

LSTM modellek felhasználása a meghibásodások előrejelzésében

Leveraging LSTM models for failure prediction

HORNYÁK Olivér, PhD

egyetemi docens

Miskolci Egyetem, Gépészmérnöki és Informatikai Kar,
Miskolc-Egyetemváros, +36 46 565 000, oliver.hornyak@uni-miskolc.hu, www.ait.iit.uni-miskolc.hu

Abstract

In today's rapidly evolving industrial landscape, the prevention of machinery and equipment failures is of paramount importance. Accurate and timely failure prediction can not only reduce operational downtime but also save resources and improve overall safety. This article presents an in-depth exploration of the application of Long Short-Term Memory (LSTM) models for failure prediction in industrial settings. The core objective of this research is to harness the power of LSTM neural networks to analyze time-series data and identify patterns that precede equipment failures. We delve into the theoretical foundations of LSTM networks, highlighting their ability to capture long-term dependencies in sequential data, making them well-suited for modeling the dynamics of industrial systems.

Keywords: LSTM, algorithm, classification, Artificial intelligence, machine learning

Kivonat

A mai gyorsan változó ipari környezetben gépek és berendezések meghibásodásainak megelőzése kiemelten fontos. Az pontos és időben történő meghibásodás-előrejelzés nemcsak csökkentheti a működési leállásokat, hanem erőforrásokat is megtakaríthat, és javíthatja az általános biztonságot. Ez a cikk egy részletes vizsgálatot mutat be a Hosszú rövidtávú memória (Long Short-Term Memory, LSTM) modellek alkalmazásáról a meghibásodások előrejelzésére ipari környezetekben. A kutatás alapvető célja a LSTM neurális hálózatok erejének kiaknázása az idősoros adatok elemzéséhez és a meghibásodásokat megelőző mintázatok azonosításához. Elméleti alapjaikba mélyedünk, kiemelve képességüket a hosszú távú függőségek rögzítésére a szekvenciális adatokban, ami kiválóan alkalmas az ipari rendszerek dinamikájának modellezésére.

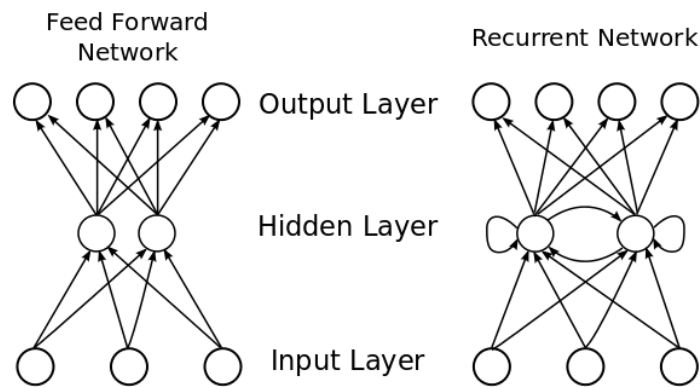
Kulcsszavak: LSTM, algoritmus, osztályozás, mesterséges intelligencia, gépi tanulás

1. BEVEZETÉS

A hosszú rövidtávú memóriával (Long Short-Term Memory, LSTM) [12] rendelkező modellek napjainkban az egyik legizgalmasabb területet képviselik a mélytanulásban és szekvenciaelemzésben. Ezek a rekurzív neurális hálózatok kivételesen hatékonyak a szekvenciális adatok feldolgozásában, és számos alkalmazási területen, mint például a természetes nyelvfeldolgozás, idősorok elemzése, és beszéd felismerés, kiemelkedő eredményeket értek el. Ebben a cikkben részletesen vizsgáljuk a LSTM modellezés alapjait, valamint megvizsgáljuk az alkalmazását meghibásodások előrejelzésében. Emellett bemutatjuk azokat a legújabb fejlesztéseket és kutatási irányokat, amelyek lehetővé teszik az LSTM modellek további javítását és alkalmazkodását az adott problémákhoz és adathalmazokhoz. A cikk célja, hogy áttekintést nyújtson a LSTM modellezés jelenlegi állapotáról, és segítse az olvasót a technológia teljes körű megértésében és sikeres alkalmazásában.

