és kórélettani alapok

A cukorbetegség egy olyan anyagcsere betegség, amely a vérben a cukor (glükóz) felhalmozódását okozza. A glükóz nélkülözhetetlen energiaforrása a szervezetnek, hasznosulása pedig a sejtekbe való belépés után megy végbe. A fent említett betegség esetében pont a sejtekbe való belépés folyamata akadályozott. A cukor a sejtekbe az inzulin hormon által a sejtmembránon aktivált glükóz transzportereken (úgynevezett GLUT ka-

pu fehérjén) keresztül jut be [4]. Itt sejttípustól függően a metabolizmus folyamatába kerülve a Szentgyörgyi-Krebs ciklus során hasznosul, vagy pedig keményítő formájában tárolódik. Jellemzően az inzulin aktivált GLUT kapuk a máj, vázizom és zsír sejteken találhatók nagy mennyiségben, mivel ezek a legnagyobb fogyasztói idegrendszer mellett a vércukornak. Az inzulint a hasnyálmirigy Langerhans-szigeteiben lévő béta-sejtek termelik, fő szerepe pedig az említett metabolikus út szabályozása, de egyéb szabályozási funkciói is vannak. Az inzulin a sejtmembrán receptorokon keresztül kapcsolódik a sejtekhez. Az inzulin hatására az említett "sejtkapuk" megnyitásával a vér cukorkoncentrációja csökken, a sejtek pedig energiához jutnak [4]. A vér cukorkoncentrációja a szervezet által termelt hormonok szabályozásának segítségével bizonyos határok között változik. A cukorkoncentráció csökkenését így tehát főként inzulin eredményezi, növekedését pedig annak antagonistája, a glukagon okozza azáltal, hogy serkenti a májban a glukoneogenezist. A glukoneogenezis az a folyamat, amely során a máj szőlőcukrot állít elő a vércukorszint megemelésére érdekében az inzulin hatására tárolt komplex cukormolekulákat lebontva [5].



1-2. ábra. Az inzulin és vércukorszint változása a nap folyamán [6].

1.1.3. A diabétesz fajtái

A diabétesznek több típusa is létezik, melyeket megkülönböztetünk attól függően, hogy milyen életkorban alakul ki a betegség, mi miatt alakul ki, illetve van-e szüksége a betegnek inzulinra. A főbb típusok ismertetése alább következik.

1.1.3.1. 1-es típus (T1DM) Ebben az esetben egy autoimmun betegségről beszélünk, ami abszolút inzulinhiányhoz vezet. Oka abban rejlik, hogy a szervezet a saját sejtjeit idegenként kezeli, támadást indít ellenük és ezen autoimmun reakció okán kialakult gyulladás miatt az inzulin termeléséért felelős béta-sejtek elpusztulnak. A betegség kezelése csak külső inzulin szervezetbe juttatásával lehetséges [7].

1.1.3.2. 2-es típus (T2DM) A másik leggyakoribb betegségi típus esetében relatív inzulinhiányról vagy inzulinrezisztenciáról beszélünk. A hosszú időn át (egészségtelen életmód miatt akár éveken át) tartó magas amplitúdójú inzulinhullámok miatt a sejtek megpróbálnak védekezni a túl sok beléjük kerülő glükóz ellen, amihez eszközként a cukortranszport folyamatát gátolják meg úgy, hogy nem reagálnak az inzulinra. Emiatt a vér glükóz szintje magas marad, ez pedig lokálisan a béta-sejteket, valamint globálisan az agy által idegi kapcsolatokon keresztül szabályozottan az inzulintermelés fokozására készíti a szervezetet. Innentől egy öngerjesztő folyamatról beszélhetünk, mivel a sejtek még inkább ellenállnak a magasabb inzulinszinteknek, a hasnyálmirigy béta-sejtjei pedig fokozzák a termelést. Egy idő után a hasnyálmirigy elkezd egyre kevesebb inzulint termelni, míg végleg leáll a folyamat [8]. A betegség diagnosztizálására jellemzően valamilyen mellékhatás kezelése okán feltárva kerül sor. A T2DM esetében jelentős eredményt lehet elérni a kezelésben életmódváltással. Előrehaladottabb esetben gyógyszeres kezelés kezdhető (orális antidiabetikum). Hosszabb távon inzulin adagolása is szükségessé válhat.

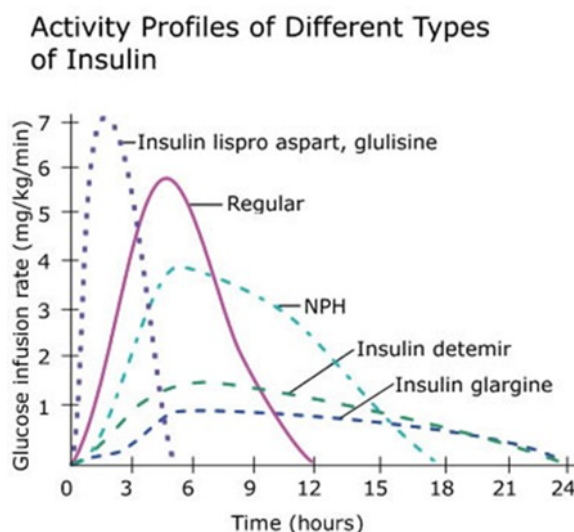
1.1.3.3. MODY típus Ez egy viszonylag ritkán előforduló cukorbetegség típus, ami a glükóz anyagcsere gén mutációja miatt fejlődik ki. Hatása szintén a béta-sejtek inzulintermelésében és kiválasztásában okoz zavart [9]. Az esetek nagy részében nincs szükség inzulinterápiára, hanem tablettás antidiabetikummal és az életmódra való odafigyeléssel karbantartható a betegség.

1.1.3.4. Terhességi diabétesz (GDM) Ennek kiváltó okai nem teljesen ismertek, de feltételezések szerint a terhesség harmadik trimeszterében a méhlepény által termelt hormonok kombinációja inzulinellenes hatást képes kiváltani [10]. Gyakran a megfelelő étkezéssel és a testmozgás rendszeresítésével kezelhető ám a szakemberek javaslatára inzulinterápia is szükséges lehet.

1.1.3.5. Pancreatogén diabétesz Hátterében a pancreas (hasnyálmirigy) kiterjedt gyulladás, vagy elhalása áll, ami miatt csökken az inzulintermelés. Miután a hasnyálmirigy pusztulása nem csak a béta-, hanem az alfa-sejtek elhalását is jelenti, ez a betegség hatással van a glukagon termelésére is. Emiatt megnő a hipoglikémiás roszullétek kockázata. Kezdetben diétával egyensúlyban tartható az anyagcsere, idővel a betegek inzulinkezelést igényelnek.

1.1.3.6. Gyógyszer okozta diabétesz Kémiai anyagok, esetleg gyógyszerek által kiváltott anyagcsere rendellenesség esetén beszélünk a diabétesz ezen formájáról [11]. Gyakran a kiváltó anyag elhagyása után a betegség megszűnik, de néhány esetben szükséges lehet gyógyszeres inzulinterápia is.

A fent említett esetek nagyrészt az inzulin külső adagolása szükséges a betegség kontrollálása miatt. Ehhez leggyakrabban humán vagy analóg inzulint alkalmaznak [12]. A humán inzulin azonos az emberi szervezetben megtalálható inzulinnal, míg az analóg típus ennek a molekulának egy módosított változata. Az alábbi ábrán a különböző fajta inzulinok hatásprofilja látható.



1-3. ábra. A különböző inzulinfajták hatásideje [13].

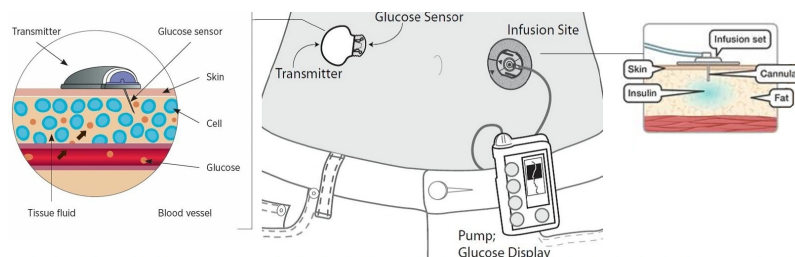
1.1.4. A vércukorszintmérés eszközei és jelentősége

Egy egészséges ember esetében a vércukorszint jól meghatározható értékek között mozog. Ez az érték éhomi esetben 4.0 - 5.4 mmol/L között változik, míg 2 órával étkezés után akár 7.8 mmol/L felett is lehet, melyről visszaáll az alapállapotba. Cukorbetegséggel küzdők esetében a vér glükóz szintje jelentősen megemelkedhet, ezt hiperglikémiának nevezzük. Ez legalább 25mmol/L cukorkoncentráció fölött relatív rövid távon az idegsejtek dehidratációján keresztül a tudatállapot beszűkülését, később pedig akár kómát is okozhat. Inzulinos kezelés esetében az injekció elmaradása, túl kis dózis beadása vagy a vércukorszint hibás mérése vezethet ehhez az állapothoz. Amennyiben a szükségesnél nagyobb dózisú inzulin került beadásra, esetleg az elvártnál korábban, a hipoglikémia veszélye áll fenn. Ez a károsan alacsony vércukorszint (2.2 mmol/L alatti érték) elérését jelenti, amit legtöbbször gyorsan felszívódó glükóz szervezetbe juttatásával lehet kezelni. Súlyos esetekben, illetve külső segítségnyújtás nélkül rohamhoz, hosszabb fennállása esetén akár ketoacidotikus kómához is vezethet [14]. A fentiekben említett szélsőséges állapotokhoz vezető úton fontos szerepet játszik a vércukorszint pontos meghatározása, hiszen anélkül nem lehet megállapítani a beadandó inzulin mennyiségét. Az adagolás történhet manuálisan egy inzulinfecskendővel, vagy inzulintollal, illetve automatikusan inzulinpumpa segítségével.



1-4. ábra. Az inzulinpumpa részei [15].

Mind az 1-es mind a 2-es típusú Diabetes Mellitus esetén szükséges a napi többszöri vércukorszint mérés, hiszen kezdetben a kezelőorvos ezáltal állítja be a bázis-bólus terápia adagolandó inzulin mennyiségét, a későbbiekben pedig önellenőrzés szempontjából fontos. További alapszabály az, hogy minden inzulin beadás előtt szükséges mérni a hipoglikémia elkerülése végett. Amennyiben a páciens inzulinpumpát használ, a folyamatos inzulinbevétel miatt gyakori vércukorszintmérés javallott, amit optimális esetben általában folyamatos vércukorszintmérő rendszer alkalmazásával lehetne megoldani. A mérés tehát történhet vércukorszintmérő készülékkel tesztsíkot használva, illetve CGM szenzorral [14]. A kettő közül az előbbi elterjedtebb. Ennek használata az ujjszúró láncza segítségével serkentett vér tesztsíkra juttatásával kezdődik. A mérőeszköz leggyakrabban kolorimetria vagy amperometria segítségével állapítja meg a cukorkoncentrációt. Ennél sokkal kényelmesebb megoldást kínál a CGM (Continuous Glucose Monitoring), ami folyamatos szöveti cukor monitorozást végez a bőr alatti szövetbe elhelyezett szenzor felhasználásával. Az ilyen eszköz nagyon hasznos, amikor a terápia beállításának kezdeti szakaszában tart a páciens, illetve valamilyen ismeretlen okból hosszabb időn keresztül visszatérő, ingadozó vércukorszintet észlel.



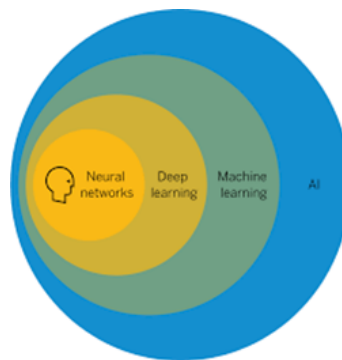
1-5. ábra. A CGMS és az inzulinpumpa elhelyezkedése [16].

1.2. Mélytanulás és gépi tanulás

A mesterséges intelligencia napjainkban igen kutatott és nagy reményekkel kecsegtető terület. Leggyakoribb felhasználási területe az előrejelzéseket nyújtó rendszerek illetve

terjedelmes adathalmazokban való mintafelismerést végző alkalmazások. Több típust is megkülönböztetünk a mesterséges intelligencia algoritmusok közt az alapján, hogy mennyire tág területen képes feladatokat elvégezni úgy, hogy az akár az emberi teljesítményt is meghaladja [17]. Ezen csoportosítás alapján megkülönböztetünk három típust:

- szűk mesterséges intelligencia
- általános mesterséges intelligencia
- mesterséges szuperintelligencia



1-6. ábra. Az AI típusok felosztása.

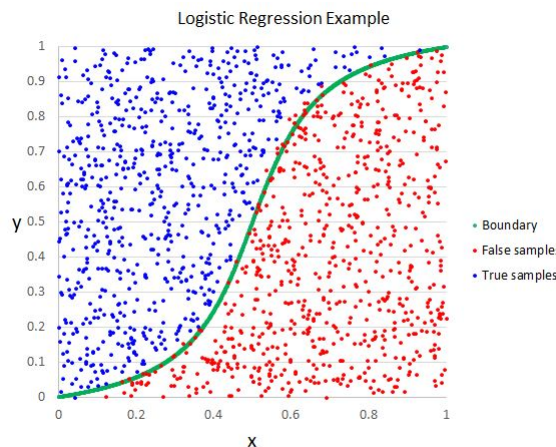
A szűk MI (weak AI) annyit takar, hogy az algoritmus egy adott feladatot el tud végezni igen nagy, adott esetben akár az embernél is nagyobb pontossággal. Az általános MI (strong AI) már képes az emberi eredményeken túlmutatni gyakorlatilag bármilyen intellektuális területen. Ekkor a gép már emberi szintű kreativitással rendelkezik. A mesterséges szuperintelligencia (ASI) az emberi képességeket minden területen képes lenne felülmúlni [18]. A kutatási eredmények azt mutatják, hogy ez még nem valósult meg, de a terület magasan kutatottsága és egyre mélyebb ismereteink idővel lehetővé teszik az ASI elkészítését is. A mesterséges intelligencia elkészítését tanulási folyamatnak szoktuk nevezni. Ennek két fajtáját különböztetjük meg: a gépi tanulást és a mélytanulást.

1.2.1. Gépi tanulás

A gépi tanulás célja, hogy az algoritmus képes legyen explicit feladatok végrehajtására azok leprogramozása nélkül. Több részkategóriákról is beszélhetünk a kategóriában. Az első a felügyelt tanulás. Ebben az esetben jelentős emberi beavatkozás szükséges. Lényegében az ember tanítja meg a modellt arra, hogy hogyan reagáljon a bemenő adatokra. A második kategória a félig felügyelt tanulás, amikor a modellt címkézett és címkézetlen adatok keverékével tanítják. Itt már kevesebb emberi beavatkozás szükséges.

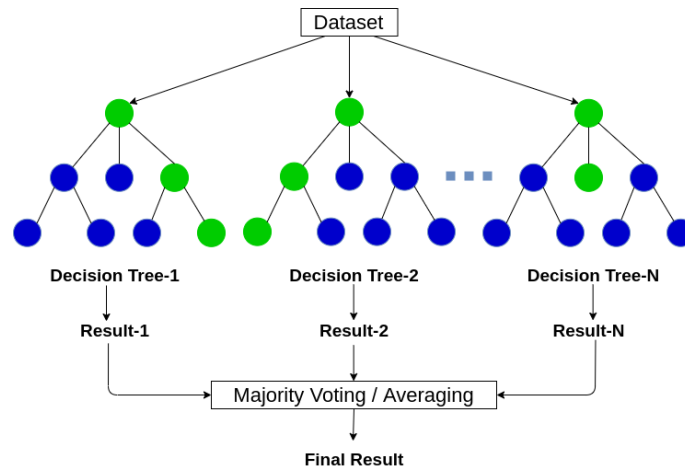
A harmadik kategória a felügyelet nélküli tanulási mód, amiben az algoritmus saját maga rendezi megadott kategóriákba a bemenő adatokat és önállóan keres mintázatokat. A negyedik kategória a megerősítéses tanulás, amikor az algoritmus a tanulás és működés során folyamatosan értékelésre kerül, amikor döntést hoz. Ha a döntése helyes, akkor pozitív megerősítést (jutalmat) kap, ha helytelen, akkor negatív megerősítést (büntetést). Ez a módszer eredményezi a leginkább rugalmas modelleket, amelyek a működésük során bármilyen bemeneti adathoz a leginkább tudnak alkalmazkodni.

1.2.1.1. Logisztikus regresszió A logisztikus regresszió megbecsüli egy bizonyos esemény (a függő változó) bekövetkezésének valószínűségét. Egy függő változó és kettő vagy több független változó közötti kapcsolat leírására szolgál. Általában a függő változó egy esemény. Az elemzés során azt vizsgáljuk, hogy ez az esemény bekövetkezik-e, vagy sem [19]. Az aktuális feladat hogy felismerjük hogy épp étkezés történik-e vagy sem, így egy bináris logisztikus regresszió alkalmas a feladatra.



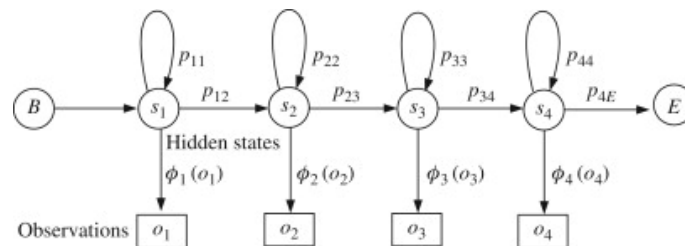
1-7. ábra. A logisztikus regresszió működése.

1.2.1.2. Random forest A random forest egy olyan felügyelt gépi tanulási algoritmus, amely döntési fából épül fel. Egy olyan speciális technika ez, mely több osztályozót ötvöz, hogy összetett problémákra is eredményt tudjon nyújtani. Maga az algoritmus generál egy "erdőt" és ezt az erdőt majd zsákolással vagy bootstrap-aggregálással tanítja be. Az eredményt a döntési fák előrejelzései alapján fogja megállapítani. Minél több fánk van (vagyis minél nagyobb az erdőnk) annál pontosabb lesz az eredmény is [20].



1-8. ábra. A random forest felépítése.

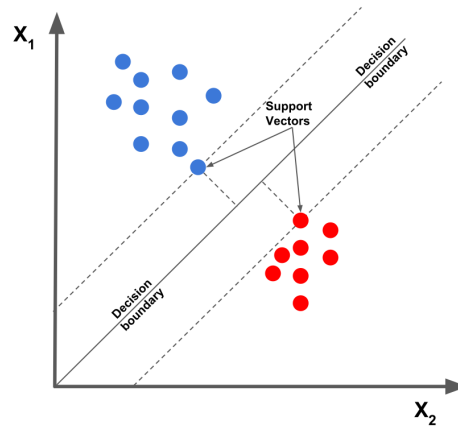
1.2.1.3. Hidden-Markov modell A HMM a Markov-lánc bővítésén alapul. A Markov-lánc egy olyan modell, amely információt szolgáltat random változók sorozatának valószínűségéről vagy állapotáról. Ezek mindegyike csak valamilyen meghatározott halmazból vehet fel értéket. A Markov-lánc feltételezi, hogy ha meg akarjuk jósolni a jövőt (a következő elem valószínűségét) a sorozatban, akkor csak a jelenlegi állapot (változó) számít. Ezek alapján a jelenlegi állapot előtti állapotoknak nincs hatása a jövőre, csakis a jelenlegi állapoton keresztül [21]. Ez olyan, mintha úgy jósolnánk meg a holnapi időjárást, hogy megvizsgáljuk a mai időjárást, de nem nézzük meg a tegnapi, vagy korábbi időjárást.



