

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
INFORMATIKAI KAR



DIPLOMAMUNKA

OE-NIK
2024

Hallgató neve:
Hallgató törzskönyvi száma:

Kisbenedek Lilla
T-009542/FI12904

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

DIPLOMAMUNKA FELADATLAP

Hallgató neve: **Kisbenedek Lilla**
Törzskönyvi száma: **T/009542/FI12904/N**
Neptun kódja: **JRP7MG**

A diplomamunka címe:

Élettani modell illesztés gépi tanulással **Physiological model fitting using machine learning**

Intézményi konzulensek: **Dr. Drexler Dániel András, Puskás Melánia**
Külső konzulens:

Beadási határidő: **2024. május 15.**



A feladat


A jövő orvoslásában az egyik ígéretes irányvonal a terápiák matematikai és mérnöki módszereken alapuló optimalizálásában rejlik, amelyekkel a betegségek kezelése személyre szabottan történik, ellentétben a ma használt, ad hoc, nagy populációs átlagra kifejlesztett, ritkán optimalizált terápiákkal. A terápiaoptimalizálás számtalan előnnyel jár, ezek közül a legfontosabb az alacsonyabb dózis, ami kevesebb mellékhatással, alacsonyabb költségekkel jár, és bizonyos esetekben előnyös hatással lehet a gyógyszerrel szemben kialakuló rezisztencia leküzdésében is. A terápiaoptimalizáláshoz azonban szükség van a gyógyszer hatását leíró modellre, ami alapján az optimalizálás elvégezhető. Ez a modell függ a betegség típusától, a páciensről és az alkalmazott gyógyszertől. A személyre szabás alapja, hogy meg tudjuk adni az adott páciens modelljében szereplő paramétereket, amik alapján a terápiaoptimalizálás elvégezhető. Ehhez parametrikus identifikációra van szükség, amihez sok mérés kell, ami a betegállásban nehezen kivitelezhető, azonban kísérleti körülmények között van rá lehetőség.

A feladat egy tumornövekedési modell alapján egy paraméterbecslő rendszer létrehozása neurális hálózat segítségével. A neurális hálózat tanításához rendelkezésre állnak valós mérési adatokból identifikált modellparaméterek, amik kemoterápiás szerrel kezelt emlőrákos egerekből származnak. Ezek alapján virtuális egereket hozunk létre, amelyekkel megfelelő számú tanítóadat generálható, azonban az így elkészült neurális hálózat nem feltétlenül tudja megfelelően kezelni a valódi mérésből származó adatokat. Ezért a neurális hálózatot megerősítéses tanulással tovább hangoljuk a mérésekből származó adatokon.

A diplomamunkának tartalmaznia kell:

- a mesterséges neurális hálózatok, megerősítéses tanulás, hiperparaméterek optimalizációja és identifikációs algoritmusok témakörében végzett irodalomkutatást,
- a mesterséges neurális hálózatok és identifikációs algoritmusok kombinálásával létrehozott paraméterbecslő eljárás rendszertervét, a megerősítéses tanulás tervét. Vizsgálja meg különböző hiperparaméterek optimalizációját végző algoritmusok használatát,
- a paraméterbecslő eljárás és megerősítéses tanulási algoritmusok implementálását,
- a paraméterbecslő eljárás tesztelését és más algoritmusokkal való összehasonlítását,
- az eredmények értékelését.





.....
Dr. Eigner György
intézetigazgató

A diplomamunka elévülésének határideje: **2026. május 15.**
(ÓE HKR 61.§ (10) bekezdés szerint)

A diplomamunkát beadásra alkalmasnak tartom:

.....
külső konzulens


.....
intézményi konzulens



HALLGATÓI NYILATKOZAT

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

Budapest, 2024. 05. 15.

Kisbenedek Lilla
.....

hallgató aláírása



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:

Kisbenedek Lilla..... JRP7MG..... nappali

Telefon: Levelezési cím (pl: lakcím):

+36306849457..... 3348 Szilvásvárad Széchenyi u. 28.

Szakedolgozat / Diplomamunka¹ címe magyarul:

Élettani modell illesztés gépi tanulással.....

Szakedolgozat / Diplomamunka² címe angolul:





Physiological model fitting using machine learning.....

Intézményi konzulens:

Külső konzulens:

Dr.Drexler Dániel.....

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2022.09.20.	Téma pontosítása, eddigi eredmények megvitatása	
2.	2022.10.12.	Rendszersepcifikáció ellenőrzése, javaslatok átbeszélése	
3.	2022.11.04.	Előrehaladás ellenőrzése, első implementációs eredmények megvitatása	
4.	2022.12.02.	Implementáció megbeszélése, javítása	

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

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

Budapest, 2022.12.09.....


.....
intézményi konzulens

¹ Megfelelő aláhúzendó!

² Megfelelő aláhúzendó!

³ Megfelelő aláhúzendó!

⁴ Megfelelő aláhúzendó!



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Kisbenedek Lilla Neptun Kód: JRP7MG Tagozat: nappali
Telefon: +36306849457 Levelezési cím (pl.: lakcím): 3348 Szilvásvárad Széchenyi u. 28.

Szakedolgozat / Diplomamunka¹ címe magyarul:

Élettani modell illesztés gépi tanulással

Szakedolgozat / Diplomamunka² címe angolul:

Physiological model fitting using machine learning

Intézményi konzulens:

Dr. Drexler Dániel András

Külső konzulens:

Kérjük, hogy az adatokat nyomtatott nagy betűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2023.02.23.	Az eddigi munka áttekintése, továbbfejlesztési irányok átgondolása.	
2.	2023.03.16.	Előrehaladás ellenőrzése.	
3.	2023.04.06.	Eredmények megvitatása.	
4.	2023.05.04.	Prezentáció ellenőrzése.	

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

A hallgató a Szakedolgozat / Szakedolgozat I. / Szakedolgozat II. / Projektlabor 1 / Projektlabor 2 / Projektlabor 3 / Záródolgozati projekt / Diplomamunka I / Diplomamunka II / Diplomamunka III / Diplomamunka IV ³ tantárgy követelményét teljesítette, beszámolóra / védésre ⁴bocsátható.

A konzulens által javasolt érdemjegy: jóles (5)

Budapest, 2023.05.11.

.....
Intézményi konzulens

¹ Megfelelő aláhúzendó!

² Megfelelő aláhúzendó!

³ Megfelelő aláhúzendó!

⁴ Megfelelő aláhúzendó!



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Kisbenedek Lilla Neptun Kód: JRP7MG Tagozat: nappali

Telefon: +36306849457 Levelezési cím (pl.: lakcím): 1148 Budapest Bolgárkertész u. 5/A 1/7

Szakedolgozat / Diplomamunka¹ címe magyarul:

Élettani modell illesztés gépi tanulással

Szakedolgozat / Diplomamunka² címe angolul:

Physiological model fitting using machine learning

Intézményi konzulens:

Külső konzulens:

Dr. Drexler Dániel András, Puskás Melánia

Kérjük, hogy az adatokat nyomtatott nagy betűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2024.03.06.	Az eddigi munka áttekintése, féléves tervek egyeztetése.	
2.	2024.04.12.	Eredmények ellenőrzése.	
3.	2024.04.24.	Dolgozat felépítésének megbeszélése.	
4.	2024.05.06.	Dolgozat és prezentáció ellenőrzése.	

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

A hallgató a Szakedolgozat / Szakedolgozat I. / Szakedolgozat II. / Projektlabor 1 / Projektlabor 2 / Projektlabor 3 / Záródolgozati projekt / Diplomamunka I / Diplomamunka II / Diplomamunka III / Diplomamunka IV³ tantárgy követelményét teljesítette, beszámolóra / védésre⁴bocsátható.

A konzulens által javasolt érdemjegy: jelas.....

Budapest, 2024.05.13.

Intézményi konzulens

¹ Megfelelő aláhúzendó!

² Megfelelő aláhúzendó!

³ Megfelelő aláhúzendó!

⁴ Megfelelő aláhúzendó!

KIVONAT

A diplomamunkám egy kutatás részeként jött létre, mely során daganatos betegek számára hozunk létre optimális terápiát. A kutatás során már korábban létrehozásra került egy tumordinamikát és a gyógyszer tumorra gyakorolt hatását leíró differenciálegyenlet-rendszer. Az egyenletrendszer négy egyenletből áll és nyolc paramétert tartalmaz. Az egyenletekben szereplő paraméterek által célunk a páciensek jellemzése, meghatározott jellemzőik alapján pedig esetleges csoportosításuk hasonló tumordinamika alapján. A betegek jellemzői ismeretében személyreszabott terápia generálható.

Diplomamunkám a korábban létrehozott tumor modell differenciálegyenleteiben szereplő paramétereknek a meghatározásához kapcsolódik. A munkám fő célja identifikáció elősegítésére alkalmas mesterséges intelligenciát alkalmazó algoritmusok létrehozása. Az első lépésben a rendelkezésre álló tumortérfigyelő adatok szűrésével foglalkoztam, a kiugró értékek detekciójával majd pedig az adatok zajtalanításával. A hibás és zajos adatok kiszűrése amiatt fontos, mivel a konvergenciát befolyásolhatják a különböző paraméteridentifikációs algoritmusokban. Ezt követően két mesterséges intelligencián alapuló algoritmust hoztam létre a páciensek csoportosítására és azok paramétereinek kezdeti meghatározására. Az első létrehozott algoritmus egy klaszterező algoritmus, mely adott kezelés esetén a hasonló tumordinamikával rendelkező pácienseket képes csoportosítani. A páciensek csoportosítása által a különböző klaszterekhez tartozó paramétereikről ezáltal információ nyerhető indirekt módon. Ezt az algoritmust a későbbiekben arra az esetre lehet felhasználni, ha a betegek kezelés kezdetén hasonló dózist kapnak, majd pedig a beadott injekcióra adott választ követjük a tumor növekedéseknek. A gyógyszerre hasonlóan reagáló páciensek számára populációra optimalizált terápia alkalmazható, mellyel a személyreszabott terápia költségei csökkenthetők. A második algoritmus egy autenkóder architektúrát követő gépi tanulási algoritmus, ahol a hálózat képes a paraméterek önálló meghatározására és megtanulására felügyelet nélküli tanúlással. Az autenkóder két komponensből épül fel, egy enkóder részből, ami esetünkben a paraméterek megtanulásáért felelős hálózat, illetve egy dekóderből, mely egy differenciálható numerikus integrátor (ODE megoldó), mely a kezdeti feltételekből és a becsült paramétereiből megoldja a tumormodell differenciálegyenleteit.

Munkámmal a tumormodell paramétereinek meghatározásához járultam hozzá, mely elengedhetetlen előzetes lépése az optimális terápia megalkotásának. Kutatásom az Innovációs és Technológiai Minisztérium ÚNKP-23-2 kódszámú Új Nemzeti Kiválósági Programjának támogatásával készült.

ABSTRACT

My thesis is part of a research project when we create an optimal therapy for patients diagnosed with cancer. In this research, there has already been established a system of differential equations describing tumor dynamics and the effect of the drug on the tumor. The system of equations consists of four equations and eight parameters. By using the parameters in the equations, we aim to characterize patients and, based on their specific characteristics, possibly group them according to similar tumor dynamics. Personalized therapy can be generated based on these characteristics of the patients.

My thesis is related to the determination of parameters in differential equations of the tumor model. The main goal of my work is to create artificial intelligence algorithms to facilitate identification. In the first step, I worked on filtering the available tumor volume data, detecting outliers, and then de-noising the time series. Filtering out erroneous and noisy data is important because it can affect convergence in different parameter identification algorithms. Subsequently, two artificial intelligence-based algorithms were created to cluster patients and predict their parameters. The first algorithm created is a clustering algorithm that can group patients with similar tumor dynamics for a given treatment. By clustering patients into separate groups, information on parameters belonging to different clusters can be obtained indirectly. This algorithm can be used in the future, by giving the patients a similar dose at the start of treatment and then tracking tumor growth in response to the injection. Population-optimized therapy can be applied to patients with similar answers to the drug, reducing the cost of personalized therapy. The second algorithm is a machine learning algorithm following an autoencoder architecture, where the algorithm can determine and learn parameters autonomously through unsupervised learning. The autocoder is composed of two components, an encoder part, which in our case is the network responsible for learning the parameters, and a decoder, which is a differentiable numerical integrator (ODE solver) that solves the differential equations of the tumor model from the initial conditions and the predicted parameters.

My work has contributed to the identification of the parameters of the tumor model, which is an essential preliminary step in the design of an optimal therapy. My research was funded by the New National Excellence Programme of the Ministry of Innovation and Technology, code number ÚNKP-23-2.

Tartalomjegyzék

Nomenklátúra	11
1. Bevezetés	13
2. Irodalmi háttér	16
2.1. A tumor dinamikáját leíró matematikai modell	16
2.2. Paraméterbecslési probléma és megoldási lehetőségei	18
2.3. Alkalmazott algoritmusok alapjai	22
3. Megvalósítás	26
3.1. Kiugró értékek detektálása és zajszűrés	26
3.2. Idősorok klaszterezése tumordinamika alapján	32
3.3. Paraméterbecslő autoenkóder létrehozása	37
4. Eredmények	47
4.1. Kiugró értékek detektálásának a kiértékelése	47
4.2. Idősorok klaszterezésének kiértékelése	52
4.3. A paraméterbecslő autoenkóder kiértékelése	58
5. Konklúzió	65
Irodalomjegyzék	67
Ábrák jegyzéke	73

Nomenklatúra

- a : Tumor növekedési ráta.
- b : Inhibíciós ráta.
- c : Módosított klírens.
- $d_{i,j}$: Az i -edik virtuális pácienshez tartozó dózis a j -edik napon.
- $D_{i,j}$: A BMU i -edik és az adott j -edik neuron közötti távolság.
- E : Bemeneti vektorok halmaza a SOM algoritmusban.
- ED_{50} : Medián effektív dózis.
- $h(0)$: A kiindulási állapot ($t=0$) a NODE architektúra esetén.
- h_t : A NODE architektúra esetén a t -edik lépésben a hálózat állapota.
- $h(T)$: A T -edik időpillanatban a kimenet a NODE architektúra esetén.
- $\mathcal{H}_{i,j}$: Gauss-féle szomszédsági függvény a SOM algoritmusban.
- k_1 : A gyógyszer áramlási sebességi együtthatója a centrális kompartmentből a perifériás kompartmentbe.
- k_2 : A gyógyszer áramlási sebességi együtthatója a perifériás kompartmentből a centrális kompartmentbe.
- n : Nekrotikus ráta.
- n : Neuronok száma a SOM algoritmusban.
- N : A tanító adatok száma, a dolgozatban $N = 20000$.
- \mathcal{N} : Differenciál-operátor a PINN architektúrában.
- L_B : A peremfeltételek teljesítéséért felelős tag a PINN költségfüggvényében.
- L_F : A differenciálegyenlet kielégítéséért felelős tag a PINN költségfüggvényében.
- L_{data} : Az illeszkedésért felelős tag a PINN költségfüggvényében.
- \mathcal{L} : Költségfüggvény a neurális hálózatban.
- p : A tumormodell összes paraméterét tartalmazó halmaz.
- p^{\max} : A paraméterek felső határa.
- p^{\min} : A paraméterek alsó határa.
- p' : A paraméterek értéke normalizálást követően.

t_k	: A k -edik injekciózás időpontja.
u	: Beadott kemoterápiás dóziseket tartalmazó vektor.
w	: Kimosódási ráta.
$W_{(NN)}$: Az autoenkóder neurális hálózatához tartozó súlyokat tartalmazó mátrix.
$W_{(SOM)}$: A SOM hálózatához tartozó súlyokat tartalmazó mátrix.
x_1	: Az élő tumortérfogat időfüggvénye [mm^3].
x_2	: A halott tumortérfogat időfüggvénye [mm^3].
x_3	: A gyógyszer szint időfüggvénye a vérben [mg/kg].
x_4	: A gyógyszer szint időfüggvénye a szövetekben [mg/kg].
X_i	: Az i -edik bemeneti érték az autoenkóderben.
\hat{X}_i	: Az X_i bemenetre adott kimeneti predikció (rekonstrukció) az autoenkóderben.
y	: A teljes tumortérfogat [mm^3].
Y	: A teljes tumortérfogatok tartalmazó vektor.
\hat{Y}	: A becsült teljes tumortérfogatok tartalmazó vektor egy mérési intervallumra, L hosszúságú.
z_i	: Az autoenkóder enkóder részének i -edik bemenetre adott kimenete (tömörített reprezentáció).
$\eta_{(NN)}$: A tanulási sebesség (hiperparaméter) a neurális hálózatban.
$\eta_{(SOM)}$: A tanulási sebesség (hiperparaméter) a SOM algoritmusban.
θ	: A neurális hálózat összes paramétereit tartalmazó halmaz.
σ	: A szigmoid függvény.
γ	: A SOM algoritmus szomszédsági függvényét befolyásoló hiperparaméter.
ω	: A PINN költségfüggvényében alkalmazott súly.

1. fejezet

Bevezetés

A daganatos megbetegedések a vezető halálokok közé tartoznak, továbbá az egyik leginkább meghatározó tényező, amely lassítja a várható élettartam növekedését [1]. Bár egyes fejlett országokban a rákos megbetegedések halálozási rátája az elmúlt évtizedekben csökkent, ezt az előrehaladást mérsékli az egyes rák típusok megnövekedett előfordulási száma [2]. Továbbá a betegek kezelése és ellátása a közegészségügyben hatalmas gazdasági erőforrásokat emészt fel, a következő 40 évre pedig növekvő tendenciát jósolnak az esetszámokat tekintve [3].

A rákos megbetegedéseket a sejtek szabályozatlan növekedése és a sejtek terjedése a származási helyükről más testrészekre jellemzi. A rák különböző betegségek összefoglaló neve, melyek különböző tünetekkel társulnak és eltérő kezelési módszereket igényelnek. Évtizedeken át csak néhány lehetőség állt rendelkezésre a rák kezelésére, amelyek magukban foglalták a műtétet, a sugárterápiát és a kemoterápiát, valamint ezeket kombinációban vagy önálló kezelési módszerként alkalmazva [4]. A sugárterápia során a cél az, hogy az ionizáló sugárzásnak kitett daganatsejtek DNS-ét károsítsák, így megakadályozzák azok szaporodását vagy elpusztítják azokat. A hagyományos kemoterápia célja szintén a gyorsan növekvő és osztódó rákos sejtek elpusztítása, különböző kemoterápiás gyógyszerek alkalmazásával. Utóbbi hátránya, hogy az alkalmazott gyógyszerek nem szelektívek, és az egészséges, normál szöveteket is károsíthatják, súlyos, nemkívánatos mellékhatásokat, például étvágytalanságot, hányingert és hajhullást okozva [5].

Az olyan tulajdonságok, mint a rák típusa, helye és súlyossága befolyásolják a kezelési mód kiválasztását. A sebészeti eltávolítás sugárterápiával kombinálva az elsődleges módszer az áttét nélküli elsődleges daganatok esetében. Mindazonáltal, számos erősen agresszív daganat esetén ezeknek a módszereknek a korlátai nyilvánvalóak a betegeknél. A leggyakrabban ajánlott terápia a műtéti beavatkozást követő sugár- és kemoterápia. Bár a két módszer súlyos mellékhatásokat okoz, meglehetősen rontva az életminőséget, alkalmazásuk nagy mértékben

csökkenti a morbiditást és a mortalitást [6].

A rákos megbetegedések kezelése során a gyógyszer-rezisztencia és a gyógyszer hordozó rendszerek kifejeletlensége jelentik az egyik legnagyobb problémát a rákgyógyításban [6]. Az utóbbi évtizedekben nagy mértékben fejlődtek a rák kezelésére szolgáló különböző modalitások. A hagyományos módszerek mellett nagy hangsúlyt kaptak az alternatív kezelési módszerek is, mint az őssejtterápia, a célzott gyógyszerek, a hormonterápia és az immunterápia. Összességében a rákellenes gyógyszer kutatások a sokkal pontosabb és kevesebb gyógyszer alkalmazása irányába fordultak.

A gyógyszerkutatáson felül a gyógyszer adagolásának szabályozása is megélnéült az utóbbi évtizedekben. Az új gyógyszeradagoló rendszerek fejlesztésének fő célja, hogy lehetővé tegyék a tartós és ellenőrzött gyógyszerleadást, a hatékony gyógyszer szintet és ezzel egyidejűleg csökkentsék a káros hatásokat [7]. A kemoterápia során beadott gyógyszerek megállapítása jelenleg egy meghatározott protokoll alapján történik, mely a betegek test-súlyát veszi figyelembe. Az adagolás során a maximálisan tolerálható dózist alkalmazzák (MTD - Maximum Tolerated Dosage), ahol pár hetes időközönként nagy dózisu kemoterápiás szert adnak be a betegeknek, melynek hátránya a hosszabb kezelésmentes időszakok alatt a rezisztencia-mechanizmusok megjelenése, továbbá a mellékhatások súlyosbodása.

A gyógyszerhordozó rendszerek létrehozásának alapja egy robusztus algoritmus, mely képes a páciens mért tulajdonságai alapján meghatározni a szükséges gyógyszer mennyiségét. A dózisok megállapítása egy matematikai modellen alapuló algoritmus alapján történik, melynek a paramétereinek a meghatározása elengedhetetlen a folyamat során. A munkám során egy korábban megalkotott tumormodellt használok fel, mely egy négy állapotváltozót tartalmazó közönséges differenciál-egyenlet rendszer. A rendszer bemenete a beadott dózis, míg a kimenete a teljes tumortérfogat. Utóbbi az általunk mérhető változó, mely leírja a tumor reakcióját a beadott gyógyszer mennyiségekre. Az egyenletek nyolc paramétert tartalmaznak, melyeknek az identifikált értéke alapján történik a személyreszabott, optimális terápia megalkotása. A paraméter identifikáció gyakran hosszas és bizonytalanságokat tartalmazó folyamat, melynek a meggyorsítására egy mesterséges intelligenciát felhasználó algoritmus alkalmas lehet. Ahhoz, hogy megfelelő adatokkal történjen az algoritmusok tanítása, kezdetben kiszűrtem a rendelkezésre álló adatokból a kiugró értékeket és a zajt. Ezt követően két különböző mesterséges intelligencián alapuló algoritmust alkalmaztam a paraméterek identifikációjára. Az első algoritmus a mért tumortérfogatot tartalmazó idősorokat csoportosította hasonló tumordinamika alapján, a másik algoritmus pedig egy autoenkóder-alapú hálózatot használt a paraméterek meghatározására.

A dolgozatom elején ismertetem a felhasznált tumordinamikai modellt (2.1. alfejezet). Mivel a modell kimenete alapján történik az egyenletekben szereplő paraméterek meghatá-

rozása, a 2.2. fejezetben annak identifikációjának lehetőségeit részletezem, továbbá kitérek a felhasznált algoritmusok működésére is. Ezt követően a 3. fejezetben először a tumortér-fogatokat tartalmazó idősorokban a kiugró értékek identifikációját, majd a zajmentesítését részletezem. Végül pedig a paraméter identifikációra alkalmas algoritmus implementációs lépéseit mutatom be. A 3. fejezetben külön bemutatom a különböző implementációs lépések során kapott eredményeket. A létrehozott rendszer alkalmas lehet a paraméterek meghatározására, illetve kezdeti értékük becslésére. Utóbbi felhasználható különböző identifikációs algoritmusok kiindulási értékeként.

A kezelések személyre szabása során lehetséges akár egy olyan gyógyszerhordozó rendszer megalkotása, mely személyre szabottan állapítja meg a kemoterápiás szerek beadott mennyiségét és annak optimális időpontját, azáltal hogy információt kap a daganat adott pillanatbeli állapotáról. Munkámmal az algoritmus paraméterbecslő szakaszának a működéséhez járultam hozzá.

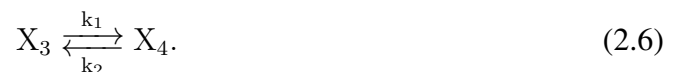
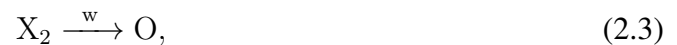
2. fejezet

Irodalmi háttér

2.1. A tumor dinamikáját leíró matematikai modell

Egy kísérletekkel validált modell segítségével előre jelezhető, hogy egy adott kezelési protokoll milyen hatással lesz a tumor dinamikájára. Ezáltal a kezelések hatékonysága növelhető, a mellékhatások csökkenthetők és a kezelési költségek optimalizálhatók. A szakirodalomban számos modell található a tumor viselkedésének leírására [8, 9, 10]. Ezek a megalkotott modellek vagy csak az időben, vagy időben és térben próbálják meg leírni közönséges és parciális differenciálegyenletek által a tumor dinamikáját.