2. AZ LSTM ALGORITMUS

Az LSTM hálózat a Recurrent Neural Network (RNN) [11] egy típusa. Az RNN-ek pedig a neurális hálózatok speciális típusai sorrendi problémákra. Adott egy előrecsatolt MLP hálózat, tekintjük úgy az RNN-t, mintha hurkok kerülnének az architektúrába (lásd 1. ábra). Például egy adott rétegben minden neuronnak lehetősége van a jelét oldalirányban átadnia, emellett továbbítja azt a következő rétegre.



1. ábra. Az előrecsatolt hálózat és a visszacsatolt hálózat különbsége

A hálózat kimenete visszacsatolást ad a következő bemeneti szintnek. Az ismétlődő kapcsolatok állapotot vagy memóriát adnak a hálózathoz, és lehetővé teszik a tanulást és kihasználják a megfigyelések rendezettségét a bemeneti szekvenciákban. Egy új dimenziót jelent, hogy a közelítendő függvényben megjelenik az időbeliség, a szekvencia. A hálózat képes megtanulni egy leképezési funkciót a bemenetek idejét is figyelembe véve egy adott kimenetre. A belső memória azt jelenti, hogy a kimenetek függvényei bemeneti szekvenciák belső kontextusának is, nem csak magának az inputnak. Bizonyos értelemben ez a képesség az, ami lehetővé teszi a szekvencia előrejelző képességet a neurális hálózatok számára. A Long Short-Term Memory sok idősoros feladatot képes megoldani, ami megoldhatatlan előrecsatolt hálózatok által használt rögzített méretű időablakok használatával.

A neurális hálózatok szekvencia-előrejelzésre való használatának általános előnyei mellett az RNN megtanulhatja és kihasználhatja az adatok időbeli függőségét. Vagyis a legegyszerűbb esetben a hálózat egy-egy megfigyelést jelenít meg egy sorozatból, és meg tudja tanulni, hogy a korábban látott megfigyelések közül melyek relevánsak, és hogy mennyire relevánsak az előrejelzés elkészítéséhez. A hosszú távú összefüggések szekvenciákból való tanulásának képessége miatt az LSTM hálózatoknak nincs szükség előre meghatározott időablakra, és képesek komplex többváltozós sorozatok modellezése [3].

A visszatérő neurális hálózatok ígérete az, hogy az időbeli függőség és a kontextuális a bemeneti adatokban lévő információk megtanulhatók. A visszatérő hálózat, amelynek bemenetei nem rögzítettek, hanem bemenetet szekvenciát képeznek használható arra, hogy egy bemeneti sorozatot kimeneti szekvenciává alakítson, miközben rugalmas módon a kontextuális információkat figyelembe veszi [4].

Számos RNN létezik, de az LSTM az, amely beváltja az RNN-ek ígéretét szekvencia előrejelzése. Ez az oka annak, hogy jelenleg olyan széles körben elterjedt az LSTM-ek alkalmazása. Az LSTM-eknek belső állapotuk van, kifejezetten tudatában vannak a bemenetek időbeli szerkezetének, több párhuzamos bemeneti sorozatot külön-külön modellezhetnek, és különböző hosszúságúak lehetnek bemeneti szekvenciák a változó hosszúságú kimeneti szekvenciák előállításához.

