



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Elektrokardiográfián alapuló migrénes rohampredikció mesterséges intelligencia segítségével

TDK dolgozat

Készítette:

Fazekas Gergő

Konzulens:

dr. Dabóczi Tamás
dr. Molnár Balázs

2022

Tartalomjegyzék

Kivonat	i
Abstract	ii
1. Bevezetés	1
1.1. Korábbi migrénpredikációs kutatások	1
1.2. Rohampredikációs rendszer fejlesztése alprojekt	2
1.3. Saját megközelítés	2
2. Háttérismeretek	4
2.1. Migrén	4
2.1.1. Migrén típusok	4
2.1.2. Megelőző és követő tünetek	4
2.1.3. Kezelése	4
2.2. Elektrokardiográfia	5
2.2.1. EKG vizsgálat	5
2.2.2. Elektrokardiogram	6
2.3. Szívfrekvencia-variabilitás	6
2.3.1. HRV paraméterek	6
2.3.2. Migrén és a HRV kapcsolata	7
2.3.3. Hipotézis	7
2.4. Gépi tanulás	7
2.4.1. Perceptron	7
2.4.2. Többretegű perceptron	8
2.4.3. Konvolúciós architektúra	8
3. Implementáció	10
3.1. Projekt környezet	10
3.2. Használt programozási nyelv és könyvtárak	10
3.3. Adatok tárolása és kezelése	10
3.4. Előfeldolgozási folyamat optimalizálása	11
3.5. Konfiguráció és logolás	11
4. Adatok	12
4.1. Rendelkezésre álló adatok ismertetése	12
4.2. Felhasználandó adatok kiválasztása	13
4.3. Nyers EKG jeltől az RR-intervallumokig	14
4.3.1. Jel szűrése	14
4.3.2. R-csúcs detekció és RR-intervallumok	15
4.4. Kiugró adatok kezelése	15
4.4.1. Kiugró éjszakák szűrése	16

4.4.1.1.	Szűrés az R-csúcsok száma alapján	16
4.4.1.2.	Szűrés a gyorsulásmérő értékek alapján	17
4.4.1.3.	Szűrés RR-intervallumok szórása alapján	17
4.4.2.	Kisebb előfeldolgozási hibák kompenzálása	19
4.4.3.	Összegzés	21
4.5.	Adatok felcímkézése	21
4.5.1.	Kontroll adatok kiválasztása	22
4.5.2.	Roham előtti adatok kiválasztása	22
4.5.3.	Megvalósítás	23
4.6.	Tanító és validációs halmazok elkülönítése	23
4.6.1.	Az elkülönítő logika	23
4.6.2.	A validációs halmazba választott adatok aránya	24
4.7.	Adatok skálázása	24
4.8.	Összefoglalás	25
5.	Mesterséges intelligencia alapú megközelítések	26
5.1.	Adatok megvizsgálása	26
5.2.	A modell kimenete és a hibafüggvény	27
5.3.	Egyszerű megközelítés	27
5.3.1.	Bemenetek és architektúra	27
5.3.2.	Eredmények	28
5.4.	Tanítási folyamat fejlesztése	29
5.4.1.	Általánosító képesség növelése	29
5.4.2.	Validáció reprezentativitásának növelése	30
5.4.3.	Kategóriák kiegyenlítetlenségének csökkentése	30
5.4.4.	Változtatások hatása	31
5.5.	Paraméter alapú megközelítés	31
5.5.1.	Megközelítés	31
5.5.2.	Modell architektúra	32
5.5.3.	Eredmények	32
5.6.	Különbség alapú megközelítés	34
5.6.1.	Ötlet	34
5.6.2.	Tanítási folyamat	34
5.6.3.	Modell architektúra	35
5.6.4.	Eredmények	36
5.7.	Összefoglalás	37
6.	Konklúzió	39
	Köszönetnyilvánítás	41
	Irodalomjegyzék	42

Kivonat

A migrén az egyik leggyakoribb fejfájásos betegség, melynek prevalenciája becslések szerint globálisan a felnőtt lakosság 7.5 - 15%-a. A tipikusan lüktető, féloldalas fájdalom általában rohamszerűen jelentkezik, amelyet hányinger, hányás, fény- és hangérzékenységgel kísérhet. Ennek megszüntetésére, illetve tüneteinek enyhítésére léteznek akut, illetve preventív megoldások, csakhogy mind a mai napig nem megoldott a migrén hatékony kezelése minden beteg számára. Ezzel szemben a sikeres előrejelzése lehetőséget biztosíthat a rohamra való felkészülésre, a használt gyógyszer bevételi idejének optimalizálására, amely ezáltal jelentős segítséget nyújthat az elszenvedőnek. [4]

Ezt a feladatot kíséri meg megoldani az E-Group ICT Software Zrt. és a Pécsi Tudományegyetem közös, „Roham predikciós rendszer fejlesztése” című alprojektjének egyik részfeladata, amely azon hipotézisen alapszik, hogy a migrénes roham a szívfrekvencia változásából előrejelezhető. A feltevés a nemzetközi szakirodalomban publikált kis mintaszámú kísérlet [22] biztató eredményein alapul. A projekt célja az ezen elmélet független kísérletekkel való igazolása (esetlegesen a hipotézis elvetése) a kutatás során összegyűjtött, elektrokardiográfiai (EKG) adatok felhasználásával, illetve pozitív eredmény esetén hatékony predikciós algoritmus kifejlesztése.

A feladatom első lépésében elkészítettem egy, nagy mennyiségű nyers EKG eredményeket hatékonyan előfeldolgozó programot, amely először megszűri, majd kinyeri és eltárolja a szívdobbanások között eltelt időkülönbségeket, majd az így kapott adatokat a migrénes roham előtti, illetve attól megfelelő időbeni távolságra lévő kategóriákba sorolja. Ezt követően a pécsi kutatók, a konzulenseim, illetve saját felvetéseimet felhasználva több mesterséges intelligencia módszert dolgoztam ki, melyek célja annak vizsgálata, hogy a hipotézis igazolható-e, a migrén prediktálható-e univerzálisan. A különböző elgondolásokon alapuló eredményeket elemeztem, kiértékeltem és összehasonlítottam.

Abstract

Migraine is one of the most common headache diseases, its prevalence is estimated to be 7.5 - 15% of the adult population globally. The typically throbbing, one-sided pain usually occurs in attacks and may be accompanied by nausea, vomiting, and sensitivity to light and sound. There are both acute and preventive solutions to eliminate and alleviate its symptoms, but the effective treatment of migraine for all patients has not been found yet. On the other hand, its prediction could provide an opportunity to prepare for the attack, and to optimize the time of taking the used medicine, which can thereby provide significant help to the sufferer. [4]

This task is attempted to be solved by one of the sub-tasks of the joint sub-project of E-Group ICT Software Zrt. and the University of Pécs entitled "Development of a seizure prediction system", which is based on the hypothesis that a migraine attack can be predicted from the change of the heart rate. The assumption is derived from the encouraging results of a published small-sample trial: [22]. The aim of the project is to verify this theory with independent experiments (possibly disprove the hypothesis) using the electrocardiographic (ECG) data collected during the research and, in case of a positive result, to develop an effective prediction algorithm.

In the first step of my task, I implemented a program that efficiently pre-processes a large amount of raw ECG data, that first filters, then extracts, and stores the time differences between heartbeats, and then classifies them into the following categories: those before the migraine attack and those at a suitable time distance from it. Subsequently, using the suggestions of the researchers from Pécs, my supervisors, and my own, I developed several artificial intelligence methods, the purpose of which is to investigate whether the hypothesis can be verified, whether migraine can be predicted universally. I analyzed, evaluated, and compared the results based on different ideas.

1. fejezet

Bevezetés

A migrén becslések szerint nagyjából minden tizedik, tehát globálisan körülbelül 800,000,000 embert érintő fejfájásos betegség. Sajnálatos módon számottevő negatív társadalmi és gazdasági hatása van, amely köszönhető például az egyén életminőségének romlásának, vagy a betegségből fakadóan kihagyott, vagy nem effektív munkanapoknak. Sajnos maga a betegség nem gyógyítható, csupán enyhíthető a rohamok intenzitása és csökkenthető a gyakorisága a megfelelő életmód, illetve gyógyszeres kezelés által. [4]

A migrén predikciója segíthet optimalizálni a gyógyszerek bevételét, felkészülési lehetőséget biztosíthat az elszenvedő számára, illetve amennyiben sikerül megfelelő predikciós eljárást kidolgozni, az jelentősen segítheti a betegség megértését és további kutatását.

1.1. Korábbi migrénpredikciós kutatások

Maga a migrénpredikciós elképzelés talán ma már nem számít igazán új ötletnek. Egy 1999-es cikk szerint a migrén kutatása azelőtt leginkább közvetlenül a roham alatti vizsgálatokra vonatkozott, és újszerűnek nevezi meg a rohamon kívül eső időtartamban való vizsgálódást. Az általa hivatkozott kutatások utalnak arra, hogy úgynevezett Event Related Potential (ERP) analízis segítségével a migrén prediktálható. Konklúziója, hogy a roham előrejelzésének kutatása új kapukat nyithat meg a betegség megértésének és kezelésének irányába. [12]

Számos kutatás irányult arra, hogy összefüggéseket találjanak különböző életvitel jellegű adatok és a rohamok között. Ezek egyike például egy 2020-as kutatás, amely során 178 ember 90 napnyi adata került összegyűjtésre (összesen 1870 roham). Ez tartalmazta a migrén előfordulását, a stressz-szintet, az alvás minőségét, a koffein- és alkoholfogyasztást, a menstruációs ciklust, illetve az egyén saját migrén predikcióját. Az így összegyűjtött adatokból sikerült összefüggéseket felfedezni bizonyos változók és a rohamok között, viszont a többváltozós modell a migrénes roham kockázatát csak valamivel jobban jósolta a véletlennél. [10]

A 2020-as évben a Headache lapban megjelent egy cikk „Predicting the Future of Migraine Attack Prediction” címmel, amely a migrénes roham előrejelzésének jövőjét hivatott kutatni. A cikk bemutatja, hogy korábbi kutatásoknak ugyan sikerült olyan összefüggéseket felfedezni, amelyek valószínűsíthetők, hogy egyes változók hajlamosítható tényezők, viszont kiemeli azt is, hogy a csoportszintű predikció nagyon nehéz és még egyelőre nem egyértelműen megoldott feladat. A szerző szerint a jobb eredmények érdekében egyrészt hosszútávú mérésekre, másrészt a figyelendő paraméterek kiterjesztésére lenne szükség. Ezen belül javasolja a különböző fiziológiás állapotok monitorozását, mint például a szívfrekvencia-variabilitás, bőrhőmérséklet, galvanikus bőrreakció, vagy akár alvásminőség, fizikai aktivitás, érzelmi állapot. [3]

Végül, hadd emeljem ki egy finn egyetem publikációját 2018-ból, amely a legerősebben kapcsolódik ezen dolgozathoz. A kutatás során a betegek által viselt szenzorok adatait használták fel, amelyek a résztvevők mozgását (gyorsulásmérő segítségével), hőmérsékletét, galvanikus bőrreakcióját, pulzusát, illetve szívfrekvencia-variabilitását monitorozták. Mivel a kiválasztott mérőműszerek kifejezetten érzékenyek a mozgásra, ezért azt a döntést hozták, hogy csupán az alanyok alvása során mért adatokat veszik figyelembe. Az egy-egy éjszakára vonatkozó eredményekből meghatározták a mért értékek átlagát, szélsőértékeit és számos más mérőszámot, illetve az adatok feldúsítása érdekében egy-egy estét az összes többi, de azonos alanyhoz tartozó estéhez hasonlítottak a kiszámolt paraméterek értékeinek különbségét képezve. Végül különböző klasszifikációs módszerekkel igyekeztek olyan modellt létrehozni, amely a lehető legnagyobb pontossággal tudja meghatározni az adott éjszakából származó adatokról, hogy az adott napon volt-e migrénes rohama az illetőnek vagy sem. Ezen eljárást használva biztató eredményekre jutottak. Volt olyan alany, akire egyénre szabottan sikerült, akár több mint 95%-os pontosságot elérniük, viszont univerzális, minden résztvevőre egyaránt alkalmas prediktorra vonatkozó próbálkozások nem jártak sikerrel. Az utóbbi oka lehet az is, hogy a kutatásban összesen csupán 7 alany vett részt, amely vélhetően nem elég egy jó általánosító képességgel rendelkező modell megalkotásához. [22]

1.2. Rohampredikciós rendszer fejlesztése alprojekt

A E-Group ICT Software Zrt. és a Pécsi Tudományegyetem közös, 2018-ban induló InnoHealth DataLake projektjének célja „egy csúcstechnológián alapuló, az egészségügyi ellátást, megelőzés és kutatást támogató, egységes folyamatokon alapuló standardjainak és akkreditációs rendszerének kialakítása” [11].

Egyik alprojektje a „Roham predikciós rendszer fejlesztése” címet viseli, amely egyik részfeladata során több, mint 100 migrénes beteg elektrogardiográfiai (EKG) értékeit monitorozták. Célja összefüggések felfedése az adatokból meghatározott HRV (heart rate variability, magyarul: szívfrekvencia variabilitás) paraméterek és a kialakuló migrénes roham között. Egyedisége, hogy ismeretem szerint nincs másik olyan kutatás, amely egyrészt ilyen mennyiségű EKG adatot gyűjtött volna betegektől migrénpredikciós kutatási céllal¹, illetve számos korábbi tanulmánnyal ellentétben nem veszi figyelembe a résztvevők más fiziológiás paramétereit, hanem egyedül a HRV értékekre hagyatkozik. Amennyiben biztató eredmények születnek, az feltehetően komoly előrelépést jelenthet a migrén kutatásának szempontjából, ha pedig populációs szinten hatékony előrejelzést lehetővé tevő felfedezésre jutnak, akkor az akár forradalminak is nevezhető, tekintve a korábbi tanulmányok eredményeit.

Ezen kutatásba való becsatlakozásra kaptam lehetőséget, amely során biztosítottak számomra hozzáférést az összegyűjtött adatokhoz, az elkészült megoldásokhoz, továbbá számítási kapacitást azok feldolgozásához². Mindezt azzal a céllal, hogy munkámmal segítsen az elsősorban mesterséges intelligencia alapú megközelítését a feladatnak.

1.3. Saját megközelítés

Tehát célom volt egy olyan gépi tanulási modell megalkotása, amely képes megtalálni azon keresett mintázatot a szív dobogásának ütemében, amely a kialakuló migrénre utal. Saját megközelitésem alapvetően abban tér el a kutatók által elsődlegesen képviselttől, hogy nem az előre meghatározott integrális paraméterekből való predikcióra fókuszál, hanem a

¹Összehasonlítva például azzal, hogy a [22] tanulmányban összesen 7 alany vett részt.

²Emellett részt vehettem konzultációkon, illetve a projekttel kapcsolatos megbeszéléseken is.

szívdobbanások közötti időkülönbség szekvenciát veszi alapul és mély tanulási módszereket alkalmaz.

Ehhez a kutatási alanyok éjszakai (0 és 6 óra közötti) EKG értékeit használtam fel (a [22] kutatáshoz hasonlóan), melyeket feldolgoztam, saját ötleteim alapján kisztisztítottam, majd egyéni megoldással csoportosítottam migrénes roham előtti, illetve úgynevezett kontroll kategóriákba. A munkám során számos mély tanulási megközelítést dolgoztam ki, amelyek eredményeit elemeztem, illetve összehasonlítottam.

2. fejezet

Háttérismeretek

2.1. Migrén

A migrén egyike a legmegterhelőbb olyan fejfájásbetegségeknek, amelyeknek nincs egyértelmű kiváltó oka. Becslések szerint globálisan a felnőtt lakosság férfi tagjainak 4 - 9,5%-át, míg a nők 11,2 - 25%-át érinti. Hazánkban a migrén prevalenciáját 9,6%-nak mérte egy 1998-as felmérés. A rohamok egyénenkénti gyakoriságát illetően a migrénben szenvedők kb. 35%-a 1-4, míg kb. 25%-a legalább 4 rohamot szenved el havonta. [4] [13]

2.1.1. Migrén típusok

A migrénes betegségek az alábbi két nagy csoportra bonthatók: migrén aurával és migrén aura nélkül.

Aura nélküli típus (másnéven közönséges migrén) esetén a visszatérő roham hossza tipikusan 4 - 72 óra, amelyet egyoldalú, pulzáló fájdalom jellemez. További tünetei közé tartozik a hányinger, illetve a fényre és hangokra való túlérzékenység.

Aurás migrén (másnéven klasszikus migrén) esetében ismétlődő, percekig tartó, általában egyoldalú fájdalom mellett neurológiai tünetek is jelentkeznek, leggyakrabban vizuális, vagy érzékelési zavarok formájában. [9]

2.1.2. Megelőző és követő tünetek

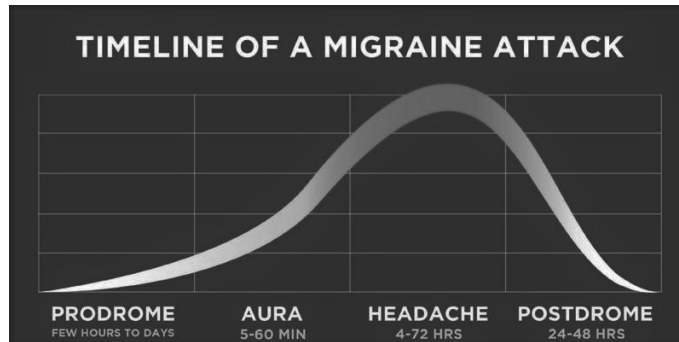
A prodromális tünetek órákkal vagy egy-két nappal kezdődhetnek a migrénes roham többi tünete előtt. Ide tartozhatnak a fáradtság, koncentrációs nehézségek, nyakmerevség, homályos látás, ásítás és sápadtság, illetve a roham alatt is tapasztalható hangokra és fényre való túlérzékenység.