1-9. ábra. A Markov lánc működési elve.

1.2.1.4. Support-vector machine A gépi tanulásban az SVM-ek felügyelt tanulási modellek, hozzájuk kapcsolódó tanulási algoritmusokkal. Leggyakrabban osztályozás és regressziós elemzés céljából elemzik az adatokat. Az SVM az egyik legrobustusabb előrejelzési módszer, mivel statisztikai tanulási keretrendszereken alapul. Ha adott egy sor betanítási példa, amelyek mindegyike két kategória valamelyikéhez tartozik, az SVM tanító algoritmus olyan modellt épít fel, amely új példákat rendel egyik vagy másik kategóriához, tehát nem valószínűségi bináris osztályozóként működik. A mintákat térbeli pontokra képezi le úgy, hogy maximalizálja a két kategória közötti rés szélességét. Az új mintát ezután ugyanabban a térben leképezi és megjósolja, hogy melyik kategóriába tartozik

az alapján, hogy a rés melyik oldalán van. A lineáris osztályozáson kívül az SVM-ek hatékonyan képesek nemlineáris osztályozást is végrehajtani az úgynevezett kerneltrükk segítségével [22].



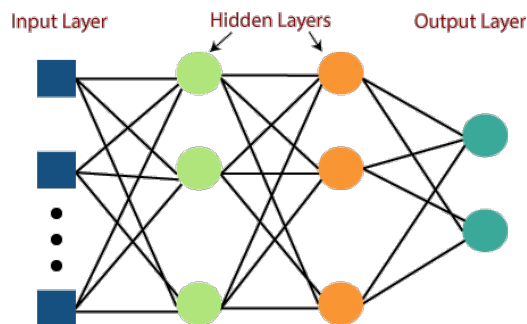
1-10. ábra. A support-vector machine működése.

1.2.1.5. Restricted Coulomb Energy Az RCE hálózatokban egy osztálynak egy vagy több régiója is lehet, és ezek a régiók átfedhetik egymást. Az RCE hálózat alapvetően három rétegből áll, a bemeneti, a kimeneti és a rejtett rétegből. A bemeneti réteg minden jellemzőhöz egy csomópontot (node-ot) tartalmaz. A kimeneti réteg minden osztályhoz egy csomópont tartozik. A rejtett rétegben minden csomópont egy prototípust képvisel. Minden osztály egy prototípushoz vagy egy csoporthoz kapcsolódik. Az RCE hálózat két módot tartalmaz, a tanulási és az osztályozási módot. A tanulási mód beállítja a bemeneti és rejtett réteg közötti kapcsolatok súlyát, és csökkenti a küszöbértékeket a rejtett csomópontokban, hogy kiküszöbölje a helytelen aktivitásokat [23]. Az osztályozási mód a prototípusok és az általuk befolyásolt mezők alapján hozza meg az osztályozási döntést. Egyes régiókhoz több osztály is tartozhat, és az RCE hálózat ezeket a régiókat "kétértelműnek" minősíti.

1.2.2. Mélytanulás

A mélytanulás az emberi agy működéséhez hasonlóan szerkezetet használ, ezt neurális hálónak nevezzük. A tanításához sokkal nagyobb adatmennyiség szükséges mint a gépi tanulás esetében, de jóval rugalmasabb és kétértelműségeket is könnyebben képes kezelni. A mélytanulás esetében több algoritmust is használhatunk, mint például a konvolúciós neurális hálózatok, az ismétlődő (recurrent) neurális hálózatok, stb. A neurális hálózatok több rétegből állnak, amely rétegek között haladva a bementi adatok egyre absztraktabbak lesznek.

1.2.2.1. Multilayer perceptron Az MLP egy speciális mesterséges neurális hálózat amely bemenetek halmazából generál kimenetek halmazát. Általában egyszerű regressziós problémák esetén használják őket. Magát a módszert az jellemzi, hogy több réteg bemeneti csomóponttal rendelkezik, amelyek irányított gráfként vannak összekötve a ki- és bemeneti rétegek között. Ez azt jelenti, hogy a csomópontokon keresztül vezető út csak egy irányba haladhat. Az MLP egy mély tanulási technika mivel több idegsejt réteggel képes dolgozni. Alapvetően arra törekszik, hogy emlékezzen a szekvenciális adatok mintáira. Legalább három réteg csomópontból áll: egy bemeneti, egy rejtett és egy kimeneti rétegből [24]. A bemeneti csomópontok kivételével minden csomópont egy neuron, amely egy nemlineáris aktivációs függvényt használ.



1-11. ábra. Egy MLP neurális hálózat leegyszerűsített képe.

1.2.2.2. Konvolúciós neurális hálók A VGG16 (Visual Geometry Group) egy mélytanulásra megalkotott konvolúciós neurális háló, ahol a 16 a modellben a tanítható rétegek számát adja meg. Az architektúra A. Zisserman és K. Simoyan munkája. A hálót arra tervezték, hogy az ImageNet adathalmazon, amely 14 millió képből áll és majdnem 1000 osztályból találhatók benne képek, tudjon klasszifikációt végrehajtani [25]. Nagy paraméterszámú háló, mivel meglehetősen magas filter számmal rendelkező rétegeket tartalmaz, ugyanakkor egyszerű architektúrája miatt számos feladat megoldására alkalmas tud lenni.

1.2.3. TensorFlow Lite 2.0

A TensorFlow Lite egy nyílt forráskódú mélytanulási keretrendszer, amelyet az eszközön történő következtetésekhez (Edge Computing) terveztek. A TensorFlow Lite olyan eszköz-készletet biztosít, amely lehetővé teszi az eszközön történő gépi tanulást azáltal, hogy lehetővé teszi a fejlesztők számára, hogy betanított modelljeiket mobil, beágyazott és IoT-eszközökön és számítógépeken futtassák. Támogatja az olyan platformokat, mint a beágyazott Linux, Android, iOS és az MCU. A TensorFlow Lite az eredeti TensorFlow (TF) könnyebb változata. A TF Lite kifejezetten mobil számítástechnikai platformokhoz és beágyazott eszközökhöz, szélső számítógépekhez, videojáték-konzolokhoz és digitális

fényképezőgépekhez készült. A TensorFlow Lite kvantálási technikákkal is optimalizálja a betanított modellt, ami következképpen csökkenti a szükséges memóriahasználatot, valamint a neurális hálózatok használatának számítási költségeit [26]. A TF-Lite előnyei:

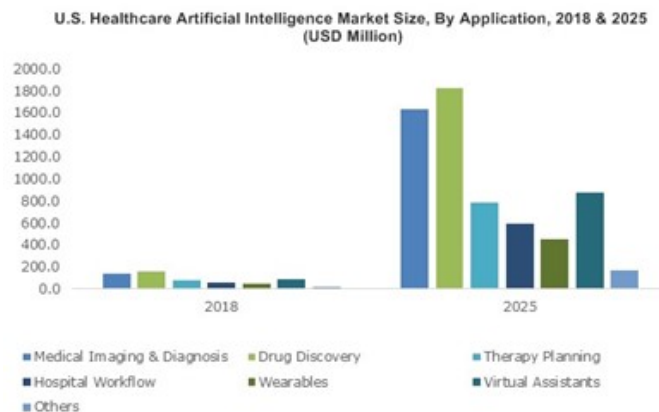
- **Modell átalakítás:** A TensorFlow modellek hatékonyan átvihetők TensorFlow Lite modellekbe a mobilbarát telepítés érdekében. A TF Lite képes optimalizálni a meglévő modelleket, hogy kisebb legyen a memória- és költségigénye, ami ideális helyzet a gépi tanulási modellek mobiltelefonon történő használatához.
- **Minimális késleltetés:** A TensorFlow Lite csökkenti a következtetési időt, ami azt jelenti, hogy a valós idejű teljesítményétől függő problémák a TensorFlow Lite-on ideálisan működnek.
- **Felhasználóbarát:** A TensorFlow Lite viszonylag egyszerű módot kínál a mobilfejlesztők számára, hogy Tensorflow gépi tanulási modelleket használó alkalmazásokat építsenek iOS és Android eszközökön.
- **Offline következtetés:** Az Edge ML következtetés nem támaszkodik internetkapcsolatra, ami azt jelenti, hogy a TF Lite lehetővé teszi a fejlesztők számára, hogy gépi tanulási modelleket telepítsenek olyan helyeken, ahol az internetkapcsolat drága vagy korlátozott. Az intelligens kamerákat például arra lehet tanítani, hogy távoli helyeken azonosítsák a vadon élő állatokat, és csak a videó adatfolyam bizonyos részeit továbbítsák.

Létrehozhatunk egy modellt a TensorFlow-ban, majd a TensorFlow Lite Converter segítségével TensorFlow Lite modellé alakíthatjuk. A TensorFlow Lite konverter optimalizálásokat és kvantálásokat alkalmaz a modell méretének és késleltetésének csökkentése érdekében. Ezek után a modell alig, vagy egyáltalán nem veszít az érzékenységből vagy a modell pontosságából. A TensorFlow Lite konverter egy optimalizált FlatBuffer formátumot hoz létre, amelyet a .tflite fájlkiterjesztés azonosít a kezdeti Tensorflow modell használatával. Munkám során vettem a legjobb teljesítményű modellt és a TensorFlow Lite konverter alkalmazásával készítettem belőle egy kisebb számítási képességű eszközön is futtatható modellt.

1.3. Az AI és a medicina

A mesterséges intelligencia alkalmazása a tudományok számos területén megfigyelhető, köztük az orvostudományokban is. Az egészségügy egy olyan terület, ahol nap mint nap rengeteg digitális adat képződik, ami az MI tanításához nélkülözhetetlen. Jelenleg ezen adatok feldolgozásának a nagy része emberi erővel történik [27]. Ez amellet, hogy sok időt és energiát igényel, magában hordozza a hibázás lehetőségét. A leletek kiértékelése leginkább hosszadalmas és odafigyelést igénylő folyamat illetve az ebben történő hibázás

akár emberi életet is követelhet. Szerencsére a mesterséges intelligencia kiválóan hasznosítható ezen a területen. Ma már számos helyen találkozhatunk vele, mint például a tüdőszűrés eredményeit vizsgáló neurális hálók, a patológia szöveteket analizáló algoritmusok, a röntgen és CT felvételeket kielemező MI-k, vagy akár olyan technológiák, amik a daganatos betegséggel küzdők kemoterápiás kezelésének optimalizálását hivatottak elvégezni [28], [29]. Ezek mellett a cukorbetegség terápiájában is van szerepük, például létezik olyan félautomata rendszer, ami a mesterséges hasnyálmirigy koncepciója szerint működik. Jelen pillanatban folynak olyan irányú kutatások, amik a megerősítéses tanulás segítségével a vércukorszint határértékek között tartását biztosítják. Ezen alkalmazások mellett még sok kisebb AI alapú alkalmazás létezik, amiknek egy nagy hányada a prevencióban betöltött szerepe által gyakorol hatást az életünkre. 2018-ban végeztek egy felmérést illetve szakértők véleményét kikérve készítettek egy predikciót a mesterséges intelligencia egészségügyben betöltött szerepét illetően. Ennek eredménye azt mutatja, hogy az AI felhasználása jelentősen növekedni fog a jövőben [30]. Ezt szemlélteti az alábbi ábra.

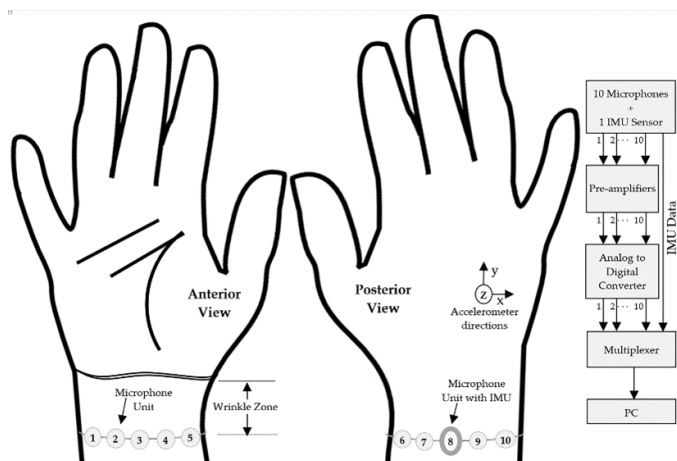


1-12. ábra. Az AI medicinában betöltött szerepköre [30].

2. Irodalomkutatás

A gesztusdetekció jelenleg egy intenzíven kutatott terület, hiszen a hétköznapiakban sok helyen alkalmazhatjuk. Például a HMI-k esetében egy teljesen új dimenziót nyithat meg a felhasználói élmény tekintetében. Emellett a pandémia is motiválta a kutatókat arra, hogy olyan érintés nélküli irányítási technikákat fejlesszenek ki, amik könnyen használhatók az átlag emberek számára [31]. Munkám során én nem a gép-ember kommunikációra összpontosítottam, hanem magának egy mozgásformának a felismerését valósítottam meg. Reményeim szerint és később már tapasztalataim alapján is a gyorsulás és szögelfordulási adatokon való neurális háló tanítás javított az algoritmusok pontosságán. Több szakirodalomban találkoztam olyan jelenséggel, hogy a kutatók kiegészítették az IMU adatait más szenzorokból nyert értékekkel, ezzel javítva azok hatásfokát. Többen próbálkoztak az EMG technológia felhasználásával, ami az alkar könyökhöz végének alsó oldalán elhelyezett 32 elektróda segítségével valósult meg. Ez az izmok illetve inak megfeszüléséből nyert ki információt. Hidden-Markov modell készítésével nem egyénfüggő algoritmust készítettek a szakemberek, aminek pontossága 96,6% [32]. Ez igen jó érték, ugyanakkor azt észre kell vennünk, hogy a hétköznapiak tekintetében nem használható a fizikai kialakítás. Emiatt szerettem volna a hangot mint információforrást használni, hiszen ott a levegő az átvivő közeg és nem szükséges extra kábel. Korábbi kutatásaim során egy MPU6050 6 tengelyű gyorsulás és elfordulásmérő szenzort használtam az adatgyűjtésre. Több helyen is találkoztam ezzel az eszközzel a szakirodalomkutatás során, ennek oka a megbízhatóság és pontosság mellett az olcsó ár és könnyű beszerezhetőség. Nem voltam egyedül azzal, hogy csupán ebből az egy szenzorból gyűjtött gyorsulás és elfordulás értékekkel próbáltam meg neurális hálót tanítani. A romániai Gheorghe Asachi Technikai Egyetem kutatói erre a datasetre RCE (Restricted Coulomb Energy) és MLP (Multi Layer Perceptron) gépi tanulási algoritmusok segítségével próbáltak gesztusdetekciót készíteni. Munkájuk gyümölcse egy 85,4%-os pontosságú és egy 88%-os pontosságú modell lett. Ezek mellett egy újabb technológiát is teszteltek, ami a DTW algoritmusban mutatkozott meg [33]. Ez a Dynamic Time Warping rövidítése. Olyan idősor elemzéses algoritmust jelent, amely két időbeli sorozat közti hasonlóságot méri. Ezzel jól meghatározható a periodikusan ismétlődő mozgásforma, illetve a különböző étkezési sebességek sem jelentenek kihívást [34]. A módszer eredménye 98,6%-os pontosság volt. A szakirodalomban leggyakrabban DTW, MLP és CNN (Convolutional Neural Network) algoritmusokkal találkozhatunk. Az általam használt mikrofonos IMU kiegészítés más kutatóknak is eszébe jutott, viszont a felhasználás kicsit különbözik. Míg én az étkezés hangjait próbálom felismerni mikrofonos adatgyűjtés révén, addig a hong kongi City University of Hong Kong egyetem kutatói az alkar bőrfelületén a nyomásváltozást vizsgálták. Minkét csuklóra 10-10 mikrofonból álló szenzoregységet helyeztek el, amely a bőrfelülethez tapadva alkalmassá vált az apró mozgások által keltett nyomásváltozások érzékeléséhez

[35]. A szakirodalomkutatás során számomra ez a megoldás tetszett a legjobban, úgy gondolom egy igen ötletes felhasználása volt ez a mikrofonoknak. Az IMU és mikrofontömb közös felhasználásából LDA (Latent Dirichlet Allocation) neurális hálót tanítva 80%-os pontosságú, míg SVM (Support Vector Machine) esetén 75%-os pontosságú modellt kaptak. A mérési elrendezés sematikus ábrája alább látható.



2-13. ábra. IMU és mikrofontömb segítségével megvalósított adatgyűjtés [35].

Mások olyan irányba indultak el, hogy a kéz mozdulatai helyett inkább a fej mozgását vizsgálták, amely egészen érdekes összefüggésekre világított rá a mintavételi frekvencia, a szabadságfokok és a tanított háló pontossága tekintetében. Munkájuk eredményéből jól látható, hogy a mozgás szabadságfokainak növelésével javult a háló teljesítménye. 3 szabadságfok esetén 75%, 9 szabadságfok esetén 87,56% volt a pontosság [36]. A mintavételi frekvencia fontosságát ott lehet észrevenni, hogy míg a kéz mozdulatainak vizsgálata esetén sokszor elég a 20 Hz, addig a fej apróbb, gyorsabb mozgásainak érzékelésére a javasolt frekvencia minimum 50 Hz. Ionut Cristian Severin és kutatótársai szerint [36]. Az Egyesült Királyság Sussexi Egyetemének kutatói az LSTM neurális hálókkal végeztek komoly munkát, mely során tizennyolc fajta kézmozdulat felismerésére tettek kísérletet. Az általuk választott MI algoritmus a finom motoros mozgások időbeni dinamikájára fókuszál és ez alapján próbál felismerni bizonyos mozgásokat. Az általuk publikált eredmények motiváltak engem arra, hogy a korábbi gyorsulási adatok mellett a szögelfordulási értékeket is adjam a tanítani kívánt algoritmus bemenetére, hiszen munkájuk kimenete azt mutatta, hogy a használt technológiák teljesítménye 15%-kal javult, amikor az accelerométer adatok mellett giroszkópból kapott értékeket is a bemenet-hez csatolták. Abban az esetben amikor a bemenő vektort tovább bővítették az előző két paraméter mellett mágneses érzékelők adataival, a kimeneten 20%-os javulás volt megfigyelhető [37].