2.1. Long Short-Term Memory

Az LSTM hálózat különbözik a klasszikus MLP-től. Az MLP rövidítés az angol „Multilayer Perceptron” kifejezésre utal [10], és magyarul többretegű perceptron néven ismert. Az MLP-hez hasonlóan az LSTM hálózat is rétegekbe szervezett neuronokból áll. A bemeneti adatokat a hálózaton keresztül továbbítják az input neuronoktól az output neuronok irányába. Az RNN-ekhez hasonlóan az LSTM-eknek is van visszatérő kapcsolata, így a korábbi aktiválások állapota része a modellnek. Az előző időlépésből származó neuron kontextusként szolgál egy kimenet megfogalmazásához. De a többi RNN-től eltérően az LSTM olyan egyedülálló formával rendelkezik, amely kiküszöböli az egyéb RNN hálózatok betanítását és skálázását gátló tényezőket. A legfontosabb technikai kihívás, amellyel korábban az RNN szembenézett, az, hogyan lehet hatékonyan tanítani azokat. A problémák két véglete az volt, hogy a súlyfrissítési eljárás gyorsan kis súly változást eredményezett (eltűnő gradiensek), vagy olyan nagyot, hogy az eljárás hamar véget ért (robbanó gradiensek). Az LSTM-ek struktúrájuknál fogva ezt megoldják. Az LSTM egy RNN architektúra, amelyet kifejezetten arra terveztek kezelni az eltűnő gradiens problémát.

Sajnos, a kontextuális információk széles köre, amelyhez a szabványos RNN-ek hozzáférhetnek a gyakorlatban meglehetősen korlátozott. A probléma az, hogy egy adott bemenet hatása a rejtett rétegen, és ezért a hálózati kimeneten vagy lecseng, vagy exponenciálisan emelkedik, miközben körbejárja a hálózat visszatérő kapcsolatait. Ez hiányosság a szakirodalomban az eltűnő gradiens problémaként emlegetett [5]. Az LSTM hálózat számítási egységét memóriacellának, memóriablokknak, illetve röviden csak sejtnek

nevezük. A neuron kifejezés, mint számítási egység, korábban annyira rögzült a, hogy az MLP-k esetén is gyakran használják az LSTM memóriacellára való hivatkozás szinimájaként. Az LSTM sejtek súlyokból és kapukból állnak. A Long Short Term Memory architektúrát a meglévő RNN-ek hibaterjedésének elemzése motiválta, amely megállapította, hogy a hosszú időközések nem érhetők el a meglévő RNN-ekben architektúrális okokból, mert a visszaterjesztett hiba vagy exponenciálisan növekszik (felrobban), vagy exponenciálisan csökken (eltűnik).

Az LSTM réteg ismétlődően összekapcsolt blokkokból áll, amelyek memóriablokkokként ismertek. Ezeket a blokkokat a digitális számítógépben lévő memóriachipek egy változatának tekinthetjük. Mindegyik tartalmaz egy vagy több ismétlődően csatlakoztatott memóriacellát és három szorzó egységet - a bemeneti, a kimeneti és a felejtő kapukat - amelyek írási, olvasási és alaphelyzetbe állítási műveletek analógiájaként tekinthetők a cellák szintjén. A hálózat csak a kapukon keresztül tud kölcsönhatásba lépni a cellákkal [5].

2.1. LSTM elemi

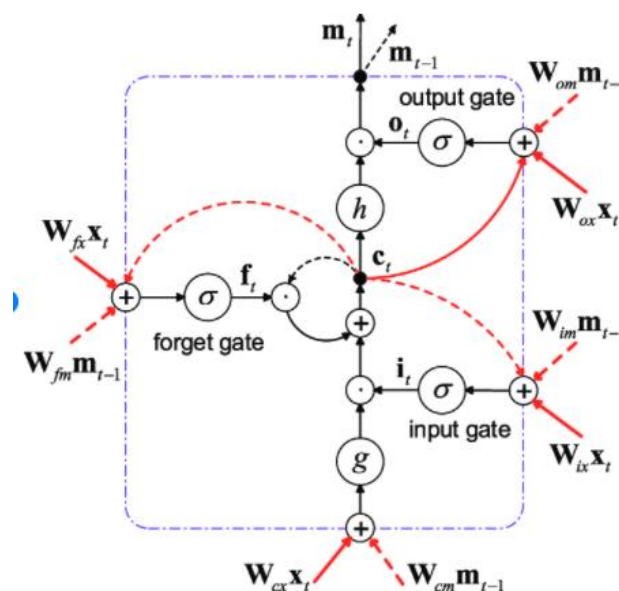
A memóriacella súlyparaméterekkel rendelkezik a bemenethez, a kimenethez, valamint a belső állapotához, amely a bemeneti időlépések hatására épül fel. Ezek lehetnek:

- bemeneti súlyok: az input súlya adott időlépésben
- kimeneti súlyok: az utolsó időlépés óta a kimenet súlya
- belső állapot súlya: milyen mértékben használjuk a belső állapotot a következő output kiszámolására.