Fontos tény a rohamelőrejelzés kutatásának szempontjából, hogy ezt számos beteg képes a saját tünetei alapján meghatározni. Ezt egy 2003-as kutatás eredménye is igazolja, amely során 97 ember 3 hónapnyi feljegyzéseiből derült ki, hogy az esetek 72%-ában a betegek előre meg tudták jósolni a roham bekövetkeztét. Ami a prodromális tüneteket illeti, a leggyakoribb ezek közül a fáradtság, kimerültség volt, míg kevesebb esetben ugyan, de koncentrációs nehézségekről és nyakmerevségről is panaszkodtak a résztvevők. [7]

A fejfájás megszűnését követően (legfeljebb 48 órán át) további, úgynevezett poszt-dromális tünetek állhatnak fenn, amelyet a prodromális szakaszhoz hasonlóan leggyakrabban fáradtság vagy kimerültség, koncentrációs nehézségek és nyakmerevség jellemezhet. [9]

2.1.3. Kezelése

A migrén kezelése történhet akut, illetve preventív módon.



2.1. ábra. A migrén fázisai [23]

Az akut kezelés célja a tünetek enyhítése, illetve megszüntetése. Elsősorban a migrén-roham erőssége, a korábbi terápiás próbálkozások tapasztalata, illetve az esetleges társbetegségek határozzák meg a roham esetén használandó gyógyszert.

Prevenció céljából léteznek különféle terápiás gyógyszerek, amelyek mindennapi szedése csökkenti a rohamok hosszát és súlyosságát, illetve segítheti a megelőzést a rohamot provokáló tényezők elkerülése is. Ilyenek lehetnek például egyes ételek, italok, gyógyszerek, a stressz, kialvatlanság, vagy akár a túlzott fizikai aktivitás.

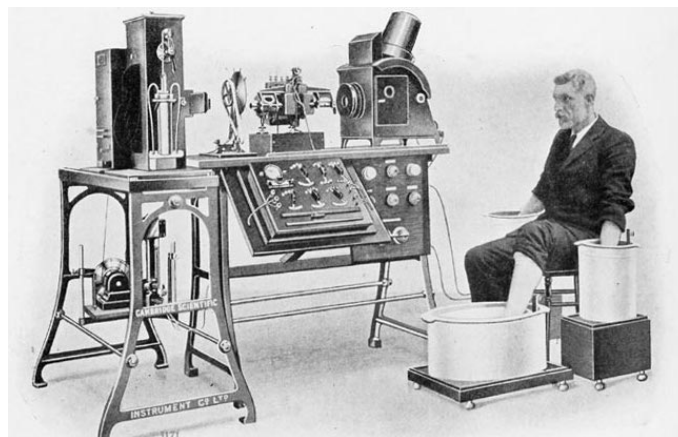
Mind a mai napig nem megoldott a migrén hatékony kezelése minden beteg számára, továbbá a használt gyógyszerek számos (akár súlyos) mellékhatással is rendelkeznek, ami adott társbetegségek jelenlétében jelentősen korlátozhatja a használatukat. [4] [16]

2.2. Elektrokardiográfia

Az elektrokardiográfia (röviden EKG) olyan eljárás, amely képes a szív által generált elektromos potenciálkülönbségek mérésére az emberi test megfelelő pontjaira helyezett elektródák segítségével. [24] [8]

2.2.1. EKG vizsgálat

Willem Einthoven holland fizikus a 20. század elején elsőként rögzített sikeresen EKG görbét, ezzel lefektetve a szívbetegségek kivizsgálásának legalapvetőbb technikájának alapjait. Munkásságáért 1924-ben orvosi Nobel díjjal jutalmazták és a mai napig az „elektrokardiográfia atyja” elnevezéssel illetik. [20]

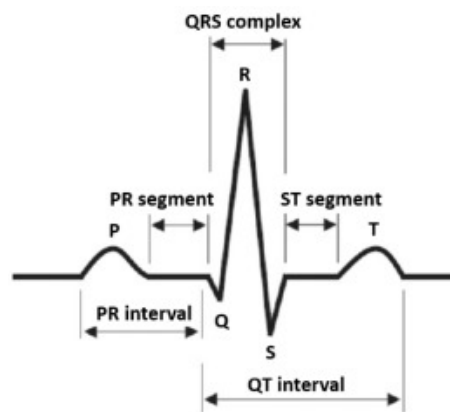


2.2. ábra. Egy teljes elektrokardiogram Willem Einthoven idejéből [8]

Érdekes módon a szívizomrostokban keletkező feszültségkülönbségek a test felületén a szívtől távolabb is érzékelhetők. Ennek oka, hogy a testünk egy jelentős százalékát jó vezetőképesseégű elektrolitoldat alkotja. A mérőeszköz elektródáinak a test különböző pontjaira (tipikusan a végtagokra és a mellkasra) való felhelyezését követően a potenciálkülönbség változásai regisztrálhatók, az így keletkező görbe az úgynevezett elektrokardiogram. Ebből lehet számos következtetést levonni, például a szívfrekvenciára, az ingerület kialakulására és terjedésére vonatkozólag. [5]

2.2.2. Elektrokardiogram

Az EKG görbéről leolvasható hullámokat ABC sorrendben P-től T-ig betűk jelölik. A szinuszcsozóban keletkezett ingerület tovaterjed és aktiválja a pitvarokat. A P hullám a pitvari depolarizációt jelzi. „A QRS csoportot alkotó hullámok jelzik a kamraizomzat depolarizációját. Ezen belül a negatív Q-hullám a kamraizomzat depolarizációjának kezdetét, a pozitív R-hullám a kamra fő tömegének depolarizációját, a negatív S-hullám pedig az utolsóként aktiválódó bázis depolarizációját jelzi.” [5]



2.3. ábra. EKG görbe részei [18]

2.3. Szívfrekvencia-variabilitás

A szívfrekvencia-variabilitás (angolul: Heart Rate Variability, röviden HRV) az egymást követő szívverések közötti időtartam ingadozása. Az egészséges szív nem metronómként viselkedik, hanem folyamatosan változik a frekvenciája, sőt a nagyon szabályos szívverés idegrendszeri károsodásra, illetve nem megfelelő egészségi állapotra utalhat, viszont a túl nagy varianciával rendelkező szívverés sem feltétlenül jelent jót, ugyanis okozhatja olyan probléma is, mint a pitvarfibrilláció. [21]

2.3.1. HRV paraméterek

A szívverések közti időkülönbségeket a relatíve könnyen detektálható R-csúcsok között értelmezzük tipikusan, amelyet RR-intervallumoknak nevezünk. Ezen intervallumok székenciája érdekes módon rengeteg információt hordoz az egyénre vonatkozóan, ezért az orvostudomány számos integrális értéket határozott meg, amelyeket HRV paramétereknek nevezünk. Ezek meghatározása történhet az RR-intervallumok időtartományából (például az értékek szórása), frekvenciatartományából, de nemlineáris, egészen összetett mérőszámok is léteznek. Ezen kiszámolt értékek összefüggésbe hozhatók különböző egészségügyi jellemzőkkel, ezáltal lehetőséget biztosítva az orvosok számára különböző diagnózisok meghatározására. [21]

2.3.2. Migrén és a HRV kapcsolata

Ezen projekt szempontjából az egyik legmeghatározóbb kérdés, hogy mi a kapcsolat a migrén és a szívdobbanások között eltelt időintervallumok szekvenciája között. A válasz alapvetően abban keresendő, hogy a vegetatív idegrendszer aktivitása összefüggésbe hozható a migrénnel¹, amely működése a megfelelő HRV paraméterekkel megfigyelhető. A nagyfrekvenciás (0,15 Hz feletti tartomány) viselkedésre vonatkozó HRV paraméterek a paraszimpatikus, míg az alacsony frekvenciára (0,04 - 0,15 Hz) vonatkozóak pedig szimpatikus aktivitáshoz kapcsolódnak. [14] [17]

2.3.3. Hipotézis

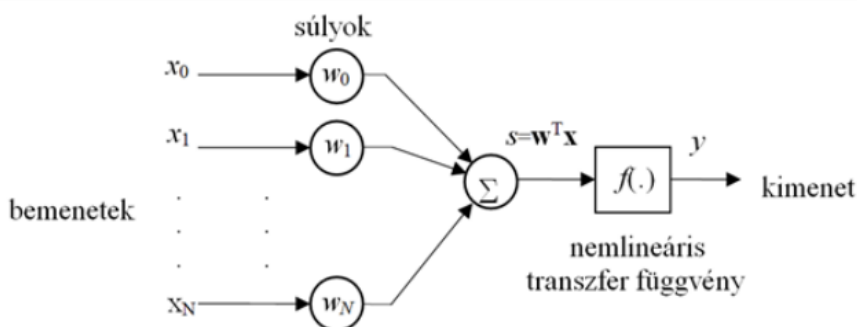
Tehát összefoglalva, a szívdobbanások között eltelt időközök szekvenciája sok hasznos információt tartalmaz, többek között a vegetatív idegrendszer aktivitására vonatkozólag. Amennyiben a migrénes rohamot megelőzően az illető szervezetében olyan jellegű elváltozás történik, amely összefüggésbe hozható a szívfrekvencia-variabilitással, akkor ez detektálható, ezáltal lehetőséget biztosíthat rohampredikcióra egy egyszerű noninvazív eljárás által.

2.4. Gépi tanulás

A gépi tanulás a mesterséges intelligencia egy részhalmaza, amely az új körülményekhez való adaptálódási képességgel, a mintázatok detektálásával és általánosításával foglalkozik.

Ezen komplex témakör alapjainak ismertetése helyett igyekeztem csak a legrelevánsabb részleteket, azaz az általam a projekt során felhasznált architektúrákat röviden és lényegretörően bemutatni annak érdekében, hogy mellőzzem az egyébként széleskörű szakirodalomban fellelhető információk ismétlését. A dolgozat során kifejezetten igyekeztem közérthető megfogalmazásokra és alapos magyarázatokra annak érdekében, hogy a témában kevesebb ismerettel rendelkezők számára is érthető legyen. A gépi tanulás iránt mélyebben érdeklődők számára tudom javasolni többek között az ingyenesen hozzáférhető [19] és [2] szakirodalmakat.

2.4.1. Perceptron



2.4. ábra. A perceptron felépítése [2]

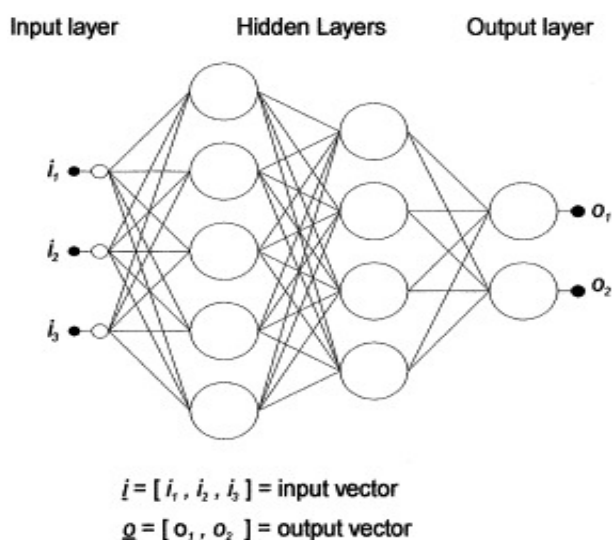
A perceptron az agyunkat felépítő neuron leegyszerűsített matematikai modellje, amely felépítése a 2.4. ábrán látható. Átviteli függvénye az $y = f(\mathbf{w}^T * \mathbf{x} + c)$ egyenlettel írható fel, ahol $\mathbf{x}$ a bemeneti értékek oszlopvektora, $\mathbf{w}$ a súlyok oszlopvektora, c

¹Például a vegetatív diszfunkció tünetei gyakoriak a migrénes betegeknél

egy konstans érték (amely nem minden esetben használatos), $f()$ az úgynevezett transzfer függvény, y pedig az egydimenziós kimeneti érték. Tehát a modell a futtatása során a bemeneti értékeket súlyozásra kerülnek a modell paramétereivel, majd ezek összegzését követően, általában nemlineáris függvényt alkalmazva jön létre a kimenet. [2]

2.4.2. Többrétegű perceptron

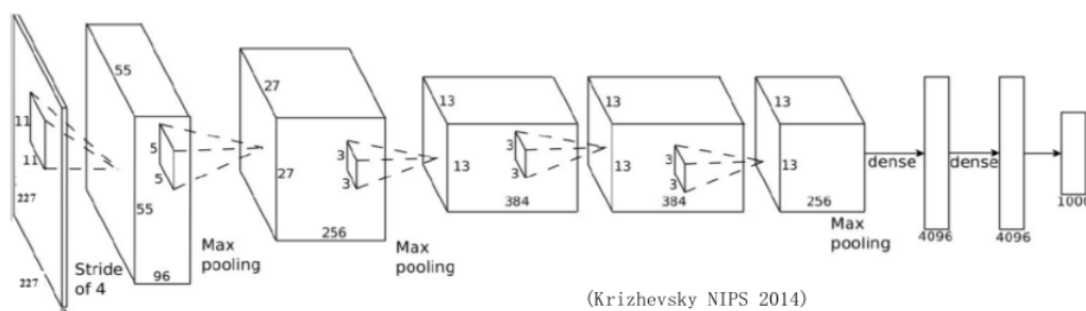
A perceptronokból összerakott többrétegű hálózatot nevezzük többrétegű perceptronnak (lásd 5.2. ábra). Természetesen előnye az egyetlen perceptronnal szemben, hogy komplexebb mintázatok felismerésére is képes. Létrehozásakor a megválasztandó hiperparaméterek a következők: a rétegek száma, az egyes rétegekben található perceptronok száma, az egyes rétegek által használt transzfer függvények. [6]



2.5. ábra. Példa egy többrétegű perceptron hálózatra [6]

2.4.3. Konvolúciós architektúra

A neurális hálózatok növekvő népszerűségével a többrétegű perceptron modellt továbbfejlesztették különböző speciális rétegekkel, amelyek segítségével adott feladatokra vonatkozóan jobb eredményeket lehet elérni. Az egyik ilyen a méltán híres konvolúciós réteg, amely meghatározott számú és dimenziójú szűrőt alkalmaz a bemenetére és ezek együttműködését állítja a tanulási folyamat során.



2.6. ábra. AlexNet architektúra. [1]

Gyakran alkalmazott képfelismerési feladatokra, ugyanis az egymást követő konvolúciós rétegek képesek kiemelni az objektumok körvonalait, majd abból komplex mintázatokat

meghatározni. Az alábbi 2.6. ábrán látható az AlexNet konvolúciós architektúra, amely első lépésben a bemenetére adott képből kinyeri a releváns információt konvolúciós rétegek segítségével, majd a végén található többrétegű perceptron dönti el, hogy mi látható rajta, vagyis melyik kategóriába tartozik. [1]

3. fejezet

Implementáció

3.1. Projekt környezet

Az adatok szenzitivitása miatt azok lokális tárolására és feldolgozására nem volt lehetőségem, ezért először egy virtuális gépen keresztül, majd a későbbiekben egy Linux szerverre csatlakozva végeztem a fejlesztést és futtatást, melyeket az E-Group ICT Software Zrt. biztosított számomra.

A virtuális gépen keresztüli fejlesztést az oda feltelepített IDE használtával végeztem, illetve annak érdekében, hogy a projekt minél rugalmasabb legyen, Docker fejlesztőkörnyezetbe telepítettem a szükséges elemeket és csatoltam hozzá a megfelelő könyvtárakat.

A későbbiekben, a *Linux* szerveren való fejlesztéshez pedig kihasználtam, hogy a *Visual Studio Code* ingyenes IDE lehetőséget biztosít szerveren való fejlesztésre SSH kapcsolaton keresztül. Így közvetlenül a fejlesztőkörnyezetből elérhettem kényelmesen a távoli fájlrendszert és erőforrásokat, illetve futtathattam és debuggolhattam a kódjaim ugyanolyan módon, ahogyan azt lokálisan is tenném.

Annak érdekében, hogy a projekt könnyen mozgatható legyen, és a változtatások követhetőek legyenek, a fejlesztés során a *Git* verziókezelő rendszert használtam és a saját *GitHub* oldalamon tárolom a projektet.

3.2. Használt programozási nyelv és könyvtárak

A gépi tanulásra és adatok feldolgozására gyakran alkalmazott *Python* programnyelvet használtam a munkám során, amely telepítését az úgynevezett *Anaconda* programon keresztül hajtottam végre. Ez többek között lehetővé teszi a csomagok telepítését és törlését, a verziók összehangolását egyszerű módon, illetve több különböző *Python* verzió kezelését. Az adatokkal való megismerkedéshez, illetve az ábrák készítéséhez az interaktív és kényelmes felületet biztosító *Jupyter Notebook*-ot használtam.

Az adatok kezelésére a *pandas*, *NumPy* és *SciPy*, míg a gépi tanulási modellek elkészítéséhez és tanításához a *Tensorflow*, illetve *Scikit-learn* könyvtárat alkalmaztam.

3.3. Adatok tárolása és kezelése

A mért EKG és gyorsulásmérő értékek sok *.EDF* kiterjesztésű fájlban álltak rendelkezésre, majd a megfelelő beolvasó függvényt felhasználva egy-egy *NumPy* tömb jött létre, melyeket táblázatos formára hoztam az elérési útvonal, páciens kódja, kezdő és végidőpont adatokkal összepárosítva.

Később az RR intervallumok meghatározását követően, azok tárolási méretének csökkentése érdekében a számábrázolási formát az alapértelmezettől *uint16*-ra állítottam,

azon *a priori* ismeretekre alapzva, hogy egyrészt két egymást követő szívdobbanás közötti időkülönbség csakis pozitív értékű lehet, illetve *ms*-ban kifejezve nem haladhatja meg a $2^{16} = 65536$ értéket (ami 65,5 másodpercnek felel meg).

Az adatokat szinte mindvégig táblázatos formában kezeltem és mentettem is el. Az utóbbihoz a *pickle* fájltypust használtam, amely előnye a *.csv* kiterjesztéssel szemben, hogy tárhely szempontjából kedvezőbb, illetve képes a táblázat soraiba ágyazott tömbök tárolására is.

3.4. Előfeldolgozási folyamat optimalizálása

A hatalmas mennyiségű nyers EKG értékek RR inverallumokra hozása időigényes. Éppen ezért igyekeztem kihasználni a rendelkezésemre álló hardver számítási kapacitását és a *multiprocess* könyvtár segítségével párhuzamosítottam a feldolgozási folyamatot.

Emellett a hosszú (akár több napos) futási idő következményeként fokozottan ügyelni kellett arra is, hogy a felmerülő hibák megfelelően legyenek lekezelve, amely egyrészt vonatkozik a hiba okának elmentésére, másrészt arra, hogy ne okozzák a program leállását. Erre azt a megoldást alkalmaztam, hogy amennyiben a feldolgozás során az egyik folyamat hibára fut, akkor a hibaüzenetet egy, az elnevezési konvenciónak megfelelő *.log* fájlba menti el.

