

Szenzor kommunikációs folyamatok állapotadatainak neurális hálózat alapú elemzése

Neural Network Based Analysis of Status Information of Sensor Communication Processes

dr. habil. GÁL Zoltán, egyetemi docens

Debreceni Egyetem Informatikai Kar,
H-4028 Debrecen, Kassai út. 26., Gal.Zoltan@inf.unideb.hu

Abstract

The Internet of Things requires the communication mechanism to be optimal not only from the data transfer but from the energy consumption point of view, too. One of most analysed types of sensor network is Low Energy Adaptive Clustering Hierarchy (LEACH) system depending on the population density, algorithm of cluster head election, heterogeneity of the energy and physical position of the nodes, velocity of the sink node, data aggregation rate and size of data frame. Complexity of the system has been analysed based on status datasets of several hundred simulation cases. The serviceability of LEACH network and dependency properties has been compared with deep learning technics using recurrent neural networks (RNN). Efficient analysis of the Big Data category of status data time series has revealed important behaviour of these sensor networks. This study work is part of PhD research task and project.

Kivonat

A Tárgyak Internete számos esetben feltételezi, hogy a szenzorok kommunikációs mechanizmusa ne csak az adatátvitel hatékonysága, hanem a továbbításhoz szükséges energia mennyisége szempontjából is optimális legyen. A szenzorhálózatok egyik leginkább elemzett típusa a LEACH (Low Energy Adaptive Clustering Hierarchy) rendszer, amelynek viselkedése olyan paraméterektől függ, mint a populáció sűrűsége, klaszterfej választási algoritmus, csomópontok energiájának, illetve fizikai helyzetének heterogenitása, nyelő csomópont sebessége, adatok tömörítésének mértéke, illetve adatkeretek mérete. A rendszer komplexitását többszáz szimulációs eset által előállított állapotadat halmaz segítségével elemeztem. A LEACH hálózat működését, illetve ennek függőségi jellemzőit a generált idősorok összevetésével dolgoztam fel, amihez visszacsatolásos neurális hálózatra (RNN) épülő mélytanuló technikákat alkalmaztam. A Big Data kategóriájú állapotadat idősorokat tartalmazó halmaz hatékony feldolgozása ezen típusú szenzorhálózatok mély viselkedési jellemzőinek megismerésére adott lehetőséget. Az elemzési tevékenység PhD kutatási munka, illetve kutatási projekt részét képezi.

Kulcsszavak: vezeték nélküli szenzor hálózat, Low Energy Adaptive Clustering Hierarchy (LEACH), visszacsatolásos neurális hálózat, kapcsolás, klaszter.

1. Bevezetés

A vezeték nélküli szenzor hálózatok csomópontjai egy érdekeltségi tartományt (AoI – Area of Interest) figyelnek meg és funkciójuktól függően adatokat mintavételeznek, amiket a huzalos hálózathoz is kapcsolódó bázisállomáshoz (BS – Base Station) vagy nyelő funkciójú (SN – Sink Node) csomópont-hoz közvetlenül vagy közvetve juttatnak el további feldolgozás céljából. Ez a továbbítási folyamat a kutatások középpontjában álló, különböző útválasztási mechanizmusok segítségével valósul meg. Mivel a szenzor eszközöknek korlátos energiájuk van, meghatározó jelentősége van az energiájuk felhasználási hatékonyságának. Ugyanakkor ezek a szenzorok erőteljesen koncentrálódnak ember által nem, vagy

csak nagyon nehezen megközelíthető fizikai területekre, így tápforrásuk gyakorlatilag nem cserélhető. Ezáltal működési idejük záloga az energiafelhasználásuk hatékonysága.

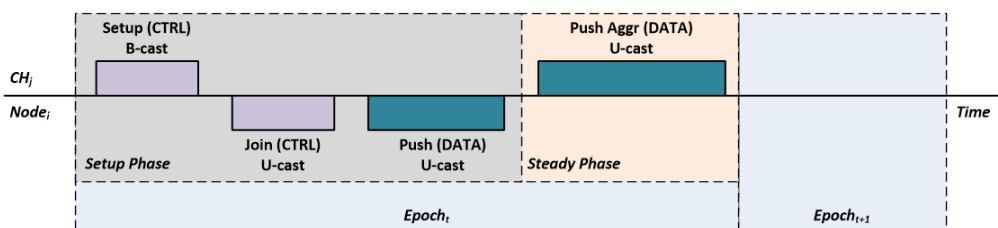
Az útválasztási protokollok kulcsfontosságú szerepet töltenek be az aggregált adatok küldésénél, ami miatt ilyen feladatokat intelligensen kell lekezelni. A WSN rendszer sikeres modellje az, amely jó kompromisszumot tud képezni a maximális mennyiségű adatgyűjtés és a minimális mértékű energiafogyasztás között. A WSN útválasztási mechanizmusoknál a klaszterezés fontos szempontként jelenik meg, mivel hálózat szinten hatékony energiamegtakarítást és adatkézbésítést nyújt. A klaszterezést is magában foglaló hierarchikus útválasztás a szenzorok előnyös elrendezési módszerének bizonyul [1], [14]. A módszer ugyanakkor növeli a skálázhatóságot, csökkenti a kieső energia mennyiségét, és a késleltetési időt, miközben jó kapcsolódást, terhelés kiegyenlítést biztosít megnövelt hálózati élettartammal [2], [17], [19].

A LEACH egy hierarchikus, klaszter alapú, energiahatékony útválasztási mechanizmus. A rendszer élettartamát véletlenszerűen választott klaszterfej (CH – Cluster Head) elemekkel növeli, amelyek a klasztertagoktól érkező kereteket aggregálás után továbbítják az SN nyelőhöz. Ez a közbeeső továbbító lépés lényegesen lecsökkenti a csomópontok által a rádiós csatorna működéséhez igénybe vett energiát, mivel az adatkeret küldési fogyasztása a távolsággal $2b$ kitevőjű hatványfüggvény szerinti. Itt b az útvonal veszteségi exponens, értéke az útvonal terjedési tulajdonságaitól függ és leggyakrabban $b \geq 2$. Könnyű belátni, hogy egy d távolságot áthidaló, félúton besegítő csomópont általi kétlépéses küldésnél az energia 2^{1-2b} része a d távolságú egylépéses küldésének, ami legrosszabb esetben is nyolcszoros nyereséget jelent a kétlépéses megoldás javára. A klaszterek kialakítása a mechanizmus számára megbízhatóságot és skálázhatóságot nyújt. A fúziós technikának köszönhetően csak értékes adatok küldésére kerül sor, mellőzve a redundáns adatok küldését a csomópontok szintjén. Az egymás utáni epoch időtartamokban véletlenszerűen kiválasztott CH csomópontok fizikai térben többnyire kiegyenlítetté teszik az energia felhasználását. Adott epoch időben egy tetszőleges CH csomópont nulla vagy annál nagyobb számú klaszter taggal rendelkezik. A két, egymással váltakozó fázis minden epoch időtartamban egyszer kerül sorra. A *Setup* (Felépítés) fázisban kerül sorra a CH csomópontok, valamint klaszterenkénti tagok kiválasztása. A *Steady* (Szilárd) fázisban a klasztertagok ütközésmentesen, TDMA vagy CDMA segítségével küldik kereteiket a saját CH csomópontjuknak, amit aggregálás után a CH ad tovább az SN nyelőnek.

A dolgozat további felépítése a következő: a második fejezetben a LEACH mechanizmus architektúráját és energiahatásait tárgyalom. A harmadik fejezetben a sokdimenziós adathalmazok elemzéséhez használt módszereket mutatom be: főkomponens analízis és a szinguláris érték felbontás matematikai háttérét, valamint a visszacsatolásos neurális hálózatok fontosabb változatait. A negyedik fejezet az újonnan javasolt, CB-LEACH nevű WSN rendszert vizsgálom háromszázhatvan különböző szimuláció segítségével előállított szintetikus állapotadatsor alapján. Az utolsó fejezetben összefoglalom az eredményeket és a kutatási munka lehetséges folytatásának irányait fogalmazom meg.

2. A LEACH mechanizmus architektúrája és működése

Mivel a LEACH hierarchikus mechanizmus jelentős energia megtakarítást biztosít, a CH funkciójú csomópontokat minden egyes epoch időtartamban véletlenszerűen választja ki, majd két váltakozó fázis szerint működteti. A CH átveszi a klasztertagok kereteit, majd aggregálás után az SN nyelőnek küldi el. Mivel a CH funkció többlet energiafogyasztással jár, ezért ezt ritkán kapják meg a WSN csomópontjai. Egy epoch időtartam két fázisból áll: *Setup*, illetve *Steady*.