A munkám során egy korábban létrehozott tumormodellt használok fel a tumortérfogatok szimulációjára [11]. A tumormodell egyenletei reakciókinetikai analógia alapján lettek létrehozva, ahol X_1 , X_2 , X_3 , és X_4 fiktív tagok.



Az X_1 az élő tumor térfogata, az X_2 az elhalt tumor térfogata, az X_3 a gyógyszer szint a centrális kompartmentben, míg az X_4 a gyógyszer szint a perifériás kompartmentben. A (2.1) egyenlet a tumorsejtek osztódását, míg (2.2) az elhalását írja le. A (2.3) egyenlet a tumorsejtek kimosódását modellezi. A (2.4) egyenlet a gyógyszer hatását, (2.5) a gyógyszer kiürülését, míg (2.6) a két kompartment közötti reverzibilis gyógyszeráramlást írja le.

A (2.1)-(2.6) fiktív reakciókból felírt differenciálegyenlet-rendszer:

$$\dot{x}_1 = (a - n)x_1 - b \frac{x_1 x_3}{ED_{50} + x_3}, \quad (2.7)$$

$$\dot{x}_2 = nx_1 + b \frac{x_1 x_3}{ED_{50} + x_3} - wx_2, \quad (2.8)$$

$$\dot{x}_3 = -(c + k_1)x_3 + k_2 x_4 + u, \quad (2.9)$$

$$\dot{x}_4 = k_1 x_3 - k_2 x_4, \quad (2.10)$$

ahol x_1 az élő tumortérfogat időfüggvénye [mm^3], x_2 az elhalt tumortérfogat időfüggvénye [mm^3], x_3 és x_4 a gyógyszer szint időfüggvénye a centrális kompartmentben [mg/kg] és a perifériás kompartmentben [mg/kg]. A modellt daganatos egereken végzett mérésekkel igazolták [12]. A létrehozott modell célja, hogy leírja a pegilált liposzómális doxorubicin (PLD) kemoterápiás szer tumorelles hatását és a tumor dinamikáját. A (2.7)-(2.8) egyenletek a tumor dinamikáját és a gyógyszer hatását (farmakodinamikát), míg (2.9)-(2.10) a gyógyszer áramlását (farmakokinetikát) írják le.

Az egyenletekben szereplő paraméterek a reakciókinetikai együtthatóknak tekinthetők a formális felírás alapján. Az együtthatók közül a , n , w esetünkben a proliferációs ráta, a nekrotikus ráta valamint a kimosódási ráta, míg b , ED_{50} a gyógyszert jellemző együtthatók (a gyógyszer maximális hatását leíró paraméter és a medián effektív dózis). A c , k_1 , k_2 paraméterek a gyógyszer áramlását írják le, a gyógyszer kimosódását, valamint a centrális és a perifériás kompartment közötti áramlását. A 3. fejezetben a felsorolt paraméterek korábbi állatkísérletek során meghatározott határait vettem figyelembe. A 2.1. táblázatban foglaltam össze a paraméterek jelölését, mértékegységét és a korábban meghatározott értékeinek a felső és alsó határát.

Korábbi munkák során megállapításra került [13], hogy a kemoterápiás szer csak akkor van hatással a tumor térfogatára, ha a

$$0 > a - b - n \quad (2.11)$$

egyenlőtlenség teljesül. Ha az összefüggés nem teljesül, az a daganat gyógyszerre való rezisztenciáját jelenti.

A rendszer bemente az u [mg/kg/nap], ami az időegység alatt bevitt gyógyszer mennyisége, melyre igaz, hogy $u \geq 0$, tehát nem vehet fel negatív értéket, mely adódik abból, hogy gyógyszert nem vonhatunk el a szervezetből. Ez a felírás folytonos bemenetet feltételez. A gyakorlatban azonban a bemenetek injekciók, azaz az egyes beadott gyógyszer dózisokat impulzív hatásnak tekintjük a vérben (x_3). Tehát, t_0, \dots, t_{K-1} időpillanatban beadott K számú

injekció esetén, mivel $t_k \geq 0$ és $k = 0, 1, 2, \dots, K - 1$, valamint $t_0 < t_1 < \dots < t_{K-1}$, így x_3 -ban folytonossági szakadás van a t_k időpontban, mely az alábbi összefüggéssel írható fel:

$$x_3(t^+_k) = x_3(t^-_k) + u_k, \quad (2.12)$$

ahol u_k a t_k időpontban beadott gyógyszer dózisa mg/kg-ban. A rendszer kimenete – az általunk mérhető változó – az élő és a halott tumor térfogatok összege:

$$y = x_1 + x_2. \quad (2.13)$$

Az x_1 és a x_2 tumortérfogat nem határozható meg külön, a teljes tumortérfogat mérése történik az egereken. A teljes tumortérfogat (y) mérése tolmérővel történik két ponton, a tumor szélességét és hosszúságát mérik, melyből egy korábban meghatározott közelítő képlet alapján történik a térfogatának kiszámítása [14]. A mérések tartalmazhatnak mérési zajt, melynek modellezése egy különálló munka a kutatás részeként [15].

Paraméter	Mértékegység	Minimum érték	Maximum érték
a	nap ⁻¹	0,5811	0,6871
b	nap ⁻¹	1,0234	1,0261
n	nap ⁻¹	$6,5696 \cdot 10^{-6}$	$1,1555 \cdot 10^{-5}$
w	nap ⁻¹	0,1742	0,3229
ED ₅₀	mg · kg ⁻¹	0,1	1,0
c	nap ⁻¹	0,3425	0,9772
k ₁	nap ⁻¹	6,7697	7,778
k ₂	nap ⁻¹	48,51	474,66

2.1. táblázat. A korábban meghatározott paraméterek minimum és maximum értékei. Az a a tumor proliferációs rátája, a b a gyógyszer maximális hatása, az n a tumor nekrotikus rátája, a w a kimosódási ráta, ED₅₀ a medián effektív dózis, c a gyógyszer kimosódási rátája, míg k₁ az áramlási együttható a vérből a szövetekbe, míg k₂ a szövetekből a vérbe.

2.2. Paraméterbecslési probléma és megoldási lehetőségei

A differenciálegyenletek esetében gyakran nem áll rendelkezésre az analitikus megoldás, így azok megoldása és a paramétereik meghatározása hosszadalmas folyamatot igényelhet. Egy

adott differenciálegyenlet vagy rendszer megoldásának keresését gyakran direkt problémának nevezzük, míg az inverz probléma alatt olyan megfelelő modell meghatározását értjük, amelynek megoldása jól egyezik néhány kísérleti vagy valós megfigyeléssel. Tehát a paraméterbecslés célja a megfigyeléseket tartalmazó adathalmazhoz legjobban illeszkedő modell ismeretlen paramétereinek meghatározása [16]. Míg a direkt problémának egyedi megoldása van, az inverz problémának több is lehet. Ebből adódóan az inverz problémában minden rendelkezésre álló információt figyelembe kell venni a meghatározandó paramétereiről. A dolgozatban a cél a 2.1. alfejezetben részletezett közönséges differenciálegyenleteket tartalmazó tumormodell paramétereinek meghatározása.

A paraméterbecslés folyamatában egy költségfüggvény optimalizálására törekszünk, melynek keretében a leggyakrabban alkalmazott módszerek a Bayes-i becslés, a maximum likelihood módszere és a legkisebb négyzetek módszere [17]. A Bayes-i becslés a Bayes-tételt használja a valószínűségi modellek paramétereinek becslésére, ezáltal kifejezi a bizonytalanságot is a paraméterbecslés körül. A maximum likelihood (ML) módszer a paramétereknek azt a legvalószínűbb értékét adja meg, ami a legnagyobb valószínűséggel következik be. Az ML módszer nem ad közvetlen becslést a paraméterek bizonytalanságáról és eloszlásáról. A legkisebb négyzetek módszere pedig a mért adatpontok és a modell predikciói között minimalizálja a négyzetes eltérést. A felsorolt paraméterbecslési módszerek által általában olyan nemlineáris optimalizálási problémát kapunk, amelyet zárt formában nem lehet megoldani, így iteratív numerikus optimalizálókat alkalmazunk a megoldás keresésére [18]. A felsorolt algoritmusok gyakran kiindulási érték megadását igénylik, melyből az iteráció elindítható. Korábbi munkákban a bemutatott tumor modell paramétereinek meghatározására többek között SAEM algoritmust alkalmaztak maximum-likelihood becslésre [19].

Az optimalizálásra alkalmazott algoritmusokat megkülönböztethetjük aszerint, hogy lokális vagy globális minimumok megkeresésére alkalmasak. Ez a megkülönböztetés csak a nemlineáris optimalizálás kontextusában szükséges, mivel a lineáris problémáknak mindig egyedi optima van. A lokális optimalizálók egy nagy csoportját alkotják a gradiens alapú módszerek, mint a Newton-módszerek, a kvázi-Newton módszerek és a konjugált gradiens módszer, melyek a paraméterek elsőrendű deriváltakon alapuló frissítésével keresik az optimális értékeket. Léteznek algoritmusok, melyek a másodrendű deriváltak értékét használják fel, mint a Broyden-Fletcher-Goldfarb-Shanno (BFGS) módszer, Gauss-Newton módszer és a Levenberg–Marquardt algoritmus. A globális optimalizálók között a legelterjedtebbek a populáció alapú algoritmusok, mint az evolúción alapulóak (például genetikai algoritmus) vagy részecske-raj optimalizálók.

A paraméterek meghatározása során fontos terület az identifikálhatósági vizsgálata a paramétereknek. Az azonosíthatósági elemzése felméri, hogy elméletileg meghatározható-e a

modell paramétereit az adatokból, megvizsgálva a méréseket, a modell szerkezetét, valamint az előforduló hibák tulajdonságait. Egy modell strukturálisan azonosítható, ha létezik egyedi paraméterezés bármely adott modell kimenetéhez. Ebből adódik, hogy egy paraméter szerkezetileg nem azonosítható, ha a paraméter megváltoztatása nem feltétlenül változtatja meg a modell kimenetét, mert a változások teljes mértékben kompenzálhatók más paraméterek megváltoztatásával. A biológiai rendszerek modellezésekor különösen fontos az azonosíthatóság vizsgálata, mert a kísérleti adatok korlátozott mennyisége és minősége csak részben megfigyelt rendszerekben gyakran rossz modellekhez és paraméterekhez vezet a modellezési folyamat során [20]. Korábbi munkákban azonosíthatósági és érzékenységi vizsgálatot végeztek az általam is alkalmazott tumormodellen, mely során megállapították, hogy az azonosítandó paraméterek és a többi névleges populációs érték kiválasztásával a rendszer azonosítható. Továbbá megállapították, hogy a paraméterek érzékenységi sorrendje az következő: a , b , n és w , ahol magasan a tumor proliferációs rátája a legérzékenyebb a tumormodell kimenetére [21].

Az irodalomban az utóbbi években számos megközelítés jelent meg különböző differenciálegyenletek és neurális hálózatok integrációjából. Ezek a hálózatok a fizikai, biológiai vagy kémiai törvényeket – jellemzően differenciálegyenletek formájában – közvetlenül a mély tanulási modellek architektúrájába illesztik be, ezáltal irányítják a tanulási folyamatot az egyenletek betartására. A legnépszerűbb megközelítés az utóbbi években az ún. fizikával informált neurális hálózatok (PINN - Physics Informed Neural Network). A PINN hálózatokat először kezdeti érték problémák pontosabb megoldásának keresésére használták a neurális hálózatok által nyújtott függvény approximáció segítségével. Mindazonáltal inverz problémákra is alkalmazható, azáltal, hogy a költségfüggvénybe bele vesszük nem csak a reziduumból álló tagot (a differenciálegyenletek betartásáért felelős tag), hanem az illeszkedésért felelős tagot is (az előre jelzett kimenetek és az aktuális megfigyelt adatok közötti eltérést) [22, 23, 24]. Az általános felírása egy PINN hálózat költségfüggvényében szereplő tagoknak az alábbi egyenlettel szemléltethető:

$$\theta^* = \arg \min_{\theta} (\omega_F L_F(\theta) + \omega_B L_B(\theta) + \omega_d L_{data}(\theta)). \quad (2.14)$$

A neurális hálózatnak meg kell tanulnia közelíteni a differenciálegyenleteket θ hálózat paramétereinek meghatározásával, egy olyan költségfüggvény minimalizálásával, amely függ a differenciálegyenlettől (L_F), a határfeltételektől (L_B), és esetlegesen néhány ismert adattól (L_{data}). Ahol az ω_F , ω_B , ω_L a tagok megfelelő súlyozásait jelentik. A tagok az egyenletben az alábbi jelentéssel bírnak. Az L_{data} az illeszkedésért felelős komponens:

$$L_{data} = \frac{1}{N} \sum_{i=1}^N \|(y_i - \hat{y}_i)\|^2, \quad (2.15)$$

ahol y_i a mért adatpont, míg \hat{y}_i a becsült adatpont. Az L_F a hálózat kimenetét arra kényszeríti, hogy eleget tegyen a megadott biológiai vagy fizikai törvényeknek, mint például egy differenciálegyenletnek:

$$L_F = \sum_{j=1}^M \|\mathcal{N}\mathcal{N}[y_{predicted}](x_j)\|^2, \quad (2.16)$$

ahol \mathcal{N} differenciál-operátor, y a kimeneti változó, x_j azok a pontok a tartományban, ahol a differenciálegyenletnek érvényesnek kell lennie, és M ezeknek a pontoknak a száma. A peremfeltételekre is felírható egy általános hibafüggvény:

$$L_B = \frac{1}{P} \sum_{k=1}^P \|(y(x_{bk}) - g(x_{bk}))\|^2, \quad (2.17)$$

ahol P a peremértékek száma, valamint $g(x_{bk})$ egy adott peremfeltétel függvény [25]. Mindazonáltal esetünkben a 2.1. fejezetben bemutatott szakadás következtében – a dózisok beadása során – a teljes mérési időintervallumra nem modellezhető a problémánk PINN-el.

A másik megközelítése a hálózatok és a differenciálegyenletek integrációjának a neurális közönséges differenciálegyenletek (NODE - Neural Ordinary Differential Equations) [26]. Az Euler-módszer az egyik legegyszerűbb eljárás a differenciálegyenletek kezdeti érték problémáinak megoldására, ahol a következő érték az előző pontban meghatározott meredekség és a lépésköz szorzata alapján számítható ki. A reziduális hálózatok hasonló koncepción alapoznak, hasonlóan a mély neurális hálózatokhoz annyi különbséggel, hogy egy réteg bemenetéhez hozzáadjuk az előző előtti réteg kimenetét is. Így a hálózat következő állapotára az alábbi összefüggés írható fel:

$$h_{t+1} = h_t + f(h_t, \theta_t), \quad (2.18)$$

ahol t a lépésszámot, h_t pedig a t -edik lépésben a hálózat állapotát jelöli. Ha a lépésközöket egyre kisebbre vesszük, akkor ez a folyamat egy differenciálegyenlet folytonos formájához konvergál:

$$\frac{dh(t)}{dt} = f(h(t), t, \theta), \quad (2.19)$$

ahol $h(0)$ a kiindulási állapot, $h(T)$ pedig a T időpillanatban a kimenet, ami megfelel az ODE kezdeti érték probléma megoldásának. A NODE felhasználásával kezdeti érték problémákat tudunk megoldani állandó memóriafelhasználással, hiszen nem szükséges a köztes lépések tárolása. A módszert többek között paraméter identifikációra is alkalmazták [27], [28], [29].

2.3. Alkalmazott algoritmusok alapjai

Önszerveződő térképek

Az önszerveződő térkép (self organizing map, SOM, más néven Kohonen-térkép) egy felügyelet nélküli mesterséges neurális hálózati módszer [30]. A SOM algoritmus hatékonyan tudja címkézetlen adatkészlet mintáit csoportosítani azáltal, hogy közöttük mintázatokat fedez fel, létrehozva egy teljesen összekapcsolt csomópontokból álló térképet. Esetünkben a jelöletlen adatok a daganat térfogatának idősorai.

A SOM algoritmus három szakaszra osztható: a kompetitív részre, a kooperatív részre és a súlyok frissítésére. A kezdeti lépésben az algoritmus inicializál egy egyrétegű hálózatot (úgynevezett SOM rács), ahol minden neuronjához egy súlyvektor tartozik, amelynek mérete megegyezik a bemeneti adatokkal. A versengés szakaszban – azaz a kompetíció során – ezt követően azonosítja a legjobban illeszkedő neuront. Tehát az algoritmus a bemenethez legjobban hasonlító, úgynevezett győztes egységet (BMU - Best Matching Unit) azonosítja, majd az ahhoz tartozó szomszédos neuronok súlyait állítja be [31]. Ezek a lépések megismétlődnek a betanítási adatkészlet minden bemeneti vektoránál, lehetővé téve a SOM számára, hogy iteratív módon tanuljon és alkalmazkodjon a bemeneti adatokhoz. A tanulási folyamat célja, hogy megtalálja azokat a súlyokat, ahol a szomszédos csomópontok hasonló értékekkel rendelkeznek. Tehát a matematikai formalizmussal élve a bemutatott három szakasz a következőképpen írható fel:

1. szakasz: Versengés

A kompetitív szakaszban a hálózat minden neuronja verseng egymással a bemenet reprezentálásáért. Minden egyes bemenethez kiválasztjuk a legjobban illeszkedő neuront (BMU) úgy, hogy összehasonlítjuk őket a bemenettel egy előre meghatározott távolságmetrika szerint. A győztes neuronnak minimális távolsága van a bemeneti adatomintától, amelyet a következő egyenlettel fejezhetünk ki:

$$\text{BMU} = \min_i (||E - W_{(\text{SOM})i}||), \quad (2.20)$$

ahol E a bemeneti vektor, $W_{(\text{SOM})}$ pedig az egységek (neuronok) súlya és $i = 1, 2, \dots, n$ jelenti az adott neuront, a neuronok száma pedig n [32].

2. szakasz: Együtműködés

A kooperatív szakaszban a győztes neuron (BMU) behatárolja a kiválasztott neuron topológiai szomszédságát. Az ebből a szomszédságból származó neuronok ezután együttműködnek. A szomszédos neuronokat a szomszédsági függvény alapján határozzák meg, amely meghatározza a szomszédos neuronok befolyásának mértékét a súlyfrissítési folyamatra. A szomszédsági függvény jellemzően figyelembe veszi a neuronok és a BMU közötti távolságot, ahol a közelebbi neuronok nagyobb befolyást gyakorolnak, ilyen például a Gauss-féle szomszédsági függvény:

$$\mathcal{H}_{i,j} = \exp\left(-\frac{D_{i,j}}{2\gamma^2}\right), \quad (2.21)$$

ahol $D_{i,j}$ az i -edik (BMU) és az adott j -edik neuron közötti távolsága, valamint γ egy hiperparaméter.

3. szakasz: Adaptáció

Az adaptáció során – melyet szinaptikus alkalmazkodásnak is neveznek – az algoritmus frissíti a súlyokat. A BMU és a hozzá közel álló neuronok súlya a SOM rácsban a bemeneti vektor irányába változik. A változás mértéke az idő előrehaladtával és a BMU-tól való távolsággal csökken. A neuronok a bemutatott bemeneti minta vektorhoz kapcsolódó értékeit súlykorrekciókkal, optimalizáló algoritmusok segítségével módosíthatják. A súlyfrissítés a következő formában írható fel:

$$W_{(\text{SOM})_i}(t+1) = W_{(\text{SOM})_i}(t) + \eta_{(\text{SOM})} \mathcal{H}_{i,j}(E(s) - W_{(\text{SOM})_i}(t)), \quad (2.22)$$

ahol t az adott iterációt jelenti, míg s az aktuális bemeneti adat indexe. A súlyok iteratíván frissülnek, és a korrekció mértékét egy, a tanulási sebességet leíró paraméter ($\eta_{(\text{SOM})}$) határozza meg [32].

Neurális hálózatok és autoenkóderek

A mesterséges neurális hálózatok (ANN - Artificial Neural Network) a gépi tanulás egy fontos részterületét alkotják. Ezek a rendszerek az emberi agy neuronhálózatainak működésének az analógiájára lettek megalkotva. Az emberi agyban lévő neuronoknak több ezer szinaptikus kapcsolata lehet más neuronokkal. Az agykéregben létrejövő összekapcsolt neuronok hálózata felelős a vizuális, hang- és érzékszervi adatok feldolgozásáért [33]. Általában egy mesterséges neurális hálózat több rétegből áll, amelyek közé tartozik a bemeneti, egy vagy

több rejtett és a kimeneti réteg. A több rejtett réteget tartalmazó hálózatot mély neurális hálózatnak nevezzük. Minden egyes neuron kapcsolódik a következő réteg neuronjaihoz, ahol az összeköttetések súlyozottak és a neuron aktiválódását egy eltolás vagy más néven torzítás szabályozza. A rétegek különböző számú neuronokat tartalmazhatnak, és minden réteg kimenete a következő réteg bemenete. Minden rétegben lineáris transzformáció történik az alábbiak szerint:

$$z^{[l]} = W_{(\text{NN})}^{[l]} a^{[l-1]} + b^{[l]}, \quad (2.23)$$

ahol $W_{(\text{NN})}^{[l]}$ a súlyokat tartalmazó mátrix, $b^{[l]}$ az eltolásokat tartalmazó vektor, $a^{[l-1]}$ az előző réteg kimenete, l pedig a réteg indexe.

Ezt követően a lineáris transzformáció kimenete egy nemlineáris transzformációs függvényen megy keresztül, mint például a szigmoid függvény. Ezeket nevezzük aktivációs függvényeknek. A szigmoid függvény az alábbiak szerint írható fel:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}, \quad (2.24)$$

ahol x a bemenete a függvénynek. A szigmoid függvény 0 és 1 közé képezi le a bemenetet. Tehát az l -edik réteg kimenete

$$a^{[l]} = \sigma(z^{[l]}). \quad (2.25)$$

A betanításhoz szükséges egy költségfüggvény (\mathcal{L}), amely méri a különbséget az előre jelzett kimenetek és a valódi címkék között, melynek értéke alapján történik a hiba visszaterjesztése a hálózat súlyaira láncszabály alkalmazásával, azaz

$$\frac{\partial \mathcal{L}}{\partial W_{(\text{NN})}^{[l]}} = \frac{\partial \mathcal{L}}{\partial z^{[l]}} \cdot \frac{\partial z^{[l]}}{\partial W_{(\text{NN})}^{[l]}}, \quad (2.26)$$

majd ezt követően a súlyok frissítése szükséges, egy válaszott optimalizáló által:

$$W_{(\text{NN})}^{[l]} = W_{(\text{NN})}^{[l]} - \eta_{(\text{NN})} \frac{\partial \mathcal{L}}{\partial W_{(\text{NN})}^{[l]}}, \quad (2.27)$$

ahol $\eta_{(\text{NN})}$ a tanulási sebesség.

A neurális hálózatok kialakítása során a választott architektúra általában az adott megoldandó problémától függ, mint például osztályozási feladatok vagy dimenziócsökkentés. A dimenziócsökkentésre alkalmazott architektúrák gyakori példája az autoenkóder [34]. Az autoenkóder egy típusa a felügyelet nélküli neurális hálózatoknak, amelyet elsősorban főbb jellemzők kinyerésére használnak [35]. Kezdetben úgy hivatkoztak rá, hogy "hibavisszaterjesztés tanár nélkül", hiszen a bemeneti adatok által tanul a hálózat [36], ezáltal nincs szükség címkézett adatkészletre.

Az autoenkóder működése formálisan is felírható. Például, ha az enkódert $h(\cdot)$ függvényként jelöljük, akkor:

$$z_i = h(X_i, \theta), \quad (2.28)$$

ahol X_i a bemenet, $i = 0, 1, \dots, N$ és N a teljes bemenete nagysága, illetve θ tartalmazza az enkóder paramétereit. A dekóder rész szintén felírható a

$$\hat{X}_i = g(z_i, \omega), \quad (2.29)$$

függvénnyel, ahol ω tartalmazza a dekóder paramétereit. A két szakasz közös költségfüggvénnyel rendelkezik, és céljuk a θ és ω paraméterkészletük optimalizálása a bemenet és kimenet közötti minimalizálás során. Az optimalizáció az alábbi általános költségfüggvényt minimalizálja:

$$SSE = \sum_{i=1}^N (\hat{X}_i - X_i)^2, \quad (2.30)$$

ahol X_i az i -edik bemenet, \hat{X}_i pedig annak a rekonstrukciója.

3. fejezet

Megvalósítás

3.1. Kiugró értékek detektálása és zajszűrés

A kiugró értékek detektálása során a cél egy olyan algoritmus létrehozása volt, mely képes azonosítani az olyan értékeket, amelyek hibásak, nem illeszkednek a tumordinamikára jellemző megváltozásra. Egy példa ezekre a kiugró értékekre, ha a mérést követően, az adatok felvétele során rossz érték kerül rögzítésre, például elgépelés esetén.