3. Módszertan, tervezett megvalósítás

Munkám során adatgyűjtést végeztem mind a mozgás mint a hang tekintetében. A kéztől származó információkat a saját készítésű adatgyűjtővel végeztem. Korábban már rendelkeztem 310.000 rekordnyi adattal, amiket szintén ezzel az eszközzel mértem le, de az általam megvalósított kombinált neurális hálókhoz új adatokra volt szükségem, mert a régi méréseket nem tudtam időben szinkronizálni a hangfelvételhez. A hang rögzítésére Keyes KY-038 hangdetektort használtam, amit hozzáépítettem a régi adatgyűjtőhöz. A mérések szinkronizálásának segítése végett minden adatrögzítést egy határozott koppintással kezdtem, mert ez az IMU adatokban és a hangfüggvényen is egy rövid de magas csúcsnak felel meg.

3.1. Adatok

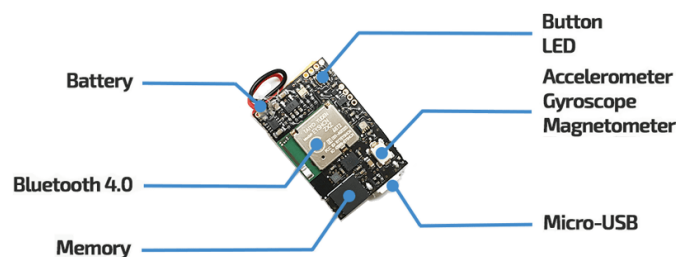
Munkám során 40 + 31 hosszú és közepesen hosszú mérést, illetve 15 rövid adatgyűjtést végeztem. Ezekből összesen 1.656.733 + 310.000 adatsornak (több mint 436 percnyi mérés) felelnek meg, melyekből 368 perc mérése tartalmaz hang adatokat is.

3.1.1. Adatgyűjtés

Jelen dolgozat elkészítéséhez 31 mérést végeztem, amiben már mindkét információs csatorna adatai rögzítésre kerültek. A többi 40 mérés csupán IMU adatokat tartalmaz. Teljes napos mérésekre nem került sor az idő hiányában, ám mindenképp szükséges a jövőben, hiszen a nap során igen sok hanghatással találkozunk. A korábbi (csak IMU-val végzett) mérések esetén 4 teljes nap került felvételre. Az új 31 mérés összesen 368 percnyi étkezést tartalmaz. A hang rögzítésének mintavételi frekvenciája 44.1 kHz volt. Emellett külön készítettem olyan hangrögzítéseket, amikben az evőeszközöket ütöttem különböző anyagú tárgyakkal, edényekkel. Az alap elképzelésem az volt, hogy a korábbi algoritmust kiegészítem egy olyan neurális hálóval, ami az tárgy és evőeszköz ütközésének hangjára triggerelődik. Később ezt még kiegészítettem egy nagyon jellegzetes hang felismerésével, ami a fémdobozos italok nyitócupakjának a reccsenését és szisszenését jelenti. Ezzel nem lehet minden étkezésfajtát felismerni, ugyanakkor a tárgyból való evés illetve az energiatartalmú és sör fogyasztásának detektálása megbízhatóbb lett. A mérések között kettő olyan található, amely étteremben került rögzítésre. Ennek fontosságát az alapzaj megfelelő kiküszöbölésének igénye adta. A sörös és energiatartalmú dobozok nyitásától 15 mérés készült.

3.1.1.1. Felhasznált eszközök Az adatgyűjtéshez gyorsulás és elfordulás mérésére alkalmas szenzort tartalmazó eszközöket kell használni. A piacon több ilyen készülék is található, de saját eszköz készítése is a kutatás részét képezi.

- **Mbientlab MetamotionS** A MetaMotionS egy 6 tengelyes IMU alapú szenzor. Kommunikációra az eszköz Bluetooth Low Energy (BLE) kommunikációs protokollt használ. A benne található 100mAh-s akkumulátornak, illetve BLE kommunikációnak köszönhetően a szenzor képes hosszabb ideig (24-48 órán keresztül) folyamatos adat továbbításra. A szenzor csak gyorsulás és szögelfordulás adatgyűjtésre és annak továbbítására lett kialakítva. Rendelkezik sleep funkcióval is, aminek megfelelő használatával még tovább növelhető az élettartam. A giroszkóp és a gyorsulásmérő adatait egy 16 bites, Bosch BMI270-es 6 tengelyes IMU szenzor szolgáltatja, melyet a gyártó leírása alapján gesztusfelismerésre és naplózásra terveztek [38]. Ennek köszönhetően a szenzor érzékenysége is mintavételezési frekvenciája is széles tartományban állítható. A gyorsulásmérő mintavételi frekvenciája egészen 0.78125 Hz-től 1600Hz-ig terjed. Számomra jelenleg a 25, 50, 100 Hz-es értékek relevánsak, ugyanis az általam vizsgált mozgásformák esetében ezen tartományok, már tudnak kellően sűrűn mintavételezett mérési adatokat szolgáltatni. A giroszkóp esetében a legkisebb mintavételezési érték a 25 Hz, legnagyobb pedig a 3200Hz [39].



3-14. ábra. MetamotionS szenzoregység [38].

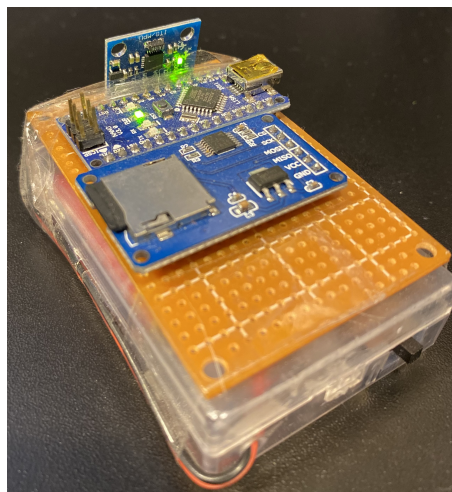
A tesztmérések alapján a beállított mintavételezési frekvenciát az eszköz $\pm 0,6-0,9\%$ -os hibával tudja tartani. Minden mintavételi pontot egyedi időbélyeggel láttam el a mérés során, így az első és utolsó mintavétel között eltelt idő különbségét elosztva a mintavételi pontok darabszámával meghatározható, a valós frekvencia értéke.

- **Panoramic Digital Health** A készülék tartalmazza a releváns szenzorokat, azzal a kiegészítéssel, hogy egy magnetométer is helyet kapott a PCB-n, amivel a giroszkóp időnkénti kalibrálása megoldható. Magát a karpántot úgy tervezték meg, hogy klinikai vizsgálatokhoz és orvosi kutatásokhoz is megfelelő legyen adatgyűjtési szempontból. 1GB belső memóriával rendelkezik, amelynek tartalmát a hubon keresztül naponta lehet szinkronizálni [40]. Az adatforgalom tanulmányozása során kiderült, hogy az eszköz már nyers adatokkal kommunikál, titkosítatlan formában, így segítségével lehetőség nyílik valós idejű adatgyűjtésre.



3-15. ábra. Panoramic Digital Health eszköz.

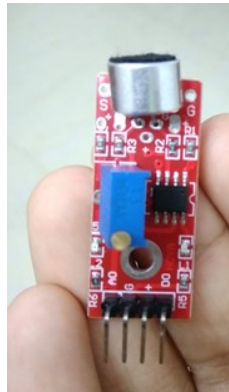
- **Saját készítésű adatgyűjtő** A fenti eszközök használata mellett úgy döntöttem, hogy én is készítek egy adatgyűjtő készüléket. Ennek megvalósítására egy Arduino Nano-t használtam mikrovezérlő gyanánt, Keyes KY-038 analóg kimenetű mikrofont hangrögzítési célból, IMU szenzornak pedig egy MPU6050 -re [41] esett a választásom. Az adatokat az eszköz egy SD kártyára menti el .txt formátumban. A mintavételi időt a kutatásokban olvasottakhoz hasonlóan 25 Hz-re állítottam be. Az eszköz tápellátásáról 2 db ceruzaelem gondoskodik. A megfelelő szíjak segítségével csuklóra rögzítés és bekapcsolás után elindul az adatgyűjtés.



3-16. ábra. Saját készítésű adatgyűjtő eszköz.

3.1.1.2. Gyorsulási és szögelfordulási adatok mérése A gyorsulási és szögelfordulási adatok rögzítésére a magam készítette adatgyűjtő eszközt használtam. A mintavételi frekvenciát 75 Hz-re állítottam be, a felhasznált szenzor pedig egy TDK gyártmányú MPU6050 volt. A mérés során kapott legnagyobb érték 19841 volt, a legkisebb pedig -6116 volt, a gyorsulás tekintetében és 311 illetve -204 a szögelfordulás esetében, melyeket a korábbiakhoz hasonlóan 0 és 1 közötti tartományba normalizáltam. Korábbi tapasztalataim alapján a normalizáció 5% - 8%-os javulást eredményezhet. Az adatgyűjtő a domináns kéz csukló fölötti részének felső oldalához volt rögzítve, a mért értékeket .csv formátumban mentettem el.

3.1.1.3. Hang adatok mérése A hangok mérésére egy Arduinohoz gyakran használt Keyes KY-038 analóg kimenetű mikrofont [42] használtam. A mintavételi frekvencia 44.1 kHz volt.



3-17. ábra. Keyes KY-038 mikrofon modul.

Az eszköz használata nem igényelt módosítást a korábban elkészített eszközön, viszont az érzékenység állítására szolgáló potenciométer sok kísérletezést igényelt. A rögzített adatokat .mp3 formátumban kaptam meg, majd a feldolgozásra saját Python scriptet készítettem.

3.1.2. Adattisztítás

Az adattisztítás során a hiányzó vagy kilógó adatok eltávolításával és odaillő értékek pótlásával foglalkoztam. A gyorsulás és szögelfordulás szenzor esetében ennek nagyobb relevanciája volt, mert a mintavételi frekvencia 75 Hz volt és az egységnyi pillanatig tartó extrém értékek nem jelenthetnek valós elmozdulást. A hang esetében nem végeztem adattisztítást, mert a környezeti zajok túl nagy mennyiségben voltak jelen. Emiatt ha a kilógó értékeket normalizáltam volna, akkor lehet hogy számomra fontos információkat veszítetek el. Az IMU adatok .csv fájlokban vannak tárolva, így egy egyszerű Python script

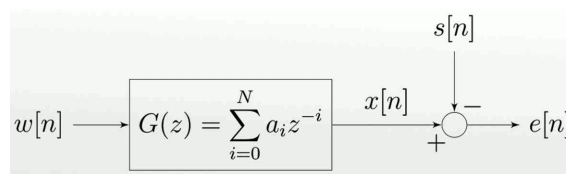
segítségével el tudtam végezni az adattisztítást. Az egységnyi outliereket helyettesítettem a környező értékek átlagával, valamint a hiányzó rekordok esetében is ugyanígy tettem.

3.1.3. Zajszűrés

A zajszűrés korábbi tapasztalataim szerint nagyon fontos, mert a mért jelre sok zaj szuperponálódik, amely aztán negatívan befolyásolhatja a mesterséges intelligencia algoritmus tanulási képességét.

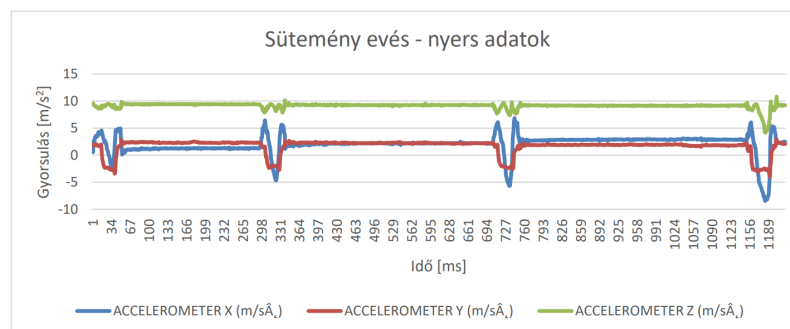
3.1.3.1. IMU adatok zajszűrése Minden olyan információt, ami a mérést torzítja el kell, hogy távolítsak. Ez lehet néhány elem, illetve lehet egy teljes mérésben jelenlévő sztochasztikus számhalmaz. A második esetben zajról beszélünk. Ennek okozója lehet a PCB nem megfelelő kialakítása, egy árnyékolatlan kábel, vagy valamilyen erős elektromágneses sugárzást kibocsátó elektronikai eszköz, esetleg a környezetből származó befolyásoló tényező (például autóban való étkezés esetén az út egyenetlenségei miatti rázkódás). Mivel ennek értéke nem determinisztikus, nem tudjuk konkrétan meghatározni az értékét. Emiatt az egész adathalmazon átfogóan kell bizonyos zajszűrési műveleteket végezni. Zajszűrés szempontjából az alábbi algoritmusokat teszteltem.

- **Átlagoló szűrő:** Használatakor egy ablakot kell megadni, amelyet az algoritmus az adathalmazon végigtolva, a benne lévő értékek átlagával közelíti az ablak középső elemének értékét. Ez valójában az adatok egymáshoz való közelítésén alapszik, ami végeredményként a zaj miatti kisebb eltéréseket kiküszöböli. Simításának erőssége az ablak méretével egyenesen arányos [43].
- **Wiener szűrő:** Célja egy ismeretlen jel statisztikai becslése, amiből következik, hogy az algoritmus statisztikai megközelítésen alapul, az elméletet pedig a minimális átlagos négyzetes hiba (MMSE) becslésének elmélete adja. Három hibafajtát képes kezelni, amiből az első a nem kauzális (nem okszerű) hibákra vonatkozik. Ennek megoldásához végtelen sok múlt és jövőbeli adat ismeretére van szükség, így a valós idejű rendszerekben nem alkalmazzuk. A második esetben a kauzális (okszerű) hibák kiküszöbölésére használható a szűrő. Ekkor végtelen sok múltbeli adat ismerete szükséges [44]. A harmadik opció a véges impulzusválaszú eset (FIR), amikor is csak a bemenő adatokra van szükség, mint jelen esetben.

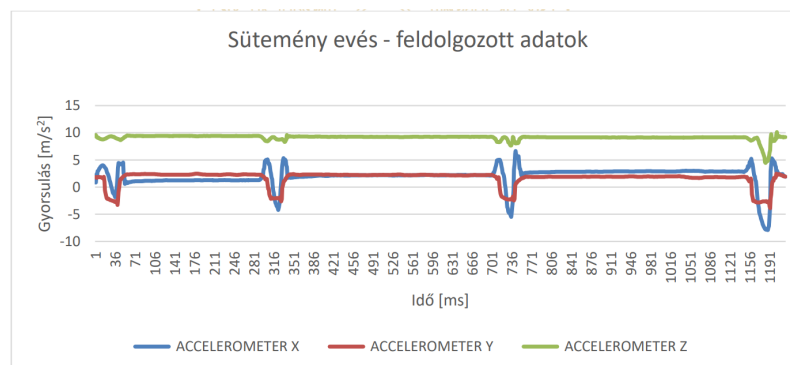


3-18. ábra. A FIR Wiener szűrő blokkvázlata diszkrét sorozatokhoz.

- **IIR szűrő:** Ezt a szűrő végtelen impulzus válasz szűrőnek is nevezik. Az auto-regresszív mozgásátlag elvét alkalmazva a szűrő a múlt és a jelen pillanat bemeneteit, illetve a múlt kimeneteit felhasználva működik. Ez az elvet leggyakrabban a sztochasztikus idősorelemzésnél használják, ahol egy kiszámíthatatlan fehér zajjal terhelt jel adja a bemenetet. A fehér zaj egy olyan fajta zajnak, amelynél a teljesítményspektrum állandó egy adott frekvencia- tartományra nézve [45]. Például az autóban való étkezés esetében nagyon nagy valószínűséggel találkozunk fehér zajjal, hiszen a kocsi sáv periodikus rezgése állandó sebesség mellett egy jól meghatározható frekvencia-tartományban lesz.



3-19. ábra. Egy sütemény elfogyasztásának szűretlen mozgási mintája.

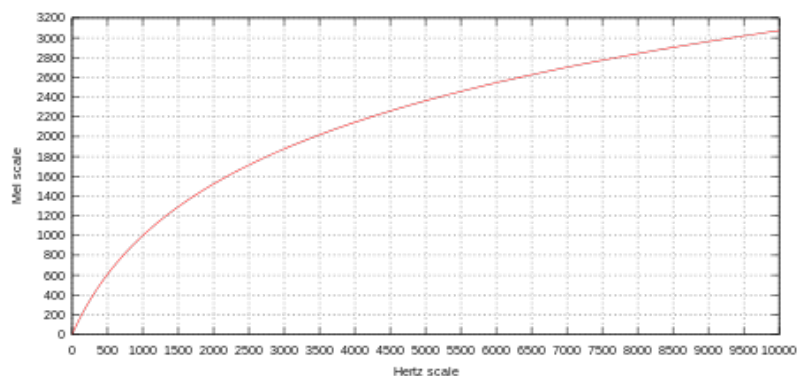


3-20. ábra. Egy sütemény elfogyasztásának szűrt és tisztított mozgási grafikonja.

- **Savitzky-Golay szűrő:** Ez egy olyan szűrő, amely mintavételezett adatpontok halmazára alkalmazható. Leginkább simítás céljából alkalmazzák úgy, hogy a jelváltozási irányok minimálisan torzuljanak csak. Ezt úgy érik el, hogy a szomszédos adatpontok egymás utáni részhalmazaira egy alacsony fokú polinomot illesztenek a legkisebb négyzetek módszerével [46]. Az így kapott közelítésekkel kapott jel lesz majd a simított jel.

3.1.3.2. Hang adatok zajszűrése A hangok esetében más zajszűrést kell alkalmazni. Erre a mel-spektrogram jelenti a megoldást, ami lényegében egy jel frekvenciájának lo-

garitmikus transzformációját tartalmazza. Ez lényegében egy olyan skála, amin az egyes értékek olyan távolságra vannak megjelenítve mint ahogyan azt az emberi fül is hallaná [47]. Tehát a mel skála segítségével a különböző frekvenciájú hangokat az emberi szem által is jól értelmezhető ábrán tudjuk megjeleníteni. Nem létezik egy univerzális mel skála, viszont a legtöbb képlet úgy határozza meg, hogy 1000 Hz-hez 1000 mel-t rendel a hallható küszöb fölött 40dB-nél és a frekvencia növekedésével csökken az együttható amivel növeljük a frekvenciához tartozó mel értéket. A mel skálát szemlélteti az alábbi ábra.

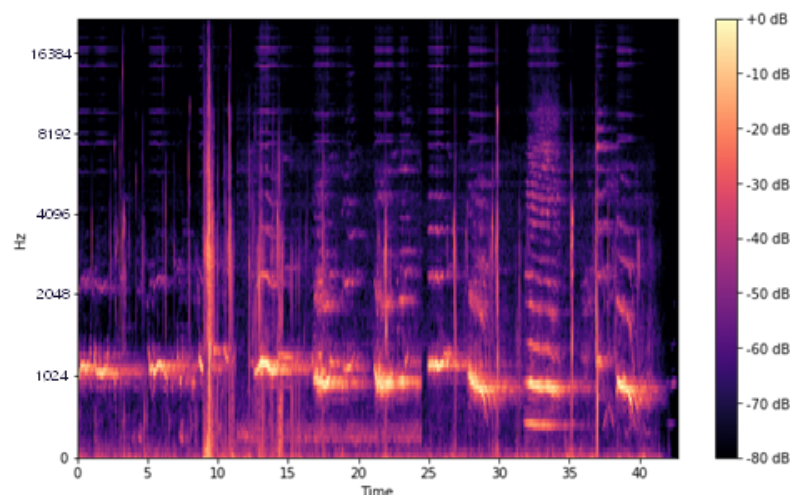


3-21. ábra. A mel skála és a frekvencia skála közti függvénykapcsolat.