1. ábra. Keretek küldési mechanizmusa epoch időtartamonként.

Így az N szenzor csomópontból álló WSN esetén az egymásutáni időszakokban e két fázis váltakozva következik be.

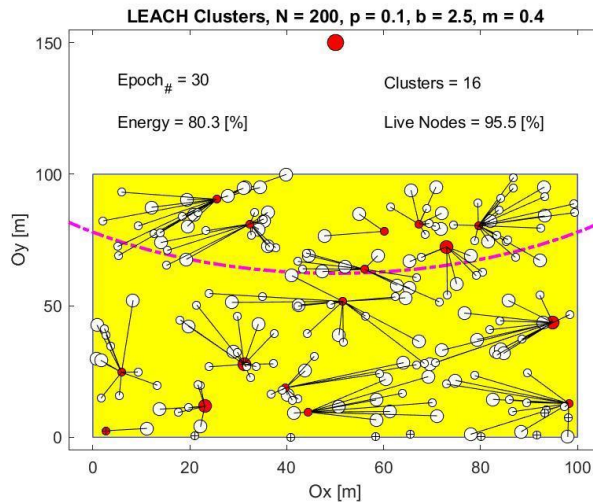
Setup (Felépítés) fázisban a CH kiválasztása és meghirdetése, valamint a klaszterek kialakítása történik. CH funkció csomópont szintű bekövetkezésének valószínűsége $p \in (0, 1)$. Mindegyik $Node_n$ csomópont $(0, 1)$ intervallumban egyenletes eloszlás szerint sorsol egy véletlen $q_n \in (0, 1)$ értéket, majd összeveti az aktuális epoch időben számolt $T(n)$ küszöbértékével:

$$T(n) = \begin{cases} \frac{p}{1-p \cdot \text{mod}(r, \frac{1}{p})}, & \text{ha } n \in G \\ 0, & \text{egyébként} \end{cases} \quad (1)$$

ahol r az aktuális epoch (round) sorszama, míg G azon csomópontok halmaza, amelyek az utóbbi $1/p$ epoch időtartamban nem szereztek CH funkciót. $Node_n$ csomópont az r -edik epoch periódusban klaszterfej funkciót szerez, ha a sorsolt $q_n < T(n)$. Ha $Node_n$ az utóbbi $1/p$ darab epoch bármelyikében CH volt, akkor $T(n) = 0$, így bármely $q_n > T(n)$, így nem szerezhethet újból klaszterfej funkciót. Ezzel a szabállyal a WSN mindegyik csomópontja azonos valószínűséggel lehet CH. A klaszterfej funkciójú csomópont üzenetszórásos módon küldött kontroll típusú kerettel értesíti a rendszer összes csomópontját a saját állapotáról. A CH funkció nélküli csomópontok mindegyike a klaszterfej forrásokból érkező jel térerőssége alapján önállóan választja ki a számára legelőnyösebb CH-t. A LEACH szimuláció során a térerőt kizárólagosan a távolságtól feltételezzük függőnek, így az $A(d)$ rádiós csillapítás bitküldésenként az alábbi összefüggés szerinti:

$$A(d) = \frac{E_{Rx}(d)}{E_{Tx}} = \begin{cases} 1, & \text{ha } d \in [0, \delta) \\ \left(\frac{\delta}{d}\right)^2, & \text{ha } d \in [\delta, d_0) \\ \left(\frac{\delta}{d_0}\right)^2 \cdot \left(\frac{d_0}{d}\right)^{2b}, & \text{ha } d \in [d_0, \infty) \end{cases} \quad (2)$$

ahol d a küldő és a vevő közötti távolság, E_{Rx} és E_{Tx} a fogadott és a kiküldött energia, δ a rádió antenna geometriai paramétere (centiméter nagyságrendű), d_0 terjedési küszöbtávolság, b az útvonalveszteség exponens, ahol $b \geq 2$. A saját adattal rendelkező klasszikus csomópont a számára legelőnyösebb, azaz legközelebbi CH-et választja ki és kapcsolódik az adott klaszterhez. Adat hiányában a klasszikus csomópont alvó állapotba megy epoch periódus ideig energia megtakarítás céljából. A CH a klaszter tagjai számára TDMA vagy CDMA kontroll típusú kerettel értesítést közöl, amivel az aktuális epoch időben garantálja az ütközésmentes keretküldést a klaszterfej felé.



2. ábra. LEACH hálózat csomópontjai és klaszterei adott epoch idején.

Az piros színű SN nyelő közepén, felül helyezkedik el, a szenzor csomópontok a sárga AoI mezőben rögzített, egyenletesen véletlen koordinátákban található, mozdulatlanul. Adott csomópont kezdeti energiája a saját körének sugarával arányos. A CH funkciók a piros színű csomópontoknál vannak. Adott klaszter a CH és a hozzá kapcsolódó, szakasszal jelölt csomópontokból áll. A lemerült csomópontokat közepén „+” karakterrel jelöljük. A nagy körív sugara d_0 ami a terjedési küszöbtávolság.

Steady (Szilárd) fázisban a klaszter csomópontok a saját CH számára ütközésmentesen küldik el saját keretüket. Az összes alvó csomópont egyetlen klaszterhez sem tartozik, így ezek sem kommunikációhoz, sem működéshez nem használnak energiát. Minden egyes CH fogadja a saját tagjaitól érkező adatkereteket, aggregálja ezeket és közvetlenül a SN nyelőhöz küldi. A klasztertagoktól érkező adatok hasonlósága vagy eltérése alapján az aggregáció foka $g \in (0, 1]$. Ha g kicsi, akkor hasonló értékeket tartalmazó keretek érkezték a CH-hez, ellenkező esetben a beérkező keretek bitjeinek száma csak kis mértékben csökkenthető. A különböző CH csomópontok ugyancsak TDMA vagy CDMA segítségével ütközésmentesen küldik saját aggregált adatkeretüket az SN nyelőhöz.

A LEACH archetípusa az elosztott útválasztási protokolloknak és egyben az egyik leghatékonyabb energia menedzsmenttel rendelkező WSN mechanizmus. Ugyanakkor nem szabad elhallgatnunk néhány hányosságát sem [1]. Tetszőleges csomópont CH funkcióbba lépése független az adott csomópont maradék energiájától, ami a lefedett területen a klaszterek fizikai elhelyezkedésében egyenletlen eloszlást eredményez [2], [11], [15-16]. Ezt fokozza az a tény is, hogy a CH csomópont akár távolabb is lehet az SN nyelőtől, mint a saját klaszterének tagjai, rossz hatékonysággal kézbesíti az adatokat. Ezáltal bizonyos CH csomópontok hamarabb lemerítik energiaforrásukat, mint más csomópontok, kialakítva energiamentes lyukakat a fizikai térben [3], [4-10], [18-20]. Más esetben a hálózat biztonságossága csökken, ha kevés energiával rendelkező csomópont kap CH funkciót, mivel az ide érkező klasztertagok kereteinek küldése éppen az SN nyelőhöz való továbbítás során szakadhat meg energiahiány miatt.

3. Költség kiegyenlítéses mechanizmus: CB-LEACH

Az általam javasolt változtatással az alap LEACH mechanizmust további készséggel, illetve intelligenciával ruháztuk fel. Ez egy költség kiegyenlítéses (CB – Cost Balanced) LEACH változat, amely összetett metrika alapján dönt a keretek továbbítását végző útvonalra vonatkozóan. CB-LEACH esetén lehetővé tesszük az SN csomópont mozgását adott nyomvonalon, valamint a csomópontok számára az optimális CH kiválasztását. Mivel előfordulhat, hogy az aktuális epoch időben adott csomópontok közelebb helyezkednek el a mozgó SN nyelőhöz, mint bármelyik választott CH, így előnyösebb ezeknek közvetlenül az SN-hez küldeniük keretüket, mint közvetve a választott CH-n keresztül. Ennek érdekében a választott CH csomópontok halmazába bevesszük a SN nyelőt is, aminek energiája időben nem csökken és a legnagyobb a WSN rendszerben. A klasszikus csomópont klaszterfej keresési fázisában nem csak a lehetséges CH-ek csomópontoktól mért távolságát, hanem azok a energia szintjét is figyelembe vesszük. Ennek érdekében az alábbi metrika alapján dönt adott szenzor csomópont, hogy melyik CH-hez kapcsolódjon:

$$COST(i, j, \alpha) = \alpha \cdot \frac{E_0}{E_{CH_j}} + (1 - \alpha) \cdot \left(\frac{D(i, j)}{d_0} \right)^{2b} \quad (3)$$

ahol $COST(i, j, \alpha)$ a $Node_i$ CH_j -hez $\alpha \in [0, 1]$ egyensúlyozási faktorról számolt metrikája, $D(i, j)$ a $Node_i$ és CH_j közötti távolság, d_0 a terjedési küszöbtávolság, b pedig az útvonalvesztés exponens. Mivel a (3) összefüggés első tagjának nevezője hosszú idő múlva nulla lehet, ezért a rendszer tulajdonságait az utolsó működő csomópontig elemezzük, így a képlet minden tagja pozitív és korlátos mennyiség marad. Aktuális epoch időben $Node_i$ azt a CH_j -t választja, amelyre $COST(i, j, \alpha)$ metrika értéke a legkisebb.