A kiugró értékek detekciójára az irodalomban számos módszer létezik [37, 38, 39]. A módszer kiválasztása az idősor jellemzőitől és az adott kiugró típustól függően változik. Esetemben a kiugró értékek a mérési adatok hibáiként vannak definiálva, amelyek hibásan vannak mérve vagy felvéve az adatok közé. Az idősorok anomáliáinak kimutatására alkalmas módszerek számát korlátozza, hogy a rendelkezésre álló idősorokra nem jellemző a szezonális (megjósolható, rendszeres időközönként ismétlődő mintázat) vagy a stacionaritás (a statisztikai tulajdonságok nem függenek a megfigyelés időpontjától). Ezenkívül a kiugró érték meghatározása kontextusfüggő, és csak egy adott környezetben releváns. Például a beadott kemoterápiás szer mennyisége és időpontja is befolyásolhatja a tumordinamikát, hiszen nagy mennyiségű kemoterápiás szer beadását követően – feltéve ha jól reagál a daganat a gyógyszerre – a tumortérfogat a meredek növekedést követően hirtelen lecsökkenhet. Utóbbi jelenséget a kiugró érték detektálása során az algoritmusok tévesen jelölhetik anomáliának.

Az általam tesztelt, elterjedt módszerek a felsorolt okok miatt nem hoztak kielégítő eredményt. Ennek következtében a feladat megoldására két újszerű megközelítést alkalmaztam. Az első esetben definiáltam egy függvényt, míg a második esetben autoenkóder segítségével tanítottam meg a tumordinamikát a hálózat számára. Mind a két algoritmus kimenete egy vektor, melynek kiszámítva az interkvartilis terjedelmét meghatároztam a potenciális kiugró értékeket. Az algoritmusok létrehozása során interpolált idősorokkal dolgoztam, melyre a Py-

thon *pandas* könyvtárának *interpolate()* függvényét alkalmaztam, mely lineáris interpolációt alkalmaz a hiányzó értékek meghatározására.

A tumortérfogatok egyetlen időSORA Y_n -el jelölhető, a (3.1)-ben leírtak szerint, ahol az n az n -edik időSOR, a t pedig az időINDEXET jelöli. Például, ha a t értéke 1 és T között van – ahol T az n -edik egér mérésének hossza napokban –, akkor az n -edik időSOR a következő képlet adja meg:

$$Y_n = \{y_{n,1}, y_{n,2}, \dots, y_{n,T}\}, \quad (3.1)$$

ahol $y_{n,t}$ az n -edik időSOR értéke a t időPONTBAN (tumortérfogat az adott napon). Esetünkben 53 egérhez tartozó időSOR állt rendelkezésre, tehát $n = 1, 2, 3, \dots, N$, ahol $N = 53$. Az összes időSOR halmaza a következőképpen jelölhető:

$$\mathcal{Y} = \{Y_n : n \in N\}. \quad (3.2)$$

Kiugró értékek detektálása a tumortérfogatokat tartalmazó időSORokban

Az első intuitív megközelítés az anomáliák kiszűrésére egy n -edik időSOR értékeiből az, hogy a rendelkezésre álló tumortérfogatok közötti különbségekből származó információkat használjam fel. A tumortérfogat különbségek vizsgálatán alapuló algoritmus folyamatábrája a 3.1. ábrán látható. Az időSOROK interpolációja során a $t + 1$ és a t idők közötti távolság különbsége 1. A valós mérési adatok interpolációját követően kiszámoltuk a tumortérfogatok különbségeit, amelyek a következőképpen írhatók fel:

$$\Delta y_t = y_{t+1} - y_t. \quad (3.3)$$

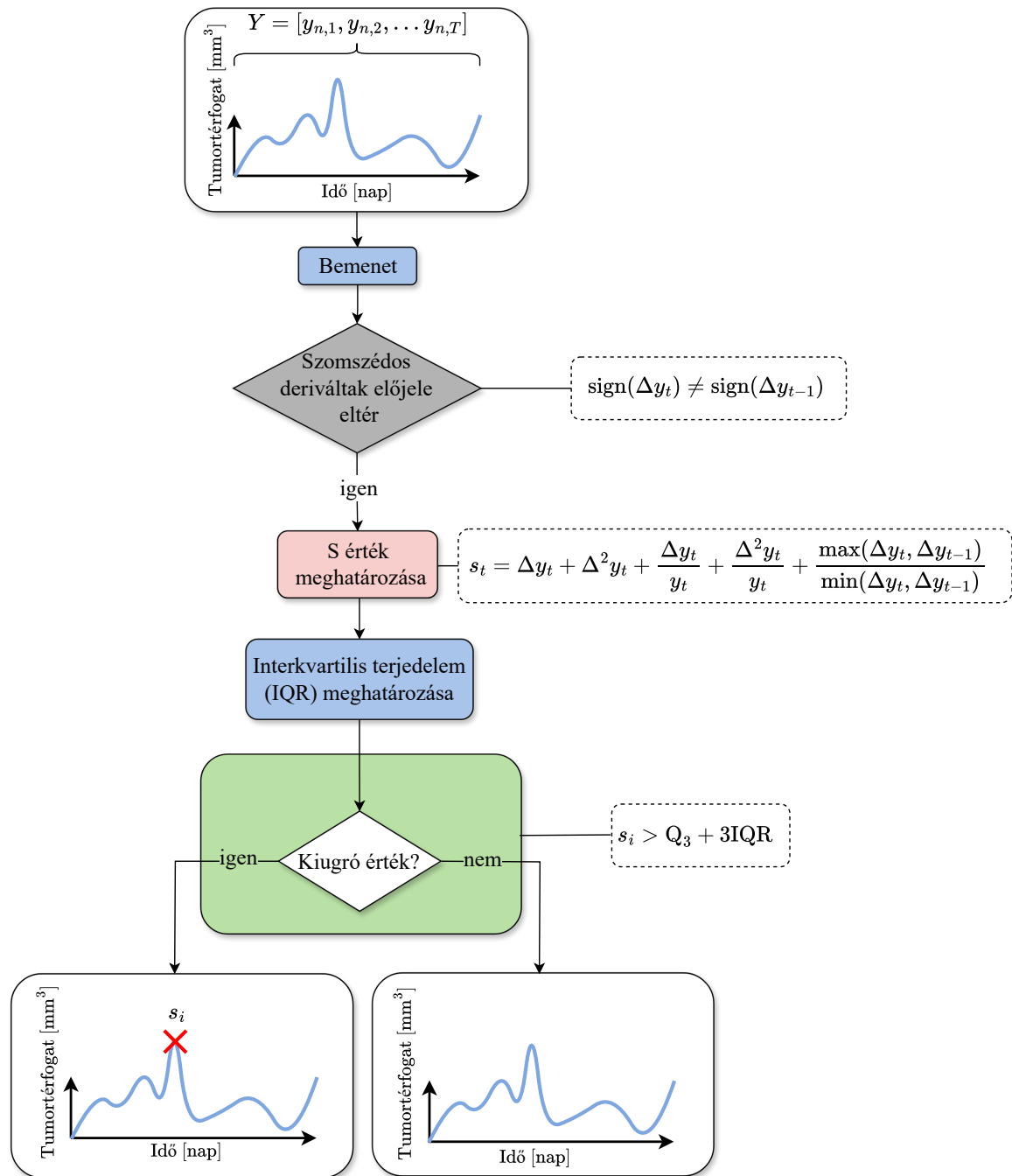
Ezenkívül kiszámítottam a másodrendű különbséget is:

$$\Delta^2 y_t = \Delta y_t - \Delta y_{t-1} = y_{t+1} - 2y_t + y_{t-1}. \quad (3.4)$$

Először a változások előjeleit hasonlítottam össze a számított differenciákat felhasználva. Csak azokat az értékeket vettük figyelembe, ahol a meredekség előjele ellentétes volt az egymást követő értékeknél. Figyelmen kívül hagytam azokat az adatokat, amelyek nem feleltek meg tehát a $\text{sign}(\Delta y_t) \neq \text{sign}(\Delta y_{t-1})$ kritériumnak. A megmaradt adatok felhasználásával a (3.5) pontban leírtak szerint pontszámot állítottam össze. Ez az s pontszám annak meghatározására lett létrehozva, hogy egy adott mérési pont milyen mértékben minősíthető kiugró értéknek:

$$s_t = \Delta y_t + \Delta^2 y_t + \frac{\Delta y_t}{y_t} + \frac{\Delta^2 y_t}{y_t} + \frac{\max(\Delta y_t, \Delta y_{t-1})}{\min(\Delta y_t, \Delta y_{t-1})}, \quad (3.5)$$

ahol Δy_t , Δy_t , $\frac{\Delta y_t}{y_t}$, $\frac{\Delta^2 y_t}{y_t}$, $\frac{\max(\Delta y_t, \Delta y_{t-1})}{\min(\Delta y_t, \Delta y_{t-1})}$ nagyságai azt jelzik, hogy az adatpont mennyiben tér el a mellette lévő értékektől. A kifejezés utolsó tagja az egymást követő különbségtagok



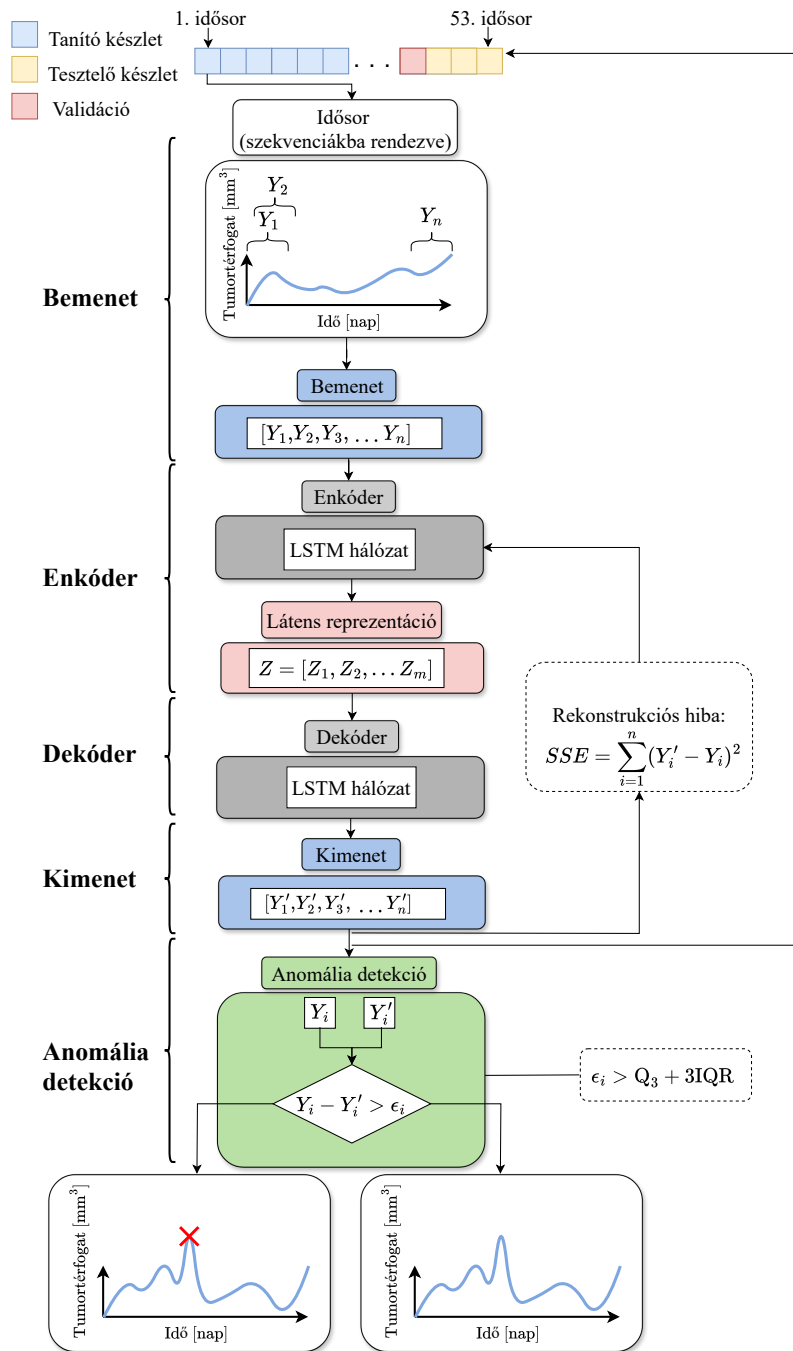
3.1. ábra. Az tumortérfogatok közötti differenciák vizsgálatán alapuló kiugró érték meghatározás folyamatábrája.

arányát írja le. A számláló értéke tehát minden esetben nagyobb, mint a nevező értéke, ezzel biztosítva, hogy minden egyes tagban nagyobb nagyságrendű érték a potenciális kiugró értékeket jelezze. Ez a pontozási módszer s_n értéket ad minden égerre (minden idősorra), ahol minden elem a t időindexhez társított pontszámokból áll:

$$s_n = \{s_{n,t} : t \in T\}. \quad (3.6)$$

A kidolgozott képlet segítségével elemeztem az s_n halmazban lévő értékek interkvartilis tartományát (IQR). Esetemben az értéket kiugró értékeknek tekintettük, és a t időpontban végzett mérést anomáliának jelöltük, ha kívül esett a $Q_3 + 3IQR$ határon, ahol $IQR = Q_3 - Q_1$ és Q_3 (felső kvartilis) az adatok alsó 75%-át, míg Q_1 (alsó kvartilis) az adathalmaz 25 %-ának a határát jelöli. Fontos megjegyezni, hogy az s_n beállított értékek normalizálásra kerültek a statisztikai mérőszámok kiszámítása előtt. Ennek a normalizálásnak az volt a célja, hogy biztosítsa az összes kifejezés egyenlő súlyozását a végső értékben. Következésképpen a normalizálás után a (3.5) minden egyes tagja 0 és 1 között változott.

Az általam kifejlesztett algoritmus hatékonyan képes volt az anomáliák észlelésére a rendelkezésre álló idősor-adatokban (melyet a 4. fejezetben ismertetek); ugyanakkor a túllilleszkedés kockázatának minimalizálása érdekében egy második, általánosabb megközelítést is implementáltam. A második – autoenkóder-alapú – algoritmus folyamatábrája a 3.2. ábrán látható. Az autoenkóder a 2.3. alfejezetben bemutatottak alapján képes megtanulni idősorok dinamikáját és azokat rekonstruálni. Ebből adódóan bármilyen anomália vagy kiugró érték rekonstrukciós hibaként jelenik meg a kimeneten. Esetemben az autoenkóder célja, hogy a rendelkezésünkre álló 53 darab tumortérfogatot tartalmazó időorból (\mathcal{Y}) megtanulja a tumordinamikát, tehát a már betanított hálózat egy kiugró értéket tartalmazó idősor esetén képes legyen azt detektálni. Az autoenkóder két LSTM neurális hálózatból épül fel. A bemenet az 53 darab idősor, azaz az 53 tumortérfogat mérés, eltérő hosszal. Az LSTM hálózat esetében lehetőség van arra, hogy eltérő hosszúságú idősorokkal tanítsuk be a hálózat súlyait. Ez azzal a konfigurációval oldható meg, ha úgynevezett *padding*-et alkalmazunk. Az utolsó tumortérfogat mérést követő fennmaradó értékeknek például -10 értékeket adunk meg, majd a hálózat felépítésénél a bemeneti rétegnél ezt az értéket megadjuk. A bemenet az idősorokat szekvenciákként kapta meg, ahol egy szekvencia (Y_n) összesen 5 tumortérfogat értékből áll. Minden idősort felbontottam mozgó ablakkal összesen 298×5 nagyságú mátrixokra, úgy hogy minden új 5 elemet tartalmazó vektort egy időegységgel eltoltam az előzőhöz képest. A 298-as érték az 53 egérből a leghosszabb mérési időintervallum alapján lett megállapítva. Ennek következtében az autoenkóder bemenete az összes tumortérfogat idősort tartalmazva $52 \times 298 \times 5$ nagyságú. Minden tanítás során egy darab idősort kivettem a teljes adathalmazból a keresztvalidáció során történő teszteléshez, melyet a 4. fejezetben részletezek.



3.2. ábra. A kiugró érték meghatározására alkalmazott, autoenkóderen alapuló algoritmus folyamatábrája. A betanított autoenkóder előrejelzéséből és az eredeti idősorok különbségéből rekonstrukciós hiba számítható. A rekonstrukciós hiba nagyságát alkalmaztam a kiugró értékek indikátoraként.

Az enkóder első rétege egy *Masking* réteg, mely a különböző hosszúságú idősorok kompenzálásaként van jelen. Ezt három LSTM réteg követi, ahol a neuronok száma rendre 128, 64, és 32. A dekóder hálózat bemenete az enkóder kimenete, tehát 32 hosszúságú vektor. Ezt követően egy *RepeatVector* réteg követi, mely plusz egy dimenziót ad a bemenethez, tehát az enkóder kimenetéhez. Ezt követően a dekóder LSTM hálózat is három LSTM réteggel rendelkezik, melyek 32, 64, 32 neuront tartalmaznak. Ezt az architektúrájú hálózatot tanítottam be az $52 \times 298 \times 5$ nagyságú idősor szekvenciákat tartalmazó bemenettel. Egy neurális hálózat 52 idősor adatai alapján tanult be, újratanítással, így mivel egy idősor betanítása 25 epochból állt, összesen $52 \times 25 = 1300$ epoch alatt tanult be. A tanítás során *Adam* optimalizálót alkalmaztam. Az LSTM autoenkóder létrehozásához és betanításához a *Tensorflow* könyvtár *keras.Model* osztályát alkalmaztam.

Zajszűrés a tumortérfigatokat tartalmazó idősorokban

A kiugró értékek kiszűrését követően az adatok zajszűrése történt. A zaj és a kiugró érték között a különbség, hogy míg előbbi számomra értékes információt tartalmaz, addig az utóbbit szerettem volna eltávolítani, hiszen hamis információt ad az adatokról. Tehát első lépésben fontos volt a kiugró értékek meghatározása, melyet követhetett a zaj csökkentése.

Az idősorok valós mérésein Wavelet-transzformációt alkalmaztunk a zaj kiszűrésére. A Wavelet-transzformáció egy módszer, amelyet a jelfeldolgozásban használnak, beleértve számos adattípust, például hangjeleket és képeket. A Fourier-transzformáció kiterjesztett változatának tekinthető, mivel szintén a jelek különböző komponenseire történő szétválasztására alkalmas. Mindazonáltal részletesebb információt nyújt, mint a hagyományos Fourier-transzformációk, különösen a nem stacionárius jellemzőket mutató jelek elemzése esetén.

A diszkrét Wavelet-transzformáció (DWT) során a zajtalanítási folyamat három fő szakaszból áll: a jel felbontása, küszöbérték alkalmazása és az eredeti jel rekonstrukciója [40]. A jel felbontásának eredménye egy hierarchikus felbontás, ahol a jel alacsony frekvenciájú és nagy frekvenciájú összetevőkre van felbontva. Ez a lépés hasonló az alul- és felüláteresztő szűrő alkalmazásához. Általában a küszöbértéket a jel felbontása után alkalmazzák, arra, hogy az annál nagyobb értékekű komponenseket nullázzuk vagy értéküket lecsökkentsük. A Wavelet-transzformációnak a zajcsökkentésre való felhasználásáról szóló irodalom számos küszöbértéket tartalmaz, amelyek közül a kemény és a lágy küszöb a leggyakrabban alkalmazott módszerek [41]. A feltételezett zajt reprezentáló kisebb együtthatók eltávolításával és a fennmaradó zajmentes együtthatókkal a rekonstrukciós algoritmus alkalmazásával az eredeti jel visszaállítható az inverz diszkrét Wavelet-transzformációval.

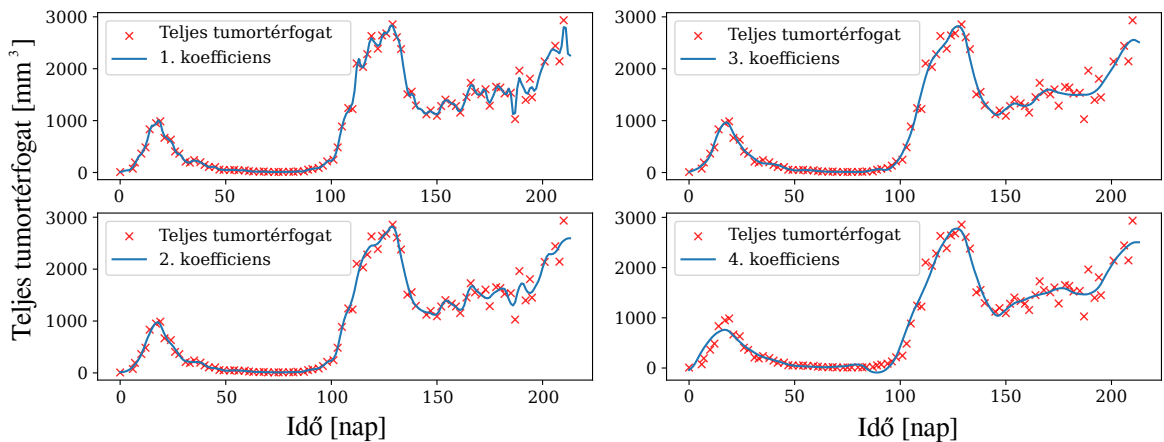
A zajcsökkentéshez először a jelet (esetünkben a tumor idősorát) N szintű együtthatókra bontottam. Az első lista tartalmazta az alacsony frekvenciájú együtthatókat tartalmazó mátri-

xokat, a következő lista pedig a magas frekvenciájú együtthatókat. A bemenet zaját a magas frekvenciák kiszűrésével csökkentettem, ami azt jelenti, hogy a koefficiensek egy részét nullára állítottam az alábbiak szerint:

$$T(W, \lambda) = \begin{cases} 0 & \text{ha } j < \lambda, \\ W & \text{különben.} \end{cases} \quad (3.7)$$

A λ -t háromnak választottam, míg a wavelet függvénynek a *Symlets sym5* függvényt határoztam meg. A küszöbértékek meghatározása után a T értékekből rekonstruáltam a jelet. A folyamat során *PyWavelet*-t Python könyvtárat használtam [42].

A 3.3. ábrán láthatunk egy példát a wavelet dekompozícióra és rekonstrukcióra különböző λ küszöbértékekkel. Fentről lefelé haladva a λ értéke növekszik, ami azt jelenti, hogy a részletesebb együtthatók (magasabb frekvenciák) nullára kerülnek.



3.3. ábra. A Wavelet-transzformáció alkalmazásának eredményei idősoros adatokon. A piros „x” jelek a megfigyelt tumortérfogat-méréseket jelölik, beleértve a zajt is, míg a kék vonal a Wavelet-transzformáció alkalmazása után közelített értékeket mutatja. A koefficiensok számának növekedésével nő a nullára állított magas frekvenciájú komponensek száma.

3.2. Idősorok klaszterezése tumordinamika alapján

Virtuális páciensek létrehozása a klaszterezéshez

A tumortérfogat mérések klaszterezése során a cél az eltérő tumordinamikát mutató virtuális páciensek eltérő csoportokba való szelektálása volt egy felügyelet nélküli algoritmus segítségével. Egy virtuális páciens alatt adott tumortérfogatokat tartalmazó idősorhoz kapcsolódó

paraméterhalmazt értjük. Az eltérő tumordinamikák meghatározását követően a paraméterek értékei is csoportosíthatók, ezáltal pedig információ nyerhető az adott betegről. A klaszterek alapján adott tumordinamikához tartozó paraméterintervallumok is felállíthatók, melyek jobb kezdeti értéket biztosíthatnak a terápiagenerálás során vagy a paraméterek identifikációja esetében.

