

Zsenák-Simon Orsolya,¹ Zsenák István²

A biztonságtechnikában is alkalmazható gépi karakterfelismerés különböző mesterségesintelligencia-módszerekkel

*Computerized Character Recognition Applicable
in Security Technology Using Various
Artificial Intelligence Methodologies*

Számos cikk született már annak vizsgálatára, hogy hogyan ismerheti fel egy algoritmus a képformátumon elhelyezkedő alakzatokat. Az általunk elvégzett vizsgálathoz Hopfield neurális hálózatok algoritmusát használtuk, és az eredményeket összevetettük egy fuzzy logikával működő eljárás eredményével. Ezek az algoritmusok külön-külön is alkalmasak a karakterfelismerésre, azonban érdekes eredményeket kapunk, ha összevetjük az algoritmusok által kapott eredményeket egymással.

A nemzetbiztonság számos aspektusából kulcsfontosságú, hogy tisztában legyünk a rendelkezésre álló mesterségesintelligencia-módszertanok képességeivel. A gépi karakterfelismerés több ponton kapcsolódik a közigazgatás és a közbiztonság tárgykörébe, így fontos ismernünk, hogy milyen programozási technológiák, milyen hatékonysággal tudják elvégezni ezt a feladatot, ugyanakkor az e tanulmányban bemutatott összehasonlítás csupán egy a sok közül, amelyet a mesterséges intelligencia témakörében el kellene végezni.

Kulcsszavak: mesterséges intelligencia, biztonság, kiberbiztonság, karakterfelismerés, fuzzy, Hopfield-háló

Several articles have been written on how an algorithm can recognize shapes placed within an image format. In our study, we utilized the algorithm of Hopfield neural networks and compared the results with a procedure based on Fuzzy logic. While these algorithms are individually suitable for character recognition, intriguing findings emerge when we compare the outcomes obtained by each algorithm.

¹ Pécsi Tudományegyetem Általános Orvostudományi Kar, e-mail: orsolya.simon1@gmail.com

² Pécsi Tudományegyetem Egészségtudományi Kar, e-mail: istvan.zsenak@etk.pte.hu

Understanding the capabilities of available artificial intelligence methodologies is crucial from various aspects of national security. Machine character recognition intersects with both public administration and public safety, hence it's important to comprehend which programming technologies can efficiently perform this task. Nevertheless, the comparison presented in this study is just one of many that should be conducted within the field of artificial intelligence.

Keywords: Artificial intelligence, security, cybersecurity, character recognition, fuzzy, Hopfield network

Bevezetés

A vizsgálat célja

Jelen tanulmányban azt vizsgáltuk, hogy milyen eredménnyel képes egy tudásalapú és egy adatalapú mesterséges intelligencia ugyanazt a képi mintavételezési feladatot ellátni. Elvégeztük az irodalomkutatást, de nem találtunk olyan tanulmányt, amely összehasonlítaná a fuzzy logika és a Hopfield neurális háló hatékonyságát karakterfelismerés esetén. Az egyes irodalmak eltérő hardveres és szoftveres környezetben végezték el a kutatásokat, így ezekből nem tudunk biztos konzekvenciát levonni a két eljárás összehasonlításához.

A kísérlet részleteit a Módszer fejezetben taglaljuk. Vizsgálatunkhoz egy 2020-as Apple MacBook Pro-t használtunk, amely 8GB RAM-mal és egy 8 magos Apple M1 chip-pel rendelkezik. A szoftvert, amelyben a teszteket elvégeztük, PHP 8.2. szkriptnyelven írtuk, amelynek futtatásához MAMP-szervert használtunk. A kísérlet során a fuzzy logika alkalmazásakor empirikus vizsgálatok után határoztuk meg a döntéshozatalhoz szükséges fuzzy függvényeket, míg a neurális háló tanítása során 20 mintavételezést végeztünk a rendelkezésre álló képi minták közül. A beállítást és tanítást követően 100-100 alkalommal futtattuk le a szoftvert, és elemeztük az eredményeket, amelyek részleteit a Módszer és az Eredmények fejezet taglalja.