Megfigyelhető, hogy ha $\alpha = 0$, akkor visszakapjuk a LEACH mechanizmus mobilitással kiegészített CH-es változatát, amit továbbfejlesztett ENH-LEACH (Enhanced) néven azonosítunk a továbbiakban. Ha nem változtatjuk az SN nyelő helyzetét időben, akkor visszakapjuk az alap (Basic) LEACH mechanizmust, amit a továbbiakban BAS-LEACH néven azonosítunk. Ugyanakkor, ha $\alpha = 1$, akkor kizárólag a CH energiája számít, ami versengéses helyzetet generál a potenciális CH-ek között és mindegyik $Node_i$ ugyanazt az egyetlen CH-et fogja választani, vagyis a maximális energiájú SN nyelőt. Ebben az esetben mindegyik csomópont közvetlenül küldi adatait az SN nyelőhöz, függetlenül a többi potenciális CH-től. Ezt az útválasztási mechanizmust DS-nek (Direct Sequence) fogjuk nevezni a továbbiakban. Szimulációs méréseink bizonyították, hogy az ENH-LEACH és a DS egymástól lényegesen eltérő útválasztási protokoll. Az α egyensúlyozási faktor két szélsőértéke különböző szempontokból előnyökkel, illetve hátrányokkal jár, így $\alpha \in (0, 1)$ tartományban az egyensúlyozási faktornak a LEACH mechanizmusra erőteljes hatása kell, hogy legyen. Ezek alapján a javasolt CB-LEACH egy protokollcsalád az alábbiak szerint:

$$CB - LEACH = \begin{cases} ENH - LEACH & , \quad ha \alpha = 0 \\ \text{Módosított LEACH család} & , \quad ha \alpha \in (0, 1) \\ DS & , \quad ha \alpha = 1 \end{cases} \quad (4)$$

A DS minden csomóponttól közvetlenül az SN nyelőhöz küldi a keretet. Ez nagy távolságokat jelent, ami jelentős mennyiségű energiát fogyaszt. Ugyanakkor a véletlenszerű küldési mechanizmus miatt a csomópontok maradék energiájának síkbeli eloszlása egyenletesnek várható. Az ENH-LEACH a WSN fizikai mezőjét két részre bontja, mivel az SN nyelő klaszterfejként való kezelése az SN nyelő d_0 sugarú környezetében lévő csomópontok számára előnyös energiafogyasztást biztosít keretküldés idejére. Nyilvánvaló kérdésként fogalmazódik meg, hogy a módosított LEACH család az α paraméter függvényében milyen tulajdonságokat örököl a két határesetből. További kérdések a CB-LEACH rendszer többi paraméterétől való függésével kapcsolatosan tehetők fel.

3. Sokdimenziós adathalmaz elemzési módszerei

A CB-LEACH rendszer jelentős mennyiségű paramétertől függ, ezért többszáz esetben történt a szimuláció lefuttatása. Minden egyes esetben többnyire milliós nagyságrendű epoch periódusidő alatt fejeződött be az adott WSN kommunikációs tevékenysége. Joggal jelenik meg annak igénye, hogy beazonosítsuk jelen hálózat viselkedését legjelentősebben befolyásoló paramétereket. Erre két elemzési technikát alkalmaztam, amelyek közös hatása erőteljes módszernek bizonyult a WSN rendszer viselkedésének tanulmányozása során.

3.1. Főkomponens analízis, illetve szinguláris érték felbontás

A főkomponens analízis (PCA – Principal Component Analysis) valamint a szinguláris érték felbontás (SVD – Singular Value Decomposition) négyzetes, illetve téglalap mátrixok egyszerűsítését teszik lehetővé. Kinyerhető a mátrix lényeges jellemzői. Mint ismeretes, egy tetszőleges $A \in M_{m \times n}$ mátrix felfogható transzformációnak is. Ha C négyzetes mátrix, azaz $C \in M_{n \times n}$, akkor a karakterisztikus polinom gyökei, $\lambda_i, i = \overline{1, n}$ a mátrix sajátértékei. Ha C mátrixunk szimmetrikus, azaz $C = C^T$ akkor a sajátértékek valós számok és nem negatívak, vagyis:

$$\det(C - \lambda_i I_n) = 0, \forall i = \overline{1, n}, \text{ ahol} \quad (5)$$

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0 \quad (6)$$

Ugyanakkor minden egyes λ_i sajátértékhez tartozó $b_i \in M_{n \times 1}$ sajátvektor esetén teljesül az alábbi lineáris transzformációs összefüggés:

$$C * b_i = \lambda_i b_i, \quad \forall i = \overline{1, n} \quad (7)$$

Az előnyös tulajdonságai miatt a bázisvektorokat unitárisnak, vagyis 2-normáját egység méretűnek választjuk:

$$\|b_i\|_2 = \sum_{j=1}^n b_{i,j}^2 = 1, \quad \forall i = \overline{1, n} \quad (8)$$

Az egyenként n sorból álló $x_i, i = \overline{1, n}$ oszlopvektorok összessége egy B mátrixot képez, ami egyben bázisa az C mátrix sajátalterének, vagyis:

$$B = (b_1, b_2, \dots, b_n) \in M_{n \times n} \quad (9)$$

$$C * B = B * \Lambda, \text{ ahol} \quad (10)$$

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 & 0 \\ 0 & \lambda_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \lambda_{n-1} & 0 \\ 0 & 0 & \dots & 0 & \lambda_n \end{pmatrix} \quad (11)$$

Jobbról szorozva a (9) összefüggés mindkét oldalát a bázismátrix inverzével, a C mátrixunk alábbi faktorizációját kapjuk:

$$C = B * \Lambda * B^{-1} \quad (12)$$

Kibontva a (12) összefüggést, megkapjuk a négyzetes mátrix egyszerűsítését lehetővé tevő összeget:

$$C = \lambda_1 b_1 * b_1^T + \lambda_2 b_2 * b_2^T + \dots + \lambda_r b_r * b_r^T + \lambda_{r+1} b_{r+1} * b_{r+1}^T + \dots + \lambda_n b_n * b_n^T \quad (13)$$

Mivel azonban a gyakorlatban a $\lambda_i, i = \overline{1, n}$ sajátértékek széles skálán veszik fel értéküket, miközben a $b_i, i = \overline{1, n}$ sajátvektorok mindegyike unitáris tulajdonsága miatt egyformán korlátos, ezért a (13) szerinti összeg r -edik tagja utáni mátrixoktól eltekinthetünk. Ezáltal kapjuk C mátrixunk egyszerűsített változatát:

$$C \approx \widehat{C} = \lambda_1 b_1 * b_1^T + \lambda_2 b_2 * b_2^T + \dots + \lambda_r b_r * b_r^T = \widehat{B} * \widehat{\Lambda} * \widehat{B}^T \quad (14)$$