A memóriacellákhoz a kapcsolódik az LSTM kapu. Ezek is súlyozott funkciók, amelyek továbbirányítják az információáramlást a cellában. Három kapu van:

- forget gate: eldönti, hogy milyen információkat kell elfelejteni (elvetni a cellából).
- input gate: meghatározza, hogy a bemenet mely értékei frissítik a memória állapotát.
- output gate: a bemenet és a cella memóriája alapján dönti el, hogy mit adjon ki.

A felejtési kapu és a bemeneti kapu a belső állapot frissítésére szolgál. A kimeneti kapu korlátozza azt, amit a cella ténylegesen kiad. A hagyományos MLP neuronokkal ellentétben nehéz tisztán lerajzolni egy LSTM memóriaegységet. Mindenütt kapcsolatok, súlyok és kapuk vannak. A következő ábra bemutatja a kapukat [6]:



2. ábra. LSTM kapuk

2.2. LSTM előnyei és hátrányai

Az LSTM-ek három fő előnyét a következőképpen foglalhatjuk össze:

- Leküzd az RNN betanításának technikai problémáit, nevezetesen az eltűnő és felrobbanó gradienseket.

- Rendelkezik memóriával a bemeneti szekvenciákkal való hosszú távú időbeli függőség problémáinak leküzdésére.
- A bemeneti és kimeneti szekvenciákat időlépésről időlépésre dolgozza fel, lehetővé téve a változó hosszúságú be- és kimeneteket.

LSTM-ek nem feltétlenül ideálisak minden szekvencia-előrejelzési problémára. Például az idősoros előrejelzésben gyakran az előrejelzés elkészítéséhez releváns információk a múltbeli megfigyelések egy kis ablakán belül vannak. Ilyenkor egy MLP ablakkal vagy lineáris modellel egy kevésbé összetett és alkalmasabb modell lehet. A szakirodalomban található idősoros benchmark problémák általában fogalmilag egyszerűbbek, mint sok, az LSTM által már megoldott feladat. Gyakran egyáltalán nem igényelnek RNN-t, mivel a következő eseményre vonatkozó összes releváns információt néhány közelmúltbeli esemény tárolja egy kis időablakon belül.

Az LSTM-ek fontos korlátja a memória. Pontosabban, hogyan lehet visszaélni a memóriával. Lehetőség van arra, hogy egy LSTM modellt arra kényszerítsenek, hogy egyetlen megfigyelésre emlékezzen nagyon hosszú számú bemeneti időlépésen keresztül. Ez az LSTM-ek rossz használata. [10] rávilágít arra, hogy az időablak alapú MLP felülmúlta az LSTM tiszta autoregressziós megközelítést bizonyos idősor-előrejelzési benchmarkokon, amelyek csak néhány közelmúltbeli bemenet figyelembevételével oldhatók meg. Így nem volt szükség az LSTM különleges erejére, nevezetesen arra, hogy megtanuljon emlékezni az egyes eseményekre nagyon hosszú, ismeretlen időszakokra.