A frekvencia és a mel közti átváltás képlete a következő [48]:

$$m = 2595 \cdot \log_{10} \left(1 + \frac{f}{700} \right) \quad (3.1)$$

A mel spektrogram segít abban, hogy csak egy meghatározott részére összpontosítsunk egy hangrészletnek. A következő képen látható 45 másodpernyi étkezés mel spektrogramja mely során levest fogyasztottam el. Jól látható ahogyan 1kHz környékén kirajzolódik a kanál és tányér ütközésének hangja. A mérést ábrázoló esetben csekély volt a háttérzaj, emiatt látható relatíve tiszta spektrogram.



3-22. ábra. Leves kanalizásának nem szűrt mel spektrogrammja.

Azokban az esetekben, amikor a háttérben egyéb zajforrások voltak, szükség volt a spektrogram szűrésére. Ehhez sáváteresztő szűrőt használtam, amit az evőeszköz és tányér koccanásakor keletkező hang illetve a fémdobozos üdítőital nyitásakor létrejövő hang frekvenciájának áteresztésére kalibráltam. Empirikus tapasztalataim alapján az első esetben a 900 Hz – 1200 Hz közti sávot, míg a második esetben 450 Hz - 650 Hz közötti tartományt kell áteresztetni. A spektrogrammal való munka a Python librosa könyvtárának használatával került megvalósításra, ami jelentősen meggyorsította a munkafolyamatot.

3.1.4. Adatnormalizáció

Az adatnormalizációra első sorban az IMU adatok esetében volt szükség, ugyanis a korábbi adathalmazom több fajta eszközzel végzett méréssel készült, melyeknek az érzékenysége különböző volt.

3.2. Gépi tanulási technikák

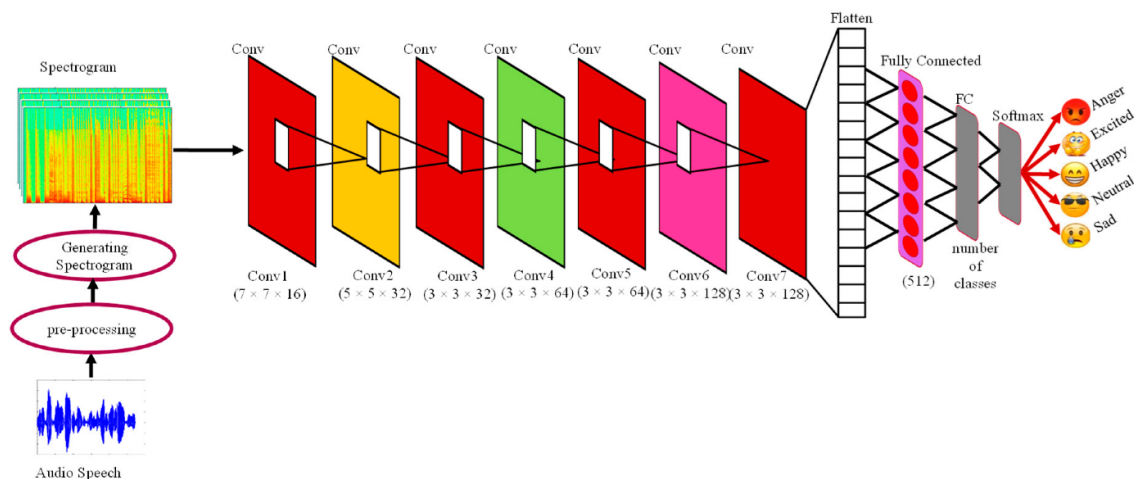
3.2.1. VGG16

A VGG16 konvolúciós neurális hálót leggyakrabban képek feldolgozására használják [49]. A 16 réteg lévén mély összefüggések megtalálására is képes. A munkám során 20 másodperces időablakot alkalmaztam, az adatokon pedig nem végeztem szűrést. A háló az eredeti 2 dimenziós konvolúcióról 1 dimenziós konvolúcióra lett átállítva. A (100,6)-os bemeneti vektor mérete függ a mintavételi frekvenciától. Jelen esetben az adathalmaz alulmintavételezésen esett át, mely után a mintavételi frekvencia 5 Hz-nek felelt meg. Ennek oka az volt, hogy igen nagy volt a bemenő adat mérete és ez az erőforráskorlátokba ütközött. Ezzel a háló megkapta a szögelfordulási adatokat is. Fontos megjegyezni,

hogy a tanítást normalizált és nem normalizált adatverzióval is kipróbáltam, de végülis korábbi tapasztalataim miatt a nyers adatokat használtam a legtöbb esetben.

3.2.2. DS-CNN

A DS-CNN (Deep Stride Convolutional Neural Network) a hang alapú detekció területén gyakran használt algoritmus. Az én választásom is a szakirodalomkutatás során olvasottak miatt esett erre az algoritmusra. A bemeneti paraméter egy két dimenziós jellemzőtérkép, ami a szűrt mel spektrogram. A vizsgálni kívánt jellemzők alapján 1 másodperces időablakokra bontottam a hangfájlt és azokból készítettem spektrogramokat. A DS-CNN felépítése alapján először egy átlagos konvolúciós rétegen megy keresztül a bemenő adat, miközben különböző szűrőket is átfuttat rajta az algoritmus. A konvolúciós rétegek után következik egy szakaszos normalizációs réteg ReLu aktivációs függvényvel. Ennek szerepe leginkább az algoritmus gyorsításában illetve a túltanulás elkerülésében nyilvánul meg. A legvégén egy fully-connected softmax réteg van, ami megmondja hogy a hallottak alapján a vizsgálni kívánt esemény történt-e, vagy valamilyen más hanghatás szerepel az adatokban [50]. A DS-CNN architektúrája a következő ábrán látható:

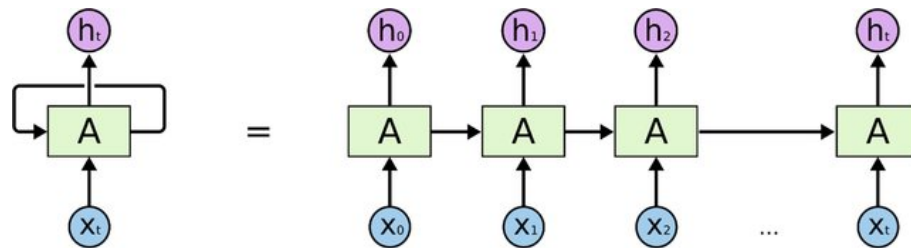


3-23. ábra. Egy DS-CNN felépítése [50].

3.2.3. LSTM

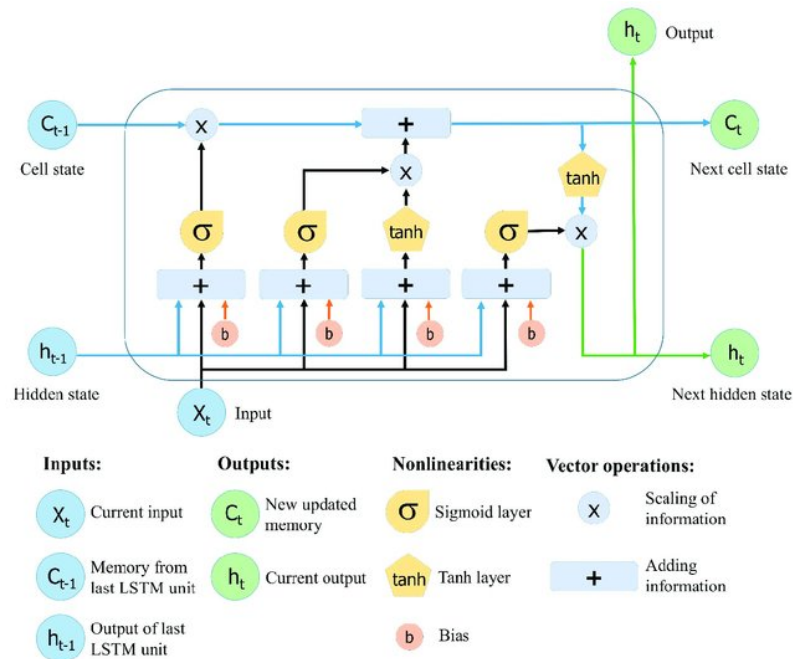
1997-ben mutatta be Hochreiter és Schmidhuber a hosszútávú memóriahálózatok koncepcióját a hosszú távú függőségi probléma elkerülésére. Azóta nagyon sokan fejlesztették már a modell alapkoncepcióját és ma már rendkívül sok helyen használják őket. Gyakorlati jelentőségük abban mutatkozik meg, hogy hosszú ideig emlékeznek információkra a múltból, amik hatással vannak a jelenbeli döntéseikre [51]. Az LSTM (Long Short Term Memory) egy speciális fajtája az ismétlődő neurális hálózatoknak (RNN - Recurrent Neural Network). Az RNN olyan neurális háló osztály, amelyben az egyes csomópontok

közötti kapcsolat irányított vagy irányítatlan gráfot alkot egy időbeli sorozat mentén [52]. Ez teszi lehetővé, hogy időben dinamikus viselkedést mutasson a hálózat. Mint minden visszatérő neurális hálózat, az LSTM is láncszerű felépítésű és működésének kulcsfontosságú eleme a rajtuk végigfutó cellaállapot. Ez egyenesen végigfut a teljes láncon, csupán néhány kisebb lineáris kölcsönhatás éri. Nagyon könnyű az információt változatlan formában átküldeni rajta. Az LSTM képes információkat eltávolítani vagy információkat hozzáadni a cellaállapotához. Ezt kapuknak nevezett struktúrák szabályozzák.



3-24. ábra. Szekvenciális feldolgozás egy RNN-ben [52].

Az ábrán jól láthatóak a korábban említett részei az LSTM neurális hálózatnak. A kép bal felső sarkában jelzett cellaállapot végighalad a hálón, miközben lineáris műveletek, mint a szorzás vagy összeadás módosíthatják az értékét. Tudjuk, hogy az RNN-ek visszatérő egységeket használnak a szekvenciaadatokból való tanuláshoz. Ugyan így működnek az LSTM-ek is. Abban azonban, hogy mi történik a visszatérő egységben, a kettő nagyon különbözik. Ha belenézünk egy szabványos RNN egyszerűsített ismétlődő egységdiagramjába, akkor azt látjuk, hogy csak két fő művelet van, az előző rejtett állapot kombinálása az új bemenettel, és átadása az aktiválási függvényen. Miután a rejtett állapotot a t lépésben kiszámoltuk, visszaadjuk az ismétlődő egységnek, és a $t+1$ lépésben a bemenettel kombinálva kiszámoljuk az új rejtett állapotot. Ez a folyamat $t+2$, $t+3$, $\dots$, $t+n$ esetén ismétlődik, amíg el nem éri az előre meghatározott számú (n) időlépést. Eközben az LSTM különféle kapukat alkalmaz annak eldöntésére, hogy milyen információkat tartson meg vagy dobjon el. Ezenkívül hozzáad egy sejtállapotot, amely olyan, mint az LSTM hosszú távú memóriája.



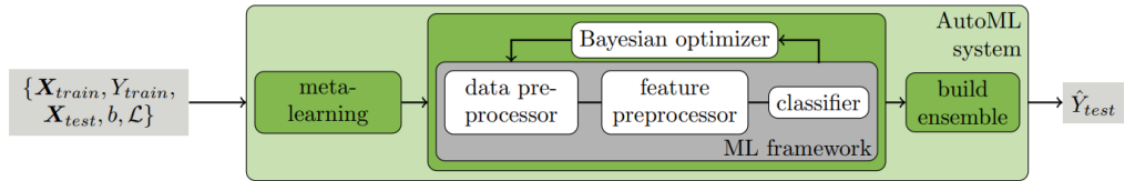
3-25. ábra. Egy LSTM neurális hálózat felépítése [53].

- Rejtett állapot és új bemenetek – egy korábbi időlépés rejtett állapota (h_{t-1}) és az aktuális időlépésben lévő bemenet (x_t) egyesítve vannak.
- Elfelejtési kapu – ez a kapu szabályozza, hogy mely információkat kell elfelejtetni. Mivel a szigmoid függvény 0 és 1 között van, beállítja, hogy a cella állapotában melyik értékeket kell elvetni, azaz 0-val szorozni, megjegyezni, azaz 1-gyel szorozni vagy részben megjegyezni, azaz 0 és 1 közötti értékkel megszorozni.
- A bemeneti kapu – segít azonosítani a fontos elemeket, amelyeket hozzá kell adni a cella állapotához.
- Cellaállapot frissítése – először a korábbi cellaállapot (C_{t-1}) megszorozódik a felejtési kapu eredményével. Ezután hozzáadunk új információt, hogy megkapjuk a legújabb cellaállapotot (C_t).
- Rejtett állapot frissítése – az utolsó rész a rejtett állapot frissítése. Az utolsó cellaállapot (C_t) átkerül a tanh aktivációs függvényen, és megszorozódik a kimeneti kapu eredményével.

3.2.4. AutoML-SKlearn 1.0 és 2.0

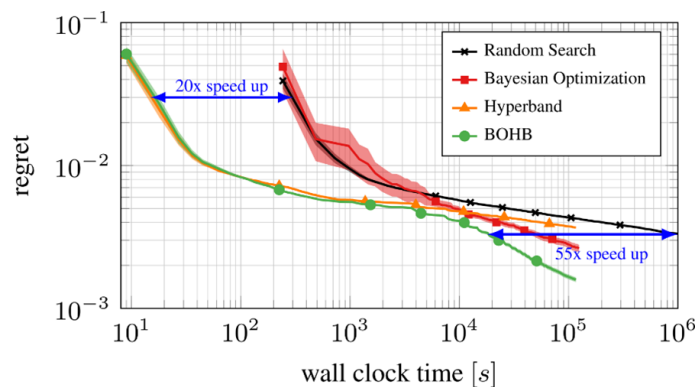
Az AutoML SKlearn a Scikit-learn könyvtárat használja az adatok megfelelő átalakításához és a gépi tanulási algoritmusokhoz [54]. Előnye, hogy segít kiválasztani több modell közül, hogy melyik a legjobban teljesítő, ehhez pedig egy Bayes-optimalizálást használ.

Végeredményként megkapjuk a legjobb AI modellt. az ahhoz tartozó hiperparaméterekkel, illetve egyéb osztályozókat is kapunk a hozzájuk tartozó paraméter javaslatokkal, amelyek közül aztán szabadon tudunk választani.



3-26. ábra. Az AutoML algoritmus felépítése [55].

Az AutoML-SKlearn három fő komponensből tevődik össze. Ezek a metatanulás, a Bayes-i optimalizálás és az együttes modellkialakítás. Az AutoML-SKlearn 2.0 már több fejlesztést is tartalmaz az elődjéhez képest [55]. Például a Bayes-optimalizálás helyett BOHB (Bayesian Optimizer and HyperBand) módszert alkalmaztak, ami egy multi-fidelity optimalizálási módszer, ami sebességben sokkal jobb az elődjéhez képest. Emellett a fejlesztők javítottak az algoritmus rugalmasságán, automatizálták a meta-optimalizálás problémáját, valamint új funkciók is megjelentek. Tehát amikor osztályozást vagy regressziót akarok alkalmazni egy új adatkészleten, az auto-sklearn a meta-funkció ki-nyerésével kezdődik, hogy megtalálja az új adatkészlet és a meta-learningen alapuló tudásbázis hasonlóságait. A következő lépésben, amikor a keresési terület kellőképpen összezsugorodik a meta-learning révén, a Bayes-optimalizálás megpróbálja megtalálni és kiválasztani a jobban teljesítő ML-modellt. Az utolsó lépésben az auto-sklearn felállítja az együttes modellt a legjobb ML modell alapján [56]. Az Auto- Sklearn 2.0 -ban olyan multi-fidelity (többhűségű) optimalizálási módszert alkalmaztak, mint a BOHB. Megmu-tatták azonban, hogy egyetlen modellválasztás nem alkalmas a probléma minden típusára, és több stratégiát is integráltak [57].



3-27. ábra. BOHB optimalizálás versus Bayes-optimalizálás sebessége [57].

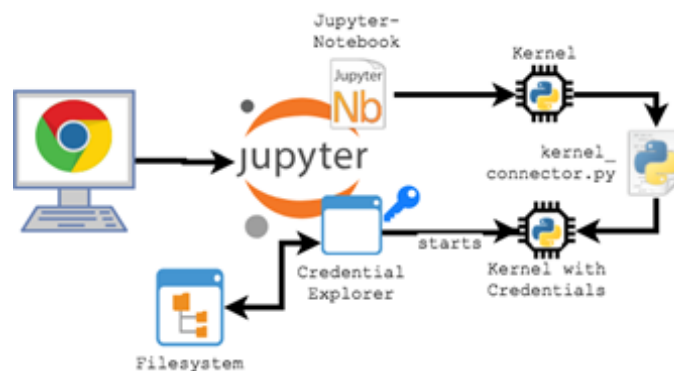
3.2.5. Neurális háló kombinációk

A tanított neurális hálók legjobban teljesítő modelljeit együttesen is teszteltem. Összesen két teszt esetet végeztem el. Az egyikben a két konvolúciós háló együttes működését vizsgáltam, a másikban az AutoML algoritmus és a DS-CNN összegzett hatásfokát derítettem fel. Az együttes vizsgálat azon az elven alapszik, hogy amennyiben mindkét háló étkezést ismer fel ugyan abban az időintervallumban, akkor biztosan mondhatjuk, hogy szénhidrátbevitel történik. Viszont ha valamelyik háló nem detektál semmit, akkor kizárom annak lehetőségét, hogy a vizsgált alany étkezik. A hálók tesztelése külön – külön történt és később, manuálisan végeztem el az időpontok szinkronizációját.

3.3. Szoftver környezetek

3.3.1. Jupyter Notebook App

A Jupyter Notebook App egy olyan server-kliens alkalmazás, ami lehetővé teszi notebook dokumentumok szerkesztését és futtatását böngészőn keresztül. A Jupyter Notebook App futtatására van lehetőség lokális számítógépen, ami nem igényel internet-hozzáférést illetve telepíthető egy távoli szerverre, ami interneten keresztül érhető el. A notebook dokumentumok megjelenítése, szerkesztése, futtatása mellett a Jupyter Notebook App rendelkezik egy "irányítópulttal" ami megjeleníti a helyi fájlokat, és lehetővé teszi a notebook dokumentumok megnyitását vagy a kernelek leállítását. A kernel egy úgynevezett "számítási motor", ami a notebook dokumentumban található kódot hajtja végre [58]. A futtatandó programtól függően a Jupyter Notebook App RAM és CPU igénye jelentősen megnőhet.