Az összeg tagjait adó r értékét az alapján határozzuk meg, hogy a csökkenő sorrendbe rendezett első r sajátérték az összes sajátérték által képviselt energia jelentős százalékát (pl. $Th = 95\%$) adja, vagyis:

$$\frac{\sum_{i=1}^r \lambda_i}{\sum_{i=1}^n \lambda_i} \leq Th \quad (15)$$

A rendszereknél mintavételezett válaszokat $Y \in \mathcal{M}_{m \times n}$ téglalap mátrixként foghatjuk fel, ahol minden egyes $y_i \in \mathcal{M}_{m \times 1}, i = \overline{1, n}$ oszlopvektor egy-egy mérési sorozat és n a mérési esetek száma:

$$Y = (y_1, y_2, \dots, y_n) \in \mathcal{M}_{m \times n} \quad (16)$$

gyakorlatban adott mérési esetben mintavételezett adatok száma lényegesen nagyobb, mint a mérési esetek száma, vagyis $m \gg n$. Ez komoly gondot okoz a számítógépes feldolgozásban az operatív memória és a processzor kapacitáskorlátja miatt. Az y_i , illetve $y_j, i, j = \overline{1, n}$ mérési esetek közötti haonlóságot, illetve különbséget a korrelációs mátrix segítségével határozhatjuk meg:

$$C = Y^T * Y \in \mathcal{M}_{n \times n} \quad (17)$$

Itt C már négyzetmátrix, szimmetrikus és a teljes főátlón 1-es értékeket tartalmaz, mivel minden mérési eset önmagával 100%-ig hasonló. Ugyanakkor C négyzetes mátrixunk már jóval kisebb dimenziójú, mint Y . Mivel azonban C nem ad betekintést adott mérési eset belső történéseibe, ezért Y mátrixot szinguláris érték felbontás alapján faktorizáljuk. Meghatározhatók $U \in \mathcal{M}_{m \times m}, \Sigma \in \mathcal{M}_{m \times n}$, illetve $V \in \mathcal{M}_{n \times n}$ mátrixok az alábbi felételeknek megfelelően:

$$Y = U * \Sigma * V^T \quad (18)$$

$$U * U^T = U^T * U = I_m \text{ és } V * V^T = V^T * V = I_n \quad (19)$$

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_{n-1} & 0 \\ 0 & 0 & \cdots & 0 & \sigma_n \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \quad (20)$$

Σ átlós mátrix σ_i , $i = \overline{1, n}$ értékeit Y téglalap mátrix szinguláris értékeinek nevezzük. Felhasználva (17) és (18) összefüggéseket, valamint (19) tulajdonságokat, kapjuk:

$$C = Y^T * Y = (U * \Sigma * V^T)^T * U * \Sigma * V^T = V * \Sigma^T * U^T * U * \Sigma * V^T = V * \Sigma^2 * V^T \quad (21)$$

Összevetve C -re kapott (12) és (21) összefüggéseket, valamint az inverzekre (19) összefüggésből származó tulajdonságokat (azaz $U^T = U^{-1}$ és $V^T = V^{-1}$), kapjuk:

$$V = B \quad \text{és} \quad \Sigma = \sqrt{\Lambda} \quad (22)$$

Utóbbi két összefüggés jelentősége meghatározó, mivel Y számára B bázis, illetve a pozitív sajátértékek négyzetgyöke pontosan a szinguláris értékeket adja:

$$\sigma_i = \sqrt{\lambda_i}, \quad \forall i = \overline{1, n} \quad (23)$$

Hasonlóan a (13) összefüggéshez, itt is felírhatjuk Y mátrixot szinguláris vektorokból előállított mátrixok súlyozott összegeként, ahol a súlyok a szinguláris értékek:

$$Y = \sigma_1 u_1 * v_1^T + \sigma_2 u_2 * v_2^T + \dots + \sigma_r u_r * v_r^T + \sigma_{r+1} u_{r+1} * v_{r+1}^T + \dots + \sigma_n u_n * v_n^T \quad (24)$$

Mivel u_i , illetve v_i^T , $i = \overline{1, n}$ korlátos vektorok, valamint σ_i súlyok csökkenő sorrendbe rendezett értékek, ezért a (24) összegben is elhanyagolhatók az r -edik tag utáni mátrixok. Ezt úgy tehetjük meg könnyen, hogy a lényegtelen szinguláris értékeket nullának vesszük, vagyis $\sigma_i = 0$, $i = \overline{r+1, n}$. Ezáltal megkapjuk Y egyszerűsített változatát, $\hat{Y} \in \mathcal{M}_{m \times n}$ mátrixot az $\hat{U} \in \mathcal{M}_{m \times r}$ és $\hat{V}^T \in \mathcal{M}_{r \times n}$ meghatározó bázisok, illetve a meghatározó szinguláris értékeket tartalmazó $\hat{\Sigma} \in \mathcal{M}_{r \times r}$ mátrix függvényében:

$$Y \approx \hat{Y} = \sigma_1 u_1 * v_1^T + \sigma_2 u_2 * v_2^T + \dots + \sigma_r u_r * v_r^T = \hat{U} * \hat{\Sigma} * \hat{V}^T \quad (25)$$

$$\hat{U} = (u_1, u_2, \dots, u_r), \quad \text{és} \quad \hat{V}^T = (v_1^T, v_2^T, \dots, v_r^T) \quad (26)$$

Nagyon fontos kiemelni az alábbiakat:

i) A gyakorlatban elemzett rendszer tulajdonságaitól függően a fontos szinguláris értékek darabszáma (r) a mérési esetek számának (n) töredéke lehet. A (25) összefüggés lényegesen csökkenti a feldolgozási mennyiséget, illetve előnyösen kiemeli a rendszer működését meghatározó legfontosabb komponenseket.

ii) \hat{Y} közelítő mátrix tulajdonképpen Y -nak zajtól jelentős mértékben mentesített változata.

iii) Az u_i , $i = \overline{1, r}$ bázisvektorok saját eseteknek megfelelő virtuális mérési sorozatokként foghatók fel, csökkenő fontossági sorrendben.

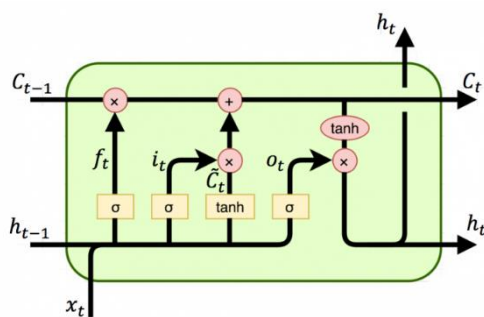
iv) A (25) összefüggés mindkét oldalát \widehat{U}^T -vel balról való szorzása alapján $Z = \widehat{U}^T * Y = \widehat{\Sigma} * \widehat{V}^T$ összefüggést kapjuk, ami azt jelenti, hogy az így kapott Z mátrix nem más, mint az Y mátrix vetülete a fontosabb mérési sorozat vektorokból képezett bázisra.

3.2. Fontosabb visszacsatolós neurális hálózatok

A gyakorlatban leginkább alkalmazott visszacsatolós neurális hálózatok az LSTM (Long-Short Term Memory), valamint a GRU (Gated Recurrent Unit). A visszacsatolás miatt mindkettő memória tulajdonsággal rendelkezik, azaz képes a beérkező adatsorokban szekvenciális mintázatok beazonosítására. Ezt azáltal képes végezni, hogy a kimenet aktuális állapota, C_t egyaránt függ a bementre kerülő aktuális x_t értéktől, a kimenet előző hozzáférhető, illetve rejtett értékétől C_{t-1} , h_{t-1} és közvetlenül azok előzetes sorrendjétől:

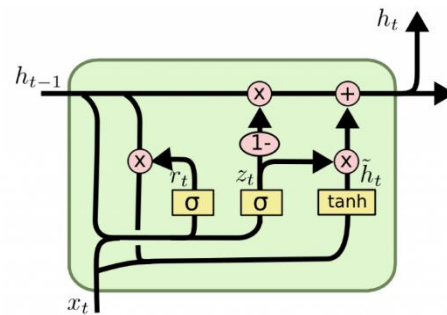
$$C_t = F_1(x_t, C_{t-1}, h_{t-1}) \quad \text{és} \quad h_t = F_2(x_t, C_{t-1}, h_{t-1}) \quad (27)$$

Az itt szereplő állapotegyenlet formálisan nem véletlenül hasonlít a digitális szekvenciális áramkörök működését leíró általános szabályhoz. A 3. ábrán és a 4. ábrán látható két hálózat tulajdonságait tömören foglalom össze.



3. ábra. LSTM RNN szerkezete.

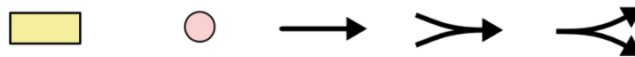
Forrás: <http://dprogrammer.org/rnn-lstm-gru>



4. ábra. GRU RNN szerkezete.