A „constant error carousel” (CRC) [13] egy olyan probléma, amely az LSTM hálózatokban előfordulhat, különösen akkor, amikor a hálózat hosszú szekvenciákat vagy idősorokat dolgoz fel. Ez a hosszú távú függőségek megőrzésének nehézségéből ered. Az LSTM hálózatok hosszú távú információkat nehezen tudnak megőrizni amiatt, hogy hibásan elfelejtik vagy torzítják azokat. A CRC probléma lényege, hogy az LSTM hálózat hibái hosszú távon felhalmozódhatnak és ismétlődhetnek, ahelyett, hogy elhalványulnának vagy javulnának az idő múlásával. Ennek eredményeként az LSTM hálózat folyamatosan hibázik ugyanazon hibák vagy torzítások sorozatát produkálva, ami akadályozza a helyes predikciókat és szekvenciaelemzést. A problémát a megfelelő hiperparaméterek beállításával és az LSTM hálózat megfelelő méretének kiválasztásával lehet enyhíteni vagy megoldani. Az ilyen hálózatok mélyítése, a tanulási sebesség beállítása és az adatok megfelelő előkészítése mind hozzájárulhatnak a CRC hatásának csökkentéséhez. Emellett egyes esetekben más RNN típusok, például a GRU (Gated Recurrent Unit) vagy a kettős LSTM, hatékonyabbnak bizonyulhatnak a hosszú távú függőségek kezelésében.

LSTM modelleket széles körben alkalmazzák a gépi tanulásban és a mély tanulásban, és számos területen kiváló eredményeket értek el, például gépi fordítás, érzelmi analízis, képalkotás, és idősorok előrejelzése terén.

3. LSTM ALAPÚ MEGHIBÁSODÁS ELŐREJELZŐ MODELL

Az LSTM alapú meghibásodás előrejelző modell egy olyan gépi tanulási modell, amely célja a műszaki berendezésekben, rendszerekben előforduló meghibásodások előrejelzése és megelőzése. [8] részletes leírást ad a műszaki felépítésről. Az alábbiakban felvázolom a lépéseket egy LSTM modell létrehozásához:

1. Adathalmaz összeállítása:

Első lépésként össze kell állítani egy megfelelő adathalmazt, amely tartalmazza a rendszer működési adatait és az esetleges meghibásodásokat jelző jeleket vagy címkéket. Az adathalmaznak megfelelő időrendi információkat kell tartalmaznia a meghibásodások előtt és után, valamint a rendszer paramétereit és mérhető jeleit.

2. Adatok előkészítése:

Az adatokat tisztítani és normalizálni kell, hogy a modell számára megfelelő formátumban legyenek. Emellett a bemeneti és kimeneti változókat megfelelően el kell különíteni.

3. Szekvenciális adatok előkészítése:

Mivel a meghibásodások előrejelzése idősorok elemzésére épül, fontos lehet a szekvenciális adatok előkészítése és részletezése, például időablakok létrehozása és azokhoz tartozó bemeneti és kimeneti értékek kiválasztása.

4. LSTM modell kialakítása:

Az LSTM hálózatot a meghibásodások előrejelzésére használjuk. A bemeneti rétegek a rendszer paramétereit, állapotát jelenti, míg a kimeneti réteg az előrejelzési eredményeket adja. Középen található a LSTM rétegek, amelyek képesek a hosszú távú függőségeket kezelni és idősorokat modellezni.

5. Tanítás és validáció:

Az adathalmazt fel kell osztani tanító- és validációs részekre. A modellt a tanító adathalmazon kell tanítani, majd a validációs halmazon kell értékelni. A hibák alapján finomhangolhatjuk a modell paramétereit.

6. Hiperparaméterek finomhangolása:

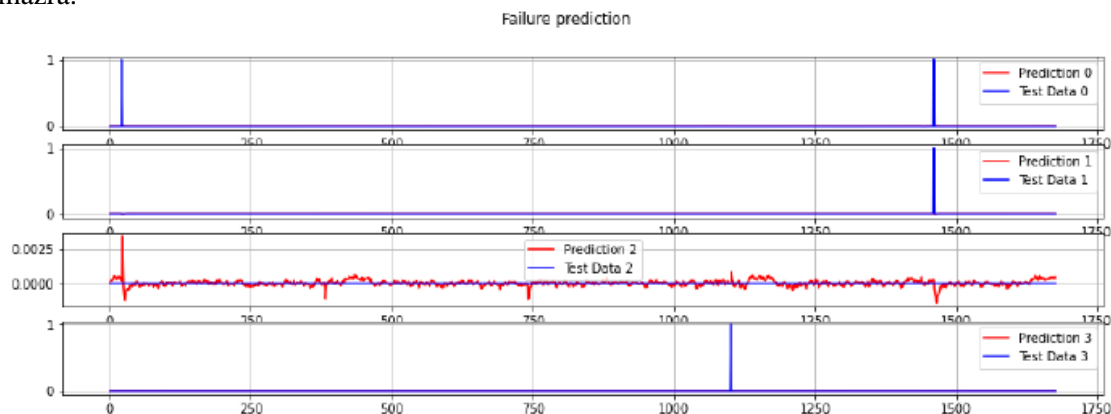
A LSTM hálózatnak számos hiperparamétere van, mint például a rejtett rétegek száma, az epizódok száma, a tanulási sebesség stb. Ezeket finomhangolva lehet maximalizálni a modell teljesítményét.

