Gépelésdinamika alapján történő személyazonosítás a SUCCESS semi-supervised osztályozási eljárás felhasználásával

TDK dolgozat

Neubrandt Dóra

ndori0713@gmail.com

+36 70 371 4724

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Számítástudományi és Információelméleti Tanszék

másodéves BSc hallgató

Dr. Buza Krisztián

buza@cs.bme.hu

Rheinische Friedrich-Wilhels-Universität

Bonn

külső konzulens

Dr. Csima Judit

csima@cs.bme.hu

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Számítástudományi és Információelméleti Tanszék

belső konzulens

„AZ EMBERI ERŐFORRÁSOK MINISZTÉRIUMA ÚNKP-16-1-I. KÓDSZÁMÚ ÚJ NEMZETI KIVÁLÓSÁG PROGRAMJÁNAK TÁMOGATÁSÁVAL KÉSZÜLT"

Absztrakt – magyar nyelvű összefoglaló

Kutatásom fő témája a gépelésminta alapján történő személyazonosítás, amihez különböző gépi tanulási módszereket használunk. Az e téma iránti egyre nagyobb érdeklődés többek közt köszönhető az online szolgáltatások növekvő népszerűségének (pl. online bankolás, online kurzusok), és annak, hogy ez az eljárás olcsó, könnyen alkalmazható, továbbá megbízható, hisz gépelésünk dinamikája egyedi, nehezen utánozható.

A kutatási program célja az, hogy hozzájáruljon az innovatív, olcsó biometrikákon alapuló személyazonosítási eljárások fejlesztéséhez a gépi tanulásra épülő megoldások alkalmazása révén. A lehetséges biometrikák közül a projekt során a gépelés dinamikájával foglalkozom. Gépelés dinamikája alatt az egyes billentyűleütések hosszát értjük. Egy gépelésdinamikán alapuló felhasználó-azonosítási rendszer használata során folyamatosan nagy mennyiségben keletkezik ún. „címkézetlen" adat (olyan adat, amelyről nem tudjuk biztosan, hogy melyik felhasználó gépelte), ezért érdemes olyan ún. félig-felügyelt (semi-supervised) eljárásokat vizsgálni, amelyek a címkézett adatok mellett a címkézetlen adatokból is képesek „tanulni".

Munkám során egy már létező, de a gépelésdinamika alapján történő személyazonosításra korábban nem alkalmazott gépi tanulási eljárást vizsgáltam. Konkrétan az idősorok félig-felügyelt osztályozására kidolgozott SUCCESS [1] eljárást adaptáltam a gépelésdinamika alapján történő személyazonosításra, és mértem az eljárás pontosságát.

Az eljárást különböző paraméterekkel publikusan elérhető adatokon teszteltem és biztató eredményeket kaptam, ugyanis a SUCCESS eljárás prediktálási pontossága lényegesen felülmúlta a baseline-nak választott random prediktálás pontosságát.

Abstract – summary in English

The main focus of my project is the person identification based on keystroke dynamics, for which we use different machine learning methods. There is an increasing interest for this topic which can be attributed to several factors including the growing popularity of online services (e.g. online banking, online courses). Furthermore, it is cheap and widely applicable as our dynamics of typing is characteristic to us, so one can hardly able to mimic another person's dynamics of typing.

The aim of this project is to contribute to the development of innovative person identification techniques which are based on cheap biometrics, using machine learning methods. Of these biometrics I will focus on keystroke dynamics. With keystroke dynamics we mean the duration of each keystroke. During the usage of a system which applies person identification based on keystroke dynamics, large amount of unlabelled data (data from which we do not know which user typed it) is generated continuously, so it is worth to examine semi-supervised methods, which can „learn" from the unlabelled data besides the labelled data.

My goal was to adopt and examine the performance of an existing machine learning method, SUCCESS [1], which have not been used in the area of person identification based on keystroke dynamics yet.