Forrás: <http://dprogrammer.org/rnn-lstm-gru>

Az LSTM modul négy darab speciális módon egymáshoz kapcsolódó rétegből áll: Forget, Input, Cell State, Output/Hidden State. A felépítésében résztvevő modulok jelölését az 5. ábra szemlélteti. Minden vonal vektort szállít. A neurális hálózati réteg (sigmoid, illetve tanh aktivációs függvény) sárga téglalappal, a pontszerű művelet (összegzés, szorzás, tanh függvény) lila körrel van jelölve. Találkozó vonalak vektor konkatenálást, szétváló vonalak vektormásolást jelentenek.



5. ábra. RNN hálózatok funkcióinak jelölése

A cella előző, illetve jelenlegi állapotát C_{t-1} , illetve C_t jelöli. Az egyes cellák állapot bementeit, illetve kimeneteit h_{t-1} , valamint h_t jelöli. Az egyes vonalakon lévő mennyiségeket az 1. Táblázat tartalmazza.

1. Táblázat: Az LSTM és a GRU rétegek funkciói

LSTM funkciók és művelet(ek)		GRU funkciók és művelet(ek)	
Forget	$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$	Update	$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$
Input	$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ $\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_f)$		$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$
Cell State	$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$	Output, Hidden State	$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t] + b_f)$
Output, Hidden State	$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$ $h_t = o_t * \tanh(C_t)$		$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$

Az LSTM és a GRU visszacsatolós hálózati rétegek a bizonyos témák esetén nagyon hatékonyan használhatók. Ilyenek a struktúrát tartalmazó adatok elemzése, ahol szükség van nem csak a rövidtávú, de a hosszútávú memóriára is. Ugyanakkor azt is hangsúlyozni kell, hogy ezek a hálózatok más, visszacsatolás nélküli neurális hálózatokhoz képes többszámolást igényelnek úgy a tanítási, mint a tesztelési fázisban. Emiatt idő struktúra nélküli adatok elemzésénél a visszacsatolós neurális hálózatok indokolatlanul sok feldolgozási erőforrást vesznek igénybe, különösebb előnyök nélkül.

4. CB-LEACH rendszer elemzése és tulajdonságai

Direkt szekvencia (DS – Direct Sequence), az alap LEACH (BAS-LEACH: Basic LEACH), valamint a továbbfejlesztett LEACH (ENH-LEACH: Enhanced LEACH) különböző paraméterek szerinti szimulációinak $n = 360$ esetéből kinyert szintetikus állapot adatokat elemzem és értékelem az előző fejezetben ismertetett módszerekkel. Ezek tulajdonképpen a CB-LEACH speciális eseteinek foghatóak fel. A szimulációs esetek során a WSN rendszer nem változtatott jellemzői a 2. Táblázatban találhatóak.

2. Táblázat. CH-LEACH rendszer állandó jellemzői

Jellemző	Érték	Jellemző	Érték
Érdekeltségi terület sugara	$R_0 = 50 \text{ m}$	Energia arány	$a + 1 = 2$
Terjedési küszöbtávolság	$d_0 = 87,71 \text{ m}$	Elektronika fogyasztás	$E_{elec} = 50 \text{ nJ/bit}$
SN nyelő koordinátái	$(x_0, y_0) = (-2R_0, 0)$	Antenna erős. fogyasztás	$E_{amp} = 0.1 \text{ nJ/bit}$
Útvonal veszteségi exponens	$b = 2$	Aggregálási fogyasztás	$E_{da} = 5 \text{ nJ/bit}$

A CB-LEACH α kiegyensúlyozási faktora a (3) és (4) összefüggések alapján befolyásolja a klaszterfej választási stratégiát, illetve a WSN hálózat típusát. A klaszterfej választási átlagos valószínűség, p az (1) összefüggésnél kap szerepet. Jellemzően viszonylag kis értékekkel szokták tesztelni a rendszer működését.

Heterogén kezdeti energiaszintek hatásának megjelenése céljából az N darab szenzor csomópontot két energiacsoportba soroljuk: NN (Normal Node), illetve AN (Advanced Node). Az NN kezdetben $a + 1 > 1$ arányban több energiával rendelkezik, mint az NN. Az összes szenzor csomópont $m \in (0, 1)$ hányada AN, a maradék pedig NN. E két csoport, illetve a teljes WSN rendszer kezdeti energiája az alábbi:

$$E_N = (1 - m) \cdot N \cdot E_0 \quad (28)$$

$$E_A = m \cdot (a + 1) \cdot N \cdot E_0 \quad (29)$$

$$E_T = E_N + E_N = (a \cdot m + 1) \cdot N \cdot E_0 \quad (30)$$

Az adatkeretek mérete ugyancsak befolyásolja a WSN rendszer működését, mivel a kontroll keretek mérete változatlanul a szokásos 20 bájt és a LEACH mechanizmus Setup fázisában forgalmazott kontroll keretek is energiát fogyasztanak. Ha az adatkeret és a kontroll keret mérete közeli, akkor az energiafogyasztás kisebb mértékben függ az adatkeretektől.

A szenzor csomópontok által mintavételezett adatok típusa lehetséges heterogén (pl. közlekedési eszközök mennyisége az autópálya különböző pontjain), vagy homogén (pl. hőmérséklet értékek egy mezőgazdasági terület különböző pontjain), Ezt $g \in (0,1)$ aggregációs szint paraméterrel jellemezzük. Adott klaszterfejnél végzett aggregáció után az SN nyelőhöz küldött FRAME adatmennyiség a klasztertagoktól beérkezett F_i adatkeret összege g részére csökken, vagyis:

$$FRAME = g \cdot \sum_{i=1}^s F_i \quad (31)$$

Ez lesz az SN nyelőhöz elküldött adatbit mennyiség, ahol s a klasztertagok száma. Heterogén adatokat kis mértékben lehet tömöríteni (g nagy), míg homogén adatokat jobban lehet tömöríteni (g kicsi).

Az SN nyelő mozgási sebességének nagysága v , ami nulla esetben az alap LEACH mechanizmust eredményezi. Pozitív esetben a nyelő egyenes vonalban egyenletesen oda-vissza mozgást végez szimmetrikusan a WSN által lefedett érdekeltségi területen (ld. 5. ábra).

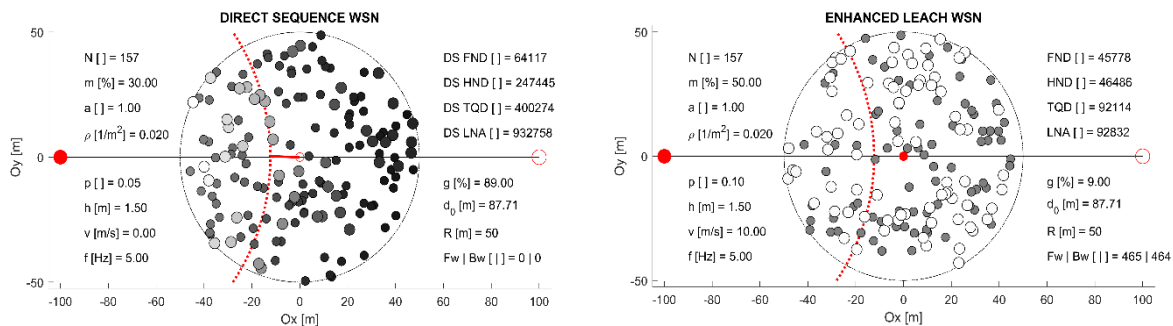
A szimulációs környezet hat paraméterének alkalmazott értékeit a 3. Táblázat tartalmazza. A szimulációk minden egyes paraméter felsorolt értéke mellett lefutottak. Bármely két szimuláció csak egyetlen paraméter értékben különbözik. Belátható, hogy ezáltal a különböző szimulációk száma:

$$n = [\alpha] \cdot [p] \cdot [m] \cdot [L] \cdot [g] \cdot [v] = 5 \cdot 2 \cdot 3 \cdot 2 \cdot 2 \cdot 3 = 360 \quad (32)$$

3. Táblázat. CH-LEACH rendszer változó paramétereinek értékei

Paraméter	Alkalmazott értékek	Paraméter	Alkalmazott értékek
α	0; 0,25; 0,5; 0,75; 1	L (bit)	1000; 4000
p	0,05; 0,1	g (%)	10; 90
m	0,3; 0,5; 0,7	v (m/s)	0; 5; 10

Példaként az 5. ábra a DS és az ENH-LEACH mechanizmusok energiafogyasztásának időbeli előrehaladását szemlélteti. A nagyobb sugarú körök AN, a kisebb sugarú körök NN csomópontok. A kör szürke árnyalata mutatja az adott csomópont időbeni lemerülését. Minél világosabb egy kör, annál később merült le a csomópont. Az SN nyelő mozgása esetén az (x_0, y_0) , illetve $(-x_0, y_0)$ pozíciók között változtatja helyzetét. Az SN nyelő kezdeti helyzete a bal oldali piros kör. Mozgása esetén a jobb oldali külső üres körig halad oda-vissza.