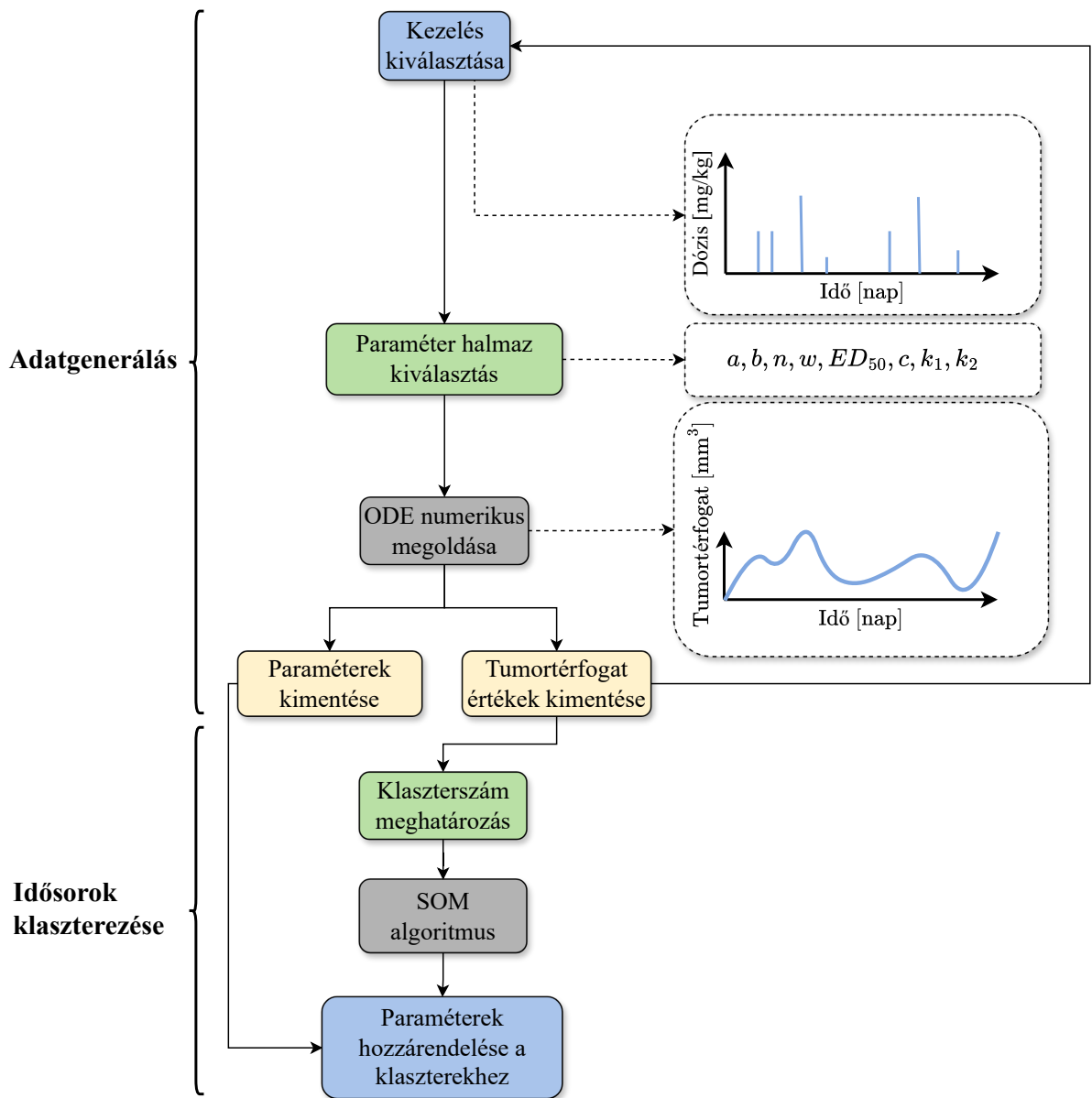
A létrehozott algoritmus a 3.4. ábrán látható. Az algoritmus kiindulási lépése az adott egér kiválasztása a rendelkezésre álló 53 egérből. Az egér kiválasztása alatt a kezelések értékeit fixáljuk, tehát a rendszer bemenetét (u) egy vektor formába rendezzük. Ezt követően egyenletes eloszlásból előre meghatározott minimum és maximum paraméter intervallumból generálunk paraméter értékeket. A pszeudo-randomszám generáláshoz a Python nyelv *numpy* nyílt forráskódú könyvtárának a *random.Generator.uniform* metódusát alkalmaztam. A megadott minimum és maximum értékek korábbi paraméterillesztési eredményekből származnak, ahol az 53 egér összes meghatározott paraméterei közül vettem a minimum és maximumot.

A paraméterek és a bemenetek ismeretében minden napra meghatároztam a tumortérfogatot a 2.1. alfejezetben bemutatott modell alapján. A közönséges differenciálegyenlet-rendszer numerikus megoldása a *scipy* csomag *integrate.odeint* metódusával történt. A kezdeti értékek a differenciálegyenlet megoldása során $x(0) = (x_1(0), x_2(0), x_3(0), x_4(0))$ formában írhatók fel, ahol x_1 az élő tumortérfogat, x_2 a halott tumortérfogat, x_3 a gyógyszerszint a vérben míg x_4 a gyógyszerszint a szövetben a kezdeti 0 időpillanatban. Esetünkben minden kezdeti érték meghatározása során $x_1(0)$ -t a teljes tumortérfogattal közelítettem, míg a többi állapotváltozót nullának vettem. Továbbá x_3 értékéhez minden nap a szimuláció során hozzáadtam az adott napi beadott kemoterápiás szer mennyiségét. Tehát a kezdeti értékeket tartalmazó vektor $x(0) = (y(0), 0, u(0), 0)$ formában írható fel.

A szimuláció során szükség volt a tumortérfogatoknak egy felső határt szabni az exponenciálisan elszálló tagok következtében, így a maximális tumortérfogat értékének minden egér esetében az adott egyed mérései során meghatározott maximális tumortérfogat értéket vettem. A mérés minden napjára szimulációval meghatároztam a tumortérfogatok értékeit. Ezeket egy két dimenziós mátrixba gyűjtöttem, ahol minden sor egy adott paraméterekkel rendelkező egyedre jelent, míg minden oszlop az adott egyed tumortérfogatát az adott napon.

Összesen egerenként 20 000 darab tumortérfogatot tartalmazó idősor került lementésre a 3.1. táblázatban látható formában. A táblázatban minden sora egy új idősort jelent, új paraméterekkel. Tehát a táblázat ugyanannak a kezelésnek az eredményeit mutatja eltérő paraméterek mellett minden napra.

A 3.1. táblázat mellett lementettem továbbá a hozzájuk tartozó paraméterek értékeit is a 3.2. táblázat szerinti formában. A két táblázatot összekötő egyedi kulcs a sorszámuk. Ez alapján a 3.1. táblázat pirossal kiemelt első sorában szereplő tumortérfogat értékek paraméterei a



3.4. ábra. A tumortérfogat értékeket tartalmazó idősorok klaszterezésére létrehozott algoritmus folyamatábrája. Az algoritmus a folyamat elején kiválaszt egy kezelést, virtuális tumortérfogat méréseket generál hozzá, majd csoportosítja azokat tumordinamikák alapján.

Sorszám	1. nap	2. nap	3. nap	4. nap	5. nap	6. nap	.	.	.	105. nap
1.	423.30	461.50	489.14	504.65	508.69	505.59	.	.	.	1.25
2.	423.30	511.52	597.38	676.02	745.44	810.76	.	.	.	5651.50
.
.
.
20 000.	423.30	634.59	911.70	1262.92	1701.37	2268.15	.	.	.	7277.66

3.1. táblázat. Az idősorokat (tumortérfogat értékeket) tartalmazó táblázat. A táblázat minden sora ugyanazon kezeléshez tartozó lehetséges tumortérfogat értékeket tartalmazza. A sorok a szimuláció során felhasznált paraméterek értékeiben térnek el.

tumormodell alapján a 3.2. táblázat a szintén pirossal kiemelt első sorában láthatók. A c , k_1 , k_2 értékei minden esetben azonosak, mivel ezek a modell farmakokinetikai paraméterei (c , k_1 , k_2).

Sorszám	a	b	n	w	ED ₅₀	c	k ₁	k ₂
1.	0,3189	8,2970	0,0132	0,0820	2,2223	1,8211	14,0008	136,2781
2.	0,5011	10,5124	0,0131	0,0799	2,1342	1,8211	14,0008	136,2781
.
.
.
20 000.	0,3870	7,5731	0,0151	0,0839	2,1685	1,8211	14,0008	136,2781

3.2. táblázat. A 3.1. táblázatban található tumortérfogat idősorok generálása során felhasznált paraméterek értékei.

Idősorok klaszterezése

A 20 000, különböző paraméterekkel rendelkező idősor létrehozását követően, az idősorokat csoportosítottam hasonló tumordinamika alapján. A csoportosítás önszerveződő térkép (SOM - Self-Organizing Map) alkalmazásával történt, mely egy klaszterező algoritmus [43]. A SOM egy, az irodalomban elterjedt módszer az idősorok csoportosítására számos területen [44, 45], többek között rendszeridentifikáció esetén is használatos [46, 47]. Korábbi munkákban már alkalmazták a tumordinamika és a farmakokinetika jellemzőinek kinyerésére és összefüggések meghatározására a tumortérfogatot tartalmazó idősorok alapján [48].

A klaszterezés lényegében felügyelet nélküli tanulási módszer, ahol címkével nem rendelkező tulajdonság vektorok állnak rendelkezésünkre. A klaszterezés célja, hogy az egymáshoz hasonló adatpontokat ugyanabba a klaszterbe csoportosítsa, miközben a különböző adatpontokat különböző klaszterbe szétválassza [49]. A hasonlóságot jellemzően távolság- vagy hasonlósági metrika határozza meg, például euklideszi távolság. A klaszterezés tehát felírható egy optimalizációs problémaként az alábbiak szerint:

$$WCSS = \sum_{i=1}^k \sum_{x \in C_i} \|x - c_i\|^2, \quad (3.8)$$

ahol $WCSS$ (Within-Cluster Sum of Squares - Klaszteren belüli négyzetösszegek) az adatpontok és a hozzájuk tartozó klaszterközéppontok közötti négyzetes euklideszi távolságok összege. Továbbá c_i a C_i klaszter középpontja, és c az összes adat középpontja. A cél tehát a diszperzió csökkentése, mely azonos klaszteren belüli elemek közötti távolság. Tehát a (3.8) egyenlőség felfogható úgy, mint egy mérőszám arról, hogy a klaszterünk mennyire homogén vagy heterogén elemeket tartalmaz. Értelemszerűen ezt igyekszünk minimalizálni, hiszen az a cél, hogy minél hasonlóbb elemeket tartalmazzon minden csoport, tehát minél jobban tudjuk őket tulajdonságaik alapján csoportosítani. Ha a fenti egyenlet szerint összeadjuk az összes klaszterhez tartozó diszperziót, akkor megkapjuk, hogy a teljes adatkészletre hogy teljesít az algoritmusunk.

Továbbá a másik cél, hogy a klaszterek közötti távolságot szeretnénk maximalizálni, mely az alábbi formában írható fel,

$$BCSS = \sum_{i=1}^k m_i \|c_i - c\|^2, \quad (3.9)$$

ahol $BCSS$ (Between-Cluster Sum of Squares - Klaszterek közötti négyzetösszegek) a klaszterközéppontok (átlag) és az összes adat középpontja (átlag) közötti négyzetes euklideszi távolságok súlyozott összege. Továbbá m_i a C_i klaszterben lévő elemek száma [50].

A klaszterezés során tehát a cél, hogy (3.8) kifejezést minimalizáljuk, míg (3.9) függvényt maximalizáljuk. Ha a két függvénynek szeretnénk optimumot találni, akkor egy triviális megoldás az, ha minden klaszter csupán egy darab elemet tartalmaz, tehát a klaszterszám megegyezik az adatbázisunk elemszámával. Ezáltal a különbözőség a klaszteren belül nulla. Ebből adódóan az algoritmusnak további korlátokat kell megadni, mint például a klaszterek közötti távolságnak egy alsó limit megadása vagy magának a klaszterek számának a megadása. Esetünkben a klaszterszám megadása kézenfekvőbb megoldásnak bizonyult, így a klaszterezést megelőzően minden esetben meghatároztam a klaszterszámot. A klaszterszám meghatározása során az irodalomban elterjedt ökölszabályt alkalmazom, mely szerint a klaszterszám az adatméret felének négyzetgyöke.

A klaszterezés előtt a bemenetet normalizáltam, a következők szerint:

$$Y_{\text{norm}} = \frac{Y - Y_{\text{min}}}{Y_{\text{max}} - Y_{\text{min}}}, \quad (3.10)$$

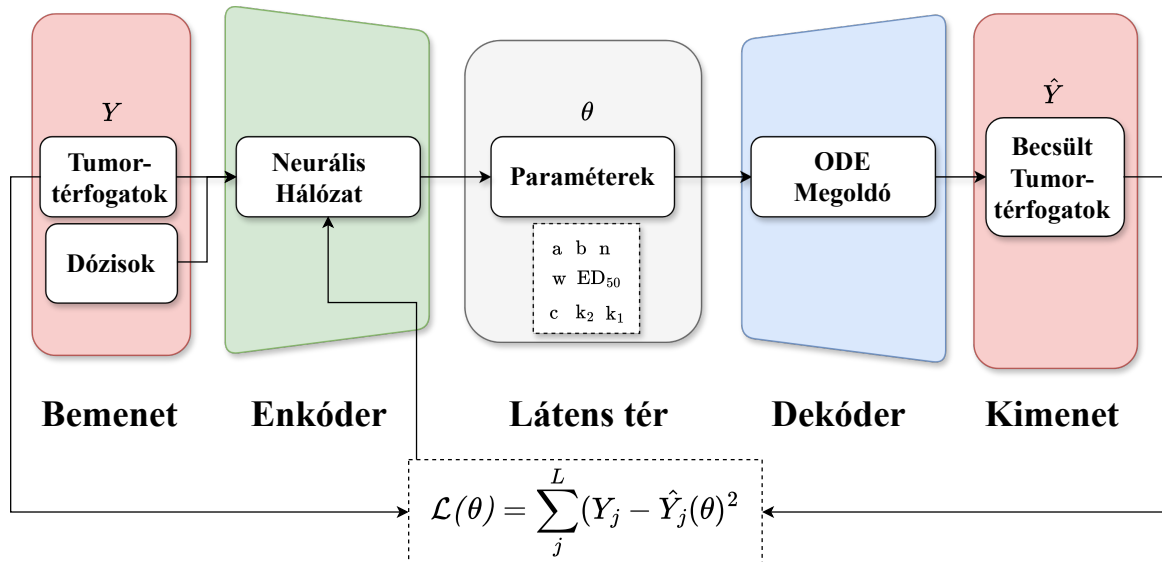
ahol Y az eredeti tumortérfogatokat tartalmazó vektor, míg Y_{max} az Y vektor maximum mértéke, míg Y_{min} a minimuma. Y_{norm} a normalizált tumortérfogatokat tartalmazó vektor. A SOM algoritmus tanítása során a hiperparaméterek az alábbi értékeket vették fel: $\gamma = 0.3$, $\eta = 0.1$, epoch = 5000. A (2.21) egyenlet alapján γ értéke meghatározza, hogy a súlyfrissítés mekkora szomszédságra terjed ki a BMU neuronnak. Az η értéke a súlyfrissítés nagyságát, tehát a tanulási sebességet befolyásolja, míg az epoch, az iterációk számát jelenti a tanulási folyamat során. A hiperparaméterek meghatározása empirikus úton történt.

3.3. Paraméterbecslő autoenkóder létrehozása

A munkám során a második algoritmus mely a tumormodell paramétereinek meghatározására alkalmas, egy speciális autoenkóder-alapú algoritmus. A létrehozott hálózat a becslt és az eredeti tumortérfogatokat tartalmazó idősorok közötti különbségek minimalizálása által képes a paraméterek meghatározására és megtanulására.

A 3.5. ábrán látható a létrehozott teljes rendszer felépítése, melynek a struktúrája megfelel egy autoenkódernek. Az autoenkóderek két fő komponensből épülnek fel: egy enkóderből és egy dekóderből. Az enkóder feladata, hogy a bemeneti adatokból állítson elő egy tömörített reprezentációt, míg a dekóder feladata, hogy a tömörített reprezentációt visszaalakítsa az eredeti bemeneti adatokhoz lehető legközelebb álló formájára. Annak mértékét, hogy mennyire sikerült jól megközelíteni az eredeti adatokat, a rekonstrukciós hibával írhatjuk le, mely a bemenet és a kimenet közötti négyzetes különbsége.

Esetünkben a bemenetek a rendelkezésünkre álló tumortérfogatok és a beadott kemoterápiás szer dózisainak értékeiből alkotott adott hosszúságú idősorok. A hálózat tanításához nem szükségesek a modell alapján hozzátartozó paraméter értékek, azokat a tanítás folyamata alatt a rendszer magától határozza meg, azáltal, hogy a hálózat súlyait hangolja úgy, hogy a bemeneti tumortérfogat és az aktuális paraméterek által megszabott kimeneti tumortérfogat között a különbség minimális legyen. Az autoenkóder tehát esetünkben abban tér el az irodalomban elterjedt alapkonceptiótól, hogy csak az enkóder egy neurális hálózat, a dekóder nem az, nem tartalmaz tanítható súlyokat. A dekóder egy ODE megoldó, ami a paramétereiből és a kezdeti feltételekből előállítja a tumortérfogatokat. Az autoenkóder tanítása során először tanító adatokat generáltam a tumormodell felhasználásával. Ezt követően a generált adatok első felével előtanítottam a hálózatot, majd pedig a második felével betanítottam az autoenkódert.



3.5. ábra. A paraméterbecslő autoenkóder felépítése. Az autoenkóder a bemeneti tumortérfogatok és a becsült tumortérfogatok közötti különbség minimalizálásával határozza meg a bemeneti tumortérfogatokhoz tartozó paraméterek értékeit.

Tanító adatok generálása

Mivel valós, egereken végzett mérésekből nem áll rendelkezésünkre egy neurális hálózat betanításához elegendő adat, így a tumormodell felhasználásával létrehoztam 20000 szimulált bemeneti adatot tartalmazó adatkészletet. A tanító adatgenerálás során pseudo-randomszám generátorral állapítottam meg dózisokat és paramétereket adott intervallumon belül, majd pedig a kezdeti értékek és paraméterek ismeretében szimuláltam a 2.1. alfejezetben ismertett tumormodell kimenetét egy numerikus integrátor segítségével. Ezáltal dózisokat és tumortérfogatokat tartalmazó idősorokat kaptam. Ezeket két mátrixba gyűjtöttem, ahol minden sor egy idősort tartalmazott. A generált paramétereket szintén egy mátrixba gyűjtöttem, ahol a sorok szintén egy virtuális pácienshez tartoztak.

Egy tumortérfogatokat tartalmazó idősor szimulációja során először egyenletes eloszlásból generáltam a 2.1. táblázatban feltüntetett tartományból egy-egy értéket, mind az $a, b, n, w, ED_{50}, c, k_1$ és k_2 paraméterek esetében. Ezt követően virtuális kezelést generáltam, azáltal, hogy figyelembe vettem a 3.6. ábrán feltüntetett kezelési rendet. Az ábrán az látható, hogy a normál protokoll szerint minden hétköznap tumortérfogat mérés van, kedden és pénteken pedig kemoterápiás szer beinjekciózása történik. Ezt a szimuláció során úgy vettem figyelembe, hogy a randomszám generátorral létrehoztam minden keddi és pénteki napra nulla és nyolc $mg \cdot kg^{-1}$ nagyságú dózisokat, egyenletes eloszlással. A felső határa a beadott dózisoknak az

Dózis	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mérés	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Hétfő	Kedd	Szerda	Csütörtök	Péntek	Szombat	Vasárnap

3.6. ábra. A tanító adatok generálása során alkalmazott mérési elrendezés. Az alkalmazott protokoll alapján hetente kétszer, kedden és pénteken történik injekciózás, míg hétköznaponta történik tumortérfogat mérés.

állatkísérletek során alkalmazott felső határ volt.

Mivel a differenciálegyenletek megoldása függvények sokasága, így az adott feltételekhez tartozó megoldást keressük, így a numerikus közelítő módszerek esetében is szükséges a kezdeti értékek megadása, amivel leszűkíthető a keresési tartomány. A tumormodell négy állapotváltozót tartalmaz, így mindegyik változónak szükséges megállapítani az értékét a kezdeti időpillanatban. A szimuláció során létrehoztam egy kezdeti értékeket tartalmazó mátrixot is, ahol minden sor szintén egy virtuális pácienshez tartozó kezdeti érték vektor, mely négy hosszúságú és tartalmazza az x_1 , x_2 , x_3 és x_4 értékeit a $t = 0$ időpillanatban. Az $x_1(0)$ értékét – mely az élő tumor térfogata – a teljes tumor értékével közelítettem, azzal a feltételezéssel élve, hogy a mérés elején még nincs elhalt tumor rész és randomszám generátorral határoztam meg, 0 és 200 mm³ között egyenletes eloszlással. Ennek következtében az elhalt tumor mennyiségét ($x_2(0)$) viszont nullának vettem. A gyógyszer koncentrációjának értékét a vérben ($x_3(0)$) és a szövetben ($x_4(0)$) szintén nullának vettem a kezdeti időpillanatban.

Tehát a négy – dózisokat, tumortérfogatókat, paraméterek, kezdeti értékeket tartalmazó – mátrix sorainak a sorszámát tekinthető az elsődleges kulcsnak vagy azonosítónak, ami összeköti a három tanításra létrehozott táblát. Minden táblázat adott sora egy adott virtuálisan generált páciens kimentett értékeit tartalmazza. Minden virtuális páciens esetében a szimulációkat mindig egy napra futtattam le *torchode* [51] közöséges differenciálegyenlet megoldójával, a kapott tumortérfogatókat kimentettem a hétköznapokon, kedden és pénteken pedig hozzáadtam az x_3 változóhoz a generált adott napra eső dózist. Összesen 20000 virtuális páciens generáltam, tehát az idősorokat tartalmazó mátrixok 20000×105 nagyságúak voltak, a kezdeti értékeket tartalmazó mátrix 20000×4 -es, a paramétereket tartalmazó pedig 20000×8 -as méretű volt. A 20000 nagyságú adatbázist elfeleztem, első felét előtanításra a második felét pedig az autoenkóder tanítására használtam fel.

A hálózat előtanítása

Ahhoz, hogy az autoenkóder a tanítás során olyan paramétereket állítson elő, amelyek biológiailag értelmezhető nagyságrendbe esnek, a hálózatot először előtanítottam a paraméterek ismeretében. A paraméterek többsége nap^{-1} mértékegységű, ennek megfelelően egy adott skálán mozognak, melynek van egy biológiailag limitált alsó és felső határa. Az előtanításra továbbá amiatt is szükség volt, mivel az enkóder kimenete a tumormodell paraméterei, ami egyben a bemenete az ODE megoldónak. Mivel a numerikus integrátor csak adott intervallumon belüli értékekkel tudja szimulálni a tumormodell kimenetét, a bemeneti paraméterek nem vehetnek fel tetszőlegesen nagy számot, különben numerikusan instabil kimenetet eredményezhetnek. Ha nem tanítottam volna elő a hálózatot, akkor olyan paraméterkombinációkat is előre jelzett volna, melynél az ODE megoldó a hirtelen nagy változások esetében a kimeneten nem tudja lekezelni az adaptív lépésköz választást.

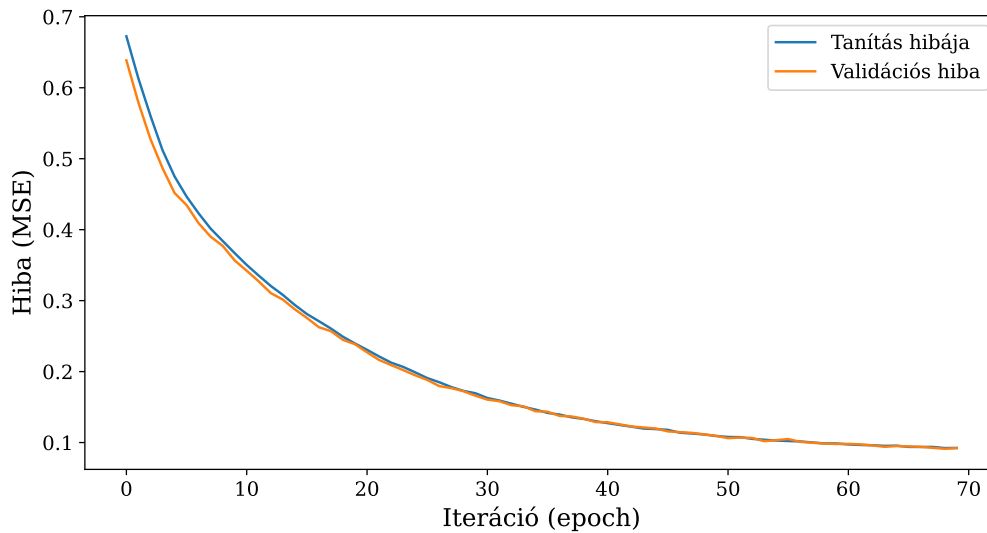
Az előtanítás során a hálózat bemenete két idősor volt: a tumortérfogatokat és a dózisokat tartalmazó mátrixok sorai, míg a kimenete a paraméterek voltak. A bemeneteket és a kimeneteket normalizáltam mielőtt betanítottam a hálózatot. Normalizálás során minden változót 0 és 1 közé skáláztam át, min-max normalizációval az alábbiak szerint:

$$p'_j = \frac{p_j - p_j^{\min}}{p_j^{\max} - p_j^{\min}}, \quad (3.11)$$

ahol p'_j a j -edik paraméter normalizált értéke, míg p_j^{\min} és p_j^{\max} a j -edik paraméter határai a 2.1. táblázatban.