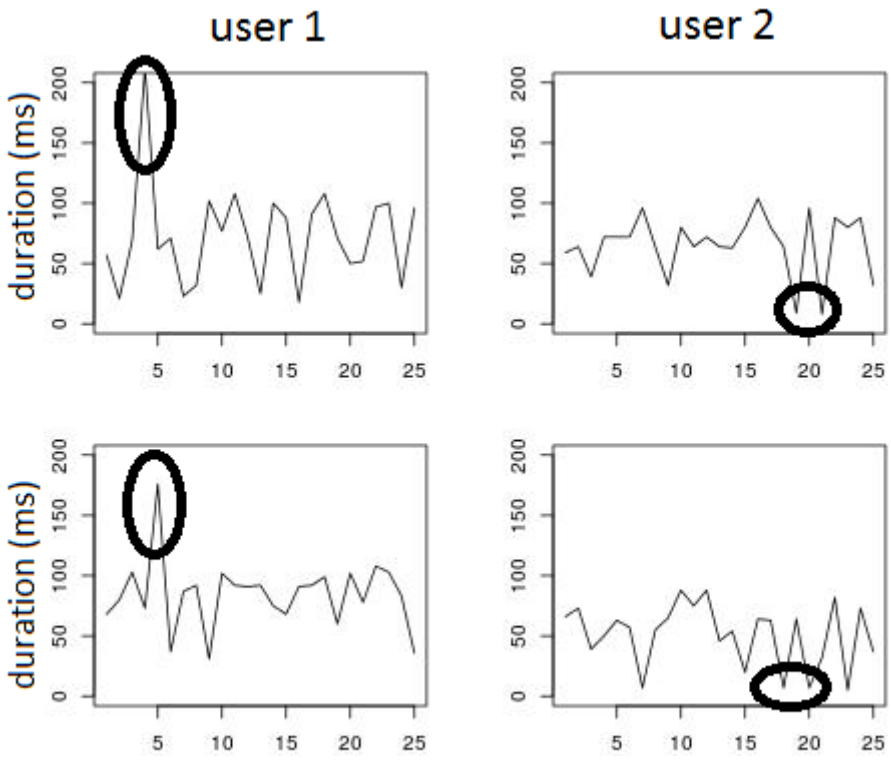
I tested the proposed method on publicly available datasets with different parameters. I have got promising results as the prediction accuracy of SUCCESS greatly outperformed the baseline which was the random prediction.

Tartalom

1. Bevezetés

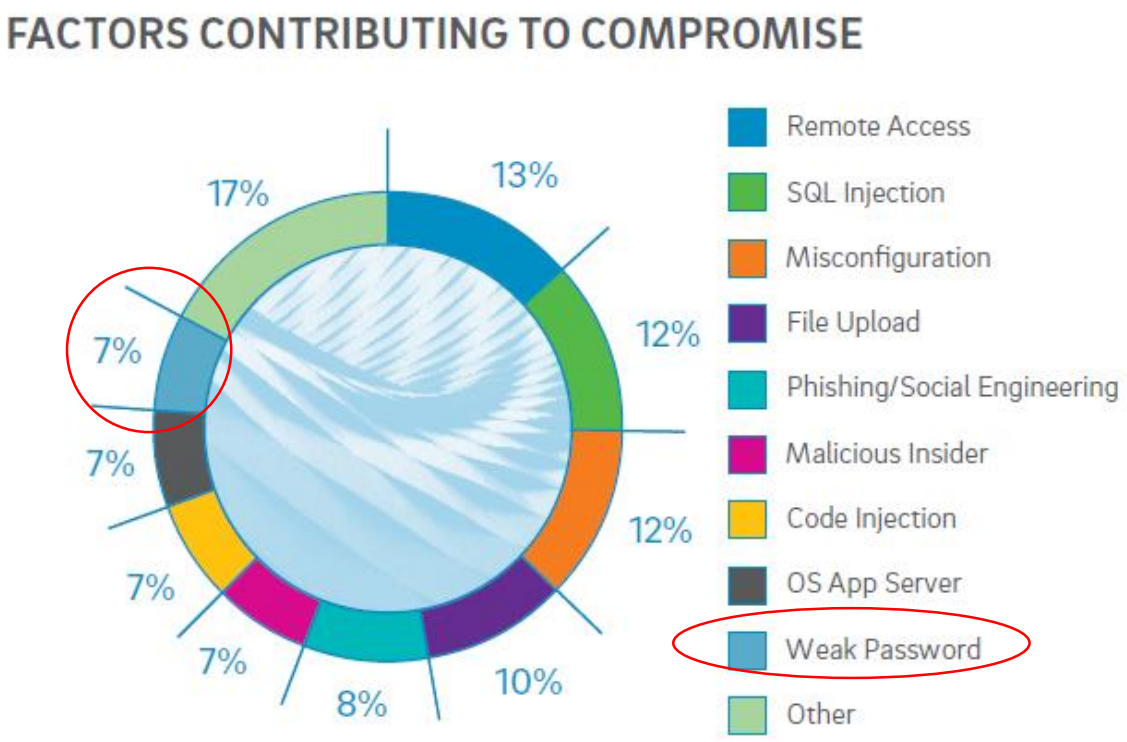
A gépelésdinamika alapján történő személyazonosítás iránt egyre növekvő érdeklődés, több tényezőnek köszönhető. Először is, gépelésünk dinamikája azért is alkalmas személyazonosítási feladatokhoz, mert egyedi, nehezen utánozható. Gépelés dinamikája alatt az egyes billentyűleütések hosszát értjük.