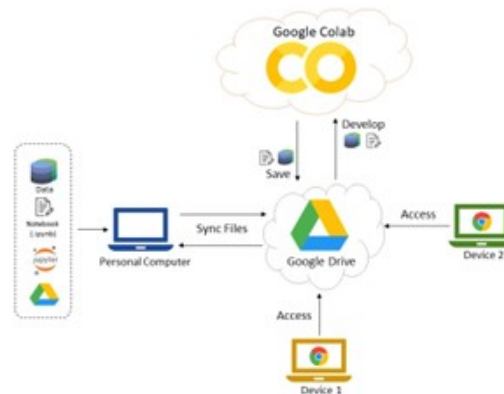


3-28. ábra. Jupyter Notebook App részei.

3.3.2. Google Colaboratory

A Google Colaboratory vagy röviden "Colab" a Google Research egyik terméke. A Colab bárki számára lehetővé teszi, hogy python-kódot írjon, illetve futtasson böngészőn

keresztül. Különösen alkalmas gépi tanulási, illetve adatelemzési programok készítésére. Technikai szempontból a Google Colab egy hosztolt Jupyter notebook szolgáltatás, aminek használata nem igényel beállítást, ám ingyenes hozzáférést biztosít a számítási erőforrásokhoz, ideértve a GPU-kat is. Ezen kívül több számítógépről is hozzáférést biztosít a jegyzetfüzetekhez és a hozzájuk tartozó összes fájlhoz, ami a Google Drive-fiókban van eltárolva. Emellett lehetőség van a projektekhez szükséges fájlok Drive-ra történő feltöltésére, ami a gyakori fájlbeolvasási folyamatokat jelentősen gyorsíthatja. Futtatásához egyedül egy böngészőre és internetkapcsolatra van szükség [59].



3-29. ábra. Google Colab működési struktúrája.

3.3.3. Docker

A Docker jelenleg az egyik legelterjedtebb container framework, amelynek implementációja a Linux kernel nyújtotta szeparációs lehetőségekre támaszkodik. Komoly előnye a sebesség. A technológiából fakadóan a Docker containerek kicsik illetve gyorsak, hiszen tulajdonképpen csak egy, a kernelen futó sandbox környezetről van szó, aminek kicsi az erőforrás igénye. A Docker alapvetően három részből áll: a daemon, a kliens, és egy REST API. A daemon a host gépen fut és parancsokat fogad a klienstől. Az API segítségével közvetlenül tudunk kommunikálhatunk a daemonnal. A containerek definiálására a dockerfile szolgál. Ez instrukciókat tartalmaz arra vonatkozóan, hogy hogyan épüljön fel a container. Ezekkel a parancsokkal például előírhatjuk az operációs rendszer verzióját, bizonyos Linux packagek installációját, portok megnyitását, és még rengeteg egyéb dolgot, mint esetemben a telepítendő python packageket [60]. A Docker konténernek egy Python 3.10- et kellett használni, mert korábbi tapasztalataim szerint ez az a python verzió, amivel minden AutoML komponens rendesen működik. A req.txt tartalmazza azokat a packageket, amelyeket telepíteni kell a kód futtatásához.

3.3.4. Microsoft Azure

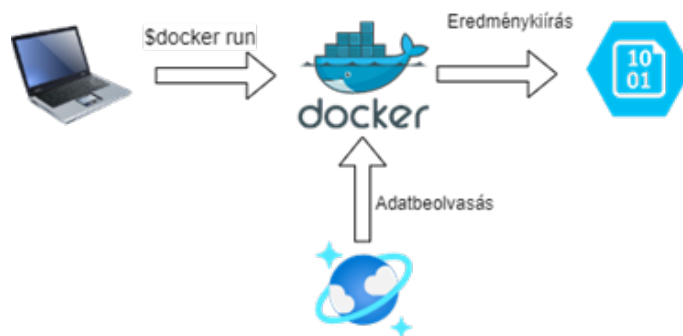
A Microsoft Azure egy olyan felhőszolgáltatási gyűjtemény, amely rendkívül sok eszközt biztosít a felhasználóknak a fejlesztéseikhez. Számomra a tárhelyek voltak a leginkább hasznosak, amiknek segítségével könnyebben tudtam az AutoML SKlearn 2.0 algoritmus tanítását Docker konténerben elvégezni. Az megvalósított struktúra a következő képen látható.

3.3.4.1. CosmosDB: Az Azure CosmosDB a Microsoft globálisan elosztott, több modelles adatbázis-szolgáltatása. Séma-agnosztikus és vízszintesen méretezhető. Belsőleg a CosmosDB az elemeket tárolókba rendezi. A tárolók adatbázisokba vannak csoportosítva, amelyek a tárolók feletti névterekhez hasonlítanak. A tárolók séma-agnosztikusak, ami annyit tesz, hogy a rendszer nem kényszeríti ki sémát az elemek hozzáadásakor. Alapértelmezés szerint az elemek minden mezője automatikusan indexelve van, ez jobb teljesítményt biztosít anélkül, hogy konkrét lekérdezési mintákat használnánk. A tárolók egyedi kulcs-kényszereket is használhatnak ha az adatok integritása ezt megkívánja [61]. A belső adatmodell a következőkön keresztül érhető el:

- egy szabadalmaztatott SQL API
- öt különböző kompatibilitási API, amik olyan végpontokat tesznek közzé, amik részben kompatibilisek a MongoDB, avGremlin, a Cassandra, az Azure Table Storage protokolljaival. Ezek az API-k lehetővé teszik bármely kompatibilis alkalmazás számára, hogy szabványos illesztőprogramokon vagy SDK-kon keresztül csatlakozzon a CosmosDB-hez.

3.3.4.2. Blob Storage: A Blob Storage a Microsoft Azure olyan funkciója, ami lehetővé teszi a fejlesztők számára, hogy strukturálatlan adatokat tároljanak egy felhőplatformon. Ezek az adatok ezután bárholnan elérhetők lesznek. Igen sok féle tartalmat lehet feltölteni: hangot, videót és szöveget. A blobok „tárolókba” vannak csoportosítva, amik felhasználói fiókokhoz vannak csatolva [62]. A blobokat .NET kód segítségével lehet módosítani. A blob-tárolás előnyeit (az objektumtárolás egy típusa) hasznosító alkalmazások a következők:

- Online médiamegosztás
- Média streamelés
- Elosztott fájl hozzáférés
- Naplófájlok tárolása és frissítése
- IoT (Internet of Things) alkalmazások



3-30. ábra. Az Azure szolgáltatás használatának logikai rendszerterve.

A blobok tárolókban vannak elhelyezve – ez a Microsoft Azure Storage-fiók alapvető tároló- egysége. A tárolók meghatározott fájl típusokat tartalmazhatnak – például az egyikben képeket, a másikon videofájlokat tárolhat a felhasználó. A konténerek olyanok mint a könyvtárszerkezeten belüli mappák. Három fő blob-fajta létezik: hozzáfűzés, blokkolás és oldalblob [62].

- **Blobok hozzáfűzése:** Ezeket kifejezetten hozzáfűzési műveletekhez tervezték. A hozzáfűzött blob leggyakrabban a naplófájlok tárolására illetve frissítésére szolgál. Az új blokkokat hozzá tudjuk fűzni egy blob végéhez, ugyanakkor a korábban már elkészített blokkokat nem tudjuk később módosítani vagy törölni.
- **Blobok blokkolása:** A blokkblobok főként médiafájlok, dokumentumok, szövegfájlok illetve bináris adatok tárolására szolgálnak.
- **Oldalblobok:** Az oldalblobok alkalmasak olvasási illetve írási műveletekre. Az oldalblobokat leggyakrabban az operációs rendszerek és a katasztrófa-helyreállítási adatok tárolására használjuk.

Az AutoML algoritmus egyik végeredményét, ami egy modelleket és azok paramétereit tartalmazó táblázat (Pipeline Profiler) sajnos Blob storage-ben nem sikerült megjeleníteni. Lementés után VS Code-ban sikerült megnyitnom és kiértékelnem.

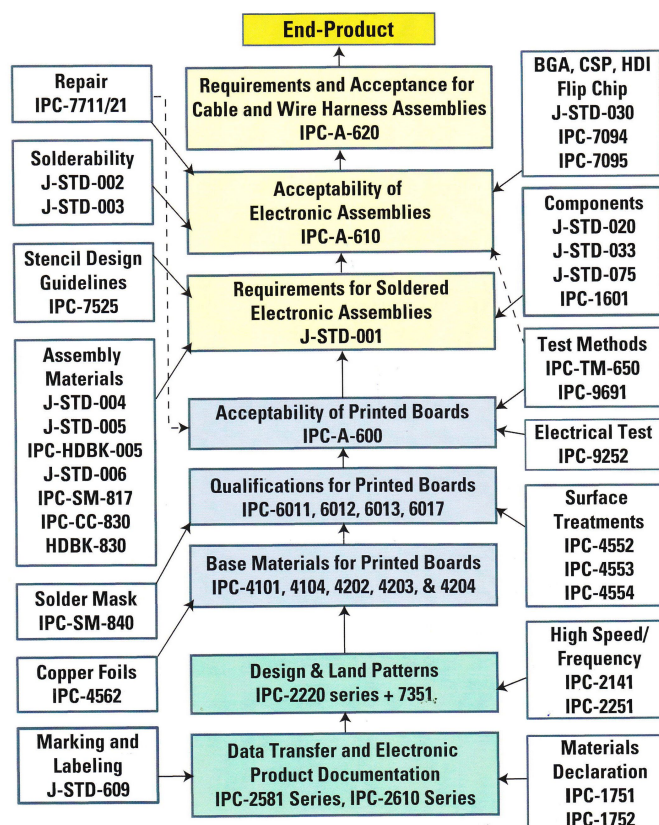
3.4. Fizikai eszköztervezés

Munkám részét képezte egy olyan eszköz megtervezése, ami képes a korábban említett fizikai paraméterek gyűjtésére. Ehhez az eszköznek tartalmaznia kell egy IMU-t és egy mikrofont. A jelenlegi periódusban még nem koncentráltam arra, hogy az készülék képes legyen neurális hálók futtatására, de a tervezés folyamán igyekeztem olyan elv szerint dolgozni, hogy később csupán a mikrokontroller lecserélésével alkalmas legyen MI futtatására is. Amennyiben a karpánt eléri kívánt szerepét és az inzulinpumpával kommunikálva fogja segíteni a precízebb inzulin adagolást, egészségügyi terméknek fog minősülni.

Emiatt teljesítenie kell az IPC szabvány kritériumait. Munkám során tanulmányoztam a szabványt és az általam megvalósított tervezési esetekben betartottam az előírt kritériumokat. Továbbá a felhasználni kívánt eszközök kiválasztása során is ezen elvek szerint jártam el. A beszerzett alkatrészeket teszteltem és a dolgozatban dokumentáltam, hogy milyen problémákba ütköztem, illetve milyen megoldási lehetőségeket javaslok azok megszüntetésére.

3.4.1. IPC szabvány

Az IPC (Institute for Printed Circuits) szabványt világszerte alkalmazzák elektronikai eszközgyártók mind a tervezés mind a gyártás folyamatában [63]. Előnye, hogy megkönnyíti a megrendelők és a gyártók közötti kommunikációt annak közös nyelvre hozása által, valamint javítja a termék minőségét és megbízhatóságát. Az IPC az eszközök minőségének garantálás miatt a teljes munkafolyamatra készített szabványokat, melyet az alábbi szabványfa szemléltet [64]. Az IPC mellett van egy másik szabványcsalád is, amelyet az orvostechikai eszközöket gyártó cégeknek és vállalatoknak nagyon szigorúan be kell tartani. Ez az ISO 13485:2016 szabvány, amely a teljes minőségirányítási rendszert átfogóan kezeli. Azok a szervezetek akik ezeket betartják, jogosultak orvostechikai eszközök tervezésére és fejlesztésére, gyártására, tárolására, forgalmazására, telepítésére illetve szervizelésére. Az általam használt IPC-2221A szabvány ennél sokkal kevésbé szigorú elveket határoz meg több szempont esetén.



3-31. ábra. IPC szabványfa [64].

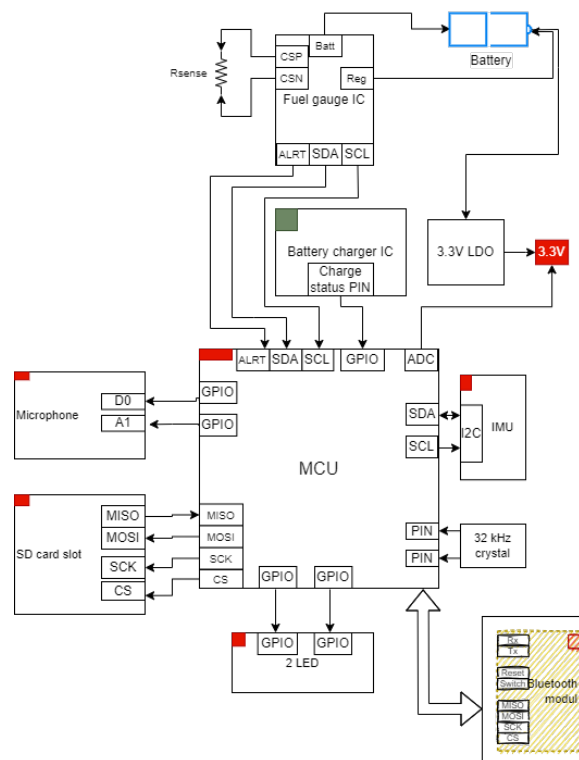
Jelen esetben számomra az IPC-2221A szabvány volt a legfontosabb, ami a nyomtatott áramkörök tervezésének elveit foglalja magában [65]. A legfontosabb szempontok az alábbiak:

- Anyagválasztás
- Anyagvastagság választás
- ESD védelem
- Strukturális felépítés
- Méret és elrendezési elvek
- Merevítések alkalmazása
- Impedancia illesztés
- Vezetősávok tervezése
- Buszok tervezése
- Viák kialakítása
- Hődisszipációs elvek
- Hővezetés
- Kényszerek és toleranciák

- Alkatrész elhelyezések
- Alkatrész választási elvek
- Tesztelhetőség
- Gyárthatóság

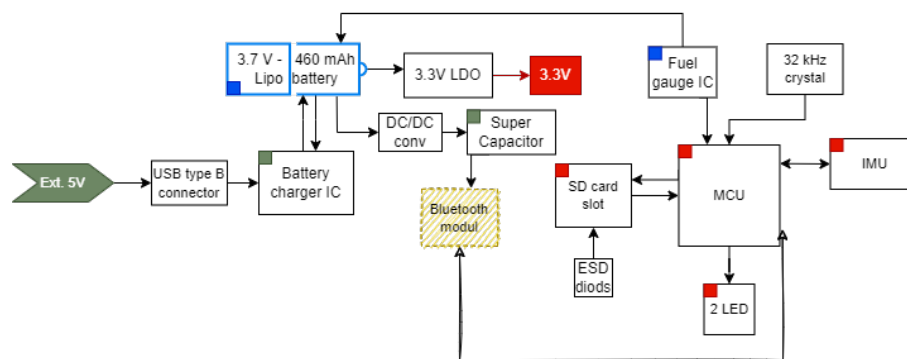
3.4.2. PCB tervezés

A tervezéshez az Altium Designer 19.1.8 tervezőprogramot használtam. A készített áramkör logikai rendszertervét szemlélteti az alábbi ábra.



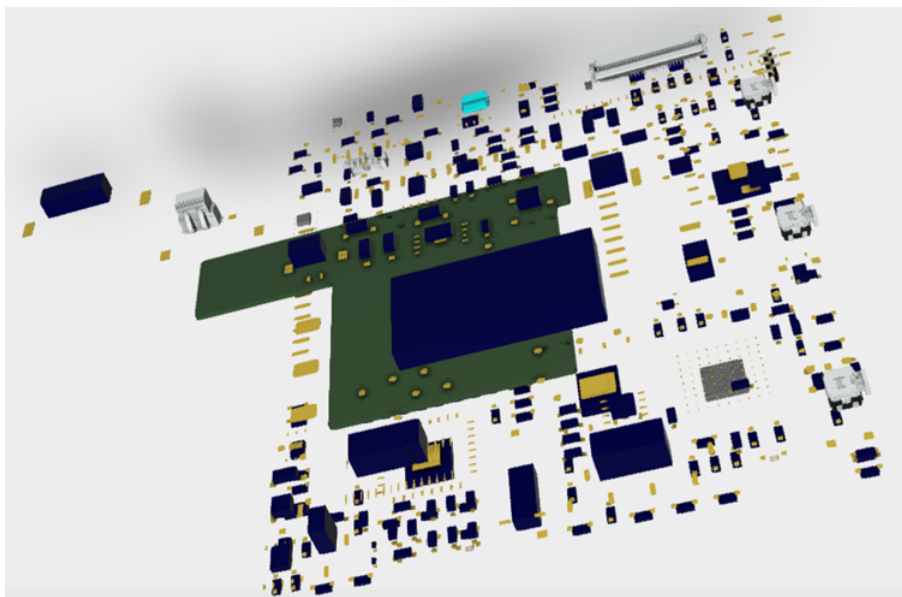
3-32. ábra. A fizikai eszköz ogikai rendszerterve.

A képen nem alkalmaztam busz jelölést, mert ebben a nézetben könnyebb átlátni a szükséges portok számát és típusait. A tápellátást 3,3 és 5V-os feszültségforrások biztosítják. A tesztpontok nincsenek jelölve az ábrán, de minden kommunikációs vezetékre illetve a különböző tápfeszültségekhez szükséges tenni egyet. A felhasználás és a tesztelés megkönny-nyítése miatt 2db LED-et helyeztem el a nyomtatott áramkörön. Az áramkör energiamenedzsmentjének tervét szemlélteti az alábbi ábra.



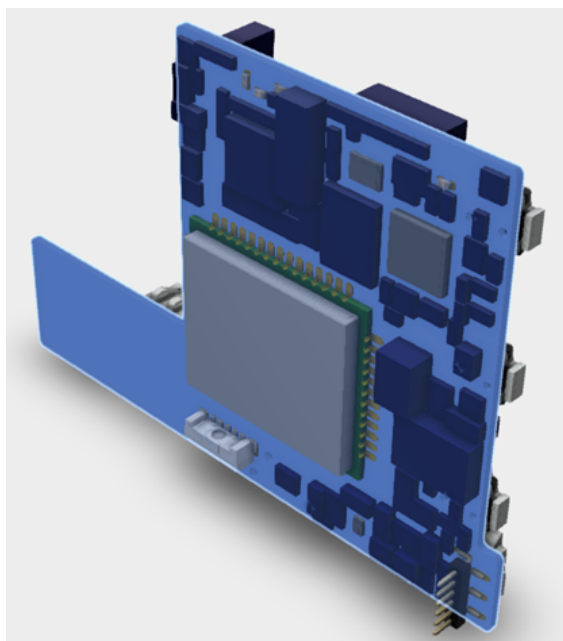
3-33. ábra. Energiamenedzsment tervrajz.