A tanítás során az adatkészletet felosztottam 0,70:0,15:0,15 arányokban tanító, validáló és teszt adatkészletre. A tanító adatkészletet használtam a hálózat előtanításához, míg a validációs adatkészletet arra, hogy a hálózat hiperparamétereit az előtanítás során hangoljam és a túlillesztést elkerüljem. Minden iteráció végén a validációs adatkészleten értékeltem ki a hálózat teljesítményét, viszont az ebből számított hibát nem terjesztettem vissza a hálózat súlyaira. A teszt adatkészletet a már betanított hálózat kiértékelésére használtam.

Az előtanítás folyamata során 10000 adatot használtam fel. Az adatkészletet 1000 nagyságú kötegekre osztottam. Egy teljes iteráció abból állt, hogy minden kötegre megtörtént a hiba kiértékelése és annak a visszaterjesztése a hálózat súlyaira, tehát összesen egy iteráción belül 10 alkalommal terjesztettem vissza a hibát. A folyamat során ún. korai leállási feltételt alkalmaztam, mely szerint a tanítás leállt, ha 5 iteráción keresztül nem változott a hiba értéke a validációs adatkészleten. A hiba számítása során átlagos négyzetes hibát alkalmaztam. A 3.7. ábrán látható az előtanítás során kapott átlagos négyzetes eltérés iterációnkénti értéke. Csak a görbéket kiértékelve megállapítható, hogy a hálózat be tudott tanulni a paraméterek érteire. A validáció és a tanítás görbéje hasonló ívet követ, mely alapján sem túl-, sem pedig



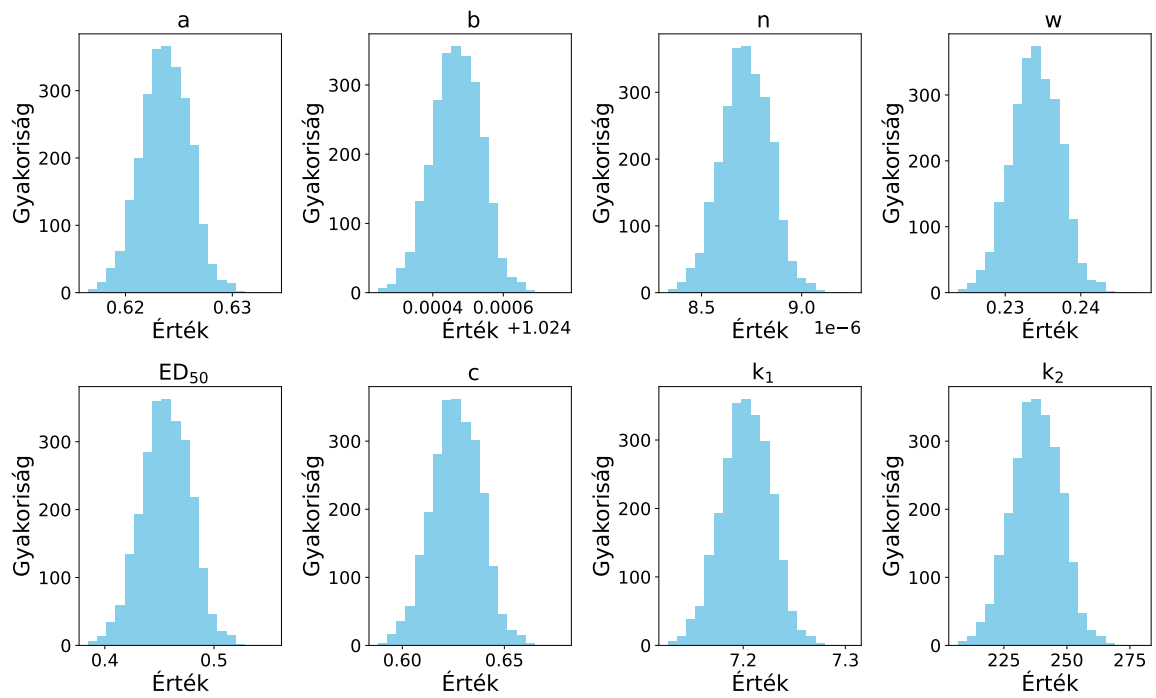
3.7. ábra. A neurális hálózat előtanítása során kapott hibák a tanító adatkészletre és a validációs adatkészletre. A hibák a becsült paraméterek és az eredeti paraméterek normalizált értékei közötti átlagos négyzetes eltérést mutatják.

alulillesztés nem figyelhető meg. Ha a validációs hiba egy idő után elkezdett volna nőni, viszont a tanítás alatt számított hiba nem, az azt jelentette volna, hogy a hálózat túlilleszkedett (overfitting) a tanító adatkészletre. Mindazonáltal a korai leállási feltétel ezt az esetet hivatott megakadályozni.

A teszt adatkészleten kiértékeltem az előtanított hálózat predikciós képességét. Az erre a célra elkülönített adatkészleten előre jeleztem a paraméterek értékeit majd pedig egy hisztogrammon ábrázoltam azokat. A paramétereket ábrázolást megelőzően denormalizáltam az eredeti intervallumukra. A 3.8. ábráról megállapítható, hogy egy szűkebb intervallumra tanult rá a hálózat, mint az eredeti intervallum. Ezáltal feltételezhető, hogy kevésbé jó általánosító képességgel rendelkezik. Mindazonáltal a cél az előtanítás során az volt, hogy a hálózat megtanuljon egy olyan intervallumból értékeket előre jelezni, amelyek az ODE megoldó bemeneteként szolgálhatnak stabil kimenetet produkálva.

Az enkóder: a neurális hálózat architektúrája és hiperparaméterei

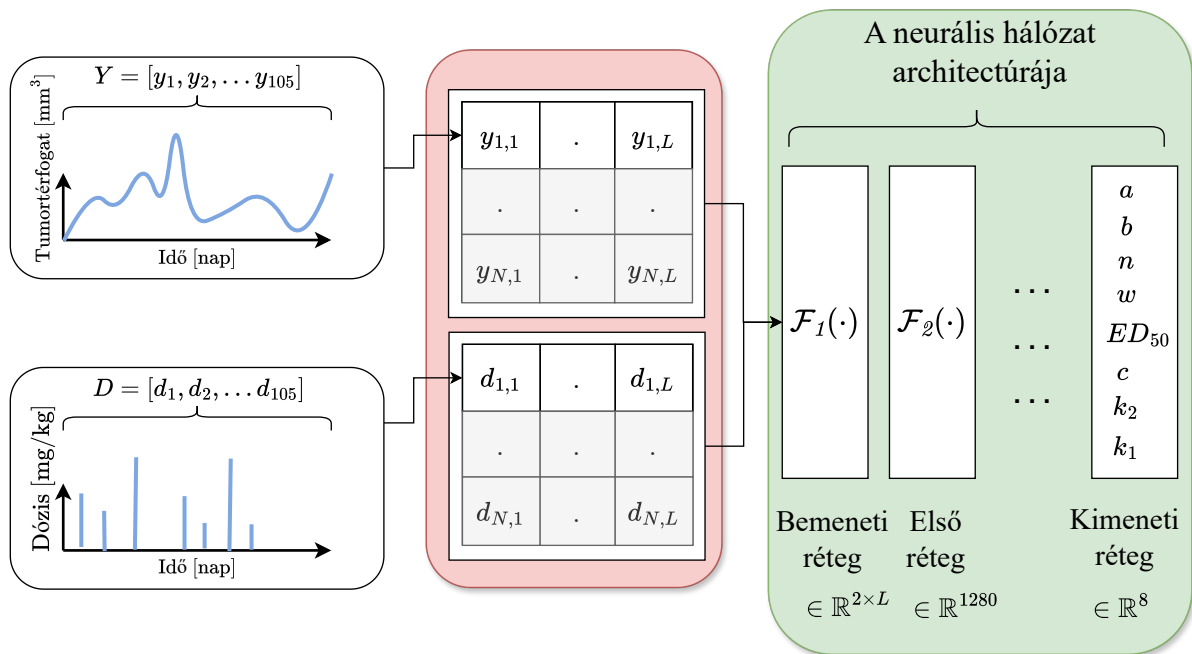
A 3.9. ábrán látható a 3.5. ábra bal oldala részletezve. Az autoenkóder bemenetei a tumortérfogatokat és a dózisokat tartalmazó két mátrix: mind a kettő $N \times L$ nagyságú. N a bemeneti adatok száma (esetünkben 10000), míg L a mérés hossza. A valós egérkísérletek során is a mért tumortérfogatok és a beadott dózisok alapján szeretnénk az adott egér paramétereit



3.8. ábra. A neurális hálózat előtanítását követően kapott paraméterek (a, b, n, w, ED₅₀, c, k₁, k₂) értékeinek gyakoriságai a tesztelésre elkülönített adatsoron.

meghatározni, majd a paramétereit alapján terápiát generálni.

A kiértékelés során három különböző hosszúságú tumortérfogat mérési hosszra tanítottam be a hálózatot és vizsgáltam meg a predikciós képességét. A mátrixok minden sora egy egyedi virtuális pácienset tartalmazott, ahogy az ábra mutatja. A két mátrixot normalizáltam, a maximum és minimum értékeik alapján, hogy egyforma súllyal vegye figyelembe a hálózat mind a két változót. A normalizálás (3.11)-hez hasonlóan történt.



3.9. ábra. Az autoenkóder első szakaszának a részletezett ábrája. Az enkóder rész a bemenetből és a neurális hálózatból épül fel. A hálózat bemenetei a tumortérfogatok és a dózisokat tartalmazó idősorok.

A hálózat hét rétegből épül fel. A bemenete a tumortérfogatok és dózisokat tartalmazó idősorok, összesen $2 \times L \times 1280$ nagyságú, míg a kimenete 64×8 méretű. A hálózat a bemeneti adatokat egyetlen vektorra alakítja, ami aztán a hálózat bemenetét képezi. Az első réteg, amely 1280 dimenziós kimenetet hoz létre, szigmoid aktivációs függvényeket használ, amelyet dropout követ. Az ún. dropout segít megakadályozni a túltanulást azáltal, hogy véletlenszerűen nullázza (kijti) az egyes bemeneteket a tanulás során. A dropout arányt 0,3-nak választottam meg.

A másodiktól az ötödik réteig minden réteg szintén 1280 egységet tartalmaz, és mindegyiket szigmoid aktiváció és dropout réteg követi. A hatodik réteg a kimenetét 64 dimenzióra csökkenti, ami előkészíti az adatokat a kimeneti réteg számára. Az utolsó (kimeneti)

réteg, amely lineáris kimenetet használ, az előre meghatározott kimeneti méretnek (paraméterek száma) megfelelően adja vissza az eredményt.

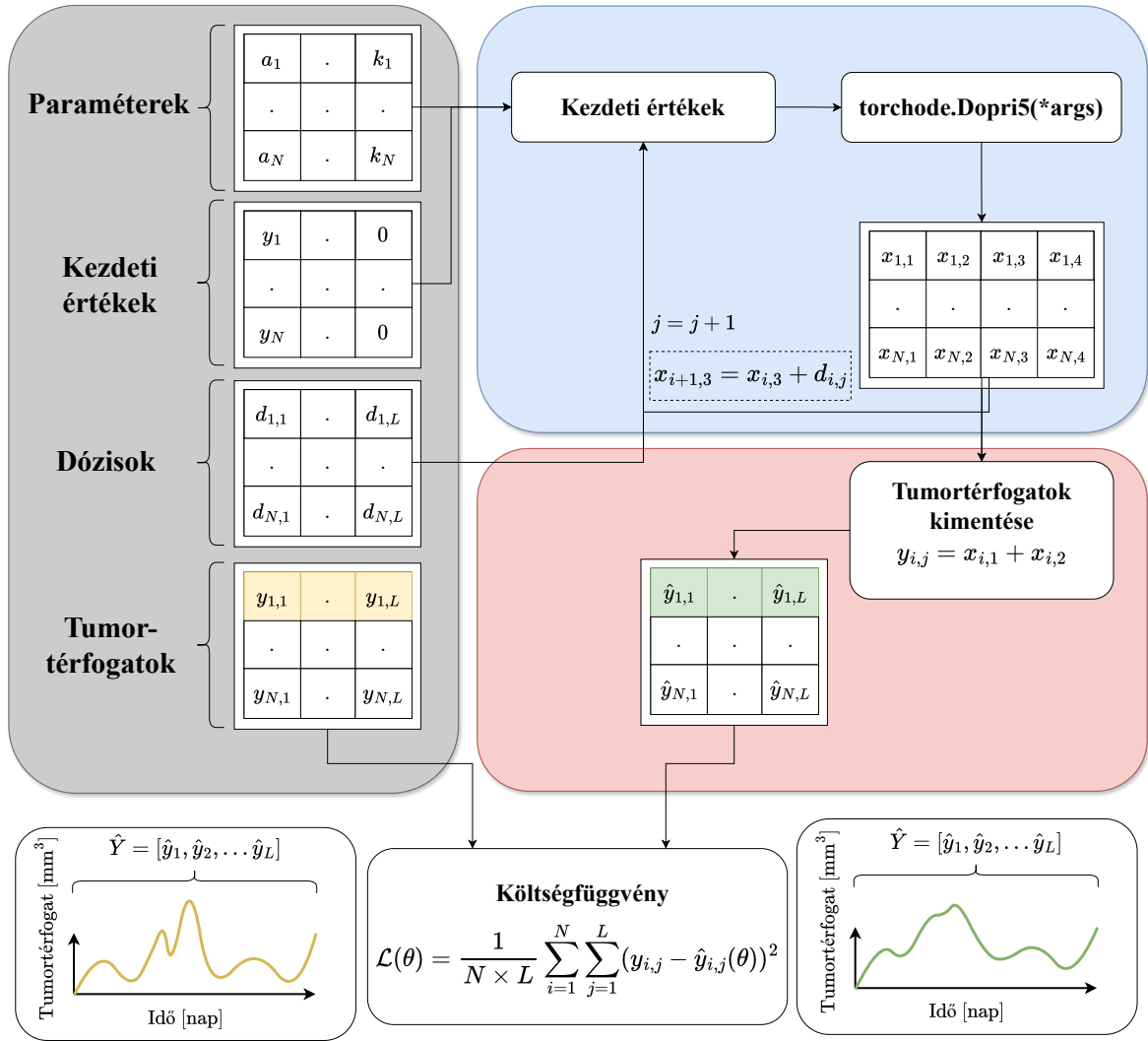
A hálózat tanítása során fontos volt az aktivációs függvény megválasztása. A kiválasztás során figyelni kellett arra, hogy mind az előtanítás, mind pedig az autoenkóder tanítása során jól teljesítsen a hálózat. Bár a ReLU aktivációs függvény az előtanítás során jobb teljesítményt nyújtott, az autoenkóderben rosszabbul teljesített. Továbbá a költségfüggvényben átlagos négyzetes hibával vizsgáltam az eltérést, mely nagyobb jelentőséget tulajdonít nagyobb hibáknak a négyzetes tulajdonságából adódóan. Ezáltal a hálózatot szerettem volna arra kényszeríteni, hogy azokra az idősorokra is rátanuljon, amik kevesebbszer fordultak elő, viszont nagyobb hibákkal. A tanuláshoz 10^{-3} tanulási sebességet választottam.

A tanításhoz NAdam (Nesterov-accelerated Adaptive Moment Estimation) optimalizálót választottam, mely az Adam optimalizáló algoritmust kombinálja a Nesterov momentummal. A különböző paraméterek adaptív tanulási sebességének meghatározásával megbecsüli a gradiensek első és második momentumát. Ezáltal a Nesterov momentum beépítésével [52] a konvergencia gyorsabbá válik.

A dekóder: a matematikai modell szimulációja

Az enkóder – a neurális hálózat – kimenetéből, azaz a paraméterekből és a tumortérfogatokból előállított kezdeti feltételekből egy ODE megoldó segítségével a rendszer kimenete előállítható. Mivel a differenciálegyenleteket nem tudjuk analitikusan megoldani, ezáltal numerikus módszerrel szimuláljuk a megoldását. A numerikus integráció a legelterjedtebb módszer a közelítő megoldások előállítására. Különböző integrációs módszerek léteznek, eltérő pontossággal és konvergencia sebességgel, mely közül a legelterjedtebb az Euler-módszer [53]. Mindazonáltal az Euler-módszer sok esetben nem kellően pontos, emiatt számos más módszer is elterjedt. A munkám során ötödrendű Dormand-Prince-módszert [54] használtam az egyenletek közelítésére. A Dormand-Prince-módszer explicit módszer, továbbá a lépésköz nagyságát automatikusan határozza meg. Emiatt fontos beállítani egy toleranciát, hogy mekkor hibát engedhet meg a szimuláció során. Az abszolút megengedett hiba toleranciáját 10^{-5} -nek, míg a relatív hiba toleranciáját 10^{-3} -nak vettem.

A neurális hálózatok tanítása során fontos, hogy a lépések differenciálhatóak legyenek és az adatok mátrixok formájában legyenek felírva. Előbbi amiatt elengedhetetlen, mert a hálózat autodifferenciálás segítségével terjeszti vissza a műveleteken keresztül a hibát a hálózat paramétereire a tanulás során. Ha olyan műveletet talál, mely nem differenciálható, akkor a hibavisszaterjesztési folyamat megszakad. A mátrixokba történő felírása az adatoknak a párhuzamosítás és a kötegenkénti adatfeldolgozás következtében számottevő. A teljes bemeneti adathalmazt a tanítás során általában kötegenként adjuk át a hálózatnak és egy-egy köteg



3.10. ábra. Az autoenkóder dekóder része, mely a paraméterekből, kezdeti értékekből és dózisokból előállítja a szimulált tumortérfogatot. A szimulált tumortérfogat és a valós tumortérfogatok közötti átlagos négyzetes eltérés értéke alapján tanítjuk az enkóder neurális hálózatát.

egészére történik a költségfüggvény kiszámítása, majd pedig a hibavisszaterjesztés. Ez jóval gyorsabb tanítást eredményez, mintha egyesével vezetnénk be a bemeneteket a hálózatnak, valamint kiegyenlítettebb tanulási folyamatot biztosít. Ahhoz, hogy kötegenkénti feldolgozást tudjunk végrehajtani, olyan differenciálható ODE numerikus integrátort használtunk a tanítás során, ami képes párhuzamosan több kezdetiérték problémát megoldani a futási idő számottevő megnövekedése nélkül [55].

A 3.10. ábrán látható a dekóder folyamatábrája. A dekóder bemenetei az enkóder által előre jelzett paraméterek. A paramétereket denormalizálom, mielőtt bevezetem az ODE megoldóba. A kezdeti értékeket, dózisokat és tumortérfogatókat tartalmazó mátrixokat az autoenkóder bemenetétől csatolom ide. A paraméterek ismeretében megoldom párhuzamosan annyi kezdetiérték problémára a tumormodellt, amekkora kötegméretet állítok be. Minden esetben egy napra oldom meg az egyenletet, majd az aznapi dózisokat hozzáadom az x_3 állapotváltozóhoz ($x_{i,3} = x_{i,3} + d_{i,j}$). Az i az adott kezdeti érték problémát jelöli (adott sort a mátrixokban), és N -ig tart, míg $j = 1, 2, 3, \dots, L$, ahol L a mérés hossza. A szimulált állapotváltozókból kimentem az x_1 (elő tumortérfogat) és x_2 (halott tumortérfogat) állapotváltozó összegét, ami a teljes tumortérfogat. Ezt követően kiszámítom az autoenkóder bemenete és kimenete közötti különbséget. Minden tumortérfogatókat tartalmazó idősort kivonok az előre jelzett paraméterek által meghatározott tumortérfogatókból. Ezt a költségfüggvényt terjesztem vissza a neurális hálózat súlyaira a `torch.backward()` függvény segítségével. A függvény arra szolgál, hogy kiszámítsa egy skalár (általában a költségfüggvény értéke) gradiensét az előre meghatározott tenzorokkal szemben. Miután a gradiensek kiszámításra kerültek, a NAdam optimalizáló felhasználja a gradienseket a hálózat súlyainak frissítésére, azzal a céllal, hogy minimalizálja a veszteséget. Ez a folyamat a `torch.step()` függvény meghívásával történik.

4. fejezet

Eredmények

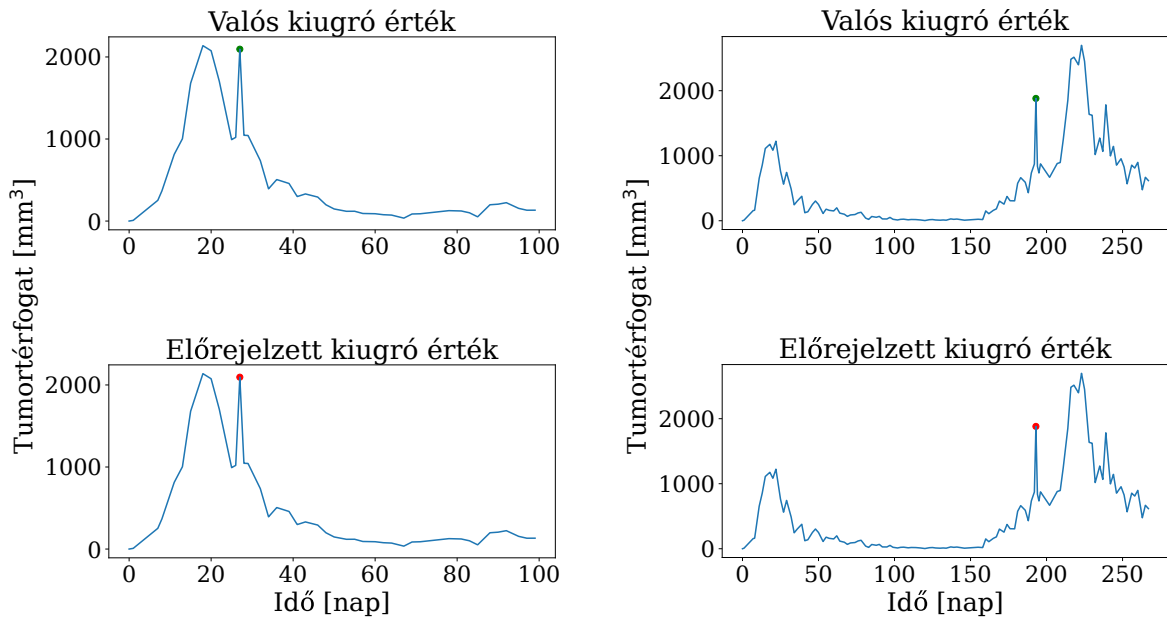
4.1. Kiugró értékek detektálásának a kiértékelése

Mivel nem állt rendelkezésre az algoritmusok tesztelésére elegendő kiugró értéket tartalmazó idősor, így mesterségesen generáltam valós idősorokba anomáliákat. Az anomáliák generálása során törekedtem arra, hogy az általunk megfigyelt anomáliás jelenséghez hasonlítson a generált kiugró érték. Egyenletes eloszlásból generáltam a kiugró érték napját a mérés első és az utolsó napja között. A pszeudo-randomszám generálásához a Python nyelv *numpy* könyvtárának a *random.Generator.uniform* metódusát alkalmaztam. A kiugró tumortérfogat létrehozásához az eredeti értéket szoroztam kettővel. Ha az eredeti érték 500 mm^3 alatt volt, akkor minden esetben az adott idősor maximum térfogatát adtam meg. Mind a két alkalmazott algoritmust a létrehozott – kiugró értékeket tartalmazó idősorokból álló – adatbázison teszteltem.

Differencián alapuló algoritmus tesztelése

A 4.1. ábra bal és jobb oldalán látható két eltérő egér példája a kiugró értékek meghatározására. A felső ábrákon a valós kiugró értékek láthatók zöld ponttal jelölve, míg az előrejelzések az alsó ábrákon láthatók piros ponttal jelölve. Megfigyelhető, hogy az algoritmus megfelelően detektálta az anomáliát mind a két esetben.

Érdemes megjegyezni az algoritmus egyik fő hiányosságát azonban, hiszen a mérések első és utolsó két adatpontjait az idősoron nem tudja vizsgálni, mivel legalább két érték szükséges a másodrendű differenciák számításához. Ez kifejezetten hátrányos számunkra, mivel gyakran az utolsó tumortérfogat értékek számunkra a legmegtévesztőbbek, hiszen rendszerint akkor kezd el a tumortérfogat exponenciálisan növekedni, így az azzal járó hirtelen meredek-ség növekedést kiugrónak tekintheti a legtöbb algoritmus.