Neurális háló

A neurális hálózatokat ma számos tudományterület alkalmazza képfelismerésre, képfeldolgozásra, ezáltal kódolás-dekódolásra, vagy akár bioinformatikai adatok elemzésére. John J. Hopfield 1982-ben fejlesztette ki modelljét, amely az agy neurális hálózatát vizsgálja, miszerint hogyan kapcsolódnak egyes neuronok egymáshoz, hogyan képesek az információátvitelre, ezen gondolat mentén pedig, hogyan képes az agy előhívni emlékeket a memóriából. Minden egyes alkalommal, amikor egy ismerős zenével vagy illattal találkozunk, azt a memóriából hívjuk elő ezeknek a neuronoknak a segítségével, mivel az agyunk már megtanulta azt az illatot, azt a hangot. Ez a felfedezés több tudományterületre is hatással volt, mint például a biológiára, de a számítástechnika tudományára is. Megismervén az agy neurális hálózatának felépítését, az informatikai szakemberek

a számítógépek neurális hálózatát kezdték építeni, kísérletet téve, hogy a gépeket megtanítsák a memóriájuk használatával tanulni.³

A neurális hálózatok működése az emberi agy működésén alapul. Az agyban neuronok hálózata található, amelyek elektromos jelek segítségével kommunikálnak egymással. A mesterséges neurális hálózatok ezen a mintán alapulnak, és mesterséges neuronokból állnak, amelyek matematikai függvények segítségével hoznak döntést.

A tanulás során a neurális hálózat gráfként képzelhető el, amelynek élei a neuronok közötti kapcsolat súlyait hivatottak reprezentálni. A bemeneti adatok alapján a kívánt kimenetet produkálják. A képfelismerés során a neurális hálózat bemenete a kép, a kimenete pedig jellemzően a kép osztálya. Az osztályokat a neurális háló trénelése előtt kell definiálni, és a háló a tanulás után képes lesz meghatározni, hogy a vizsgált kép mely osztály képeivel mutatja a legszorosabb kapcsolatot.⁴

A neurális hálózatok általában véve hatékony eszközök a képfelismeréshez. Különböző típusú neurális hálózatokat lehet használni erre a feladatra (Hopfield neurális háló, mesterséges neurális háló, mély neurális háló), és törekedni kell a leghatékonyabb eljárás kiválasztására, amely mindig a feladattól függ. A neurális hálózatok előnyei közé tartozik a nagy pontosság, a robusztusság és az alkalmazkodóképesség. E tulajdonságok miatt a neurális hálózat kimondottan alkalmas változó környezetben történő működésre, imitálva az emberi tanulás folyamatát. Működése közben a meglévő adathalmazra támaszkodik, és programozása viszonylag könnyű, ami miatt széles körben biztosított a humán erőforrás az ilyen típusú mesterséges intelligenciák fejlesztésére. A neurális hálózat hátrányai közé sorolhatjuk a magas számítási igényt, ami megnehezíti a kisméretű, kompakt rendszerekbe történő integrációját. Hiába az utóbbi évek nagy népszerűségi hulláma, az adatalapú mesterséges intelligenciák alkalmazásához továbbra is speciális chipek, komplex architektúra szükséges. Ennek egyszerűsítésére jelenleg nem látszik kellő erőfeszítés a nagyobb gyártók és szolgáltatók irányából. A másik komoly hátrány a tanításhoz szükséges nagy mennyiségű adatigény. Amennyiben ez nem áll rendelkezésre, úgy a feladat elvégzésére nem tudunk adatalapú mesterséges intelligenciát alkalmazni.⁵