A real-time adatgyűjtés miatt felmerült bennem a Bluetooth kommunikáció alkalmazása. Viszont ennek a technológiának igen magas energiaigénye van a többi alkatrészhez képest, ami miatt kialakult az igény egy szuperkapacitás alkalmazására. Ez az eszköz sokkal több energiát képes tárolni mint egy átlagos kondenzátor, ugyanakkor kisebb helyen. A Cap-XX nevű cég által gyártott szuperkapacitás 5,5V-os feszültségen 1F elektromos kapacitást tud tárolni 21x14 mm-es tokban [66]. A nyomtatott áramkör 3D nézetének robbantott és egyszerű 3D nézetének ábrája látható alább.



3-34. ábra. A PCB robbantott ábrája.

A jelenleg megtervezett nyomtatott áramkör méretei 35mm x 48mm. A PCB alapja FR4-es anyag ami 0,35mm vastagságú. Ebből összesen 3 található. A kész áramkör összesen 4 réteget tartalmaz, melyek közül a középső kettő a tápfeszültséget és a földet foglalja magában, a szélső kettő pedig a jelátviteli réteg. Az IPC által meghatározott minimum vezetősáv szélesség 0,308mm a két vezeték közötti távolság pedig 0,64 mm.



3-35. ábra. A PCB 3D ábrája.

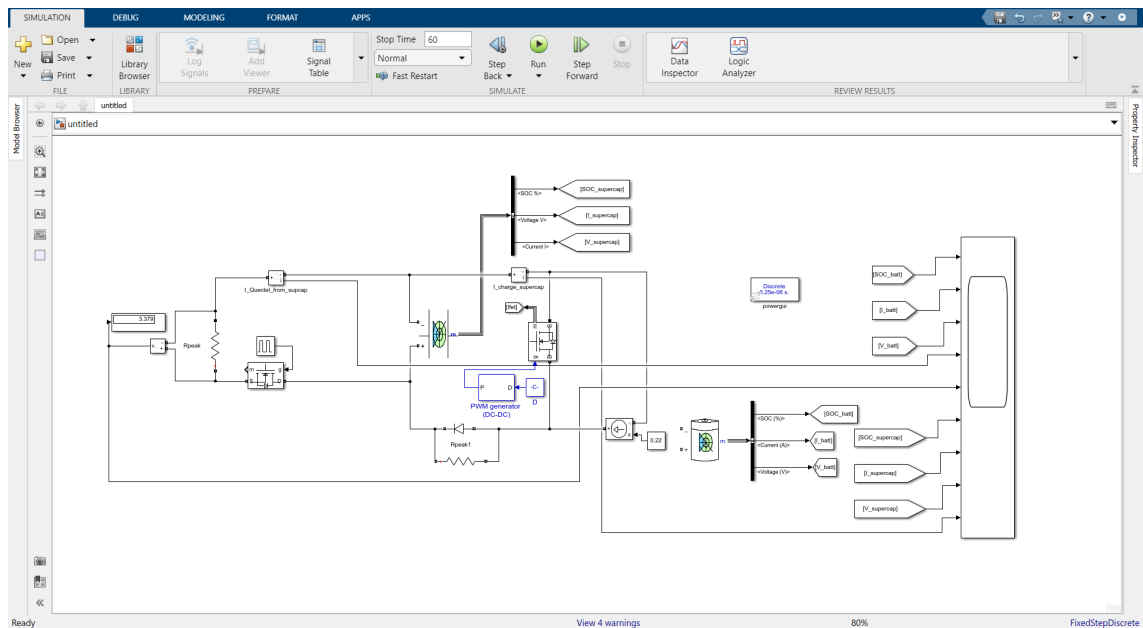
3.4.3. Alkatrész választás

Az alkatrészek esetében igyekeztem a nagyobb gyártók termékei közül választani, mert tapasztalataim szerint pontosabb gyártástechnológiával dolgoznak. Az IPC szabvány javaslata szerint igyekeztem Class 1 besorolású alkatrészeket választani. Ez azt jelenti, hogy kapacitások esetén a C0G osztályú kondenzátorokat részesítettem előnyben. Ezek szélesebb hőmérsékleti tartományban tudnak működni, nagyobb pontossággal, illetve használatukkal elkerülhető az „éneklő kondenzátor” hatás, ami a nagy frekvenciás jelek esetén a kondenzátor által kibocsátott sípoló hangot jelenti. A mikrofonnal történő mérés miatt ez igen fontos tényező. Az ellenállások tekintetében 0,1% pontosságú alkatrészeket választottam. A korábban használt MPU6050 helyett most a Bosch BMI270 IMU-t [67] választottam, ami jobb pontosságot szolgáltat és az adatlapja szerint alkalmas gesztusdetekciós eszközök fejlesztésére.

3.4.4. Tesztelés

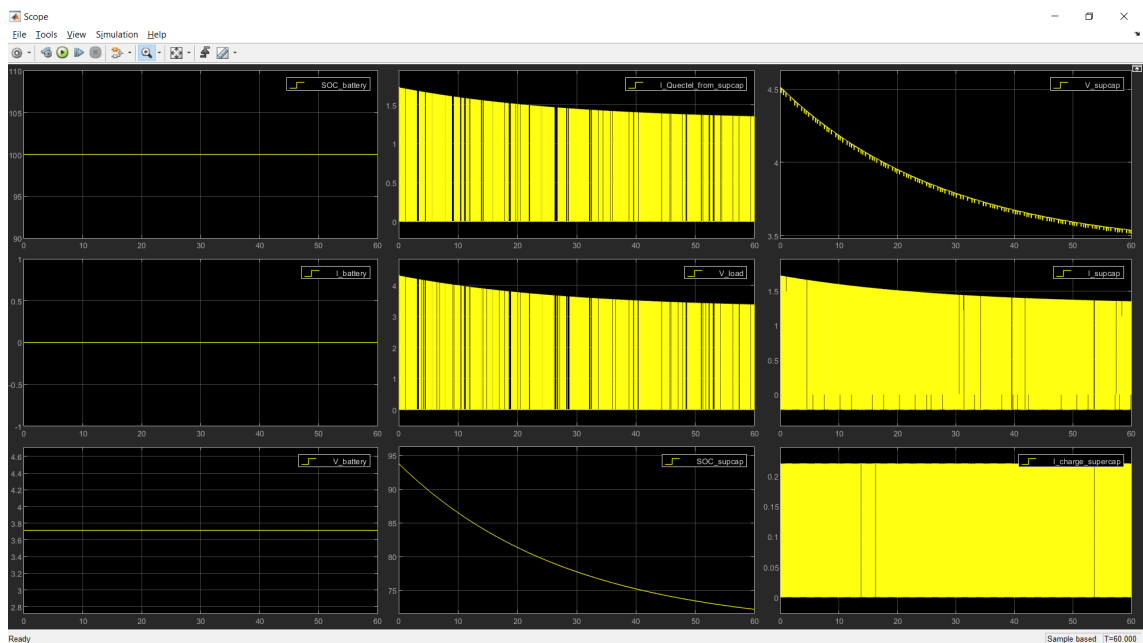
Az alkatrészek közül eddig a mikrofont, az IMU-t, egy STM32 mikrovezérlőt, illetve a tápellátás biztosításához szükséges darabokat szereztem be és teszteltem le. Problémával eddig egyedül a szuperkapacitás esetén találkoztam. A legtöbb Bluetooth modul használat közben 350-550 mA-es áramfelvétel csúcsokat produkál körülbelül másodpercenként kétszer. Ez egy egyszerű akkumulátor esetében nem tolerálható áramfelvétel, mert a cella idő előtti tönkremeneteléhez vezet. Emiatt kerülne beépítésre a szuperkapacitás. Viszont egy általam készített szimuláció alapján ez a 0,5 másodperc nem elég arra, hogy a szu-

perkapacitás feltöltőn és ki tudja szolgálni a Bluetooth modult. Nagyobb akkumulátor esetleg megoldást jelenthet, mert nagyobb áram leadására képes, ugyanakkor a méret is jelentősen megnövekedne. A szimulációt Matlab szoftverben készítettem el.

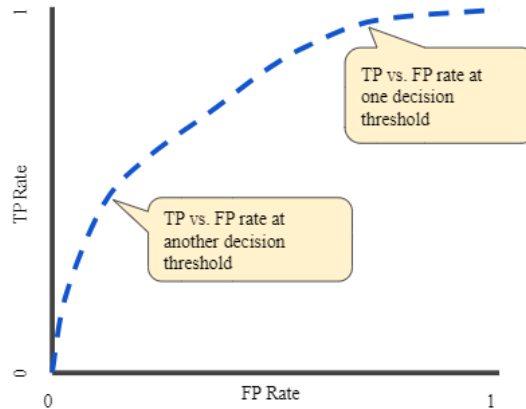


3-36. ábra. A szimuláció felépítése.

Az alábbi ábrán jól látható, ahogyan a szuperkapacitás feszültsége az idő előrehaladtával egyre csökken, ami miatt 1 perc után már nem tudja ellátni a feladatát.



3-37. ábra. 60 másodperces szimuláció eredménye.



4-39. ábra. ROC görbe.

4.2. Konfúziós mátrix

A konfúziós mátrix egy osztályozási probléma előrejelzési eredményeinek összefoglalását mutatja be. A helyes és hibás előrejelzések számát összegzi, és osztályonként lebontja. A konfúziós mátrix megmutatja, hogy az osztályozási modell milyen módon téved amikor előrejelzéseket készít. Nemcsak az osztályozó által elkövetett hibákba számába nyújt be- tekintést, hanem ami még fontosabb, az elkövetett hibák típusaiba is [68]. Az elkészítéséhez szükség van egy teszt/validációs halmazra a várható eredményekkel. A teszt/ validációs halmazon készíthetünk egy előrejelzést. A várható eredményekből és előrejelzésekből kiszámítható:

- A helyes előrejelzések száma az egyes osztályokhoz.
- A hibás előrejelzések száma az egyes osztályokhoz, a megjósolt osztály szerint.

Ezeket a számokat ezután táblázatba vagy mátrixba rendezzük az alábbiak szerint:

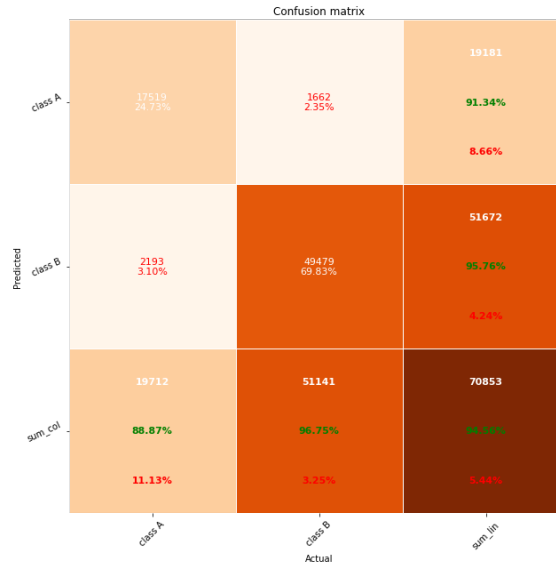
- **Várható:** A mátrix minden sora egy előre jelzett osztálynak felel meg.
- **Megjósolt:** A mátrix minden oszlopa egy aktuális osztálynak felel meg.

Az osztályra vonatkozó helyes előrejelzések száma az adott osztályérték várt sorába és az osztályérték előrejelzett oszlopába kerül. Ugyanígy az osztályra vonatkozó helytelen előrejelzések száma az adott osztályérték várt sorába és az osztályérték előrejelzett oszlopába kerül. A mátrixban az alábbi eshetőségeket találhatjuk:

- **”Igazi pozitív”:** a helyesen előrejelzett eseményértékek
- **”Hamis pozitív”:** a helytelenül előre jelzett eseményértékek

- **”Igaz negatív”**: a helyesen előre jelzett esemény nélküli értékek
- **”Hamis negatív”**: a helytelenül előre jelzett esemény nélküli értékek

A konfúziós mátrix értelmezése alapján egy osztályozó akkor a legjobb, ha csak mátrixa főátlójában található 0-tól eltérő érték.



4-40. ábra. Egy VGG16 háló eredményét bemutató konfúziós mátrix.

4.3. Precision, recall és accuracy

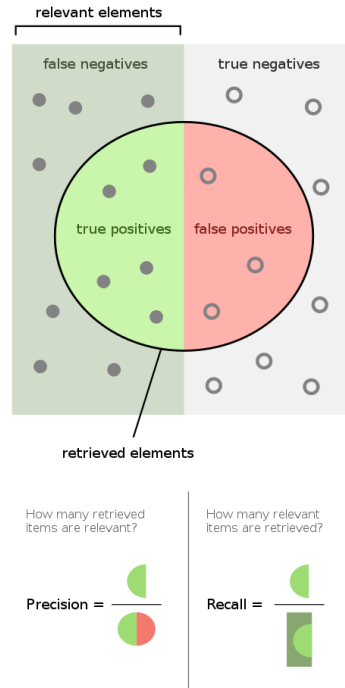
- **Precision**: A pozitívnak prediktált osztály értéke, azaz egy olyan arány amely megmondja, hogy a pozitívnak mondott érték mennyire biztosan pozitív és mennyire lehetne negatív [69].

$$Precision = \frac{TP}{TP + FN} \quad (4.1)$$

- **Recall**: Az igaz/valós pozitívok aránya, azaz ez a metrika mutatja, hogy az osztályozónk milyen arányban téveszti el a helyes osztályt és prediktál helyette helytelen osztályt [69].

$$Recall = \frac{TP}{TP + FP} \quad (4.2)$$

A recall és precision paraméterek szemléltetése az alábbi ábrán látható.



4-41. ábra. A recall és a precision szemléltetése.

- **Accuracy:** Az osztályozónk pontossága. Ez a metrika mutatja, hogy mennyire pontosan képes osztályozni az osztályozónk.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.3)$$

4.4. F1 score

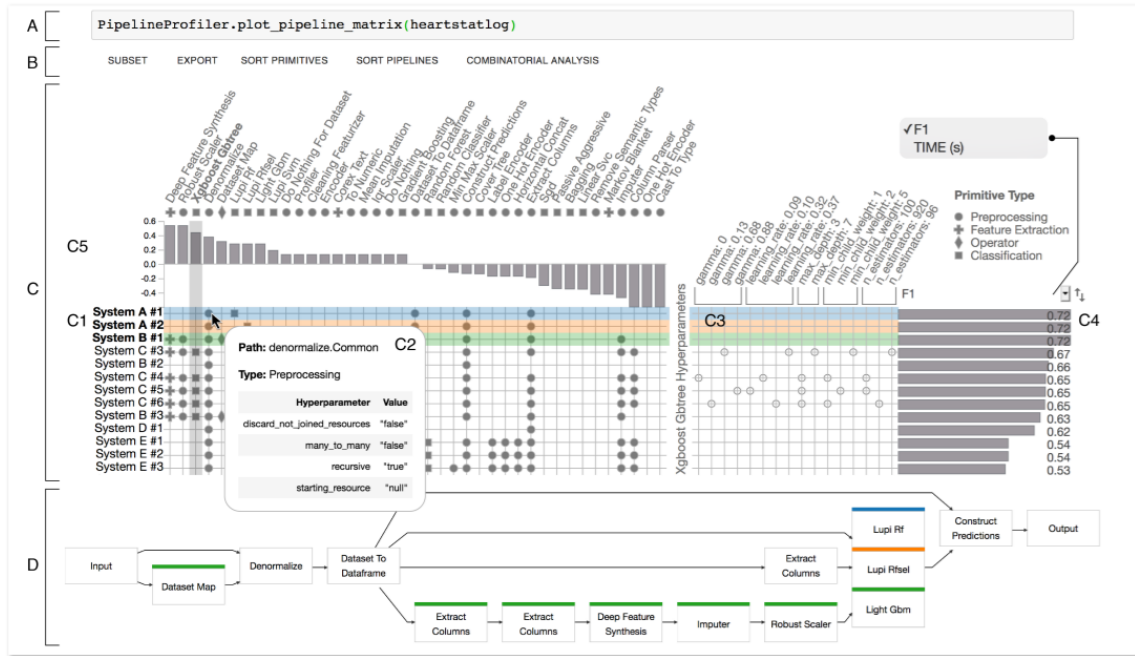
A recall és a precision közötti egyensúlyt mutatja meg. Akkor érdemes használni, ha nagyon sok negatív vagy pozitív mintánk van, mert ilyenkor előfordulhat, hogy a precision magas értéket ad, de közben maga az osztályozó rosszul működik [69].

$$F1score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (4.4)$$

4.5. Pipeline Profiler

Az Auto-Sklearn-ben az eredmények megjelenítésére egy alternatíva a PipelineProfiler használata. Ez egy vizuális elemző eszköz, ami vizuális reprezentációkat és interakciókat vezet be:

- Pipeline Mátrix, egy vizuális megjelenítés, amely hatékonyan összegzi az AutoML rendszerek által előállított folyamatokat, primitíveket és hiperparaméterértékeket
- Grafikonok összehasonlító nézete, amely grafikon-illesztést alkalmaz, hogy kiemelje a strukturális különbségeket és hasonlóságokat a fontos folyamatok között



4-42. ábra. A Pipeline Profiler eredmény megjelenítő.

Az előző ábra a PipelineProfiler megjelenítőt mutatja be. A fő összetevői a Pipeline Mátrix (C) és a Pipeline Comparison View (D). A Pipeline Mátrix (C) táblázatos összefoglalót jelenít meg a gyűjteményben található összes folyamatról. A felhasználó egy vagy több pipeline-t is részletesebben megnézhet – a kiválasztott pipeline grafikon szerkezete kattintásra megjelenik a Pipeline Comparison View (D) nézetben. A rendszer-menü (B) lehetővé teszi a felhasználó számára, hogy a folyamatok egy részhalmazára összpontosítson, bizonyos folyamatokat Pythonba exportáljon, rendezze a táblázat sorait és oszlopait, és automatizált elemzéseket hajtson végre primitív csoportokon [70].

5. Eredmények

A különböző modelleket számos tanító/teszthalmaz aránnyal illetve beállítással kipróbáltam. A korábbi munkám eredményeként már rendelkeztem 310.000 rekordot tartalmazó adathalmazzal, ami gyorsulás és elfordulás értékeket tartalmaz, később pedig 368 percnyi mérést rögzítettem. Ez a gyorsulási értékek szempontjából 1.656.733 rekordot jelent, a hangfeldolgozás esetében pedig 22.089 másodpercnyi mp3 fájlt.

5.1. VGG16

A neurális háló tanításához a korábban és az újonnan gyűjtött adatokat egyaránt felhasználtam. Ez összesen 1.966.733 rekordot jelent. Az általam beállított időablak 20 másodperc volt. Az eredményeket vizsgálva azt tapasztaltam, hogy a gyorsulási és szögelfordulási adatokat együttesen felhasználó konvolúciós neurális háló jobb eredményt szolgáltat mint a csak gyorsulási értékeken tanuló verziója. Ez a várakozásokkal egybevágó eredmény.

Szempont \ Felosztás	Önvalidáció	75/25	80/20
Accuracy	96,56	98,01	98,04
Precision	94,13	97,69	97,53
F1-score	89,82	90,12	98,70

5-1. táblázat. VGG16 CNN eredménye.