Annak érdekében, hogy a feldolgozás állapotát nyilván lehessen tartani, egy táblázatban összegzi a rendszer ezt az információt, amely minden sorában az adott fájl elérési útvonala mellé társítja a státuszát, amely a következők valamelyike lehet: *sikeres*, *elbukott*, vagy *még nem lett feldolgozva*.

3.5. Konfiguráció és logolás

Mivel egyrészt a teljes folyamat számos helyen paraméterezhető, ezért igyekeztem ezeket külön kezelni a programkódtól az átláthatóság és egyszerű megváltoztathatóság érdekében. Másrészt hosszú futások esetén fokozottan fontos az események naplózása, azaz a megfelelő logolás. Mindkettő együttes megoldására a *Python könyvtárként* telepíthető *Hydra* eszközt használtam fel, amely egyszerre biztosít keretet konfigurációs fájlok készítésére és felhasználására, illetve kényelmes módot a log fájlok automatikus létrehozására.

4. fejezet

Adatok

4.1. Rendelkezésre álló adatok ismertetése

A kutatásban 102 résztvevő alany EKG és gyorsulásmérő szenzor értékei, továbbá a migrénes rohamok időpontjai álltak rendelkezésemre. Viszont sajnos vannak olyan betegek, akik adatgyűjtése meghiúsult valamilyen oknál fogva, vagy nincs megfelelően dokumentálva a migrénes rohamjainak időpontja. Miután átnéztem alaposan a kapcsolódó feljegyzéseket, úgy döntöttem hogy a fenti okok miatt 17 illető adatát nem veszem figyelembe, tehát a maradék 85 résztvevőre vonatkozóakat tudtam a következőkben csak felhasználni.

Ahogy az az alábbi, 4.1. táblázatról leolvasható, a nem összefüggő felvételek összegzett hossza betegenként 1 és 36 nap közé esik, körülbelül 18,7 napos átlaggal.

átlag	18 nap 17:33
szórás	5 nap 23:02
minimum	5 nap 14:33
25. percentilis	15 nap 11:08
50. percentilis	19 nap 13:48
75. percentilis	20 nap 12:47
maximum	35 nap 20:48

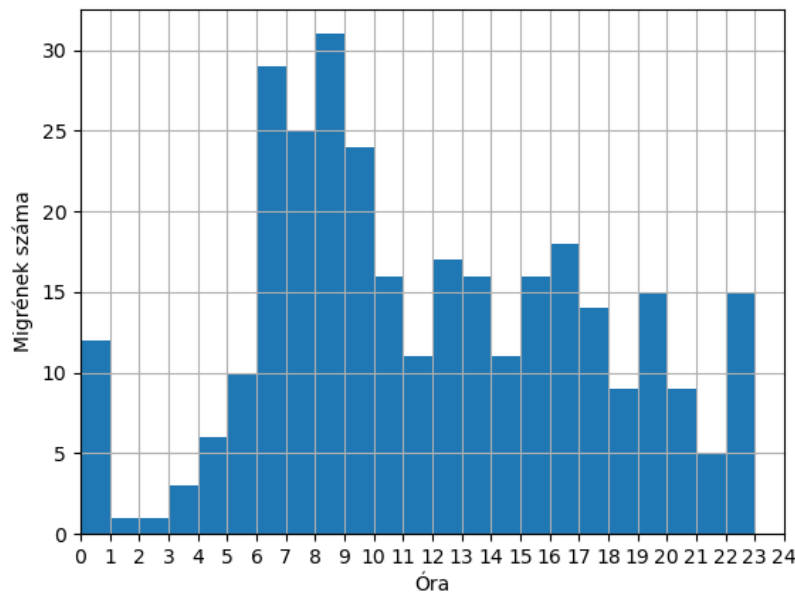
4.1. táblázat. Statisztika a felvételek összegzett hosszáról betegenként.

A kiválasztott 85 beteg összesen 314 migrénes rohamot regisztrált, illetve 11 alany egyáltalán nem tapasztalt egyet sem az adatgyűjtés során.

átlag	3.64
szórás	2.48
minimum	0
25. percentilis	2
50. percentilis	4
75. percentilis	6
maximum	8

4.2. táblázat. Statisztika a rohamok számáról betegenként.

A kutatásban résztvevő betegek kiválasztási folyamatáról, az egyének koráról, neméről, az általuk használt gyógyszerekről, a rohamok intenzitásáról, illetve általánosságban további adatairól információ nem áll rendelkezésemre.



4.1. ábra. A migrénes rohamok kezdeti időponjának eloszlása óránkénti felbontással.

4.2. Felhasználandó adatok kiválasztása

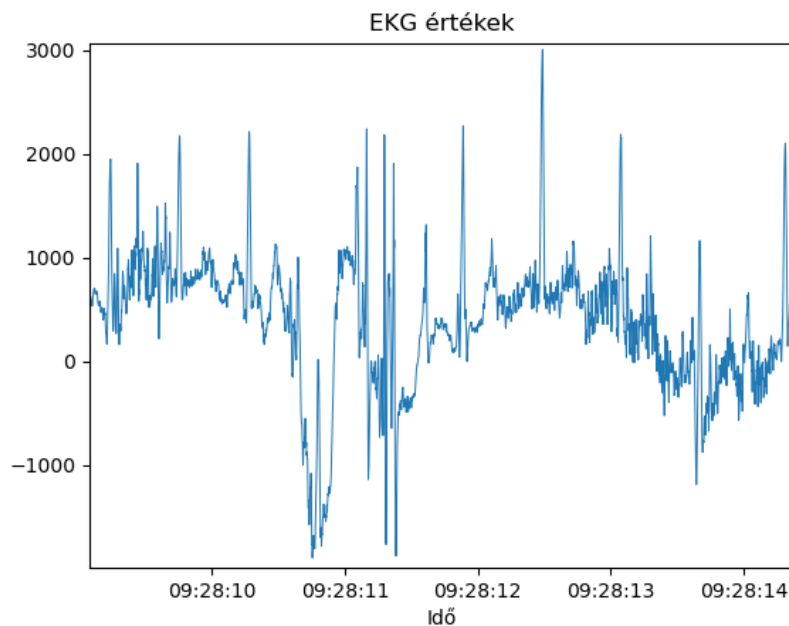
A túlságosan zajos adatok jelentős negatív hatással lehetnek a gépi tanulásra, ezért célszerű törekedni a minél tisztábbak kiválasztására a lehető legjobb eredmények érdekében.

Két különböző kategóriába sorolnám, hogy mit tekintünk jelen esetben zajnak. Az egyik természetesen a mért EKG jelre vonatkozó, amely megnehezíti, torzítja, vagy ellehetetleníti a szívdobbanások között eltelt idő detektálását. A másik minden olyan tevékenység, vagy fiziológiás folyamat, amely jelentősen befolyásolja a szívritmust és ezáltal elnyomja a számunkra releváns információt.

A [22] cikk megoldása erre a problémára az, hogy az alany alvása során összegyűjtött adatokat használja csak fel. Ezen megközelítés várhatóan tisztább EKG értékeket eredményez, illetve biztosítja azt, hogy az alany nem végzett aktív mozgással járó tevékenységet. Viszont hátránya, hogy ezen kompromisszum által rengeteg gyűjtött adatot hagyunk figyelmen kívül, illetve nem tudhatjuk, hogy az alvás hatása a szívritmusra mennyire fedik el a migrénes roham közeledtét jelző mintázatot. Mindenesetre az említett kutatás biztató eredményei alapján úgy döntöttem, hogy én is csupán az éjszakai adatokra támaszkodok. Az egyszerűség kedvéért az éjfél és reggel 6 óra közötti időintervallumot választottam, ahol feltehetően az alanyok nagy része pihen.

átlag	15,24
szórás	6,12
minimum	3
25. percentilis	11
50. percentilis	16
75. percentilis	18
maximum	33

4.3. táblázat. Statisztika az éjszakák számáról alanyonként.



4.2. ábra. Zajos részlet az egyik EKG felvételtől.

4.3. Nyers EKG jeltől az RR-intervallumokig

Az adatok feldolgozásának célja a nyers EKG adatokból meghatározni és elmenteni a szívdobbanások között eltelt időtartamokat. Ennek egyik lehetséges módja, hogy az EKG görbe csúcsait detektáljuk (az úgynevezett R-csúcsokat), amelyek időpontjait egymásból kivonva kapjuk meg az RR-intervallumokat, ami egyúttal a szívdobbanások között eltelt időkülönbségek szekvenciája.

Szeretném kiemelni, hogy ezen folyamat kidolgozását az E-Group ICT Software Zrt. munkatársai végezték, én csupán az általuk elkészített függvényeket használtam fel a saját előfeldolgozó modulomban¹, viszont ennek ellenére a teljesség érdekében nagy vonalakban ismertetem ezen alfejezetben. Ezzel kapcsolatos további részletek az általuk publikált [15] cikkben olvashatók.

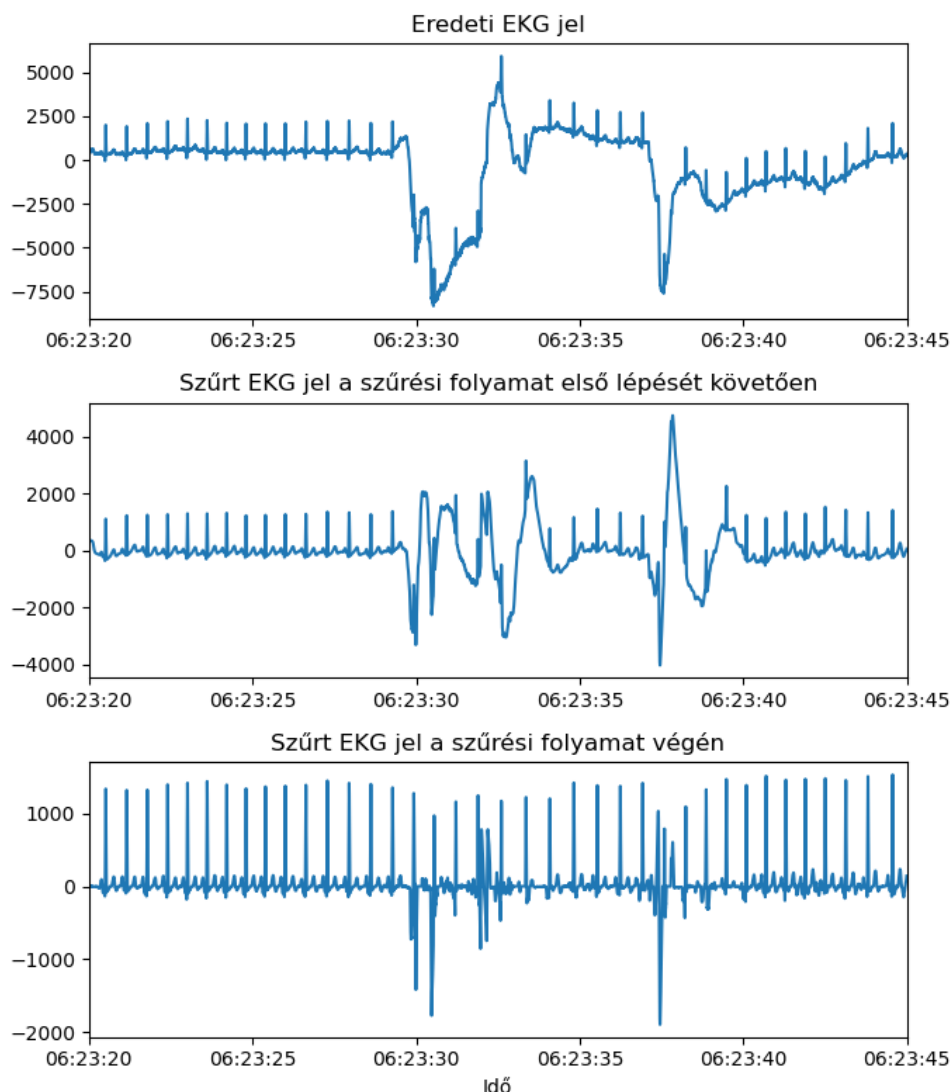
4.3.1. Jel szűrése

Annak érdekében, hogy az R csúcsok minél pontosabban detektálhatóak legyenek, szűrjük az EKG jelet. A 4.3. ábrán látható a folyamat hatása egy adott EKG részletre, amelyet az egyik, a kutatás során összegyűjtött felvételtől választottam ki.

Első lépésben egy 0,5 Hz-es kritikus frekvenciájú Butterworth-szűrőt, az 50 Hz-es zaj eltávolítására egy sávzáró szűrőt, majd a jel-zaj viszony további növelésére egy átlagoló szűrőt használunk.

Végül egy olyan szűrési eljárást alkalmazunk, amely az adott időablak mediánját vonja ki az aktuális értékből.

¹Természetesen a korábban meghatározott paramétereket én magam is állítottam annak érdekében, hogy a lehető legjobb megoldásra jussak, de mivel nem sikerült látványos javulást elérni, ezért végül maradtam az eredetnél.



4.3. ábra. EKG jel szűrési folyamata.

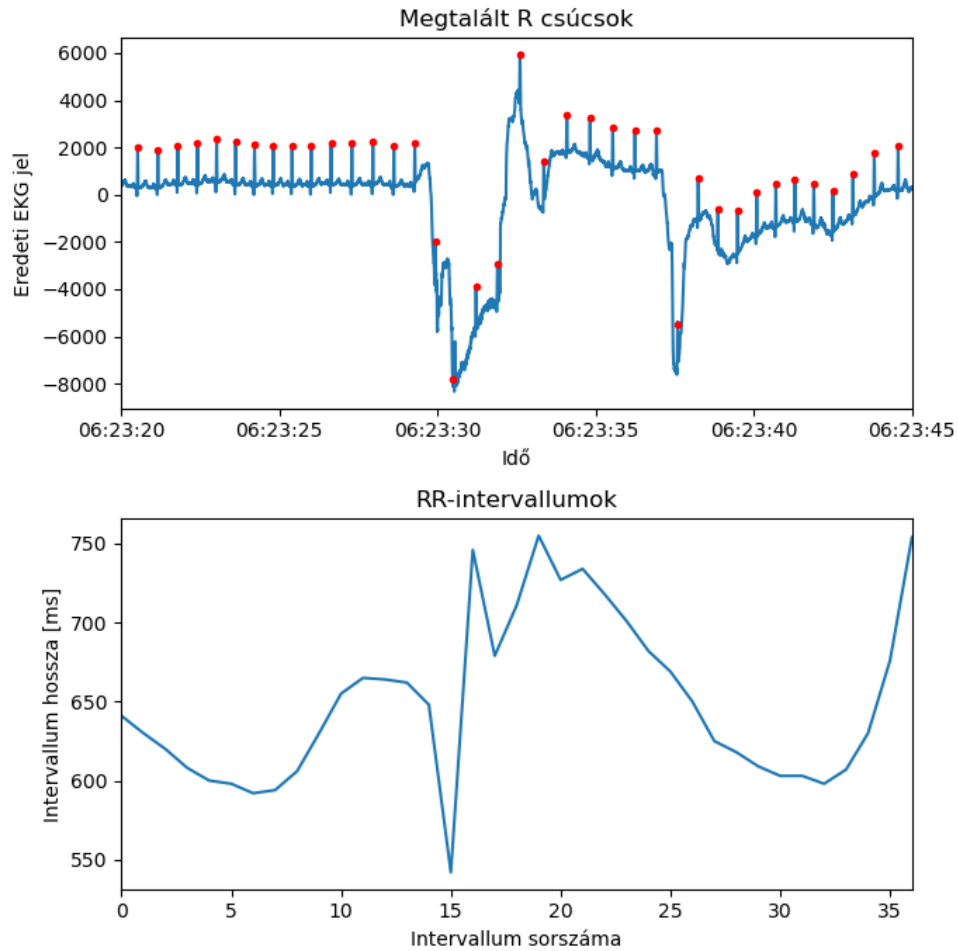
4.3.2. R-csúcs detekció és RR-intervallumok

Az utolsó lépés az R hullámok csúcsainak meghatározása a szűrt jelből, majd az egymást követő csúcsok időponjainak különbségét képezve meg is kapjuk a kívánt RR-intervallumokat. Ehhez az E-Group ICT Software Zrt. munkatársai összehasonlítottak számos különböző algoritmust az MIT arrhythmia adatbázison, majd implemetáltak egy egyedi megoldást a legjobbnak talált programkönyvtárat felhasználva.

A 4.4. ábrán láthatóak (a 4.3 és 4.4 ábrákon kirajzolttal azonos EKG felvételre vonatkozóan) a megállapított R csúcsok, illetve az RR-intervallumok. Ezen kis adatmennyiségen is láthatjuk, hogy annak ellenére, hogy komoly zaj terheli a jelet, az eljárás majdnem minden csúcsot közel megfelelően detektált.

4.4. Kiugró adatok kezelése

Annak ellenére, hogy az adatok feldolgozása alaposan megtervezett és leellenőrzött folyamat, nem tévedhetetlen az algoritmus az R-csúcs detekció tekintetében (feltételezhetően elsődlegesen az EKG értékeket terhelő zaj következményeként). Annak érdekében, hogy az



4.4. ábra. Az algoritmus által megtalált R csúcsok (felső) és a meghatározott RR-intervallumok (alsó).

ebből fakadó hibákat csökkentsem, a következőkben bemutatott módszereket dolgoztam ki.

4.4.1. Kiugró éjszakák szűrése

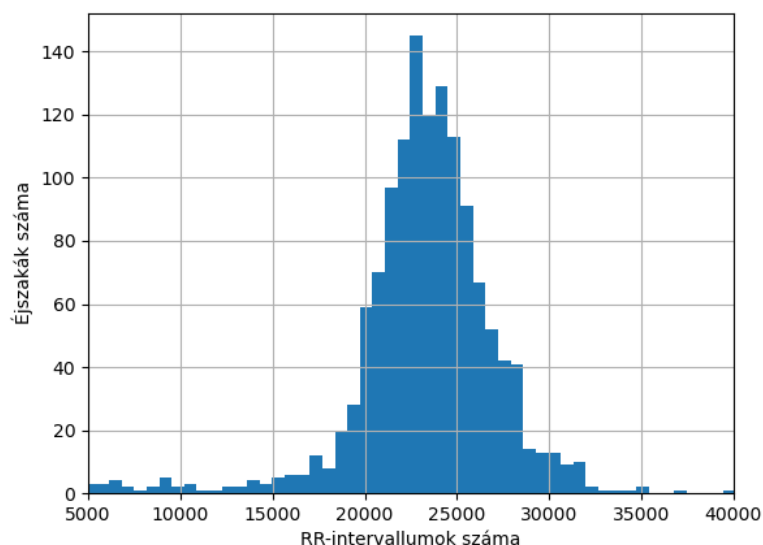
4.4.1.1. Szűrés az R-csúcsok száma alapján

A 6 órás EKG felvételen talált R-csúcsok száma könnyen megmutathatja számunkra, hogy ha valami nagyon nincs rendben a feldolgozásával, ezért első lépésben az így meghatározott kiugró adatoktól szabadultam meg.