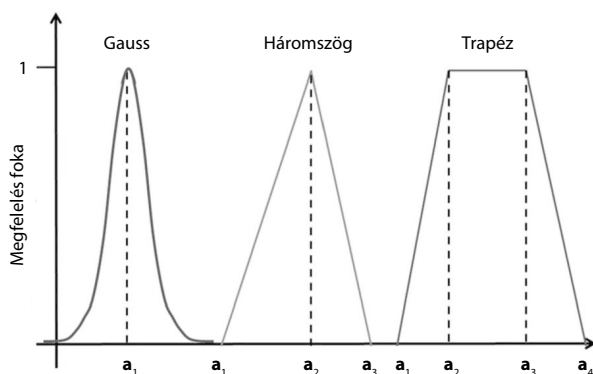
Fuzzy logika

A fuzzy logika az elmosódott halmazok modelljét használja egy probléma leírására. A halmazokat matematikai függvények szimbolizálják. A leggyakrabban háromszög-, trapéz- és Gauss-függvényeket alkalmaznak, de bármilyen saját magunk által meghatározott, a rendszer működését legjobban leíró függvényt is alkalmazhatunk a fuzzy logika szabályainak meghozatalakor. A háromszögfüggvény a legegyszerűbb fuzzy függvény, egy egyenes vonallal ábrázolható. Három paramétere van: a minimum, a maximum és a csúcserték. A trapézfüggvény szintén egyenes vonallal ábrázolható, de a háromszögfüggvényhez képest két csúcserték van. A trapézfüggvénynek négy paramétere van: a minimum, a maximum, az alsó csúcserték és a felső csúcserték. A Gauss-függvény harang alakú

³ The Franklin Institute [é. n.].

⁴ OLUARE et al. 2018.

⁵ OLUARE et al. 2018.



1. ábra: A leggyakoribb fuzzy függvények

Forrás: GONZÁLEZ et al. 2018

görbével ábrázolható, amelynek két paramétere van: a csúcserték és a szórás. Értékei e két paraméterből képződnek a Gauss-féle normális eloszlás szabályai szerint, azaz a görbe csúcsa az átlagértéket (μ) jelöli, és az eloszlás szélessége a szórás (σ) mértékét mutatja.⁶ A fent említett 3 függvény alakjait az 1. ábra szemlélteti.

A fuzzy logika legfontosabb eltérése a klasszikus matematikai halmazelmélethez képest, hogy a halmazok közötti határ nem éles, hanem egy elem valamennyire az egyik és valamennyire a másik halmazba is tartozhat. Ennélfogva a halmazelméleti műveletek, mint például az unió, metszet és különbség más módon, valamilyen függvénytranszformációs eljárás hatására állnak elő. A víz hőmérsékletének osztályozása egy élesen definiált szabályalapú rendszerben úgy történik, hogy meghatározzuk, hány foktól hány fokig tekintjük hidegnek, langyosnak és melegnek a vizet. A fuzzy logikában azt mondhatjuk, hogy a víz bizonyos hőmérsékleten hideg, bizonyos hőmérsékleten pedig valamennyire hideg és valamennyire langyos, illetve valamennyire meleg. A halmazba tartozás mértékét a függvények szélsőértékei és meredeksége határozza meg. A fuzzy logika kivételesen alkalmas a bizonytalan rendszerek leírására, ezért előszeretettel alkalmazzák vezérlés-technikában és mintavételezésben. Fontos különbség a neurális hálózhoz képest, hogy a fuzzy egy tudásalapú mesterséges intelligencia, tehát a tanítását nem nagy mennyiségű adatból végezzük, hanem az emberi tudás és tapasztalat által diktált függvényekkel, következésképp nincs szükségünk akkora adathalmazra, mint az adatlapú mesterséges intelligenciák tanításához. Az érintett rendszer működéséről szerzett tapasztalatok, tudáselemek megjelennek a rendszer *input* és *output* változóit leíró függvények értékeiben.⁷

Módszer

Az összehasonlítást a Hopfield neurális háló és a fuzzy logika között végeztük el. Mindkét esetben a működéshez szükséges minimális lépésszámú algoritmust programoztuk le.

⁶ LIMA et al. 2021.

⁷ GONZÁLEZ et al. 2018.

Az összehasonlítás során két indikátort vizsgáltunk, a karakterfelismerés során a vizsgálandó karakter egyezését a betanított karakterrel és az algoritmus lefutásának idejét.

Hopfield neurális háló

Az alkalmazás betanításához 11 betűtípust használtunk, amelyeket véletlenszerűen választottunk ki az Apple MacBook Pro telepített betűtípusai közül. Mivel kizárólag a betanítás során keletkező információra kívántunk támaszkodni, fontosnak tartottuk, hogy semmilyen előzetes ismeretünk vagy predesztinációnk ne legyen a betűtípusokat illetően, ezért a kiválasztást egy program végezte, és azonnal átnevezte a kijelölt betűtípusokat.