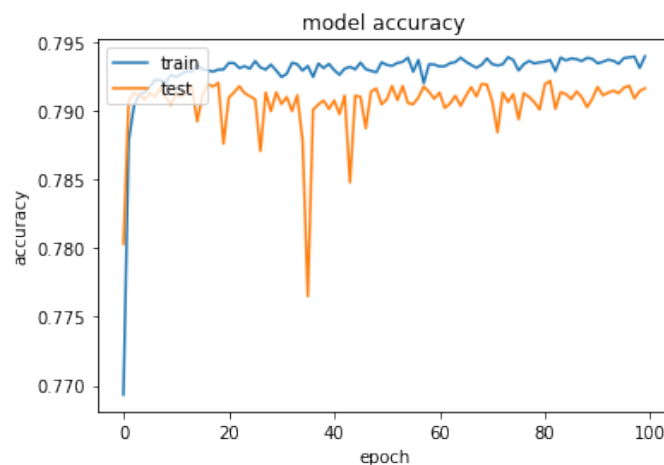
5.2. LSTM

Először a leggyakrabban használt 75/25 arányú train / teszt halmaz felosztást választottam és így tanítottam a neurális hálót. Később kipróbáltam a 80/20 arányt is, illetve teszteltem azt, hogy 3 teljes nap tanítási adatként való beadása esetén milyen pontossággal képes az algoritmus a 4. nap cselekvéseit osztályozni. Annak ellenőrzésére, hogy a neurális háló képes-e rátanulni a beküldött adatokra, egy önmagát validáló hálót hoztam létre. Ebben a tanítási és a tesztelési adathalmaz megegyezett. Az eredményeket a következő táblázat foglalja össze.

Szempont \ Felosztás	Önvalidáció	75/25	80/20	3nap/1nap
Pontosság	0.9567	0.9574	1.0000	0.7917
F1-score	0.9309	0.9298	1.0000	0.6828

5-2. táblázat. Pontosság és F1-score értékek az LSTM hálózat esetén

A táblázat első oszlopában láthatjuk, hogy a pontosság 95%-os volt önmaga validálása estében, ami azt jelenti, hogy a neurális háló rá tudott tanulni a bemenő adatokra. A 75/25 arány esetében a kisebb részmérésekből készített adathalmazzal végeztem tanítást, míg a 80/20 esetben a 4 hosszú nap adatait bontottam szét az említett arányban. Az utolsó oszlopban látható, napokra bontott tesztelés pontosságának értéke jelentősen eltér a többitől. Ennek oka az, hogy a teszt nap egy mérési hiba miatt rövidebb a többinél, illetve egy olyan eseményt is tartalmaz a mérés, amit a háló rosszul ismert fel. Az időbélyegek alapján történő azonosítás szerint ez az esemény egy pohár kávé elfogyasztása volt séta közben. Az első fajú hiba bekövetkezése várható volt ebben az esetben, hiszen étkezést eddig csak ülő, vagy állá helyzetben rögzítettem, így az adathalmazban hasonló esemény még nem volt. A következő ábrán látható, hogy a lefuttatott 100 epoch során hogyan változott a modell pontossága, illetve milyen volt a teszhalmaz pontossága.



5-43. ábra. A 3nap/1nap train/test arányú neurális háló pontosság függvénye.

5.3. AutoML-SKlearn 1.0

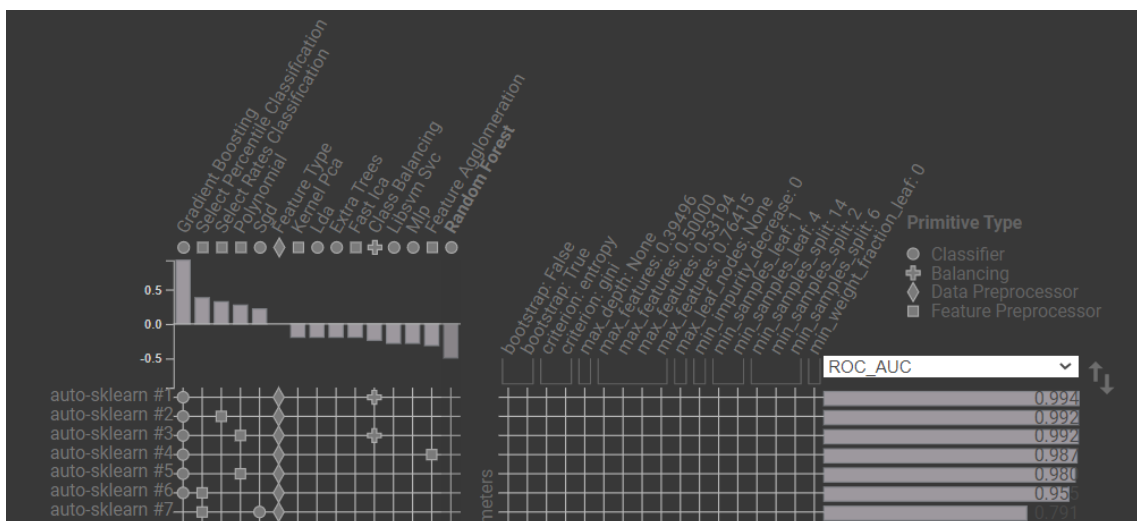
Mind az AutoML-SKlearn 1.0 és 2.0 esetében is teszteket hajtottam végre a filterezés nélküli, 0-1 közé normalizált adatokon és a 4 különböző szűrővel tisztított adatokon egyaránt. Mindkét algoritmusnál az IIR filterrel manipulált adatok adták a legjobb eredményt. Ez az SKlearn 1.0 esetében 96,55%-os pontosság volt. Ezek az értékek természetesen csak akkor értelmezhetők, ha tudjuk hogy pontosan milyen modell is áll mögöttük. Ebben is kaptam eredményt az SKlearn-től. Az SKlearn 1.0 típusú AutoML esetében a Gradient Boosting gépi tanulási technika volt a javasolt algoritmus. A Gradient Boosting működése egy kezdeti modell (például fa vagy lineáris regresszió) illesztésével kezdődik. Ezután egy második modell készül, amely az első modell gyengén teljesítő eseteinek előrejelzésére összpontosít. A két modell kombinációja várhatóan jobb lesz, mint bármelyik modell önmagában. Ezután többször megismétli ezt a hatékonyság fokozási folyamatot. Minden egymást követő modell megpróbálja kijavítani az összes korábbi

modell együttesének hiányosságait. A többi növelési módszerhez hasonlóan a gradiens növelés is iteratív módon egyesíti a gyenge "tanulókat" egyetlen erős tanulóvá [71]. A cél az, hogy találjak egy olyan függvényt, amely a legjobban közelíti a kimeneti változót a bemeneti értékek alapján. Ezt normalizálok valamilyen veszteségfüggvény bevezetésével és minimalizálásával.

Szempont \ Felosztás	Nem filterezett	Savgol	Wiener	IIR	Medián
Pontosság	0.8799	0.9433	0.9704	0.9655	0.9447
F1-score	0.9102	0.9549	0.9758	0.9786	0.9534

5-3. táblázat. Az SKlearn 1.0 eredménye

A következő képen láthatjuk az SKlearn 1.0 algoritmus kiértékeléséhez használt pipeline profiler vizuális elemző eszközt. A legjobb értéket a bal oldalon lévő Gradient Boosting algoritmus érte el. Jobb oldalt dőlten az egyes paraméterek javasolt beállításait tüntetjük fel.



5-44. ábra. Az SKlearn 1.0 pipeline profiler kiértékelési felülete.

5.4. AutoML-SKlearn 2.0

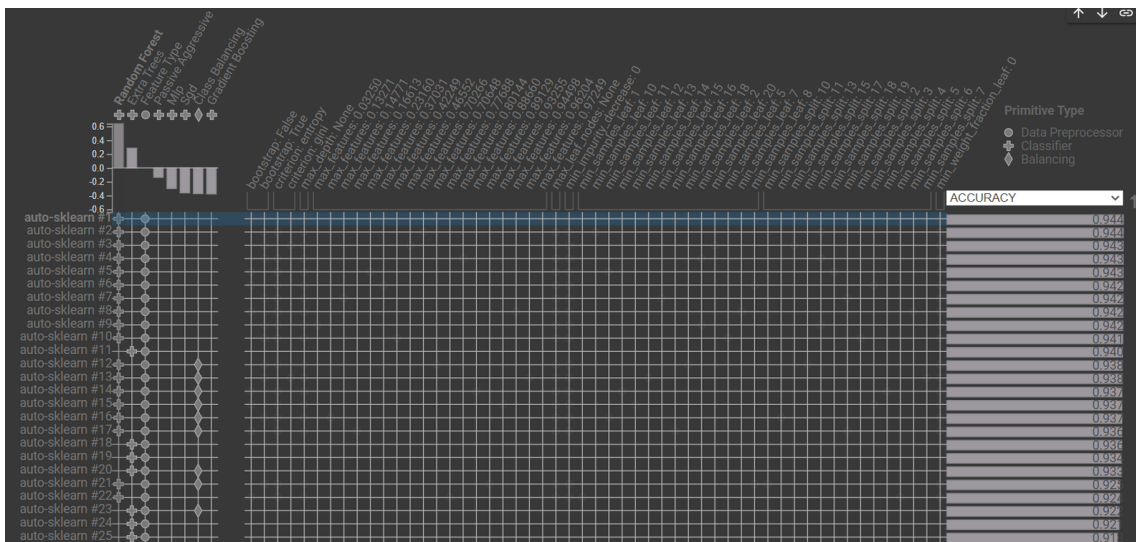
Az SKlearn 2.0 esetén a Random Forest-et kaptam eredményül. Ez egy olyan felügyelt gépi tanulási módszer, ami több döntési fából épül fel. Több osztályozót ötvöz magében, emiatt összetettebb problémákra is tud megoldást nyújtani. Az algoritmus működése során egy erdőt generál, amit bootstrap aggregálással tanít be. Az eredményt végül a döntési fák által adott előrejelzések alapján fogja meghatározni. Az eredmény annál pontosabb, minél nagyobb az erdő. Ezt az algoritmust korábbi munkám során már megvizsgáltam, akkor még számottevően kisebb adathalmazzal. Akkor 88.1%-os pontosságot

sikerült vele elérni. A jelenleg rendelkezésre álló mérési adatokkal az SKlearn 2.0 szerint 98,76%-os pontosságot is el lehet érni.

Szempont \ Felosztás	Nem filterezett	Savgol	Wiener	IIR	Medián
Pontosság	0.9089	0.9456	0.9621	0.9822	0.9187
F1-score	0.9134	0.9675	0.9737	0.9876	0.9210

5-4. táblázat. Az SKlearn 2.0 eredménye

A következő képen látható az Sklearn 2.0 eredményének elemzéséhez használt pipeline profiler elemző eszköz. A legjobb eredményt a bal oldalon látható Random Forest kapta. Jobb oldalt dőlten az egyes paraméterek javasolt beállításait láthatjuk.



5-45. ábra. Az SKlearn 2.0 pipeline profiler kiértékelő felülete.

Az SKlearn 2.0 esetében végeztem olyan tanítást, amiben felhasználtam mind a frissen mért 31 étkezés adatait, mind a korábbi dataset értékeit.

Szempont \ Felosztás	Önvalidáció	75/25	80/20
Accuracy	88,50	95,19	94,88
Precision	87,89	94,00	94,14
F1-score	86,34	94,32	93,76

5-5. táblázat. AutoML SKlearn 2.0 eredménye.

Az remélt értékekkel sajnos ellenkező eredményt kaptam a vizsgálati szempontok értelmezése után. A szögelfordulási értékeket is felhasználó SKlearn 2.0 algoritmus minden tanító és tesztalmaz arány esetében rosszabbul teljesített mint a csak gyorsulási értékeket

felhasználó verziója. A javasolt algoritmus jelen esetben is a Random Forest volt. Véleményem szerint további algoritmusok kipróbálása is szükséges, illetve az idő rövideje miatt nem volt időm az IIR filteren kívül más szűrők által generált adathalmazokon is futtatni az algoritmust.

5.5. DS-CNN

A hangok feldolgozására használt neurális háló tanításához csak az újonnan rögzített adathalmazt használtam. A hangfelvételeket 1 másodperces ablakokra bontottam, mert számításaim alapján a számomra fontos hangeffekteket egy 1 másodperces időablakba beleférnek. Az így kapott kereteken elvégzett megfelelő transzformációk után a háló bemenetére kerültek az adatok.

Szempont \ Felosztás	Önvalidáció	75/25	80/20
Accuracy	90,32	84,23	86,49
Precision	88,82	81,09	86,33
F1-score	89,33	83,76	85,17

5-6. táblázat. DS-CNN eredménye.

A kapott eredmények alapján kijelenthetem, hogy a neurális háló sikereresen rátanult az adat- halmazra és megtalálja a releváns hangeffekteket. A pontosság sajnos nem éri el a 87%-ot, ami miatt a háló rosszabbul teljesít mint a mozgási adatokat feldolgozó algoritmusok. Véleményem szerint további adatok gyűjtésével és az adathalmaz bővítésével illetve a hanghatások hosszához optimalizált ablakmérettel jobb hatásfok érhető el.

5.6. VGG16 + DS-CNN

A háló tanítására egyaránt használtam a régi és az új adatbázist, viszont a teszteléshez kizárólag azt az adathalmazt használtam, ami mozgás és hang adatokat is tartalmaz. Ez biztosította a lehetőségét a hang alapú és a mozgás alapú étkezés detekció együttes működésének vizsgálatára.

Szempont \ Felosztás	80/20
Accuracy	97,38
Precision	97,11
F1-score	93,57

5-7. táblázat. VGG16 és DS-CNN kombinációjának eredménye.

A tesztelés után arra a következtetésre jutottam, hogy a két neurális háló együttesen rosszabbul teljesít mint a VGG16 konvolúciós algoritmus önmagában. Ennek okát abban

vélem felfedezni, hogy a hangfelismerés még nem működik elég pontosan és ez lerontja a másik háló végeredményét.

5.7. SKlearn 2.0 + DS-CNN

A háló tanítására az előző esethez hasonlóan mindkét adathalmazt felhasználtam, viszont tesztelni csak az új dataseten teszteltem. Egyedül így lehetett vizsgálni azt, hogy a hang és mozgás felismerés is működik egy időben.

Szempont \ Felosztás	80/20
Accuracy	93,19
Precision	91,98
F1-score	91,02

5-8. táblázat. SKlearn 2.0 és DS-CNN kombinációjának eredménye.

Az eredmény a másik hálókombinációhoz hasonló. A két háló együttes működése minden vizsgált paraméter esetén rosszabb eredményt hozott mint az SKlearn 2.0 önmagában. Ennek oka itt is abban rejlik, hogy a pontatlanul működő hangfelismerési algoritmus lerontja az automatizált gépi tanulási folyamat kimenetét.

6. Jövőbeni tervek

A projektet véleményem szerint mindenképp érdemes tovább folytatni. A jövőben további hang- adatok gyűjtését fogom végezni, hogy a DS-CNN neurális hálóval jobb eredményeket tudjak elérni, illetve más hangok feldolgozására alkalmas algoritmusokat is ki fogok próbálni. Először egy rekurrens AI lesz a tesztelendő algoritmus, mert a szakirodalomkutatás során több esetben olvastam az által szolgáltatott pozitív eredményekről. Reményeim szerint ha a hangfelismerés megbízhatóbban működik, akkor a kombinált neurális hálók is jobb hatásfokot fognak elérni. Az eszköz tervezése szempontjából a PCB jelenleg gyártás alatt van, de a dolgozat leadásának időpontjáig nem érkezett meg, ezért nem szerepel róla kép a dokumentumban. Az energiaellátás problémáját egy nagyobb energiatároló képességű akkumulátorral fogom orvosolni, vagy pedig olyan mikrovezérlőt választok ami a kívánt igények mellett rendelkezik beépített Bluetooth modullal, különben a Bluetooth kommunikáció nem valósul meg. Amint minden eszközt leteszteltem, a NYÁK beültetése lesz a következő lépés illetve annak tesztelése.

7. Összefoglalás

Munkám során adatgyűjtést végeztem és az első periódusban gyűjtött 310.000 rekordból álló gyorsulás és szögelfordulási adatokat tartalmazó adathalmazt később kiegészítettem további 1.656.733 adatsorral, ami 368 pernyi mérésnek felel meg. Tanítást végeztem módosított VGG16, LSTM, AutoML SKlearn 1.0 és AutoML SKlearn 2.0 algoritmusokkal, illetve kipróbáltam a DS-CNN MI algoritmust, amivel az evőeszközök és tányérok egymáshoz ütődésének hangját valamint fémdobozos üdítőitalok kibontásának hangját ismerttettem fel. Az LSTM esetében a legjobb eredmény 95%-os pontosság volt, VGG16 esetén 98%-os pontosságot értem el, az SKlearn 1.0-nál 96,55%, az SKlearn 2.0 esetén ez a paraméter 94% volt. A hangfelismerő algoritmusom 86%-os pontossággal működött. Kipróbáltam a különböző mozgás és hangfelismerő algoritmusok kombinációját, aminek eredménye sajnos elmarad az elvártaktól. Ennek oka a hangfelismerő háló pontatlansága volt. Kutatásom folyamán rengeteg szakirodalmat áttekintettem a mesterséges intelligencia alapú gesztusfelismerésről illetve megismerkedtem behatóbban az IPC-2221A szabvánnyal. Ez alapján PCB-t terveztem és küldtem gyártásba. Bosch BHI270 IMU-t illetve Keyes KY-038 mikrofon modult teszteltem, valamint a Bluetooth alkatrész működtetéséhez szükséges energiafogyasztásról Matlab szimulációt készítettem. Emellett az energia menedzsment áramkört leteszteltem. Munkámat a jövőben a továbbfejlesztési lehetőségek részen belül ismertetekkel fogom folytatni.

8. Summary

In the course of my work, I collected data and later supplemented the data set containing 310,000 records of acceleration and angular rotation data collected in the first period with another 1,656,733 data lines, which corresponds to 368 measurements of seconds. I taught with modified VGG16, LSTM, AutoML SKlearn 1.0 and AutoML SKlearn 2.0 algorithms, and I tried the DS-CNN MI algorithm, with which I explained the sound of cutlery and plates hitting each other and the sound of opening soft drinks in metal cans. For LSTM, the best result was 95% accuracy, for VGG16 I achieved 98% accuracy, for SKlearn 1.0 it was 96.55%, for SKlearn 2.0 this parameter was 94%. My voice recognition algorithm worked with 86% accuracy. I tried a combination of different motion and voice recognition algorithms, the results of which unfortunately fell short of expectations. The reason for this was the inaccuracy of the voice recognition network. In the course of my research, I reviewed a lot of literature on artificial intelligence-based gesture recognition and became more familiar with the IPC-2221A standard. Based on this, I designed a PCB and sent it to production. I tested the Bosch BHI270 IMU and the Keyes KY-038 microphone module, and I made a Matlab simulation of the energy consumption required to operate the Bluetooth component. I also tested the energy management circuit. In the future, I will continue my work with what is described in the further development opportunities section.