A 4.5. ábrán látható az algoritmus által meghatározott RR-intervallumok számának eloszlása. Leolvasható, hogy körülbelül 22 500 darab R-csúcsot detektált általánosan az algoritmus, ami azt jelenti, hogy ezekben az esetekben átlagosan percenként kb. 60-at vert az illető szíve az adott éjjel. Az ábrát tovább vizsgálva, és számos éjszaka RR-intervallum szekvenciáját átnézve arra a következtetésre jutottam, hogy a 15 000 - 32 500 tartományon² kívül kevésbé megbízhatóak az adatok, vagyis ezeket nem használom fel a későbbiekben. Természetesen ezzel megvan a valószínűsége annak, hogy eldobok olyan éjszakát, ahol nem jelentős az előfeldolgozás hibája, viszont így az éjszakára vonatkozó átlagos pulzus

²Ez az intervallum a kb. 41 - 90 átlagos pulzusnak felel meg.

szempontjából extrém adatoktól is megszabadulok, ami remélhetőleg szintén jól tesz a modell tanításának tekintetében.



4.5. ábra. Egy-egy éjszakához tartozó RR-intervallumok számának eloszlása.

4.4.1.2. Szűrés a gyorsulásmérő értékek alapján

Ahogy az a 4.2. alfejezetben ismertetésre került, számos előnye miatt csak olyan adatokat szeretnék felhasználni, ahol az illető alszik, vagy legalább relatíve nyugalmi állapotban van. Ennek érdekében eddig csupán annyit tettem, hogy kiválasztottam az éjfél és 6 óra közötti intervallumokat a felvételekből, viszont nem vizsgáltam, hogy az illető valóban pihent-e ezen időtartam nagy részében.

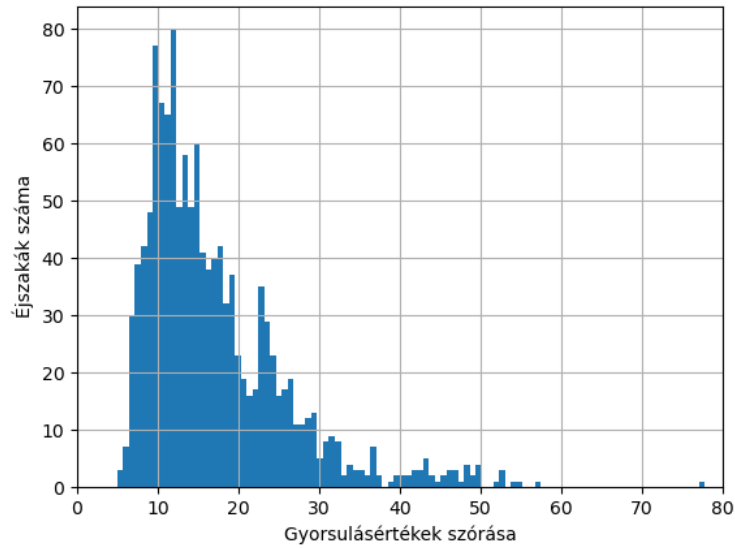
Az EKG felvétel során szintén monitorozott gyorsulásértékek erre biztosítanak lehetőséget. Annak érdekében, hogy a mozgásra vonatkozó értéket kapjak, először vettem a három dimenziós gyorsulásvektorok hosszát, majd kiszámoltam az értékek szórását az adott éjszakára³.

A 4.6. ábrán látható ezen értékek eloszlása. Leolvasható, hogy az éjszakák nagy része a 10 - 20 intervallumban van, illetve látható egy olyan eset, ami a 80 körüli értéket is eléri. Annak érdekében, hogy minél jobb döntést hozzak azzal kapcsolatban, hogy mi az elfogadható határérték, megvizsgáltam a konkrét RR-intervallumokat is. Azt tapasztaltam, hogy 50 fölött egyértelműen zajos az adat, de 40 - 50 között is meglehetősen, viszont a 30 - 40 intervallumban már jóval zajmentesebb görbék láthatóak. Ezáltal úgy döntöttem, hogy a 40-es szórásérték alatti éjszakákat használom csak fel a későbbiekben.

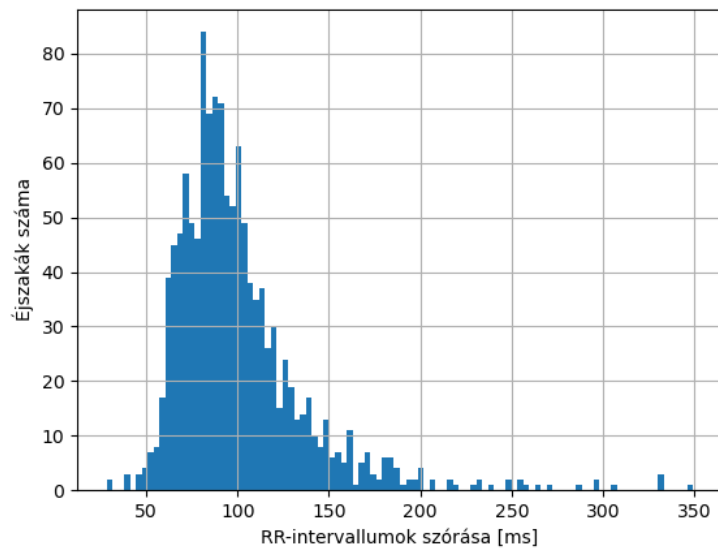
4.4.1.3. Szűrés RR-intervallumok szórása alapján

Miután az egy éjszakára vonatkozó RR-intervallumok száma és a gyorsulásértékek alapján megszabadultam számos kiugró éjszakától, megvizsgáltam az RR-intervallumok szórását is, ugyanis amennyiben annak értéke kiugróan nagy, akkor szintén felmerülhet, hogy valami nincs rendben az adattal.

³Természetesen ez egy kissé pongyola megoldás, de úgy véltem, hogy egy-két kiugróan intenzív mozgás detektálására elégséges.



4.6. ábra. Gyorsulásértékek szórásának eloszlása.



4.7. ábra. Az RR-intervallumok szórásának eloszlása.

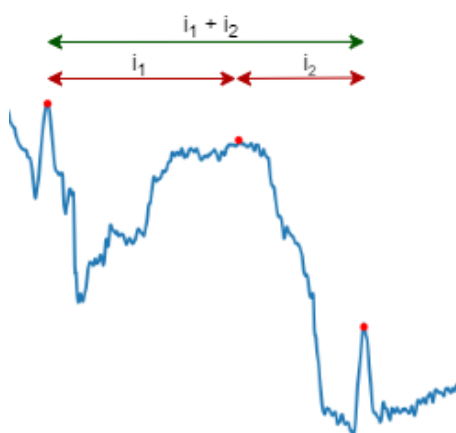
A 4.7. ábrán látható az RR-intervallumok szórásának eloszlása éjszakánként, amelyről azonnal leolvasható, hogy a szórásértékek jelentős része a 75 - 100 ms intervallumban van, viszont található akár 300 ms körüli értékkel rendelkező éjszaka is. Felmerülhet bennünk, hogy ugyan mi okozhatja ezeket a meglepően hatalmas értékeket. Találtam olyan éjszakát, ahol nagy valószínűséggel nagyon zajos lehetett az EKG jel, mivel meglehetősen kaotikusnak tűnt az RR-intervallum szekvencia, illetve olyat is találtam, ami alapján azt feltételezem, hogy az illető szívritmusa a normáltól eltérő (szabályszerűen néhány egymást gyorsan követő szívdobbanást egy lassabb követett). Természetesen az ilyen jellegű anomáliákat, illetve a kaotikus RR-intervallum szekvenciákat tartalmazó felvételektől jobbnak

láttam megszabadulni, ezért a 200 ms feletti szórásértékkel rendelkező estét nem használtam fel a későbbiekben.

4.4.2. Kisebb előfeldolgozási hibák kompenzálása

Annak ellenére, hogy az egész éjszakányi adatot nem torzítja el annyira, hogy érdemes lenne kidobni, egy-egy kiugró, extrém RR-intervallum érték negatívan befolyásolhatja a gépi tanulási modell tanítási folyamatát. Ennek megakadályozására a következőkben bemutatott megoldást dolgoztam ki.

Abban az esetben, ha kihagy az algoritmus egy R-csúcsot, vagy két valódi csúcs között hibásan detektál még egyet, akkor lehetőség van a hiba kompenzálására utólag. Az első esetben az intervallum kettéosztásával, míg második esetben a következő kettő összegzésével (a 4.8. ábrán látható módon) tehetjük ezt meg.



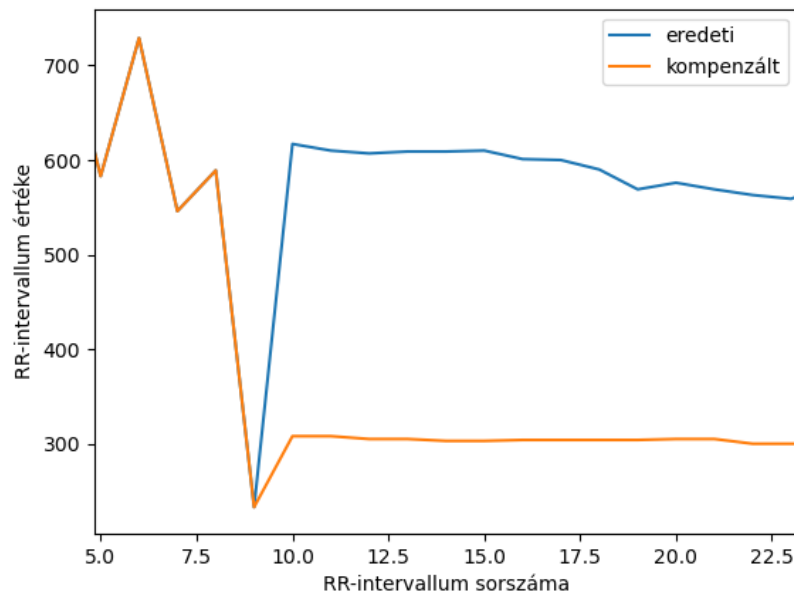
4.8. ábra. Hibás R-csúcs detekció kompenzálása.

Tehát az elkészítendő algoritmus működése a következő: amennyiben az adott RR-intervallum érték és az azt követő két érték összege közel azonos, akkor azok összegzéssel összevonásra kerülnek, míg amennyiben az adott érték és az azt követő fele közel azonos, akkor az utóbbit megfeleztve két intervallumba darabolja fel.

Ha a programot úgy implementálom, hogy végigiteráljon az RR-intervallumokon és amennyiben a korábban megfogalmazott feltételek valamelyike teljesül és végrehajtja a megfelelő műveletet, akkor komoly hibát vihet az adatba. A 4.9. ábrán egy ilyen eshetőséget láthatunk, ahol körülbelül 600-ról 300-ra csökken az intervallum, viszont mivel a következő megint 600-hoz közeli érték, ezért az összegzéses feltétel nem teljesül, viszont a duplájára nő, vagyis a felezési eljárást alkalmazza, amit folytat is hibásan. Ennek elkerülésére azt a megoldást választottam, hogy először vettem az egymást követő RR-intervallumok különbségértékeinek az abszolút értékét, majd csak azokra futtattam az algoritmust, ahol az említett különbségérték nagyobb, mint a különbségértékek 90. percentilise, tehát ebben az esetben csak a legkiugróbb 10%-ot van lehetősége maximum megváltoztatnia az algoritmusnak.

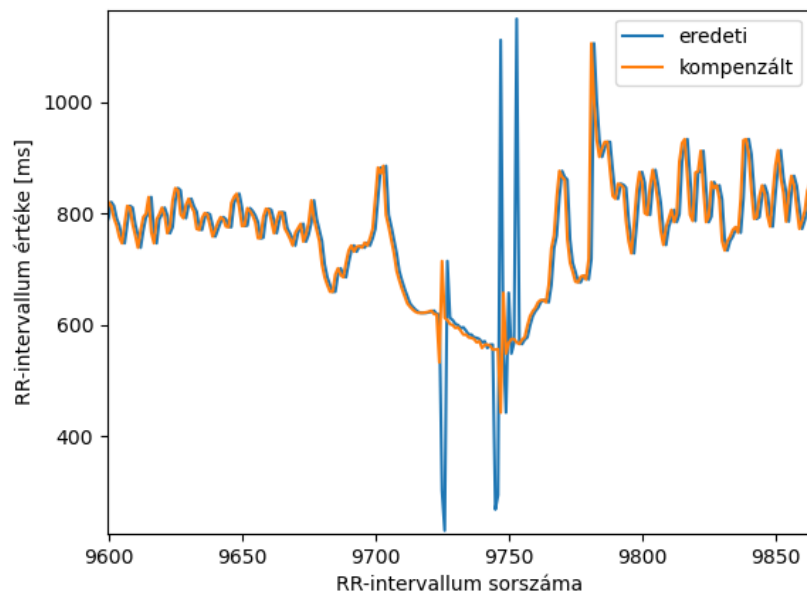
Annak meghatározására, hogy mi az a különbségérték, amivel közel azonosnak tekinthetünk két egymást követő RR-intervallumot, a korábban már ismertetett, 4.7. ábrára tekintettem vissza, amiről leolvasható, hogy az RR-intervallumok átlagos szórása valahol 75 - 100 ms között van. Ebből kiindulva én végül a 100 ms-ot választottam ezen határértéknek, mivel úgy tapasztaltam, hogy ezzel a beállítással sikerült a legjobb eredményeket elérni.

Emellett szeretném megjegyezni, hogy ez a megoldás nem minden esetben teszi tökéletessé az eredményt. Például, ha a hibás csúcsdetekció a következő csúcs előtt van



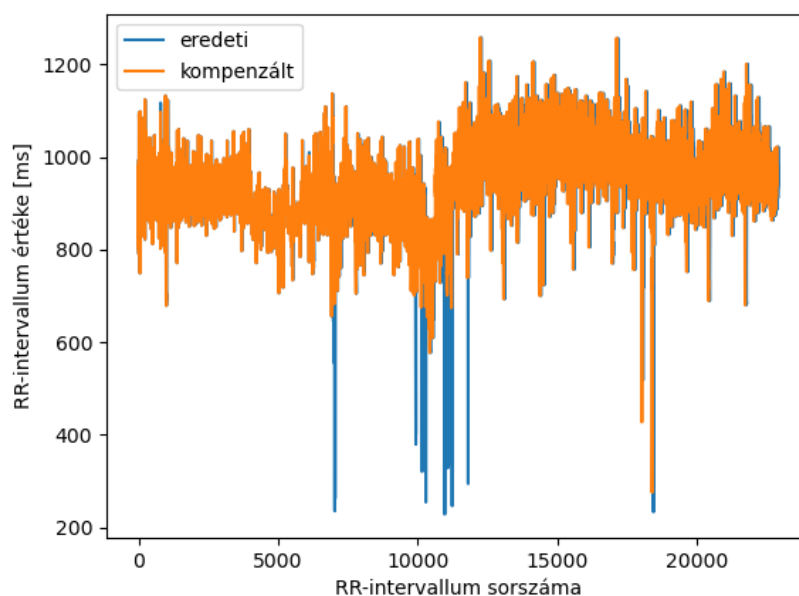
4.9. ábra. Kompenzáló algoritmus hibát visz az adatokba.

közvetlenül, akkor hibásan a következő két dobbanás között eltelt időkülönbséget növeli. Továbbá megvan annak is a valószínűsége, hogy az algoritmus megfelelően detektálta az R-csúcsokat, amik valamilyen oknál fogva kiugró értéket vesznek fel. Viszont azzal, hogy a kiugró adatok számát csökkentjük (még ha adott esetben ez a jó értékek torzulásával is járhat) tisztább adatokat és ezáltal jobb eredményeket remélhetünk.



4.10. ábra. Az elkészült kompenzáló algoritmus futtatásának hatása egy zajos adatrészleten.

Végül, mivel azt tapasztaltam, hogy továbbra is maradtak meglehetősen kiugró értékek az RR-intervallum sorozatban, ezért úgy döntöttem, hogy azokat, ahol az egymást követő intervallumok hosszának különbsége a legnagyobb 0,2%-ban van, azokat egyszerűen eldobom. Ez egy egész estés adaton, ami nagyságrendileg 25 000 RR-intervallumot tartalmaz, körülbelül 50 adatpont eldobását jelenti, amely remélhetőleg nem torzítja túlságosan az adatot, viszont a kiugró értékektől mentesen tisztább lesz.



4.11. ábra. Az elkészült megoldás eredménye egy 0 - 6 órás időszakra vonatkozólag.

4.4.3. Összegzés

Tehát a kiugró adatok csökkentésének érdekében alapvetően a következő két megközelítést alkalmaztam.

Egyrészt vizsgáltam az egyes éjszakákhoz tartozó különböző paraméterek eloszlását (RR-intervallumok száma, gyorsulásértékek, RR-intervallumok szórása) és ezáltal számos olyan éjszakát szűrtem ki, amelyek esetleg hibásan lettek feldolgozva, vagy valamilyen aspektusból extrém, vagy nem megfelelő a bennük lévő információ. Ennek eredményeként összesen 1222 éjszakányi adat maradt, összesen 83 betegtől.

Másrészt egy saját algoritmussal igyekeztem a feldolgozó program által vétett egy-egy hibás R-csúcs detekciót orvosolni, amely az ilyen szempontból kiugró értékek csökkentését szolgálja, mégpedig úgy, hogy az a lehető legkevesebb torzítást vigye az adatba.