Az alábbi ábrán láthatjuk 25 db egymást követő billentyűlenyomás hosszát két különböző felhasználónál (user1 és user2), amint ugyanazt a szöveget gépelték be kétszer. Megfigyelhetjük a jellegzetességeket az ugyanazon felhasználótól kapott gépelésminták között. Például user1-re jellemző egy hosszabb ideig tartó billentyűlenyomás az 5. billentyűlenyomás környékén, míg user2-re több különösen rövid billentyűlenyomások jellemzőek a 20. billentyűlenyomások környékén. Ezeket a karakterisztikákat így szabad szemmel is meg lehetett állapítani, ránézésre el tudtuk dönteni, hogy user1 gépelésmintája sokkal inkább hasonlít a saját gépelésmintájára, mint user2-ére és fordítva. Azonban egy olyan rendszernél, ahol több ezer felhasználót tartunk nyilván, nehézkes, időigényes és pontatlanabb lenne az ilyesfajta kiértékelés emberi munkával, ezért erre a feladatra gépi tanulási módszerek alkalmazása ajánlott.



1. ábra: Két felhasználó első 25 billentyűlenyomásának a hossza

A gépelésdinamika alapján történő személyazonosítás iránti növekvő érdeklődés többek között köszönhető az egyre népszerűbb online szolgáltatásoknak, mint például az online kurzusok vagy online bankolás, amik gyors, megbízható és olcsó személyazonosítási módszereket igényelnek. Az olyan rendszerekben, ahol különböző személyazonosítási eljárásokat alkalmaznak párhuzamosan, mint például az online bankolásnál, a felhasználó gépelésének dinamikáját is fel lehetne használni a személyazonosság megerősítéséhez.
Azért is van szükség új személyazonoítási módszerekre a régi módszerek mellett, mert a meglévő megoldásokat folyamatosan, egyre kifinomultabb technikákkal támadják. Ezt az is mutatja, hogy manapság az egyik legelterjedtebb személyazonosítási módszer a jelszavak használata ,mindemellett Trustwave 2016 Global Security Report [2] szerint a gyenge jelszavak még mindig igen jelentős százalékban járulnak hozzá a támadhatósághoz.



2. ábra: Hozzájáruló tényezők egy betöréshez [2]

Így például több személyazonosítási eljárás kombinálása, segíteni biztosabb védelmet kapni.

2. Szakirodalmi áttekintés

Az elmúlt évtizedben elindult a gépi tanulás rohamos mértékű fejlődése, ami napjainkban is tart. Az életünkben szinte mindenhol megtalálható a gépi tanulás, pl spam levélszűrés a levelezőprogramokban [3] , ajánló rendszerekben [4], sőt még az agykutatásban [5] is.
A gépi tanulásnak a lényege, hogy bizonyos adatokon tanítva (train data) különböző modelleket, azok képesek felismerni, „megtanulni" az összefüggéseket, így legközelebb ismeretlen adatokon is képesek prediktálni, az eddigi „tanulmányaikra" támaszkodva.
A gépi tanulási feladatok és a hozzájuk használt módszerek 3 nagy csoportra bonthatók aszerint, hogy mennyi információnk van az adatokról.