A PHP képgeneráló függvényeinek használatával kirajzoltunk egy C betűt a képernyőre, majd egy dupla ciklussal minden pixel színét megvizsgáltuk. Ahol eltérés volt a háttérszíntől, ott 1-est, ahol a háttérszint érzékeltük, ott -1-est írtunk egy egydimenziós tömb következő indexére. Az 50 pixel széles és 50 pixel magas képünkben így egy 2500 elemű egydimenziós tömb készült. Ezután a Hopfield neurális háló betanítási eljárásának megfelelően a tömb minden i -edik elemét megszoroztuk a tömb minden elemével, így egy 2500×2500 elemű mátrixot kaptunk. Az eljárásnak megfelelően a főátló elemeit kinulláztuk. Ezt az eljárást összesen 20-szor végeztük el, és az így elkészült 20 darab mátrixot a mátrixok összeadásának szabálya szerint adtuk össze. Az eredményt tartalmazó 2500×2500 -as mátrixot egy csv fájlba mentettük el.

A betanítás után összesen 100-szor végeztünk vizsgálatot C betűvel, és szintén 100-szor az O betűvel. A vizsgálati eljárás szintén a minimális lépésszám figyelembevételével készült. A fájlba mentett mátrixot beolvastuk a memóriába, majd a Hopfield neurális háló vizsgálati módszerének megfelelően a vizsgálni kívánt, a PHP beépített függvényeivel kirajzolt betűt tartalmazó kép pixeleiből ismét előállítottuk a 2500 elemű egydimenziós tömböt. A beolvasott mátrixot a mátrix és vektor szorzására vonatkozó szabály szerint szoroztuk meg az egydimenziós tömbbel. Az eredményként kapott egydimenziós tömbben minden negatív értéket -1-gyel, minden nem negatív értéket 1-gyel helyettesítettünk. Ezután megvizsgáltuk, hogy a vizsgálatban kirajzolt betűből képzett egydimenziós tömb és az eredményként elkészült egydimenziós tömb elemei milyen arányban egyeznek. A fájlbeolvasás utáni első lépéstől az összehasonlításig mértük az időt.

Fuzzy logika

A fuzzy tagfüggvények és szabályok kialakításához 20 alkalommal rajzoltuk ki a korábban kiválasztott betűtípusok felhasználásával a C betűket a kép ugyanazon pontjára 99%-os átlátszósággal. Így meg tudtuk vizsgálni, hogy a véletlenszerűen választott betűtípusok a kép mely szegmenseibe, milyen méretben és milyen elnyújtással helyezik a betű részeit. Felosztottuk a képet 10×10 pixeles négyzetekre, így vízszintesen és függőlegesen is 5-5 négyzetet képeztünk, tehát összesen 25 négyzetet kaptunk. Ezek közül a legfelső és legalsó sorban, illetve a bal és jobb oldali oszlopban egyszer sem tapasztaltuk, hogy a betű bármely része megjelent volna, így csak a 2-4. sor 2-4. oszlopait kellett vizsgálnunk, ami

egy 3×3 -as rács. Ezen négyzetek lettek a fuzzy inputváltozói. Az outputváltozó a C betű bekövetkezését volt hivatott reprezentálni.

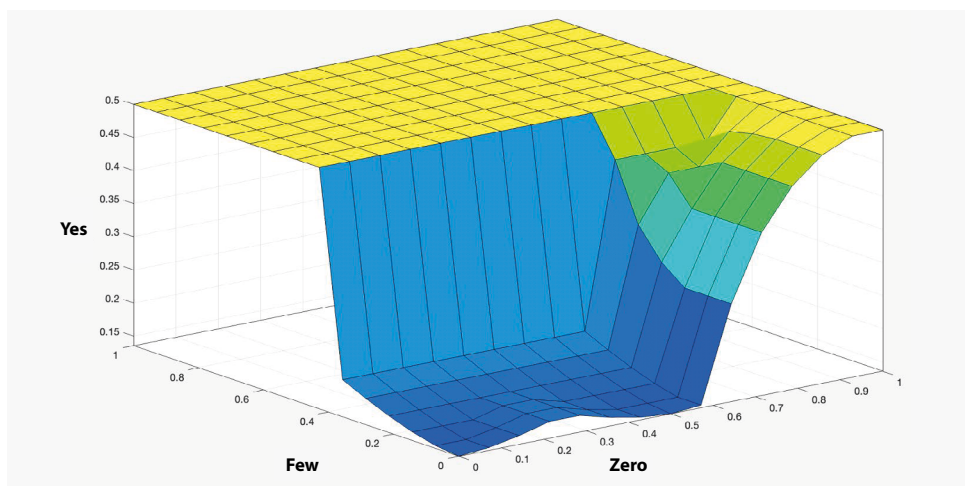
Minden inputváltozót ugyanazzal az 5 tagfüggvénnyel láttuk el, amelyek típusa és meredeksége mind az 5 változóban azonos volt. Egy-egy tagfüggvény azt mutatta meg, hogy mennyi, a háttérszíntől eltérő szín található az adott 10×10 pixeles négyzetben. A tagfüggvények és definíciójuk az alábbiak voltak:

- zero: szinte nincs pixel az adott régióban;
- few: kevés pixel van az adott régióban;
- mid: közepes mennyiségű pixel van az adott régióban;
- lot: sok pixel van az adott régióban;
- full: szinte kizárólag a karakterhez tartozó pixelek vannak az adott régióban.

Az outputváltozó tagfüggvényei azt mutatták meg, hogy mennyire tekinthetjük C betűnek a megadott mintát. A tagfüggvények és definíciójuk az alábbiak voltak:

- no: nem C betűt észlelt;
- maybe: talán C betűt észlelt;
- yes: valószínűleg C betűt észlelt.

Összesen 48 szabályt alkottunk a C betű felismeréséhez. Megfigyeléseink szerint a 3×3 -as rács 1. sorának 1. és 2. oszlopában, 2. sorának 1. oszlopában és utolsó sorának 1. és 2. oszlopában közepesen sok pixel, a 2. sor 2. és 3. oszlopát leszámítva a többiben kevés pixel tér el a háttértől. A „yes” halmazba tartozást meghatározó szabályok főként a fent említett cellákban lévő pixelek „few”-ba vagy „mid”-be történő besorolását valósították meg, a „no” halmazba tartozást meghatározó szabályok pedig főként a 2. sor 2. és 3. cellájára fókuszáltak, és azok „zero”-ba vagy „few”-ba történő besorolását végezték. Egy példát, a few és zero változó kapcsolatáról, a yes halmazba tartozás outputja tekintetében, a 2. ábra szemléltet.



2. ábra: A változók kapcsolata a 2. sor és 2. oszlopból vett minta esetén

Forrás: a szerzők szerkesztése

A szabályokat nem kellett fájlba menteni, így a vizsgálatnál a fuzzy logikát alkalmazva ezt az egyszeri időtöbbletet is megspórolhatjuk, jöllehet, a neurális háló vizsgálatakor sem számoltuk bele az ebből keletkező plusz milliszekundumokat az összehasonlításba. A memóriában tárolt fuzzyváltozók, tagfüggvények és szabályok felállítása után 100-szor végeztük el a vizsgálatot C betűre, és ugyancsak 100-szor végeztük el O betűre. A feldolgozni kívánt betű kirajzolásától kezdve a fuzzy logika kimeneti változójának értékeléséig mértük az időt.

Eredmények

A betűk kirajzolásához szükséges pixelek száma nem befolyásolta szignifikánsan a futási időt. Minden esetben a fuzzy logikával készült algoritmus volt a gyorsabb, mi több, jellemzően 1000-szer gyorsabb volt, mint a neurális háló. A betanított és a vizsgált karakterek közötti egyezés tekintetében a neurális háló pontosabb egyezést mutatott azokban az esetekben, amikor a betanított karakter és a vizsgált karakter azonosak voltak, és akkor is, amikor különböztek. Ez az érték átlagosan 0,8 és 0,97 között mozgott, ami komoly probléma, mert így megnehezíti annak a thresholdértéknek a meghatározását, amely elválasztja a C betűt az O betűtől. A fuzzy logika 0,7 körüli értékeket produkált azokban az esetekben, amikor a két karakter egyezett, és a legtöbb esetben 0-t, ha a két karakter nem egyezett. Ez és az algoritmus sebessége előnyt jelent a neurális hálóval szemben. Az egyezésre és a lefutás idejére vonatkozó részletes adatokat az 1. táblázat szemlélteti.