4.5. Adatok felcímkézése

A felcímkézés során a feldolgozott és kiválasztott adatok a migrénes roham fázisai szerint kerülnek csoportosításra. Célja, hogy kiválasszuk azokat a felvétel részleteket, amelyekben feltételezhetően megtalálható az az információ, amely a közelebbi rohamot előrejelzi, illetve meghatározzunk úgynevezett kontroll szakaszokat, amelyekben a lehető legnagyobb valószínűséggel ugyanez nincs meg. Mivel a nem megfelelően felcímkézett adatok (mennyiségüktől függően) ronthatják, vagy akár teljesen tönkre is tehetik a gépi tanulási modell

tanítását, ezért minden bizonnyal joggal nevezhető ez a munkafolyamat egyik, ha nem a legkritikusabb lépésének.

4.5.1. Kontroll adatok kiválasztása

A kontroll csoportba szeretnénk választani azokat az estéket, amelyek nagy valószínűséggel nincsenek összefüggésben migrénes rohammal. Ehhez azt kell elsődlegesen figyelembe venni, hogy (ahogyan az a 2.1.2. pontban is ismertetésre került) a prodromális (megelőző) és posztprodromális (követő) tünetek feltehetőleg maximum egy-két napig tartanak, vagyis ebben az időszakban nem elhanyagolható az esélye annak, hogy az illető szervezetére hatást gyakorol a roham. Tehát amennyiben azokat az éjszakákat választjuk ezen csoportba, amelyek a roham előtt és után több, mint két nap távolságra vannak, akkor bízhatunk abban, hogy a tévesen ezen kategóriába sorolt adatok mennyisége nem lesz jelentős.

4.5.2. Roham előtti adatok kiválasztása

A roham előtti kategória meghatározása sajnos több nehézséget rejt magában. Ennek legfőbb oka, hogy még abban sem lehetünk biztosak, hogy létezik-e a keresett információ, nemhogy abban, hogy mikortól jelenik meg olyannyira az illető szívritmusában, hogy az kimutatható legyen. Ez az időtartam ráadásul egyénenként jelentősen eltérhet, ami további bizonytalanságot eredményez, illetve a rohamokra vonatkozó feljegyzések hiányosságai és esetleges pontatlanságai is nehezítik a feladatot.

A korábban többször hivatkozott [22] kutatás ezen problémát jelentősen leegyszerűsíti azzal, hogy azon éjszakákat, amelyeket követő nap során a beteg rohamot tapasztal, ezen csoportba választja. Én a saját megoldásomban óvatosabb megközelítést választottam, amely során azokat az éjszakákat sorolja az algoritmus a roham előtti kategóriába, ahol az adott napon az aznapi első roham kezdete a 4:00 - 12:00 intervallumba esik. Ezáltal kizártam azokat az éjjeleket, ahol a vizsgált éjjél és hat óra közötti időintervallum nagy részében fennáll a roham (ugyanis ekkor nem előrejelzés, hanem detekció története), illetve azokat, ahol csak délután kezdődik. Az utóbbit természetesen azért gondoltam szükségesnek, mert ezáltal csökkentem az eltelt időt az éjszakai adatok időtartama és a roham kezdete között, ezzel növelve a valószínűségét annak, hogy a keresett információ megtalálható az ezen kategóriába sorolt adatokban. Tehát a 4.2. pontban bemutatott megfontolásokhoz hasonló módon itt is kompromisszumot kell kötnünk abban, hogy a több, vagy a vélhetően tisztább adat mellett döntünk.

Roham kezdete	roham előtti	kontroll	kihagyott
06:00 - 12:00	101	556	555
04:00 - 12:00	114	556	542
04:00 - 23:59	187	556	469
00:00 - 23:59	194	556	462

4.4. táblázat. Éjszakák száma a különböző kategóriákban a migrén kezdete szerint csoportosítva.

A 4:00 óra választása elsőre talán meglepőnek tűnhet, viszont ahogyan az a 4.4. táblázatról is leolvasható, a roham előtti kategóriába tartozó éjszakák száma jóval kevesebb, mint a kontroll csoportba tartozóké, ezáltal úgy véltem, hogy a plusz 13 migrénes éjszaka (ami az intervallum kezdetének előbbre hozásával ezen csoportba került) több pozitív

hatással lesz a tanításra, mint amennyi pontatlanságot okoz azáltal, hogy már a vizsgált időszakban megkezdődik a roham⁴.

4.5.3. Megvalósítás

Tehát összefoglalva azt a döntést hoztam, hogy a migrénes roham kezdete előtt, illetve a végét követően maximum 2 nappal lévő éjjeleket kihagyom, továbbá csak azokat választom a migrénes csoportba, ahol az adott napon az első roham kezdete egy adott időintervallumba esik. Az ehhez tartozó, általam implementált programlogikát az alábbi, 4.1. pszeudokód ismerteti.

```
Function label_patient(seizures, interval = [4, 12]):
    for seizures_on_day in seizures.groupby(seizures.start.date):
        first_seizure, last_seizure = seizures_on_day[0], seizures_on_day[-1]
        for unused_day in [first_seizure.start.day - 2 : last_seizure.end.day + 2]:
            if not is_labeled(unused_day):
                label_day(unused_day, "unused")
        if first_seizure.start in interval:
            label_day(first_seizure.start.day, "migraine")
    label_rest("control")
```

4.1. lista. Az adatok címkézésének pszeudokódja.

4.6. Tanító és validációs halmazok elkülönítése

A gépi tanulási modell számára szokás elkülöníteni tanító és validációs halmazokat. Az előbbin tanul a modell, míg a másodikon értékeljük ki az eredményét. Lényegében két fontos döntést igényel a halmazok meghatározása: milyen logika alapján történjen a validációs halmazok elkülönítése és milyen arányban.

4.6.1. Az elkülönítő logika

Az egyik felmerülő lehetőség, hogy elkülönítsünk néhány beteget és azok adatait nevezzük ki validációs halmaznak. Ezzel azt teszteljük, hogy sikerült-e olyannyira általános érvényű mintázatot felfednie a modellnek, amely feltehetően új, a kutatásban nem szereplő egyénekre is alkalmazható. Ennek hátránya, hogy amennyiben nem elegendő az ezen csoportba választott adat, akkor az eredmény sem lesz megfelelően reprezentatív, továbbá nem ad arról visszajelzést, hogy a résztvevő alanyok újabb rohamjaira milyen minőségű lenne a predikció.

Ezzel ellentétben a másik opció, hogy úgy igyekezzünk adatokat megválasztani, hogy az minél inkább azt legyen hivatott reprezentálni, hogy az adott beteg, a modell számára ismeretlen éjszakáira vonatkozólag milyen eredmények születnének. Ebben az esetben jobb validációs eredményeket várhatunk azáltal, hogy vélhetően az azonos alanyhoz tartozó éjszakák egymással valamilyen szinten korreáláltak. Ezen megoldás azzal az előnnyel is jár, hogy azonos mennyiségű éjszakákat választhatunk a kontroll és roham előtti csoportokból, ami eredményeképp az olyan mérőszámok, mint a pontosság egyszerűbben kiértékelhetők.

Végül azt a döntést hoztam, hogy elsőként az utóbbi lehetőséget választom, mégpedig úgy, hogy azon betegek adataiból választok ki egyet-egyet a kontroll és roham előtti

⁴A 4.4. ábrán lévő értékeket megvizsgálva ellentmondásosnak érezhetjük, hogy a 4.1 pontban jóval több, 314 migrénes roham került említésre. Ez egyrészt és legfőképp annak köszönhető, hogy sajnos sok éjszakáról nincs összefüggő felvétel, így én azokat nem használtam fel, másrészt kisebb mennyiségben ugyan, de vannak olyan esetek, ahol egy nap több rohamot is tapasztalt az illető és természetesen azok azonos napra lettek besorolva, harmadrészt a kiugró adatok kezelése során is számos este került kidobásra, de ezek mennyisége sem túlságosan jelentős a roham előtti éjszakák számára vonatkozólag.

kategóriába tartozó éjszakák közül, melyekből a legtöbb áll rendelkezésre. Ezzel a megoldással várhatjuk a lehető legjobb eredményeket a validációs halmazon, ugyanis ebben az esetben a modell a lehető legtöbb másik éjszakai adatát megismerte a tanítási folyamat során. Amennyiben biztató eredményekre sikerül így jutni, akkor meghatározok még egy halmazt az első ismertett elkülönítő logika szerint, amely segítségével felmérhetem a modell populációs szintre vonatkozó általánosítóképességét.

4.6.2. A validációs halmazba választott adatok aránya

Összesen 114 roham előtti, illetve 556 kontroll éjszaka állt rendelkezésemre (lásd 4.4. ábra) a címkézést követően. Úgy véltem, hogy legalább 20-20 adatra lenne szükség mindkét kategóriából a validáláshoz, annak érdekében, hogy megfelelően reprezentatív eredményt kapjak, ami körülbelül a roham előtti adatok 17,5%-át jelenti.

Tehát a validációs halmaz elkülönítéséhez azt az eljárást alkalmaztam, hogy kijelölöm azt a 20 beteget, akiktől a lehető legtöbb adat áll rendelkezésemre (a címkézést követően), majd mindegyikből véletlenszerűen kiválasztottam egy-egy roham előtti és kontroll éjszakát.

Halmaz	roham előtti	kontroll
tanító	94	536
validáció	20	20

4.5. táblázat. Éjszakák száma az elkülönített halmazokban.

4.7. Adatok skálázása

Annak érdekében, hogy a gépi tanulási modell tanítása minél stabilabb és effektívebb legyen, érdemes az adatokat skálázni. Erre több megoldás is létezik, mint például az értékek egy bizonyos intervallumba transzformálása, vagy standardizálása.

Én az utóbbit választottam, amely alkalmazásával az adatok átlaga zérusra, szórása pedig egységre változik. Képlete az alábbi (4.1. -es ponttal jelölt), ahol x az eredeti adatsort, μ az átlagát, σ a szórását, y pedig értelemszerűen a standardizált értékeket jelöli.

$$y = \frac{x - \mu}{\sigma} \quad (4.1)$$

Ennek előnye (például az adott intervallumba való transzormálással szemben), hogy a kiugró értékek nem csökkentik le nagyon az adatok varianciáját. A skálázás ezen tulajdonsága kifejezetten fontos jelen esetben, ugyanis tartalmaznak kiugró értékeket az RR-intervallum szekvenciák annak ellenére, hogy igyekeztem nagyon alaposan megtisztítani (lásd 4.4.2) az adatokat.

Mi elsődlegesen azt keressük, hogy mi a különbség az RR-intervallumokban egy adott beteg migrénes roham előtti időszakában egy kontroll időszakhoz képest, ezért számunkra az egyénenkénti szívritmusbeli különbségek nem relevánsak. Ebből kiindulva felmerült bennem, hogy a skálázást egyénenként alkalmazom, ami által az alanyok közti különbségeket csökkenthetem, ezzel megkönnyítve a gépi tanulási modell feladatát. Viszont végül elhatárolódtam ettől a megoldástól, ugyanis tartottam attól, hogy a következő két dolog következtében a megoldás inkább negatív hatást eredményez. Egyrészt, ha egy betegről csak kevés éjszakányi (mondjuk extrém esetben csak egyetlen adat áll rendelkezésre), ak-

kor a skálázási paraméterek nem lesznek reprezentatív jellemzők az egyén RR-intervallum szekvenciáira, csak arra az egy-két estére. Másrészt tartottam attól, hogy azáltal, hogy egyes betegek RR-intervallumainak szórása túlságosan különbözik, azáltal túlságosan különbözhetnek a skálázási paraméterek egyénenként.

4.8. Összefoglalás

A munkafolyamat első lépésben implementáltam egy adatafeldolgozó programot (részben a rendelkezésemre álló megoldás alapján), amely a nyers EKG adatoktól eljut a kiválasztott, tisztított, címkézett, skálázott RR-intervallum értékekig, amik a modell bemeneteként szolgálnak. Először sok időt szántam a feldolgozó program futásidejének optimalizálására és a létrejövő hibák kezelésére a robusztus működés érdekében. Ezt követően kiemelt figyelmet fordítottam a kiugró adatok kiszűrésére, illetve a hibák kompenzálására. Emellett újragondoltam a [22] cikkben használt megoldást az éjszakák kategorizálására vonatkozóan és úgy döntöttem, hogy roham előtti estének azt választom, ahol másnap délelőtt van a roham kezdete, illetve a kontroll csoportba azon éjszaka kerül, ami több mint két nap távolságra van az előtte lévő roham végétől és a következő kezdetétől. Végül az adatokat tanító és validációs halmazokra bontottam fel, majd az általam kiválasztott eljárással skáláztam a modell effektívebb tanításának érdekében.

Úgy vélem, hogy a létrejött feldolgozóprogram ezen lépések által alaposan átgondoltan készült el, illetve a kiválasztott programkönyvtárak felhasználása és a különböző optimalizációs technikák által egy hatékony és flexibilis megoldást sikerült elkészítenem, ami megfelelő alapként szolgált a további munkafolyamat számára.

5. fejezet

Mesterséges intelligencia alapú megközelítések

A kutatók rövid RR-intervallum szekvenciákból igyekeztek olyan integrális paramétereket meghatározni, amelyek összefüggésbe hozhatók a roham előtti időszakkal. Az ezen megközelítéssel elért kezdeti, nem túl biztató eredményeket követően kidolgoztam egy egyéni adatfeldolgozó, tisztító és címkéző eljárást a korábbi, meglévő megoldást továbbfejlesztve, illetve úgy döntöttem, hogy elhatárolódok ezen integrális paraméterektől és tisztán az RR-intervallum szekvenciákat felhasználó modellt igyekszem megalkotni.

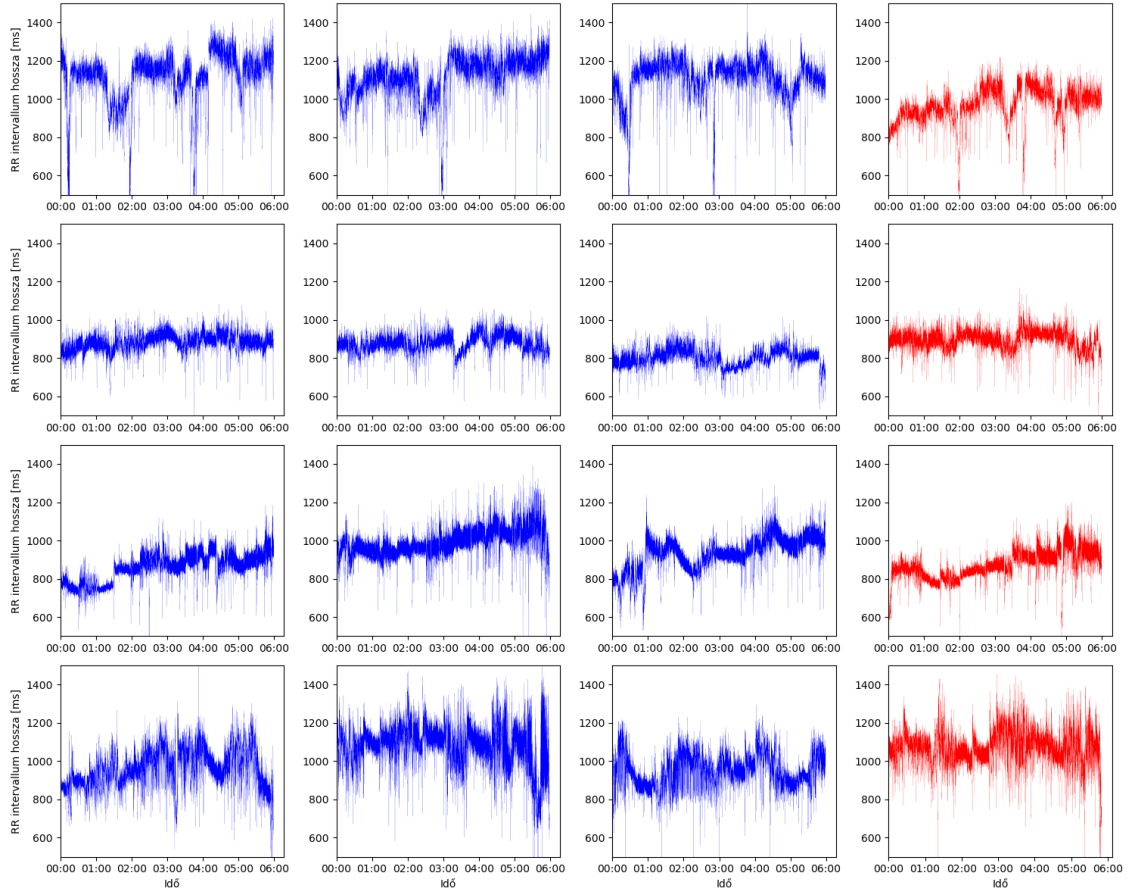
5.1. Adatok megvizsgálása

Mielőtt a tanítási folyamat, illetve a modell architektúra tervezésének részleteit ismertetném, szeretném röviden összefoglalni, hogy az előkészített adatok megvizsgálása során milyen alapvető, egyszerű következtetésekre jutottam. Ezzel igyekeztem megbecsülni a gépi tanulási modellre váró feladat komplexitását, illetve felmérni, hogy mit tartok elképzelhető megközelítésnek és nem melleleg becslést tenni arra vonatkozólag, hogy milyen eredményt tarthatok reálisan elérhetőnek.

Az 5.1. ábrán látható néhány tetszőlegesen kiválasztott kutatási alany 4 éjszakájának RR-intervallum sorozata, külön színnel megjelölve a migrénes roham előtti, illetve kontroll kategóriába soroltakat. Az ábra jól szemlélteti, hogy az egyes alanyok szívritmusa mennyire eltérő, illetve azt is, hogy azonos beteg két különböző éjszakáján mennyire hasonló a szekvencia. Végül, ami a legfontosabb, hogy ez alapján (illetve sok másik alany adatát megvizsgálva) arra a következtetésre jutottam, hogy triviális különbséget ránézésre a migrénes és kontroll éjszakák között nem lehet egyszerűen felfedezni.

Ezen megállapításból több minden is következik. Egyrészt az RR-intervallumok egész estére vett átlaga és mediánja minden bizonnyal nem egyértelmű indikátorai a rohamnak, ugyanis ellenkező esetben ez vélhetően leolvasható lenne az ábrákról. Másrészt hosszabb, akár egész éjszakára vonatkozó tendenciabeli egyértelmű eltérések sincsenek a kontroll és migrénes adatok között¹, ezáltal a több, mint egy-két óra periódusidejű frekvenciatartománybeli komponensek ezen szemponból vett hasznossága is megkérdőjelezhető. Harmadrészt sajnálatos módon az adatok szórásával kapcsolatban sem vélhetünk felfedezni valamilyen nyilvánvaló mintázatot, hanem sokkal inkább az figyelhető meg, hogy nagyon eltérhet egyénenként, ezzel rámutatva, hogy a populációsintű predikció miért is nehéz feladat.