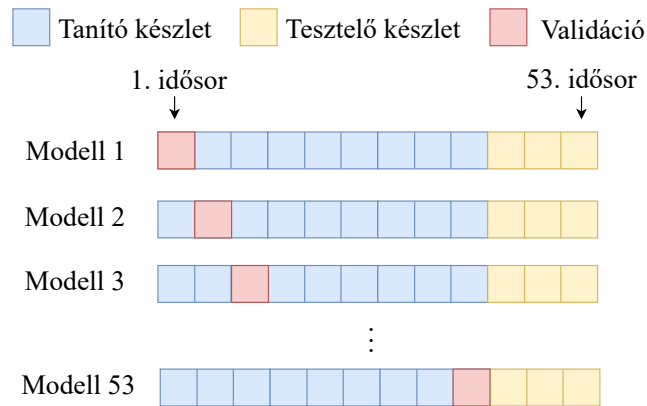


4.1. ábra. Két példa a kiugró értékek meghatározására két egér esetében. Az ábrákon szereplő idősorokon a kiugró értékek meghatározása a differenciák vizsgálatán alapuló algoritmussal történt.

Autoenkóder tesztelése

A kiugró értékek detektálására alkalmas autoenkóder betanításhoz felhasználtam a rendelkezésre álló idősorokat, viszont minden idősoron tesztelni szerettem volna az algoritmust. Ennek megoldására a tesztelés során keresztvalidációt alkalmaztam. A keresztvalidáció azt jelenti, hogy az adathalmazt felosztjuk N egyenlő részre, amelyből $N - 1$ részt használunk a modell tanítására, a fennmaradó részt pedig a validálásra. Ezt megismételjük N alkalommal, más-más felosztásban tanítva/tesztelve a modellt. Az N darab keresztvalidáció átlagos eredménye tekinthető a modell minőségének. Az irodalomban a keresztvalidációt a modell tanítása során alkalmazzák általában, mindazonáltal a már betanított modell predikciójának tesztelésére is alkalmas kis tanító adatkészlet esetén. A 4.2. ábrán látható a tanítás, tesztelés és validáció felosztásának ábrája.

Az összes 53 darab idősor mindegyikéhez külön betanított modellt hoztam létre az algoritmus validációjára. Tehát összesen 52 egeret használtam fel minden esetben tanításra és tesztelésre (0,75:0,25 arányban), majd minden egér tumortérfogat idősorát külön teszteltem. Tehát az 52 egeren betanított hálózatot az 53. egér esetében kiugró érték detekciójára alkalmaztam. Ezt a folyamatot rotáltam amíg minden egér esetében rendelkezésünkre nem állt a predikció.



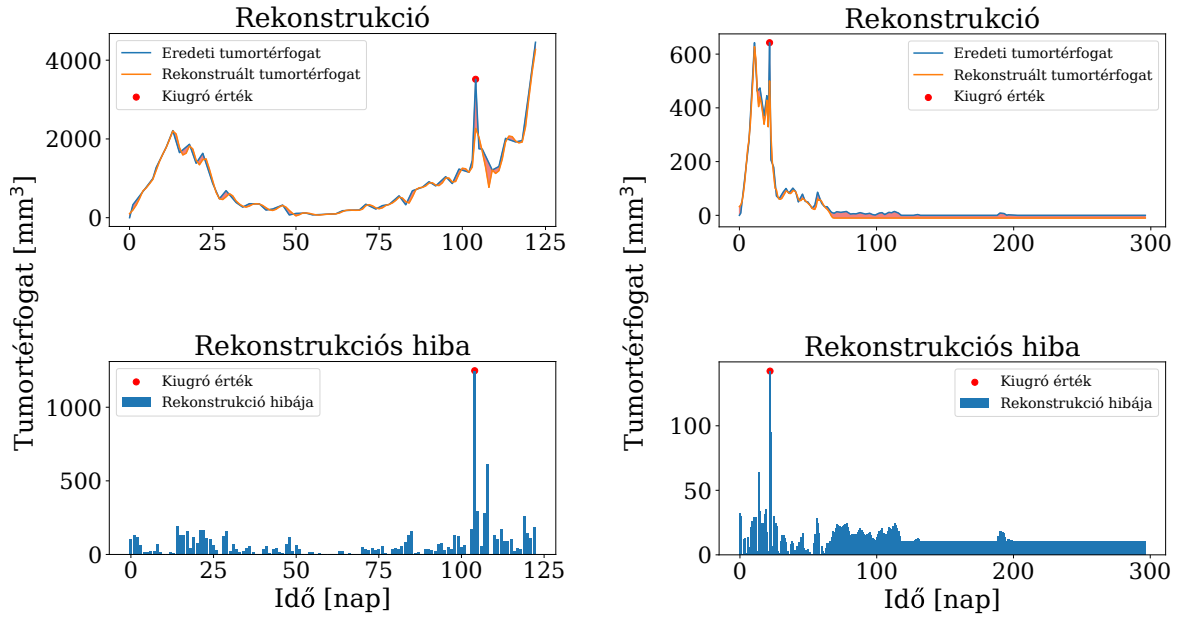
4.2. ábra. A kiugró érték detektálásának validációja az autoenkóder esetében a kis tanítókészlet miatt keresztvalidációval történt.

Az anomáliákat tartalmazó adatokon való tesztelés során a hálózat rekonstruálja a bemenet minden értékét valamekkora ϵ_i rekonstrukciós hibával. A 4.3. ábrán látható két anomáliát tartalmazó idősor rekonstrukciója, míg az alsó ábrákon a hozzátartozó rekonstrukciós hibák abszolút értéke oszlopdiagramokon ábrázolva. A kiugró érték detekciója a 3.2. folyamatábrán bemutatott lépések szerint történt. A rekonstrukciós hibák abszolút értékét vettem, majd meghatároztam a kapott értékeknek az interkvartilis terjedelmét ($\epsilon_i > Q_3 + 3IQR$), végül pedig megjelöltem a kiugró értékeket.

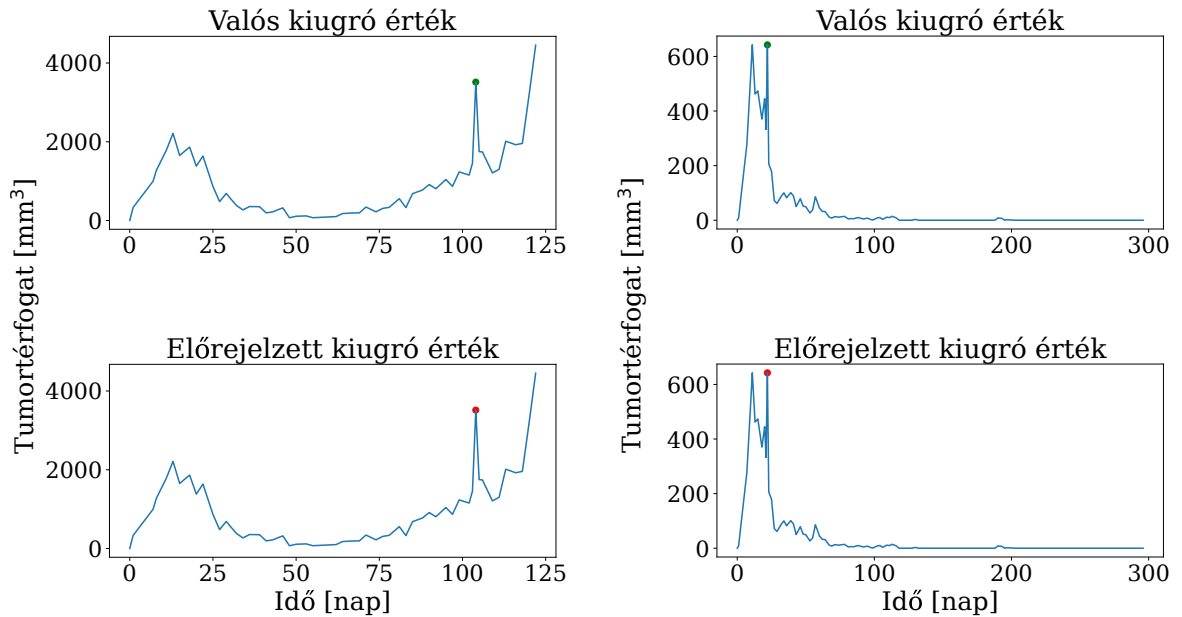
A 4.4. ábrán láthatóak az előző 4.3. ábra rekonstrukciós hibájából meghatározott kiugró értékek. Az eredeti kiugró értékek a felső ábrákon láthatók zöld ponttal, míg az előrejelzett értékek piros ponttal az alsó ábrákon. Megállapítható, hogy az algoritmus helyesen detektálta az anomáliákat.

A két algoritmus összehasonlítása

A tesztelés során minden lehetséges naphoz, amihez tumortérfogat érték tartozik hozzárendeltem egy *igaz* vagy *hamis* értéket annak függvényében, hogy azon a napon normál vagy kiugró tumortérfogat van. Ezt hasonlítottam össze a predikcióhoz is analóg létrehozott *igaz* és *hamis* értékeket tartalmazó vektorral. A 4.5. ábrán foglaltam össze a kapott eredményeket mind a differenciák vizsgálatán alapuló algoritmus, mind pedig az autoenkóder-alapú algoritmus esetében. Az ábrán egy igazságmátrix (confusion matrix) szerepel, mely általában klasszifikációs algoritmusok teljesítményének kiértékelését teszi lehetővé. Esetünkben az igazságmátrix egy 2×2 -es tábla, mely információt szolgáltat arról, hogy a vizsgált esetek mekkora részében sikerült helyesen megbecsülni a kiugró és a nem kiugró értékeket. Az igazságmátrixok tartalmazzák mind a két algoritmus esetében mind az 53 idősor minden napjának



4.3. ábra. Két egér kiugró értékének a detektálása autoenkóderen alapuló algoritmussal. A felső ábrákon késsel az eredeti tumortérfogatok tartalmozó idősorok láthatók, míg narancssárgával a rekonstruált tumortérfogat (az autoenkóder előrejelzése). Az alsó ábrákon a két görbe közötti eltérést ábrázoltam.

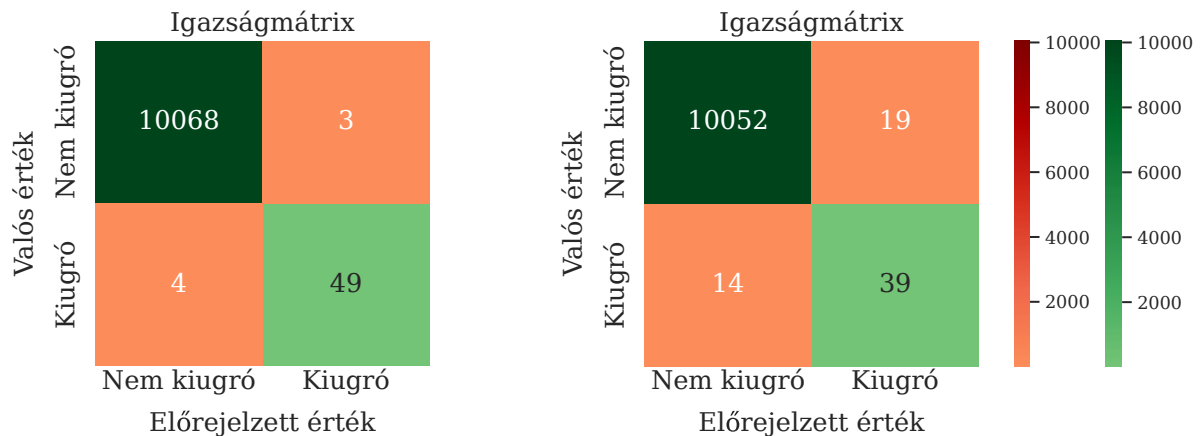


4.4. ábra. Két példa a kiugró értékek meghatározására két egér esetében. Az ábrákon szereplő idősorokon a kiugró értékek meghatározása autoenkóderen alapuló algoritmussal történt.

tumortérfoogat értékeit, mindegyik idősor esetében egy darab kiugró értékkel.

A táblázatokban zölddel vannak jelölve a helyes, míg pirossal a hibás előrejelzések. A 4.5. ábrán látható első igazságmátrix a differencián alapuló algoritmus teszt eredményeit foglalja össze. A bal felső sarokban zölddel jelöltem azokat a normál értékeket az idősorokban, melyeket az algoritmus helyesen ignorált. Ez összesen 10 068 darab tumortérfoogat értéket jelentett. A jobb felső sarokban azoknak az értékeknek a száma látható (összesen 3 eset), melyeket az algoritmus kiugró értéknek prediktált, de valójában nem voltak azok. A bal alsó sarokban összesen 4 eset látható, mely során az algoritmus normál értéknek detektálta a kiugró értékeket. Végül pedig a jobb alsó sarokban látható, hogy összesen 49 esetben a létrehozott algoritmus helyesen határozta meg a kiugró értékeket, ez az 53 esetnek a 92,45%-a.

A 4.5 ábrán látható jobb oldali igazságmátrix az autoenkóder előrejelzéseit szemlélteti a betanítást követően. Megállapítható, hogy ez az algoritmus kevesebb esetben találta el megfelelően a kiugró értékeket. Bár 53 esetből 39-et detektált az anomáliák közül – ami az esetek 73,58%-ban helyes detekciót jelent – viszont 19 érték esetében jelzett tévesen kiugró értéket. Bár az autoenkóder-alapú algoritmus a teljes intervallumon képes volt előrejelzést adni, mindazonáltal rosszabbul teljesített összességében.



4.5. ábra. A kiugró értékek meghatározásának eredményei igazságmátrixok formájában. Az ábra bal oldalán a differencián alapuló algoritmus teszt eredményei, míg a jobb oldalán az autoenkóderen alapuló algoritmus teszt eredményei láthatók.

A jobb összehasonlítás érdekében kiszámítottuk a hamis negatív és pozitív arányokat (FNR - False Negative Rate és FPR - False Positive Rate), valamint az igaz negatív és pozitív arányokat (TNR - True Negative Rate és TPR - True Positive Rate) az előre jelzett és a valós értékekből. A kiugró értékek detekciója során a hamis negatív arány (FNR) a tényleges kiugró értékek azon aránya, amelyet az algoritmus nem észlel. Azt méri, hogy az algoritmus milyen mértékben nem képes a rendellenes értékeket detektálni. A magas FNR azt jelenti,

hogyan az algoritmus jelentős számú kiugró értéket hagy ki. Az FPR a normális adatpontok azon részaránya, amelyeket tévesen azonosítanak kiugró értéként. A téves riasztások vagy téves észlelések gyakoriságát tükrözi. A magas FPR azt jelzi, hogy az algoritmus túlérzékeny, túl sok normális pontot jelöl kiugrónak. A TNR, más néven specificitás, a tényleges normális adatpontok azon aránya, amelyeket az algoritmus helyesen azonosít nem kiugrónak. A magas TNR és TPR azt jelenti, hogy a módszer nagy pontossággal képes felismerni a normális méréseket és a kiugró értékeket.

A 4.1. táblázatban összehasonlítottam a különböző metrikák eredményeit. Megállapíthatjuk, hogy a hamis pozitív arányok (FPR) mindkét esetben kicsik, ami azt jelenti, hogy mindkét algoritmus viszonylag ritkábban mutat kiugró értékeket, amikor az adott napon nem volt kiugró érték. Az autoenkóder-alapú algoritmus azonban több téves előrejelzést adott, mint a különbség-alapú módszer. A hamis negatív arány (FNR) is magasabb volt a második algoritmus esetében, ami azt jelenti, hogy több kiugró értéket is kihagyott. Ezenkívül a kiugró értékek helyes észlelési aránya (TPR) is jobb az első algoritmus esetében.

Metrika	Első algoritmus	Második algoritmus
Hamis Negatív Arány (FNR)	0.075	0.264
Hamis Pozitív Arány (FPR)	0.0003	0.0019
Igaz Negatív Arány (TNR)	0.9997	0.9981
Igaz Pozitív Arány (TPR)	0.925	0.736

4.1. táblázat. A számított metrikák összehasonlítása a két kiugró értéket detektáló algoritmus esetében.

4.2. Idősorok klaszterezésének kiértékelése

***In silico* idősorok tesztelése**

Az idősorok klaszterezése során ahhoz, hogy tesztelni tudjuk a már betanított algoritmus csoportosító és predikciós képességét, virtuális pácienseket generáltam minden kezelés esetére. Összesen 53 kezeléshez, kezelésenként 100 virtuális pácienszt hoztam létre eltérő paraméterekkel. Tehát a klaszterező algoritmus tesztelésére 5300, ismert paraméterű tumortérfigatokat tartalmazó idősor állt rendelkezésre. Az 5300 virtuális páciens mindegyikét előrejeleztem a SOM algoritmussal, mely a már létrehozott klaszterek közül kiválasztja, hogy a virtuális páciensünk melyikbe csoportba tartozik a tumordinamikája alapján leginkább. Ezt követően

a klaszterekhez hozzárendelt paraméterek intervallumát ábrázoltam és ellenőriztem, hogy a tesztelésre létrehozott virtuális pácienseink paraméterei beleesnek-e a klaszter paraméter intervallumába, illetve milyen távol esnek a klaszter paramétereinek mediánjától. Egy klaszter paraméter intervallumát az interkvartilis terjedelme alapján határoztam meg, ahol a klaszter paramétereinek a mediánjához (Q_2) hasonlítottam a virtuális páciensünk paraméterét. A (4.1) egyenlet szerint számoltam annak a hibáját, hogy az ismert paraméterű virtuális páciens paraméterei mennyire térnek el az előrejelzett klaszter paramétereitől:

$$e_i^{(a)} = \frac{|p_i^{(a)} - p_{Q_2}^{(a)}|}{p_{\max}^{(a)} - p_{\min}^{(a)}} \cdot 100, \quad (4.1)$$

ahol $p_i^{(a)}$ az adott virtuális páciens ismert a paraméter értéke, $p_{Q_2}^{(a)}$ az előrejelzett klaszter a paraméterének mediánja, továbbá $p_{\max}^{(a)}$ és $p_{\min}^{(a)}$ az a paraméterhez tartozó randomszám generálás során felhasznált felső és alsó határok. Tehát az eltérést leosztottam azzal a tartománnyal, ahonnan a randomszámokat generáltam, így megkaptam az a paraméter i . virtuális páciens hibáját ($e_i^{(a)}$). A teljes tartománnyal való leosztás célja az volt, hogy függetlenítsük a tartomány hosszától a kapott hibánkat. A (4.1) kiértékelését mind az 5300 virtuális páciensre elvégeztem, majd vettem a mediánját, átlagát és szórását a kapott eredményeknek.

In silico méréseken történő tesztelés során a farmakokinetikai paramétereket (c , k_1 , k_2) konstansnak vettem, csak az a , b , n , w , ED_{50} paramétereket változtattam. Először mind az öt paramétert egyszerre változtattam, majd aszerint, hogy mely paraméterekre adott alacsony hibát a klaszterezés, lecsökkentettük háromra ezt a számot. Végül megvizsgáltam úgy is a rendszert, hogy csak egy paramétert változtattam minden paraméter esetére. A létrejött generált paraméterekhez tumortérfogatokat tartalmazó idősorok tartoztak. Az idősorokat felhasználva tanult be a SOM algoritmus, majd ezt a már betanított algoritmust alkalmaztuk virtuális pácienseken történő tesztelésre. Az eredmények összefoglalását a három különböző számú változó paraméterek esetén a 4.2. táblázatban foglaltam össze. Látható, hogy a fixált paraméterek számának növelésével, azaz a változó paraméterek számának csökkentésével a paraméterek becslésének a hibája is csökkent. Továbbá megfigyelhető, hogy egyszerre több paraméter identifikációja esetén (a táblázat első két sora) az a paraméter adta a legjobb eredményt, mely megegyezik azzal a megállapítással, amit korábban az érzékenység vizsgálatok bizonyítottak [21]. Ezen felül látható, hogy ha egyesével változtattam a paramétereket (a táblázat alsó sora), alacsony (1% körüli) hibákat kaptam, melyből kijelenthető, hogy az algoritmus helyesen klaszterezte a tumortérfogat idősorokat.

Összesen 100 virtuális páciens hoztam létre egerenként, mely azt jelenti, hogy a 2.1. alfejeztben bemutatott tumormodellt azonos kezelés mellett 100 eltérő paraméterhalmazra oldottam meg az ODE megoldó által a kezelés minden napjára. Ezt követően a már betanított

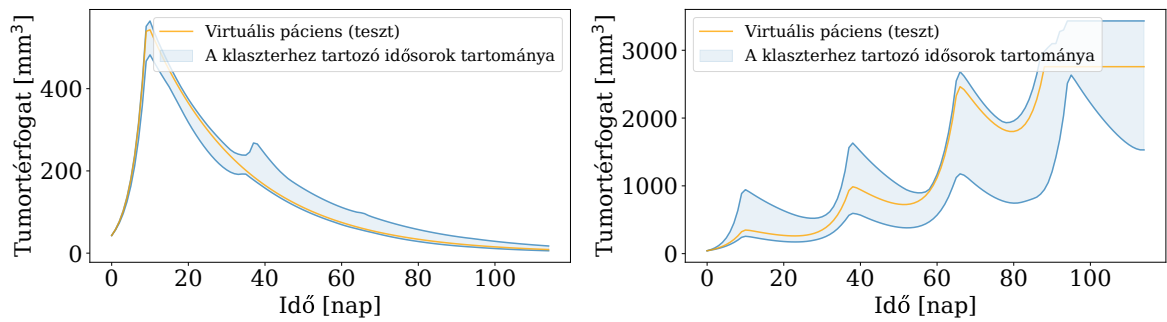
Eset	Mutató	a	b	n	w	ED ₅₀	c	k ₁	k ₂
5 változó paraméter	Medián [%]	3.47	13.54	14.39	25.97	24.32	-	-	-
	Átlag [%]	7.34	15.62	18.20	26.00	24.67	-	-	-
	SD [%]	9.44	11.93	14.41	15.07	14.90	-	-	-
3 változó paraméter	Medián [%]	2.6	12.31	16.96	-	-	-	-	-
	Átlag [%]	7.47	15.58	19.79	-	-	-	-	-
	SD [%]	10.62	12.93	13.84	-	-	-	-	-
1 változó paraméter	Medián [%]	0.982	1.20	1.04	0.81	0.90	-	-	-
	Átlag [%]	1.39	4.23	1.86	1.50	2.94	-	-	-
	SD [%]	1.38	11.22	2.84	2.32	6.93	-	-	-

4.2. táblázat. A 3×5300 teszt idősorhoz tartozó paraméterek eltéréseinek mediánja, átlaga és szórása az eredeti értékeiktől. A paramétereltéréseket különböző hálózatokon teszteltem, ahol eltérő számú paramétereket változtattam. Az első sorban egyszerre 5, míg a második sorban egyszerre 3 paramétert változtattam. Az utolsó sor esetén minden paramétert egyesével, külön-külön változtattam, míg a maradék hetet rögzítettem egy adott értéken.

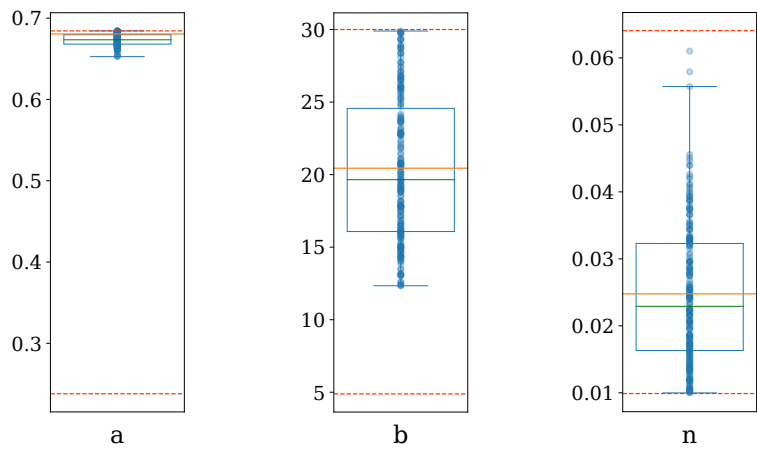
SOM algoritmussal előrejeleztem, hogy mely klaszterbe tartozik a tesztegységünk, majd az előrejelzett klaszterhez tartozó idősorok paramétereit ábrázoltam.