1. táblázat: A Hopfield neurális háló és a fuzzy logika közötti különbségek az egyezésre és futási időre vonatkozva 100 iteráció alatt

	Betanítás: C, vizsgálat: C				Betanítás: C, vizsgálat: O			
	Hopfield neurális háló		Fuzzy		Hopfield neurális háló		Fuzzy	
	Egyezés	Feldolgozási idő (ms)	Egyezés	Feldolgozási idő (ms)	Egyezés	Feldolgozási idő (ms)	Egyezés	Feldolgozási idő (ms)
Átlag	0,923	1878,05	0,76	1,746	0,901	1809,5	0,138	1,968
Összesen (ms)		187805		174,667		180950		196,825
Összesen másodpercenben		187,805		0,174		180,95		0,196

Forrás: a szerzők szerkesztése

Egyes esetekben a fuzzy logika 0,7 felettinek érzékelte az egyezést az O betű esetében is, azonban ez csak egy bizonyos betűtípusnál volt megfigyelhető. Ezek az egyezések, illetve egy másik betűtípus 0,5-ös értéke növelte az átlagot 0-ról 0,138-ra. Ezek az esetek azért problémásak, mert látható, hogy 0,7 körül lenne érdemes meghatározni azt a threshold-értéket, amelynél magasabb egyezés esetén elfogadjuk, hogy a vizsgált karakter C betű, viszont egyes esetekben az O betű is eléri ezt a mértékű egyezést. A 0,5-ös kimeneti érték tekintetben nem okoz problémát. 100 futtatás során 0,7 feletti értéket 11 esetben kaptunk. Meggyőződésünk, hogy a fuzzy tagfüggvények javításával vagy a betűtípusok limitálásával csökkenthető ez az előfordulás.

Összegzés

A kutatás értékelése

Hasonló kutatásokban is sikerült 85% fölötti arányban egyezést találni a karakterek között, mesterséges neurális hálóval, Debananda Padhi például 94%-os egyezést, Pradeepta K. Sarangi 85,30%-os egyezőséget mutató eredményeket kapott.⁸ További, nem elhanyagolható eredményeket mutatott K. Roy és társai munkássága, ahol 97,69%-os bizonyossággal ismerte fel a mesterséges neurális háló a karaktert.⁹ Pradeepta Sarangi és társai, akik vizsgálati módszerükben Hopfield neurális hálót alkalmaztak, 95,40%-os eredményt értek el.¹⁰

A fuzzy logikát Alata 2006-ban bemutatott megoldása alapján fejlesztettük, de több képkockára vágtuk a karaktereket. Alata munkájában a kép statikai nyomatékát használva vágták több részletre a karaktert. A fuzzy függvényeket hasonló módon definiálták. Eredményeiket összehasonlították négy karakterfelismerő algoritmussal. Figyelemre méltó eredmény, hogy fals pozitív értékeket egy alkalommal sem kaptak, míg az összehasonlítás gyanánt kipróbált algoritmusok minden esetben szolgáltatott fals pozitív értékeket, egyes esetekben 50%-nál nagyobbakat is. Kutatásunkban a fuzzy logika az Alata munkájában szereplő algoritmusoknál nem teljesített rosszabbul. Esetünkben fals pozitív eredmények 13,8%-ban fordultak elő. A hivatkozott forrásban a négy algoritmus közül az elfogadható eredményeket szolgáltató három algoritmus a következőképp szerepelt a fals pozitív eredmények tekintetében: ABBYY: 8%, Softwriting: 5%, Kwang-algoritmus: 19%. A negyedik, Matti-algoritmus 10 esetből 7 esetben 50% felett adott fals pozitív eredményt. A fentiek alapján az általunk alkotott fuzzy logika a Kwang-algoritmushoz hasonló eredményt ért el.¹¹

Kutatásunk megerősíti, hogy igény mutatkozik a különböző mesterségesintelligencia-módszerek teljesítményének összehasonlítására, ami megkönnyítené a feladat elvégzésére leghatékonyabb megoldás kiválasztását, vagy azt, hogyan építhetők egymásra, egymást segítve a különböző módszertanok.¹²

Kapcsolat a nemzetbiztonsággal

A generatív mesterséges intelligencia 2022-es berobbanása óta több szakértő adott interjút, szerepeltek podcastadásokban, amelyekben a laikus közönség számára igyekeztek érthető módon megfogalmazni a mesterséges intelligencia működését, határait és felhasználási lehetőségeit. Többen egyetértenek abban, hogy a meglévő módszertanokat vegyítve érdemes alkalmazásokat fejleszteni az élet minden területén. Az információbiztonság tekintetében a publikus adatok alapján nehéz megmondani, hogy mely szolgáltató használ mesterségesintelligencia-megoldást, és milyen.¹³

⁸ SARANGI-SAHOO-AHMED 2012.