¹Ilyen lehet a szívdobogás folytonos, lassú gyorsulása illetve lassulása.



5.1. ábra. Kontroll (kék) és roham előtti (piros) csoportba tartozó egész éjszakás RR-intervallumok, ahol egy-egy sorhoz egy-egy kutatási alany tetszőlegesen kiválasztott éjszakái tartoznak.

5.2. A modell kimenete és a hibafüggvény

Az elkészítendő modell célja meghatározni, hogy roham előtti, vagy kontroll csoportba tartozik az aktuális bemenet. Ezáltal kézenfekvő úgy megalkotni, hogy egyetlen, 0 és 1 közötti számérték legyen a kimenete, amely azt jelenti, hogy az aktuális bemenet a modell szerint milyen valószínűséggel tartozik a migrénes kategóriába. Annak érdekében, hogy a kimenet a kívánt (0, 1) intervallumba tartozzon, a modell kimenetén található transzfer függvénynek a szigmoid-ot választottam.

A tanításhoz szükséges definiálnunk egy hibafüggvényt, amely meghatározza a kimenet jóságát, majd ezt az értéket felhasználva állítja az algoritmus a modell paramétereit. Én ezen célra az úgynevezett *binary crossentropy* függvényt választottam, amely a korábban említett szigmoid függvény használata esetén alkalmazandó.

5.3. Egyszerű megközelítés

5.3.1. Bemenetek és architektúra

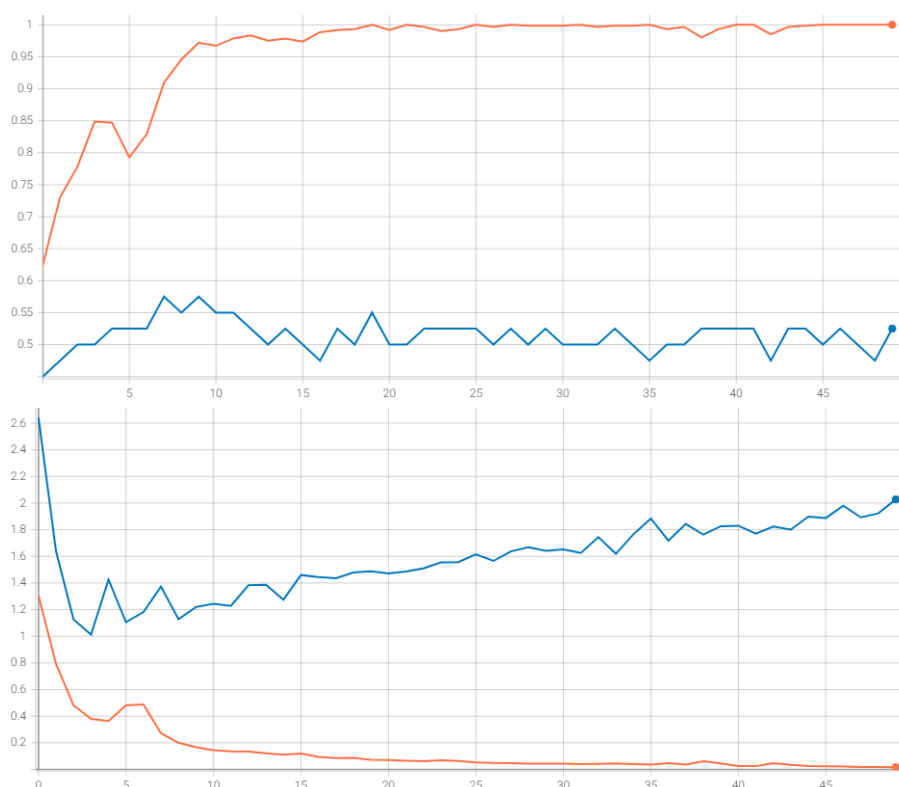
A talán legegyszerűbb megoldás, hogy a rendelkezésünkre álló, egy-egy éjszakányi RR-intervallum szekvencián egyesével, egymástól függetlenül tanítjuk a lehető legegyszerűbb neurális hálózatot, amelyet mindösszesen egyetlen perceptron alkot.

Ezen architektúra esetén a bemeneteknek azonos hosszúnak kell lenniük, ezért mindegyik RR-intervallum sorozat közepéből kivágtam egy ablakot (mivel az éjjél - 6 óra intervallum közepén alszik a legnagyobb valószínűséggel az illető, tehát feltehetően ott a legkevésbé zajos az adat), amely a legrövidebb bemenet hosszával egyenlő.

Mivel a legekevesebb RR-intervallumot tartalmazó éjszaka kb. 15 000 elemből áll, ezért a perceptron modellnek ezzel megegyező számú bemenete lesz, illetve egyetlen kimenete, vagyis körülbelül 15 000 tanítható paraméter² alkotja.

5.3.2. Eredmények

Az 5.2. felső ábrán látható a tanítás eredménye a pontosság szempontjából, amelyről egyértelműen leolvasható, hogy nagyon gyorsan és effektíven tanult a modell a tanító adaton, viszont a validációs halmazon elért pontossága nem mutatott hasonló tendenciát. Észrevehetjük emellett, hogy az első néhány tanítási ciklus végén a validációs halmazon 50% fölé nőtt a pontosság, sőt a legjobb esetben az 57,5%-ot is elérte az ugyanezen mérőszám értéke, ami azt jelenti, hogy a 40 éjszakából 23-at helyesen kategorizált. Az, hogy egy ilyen egyszerű megoldással már képes olyan mintázatot találni a modell a roham előtti és kontroll adatokban, ami a validációra is minimális mértékben alkalmas, elsőre akár nagyon biztató eredménynek is tűnhet, de mielőtt elhamarkodottan messzemenő következtetéseket vonnánk le ezekből a számokból, érdemes kicsit alaposabban megvizsgálni az eredményeket.



5.2. ábra. Pontosság (felső) és loss (alsó) értékek a tanító (narancssárga), illetve validációs (kék) adatokon a tanítási iterációk függvényében.

Az 5.2. alsó ábrán szerepel a kimeneti hiba átlagos értéke, ami árnyaltabb képet biztosít számunkra, ugyanis arról ad visszajelzést, hogy mennyire volt távol a roham előtti

²Egy perceptron paramétereinek száma a bemenetek számánál egyel több, amennyiben a konstans eltolás is engedélyezett, ellenkező esetben pedig azonos vele.

éjszakára vonatkozó kimenet a kívánt 1-es értéktől, illetve ellenkező esetben a 0-tól. Látható például, hogy az imént említett 57,5%-os pontosságú esetben a hiba mértéke közel azonos volt az 50%-ossal, vagyis annak ellenére, hogy a kimeneti értékek némelyike jó irányba billent a 0,5-ös határon, összességében kevésbé magabiztos jó, illetve magabiztosabb hibás becslést adott. Tehát hiába láthatunk 50% feletti pontosságot, ha az más szempontból vizsgálva nem is jelent jobb eredményt.

Mivel a kiválasztott 40 validációs este egyértelműen kevés ahhoz, hogy a pontosság, illetve egyetlen kísérlet megfelelően reprezentatív eredményt szolgáltatson, ezért többször megismételtem a tanítást másik estét választva validációra. Számos esetben a validációs pontosság el sem érte az 50%-ot, viszont ami minden esetben azonos volt, hogy már a legelső néhány (vagy akár a legelső) tanítási ciklus végétől fokozatosan nőtt a validációs hiba értéke. Tehát amit egyértelműen láthatunk, az a túltanulás jelensége, ami annyit tesz jelen esetben, hogy a mintázat, ami alapján a modell elkülöníti a migrénes és kontroll éjszakákat, az csak a tanító halmazon megfelelő, általánosan nem. Tehát hiába jut oda a tanítás, hogy a kb. 630 éjszaka nagy részét megfelelően sorolja be a modell, nagy valószínűséggel új predikcióra nem alkalmas a megoldás. Ennek oka, hogy az éjszakákat könnyebben különíti el a zaj, mint a releváns információ, ami előzetes félelmeinket beigazolván valóban nem egyszerűen fellelhető.

5.4. Tanítási folyamat fejlesztése

5.4.1. Általánosító képesség növelése

Tehát elsődleges problémánk, hogy a modell általánosító képessége nem megfelelő. Ezen egyrészről egy jobb architektúra megválasztásával (amire a későbbiekben kerül majd csak sor), illetve a tanítóadatok augmentálásával lehet segíteni. Az utóbbira vonatkozó megfontolásokat szeretném röviden összefoglalni ebben a pontban.

Összesen néhány száz éjszaka áll rendelkezésünkre tanítási célra, ami a korábban bemutatott folyamatban azt jelenti, hogy a modell összesen ezzel megegyező számú különböző bemenettel találkozik. Amennyiben mindig a közepét vágjuk ki az RR-intervallum szekvenciáknak, akkor egyrészt elvesztjük vele azokat az adatokat, amik ezen az ablakon kívül esnek, másrészt egy adott éjszakára vonatkozóan mindig ugyanahhoz a bemenethez ugyanazon súly tartozik. Annak érdekében, hogy ez ne így történjen, egy olyan megoldást készítettem, amely képes arra, hogy a tanítás során minden alkalommal véletlenszerűen kerüljön kiválasztásra a paraméterként megadható méretű ablak az adott szekvenciából.

Annak meghatározására, hogy ezen megoldással hány különböző bemenete lehet a modellnek az 5.1-es egyenlet szolgál, amelyben N az éjszakák száma, l az adott éjszakához tartozó RR-intervallumok száma, míg w a választott ablakméretet jelöli (és az *avg* címke természetesen az átlagértékre vonatkozik).

$$y = \sum_{n=1}^N (l_n - w + 1) = N \cdot (l_{avg} - w + 1) \quad (5.1)$$

Az említett egyenletbe az 5.1-es táblázatból a megfelelő értékeket behelyettesítve azt kapjuk, hogy az összes lehetséges bemenet száma $671 \cdot (23536 - w)$. Ez azt jelenti, hogy a lehető legnagyobb ablakméretet (15044) választva körülbelül 5 700 000, míg mondjuk 1024-es ablakmérettel körülbelül 15 100 000 különböző bemenet lehetséges³.

A véletlen ablakozás mellett az egyik gyakran alkalmazott megoldás, hogy az idősoros adatokat véletlenszerű zajjal terhelik minden egyes tanítási iterációban. Ez az eljárás szín-

³Szeretném hangsúlyozni, hogy itt a bemenetek nem függetlenek egymástól.

éjszakák száma	671
átlag	23 535
szórás	2 785
minimum	15 044
maximum	32 312

5.1. táblázat. Statisztika az RR-intervallumok számáról éjszakánként.

tén növelheti a modell általánosító képességét, viszont amennyiben elnyomja a releváns adatot, akkor az kontraproduktív is lehet. Ezen oknál fogva végül úgy döntöttem, hogy az egyébként is zajos adatot további, additív zajjal nem terhelem.

5.4.2. Validáció reprezentativitásának növelése

Annak érdekében, hogy a validáció valamivel reprezentatívabb eredményt szolgáltatson, úgy döntöttem, hogy az előző (5.4.1 pontban) ismertetett eljáráshoz hasonlóan itt is felúsítom az adatokat ablakozással. Azért, hogy a validációs halmaz ne változzon a tanítási folyamat során, itt nem a véletlenszerűséget alkalmaztam, hanem előre legeneráltam minden adatot. Emellett úgy is véltem, hogy az összes lehetséges kombináció túl nagy validációs halmazt generálna, ezért meghatároztam egy lépésközt az ablak csúsztatására vonatkozóan, amivel csökkenthettem az adatok számát.

5.4.3. Kategóriák kiegyenlítetlenségének csökkentése

Jelen esetben jóval kevesebb migrénes este áll rendelkezésünkre, mint amennyi kontroll, ami negatív hatással lehet a tanításra. Ennek oka egyszerűen az, hogy a modell hibásan arra tanulhat rá, hogy mindig a kontroll estét prediktálja, ugyanis ezzel egészen jól csökkentheti a hiba értékét.

Ez ellen az egyik lehetőség, hogy súlyozzuk a kategóriákat, amit a hibaszámításnál vesz a tanító algoritmus figyelembe, ezáltal a paraméterek állításánál nagyobb szerepet kaphat a kisebb mintaszámmal rendelkező osztályra vonatkozó hiba értéke.

En ezen megoldás helyett inkább úgy döntöttem, hogy egy talán szebb, általam kidolgozott megoldást alkalmazok, amely azon az elgondoláson alapszik, hogy minden egyes mini batch-be és epoch-ba azonos mennyiségű migrénes és kontroll adat legyen. Ezt úgy valósítottam meg, hogy minden epoch-ba mindegyik roham előtti éjszakából kerül adat, viszont a kontroll éjszakák közül minden alkalommal véletlenszerűen, egyenletes eloszlással választ a program, majd ezeken iterál végig úgy, hogy minden batch-be azonos mennyiségű adat kerüljön a két kategóriából.

Érdemes kiszámolni, hogy átlagosan hány epoch-ra van szükség ahhoz, hogy mindegyik kontroll estéből legalább 1 részletet kapjon a modell. Ehhez az 5.2 képlet alkalmazható, amely meghatározza, hogy egyenletes eloszlás esetén átlagosan hány kísérletre van szükség ahhoz, hogy mindegyik lehetőség egyszer szerepeljen, ahol N a lehetőségek száma.

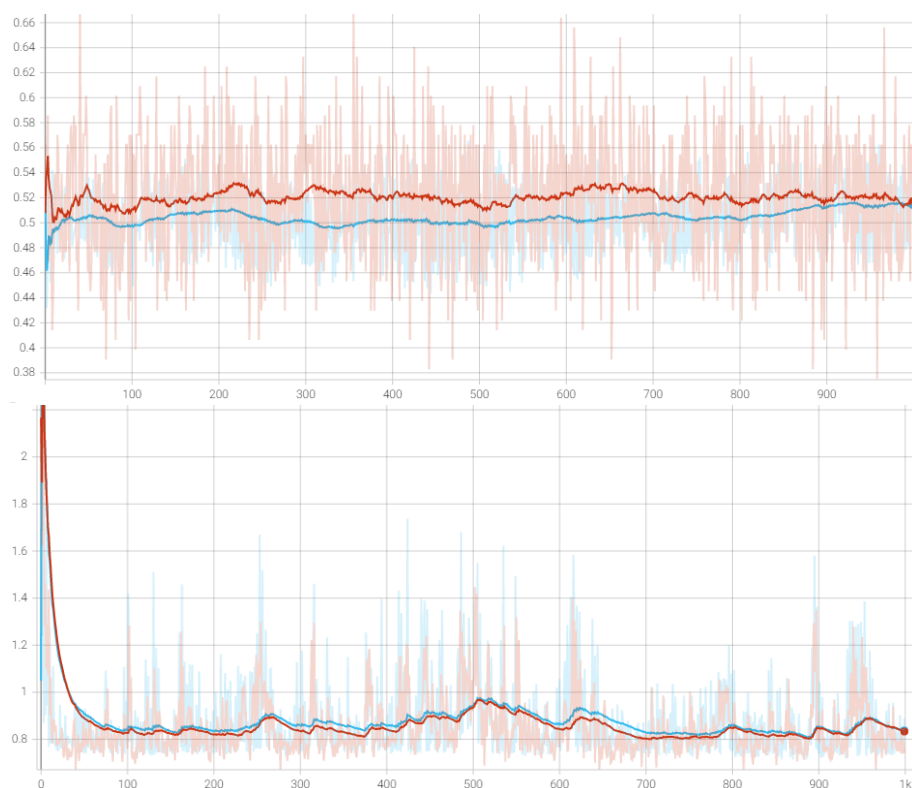
$$k = \sum_{n=1}^N \frac{N}{n} \quad (5.2)$$

Tehát a kontroll éjszakák számát a képletbe behelyettesítve megkapjuk, hogy hányszor kell véletlenszerűen választani a kontroll esték közül átlagosan, hogy legalább egyszer az összes szerepeljen. Ahhoz, hogy az átlagosan szükséges epoch-ok számát kapjuk meg, el kell osztani az eredményt az egy epoch során választott esték számával, ami megegyezik a roham előtti éjszakák számával (ugyanis minden epoch-ban mindegyik roham előtti éjszaka

szerepel). Behelyettesítve a 4.5 táblázatban található adatokat azt kapjuk, hogy átlagosan körülbelül 39 epochra van szükség ehhez.

5.4.4. Változtatások hatása

Az 5.3. ábrán látható az előző változtatások hozzáadását követően, a korábbival azonos architektúrájú, egyetlen perceptront tartalmazó modell tanításának eredménye a korábbival azonos ablakméret esetén. Az 5.2 ábrával összevetve, a vártnak megfelelően azt tapasztalhatjuk, hogy megnövekszik a pontosság és a kimeneti hiba szórása (mivel minden tanítási ciklus során más bemenetet kap a modell), illetve lassul a tanulás hatékonysága. Végül azt olvashatjuk le az ábrákról, hogy a pontosság mindkét halmazon 50% környékén maradt mindvégig, illetve a hiba sem csökkent le egy adott szint alá. Mindebből arra következtethetünk, hogy a modell túl egyszerű, tehát egy több paraméterű architektúrával érdemes próbálkozni a későbbiekben.



5.3. ábra. Pontosság (felső ábra) és loss (alsó ábra) értékek a tanító (piros), illetve validációs (kék) adatokon a tanítási iterációk függvényében. Az erősebb szín a mozgóablakos átlagot jelenti a könnyebb átláthatóság érdekében.

5.5. Paraméter alapú megközelítés

5.5.1. Megközelítés

Tehát a tanítási folyamat fejlesztését követően arra a konklúzióra jutottam, hogy érdemes az eddigi egyetlen perceptront tartalmazó architektúrát továbbfejleszteni. Az egyik lehetőség, hogy a meglévő modellt egyszerűen többrétegű perceptronná alakítjuk, vagy bevezetünk olyan rétegeket, amelyek segítségével jobb eredményeket remélhetünk. Emellett eddig nem foglalkoztam a megfelelő ablakméret megválasztásával sem, amelyet szintén

meg kell határozni az effektív továbblépéshez. Tehát lényegében a következő két kérdésre kerestem a választ: a megoldandó problémához milyen modell architektúra és mekkora ablakméret illik leginkább.