- Az első a **felügyelt tanulás (supervised learning),** ahol a tanítóadatoknak tudjuk a címkéjét is. A gépelésdinamikai kontextusban ez azt jelenti, hogy minden egyes gépelésmintáról a tanító adathalmazból tudjuk, hogy melyik felhasználó gépelte.
- A második a **felügyelet nélküli tanulás (unsupervised learning),** amikor a tanítóadat címkézetlen. Tehát nem tudjuk, hogy a tanító adatokban melyik gépelésminta melyik felhasználóhoz tartozik.
- A harmadik a **félig-felügyelt tanulás (semisupervised learning)** ahol vannak felcímkézett és címkézetlen tanító adataink is, tehát itt a címkézetlen adatokból is tanul a rendszer. A gépelésdinamikás analógiát követve, ez azt jelenti, hogy rendelkezésünkre állnak olyan gépelésminták melyekről tudjuk, hogy melyik felhasználó gépelte és vannak olyanok is, amelyekről nem.

A gépelésdinamika, mint biometrika, gépi tanulással történő felhasználása különböző felismerési feladatokra számos területen egyre elterjedtebb.
Figyelemre méltó Antal Margit munkássága. Például megvizsgálta, hogy touchcreen-el rendelkező okostelefonokon gyűjtött gépelésmintákból és swipe-mintákból meghatározható-e a felhasználó neme.
Továbbá a gépelésdinamika alapján történő személyazonosítással is foglalkozott Android platformon. Azt vizsgálta, hogy ha a gépelésdinamika attribútumain kívül figyelembe vesszük az érintőképernyő által kínált egyéb paramétereket is (mint a nyomás és az ujjlenyomat terület),

akkor javul a gépelésdinamika alapján történő személyazonosítás pontossága. Összesen 42 ember vett részt ebben a tanulmányban, akiknek az adatait kétfajta Androidos készüléken: Nexus7 tableten és LGOptimus L7 II P710 telefonon gyűjtötték. Először egy Android platformon készített regisztrációs űrlapot kellett kitölteniük, majd egy előre megadott, erősnek számító jelszót kellett 30-szor begépelniük. A kutatás során a gépelésdinamikának 4 attribútumát vették figyelembe (billentyűlenyomások hossza, egymás utáni billentyűleütések közt eltelt idő, billentyű elengedése és a következő megnyomása közt eltelt idő, a billentyűlenyomások átlagos ideje). Ezek mellet még mérték az Androidos készülékeken mérhető további attribútumokat, mint a billentyűlenyomás közbeni nyomás és az ujjlenyomat területe, nyomások átlaga és az ujjlenyomatok területének átlaga. A kitűzött feladathoz több különböző ismert gépi tanulási módszert próbáltak ki, és eredményként azt kapták, hogy az általuk ajánlott attribútumok figyelembevételével tényleg nő a klasszifikáció pontossága.
Megjegyzendő, hogy az előbb említett kutatás eredményei azért nem összehasonlíthatók a jelenlegi projekt eredményeivel, mert ott más platformon, több attribútumot vettek figyelembe. Ebben a kutatómunkában azért is esett a döntés csak a billentyűleütések hossza mellett, mint figyelembe vehető attribútum, mivel egy szélesebb körben használható eljárás fejlesztése a cél, és ez egy olyan adat, amit mindenhol fel lehet venni, sokkal szélesebb körben, mint például a billentyűlenyomás erősségét, amit egy normál billentyűzet nem mér. Valamint az volt a célunk, hogy egy olyan gépi tanulási eljárás, konkrétan SUCCESS eljárás, alkalmazási lehetőségeit vizsgáljuk, amelyet korábban nem használtak erre a feladatra.

Ebben a projektben mi a gépelésdinamika alapján történő személyazonosítás problémájára koncentráltunk.

2.1 Kapcsolódó gépi tanulási eljárások

Ahhoz, hogy megértsük a SUCCESS eljárás működését és kiértékelését, először néhány gépi tanulási eljárást ismertetünk.