⁹ ROY 2005.

¹⁰ SAHOO-KUMAR 2020.

¹¹ ALATA-AL-SHABI 2006.

¹² WAN 2023.

¹³ SÁFRÁN 2023.

Az adatbiztonság kérdése az informatikai rendszerekhez kevésbé értő vállalkozásokat (amelyek jellemzően kkv-k) érintik a legjobban, így átfogó hazai elemzés szükséges, amely a kkv-szektor szereplőinek lehetőségeit méri fel, informatikai rendszereik biztonságának mesterséges intelligencia általi javítását célozza meg. A kkv-szektor mellett a közigazgatási egységek lehetőségeit is érdemes elemezni, mivel az állampolgárok érzékeny adataival dolgoznak. Az általános szakértői vélekedés és jelen tanulmány is rávilágít arra, hogy nem feltétlenül a piacon elérhető mesterséges intelligenciát is alkalmazó szoftveres megoldásokat érdemes implementálni, hanem a probléma oldaláról közelíteni, és a mesterségesintelligencia-módszertanok széles palettájáról választani hatékony megoldást. Ehhez a módszertanok azonos hardveres és szoftveres környezetben történő összehasonlítása kulcsfontosságú. Az egyes vállalati vagy közigazgatási területen felmerülő adatbiztonsági problémákra kipróbált több módszer összehasonlítása alapot adhat egy standard felállításához.

Felhasznált irodalom

- ALATA, Mohanad – AL-SHABI, Mohammad (2006): Text Detection and Character Recognition Using Fuzzy Image Processing. *Journal of Electrical Engineering*, 57(5), 258–267. Online: www.researchgate.net/publication/250802079_Text_detection_and_character_recognition_using_fuzzy_image_processing
- GONZÁLEZ, P. et al.(2019): Approximation of Fuzzy Functions by Fuzzy Interpolating Bicubic Splines: 2018 CMMSE Conference. *Journal of Mathematical Chemistry*, 57, 1252–1267. Online: <https://doi.org/10.1007/s10910-018-0946-x>
- LIMA, Bráulio Nascimento et al. (2021): Artificial Intelligence Based on Fuzzy Logic for the Analysis of Human Movement in Healthy People: A Systematic Review. *Artificial Intelligence Review*, 54, 1507–1523. Online: <https://doi.org/10.1007/s10462-020-09885-8>
- OLUDARE, Isaac Abidoun et al. (2018): State-Of-The-Art in Artificial Neural Network Applications: A Survey. *Heliyon*, 4(11). Online: <https://doi.org/10.1016/j.heliyon.2018.e00938>
- ROY, K. et al. (2005): Oriya Handwritten Numeral Recognition System, in Eighth International Conference on Document Analysis and Recognition. *IEEE*, 770–774. Online: <https://doi.org/10.1109/ICDAR.2005.183>
- SAHOO, Ramesh – KUMAR, Sateesh (2020): Behavior of Learning Rules in Hopfield Neural Network for Odia Script. *International Journal of Advanced Computer Science and Application*, 11(1). Online: <https://doi.org/10.14569/IJACSA.2020.0110155>
- SÁFRÁN József (2023): A mesterséges intelligencia és a rendvédelmi szervek, valamint a közigazgatás kapcsolata. *Nemzetbiztonsági Szemle*, 11(4), 20–34. Online: <https://doi.org/10.32561/nsz.2023.4.2>
- SARANGI, Pradeepta K. – SAHOO, Ashok K. – AHMED, P. (2012): Recognition of Isolated Handwritten Oriya Numerals Using Hopfield Neural Network. *International Journal of Computer Applications*, 40(8), 36–42. Online: <https://doi.org/10.5120/4986-7250>
- The Franklin Institute [é. n.]: *John J. Hopfield*. Online: <https://fi.edu/en/awards/laureates/john-j-hopfield>
- WAN, Xiaojuan et al. (2023): An Algorithm for English Capital Letters Based on Discrete Hopfield Network and BP Network. Preprint. Online: <https://doi.org/10.21203/rs.3.rs-3714029/v1>