5. ábra. Elemzett WSN rendszer kör alakú érdekeltségi területtel.

Bal: DS ($\alpha = 0,5; p = 0,1; m = 0,3; L = 1000$ bit; $g = 90\%; v = 0$ m/s)

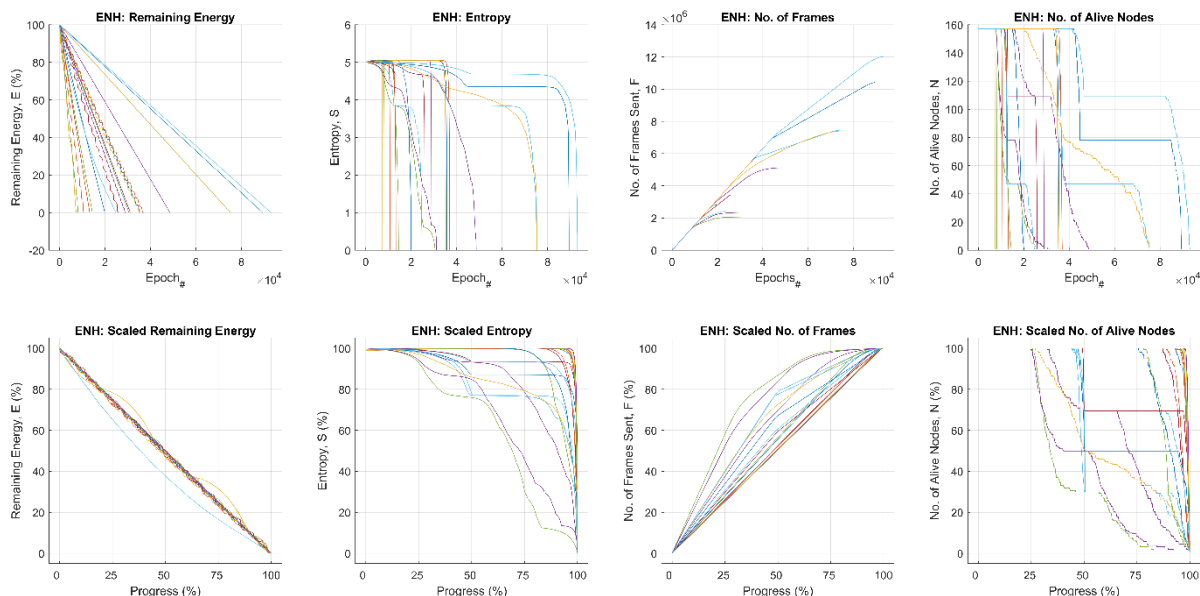
Jobb: ENH ($\alpha = 0,5; p = 0,1; m = 0,5; L = 1000$ bit; $g = 10\%; v = 10$ m/s)

Megfigyelhető, hogy mozdulatlan SN esetén a távolabbi DS csomópontok hamarabb lemerülnek (bal oldali ábra), míg mozgó SN esetén az ENH-LEACH csomópontok az érdekeltségi területen belül közel azonos időben merülnek le (jobb oldali ábra). A kezdeti energia-középpont elmozdulását a nagy kör középetől balra irányuló szakasz mutatja. Ez is igazolja, hogy energia fogyasztása a függőleges tengely mentén egyenletes. Mozgó SN nyelő esetén az energia-középpont gyakorlatilag nem mozdul, ami az érdekeltségi területen belüli teljesen egyenletes fogyasztást igazolja. Ugyanakkor az AN csomópontok maradnak tovább működőképesek az NN csomópontokhoz képest és minden csomópont a saját energiaosztályának megfelelően, közel azonos időpillanatban merül le (jobb oldali ábra). Az ábrák jobb felső sarkában található értékek az adott szimulációs esetben lemerülő epoch periódusokat azonosítják: FND (First Node Die) - első csomópont lemerült, HND (Half Node Die) - fele a csomópontoknak lemerül, TQD (Third Quarter Die) - háromnegyede a csomópontoknak lemerült, LNA (Last Node Alive) – már csak egyetlen csomópont maradt. Látható, hogy DS esetén milliós, ENH_LEACH esetén pedig százezres nagyságrendű epoch ideig működik a WSN hálózat.

Az összes ($n = 360$) szimulációs esetben a szintetikus idősorok gyűjtése megtörtént a 4. Táblázatban felsorolt jellemzőkre. DS mechanizmus esetén az utolsó négy jellemző időben konstans, ezért ezeket nem mértük.

4. Táblázat. Mintavételezett szintetikus adatok minden egyes szimuláció esetén

Idősor neve	Jelölés	DS	BAS-LEACH	ENH-LEACH
Maradék energia (%)	E	+	+	+
Shannon entrópia ()	S	+	+	+
Küldött keretek száma (db)	F	+	+	+
Működő csomópontok száma (db)	N	+	+	+
Tranzakciók száma (db/epoch)	T		+	+
Klaszterek száma (db/epoch)	K		+	+
CH-SN átlagos távolság (m/ d_0)	D		+	+
Klaszter átlagos sugár	R		+	+



6. ábra. Húsz darab ENH-LEACH szimuláció négy szintetikus idősorának görbéi epoch idő, illetve százalékos lefolyás függvényében

Mivel a paraméterek hatására a különböző szimulációs esetekben eltérő epoch szám alatt merült le a WSN hálózat, az egyes idősorokat interpoláció segítségével időben azonos hosszúságúra (e_0)

skaláztuk. Mechanizmusonként az idősorok közös hossza az egyes idősorok átlaga lett. Ezután mindegyik idősort normalizáltuk, hogy egymás között összemérhető legyenek. A normalizálás X idősor esetén az alábbi:

$$X' = \frac{X - \text{MEAN}(X)}{\text{STD}(X)} \quad (33)$$

Ez alapján adott szimulációs esetben a WSN rendszerünk által mechanizmusonként választott idősor a következő vektor konkatenációval valósul meg:

$$y_i^{DS} = (E_i^{DS}, S_i^{DS}, F_i^{DS}, N_i^{DS}) \in M_{1 \times 8e_0} \quad (34)$$

$$y_i^{BAS} = (E_i^{BAS}, S_i^{BAS}, F_i^{BAS}, N_i^{BAS}, T_i^{BAS}, K_i^{BAS}, D_i^{BAS}, R_i^{BAS}) \in M_{1 \times 8e_0} \quad (35)$$

$$y_i^{ENH} = (E_i^{ENH}, S_i^{ENH}, F_i^{ENH}, N_i^{ENH}, T_i^{ENH}, K_i^{ENH}, D_i^{ENH}, R_i^{ENH}) \in M_{1 \times 8e_0} \quad (36)$$

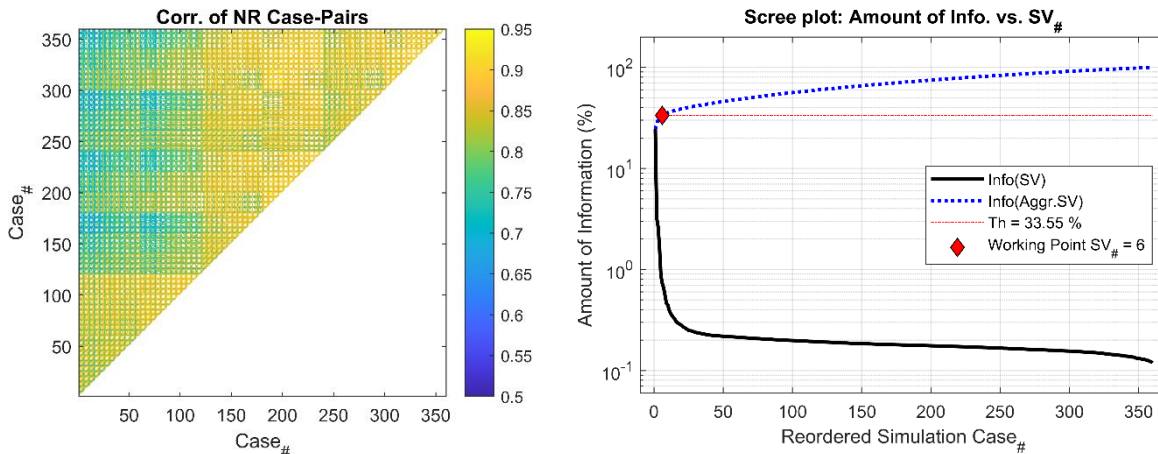
A fenti három sorvektor mindegyike $8e_0$ darab elemet tartalmaz. Mivel DS mechanizmus csak fele anyyi idősort ad, mint BAS vagy ENH, ezért DS részvektorait dupla hosszúságúra skaláztuk. A fenti jelölésekkel azonosított sorvektorokból az WSN rendszer i -edik szimulációjában keletkezett $y_i, i = \underline{1, n}$ válaszát az alábbi oszlopvektor jelenti:

$$y_i = (y_i^{DS}, y_i^{BAS}, y_i^{ENH})^T \in M_{m \times 1} \quad (37)$$

$$Y = (y_1, \dots, y_n) \in M_{m \times n}$$

ahol $m = 24e_0 = 670.872$ és $n = 360$.