Ezek megválaszolásához visszatérnék oda, hogy a megközelítés alapja az a hipotézis, amely szerint az RR-intervallumok szekvenciájából ki lehet nyerni olyan integrális értékeket, amelyek előrejelzik a közelgő rohamot. Ez alapján érdemes lenne olyan architektúrát választani, amely ezt reprezentálja valamilyen formában. Éppen ezért úgy döntöttem, hogy a modell bemenetén konvolúciós rétegeket alkalmazok, amelyek a bemenő adaton több nemlináris szűrést hajtanak végre (ezzel meghatározva a kívánt paramétereket), majd a kimenetére egy többrétegű perceptront illeszték, amely leképezi a kimenetre ezeket az értékeket.

Az ablakméret és architektúra részleteinek megválasztása nem könnyű feladat, amelyhez számos dolgot kell végiggondolni. Egyrészt az említett HRV paraméterek már 5 - 10 percnyi RR-intervallumból jól számíthatóak. Ez azt jelenti, hogy ez alapján körülbelül 300 - 600 RR-intervallum már elegendő lehet, viszont fontos figyelembe venni, hogy a körületekintő adatfeldolgozás és tisztítás ellenére komoly zaj terhelheti az adatot, amely negatív hatását a nagyobb időablak feltételezhetően csökkenteni képes. Másrésztől intuitíve felmerülhet bennünk, hogy a rohamot megelőző éjszaka során a roham kezdeti időpontjához közeledve egyre erősebben jelentkezik az adott mintázat. Ezáltal elképzelhető, hogy a paraméterek változásának tendenciája kifejezetten releváns, illetve azt is jelenti, hogy adott esetben az éjszaka elején még nem is igazán fellelhető a keresett információ.

Tehát összességében arra a konklúzióra jutottam, hogy relatíve nagy (több órányi adatot magába foglaló ablakmérettel) érdemes próbálkozni, illetve az alkalmazott első szűrők által lefedett időtartam az 5-10 perc nagyságrendjébe tartozzon, amelyek segítségével meghatározott paraméterek tendenciáját kapja meg a modell végén található többrétegű perceptron.

5.5.2. Modell architektúra

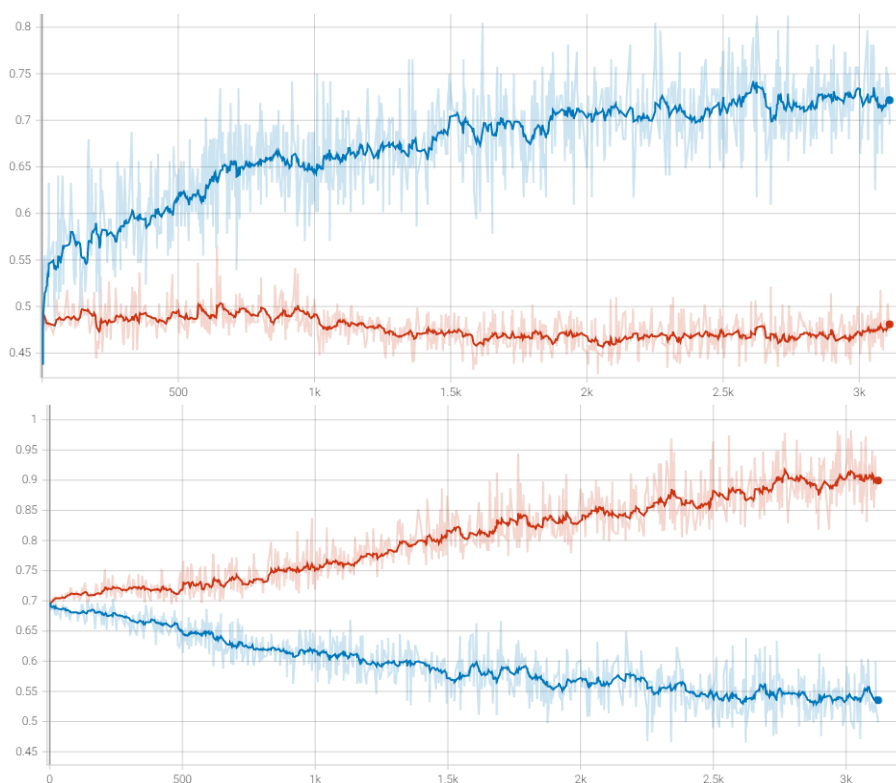
Végül a $2^{12} \cdot 3 = 12288$ széles ablakot választottam a bemenő adat számára, ami hozzávetőlegesen az 0:00 - 6:00 óra közötti időtartam felét foglalja magába. Az elkészült architektúra az 5.2 táblázatban látható. A bemeneten 2 darab konvolúciós réteg található, amelyek szűrői 32 hosszúak, vagyis $32 \cdot 32 = 1024$ egymást követő RR-intervallumra vannak hatással egyszerre, amely körülbelül 15 - 20 percnyi adatot jelent. Ezáltal teljesül az elképzelés, miszerint az alkalmazott szűrési eljárás a HRV paraméterek meghatározásához szükséges időablakból történjen. Ezt követően egy átlagoló réteg a létrejött 32 darab szűrt adatot egyenként 32 részre darabolja fel és azok átlagát képezi, ezzel meghatározva 32 paraméter 32 adatból álló tendenciáját. Végül mindezeket együtt, vagyis $32 \cdot 32 = 1024$ paramétert kap meg a multilayer perceptron, amely először 32 dimenzióba transzformálja le a megkapott információt, majd végül az egy dimenziós kimenetre. A modell szabadságfokát illetően összesen 100 513 állítható paraméterrel rendelkezik.

5.5.3. Eredmények

A tanítás eredménye az 5.4 ábrára kirajzolt grafikonokon látható. Leolvasható, hogy képes volt tanulni a modell, fokozatosan nőtt a tanító halmazon való pontosság és csökkent a hiba. Viszont az alsó, hibaértékekre vonatkozó ábra mutatja a legjobban, hogy már a legelső tanítási ciklust követően a validációs halmazra vonatkozó hiba folyamatosan nőtt, vagyis már a tanítás legelejétől túltanult a modell, még akkor is, ha akkor csak 50% körüli pontosságot ért el mindkét halmazon.

Réteg neve	Kimeneti dimenziója	Paraméterei
Input	12 288 x 1	-
Conv1D	12 257 x 64	kernel méret: 512, szűrők száma: 64
ReLU		-
Conv1D	12 226 x 32	kernel méret: 32, szűrők száma: 32
ReLU		-
AveragePooling1D	32 x 32	méret: 12 226 // 32
Flatten	1 024	-
Dense	32	méret: 32
ReLU		-
Dense	1	méret: 32
Sigmoid		-

5.2. táblázat. Implementált konvolúciós hálózat architektúra.



5.4. ábra. Pontosság (felső) és loss (alsó) értékek a tanító (kék), illetve validációs (piros) adatokon.

Természetesen ezt követően még számos más beállítással is elvégeztem a kísérletet⁴. Állítottam a szűrők paraméterein, változtattam a többrétegű perceptron rejtett dimenzióját és igyekeztem minél kisebb, de még a tanítóhalmazon megfelelően tanulni képes modellt létrehozni annak reményében, hogy a kevésbé komplex mintázatot találó modell majd kevésbé fog túltanulni, de azt tapasztaltam, hogy még az elkészített legkisebb szabadságfokú (kb. 2000 paramétert tartalmazó) modell tanulása során is azonnal nőtt a hiba értéke a validációs halmazon. Tehát összefoglalva, lényegében mindegyik esetben a bemutatotthoz hasonló eredményre sikerült jutnom.

⁴És természetesen alaposan átnéztem a teljes folyamatot reménykedve abban, hogy esetleg a programkód hibája miatt kapok ilyen eredményeket, de nem találtam benne olyan hibát, ami miatt teljesen elcsúszott volna a tanítás.

5.6. Különbség alapú megközelítés

Az előző alfejezetben bemutatott megközelítéssel azt tapasztaltam, hogy nagyon hamar túltanult a modell, ráadásul a rétegek paramétereitől függetlenül. Felmerült tehát bennem a kérdés, hogy létezik-e olyan megoldás, ami jobban a lényegénél kezeli a problémát, ezáltal a tanult mintázatot minél inkább általános érvényű lehet.

5.6.1. Ötlet

Az előző megközelítésben arra tanítottam a modellt, hogy tetszőleges személy, tetszőleges éjszakájának szívverésébe pár órára belehallgatva képes legyen megállapítani, hogy számíthat-e a közeljövőben migrénes rohamra. Ez tulajdonképpen egy meglehetősen erős elvárás egy algoritmustól, tekintve, hogy nem kap semmi többletinformációt az illetőre vonatkozóan. Nem tudja, hogy milyen nemű, mi az életkora, vagy milyen egészségnek örvend. És ami a legfontosabb, azt sem ismeri, hogy általában hogyan szokott verni az illető szíve, csak az aktuális értékeket kapja meg és abból kell prediktálnia. Viszont ez ellentmondásos azzal, hogy a valódi információ a különbségben rejlik, vagyis azt keressük, hogy miben tér el az RR-intervallumok szekvenciája a roham előtti éjjelen az azonos illető kontroll éjjeleitől⁵.

Ezen gondolatmenetből kiindulva megalkottam egy olyan tanítási folyamatot és modell architektúrát, amely azonos beteg kontroll és roham előtti éjszakáinak egymáshoz való hasonlításán alapszik, bízva abban, hogy ezzel egy robusztusabb megoldást sikerül létrehozni, amely általánosító képessége is megfelelőbb.

5.6.2. Tanítási folyamat

Tehát az előzőekben megfogalmazottak alapján az a cél, hogy a modell a kimenetét az éjszakák közti különbségekből határozza meg. Ezáltal az előzőektől eltérően nem egy, hanem két éjszakát kell a modellnek egyszerre átadni, tehát két RR-intervallum szekvencia lesz a bemenete, amelyeket cél (*target*), illetve referencia (*reference*) bemenetnek neveztem el. Az előbbiről feladata a modellnek eldönteni, hogy melyik kategóriába tartozik, míg az utóbbit viszonyítási alapként használja. Referencia csakis kontroll éjszakái adatból származhat, illetve fontos, hogy minden esetben azonos beteghez kell, hogy tartozzon mindkét bemenet.

Mivel az előzőektől eltérően itt megköveteli az eljárás, hogy legalább egy kontroll csoportba tartozó éjszakája legyen az adott páciensek, ezért el kellett dobnom az összes olyan adatot, ahol ez nem teljesül. Ennek következményeképpen az eredmények nem egyértelműen összehasonlíthatóak a korábbiakkal, ugyanis csak a korábbi halmaz egy részhalmazán tanul a modell, ami önmagában is hatással lehet az eredményre (főképp abban az esetben, ha összességében kevés adat áll rendelkezésre).

Mivel úgy véltem, hogy az 5.4.1 alfejezetben ismertetett megoldások nem elvetendőek, ezért az adatok kiválasztását alapjaiban nem változtattam meg, csak új funkciókkal bővítettem. A bemenő adatokat kezelő programlogika minden egyes epoch előtt egyénenként a kontroll éjszakák közül véletlenszerűen kiválaszt egyet és kinevezi referenciának, majd a tanulási iteráció során az aktuális céléjszakához társítja a hozzátartozó referencia estét.

Azáltal, hogy a referencia esték halmaza is minden alkalommal változik, illetve véletlenszerűen történik az RR-intervallum szekvencia egy szeletének kiválasztása belőle, a korábban említett lehetséges bemenetek egyébként is magas száma sokszorozódik, amely szintén pozitív hatással lehet a modell általánosító képességére.

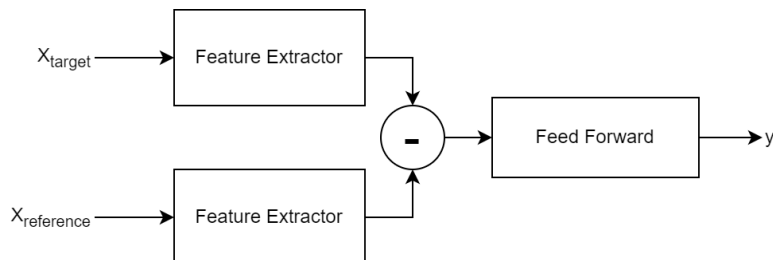
⁵Szeretném felidézni, hogy a sokat hivatkozott, biztató eredményeket közlő [22] cikk is az éjszakák különbségén alapuló megoldást alkalmazott.

5.6.3. Modell architektúra

Összegezve az előző pontokban foglaltakat, a tervezendő modell két bemenettel rendelkezik, amelyeknek valamilyen módon képzett különbségéből határozza meg, hogy melyik kategóriába tartozik az úgynevezett cél bemenet. Tehát elsődlegesen a felmerülő kérdés az, hogy hogyan érdemes két, sok elemből álló RR-intervallum szekvencia különbségét képezni.

Ennek megválaszolásához visszatérnek az 5.1. ábrához, amelyet vizsgálva igyekeztem különbséget felfedezni a roham előtti és kontroll éjszakák között. Ehhez azt az eljárást alkalmaztam, hogy például szemmel meghatároztam az egyes éjszakákra vonatkozó RR-intervallum értékek átlagát és megnéztem, hogy esetleg található-e szignifikáns, azonos irányú különbség a roham előtti időszakokban a többihez képest. Más szóval meghatároztam integrális paramétereket és azok eltérését figyeltem. Ebből kiindulva azt szeretném elérni, hogy a modell meghatározzon olyan transzformációkat, amikkel a bemenetből megfelelő paramétereket határoz meg, majd vegye azok különbségét és abból határozza meg a kimenetet.

Tehát a modell architektúra alapvetően két fő komponensből épül fel. Az első célja az integrális paraméterek meghatározása (*feature extractor*), az utóbbi feladata pedig a kimenet megállapítása ezen értékekből (*feed forward*). Az 5.5. ábrán illusztráltam az architektúrát, amelyről leolvasható, hogy a modell, a futtatása során mindkét bemenetre azonos eljárással (a *feature extractor* komponens segítségével) meghatároz N darab értéket, majd ezek különbségét képi és adja át az utóbbi, *feed forward* komponensnek. Tehát ezzel a megoldással az azonos transzformációval kiszámított értékek különbsége lesz az az információ, amelyből a modell végül eldönti, hogy migrén előtti, vagy kontroll kategóriába tartozik a céléjszaka.



5.5. ábra. Leegyszerűsített modell architektúra.

Amennyiben az N érték túl nagynak, vagy a *feed forward* komponenst túl komplexnek választjuk, akkor beláthatjuk, hogy valószínűleg a korábbiakhoz hasonlóan hamar túltanul a modell. Ennek oka, hogy az integrális paraméterek meghatározása kevésbé lesz kritikus feladat, mivel a modell végén található transzformációk a kevésbé jól megválasztott paramétereket is le tudják képezni megfelelő kimenetre. Ellenkező extrém esetben, vagyis mondjuk amennyiben $N = 1$, akkor minden feladat a *feature extractor* komponensre hárul és valószínűleg szintén hasonló eredményre juthatunk, mint a korábbi megközelítéssel, mivel a modell feladata lényegében ugyanaz, vagyis hogy minél nagyobb értékeket társítson az egyik kategóriához és minél kisebbet a másikhoz. Mindezekből arra a következtetésre juthatunk, hogy az N megfelelő megválasztása meglehetősen kritikus feladat, illetve nem érdemes túl kicsire, illetve túl nagyra állítani az értékét.

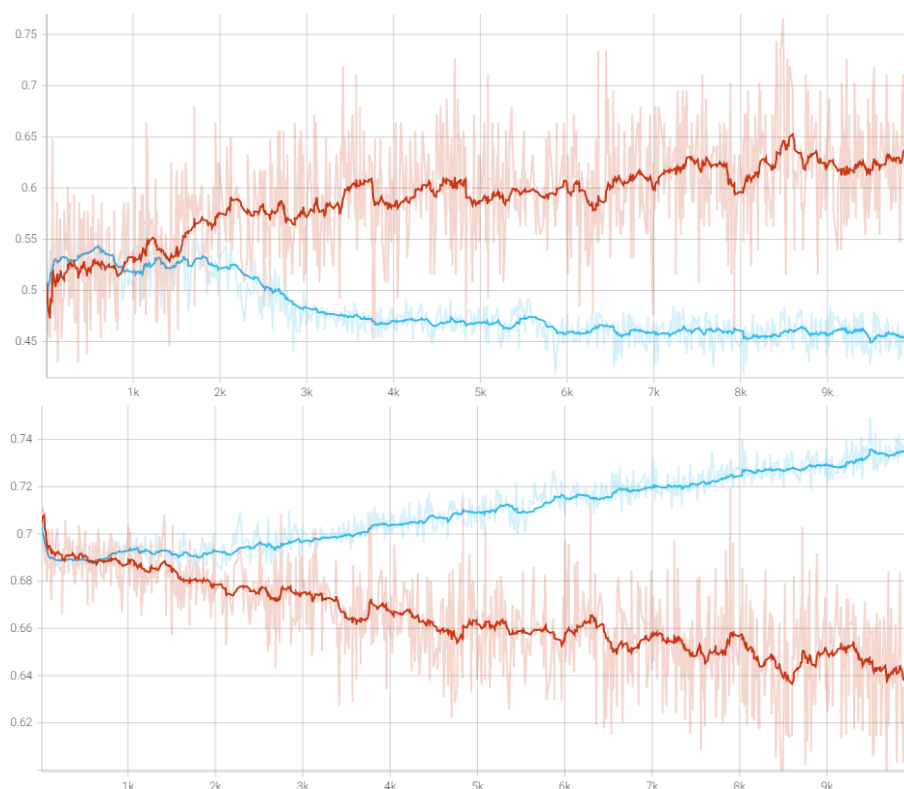
Az 5.3 táblázatban foglaltam össze az elkészített *feature extractor* komponens architektúráját, amely legfőképp abban tér el a korábban (5.2. táblázatban ismertetett) alkalmazottól, hogy jelentősen csökkentettem a szabadságfokát, illetve a végén nem egyetlen, hanem 4 kimenet található. Az előbbinek az volt az oka, hogy a túltanulástól tartva nem szerettem volna, ha a model túl összetett módon határozza meg a kívánt 4 értéket, ezért

igyekeztem úgy megválasztani a hiperparamétereket, hogy relatíve egyszerű mintázatban keresse a különbséget. Továbbá említést érdemel, hogy ezen komponens kimenetére a tanh transfer függvényt választottam annak érdekében, hogy a $(-1, 1)$ intervallumba essen a kiválasztott 4 paraméter, vagyis ne adjon tovább kiugró számértékeket. A *feed forward* komponenst pedig egyetlen perceptronnal alkottam meg, azzal a céllal, hogy ne vegye le a feladatot *feature extractor* komponens válláról. Az így elkészült modellt összesen 3,033 állítható paraméter alkotja.