A 4.6. ábrán látható két azonos kezelésre betanított, hasonló tumordinamikával rendelkező idősorokat tartalmazó klaszter. Az ábrán a kék tartomány a klaszterhez tartozó idősorok minimumából és maximumából tevődik össze, míg a narancssárga görbe a tesztelésre létrehozott virtuális páciens jelöli. Megállapítható, hogy a tumordinamika alapján megfelelő csoportba helyezte a tesztet az algoritmus.

Az egy klaszterhez tartozó paraméterek interkvartilis tartományának kiértékelése során dobozdiagramokat alkalmaztam, ahol a dobozdiagram mediánjához hasonlítottam a tesztegység paramétereit. A 4.6. ábra bal oldalán látható klaszterhez tartozó paraméterek a 4.7. dobozdiagramokon láthatók. Az ábra létrehozása során három paramétert változtattam, az a, b és n-t, a többi paramétert fixáltam. A dobozdiagramokon a piros szaggatott vonalak azt a tartományt jelölik, amin belül a randomszám generálás történt egyenletes eloszlással. A kék pontok a klaszterhez tartozó paraméterek értékei, amiknek a mediánja zöld vonallal van jelölve. A tesztelésre létrehozott virtuális páciens paramétereinek értékeit narancssárga vonallal jelöltem.



4.6. ábra. Egy adott kezeléshez tartozó két klaszter. A kék intervallumok az előrejelzett klaszterhez tartozó idősorok minimumai és maximumai közötti szakaszok. A narancssárga vonal a teszt egereket jelenti.

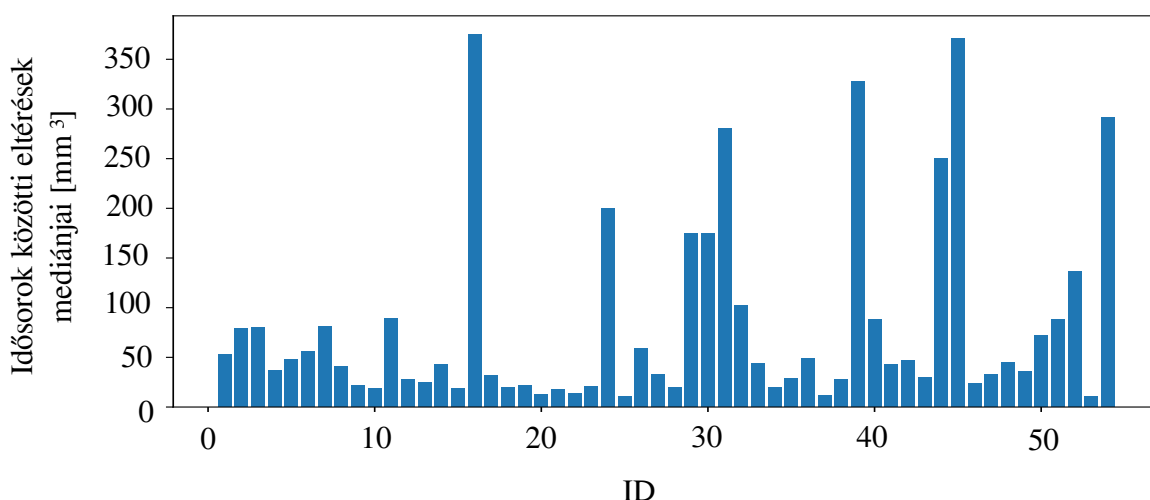


4.7. ábra. A 4.6. ábra bal oldalán látható klaszterhez tartozó paraméter intervallumok.

***In vivo* idősorok tesztelése**

A végső cél a klaszterezés során az volt, hogy ne csak *in silico*, azaz virtuális páciensek adatait csoportosítsuk tumordinamika alapján, hanem valós, *in vivo* méréseken is teszteljük. Összesen 54 egérnek a tumordinamikáját tartalmazó idősor állt rendelkezésre erre a feladatra. A különbség az *in silico* és az *in vivo* mérések között, hogy az utóbbi zajos, melynek következtében pontatlanabban klaszterezhető. Ennek kiküszöbölése érdekében először a kiugró értékeket és a zajt távolítottam el az idősorokból, majd ezt követően klasztereztem.

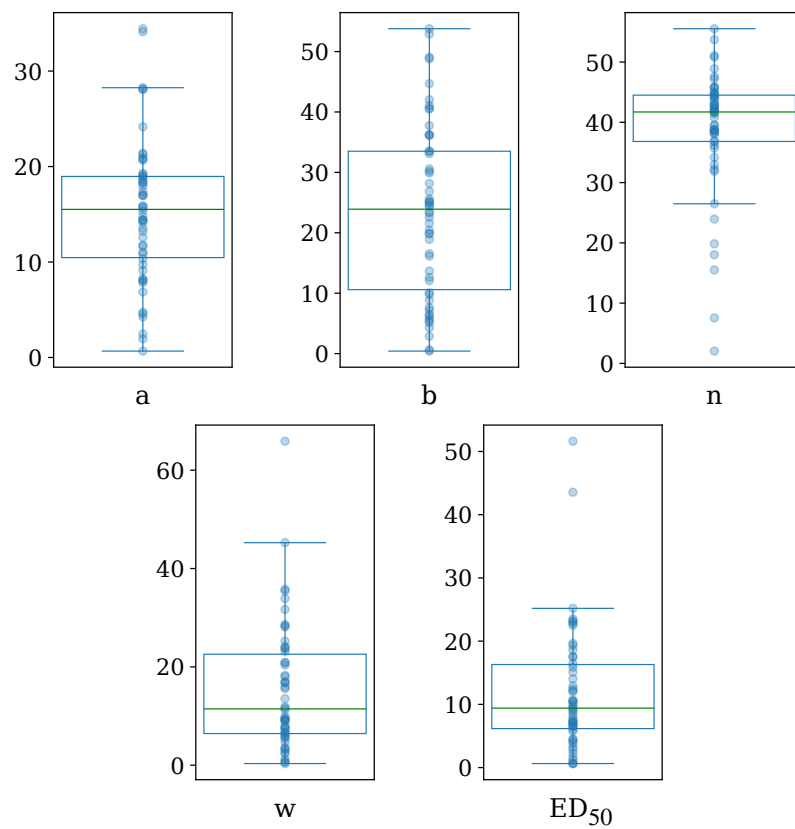
A klaszterezés pontosságának számszerűsítéséhez kiszámítottam a teszt idősor távolságát a megjósolt klaszter idősorának mediánjától. Az 54 tesztegérre vonatkozó eredmények a 4.8. ábrán láthatók. Az összes egér esetében az eltérések átlaga és mediánja 80,8, illetve 43,0.



4.8. ábra. Az ábrán az 54 rendelkezésünkre álló *in vivo* mérés hibái láthatók. A hibaszámítás egerek tumortérfogatának abszolút eltérését számítottam a becsült klaszter medián tumortérfogatától.

Korábbi mérésekből rendelkezésünkre álltak az 54 idősorhoz tartozó illesztett paraméterek értékei. A tesztelés során az eredeti idősor paramétereit tehát összehasonlítottam a klaszterhez tartozó paraméterek mediánjával. Az egyes paraméterek eredményeit a 4.9. ábra összegzi. Megfigyelhető, hogy a paraméterhibák közül az ED_{50} paraméterek mutatják a legkisebb hibát. Annak ellenére, hogy az ED_{50} a legkisebb hibával rendelkezik, azt tapasztaltuk, hogy az a és b paraméterek minden egyes klaszterben külön-külön lényegesen kisebb szórást mutattak. Emellett fontos megjegyezni, hogy a paraméterek kiértékelési folyamata során felhasznált alapigazság is egy becslési folyamat eredménye volt.

Összeségében tehát megállapítható, hogy a létrehozott klaszterező algoritmus képes volt



4.9. ábra. A paraméterek százalékos hibáinak abszolút értékei dobozdiagramokon ábrázolva. A hibák számítása során a paramétereket a korábban paraméter identifikáció útján megállapított értékekhez viszonyítottam.

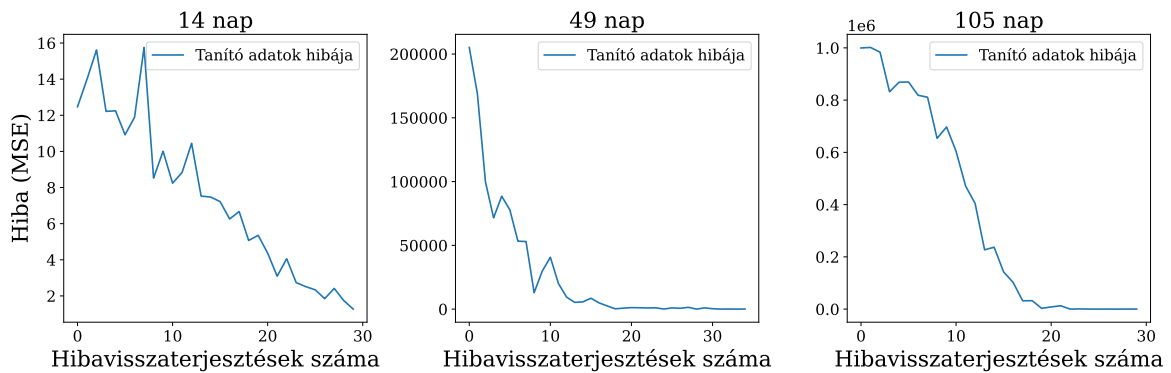
a tumortérfogatok értékeiből a mind a virtuális, mind pedig a valós mérésekből származó idősorokat csoportosítani tumordinamika alapján. A kiértékelések során összehasonlítottuk a klaszter mediánjának az idősorát a tesztesetekkel, továbbá megvizsgáltuk a hozzátartozó paramétereket is.

A jövőben a létrehozott algoritmus felhasználható úgy, hogy a terápia megkezdésének időpontja előtt a betegek kapnak egy inicializáló injekciót a kemoterápiás szerből, majd nézzük a tumordinamikájának a változását. A létrehozott klaszterező algoritmussal ez a folyamat automatizálható, a betegeknek a gyógyszerre való reakciója csoportosítható, ezáltal páciens-halmazok hozhatók létre. A páciensek halmazára terápia optimalizálható, mellyel a személyre szabott terápia költségei lecsökkenthetők.

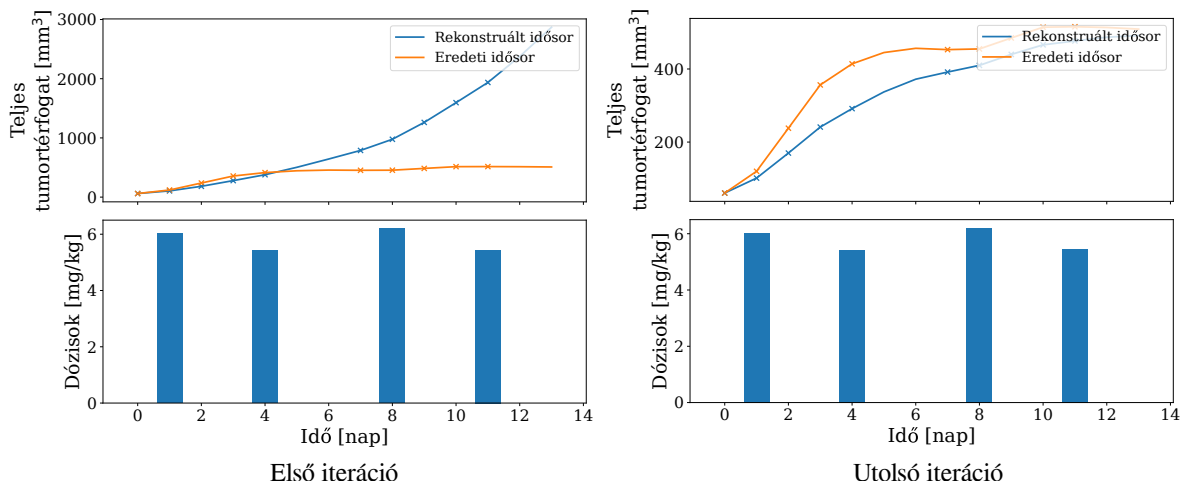
4.3. A paraméterbecslő autoenkóder kiértékelése

Az autoenkóder tanítása során három eltérő hosszúságú mérési szakaszra tanítottam be a hálózatot a kiértékelések során. A cél az volt, hogy meghatározzam, hogy rá tud-e tanulni különböző hosszúságú tumortérfogatokat tartalmazó idősorokra. A legrövidebb a 2 hetes szakasz volt, majd 7 hétre, végül pedig 15 hétre tanítottam be a hálózatot. A tanítás során a validációs adatkészlet hibájától függő megállási feltételt alkalmaztam (early stopping). Ha a hiba nem változott 3 iteráción keresztül, akkor leállt a tanítás és a legjobban teljesítő hálózat került kimentésre. A teljes adathalmazt kötegekbe rendeztem, ahogy a 3. fejezetben részleteztem. Minden köteg után hibavisszaterjesztés történt a hálózat súlyaira. Mivel az alkalmazott hálózat nem hagyományos – pusztán adatokon alapuló – tanítást alkalmaztam, hanem a tumormodell beillesztésével plusz információt szolgáltattam a tanuláshoz, így már egy iteráción belül is nagy hibacsökkenést tapasztaltam gyakran. Ebből adódóan nem az iterációk során értékeltem ki a validációs adatot, hanem minden hibavisszaterjesztést követően, ezáltal jobb képet kapva a tanulás folyamatáról.

A 4.10. ábrán látható a három intervallumra kapott ún. loss görbe. Látható, hogy mind a három esetben körülbelül 30 hibavisszaterjesztés volt szükséges ahhoz, hogy a tanító adatkészletre rátanuljon a hálózat. Továbbá megfigyelhető, hogy minél rövidebb az előrejelzés időintervalluma, annál kisebb hibáról indul a tanítás. Ez abból adódik, hogy minél hosszabb az időintervallum, annál nagyobb értékeket vehet fel a tumortérfogat az exponenciális növekedésének következtében (ahogy a 4.13. ábra is igazolja).



4.10. ábra. Az autoenkóder három tanulási görbéje a három intervallumra. Mivel a hálózat nagy mértékben tanult minden kötegen történő hibavisszaterjesztés után, így a hagyományos iterációk (epoch) helyett a hibavisszaterjesztések számát ábrázoltam a vízszintes tengelyen.



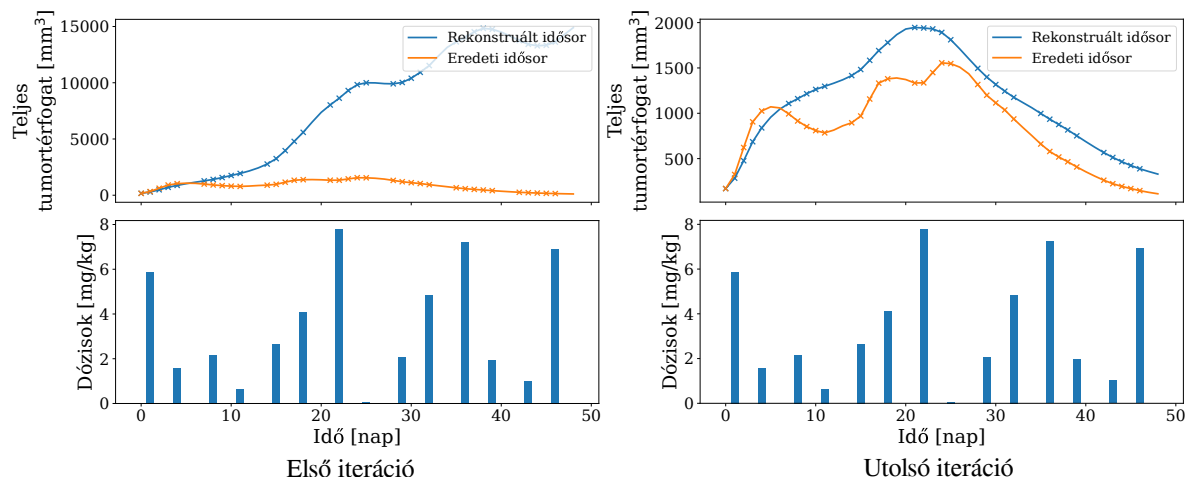
4.11. ábra. Példa a validációs adathalmazból a hálózat tanulása alatti teljesítmény változásra két hetes időintervallumon. A tanítás folyamata során kezdetben nagyobb hibával, majd ahogy tanult a hálózat, egyre kisebb hibával becsülte meg a tumortérfogatot.

Tanítás 14 napos intervallumon

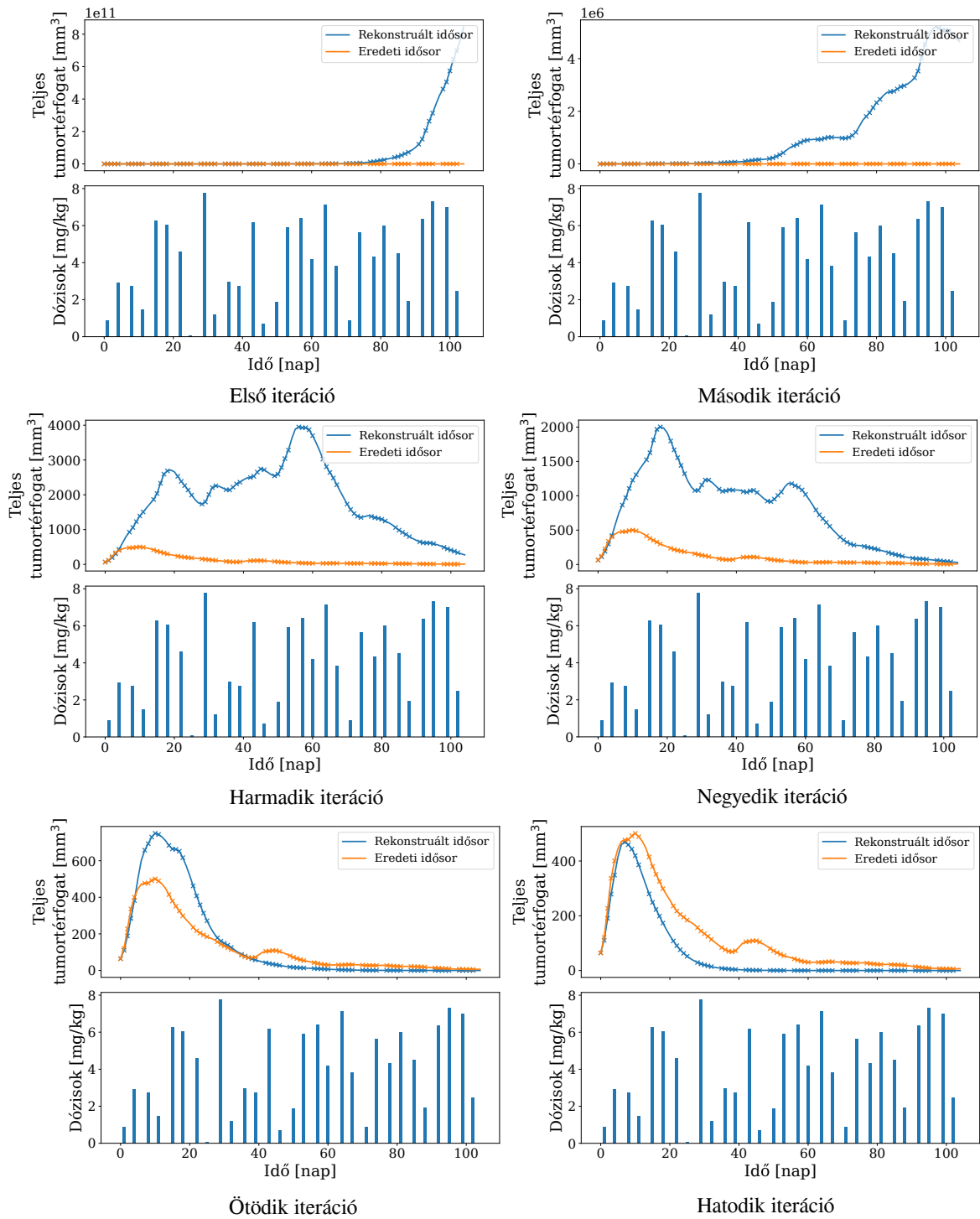
A 4.11. ábrán látható két példa arra, hogy az autoenkóder teljesítménye hogyan fejlődött az iterációk alatt. Mind a két ábrán az előrejelzett tumortérfogatokat ábrázoltam késsel, míg narancssárgával az eredeti idősort. A normál protokoll szerinti tumortérfogat mérési időpontokat (hétfőtől péntekig) kereszttel jelöltem. A random generált dózisokat oszlopdiagrammal ábrázoltam a tumortérfogat görbék alatt. A bal oldali ábrán az első iterációt követően, míg a jobb oldali ábrán már az autoenkóder betanítását követően történt a predikció. Megállapítható, hogy az előre jelzett tumortérfogatokat tartalmazó idősor jobban illeszkedik az eredetihez a betanítást követően.

Tanítás 49 napos intervallumon

A 7 hetes tanítás során szintén kimentettem a validáció során kapott rekonstrukciókat. A 4.12. ábrán látható egy példa a 49 napos előrejelzésre. Ebben az esetben kezdetben túl agresszív tumordinamikát prediktált az autoenkóder. Később azonban, ahogy a tanító adatkészleten betanult – hasonlóan a két hetes verzióhoz – a validációs adatokon is jobban teljesített, ahogy a jobb oldali görbén látszik.



4.12. ábra. Példa a validációs adathalmazból a hálózat tanulása alatti teljesítmény változásra hét hetes időintervallumon. A tanítás folyamata során kezdetben nagyobb hibával, majd ahogy tanult a hálózat egyre kisebb hibával becsülte meg a tumortérfogatokat.



4.13. ábra. Példa a validációs adathalmazból a hálózat tanulására a 105 napos intervallumon.

Tanítás 105 napos intervallumon

A 105 napos tanítás során – ahogy a 4.13. ábrán is látszik – minden iteráció utáni kimenetet lementettem, hogy vizualizálni tudjam a hálózat tanításának folyamatát. Az ábrán a bal felső sarokban az első iterációt követően látható az előrejelzett paraméterek által meghatározott tumortérfogat érték. Megfigyelhető, hogy az előre jelzett tumortérfogat értéke exponenciálisan elszáll. Mindazonáltal, az iterációk számának növekedésével a két görbe egyre közeledik egymáshoz, míg a tanítás végén egészen hasonló tumordinamikájú görbéket nem kapunk. A tapasztalatok alapján el lehet mondani, hogy az egyszerűbb, exponenciálisan lecsengő görbéket (kezelésre jól reagáló virtuális páciensek esetét) az esetek többségében jól megtanulta a hálózat. Ezzel szemben, a bonyolultabb tumordinamikát (mely a tanító adatkészlet kisebb részét képezi) az esetek csak kis százalékában tudta helyesen előre jelezni. A jövőben ebből adódóan érdemes a hálózat általánosító képességét növelni, valamint a tanító adatkészlet létrehozása során a különböző tumordinamikájú idősorokat figyelembe venni.

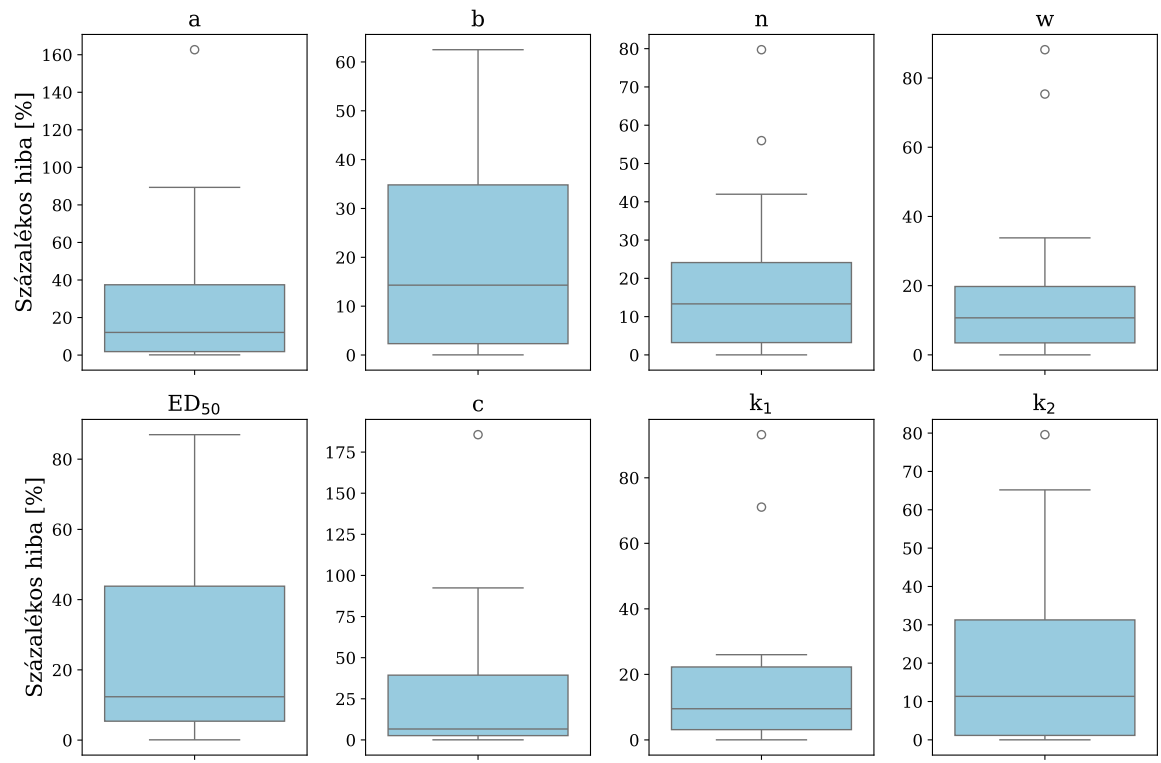
A betanítást követően a paraméterek százalékos eltéréseit is meghatároztam az eredeti értékektől, mely a nyolc paraméterre a 4.14. ábrán látható. Az eltéréseket dobozdiagrammon ábrázoltam. Az alsó kvartilis, azaz doboz alja (Q_1) a hibák alsó 25%-ának alsó határát mutatja, míg a felső kvartilis, azaz doboz teteje (Q_3) a felső 25%-ának az alsó határát jelzi. A medián a dobozt keresztbe szelő egyenes. Az ábráról leolvasható, hogy a relatív százalékos hiba mindenhol 10% körüli volt.