7. Tesztelés és monitorozás:

A modellt a valós időben működő rendszerre lehet alkalmazni a meghibásodások előrejelzésére. A modell folyamatosan figyelheti a rendszer állapotát és a bemeneti adatokat, és amikor meghibásodást észlel, előrejelezheti azt.

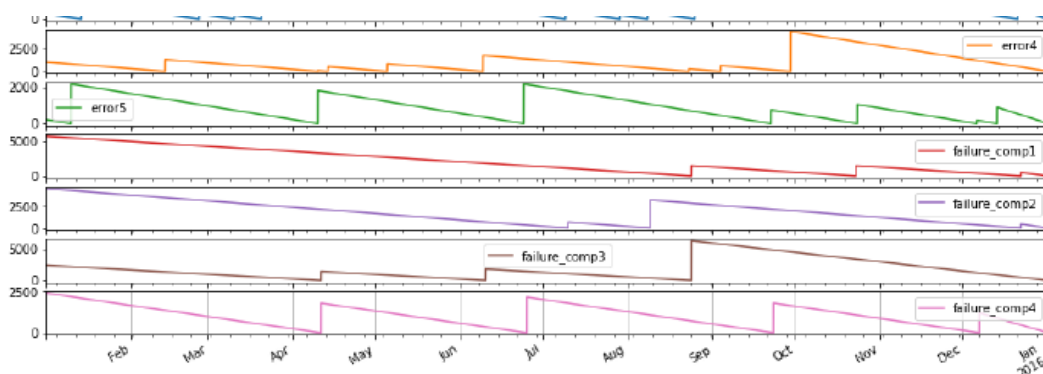
4. A MODELL FUTTATÁSÁNAK EREDMÉNYEI

A vizsgálathoz tartozó adathalmazt [8] először binárisan tekintettük. Ebben 1 jelentette a meghibásodást, 0 a helyes működést. Az alábbi ábrán a piros vonal mutatja a modell által előrejelzett hibákat, a kék vonalak a tesztadatokat tartalmazzák. Az LSTM hálózatok a tanulási folyamat során az adatok statisztikáját veszik figyelembe, és időbeli mintázatokat próbálnak kinyerni. Bináris adatok esetén a statisztikák és a mintázatok jelentősen eltérhetnek a folytonos vagy idősoros adatoktól, ami megnehezíti az LSTM hálózatoknak a helyes modellezést. Az LSTM hálózatok hajlamosak az overfittingre, különösen kisebb méretű adathalmazok esetén. Bináris adatok esetén, ahol a címkék korlátozottak lehetnek (például 0 vagy 1), az overfitting könnyen előfordulhat, és a modell hajlamos lehet túlzottan specializálódni az adathalmazra.