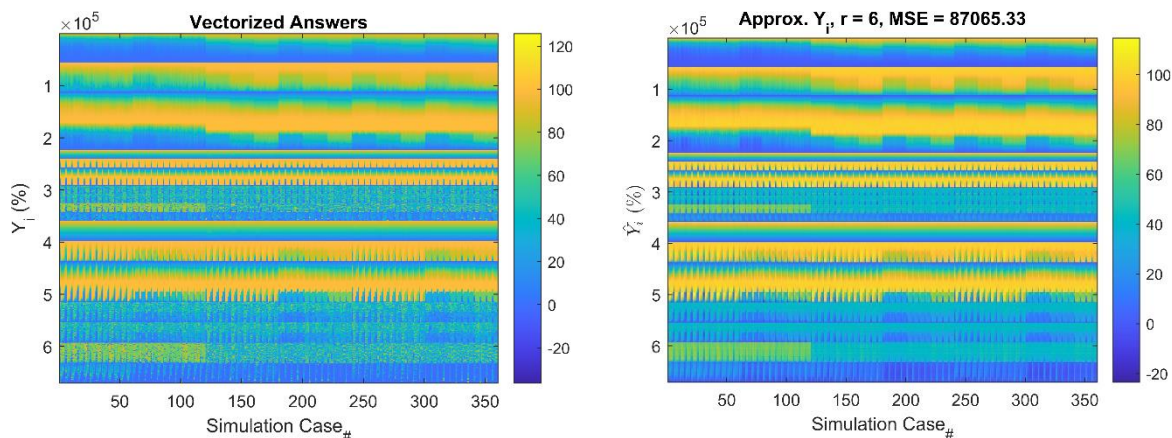
Az Y mátrix mindegyik oszlopa a 4. Táblázat az adott szimuláció mindegyik idősorát tartalmazza, tehát a mátrix összesen $20 \cdot 360 = 7.200$ normalizált idősort foglal magában. Könnyen belátható, hogy a 360 szimuláció összehasonlítása egymás között szabad szemmel nem lehetséges, proceszálást igényel. A 3.1 alfejezetben tárgyalt módon az Y válaszmátrixot szinguláris érték felbontással egyszerűsítjük. A 7. ábra mutatja az Y válaszok C korrelációs mátrix képét, illetve a szinguláris értékek „Scree Plot” ábráját. Mivel C szimmetrikus, ezért csak a főátló alatti részét ábrázoljuk.



7. ábra. WSN rendszer Y válaszainak korrelációs mátrixa (bal), illetve a szinguláris értékek (jobb).

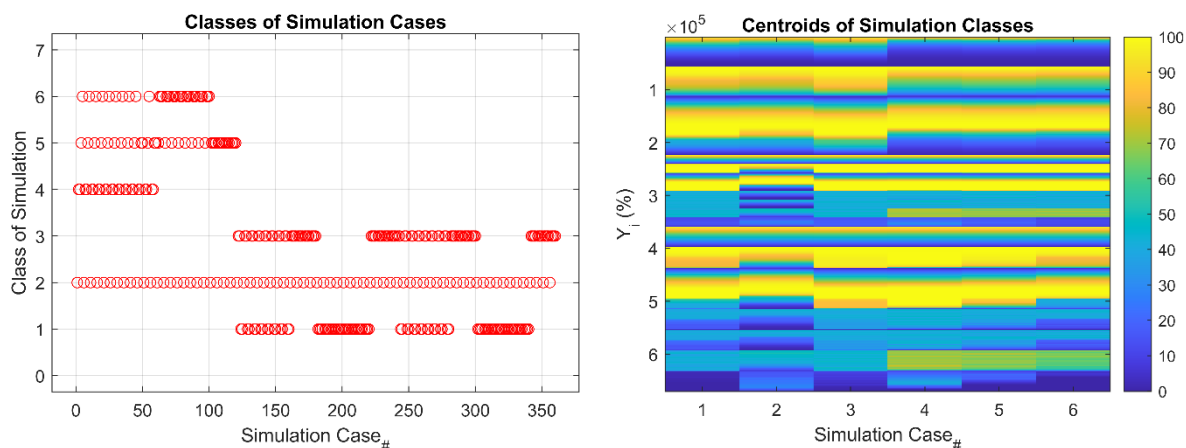
Megállapítható, hogy mivel a szimuláció esetet képviselő y_i oszlopvektor szerkezete (34), (35), (36) és (37) összefüggések szerint rögzített, ezért ez felfogható vektorizált képek. A 360 különböző szimuláció pedig húsz-húsz képbjektumból összerakott, azonos szerkezetű, de egymástól eltérő képek tekinthető. A 7.bal ábra szerint az $y_i, i = \underline{1, n}$ válaszok közötti korreláció a (0,5; 0,95) intervallumban van, zömében közelebb a nagyobb értékhez. Ezt igazolja számszerűen a 7.jobb ábra is, mivel a szinguláris értékek közül $r = 6$ legnagyobb képviseli az $n = 360$ szinguláris érték által képviselt információ 33,55%-át. A „Scree Plot” könyök része igazolja, hogy a 360 szimulációból egy 6 virtuális esetből álló főkomponens bázis képezhető. Tehát létezik 6 darab különböző virtuális (α, p, m, L, g, v) paraméter tenzor érték, ami hat darab olyan virtuális szimulációt képvisel, ami elsősorban jellemzi a CB-LEACH WSN rendszerünket. A többi bázisvektor (virtuális szimuláció eset) közel azonos súllyal vesz részt az eredeti Y mátrix oszlopainak előállításában, ami azt jelenti, hogy a többi szimulációs eset, vagyis a 3. Táblázatban felsorolt (α, p, m, L, g, v) , változó szimulációs paraméter-tenzor a WSN rendszer működését enyhén befolyásolja.

A (25) összefüggés alapján előállított \hat{Y} , egyszerűsített válasz mátrix az elemi idősorokat zajoktól és mellékhatásoktól jelentősen mentesített formában tartalmazza. A 8. ábra erre ad példát eredeti és szűrt esetben. A 8. ábra mindkét része egyenként 7.200 idősort tartalmaz. Az idősorok az oszlopok mentén fentről lefelé haladva tartalmazzák a (37) szerinti idősorokat. A normalizált és skálázott idősorok értékeit a színek adják. Megfigyelhető, hogy a két „kép” elrendezésben és élességben is nagyon hasonlít egymáshoz. Az egyenként $m \cdot n = 241.513.920$ elemből álló két mátrix átlagos négyzetes hibája (MSE: Mean Square Error) $MSE = 87.065,33$, ami elemenként átlagosan $36 \cdot 10^{-5}$, elenyészően kicsi érték. Ugyanúgy a két mátrix közötti keresztkorreláció a $\tau = 0$ esetén nagyon kicsi érték, $-26,67$.



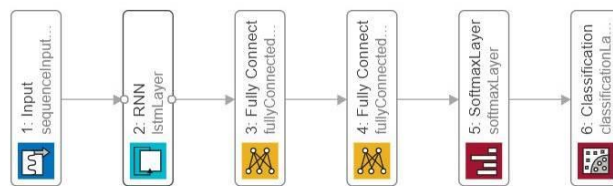
8. ábra. WSN rendszer Y eredeti (bal), illetve \hat{Y} egyszerűsített (jobb) válaszainak egy-egy idősora.