Klaszterezés korlátok mellett (constrained clustering)

Klaszterezésnek nevezzük, amikor az adatokban automatikusan felismerjük a hasonló példányokból álló csoportokat. Amikor korlátok mellett klaszterezünk, akkor az algoritmusnak különböző kikötéseket adhatunk meg a példányok között. A cannot-link (CN) kikötés leírja, hogy két példány nem lehet ugyanabban a klaszterben, míg a must-link (ML) kikötés leírja, hogy két példánynak ugyanabban a klaszterben kell lennie.

Hierarchikus agglomerative klaszterező (HAC) algoritmusoknál kezdetben minden példány külön klaszterbe tartozik. Majd iteratív módon egyesítjük a klasztereket. Minden iterációban a két leghasonlóbb klasztert vonjuk össze. Ezt addig folytatjuk, amíg el nem érjük a kívánt klaszterszámot, vagy már a két legközelebbi klaszter távolsága is túl nagy (átlép egy meghatározott küszöbértéket). A kívánt klaszterszám és a távolság küszöbértéke egy külső, a felhasználó által megadott paraméter. A constrained klaszterezés esetében még a kikötéseket is figyelembe vesszük az iteratív folyamat során, azaz például nem vonunk össze két olyan klasztert, amelyek között cannot-link kikötés van, a must-link kikötésekkel összekötött példányokat pedig rögtön az első iterációban összevonjuk.

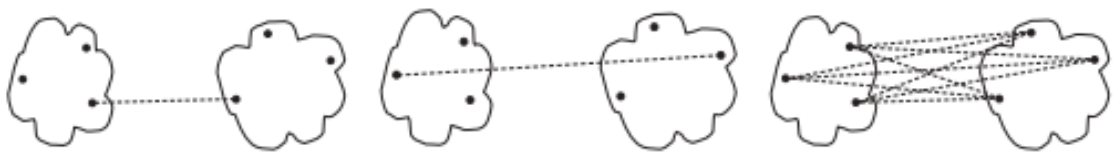
Legelterjedtebb hierarchikus kaszterező eljárások [7] :

Single link esetben, azon két klasztert tekintjük leghasonlóbbnak, amelyeknél a két külön klaszterba tartozó példányok közötti távolság minimális.

Ezzel szemben, a **complete link** esetben, azon két klasztert tekintjük leghasonlóbbnak, amelyeknél a két külön klaszterba tartozó példányok közötti távolság maximális.

A **group average** verzióban azon két klasztert tekintjük a leghasonlóbbnak, ahol az összes külön klaszterbeli példánypárok hasonlóságának az átlaga maximális.

A 3. ábra szemléletesen bemutatja, hogy ez a 3 eljárás hogyan számítja a klaszterek hasonlóságát.



3. ábra: Hogyan számítja a klaszterek hasonlóságát a single link, complete link és group average link hierarchikus klaszterező eljárás [7]

Osztályozás és címkézés (cluster and label)

A klaszterezés és címkézés technikában először klaszterezzük az adatainkat (korlátozott vagy nem korlátozott klaszterezéssel). Ezután a klasztereket leképezzük osztályokra valamilyen algoritmus segítségével. Például ilyen leképzés lehet, az, ha többségi szavazat alapján, a klaszter arra az osztályra lesz leképezve amelyikből a legtöbb címkézett elemet tartalmazza.
Az osztályozás és címkézés eljárás akkor működik jól, ha az előzetes klaszterezés felismerte az adatok igazi struktúráját.