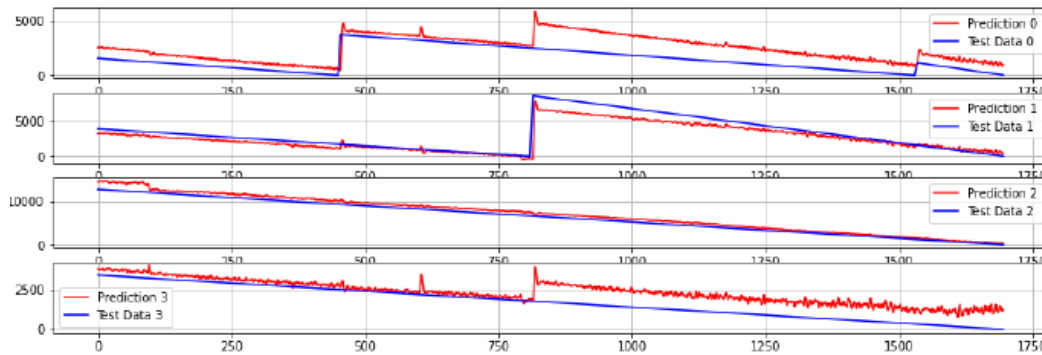
3. ábra. Bináris LSTM modell

A modellt ezután átalakítottuk fűrészfog-szerűen. Ez a bináris meghibásodási adatokból a lineárisan közelített maradó élettartamra transzformálja a problémát (Remaining Useful Life, RUL) [9].



4. ábra. RUL modell

Az előrejelzést így már nagyobb pontossággal tudtuk előállítani.



5. ábra. RUL modell pontossága

5. ÖSSZEFOGLALÁS

A cikkben áttekintettem, az LSTM modellek struktúráját. Felépítettem egy LSTM alapú tanuló modellt meghibásodások előrejelzésre. A modell bináris adatok előrejelzésére nem adott kielégítő eredményt. Az adathalmazt átranzformáltam a maradó élettartamra, így az előrejelzés megfelelő pontosságú lett.

KÖSZÖNETNYILVÁNÍTÁS

A kutatás a Nemzeti Kutatási Fejlesztési és Innovációs Hivatal 2020-1.1.2-PIACI-KFI-2020-00147 pályázati támogatásával valósult meg. A Szerző köszöni a támogatást.

IRODALMI HIVATKOZÁSOK

- [1] Graves, A. *Generating sequences with recurrent neural networks*. arXiv preprint arXiv:1308.0850 (2013).
- [2] Sutskever, I. Oriol V., and Quoc V. L. *Sequence to sequence learning with neural networks*. Advances in neural information processing systems. 2014.
- [3] Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015, April). *Long Short Term Memory Networks for Anomaly Detection in Time Series*. In Esann (Vol. 2015, p. 89).
- [4] Bengio, Y., Simard, P., and Frasconi, P. *Learning long-term dependencies with gradient descent is difficult*. IEEE transactions on neural networks 5.2 (1994): 157-166.
- [5] Graves, A., Schmidhuber, J. *Framewise phoneme classification with bidirectional LSTM and other neural network architectures*. Neural networks 18.5-6 (2005): 602-610.
- [6] Cai, M., & Liu, J.. *Maxout neurons for deep convolutional and LSTM neural networks in speech recognition*. Speech Communication, 77, 53-64. 2016
- [7] Gers, Felix A., Eck, D., Schmidhuber, J. *Applying LSTM to time series predictable through time-window approaches*. Neural Nets WIRN Vietri-01. Springer, London, 2002. 193-200
- [8] Hornyák, O., Iantovics, L. B. *AdaBoost Algorithm Could Lead to Weak Results for Data with Certain Characteristics*. Mathematics, 11(8), 1801. 2023.
- [9] Liu, K., Shang, Y., Quan, O., & Widanage, W.D. (2021). *A Data-Driven Approach With Uncertainty Quantification for Predicting Future Capacities and Remaining Useful Life of Lithium-ion Battery*. IEEE Transactions on Industrial Electronics, 68, 3170-3180.
- [10] Murtagh, F. (1991). *Multilayer perceptrons for classification and regression*. Neurocomputing, 2(5-6), 183-197.
- [11] Medsker, L. R., & Jain, L. C. (2001). *Recurrent neural networks*. Design and Applications, 5(64-67), 2.
- [12] Sherstinsky, A. (2020). *Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network*. Physica D: Nonlinear Phenomena, 404, 132306.
- [13] Hochreiter, S., & Schmidhuber, J. (1997). *Long short-term memory*. Neural computation, 9(8), 1735-1780.
- [14]