Az eddig megállapított tények a visszacsatolós neurális hálózat létrehozásában, illetve annak segítségével történő elemzéshez hasznosnak bizonyulnak, mivel lecsökkentik a modellezéshez szükséges tanulási folyamatot. „k-Mean clustering” módszerrel r csoportba soroljuk az n darab szimuláció válaszait, azaz $y_i, i = \underline{1}, n$ oszlopvektorokat. Ennek eredményét a 9. ábra szemlélteti. Az r csoportba tartozó szimulációk száma egyenként: 96, 72, 96, 24, 38, 34. Mivel a tagok száma csoportonként azonos nagyságrendű, ezért mindegyik csoport fontos.



9. ábra. Szimulációk osztályokba sorolása (bal), illetve centroid szimulációk (jobb)

A kialakított neurális hálózat a 10. ábrán látható, ahol a visszacsatolást a második blokkban lévő LSTM réteg tartalmazza.



10. ábra. Visszacsatolósos neurális hálózat szerkezete

A rendelkezésre álló 360 válasz mindegyikéről tudjuk, hogy melyik osztályba tartozik. Az y_i válaszok közül 180-at véletlenszerűen elkülönítjük, amit az összeállított neurális hálózat tanításához használunk fel. További 90 darab véletlenszerűen kiválogatott y_i szimulációs válaszvektort a hálózat tanításának validálásához használjuk. A maradék 90 darab y_i válaszvektort a visszacsatolósos neurális hálózat teszteléséhez használjuk. Az RNN tanításához „ADAM” algoritmust alkalmaztunk, a gradiens küszöb módszere pedig az „L2Norm”. A tanítási opciók választott értékét az 5. Táblázat tartalmazza.

5. Táblázat. Az RNN hálózat tanítási opcióinak értéke

Tanítási opció	Érték	Tanítási opció	Érték
GradientDecayFactor	0.9000	GradientThreshold	1
SquaredGradientDecayFactor	0.9990	MaxEpochs	10
InitialLearnRate	0.0200	Number of Classes	6

Az így kapott RNN $\eta = 97,8\%$ pontossággal tudta meghatározni a tesztelésre maradt y_i szimulációs válaszvektorok csoporthoz tartozását. A megoldás viszonylag gyors tanulási folyamat alapján zajlott le, viszont a futtató számítógép memória kapacitása legalább 64 GB kell, hogy legyen.

5. Összefoglalás, következtetések

A dolgozatban bemutatott módszer a kiegyenlítéses, CB-LEACH vezetéknélküli szenzor hálózat viselkedésének elemzését teszi lehetővé. A CB-LEACH a klasszikus LEACH energiahatékony változata, aminek működését hat paraméter alapján lehet befolyásolni. Az optimális paraméter meghatározásához a 360 különböző szimulációval előállított szintetikus állapot adathalmaz elemzése Big Data kategóriájú metódusok használatát tette szükségessé. Ennek érdekében szinguláris érték felbontáson alapuló főkomponens analízist alkalmaztam a legfontosabb szimulációs esetek meghatározásához. Ezek száma $r = 6$, ami nem az (α, p, m, L, g, v) paraméterek száma miatti egybeesés, hanem a CB-LEACH rendszer speciális készségei miatti tulajdonság. A 360 szimulációs esetet r csoportba sorolásához mintázatok beazonosításához alkalmazható, LSTM alapú visszacsatolósos neurális hálózatot használtam fel, amely $\eta = 97,8\%$ pontossággal tudta beazonosítani a véletlenszerűen választott szimulációs válaszvektorok csoporthoz tartozását. A kutatási munka folytatásaként további elemzések szükségesek a válaszvektorok elemi idősorainak zajmentesítésére vonatkozóan a fontosnak tartott szinguláris értékek számának függvényében.

Köszönetnyilvánítás

A kutatást az „Integrált kutatói utánpótlás-képzési program az informatika és számítástudomány diszciplináris területein” (EFOP-3.6.3-VEKOP-16-2017-00002) című projekt támogatta. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg. A kutatást a QoS-HPC-IoT Lab támogatta.

Irodalmi hivatkozások

- [1] Heinzelman, W. B, Chandrakasan, A. P., An application-specific protocol architecture for wireless micro sensor networks, *IEEE Transactions on Wireless Communications* Vol. 1 (2002), 660–670.
- [2] Moh'd Alia, O., Dynamic relocation of mobile base station in wireless sensor networks using a cluster-based harmony search algorithm , *Information Sciences*, (2017), 76–95.
- [3] Kim, J., In, J., Hur, K., Kim, J., & Eom, D., An intelligent agent-based routing structure for mobile sinks in WSNs, *IEEE Transactions on Consumer Electronics* Vol. 4 (2010), 2310–2316.
- [4] Liang, W., Luo, J., & Xu, X., Prolonging network lifetime via a controlled mobile sink in wireless sensor networks, *IEEE global telecommunications conference GLOBECOM* (2010), 1–6.
- [5] Salarian, H., Chin, K.W., & Naghdy, F., An energy-efficient mobile-sink path selection strategy for wireless sensor networks, *IEEE Transactions on Vehicular Technology* (2014), Vol. 63 2407–2419.
- [6] Ismat, N., Qureshi, R., & Mumtaz, I., Efficient Clustering for Mobile Wireless Sensor Networks, *17th IEEE International Multi Topic Conference* (2014), 110–114.
- [7] Zhou, Z., Du, C., Shu, L., Hancke, G., Niu, J., & Ning, J., An Energy-Balanced Heuristic for Mobile Sink Scheduling in Hybrid WSNs, *IEEE Transactions on Industrial Informatics*, Vol. 12 (2016), 28–40.
- [8] Thakur, M., Mobile Sink Based NLEACH Protocol by using Ant Colony Optimization, *International Journal of Science Research and Technology*, Vol. 2 (2016), 1–10.
- [9] Chi, Y., P., Chang, H., P., An energy-aware grid-based routing scheme for wireless sensor networks, *Telecommunication Syst.*, (2013).
- [10] Gal, Z., Korteby, M., A., Energy Sparing of the Leach Communication Mechanism in Heterogeneous WSN, *8th International Conference on Advanced Computer Science and Information Technology*, (2019), 53–64.
- [11] Abo-Zahhad, M., Ahmed, S. M., Sabor, N., & Sasaki, S., Mobile sink-based adaptive immune energy-efficient clustering protocol for improving the lifetime and stability period of wireless sensor networks, *IEEE Sensors Journal*, Vol. 15 (2015), 4576–4586.
- [12] Ghosh, N., & Banerjee, I., An energy-efficient path determination strategy for mobile data collectors in wireless sensor network, *Computers & Electrical Engineering*, Vol. 48 (2015), 417–435.
- [13] Khan, A. W., Abdullah, A. H., Razzaque, M. A., & Bangash, J. I., Vgdra: A virtual gridbased dynamic routes adjustment scheme for mobile sink-based wireless sensor networks, *IEEE Sensors Journal*, Vol. 15 (2015), 526–534.
- [14] Moh'd Alia, O., Dynamic relocation of mobile base station in wireless sensor networks using a cluster-based harmony search algorithm, *Information Sciences*, (2017), 76–95. 178
- [15] Wang, J., Cao, Y., Li, B., Kim, H. J., & Lee, S., Particle swarm optimization based clustering algorithm with mobile sink for WSNs, *Future Generation Computer Systems*, Vol. 76 (2016), 452–457.
- [16] Salarian, H., Chin, K. W., & Naghdy, F., An energy-efficient mobile-sink path selection strategy for wireless sensor networks, *IEEE Transactions on Vehicular Technology*, Vol. 63 (2014), 2407–2419.
- [17] Baljinder, S., Amit, V., Manit, K., A survey on various energy-efficient routing protocols in WSN, *International Journal of Advanced Research, Ideas and Innovations in Technology*, Vol. 4 (2019), 862–865.
- [18] Harjit, S., Varun, S., Energy Efficient Clustering for Network Stability and Longevity for Heterogeneous Wireless Sensor Network, *International Journal of Engineering Science and Computing*, Vol. 8 (2018), 18867–18872.
- [19] Mittal, N., Singh, U., Sohi, B., A stable energy efficient clustering protocol for wireless sensor networks, *Wireless Networks*, Vol. 23 (2017), 1809–1821.
- [20] Preeti, Belwal, M., Energy Efficient LEACH and Improved LEACH: A Review, *International Journal of Advanced Research in Computer Science*, Vol. 10 (2019), 51–52.