Réteg neve	Kimeneti dimenziója	Paraméterei
Input	12 288 x 1	-
Conv1D	12 225 x 16	kernel méret: 64, szűrők száma: 16
ReLU		-
AveragePooling1D	32 x 32	méret: 12 225 // 32
Flatten	1 024	-
Dense	4	méret: 4
Tanh		-

5.3. táblázat. Implementált *feature extractor* komponens architektúra.

5.6.4. Eredmények

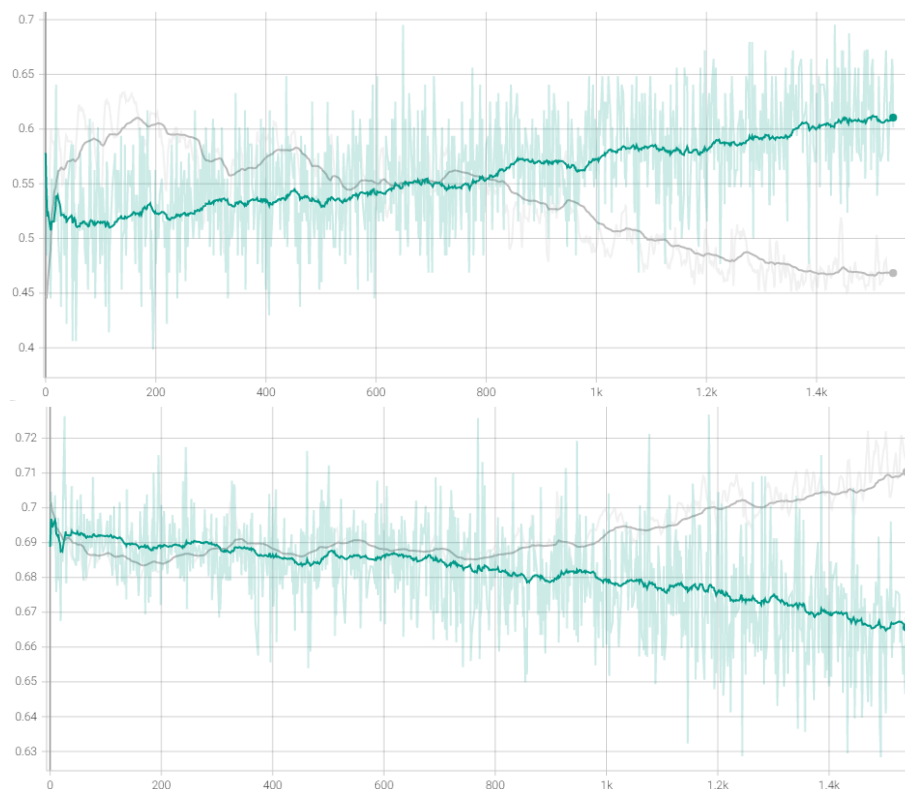


5.6. ábra. Pontosság (felső) és loss (alsó) értékek a tanító (piros), illetve validációs (kék) adatokon.

Az 5.6. ábrán látható a modell tanításának eredménye. Egyrészt leolvasható, hogy a modell képes tanulni a tanító halmazon, vagyis folyamatosan csökkent a kimeneti hiba értéke és nőtt a pontosság egészen 65%-ig (sőt, a tendenciából ítélve valószínűleg tovább is nőtt volna, ha még tovább tart a tanítás). Másrészt a validációs halmazon elért eredményekből az látható, hogy eleinte együtt javultak az értékei a tanító értékekkel, ami

megközelítőleg maximum 55%-os pontosságot jelentett mindkettőn, majd ezt követően a már jól ismert túltanulás mintázata a megfigyelhető, vagyis a hibaérték a tanító eredményekkel ellentétes irányba tendált.

Természetesen ezzel a megközelítéssel sem ez volt az egyetlen kísérlet, amit elvégeztem. Az 5.7. ábrán látható egy másik eredmény, amely során egy jóval nagyobb szabadságfokú *feature extractor* komponenssel próbálkoztam. Leolvasható, hogy összességében hasonló ezen megoldás kimenete is, ugyanis látható, hogy a két halmazon viszonylag együtt csökken a hiba, a pontosságértékek 55% környékén találkoznak, majd a következőkben a validációs halmazon egyfolytában romlik az eredmény.



5.7. ábra. Pontosság (felső) és loss (alsó) értékek a tanító (zöld), illetve validációs (szürke) adatokon.

5.7. Összefoglalás

Megközelítesemben ahelyett, hogy az RR-intervallumok szekvenciájából határoztam volna meg az orvostudomány által meghatározott HRV paramétereket, egy általánosabb megoldásra törekedtem, ahol az említett paraméterek meghatározását is modell végzi.

Először a lehető legegyszerűbb megoldással próbálkoztam, amely segítségével igyekeztem felmérni, hogy milyen irányban érdemes továbbhaladni. Az első eredmények arra utaltak, hogy minimális mennyiségben ugyan, de található olyan mintázat az adatokban, ami a validációs halmazon is fellelhető, viszont a megoldás egyszerűsége következményeként a létrejött eredmények nem bizonyultak olyan szinten reprezentatívnak, hogy abból messzemenő következtetéseket lehessen levonni.

Ezt követően igyekeztem úgy fejleszteni a tanítási folyamatot, hogy az a modell általánosító képességére jó hatással legyen, illetve feldúsítottam a validációs adatokat, ezáltal növelve annak reprezentativitását.

Emellett a modell architektúrát is átalakítottam olyan formára, amelytől a lehető legjobb eredményeket reméltem. A rétegek és hiperparaméterek meghatározásánál figyelembe vettem, hogy milyen megoldásokat alkalmaznak idősoros adatok mesterséges intelligencia alapú feldolgozására, illetve végiggondoltam, hogy milyen megközelítés illeszkedik a legjobban a megoldandó feladathoz. A modell tanítások során azt tapasztaltam, hogy kisebb-nagyobb hiperparaméterekkel rendelkező megoldások megközelítőleg azonos eredményt szolgáltatnak a validációs halmazra vonatkozólag, amely pontossága 50% környékén mozgott minden esetben. Ezek az eredmények arra készítettek, hogy újragondoljam a feladatot és egy, az éjszakák különbségére épülő eljárást és architektúrát dolgoztam ki, amely legjobb eredménye a körülbelül 55%-os pontosság mindkét halmazon egyidőben.

6. fejezet

Konklúzió

A dolgozat során arra a kérdésre kerestem a választ, hogy az RR-intervallumok éjszakai szekvenciájából felismerhető-e a közelgő migrénes roham kialakulása.

Az első megközelítésem szerint a modell feladata egy körülbelül 3 órányi RR-intervallum szekvenciából meghatározni, hogy az adott éjszaka roham előtti vagy kontroll éjszakába tartozik. Ennek során nagyon hamar túltanulást tapasztaltam annak ellenére, hogy számos különböző módon fejlesztettem a használt eljárást az általánosító képesség növelésére.

Amennyiben különbségalapú megközelítést alkalmaztam, valamivel jobb eredményeket¹ sikerült elérni, ami körülbelül 55%-os pontosságot jelent mindkét halmazon egyidőben. Konklúzióként arra jutottam az eredmények alapján, hogy az alábbi okai lehetnek annak, hogy a modell általánosító képessége nem megfelelő:

1. Hiba van az adatok feldolgozásában, vagy címkézésében.

Annak érdekében, hogy ennek a valószínűségét csökkentsem, sokat dolgoztam azon, hogy minden egyes lépést alaposan megvizsgáljak és leellenőrizzek.

2. Nagyon sok zaj terheli az adatokat.

Annak ellenére, hogy nagyon alaposan válogattam ki az éjszakákat és igyekeztem a kiugró adatokat eldobni, vagy kompenzálni, sajnos számos olyan zajforrás van, amit nem tudtam kezelni. Néhányat említve a teljesség igénye nélkül ilyen például, ha az adott éjszaka felületesen, vagy rosszul alszik az illető, vagy stresszhatás alatt van. Elképzelhetőnek tartom, hogy ezek a környezeti hatások olyan szinten befolyásolják az illető fiziológiás állapotát, hogy a számunkra releváns információt javarészt elnyomja az ilyen típusú zaj.

3. Kevés a rendelkezésemre álló adat.

Összesen körülbelül 100 migrén előtti kategóriába tartozó éjszaka állt rendelkezésemre, ami minden bizonnyal csekély mennyiségű egy megfelelő minőségű, populációs szinten működő, általános megoldáshoz. Emellett fontos kihangsúlyozni, hogy egy-egy betegről csak néhány éjszaka a felhasználható, ami legfőképp annak köszönhető, hogy nagyon óvatos az adatok kiválasztása és kategorizálása. Mivel összesen 20-20 adatot választottam a validációs halmazba mindkét csoportból, ezért nem tekinthetjük reprezentatívnak az ezen a halmazon elért eredményeket.

4. A hipotézis nem állja meg a helyét.

¹Szeretném kihangsúlyozni, hogy a két megközelítés eredménye nem egyértelműen összehasonlítható, mivel csekély mértékben ugyan, de különbözött a tanító halmaz.

Amennyiben a szívritmusban nincsen olyan változás órákkal a roham kezdete előtt, amit a közelgő roham idézne elő, akkor hiába minden erőfeszítés és rengeteg adat, nem lehetséges a rohampredikció.

5. Nem jó a megközelítés.

Végül lehet, hogy egy jobb modell architektúra és tanítási folyamat segítségével az elért eredményeket lehet javítani.

A különbség alapú megközelítésem által a validációs halmazon elért körülbelül 55%-os eredményből messzemenő következtetéseket úgy vélem nem érdemes levonni. Valamilyen szinten egyszerre nevezhető biztató eredménynek, illetve a véletlen művének. Tehát mindent összevetve, a hipotézis, miszerint az éjszakai RR-intervallum szekvenciákból a migrénes roham már órákkal a roham kezdete előtt kimutatható, nem nyert bizonyítást, viszont egyértelműen megcáfolásra sem került. Az eredmények viszont arra engednek következtetni, hogy amennyiben a hipotézis igaz, vagyis előrejelzi a szív ritmusa a közelgő rohamot, az feltételezhetően komoly zajjal terhelt és nem könnyedén megtalálható, mindenkinél azonos mintázat. Tehát összességében én is arra a következtetésre jutottam, amelyre számos szakirodalom, hogy a populációs szintű predikció kifejezetten nehéz, illetve egy továbbra is megoldásra váró feladat, amellyel remélhetőleg lehetőség nyílik rengeteg migrénes beteg életminőségének javítására.

Köszönetnyilvánítás

Szeretnék köszönetet nyilvánítani elsődlegesen egyetemi konzulensemnek, Dr. Dabóczi Tamásnak a témaválasztásban való közreműködéséért, a rengeteg rászánt idejéért és példaeértékű segítőkészségéért. Emellett szeretnék köszönetet mondani a céges konzulensemnek, Dr. Molnár Balázsnak a technikai részletekben, illetve a releváns egészségügyi kérdésekben való kisegítéséért, valamint Dr. Tényi Ákosnak a céges kooperációra kapott lehetőségért.

Irodalomjegyzék

- [1] Saad Albawi – Tareq Abed Mohammed – Saad Al-Zawi: Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)* (konferenciaanyag). 2017, 1–6. p.
- [2] Márta Altrichter – Gábor Horváth – Béla Pataki – György Strausz – Gábor Takács – József Valyon: *Neurális hálózatok*. Budapest, 2006, Panem Könyvkiadó Kft. ISBN 978-963-545-464-8.
- [3] Dawn C. Buse – James S. McGinley – Richard B. Lipton: Predicting the Future of Migraine Attack Prediction. *Headache: The Journal of Head and Face Pain*, 60. évf. (2020) 10. sz., 2125–2128. p. ISSN 1526-4610.
URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/head.14025>.
_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/head.14025>.
- [4] Eva Csepany – Máté Magyar – Tamas Gyure – Gyorgy Bozsik – Csaba Ertsey: A migrén korszerű farmakoterápiája. *Neuropsychopharmacologia Hungarica: a Magyar Pszichofarmakológiai Egyesület lapja = official journal of the Hungarian Association of Psychopharmacology*, 17. évf. (2016. január), 169–176. p.
- [5] Zsuzsanna Csík – Kornélia Hegedűs – György Dr Lévay – Ágnes Dr Valló – György Dr Lévay – Péter Dr Enyedi: *Alkalmazott élettan - kórélettan*. Budapest, 2010, Semmelweis Egyetem Egészségtudományi Kar. ISBN 978-963-7152-75-7.
- [6] M. W Gardner – S. R Dorling: Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32. évf. (1998. augusztus) 14. sz., 2627–2636. p. ISSN 1352-2310. URL <https://www.sciencedirect.com/science/article/pii/S1352231097004470>.
- [7] N. J. Giffin – L. Ruggiero – R. B. Lipton – S. D. Silberstein – J. F. Tvedskov – J. Olesen – J. Altman – P. J. Goadsby – A. Macrae: Premonitory symptoms in migraine: an electronic diary study. *Neurology*, 60. évf. (2003. március) 6. sz., 935–940. p. ISSN 1526-632X.
- [8] Jean-Jacques Goy – Jean-Christophe Stauffer – Jürg Schlaepfer – Pierre Christeler: *Electrocardiography (ECG)*. 2013. június, Bentham Science Publishers. ISBN 978-1-60805-479-4. Google-Books-ID: R2viAwAAQBAJ.
- [9] Headache Classification Committee of the International Headache Society (IHS) The International Classification of Headache Disorders, 3rd edition. *Cephalalgia: An International Journal of Headache*, 38. évf. (2018. január) 1. sz., 1–211. p. ISSN 1468-2982.
- [10] Katherine K. Holsteen – Michael Hittle – Meredith Barad – Lorene M. Nelson: Development and Internal Validation of a Multivariable Prediction Model for Individual Episodic Migraine Attacks Based on Daily Trigger Exposures. *Headache: The*

- Journal of Head and Face Pain*, 60. évf. (2020) 10. sz., 2364–2379. p. ISSN 1526-4610. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/head.13960>.
_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/head.13960>.
- [11] IHDL 2021 Szimpózium | InnoHealth Datalake.
URL <https://innohealth.eu/ihdl-2021-szimpozium/>.
 - [12] P. Kropp: Prediction of migraine attacks. *Cephalalgia: An International Journal of Headache*, 19. évf. (1999. június) 5. sz., 477. p. ISSN 0333-1024.
 - [13] Richard B. Lipton–Marcelo E. Bigal–Ann I. Scher–Walter F. Stewart: The global burden of migraine. *The Journal of Headache and Pain*, 4. évf. (2003. március) Suppl 1. sz., s3–s11. p. ISSN 1129-2369.
URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3611680/>.
 - [14] Mitchell G. Miglis: Migraine and Autonomic Dysfunction: Which Is the Horse and Which Is the Jockey? *Current Pain and Headache Reports*, 22. évf. (2018. február) 3. sz., 19. p. ISSN 1534-3081. URL <https://doi.org/10.1007/s11916-018-0671-y>.
 - [15] Balázs Molnár–László Micsinyei–Gábor Perlaki–Gergely Orsi–László Hejmel–Tamás Dóczy–József Janszky–Norbert Laky–Ákos Tényi: Data Quality Enhancement for Machine Learning on Wearable ECGs. In Ignacio Rojas–Olga Valenzuela–Fernando Rojas–Luis Javier Herrera–Francisco Ortuño (szerk.): *Bioinformatics and Biomedical Engineering*, Lecture Notes in Computer Science konferenciasorozat. Cham, 2022, Springer International Publishing, 269–279. p. ISBN 978-3-031-07704-3.
 - [16] Golden L. Peters: Migraine overview and summary of current and emerging treatment options. *The American Journal of Managed Care*, 25. évf. (2019. január) 2 Suppl. sz., S23–S34. p. ISSN 1936-2692.
 - [17] Veronika Rauschel–Andreas Straube–Frank Süß–Ruth Ruscheweyh: Responsiveness of the autonomic nervous system during paced breathing and mental stress in migraine patients. *The Journal of Headache and Pain*, 16. évf. (2015. szeptember) 1. sz., 82. p. ISSN 1129-2377. URL <https://doi.org/10.1186/s10194-015-0567-8>.
 - [18] Ricardo Rodríguez-Jorge–I. De León-Damas–Jiri Bila–Jiří Škvor: Internet of things-assisted architecture for QRS complex detection in real time. *Internet of Things*, 14. évf. (2021. június), 100395. p. ISSN 2542-6605. URL <https://www.sciencedirect.com/science/article/pii/S2542660521000391>.
 - [19] Stuart Russell–Peter Norvig: *Artificial Intelligence: A Modern Approach*. 3rd edition. kiad. Upper Saddle River, 2009, Pearson. ISBN 978-0-13-604259-4.
 - [20] Yasar Sattar–Lovely Chhabra: Electrocardiogram. In *StatPearls*. Treasure Island (FL), 2022, StatPearls Publishing.
URL <http://www.ncbi.nlm.nih.gov/books/NBK549803/>.
 - [21] Fred Shaffer–J. P. Ginsberg: An Overview of Heart Rate Variability Metrics and Norms. *Frontiers in Public Health*, 5. évf. (2017. szeptember), 258. p. ISSN 2296-2565. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5624990/>.
 - [22] Pekka Siirtola–Heli Koskimäki–Henna Mönttinen–Juha Rönning: Using Sleep Time Data from Wearable Sensors for Early Detection of Migraine Attacks. *Sensors (Basel, Switzerland)*, 18. évf. (2018. április) 5. sz., E1374. p. ISSN 1424-8220.

- [23] The Timeline of a Migraine Attack. URL <https://americanmigrainefoundation.org/resource-library/timeline-migraine-attack/>.
- [24] H. Kenneth Walker – W. Dallas Hall – J. Willis Hurst (szerk.): *Clinical Methods: The History, Physical, and Laboratory Examinations*. 3rd. kiad. Boston, 1990, Butterworths. ISBN 978-0-409-90077-4.
URL <http://www.ncbi.nlm.nih.gov/books/NBK201/>.