Hivatkozások

- [1] WHO. “Diabetes”. (2021). URL: <https://www.who.int/news-room/fact-sheets/detail/diabetes>.
- [2] Medical Online. “A cukorbetegség nem csak az idősök betegsége”. (2019). URL: http://medicalonline.hu/gyogytas/cikk/a_cukorbetegseg_nem_csak_az_idosek_betegsege.
- [3] X. Lin és mtsai. “Global, regional, and national burden and trend of diabetes in 195 countries and territories: an analysis from 1990 to 2025”. (). URL: <https://www.nature.com/articles/s41598-020-71908-9> (elérés dátuma 2020).
- [4] Prof. Dr. Wittmann István. “Diabetológia jegyzet orvostanhallgatók számára”. (2014). URL: <https://kk.ptt.hu/hu/download/index/18362>.
- [5] Dr. Gerő László. “A pancreatogen diabates okai, tünettana és kezelése”. (). URL: <http://www.diabet.hu/folyoiratcikk.aspx?articleid=2505>.
- [6] Wikipedia. “Blood sugar level”. (). URL: https://en.wikipedia.org/wiki/Blood_sugar_level#cite_note-1 (elérés dátuma 2022).
- [7] Dr. Pánczél Pál. “A cukorbetegség etiológiája és patogenezise”. (2017). URL: <https://semmelweis.hu/belgyogyaszat3/files/2017/05/A-diabetes-mellitus-etiol%5C%c3%5C%b3gi%5C%c3%5C%a1ja-%5C%c3%5C%a9s-patogenezise.pdf>.
- [8] Scientific Reports. “Artificial Pancreas, Injection Sites, and Fast-Acting Insulin: Diabetes Trends and Challenges”. (). URL: <https://www.mddionline.com/components/artificial-pancreas-injection-sites-and-fast-acting-insulin-diabetes-trends-and> (elérés dátuma 2022).
- [9] Cukorbetegközpont. “Mody- a cukorbetegség egy ritka fajtája”. (2019). URL: <https://www.cukorbetegkozpont.hu/mody-a-cukorbetegseg-egy-ritka-fajtaja>.
- [10] Magyar Diabetes Társaság. “A diabétesz és a terhesség”. (2011). URL: http://www.diabet.hu/info.aspx?web_id=&sp=179.
- [11] Dr.Diag. “A pancreatogen diabates okai, tünettana és kezelése”. (). URL: <http://www.drdiag.hu/kereso/diagnosztika.adatlap.php?id=90393>.
- [12] Diabetes Education Online. “Types of Insulin”. (). URL: <https://dtc.ucsf.edu/types-of-diabetes/type2/treatment-of-type-2-diabetes/medications-and-therapies/type-2-insulin-rx/types-of-insulin/>.

- [13] Diabetes Teaching Center at the University of California. “Types of Insulin”. (). URL: <https://dtc.ucsf.edu/types-of-diabetes/type2/treatment-of-type-2-diabetes/medications-and-therapies/type-2-insulin-rx/types-of-insulin/> (elérés dátuma 2022).
- [14] Ilka Gdanietz. “Postprandial Blood Glucose - Why it’s important after eating”. (2018). URL: <https://www.mysugr.com/en/blog/postprandial-glucose-why-your-blood-sugar-is-so-important-after-eating/>.
- [15] Medtronic. “Insulin pump therapy”. (). URL: <https://www.medtronic-diabetes.com/en-MENA/about-diabetes/insulin-pump-therapy> (elérés dátuma 2022).
- [16] Mufti Adil Farooki MD; Shaykh Mateen Khan MD. “Insulin Pumps and CGM Devices: Islamic Permissibility of Use During Ghushl”. (). URL: <https://muslimmed.org/2021/12/22/insulin-pumps-and-cgm-devices-islamic-permissibility-of-use-during-ghushl> (elérés dátuma 2021).
- [17] Michael Middleton. “Deep Learning vs. Machine Learning — What’s the Difference?”. (2021). URL: <https://flatironschool.com/blog/deep-learning-vs-machine-learning/>.
- [18] IBM Cloud Education. “What is Strong AI?”. (2020). URL: <https://www.ibm.com/cloud/learn/strong-ai>.
- [19] S. Swaminathan. “Logistic Regression — Detailed Overview”. (2019). URL: <https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc> (elérés dátuma 2021).
- [20] T. Yiu. “Understanding Random Forest”. (2021). URL: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2> (elérés dátuma 2022).
- [21] ScienceDirect Topics. “Hidden Markov Models - an overview”. (2019). URL: <https://www.sciencedirect.com/topics/mathematics/hidden-markov-models> (elérés dátuma 2021).
- [22] C. Cortes és V. Vapnik. “Support-vector networks”. (1995). DOI: 10.1007/BF00994018. URL: [Mach%20Learn,%20k%3%B6t.%2020,%20sz.%203,%20o.%20273%E2%80%9393297](#) (elérés dátuma 2021).
- [23] Y. Jung; S. Lee; és Y. Jung J. Cho. “VLSI Implementation of Restricted Coulomb Energy Neural Network with Improved Learning Scheme”. (2019). DOI: 10.3390/electronics8050563. URL: [Electronics,%20k%3%B6t.%208,%20o.%20563](#) (elérés dátuma 2022).

- [24] C. Bento. “Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis”. (2021). URL: <https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141> (elérés dátuma 2022).
- [25] Karen Simonyan; Andrew Zisserman. “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION”. (2015). URL: <https://arxiv.org/pdf/1409.1556.pdf>.
- [26] Gaudenz Boesch. “TensorFlow Lite – Real-Time Computer Vision on Edge Devices (2022)”. (2022). URL: <https://viso.ai/edge-ai/tensorflow-lite/>.
- [27] MaBisz. “Így javíthatja a mesterséges intelligencia az egészségügyi ellátást”. (). URL: <https://mabisz.hu/szemle/?p=40334> (elérés dátuma 2022).
- [28] J. R. Fielding és mtsai. “Choosing a Specialty in Medicine: Female Medical Students and Radiology”. *American Journal of Roentgenology* 188 (2007), 897–900. old. DOI: [10.2214/AJR.06.0539](https://doi.org/10.2214/AJR.06.0539).
- [29] A. Hosny ; C. Parmar ; J. Quackenbush ; L. H. Schwartz és H. J. W. L. Aerts. “Artificial intelligence in radiology”. *Nat Rev Cancer* 18 (2018), 500–510. old. DOI: [10.1038/s41568-018-0016-5](https://doi.org/10.1038/s41568-018-0016-5).
- [30] Magyar Egészségügyi Menedzsment Társaság. “Biztonságos-e a mesterséges intelligencia alkalmazása az egészségügyben?”. (). URL: <https://memt.hu/biztonsagos-e-a-mesterseges-intelligencia-alkalmazasa-az-egeszsegugyben/> (elérés dátuma 2022).
- [31] Kim Minwoo; Jaechan Cho; Seongjoo Lee és Yunho Jung. *IMU Sensor-Based Hand Gesture Recognition for Human-Machine Interfaces*. Sensors, 2019. DOI: <https://doi.org/10.3390/s19183827>.
- [32] Christoph Amma Marcus Georgi és Tanja Schultz. *Recognizing Hand and Finger Gestures with IMU based Motion and EMG based Muscle Activity Sensing*. Karlsruhe, Germany: Cognitive Systems Lab, Institute for Anthropomatics és Robotics, Karlsruhe Institute of Technology, 2015.
- [33] Jeffrey L. Elman. “Finding Structure in Time”. (1990). URL: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1.
- [34] Meinard Müller. “Dynamic Time Warping. In: Information Retrieval for Music and Motion. Springer, Berlin, Heidelberg”. (2007). DOI: [10.1007/978-3-540-74048-3_4](https://doi.org/10.1007/978-3-540-74048-3_4). URL: https://link.springer.com/chapter/10.1007/978-3-540-74048-3_4#citeas.

- [35] N. Siddiqui és R. H. M. Chan. *Multimodal hand gesture recognition using single IMU and acoustic measurements at wrist*. PLOS ONE, Januar. 2020. DOI: <https://doi.org/10.1371/journal.pone.0227039>.
- [36] Severin D. Dobrea; és M.-C. Dobrea. “Head Gesture Recognition using a 6DOF Inertial IMU”. (2020). DOI: [10.15837/ijccc.2020.3.3856](https://doi.org/10.15837/ijccc.2020.3.3856). URL: <http://univagora.ro/jour/index.php/ijccc/article/view/3856>.
- [37] Georgi Marcus; Christoph Amma; és Tanja Schultz. “Recognizing Hand and Finger Gestures with IMU based Motion and EMG based Muscle Activity Sensing”. (2015). DOI: [10.5220/0005276900990108](https://doi.org/10.5220/0005276900990108). URL: <https://www.scitepress.org/papers/2015/52769/52769.pdf>.
- [38] MbientLab. “METAMOTIONS”. (2022). URL: <https://mbientlab.com/metamotions/>.
- [39] Mbientlab. “MetamotionS description”. (2021). URL: <https://mbientlab.com/metamotions/>.
- [40] Panoramic Digital Health. “Panoramic Digital Wearable Health Sensors”. (2021). URL: <https://www.panoramicdigitalhealth.com/what-we-do.html>.
- [41] TDK. “MPU-6050 — TDK”. (2021). URL: <https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/>.
- [42] Electronic Clinic. “KY-038 Sound Sensor with Arduino Circuit Code, Clap project”. (2021). URL: <https://www.electronicclinic.com/ky-038-sound-sensor-with-arduino-circuit-code-clap-project/> (elérés dátuma 2022).
- [43] József Fellegi. *Digital signal processing I*. Székesfehérvár: BUDAPEST TECHNICAL COLLEGE KÁLMÁN KANDÓ FACULTY OF COMPUTER ENGINEERING, FACULTY OF ELECTRICAL ENGINEERING, 2004.
- [44] Norbert Wiener. *The interpolation, extrapolation and smoothing of stationary time series*. Cambridge: MIT Press, 1942.
- [45] Arnold Pető. *FAN NOISE REDUCTION*. Budapest: Budapest University of Technology et al., 2012.
- [46] Wikipedia. “Savitzky–Golay filter”. (September. 2021). URL: https://en.wikipedia.org/w/index.php?title=Savitzky%E2%80%5C%93Golay%20%5C_filter%5C&oldid%5C=1041940665.
- [47] L. Roberts. “Understanding the Mel Spectrogram”, Analytics Vidhya”. (2021). URL: <https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53> (elérés dátuma 2022).

- [48] Wikipedia. “Mel scale”. (2021). URL: https://en.wikipedia.org/w/index.php?title=Mel_scale&oldid=1117532023 (elérés dátuma 2022).
- [49] GeeksforGeeks. “VGG-16 — CNN model”. (2020). URL: <https://www.geeksforgeeks.org/vgg-16-cnn-model/>.
- [50] és T. May P. M. Sørensen B. Epp. “A depthwise separable convolutional neural network for keyword spotting on an embedded system”. (2021). URL: <https://asmp-urasipjournals.springeropen.com/articles/10.1186/s13636-020-00176-2> (elérés dátuma 2022).
- [51] Xuan Hien Le; Sungho Jung; Giha Lee. “Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting”. (2019). DOI: 10.3390/w11071387. URL: https://www.researchgate.net/publication/334268507_Application_of_Long_Short-Term_Memory_LSTM_Neural_Network_for_Flood_Forecasting.
- [52] Heiga Zen; Hasim Sak. “UNIDIRECTIONAL LONG SHORT-TERM MEMORY RECURRENT NEURAL NETWORK WITH RECURRENT OUTPUT LAYER FOR LOW-LATENCY SPEECH SYNTHESIS”. (2019). URL: <https://static.googleusercontent.com/media/research.google.com/en/pubs/archive/43266.pdf>.
- [53] Francisco Javier Ordóñez ; és Daniel Roggen. “Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition”. (2015). DOI: 10.3390/s16010115. URL: <https://www.mdpi.com/1424-8220/16/1/115/htm>.
- [54] Katharina Eggenberger; Matthias Feurer. “Auto-Sklearn 2.0: The Next Generation”. (2020). URL: <https://www.automl.org/auto-sklearn-2-0-the-next-generation/>.
- [55] Feurer; Matthias, Klein; Aaron, Eggenberger; Katharina Springenberg; Jost, Blum; Manuel és Hutter; Frank. “Efficient and Robust Automated Machine Learning”. (2015), 2962–2970. old.
- [56] Matthias Feurer, Katharina Eggenberger, Stefan Falkner, Marius Lindauer és Frank Hutter. “Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning”. (2020).
- [57] M. J. Bahmani. “HyperBand and BOHB: Understanding State of the Art Hyperparameter Optimization Algorithms”. (2020). URL: <https://neptune.ai/blog/hyperband-and-bohb-understanding-state-of-the-art-hyperparameter-optimization-algorithms> (elérés dátuma 2022).

- [58] Bernadette M; RandlesIrene V; PasquettoMilena GolshanMilena GolshanChristine L; Borgman. “Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study”. (2017). DOI: [10.1109/JCDL.2017.7991618](https://doi.org/10.1109/JCDL.2017.7991618). URL: https://www.researchgate.net/publication/318751639_Using_the_Jupyter_Notebook_as_a_Tool_for_Open_Science_An_Empirical_Study.
- [59] Ekaba Bisong. “Google Colaboratory”. (2019). DOI: [10.1007/978-1-4842-4470-8_7](https://doi.org/10.1007/978-1-4842-4470-8_7). URL: https://www.researchgate.net/publication/336108351_Google_Colaboratory.
- [60] Docker Documentation. “Use containers for development”. (2021). URL: <https://docs.docker.com/language/java/develop/> (elérés dátuma 2022).
- [61] Microsoft Azure. “Azure Cosmos DB - NoSQL and Relational Database”. (2021). URL: <https://azure.microsoft.com/en-us/products/cosmos-db/> (elérés dátuma 2022).
- [62] Microsoft Azure Documentation. “Introduction to Blob (object) Storage - Azure Storage”. (2021). URL: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction> (elérés dátuma 2022).
- [63] Microsolder. “IPC szabványok és kiadványok”. (). URL: <http://microsolder-ipc.com/szabvanyok-es-kiadvanyok.html> (elérés dátuma 2022).
- [64] R. Péter. “Az IPC szabványrendszere”. (2021). URL: <https://www.elektro-net.hu/gyartosor/5062-az-ipc-szabvanyrendszere> (elérés dátuma 2022).
- [65] ASSOCIATION CONNECTING ELECTRONICS INDUSTRIES. “Generic Standard on Printed Board Design”. (2021). URL: <https://www.ipc.org/TOC/IPC-2221A.pdf> (elérés dátuma 2022).
- [66] CAP-XX Designs Manufactures the Thinnest. “PRISMATIC ULTRA THIN SUPERCAPS”. (2019). URL: <https://www.cap-xx.com/products/prismatic-overview/>.
- [67] Bosch. “BMI270 IMU combining accelerometer and gyroscope”. (October. 2021). URL: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmi270-ds000.pdf>.
- [68] S. Narkhede. “Understanding AUC - ROC Curve”. (2021). URL: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>.

- [69] T. Kanstrén. “A Look at Precision, Recall, and F1-Score”. (2021). URL: <https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec>.
- [70] Jorge Piazzentin Ono és mtsai. “PipelineProfiler: A Visual Analytics Tool for the Exploration of AutoML Pipelines”. (2020). URL: <https://deepai.org/publication/pipelineprofiler-a-visual-analytics-tool-for-the-exploration-of-automl-pipelines>.
- [71] J. Brownlee. “A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning”. (2016). URL: <https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning> (elérés dátuma 2022).

Ábrajegyzék

1-1. A diabéteszesek számának változása az idő függvényében [3].	5
1-2. Az inzulin és vércukorszint változása a nap folyamán [6].	6
1-3. A különböző inzulinfajták hatásideje [13].	8
1-4. Az inzulinpumpa részei [15].	9
1-5. A CGMS és az inzulinpumpa elhelyezkedése [16].	9
1-6. Az AI típusok felosztása.	10
1-7. A logisztikus regresszió működése.	11
1-8. A random forest felépítése.	12
1-9. A Markov lánc működési elve.	12
1-10. A support-vector machine működése.	13
1-11. Egy MLP neurális hálózat leegyszerűsített képe.	14
1-12. Az AI medicinában betöltött szerepköre [30].	16
2-13. IMU és mikrofontömb segítségével megvalósított adatgyűjtés [35].	18
3-14. MetamotionS szenzoregység [38].	20
3-15. Panoramic Digital Health eszköz.	21
3-16. Saját készítésű adatgyűjtő eszköz.	21
3-17. Keyes KY-038 mikrofon modul.	22
3-18. A FIR Wiener szűrő blokkvázlata diszkrét sorozatokhoz.	23
3-19. Egy sütemény elfogyasztásának szüretlen mozgási mintája.	24
3-20. Egy sütemény elfogyasztásának szűrt és tisztított mozgási grafikonja.	24
3-21. A mel skála és a frekvencia skála közti függvénykapcsolat.	25
3-22. Leves kanalizásának nem szűrt mel spektrogramja.	26
3-23. Egy DS-CNN felépítése [50].	27
3-24. Szekvenciális feldolgozás egy RNN-ben [52].	28
3-25. Egy LSTM neurális hálózat felépítése [53].	29
3-26. Az AutoML algoritmus felépítése [55].	30
3-27. BOHB optimalizálás versus Bayes-optimalizálás sebessége [57].	30
3-28. Jupyter Notebook App részei.	31
3-29. Google Colab működési struktúrája.	32
3-30. Az Azure szolgáltatás használatának logikai rendszerterve.	34
3-31. IPC szabványfa [64].	36
3-32. A fizikai eszköz ogikai rendszerterve.	37
3-33. Energiamenedzsment tervrajz.	38
3-34. A PCB robbantott ábrája.	38
3-35. A PCB 3D ábrája.	39
3-36. A szimuláció felépítése.	40
3-37. 60 másodperces szimuláció eredménye.	40

4-38. Az AUC szemléltetéséhez használatos regressziós előrejelzések.	41
4-39. ROC görbe.	42
4-40. Egy VGG16 háló eredményét bemutató konfúziós mátrix.	43
4-41. A recall és a precision szemléltetése.....	44
4-42. A Pipeline Profiler eredmény megjelenítő.....	45
5-43. A 3nap/1nap train/test arányú neurális háló pontosság függvénye.....	47
5-44. Az SKlearn 1.0 pipeline profiler kiértékelési felülete.....	48
5-45. Az SKlearn 2.0 pipeline profiler kiértékelő felülete.	49

Táblázatok jegyzéke

5-1. VGG16 CNN eredménye.....	46
5-2. Pontosság és F1-score értékek az LSTM hálózat esetén	46
5-3. Az SKlearn 1.0 eredménye	48
5-4. Az SKlearn 2.0 eredménye	49
5-5. AutoML SKlearn 2.0 eredménye.	49
5-6. DS-CNN eredménye.	50
5-7. VGG16 és DS-CNN kombinációjának eredménye.	50
5-8. SKlearn 2.0 és DS-CNN kombinációjának eredménye.....	51