Self training

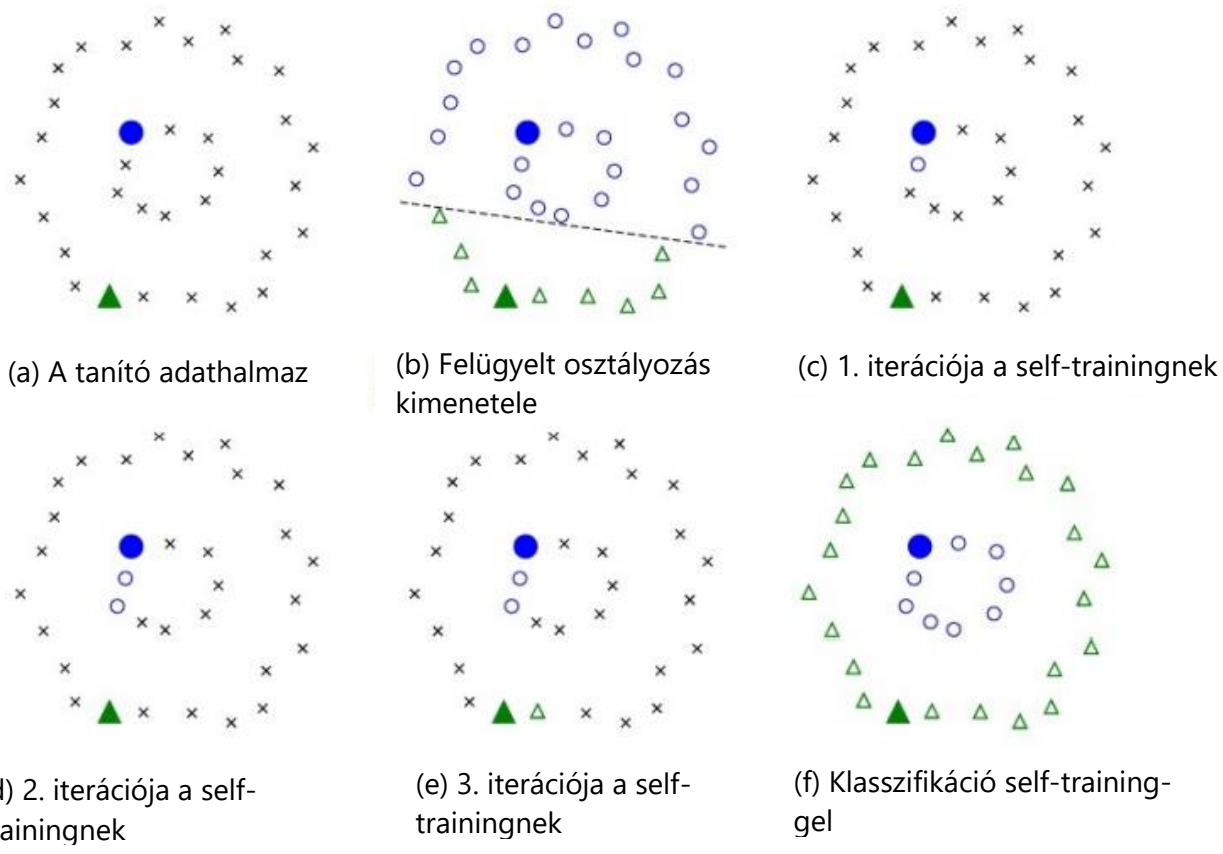
Ez az egyik leggyakrabban használt félig-felügyelt algoritmus. Azért kerül itt ismertetésre, mert a SUCCESS eljárást bemutató cikkben [1], a self-traininget választották baseline-nak, aminél a cikkbeli eredmények szerint általában jobb a SUCCESS. Ezért is esett a választás a SUCCESS alkalmazására jelen projektben.

SELF-TRAINING(L, U)
1 $L_0 = L$
2 $U_0 = U$
3 $t = 0$
4 **repeat**
5 $M = \text{SUPERVISED-LEARNING}(L_t)$
6 $x_{best} = \arg\max_{x \in U_t} \text{CERTAINTY}(M, x)$
7 $\hat{y} = \text{CLASSIFY}(M, x_{best})$
8 $L_{t+1} = L_t \cup \{(x_{best}, \hat{y})\}$
9 $U_{t+1} = U_t \setminus \{x_{best}\}$
10 $t = t + 1$
11 **until** $|U_t| == 0$
12 **return** M

4. ábra: Self-training pszeudokódja [1]

A self-training egy olyan általános eljárás, ami széleskörben használható különböző osztályozókkal, azok teljesítményének növelésére. Ahhoz, hogy a self-training-et használni tudjuk olyan osztályozóra van szükségünk, ami a példányoknak nem csak az osztálycímkéjét prediktálja, hanem visszaad egy „valószínűségi" értéket is, ami azt mondja meg, hogy mennyire valószínű, hogy az általa prediktált osztálycímke helyes.