Mérés hossza [nap]	14	49	105
Medián	260.5	303.2	378.7
Átlag	630.6	724.8	1044.4

4.3. táblázat. Az idősorok közötti abszolút eltérés a teszt adatkészleten.

A 4.3. táblázatban összegyűjtöttem a teszt adatokon végzett kiértékelés hibáit az idősorokra. A medián és az átlag közötti eltérés azt a feltételezést igazolja, hogy egyes tumordinamikára jól rátanult a hálózat, viszont voltak kiugróan rossz eredmények, mely a hiba átlagát eltolták. Továbbá megfigyelhető az a tendencia is, hogy az idősor hosszának növekedésével az átlagos hiba nagysága is nő.

Összeségében tehát elmondható, hogy az autoenkóder a rendelkezünkre álló differenciálegyenletrendszer beépítésével képes volt megtanulni a modell paramétereit. Három különböző időintervallumra megvizsgálva a rendszert, az a konklúzió, hogy a hálózat jól teljesített azokon az adatokon, amik az adatkészlet nagy részét képezték. A futási idő az időintervallum megnöve-



4.14. ábra. A tesztadatokon a paraméterek eltérései az eredeti értékektől mind a három intervallum esetén.

lésével nem nőtt szignifikánsan, ezáltal a jövőben akár hosszabb intervallumokat is meg lehet vizsgálni.

A jövőben a modell módosítható úgy, hogy képes legyen nem csak az egyedek paramétereinek meghatározására, hanem hasonló paraméterű, adott eloszlással rendelkező egérpuláció paramétereinek a várható értékének és a szórásának a meghatározására is. Továbbá a hálózat architektúrájának fő jelentősége az, hogy nem csak *in silico*, de *in vivo* adatokon is képes tanulni. Tehát nem szükségesek hozzá az adott tumordinamikához tartozó paraméterek értékei, így egy fontos lépés lehet a jövőben a valós és a zajos mérési adatokon történő továbbtanítása a hálózatnak.

5. fejezet

Konklúzió

A betegségek matematikai modellezése révén azok lefolyása és reakciója különböző kezelésekre leírható és előre jelezhető. Azonban ezek a modellek önmagukban nem elegendőek, a modell paramétereinek pontos meghatározása nélkülözhetetlen. A diplomamunkám egy kutatás köztes lépését adja, azáltal, hogy a tumordinamikai modell paramétereinek becslésére alkalmas algoritmusokat hoztam létre. A paraméteridentifikációra alkalmazott iteratív módszerek jelentős időt igényelhetnek az optimális megoldás eléréséhez, továbbá nem garantált, hogy a globális optimumhoz konvergálnak, különösen, ha a kiindulási paraméter értékek távol esnek azok optimális értékeitől. Mindazonáltal, a rosszul meghatározott értékek által a betegség rossz csoportosítása a kemoterápiás gyógyszer mennyiségének rossz meghatározását eredményezheti. A paraméterek gyors és pontos ismerete által terápia generálható különböző algoritmusokkal egyénre, vagy akár hasonló tumordinamikával rendelkező populációra.

A munkám három szakaszra osztható. Az első szakaszában a tumortérfogatok kiugró értékeinek detektálásával és a zaj szűrésével foglalkoztam. Mivel a paraméterek pontos azonosítása a modell kimenetén alapul, amely a teljes tumortérfogat, fontos, hogy ne tartalmazzanak hibás értékeket a felhasznált idősorok. A tumortérfogat tolmérővel mért változó, és nem tartalmazhat jelentős hibákat vagy kiugró értékeket, mivel ezek befolyásolhatják a paraméterek meghatározásának az eredményét. Ebben a munkában két algoritmust hoztam létre a kiugró értékek azonosítására. A két algoritmust kiértékelünk és összehasonlítottunk ugyanazzal az idősorral és ugyanazokkal a kiugró értékekkel. Az eredmény szerint a különbség alapú algoritmus jobb teljesítményt mutatott, azonban az autoenkóder-alapú algoritmus általánosabb megoldást nyújtott a problémára.

A munkám második szakaszában tumordinamika alapján csoportosítottam az *in silico* és *in vivo* tumortérfogatokat tartalmazó idősorokat. A klaszterezéshez önszerveződő-térképet alkalmaztam. A kiértékelés során megvizsgáltuk a klaszterek medián idősorai és a teszt idősor közötti eltéréseket. Mivel a klaszterekben található idősorokhoz paraméterek tartoztak,

így a hozzájuk rendelt paraméterek és a teszt esethez tartozó paraméterek közötti eltéréseket is kiértékeltem. A kiértékelés során *in silico* esetben, a rögzített paraméterek számának növelésével a pontosság növekedett. *In vivo* esetben a klaszterezést megelőzően Wavelet-transzformációval csökkentettük a zajt az adatokon. A jövőben a cél az algoritmus felhasználásával a hasonló tumordinamikájú betegek betegcsoportokba történő klaszterezése és populáció-alapú terápia alkalmazása.

A munkám harmadik szakaszában egy olyan speciális autoenkóder architektúrát fejlesztettem ki, mely a dózisokból és a tumortérfogatokat tartalmazó idősorokból képes a paraméterek meghatározására és megtanulására. Ez az architektúra alkalmas lehet nem csak *in silico* (szimulált adatokon alapuló), de akár *in vivo* (valós méréseken alapuló) adatokból történő betanításra, mivel nem igényli a paraméterek előzetes ismeretét, mert architektúrájából adódóan felügyelet nélküli tanulással tanul. Ezáltal a valósághoz közelebbi eredményt kaphatunk, ha mind szimulált, mind pedig valós adatkészleten be tudjuk tanítani a hálózatot. Az eredmények kiértékelése során három különböző idősor hossza tanítottam be az autoenkódert, majd pedig értékeltem ki. Először két hétre vizsgáltam meg az előrejelzés pontosságát, majd pedig hét hétre, végül pedig 15 hétre. Az eredmények azt mutatták, hogy a hálózat képes volt megtalálni azokat a paramétereket, amelyek leginkább visszaadják a tumortérfogat idősorokat. A hálózat teljesítőképességét előre elkülönített teszt adatokon határoztam meg. Megvizsgáltam a paraméterek eltérését az eredeti értékhez képest, valamint a rekonstruált idősorok eltérését a bemeneti idősorhoz viszonyítva. Összességében elmondható, hogy a létrehozott architektúra alkalmas a differenciálegyenletek – esetünkben tumormodell – paramétereinek meghatározására és megtanulására.

A kutatás végső célja egy olyan gyógyszerhordozó rendszer kifejlesztése, mellyel a kemoterápiás gyógyszerek adagolásával a tumor mérete szabályozható, ezáltal a beteg életminősége és élethossza megnövelhető. Egy ilyen eszköz alapja egy olyan algoritmus, melynek nélkülözhetetlen részét képezi egy robusztus, paraméterek identifikálására alkalmas módszer. Munkám során a paraméterek gyorsabb meghatározásához fejlesztettem gépi tanulási algoritmust.

A "Tumormodell paramétereinek meghatározása MI alkalmazásával" projekt nevében köszönetet mondunk a HUN-REN Cloud (lásd: Héder et al. 2022; <https://science-cloud.hu/>) használatáért, ami hozzájárult a publikált eredmények eléréséhez [56]. Kutatásom az Innovációs és Technológiai Minisztérium ÚNKP-23-2 kódszámú Új Nemzeti Kiválóság Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs alapról finanszírozott szakmai támogatásával készült.

Irodalomjegyzék

- [1] Freddie Bray, Mathieu Laversanne, Elisabete Weiderpass, and Isabelle Soerjomataram. The ever-increasing importance of cancer as a leading cause of premature death worldwide. *Cancer*, 127(16):3029–3030, 2021.
- [2] Rebecca L Siegel, Kimberly D Miller, Nikita Sandeep Wagle, Ahmedin Jemal, et al. Cancer statistics, 2023. *Ca Cancer J Clin*, 73(1):17–48, 2023.
- [3] Camilla Mattiuzzi and Giuseppe Lippi. Current cancer epidemiology. *Journal of Epidemiology and Global Health*, 9(4):217, 2019.
- [4] Zaigham Abbas and Sakina Rehman. An overview of cancer treatment modalities. In *Neoplasms*. InTech, sep 2018.
- [5] Sudipta Senapati, Arun Kumar Mahanta, Sunil Kumar, and Pralay Maiti. Controlled drug delivery vehicles for cancer treatment and their performance. *Signal Transduction and Targeted Therapy*, 3(1), March 2018.
- [6] Dejene Tolossa Debela, Seke GY Muzazu, Kidist Digamo Heraro, Maureen Tayamika Ndalama, Betelhiem Woldemedhin Mesele, Dagimawi Chilot Haile, Sophia Khalayi Kitui, and Tsegahun Manyazewal. New approaches and procedures for cancer treatment: Current perspectives. *SAGE Open Medicine*, 9:205031212110343, January 2021.
- [7] Flavia Laffleur and Valerie Keckeis. Advances in drug delivery systems: Work in progress still needed? *International journal of pharmaceutics*, 590:119912, 2020.
- [8] Xiaoqiang Sun and Bin Hu. Mathematical modeling and computational prediction of cancer drug resistance. *Briefings in Bioinformatics*, 19(6):1382–1399, June 2017.
- [9] Angela M. Jarrett, Ernesto A.B.F. Lima, David A. Hormuth, Matthew T. McKenna, Xinzeng Feng, David A. Ekrut, Anna Claudia M. Resende, Amy Brock, and Thomas E. Yankeeelov. Mathematical models of tumor cell proliferation: A review of the literature. *Expert Review of Anticancer Therapy*, 18(12):1271–1286, October 2018.

- [10] Hsiu-Chuan Wei. Mathematical modeling of tumor growth and treatment: Triple negative breast cancer. *Mathematics and Computers in Simulation*, 204:645–659, 2023.
- [11] Dániel András Drexler, Tamás Ferenci, Anna Lovrics, and Levente Kovács. Tumor Dynamics Modeling based on Formal Reaction Kinetics. *Acta Polytechnica Hungarica*, 16:31–44, 2019.
- [12] Dániel András Drexler, Tamás Ferenci, András Füredi, Gergely Szakács, and Levente Kovács. Experimental data-driven tumor modeling for chemotherapy. In *Proceedings of the 21st IFAC World Congress*, pages 16466–16471, 2020.
- [13] Dániel András Drexler, Johanna Sápi, and Levente Kovács. Modeling of tumor growth incorporating the effects of necrosis and the effect of bevacizumab. *Complexity*, 2017:1–10, 2017.
- [14] Johanna Sápi, Levente Kovács, Dániel András Drexler, Pál Kocsis, Dávid Gajári, and Zoltán Sápi. Tumor volume estimation and quasi-continuous administration for most effective bevacizumab therapy. *PloS one*, 10(11):e0142190, 2015.
- [15] Melánia Puskás, Borbála Gergics, Balázs Gombos, András Füredi, Gergely Szakács, Levente Kovács, and Dániel András Drexler. Noise modeling of tumor size measurements from animal experiments for virtual patient generation. In *2023 IEEE 27th International Conference on Intelligent Engineering Systems (INES)*. IEEE, July 2023.
- [16] Xin-She Yang. *Mathematical modeling with multidisciplinary applications*. John Wiley & Sons, Nashville, TN, December 2012.
- [17] Lennart Ljung. *System Identification*, pages 163–173. Birkhäuser Boston, Boston, MA, 1998.
- [18] Adriaan van den Bos. *Parameter estimation for scientists and engineers*. Wiley-Blackwell, Hoboken, NJ, June 2007.
- [19] Levente Kovács, Tamás Ferenci, Balázs Gombos, András Füredi, Imre Rudas, Gergely Szakács, and Dániel András Drexler. Positive impulsive control of tumor therapy—a cyber-medical approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–12, 2023.
- [20] Franz-Georg Wieland, Adrian L. Hauber, Marcus Rosenblatt, Christian Tönsing, and Jens Timmer. On structural and practical identifiability. *Current Opinion in Systems Biology*, 25:60–69, 2021.

- [21] Mate Siket, Gyorgy Eigner, and Levente Kovacs. Sensitivity and identifiability analysis of a third-order tumor growth model. In *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*. IEEE, June 2020.
- [22] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [23] Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, 2022.
- [24] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, May 2021.
- [25] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics-informed neural networks: Where we are and what’s next. *Journal of Scientific Computing*, 92(3), July 2022.
- [26] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- [27] William Bradley and Fani Boukouvala. Two-stage approach to parameter estimation of differential equations using neural odes. *Industrial & Engineering Chemistry Research*, 60(45):16330–16344, 2021.
- [28] Subiksha Selvarajan, Aike Aline Tappe, Caroline Heiduk, Stephan Scholl, and René Schenkendorf. Parameter identification concept for process models combining systems theory and deep learning. In *ECP 2022*. MDPI, June 2022.
- [29] Qiang Yin, Juntong Cai, Xue Gong, and Qian Ding. Local parameter identification with neural ordinary differential equations. *Applied Mathematics and Mechanics*, 43(12):1887–1900, December 2022.
- [30] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

- [31] Marc M Van Hulle. Self-organizing maps. *Handbook of natural computing*, 1:585–622, 2012.
- [32] Simon O Haykin. *Neural Networks and Learning Machines*. Pearson, Upper Saddle River, NJ, 3 edition, November 2008.
- [33] Bernhard Mehlig. *Machine learning with neural networks*. Cambridge University Press, Cambridge, England, October 2021.
- [34] Charu C Aggarwal. *Neural networks and deep learning*. Springer International Publishing, Cham, Switzerland, 1 edition, September 2018.
- [35] Dor Bank, Noam Koenigstein, and Raja Giryes. *Autoencoders*, pages 353–374. Springer International Publishing, Cham, 2023.
- [36] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 71:599–607, 1986.
- [37] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)*, 54(3):1–33, 2021.
- [38] Tung Kieu, Bin Yang, Chenjuan Guo, and Christian S Jensen. Outlier detection for time series with recurrent autoencoder ensembles. In *IJCAI*, pages 2725–2732, 2019.
- [39] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797, 2022.
- [40] Samir Avdakovic, Amir Nuhanovic, and Mirza Kusljugic. Wavelet theory and applications for estimation of active power unbalance in power system. In *Advances in Wavelet Theory and Their Applications in Engineering, Physics and Technology*. InTech, April 2012.
- [41] H-Y Lin, S-Y Liang, Y-L Ho, Y-H Lin, and H-P Ma. Discrete-wavelet-transform-based noise removal and feature extraction for ecg signals. *Irbm*, 35(6):351–361, 2014.
- [42] Gregory Lee, Ralf Gommers, Kai Wohlfahrt, Filip Wasilewski, Aaron O’Leary, Holger, Alexandre Sauv e, Jarrod Millman, Ankit Agrawal, Christian Clauss, Daniel M. Pelt,

- Helder Oliveira, Frank Yu, Matthew Brett, Michel Pelletier, SylvainLan, Daniele Tricoli, Saket Choudhary, Ahmet Can Solak, asnt, Arfon Smith, 0-tree, Corey Goldberg, Daniel Goertzen, Dawid Laszuk, ElConno, Evans Doe Ocansey, Jacopo Antonello, Jakub Mandula, and jakirkham. *Pywavelets/pywt*: v1.5.0, 2023.
- [43] Giuseppe Vettigli. *Minisom*: minimalistic and numpy-based implementation of the self organizing map, 2018.
- [44] Tak-chung Fu, Fu-lai Chung, Vincent Ng, and Robert Luk. Pattern discovery from stock time series using self-organizing maps. In *Workshop notes of KDD2001 workshop on temporal data mining*, volume 1. Citeseer, 2001.
- [45] Xiaozhe Wang, Kate Smith, and Rob Hyndman. Characteristic-based clustering for time series data. *Data mining and knowledge Discovery*, 13:335–364, 2006.
- [46] Ming Ge, Min-Sen Chiu, and Qing-Guo Wang. An extended self-organizing map for nonlinear system identification. *Industrial & engineering chemistry research*, 39(10):3778–3788, 2000.
- [47] G.A. Barreto and A.F.R. Araujo. Identification and control of dynamical systems using the self-organizing map. *IEEE Transactions on Neural Networks*, 15(5):1244–1259, September 2004.
- [48] Daria Kurz, Carlos Salort Sánchez, and Cristian Axenie. Data-driven discovery of mathematical and physical relations in oncology data using human-understandable machine learning. *Frontiers in Artificial Intelligence*, 4, November 2021.
- [49] Anil K Jain and Richard C Dubes. *Algorithms Clustering Data*. Prentice Hall advanced reference series. Prentice Hall, Old Tappan, NJ, January 1988.
- [50] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3(1):1–27, 1974.
- [51] Marten Lienen and Stephan Günnemann. torchode: A parallel ode solver for pytorch. *arXiv preprint arXiv:2210.12375*, 2022.
- [52] Timothy Dozat. Incorporating Nesterov Momentum into Adam. In *Proceedings of the 4th International Conference on Learning Representations*, pages 1–4.
- [53] J C Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, Nashville, TN, 3 edition, August 2016.

- [54] John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.
- [55] Ricky T. Q. Chen. torchdiffEq, June 2021.
- [56] M. Héder, E. Rigó, D. Medgyesi, R. Lovas, Sz. Tenczer, F. Török, A. Farkas, M. Emődi, J. Kadlecik, Gy. Mező, Á. Pintér, and P. Kacsuk. The past, present and future of the ELKH cloud. *Információs Társadalom*, 22(2):128, aug 2022.

Ábrák jegyzéke

3.1.	Az tumortérfogatok közötti differenciák vizsgálatán alapuló kiugró érték meghatározás folyamatábrája.	28
3.2.	A kiugró érték meghatározására alkalmazott, autoenkóderen alapuló algoritmus folyamatábrája. A betanított autoenkóder előrejelzéséből és az eredeti idősorok különbségéből rekonstrukciós hiba számítható. A rekonstrukciós hiba nagyságát alkalmaztam a kiugró értékek indikátoraként.	30
3.3.	A Wavelet-transzformáció alkalmazásának eredményei idősoros adatokon. A piros „x” jelek a megfigyelt tumortérfogat-méréseket jelölik, beleértve a zajt is, míg a kék vonal a Wavelet-transzformáció alkalmazása után közelített értékeket mutatja. A koeficienszek számának növekedésével nő a nullára állított magas frekvenciájú komponensek száma.	32
3.4.	A tumortérfogat értékeket tartalmazó idősorok klaszterezésére létrehozott algoritmus folyamatábrája. Az algoritmus a folyamat elején kiválaszt egy kezelést, virtuális tumortérfogat méréseket generál hozzá, majd csoportosítja azokat tumordinamikák alapján.	34
3.5.	A paraméterbecslő autoenkóder felépítése. Az autoenkóder a bemeneti tumortérfogatok és a becsült tumortérfogatok közötti különbség minimalizálásával határozza meg a bemeneti tumortérfogatokhoz tartozó paraméterek értékeit.	38
3.6.	A tanító adatok generálása során alkalmazott mérési elrendezés. Az alkalmazott protokoll alapján hetente kétszer, kedden és pénteken történik injekciózás, míg hétköznaponta történik tumortérfogat mérés.	39
3.7.	A neurális hálózat előtanítása során kapott hibák a tanító adatkészletre és a validációs adatkészletre. A hibák a becsült paraméterek és az eredeti paraméterek normalizált értékei közötti átlagos négyzetes eltérést mutatják.	41
3.8.	A neurális hálózat előtanítását követően kapott paraméterek (a , b , n , w , ED_{50} , c , k_1 , k_2) értékeinek gyakoriságai a tesztelésre elkülönített adatsoron.	42

3.9.	Az autoenkóder első szakaszának a részletezett ábrája. Az enkóder rész a bemenetből és a neurális hálózatból épül fel. A hálózat bemenetei a tumortérfogatokat és a dózisokat tartalmazó idősorok.	43
3.10.	Az autoenkóder dekóder része, mely a paraméterekből, kezdeti értékekből és dózisokból előállítja a szimulált tumortérfogatot. A szimulált tumortérfogat és a valós tumortérfogatok közötti átlagos négyzetes eltérés értéke alapján tanítjuk az enkóder neurális hálózatát.	45
4.1.	Két példa a kiugró értékek meghatározására két egér esetében. Az ábrákon szereplő idősorokon a kiugró értékek meghatározása a differenciák vizsgálatán alapuló algoritmussal történt.	48
4.2.	A kiugró érték detektálásának validációja az autoenkóder esetében a kis tanítókészlet miatt keresztvalidációval történt.	49
4.3.	Két egér kiugró értékének a detektálása autoenkóderen alapuló algoritmussal. A felső ábrákon késsel az eredeti tumortérfogatokat tartalmazó idősorok láthatók, míg narancssárgával a rekonstruált tumortérfogat (az autoenkóder előrejelzése). Az alsó ábrákon a két görbe közötti eltérést ábrázoltam.	50
4.4.	Két példa a kiugró értékek meghatározására két egér esetében. Az ábrákon szereplő idősorokon a kiugró értékek meghatározása autoenkóderen alapuló algoritmussal történt.	50
4.5.	A kiugró értékek meghatározásának eredményei igazságmátrixok formájában. Az ábra bal oldalán a differencián alapuló algoritmus teszteredményei, míg a jobb oldalán az autoenkóderen alapuló algoritmus teszteredményei láthatók.	51
4.6.	Egy adott kezeléshez tartozó két klaszter. A kék intervallumok az előrejelzett klaszterhez tartozó idősorok minimumai és maximumai közötti szakaszok. A narancssárga vonal a teszt egereket jelenti.	55
4.7.	A 4.6. ábra bal oldalán látható klaszterhez tartozó paraméter intervallumok.	55
4.8.	Az ábrán az 54 rendelkezésünkre álló <i>in vivo</i> mérés hibái láthatók. A hibaszámítás egerek tumortérfogatának abszolút eltérését számítottam a becsült klaszter medián tumortérfogatától.	56
4.9.	A paraméterek százalékos hibáinak abszolút értékei dobozdiagramokon ábrázolva. A hibák számítása során a paramétereket a korábban paraméter identifikáció útján megállapított értékekhez viszonyítottam.	57

4.10. Az autoenkóder három tanulási görbéje a három intervallumra. Mivel a hálózat nagy mértékben tanult minden kötegen történő hibavisszaterjesztés után, így a hagyományos iterációs (epoch) helyett a hibavisszaterjesztések számát ábrázoltam a vízszintes tengelyen.	59
4.11. Példa a validációs adathalmazból a hálózat tanulása alatti teljesítmény változásra két hetes időintervallumon. A tanítás folyamata során kezdetben nagyobb hibával, majd ahogy tanult a hálózat, egyre kisebb hibával becsülte meg a tumortérfogatokat.	59
4.12. Példa a validációs adathalmazból a hálózat tanulása alatti teljesítmény változásra hét hetes időintervallumon. A tanítás folyamata során kezdetben nagyobb hibával, majd ahogy tanult a hálózat egyre kisebb hibával becsülte meg a tumortérfogatokat.	60
4.13. Példa a validációs adathalmazból a hálózat tanulására a 105 napos intervallumon.	61
4.14. A tesztadatokon a paraméterek eltérései az eredeti értékektől mind a három intervallum esetén.	63