A self-training egy iteratív eljárás. A címkézett példányok halmaza addig nő, ameddig az összes példány címkézett nem lesz. Legyen L_1 a kezdetben címkézett példányok halmaza. Továbbá jelöljük L_t-vel a címkézett példányok halmazát a t-edik iterációban (t>=1). Minden egyes iterációban a felügyelt osztályozót a címkézett adatok halmazán tanítjuk, majd ezzel az osztályozóval osztályozzuk a címkézetlen példányokat. Az osztályozott példányok közül, azokat, amiknek legnagyobb az ún. valószínűségi értéke, a címkéjükkel együtt bevesszük a már címkézett példányok halmazába. Így megkaptuk az L_{t+1} –et, amely halmazon az t+1-edik iterációban tanítjuk az osztályozónkat. A legegyszerűbb esetben, minden egyes iterációban egy példányt adunk hozzá a címkézett példányok halmazához.



(a) A tanító adathalmaz

(b) Felügyelt osztályozás kimenetele

(c) 1. iterációja a self-trainingnek

(d) 2. iterációja a self-trainingnek

(e) 3. iterációja a self-trainingnek

(f) Klasszifikáció self-training-gel

5. ábra: Self-training legközelebbi szomszéd osztályozóval [1]

Az 5. ábrán láthatjuk a self-training használatát legközelebbi szomszéd osztályozóval. Két osztály van, a körök és a háromszögek. A kiszínezett alakzatok felelnek meg a kezdetben címkézett tanítóadatoknak L_1, míg a címkézetlen példányokat x –szel jelölik. A (c) -(f) ábrák

mutatják be a self-training lépéseit, és az (f) ábrán láthatjuk a self-training kimenetét. Ezzel szemben látjuk, a (b) ábrán, hogy ha csak simán a felügyelt osztályozót használtuk volna, akkor mit kaptunk volna eredményül. Látszik, hogy ez az osztályozó self-traininggel használva, pontosabban követi az adatok mintáját.

Idősorok félig-felügyelt osztályozása

Az idősorok osztályozásával kapcsolatban egy igen meglepő tudományos eredmény az, hogy az egyszerű legközelebbi szomszéd osztályozók, a speciális DTW (Dynamic Time Warping) távolságmérő eljárással használva, igen kompetitívek a sokkal bonyolultabb eljárásokkal szemben. Ezért is a SUCCESS eljárás a DTW alapú legközelebbi szomszéd klasszifikációra épül.

2.2. SUCCESS módszer

A SUCCESS [1] eljárást Marussy Kristóf mutatta be 2013-ban az International Conference on Artificial Intelligence and Soft Computing-on, Zakopánéban, és jelentette meg a konferencia kiadványban. A következőkben ezeket az eredményeket tekintjük át.

A SUCCESS egy félig felügyelt, idősorok osztályozására készített eljárás. Tehát olyan problémákra használható, ahol a tanító adatokban vannak úgynevezett címkézett és címkézetlen idősorok is. A SUCCESS ezeken az adatokon tanítva, próbál egy új, eddig ismeretlen idősort osztályozni.

A SUCCESS eljárás lépései

- Constrained single-link hierarchikus klaszterezővel klaszterezzük mind a címkézett és címkézetlen példányait a tanítóhalmaznak. Klaszterezés közben az idősorok távolsága alatt a DTW távolságukat értjük Továbbá cannot-link kikötéseket szabunk ki minden címkézett példánypár között, még akkor is, ha azonos címkéjűek.
- Az így kapott klasztereket a bennük levő egyetlen címkézett adat címkéjével címkézzük fel.

- Majd az így kapott most már csak címkézett idősorokat tartalmazó tanítóadatokon egy legközelebbi szomszéd osztályozót tanítunk, és ez lesz a végső osztályozónk.

A SUCCESS eljárás készítői továbbá vizsgálták az eljárás feszítőfa kereséssel, konkrétan Kruskal algoritmussal való rokonságát is.

3. SUCCESS módszer alkalmazása gépelésdinamika alapján történő személyazonosításra

Mi a gépelésdinamika alapján történő személyazonosítással foglalkozunk. Mint már fentebb is említettük, a gépelés dinamikája alatt az egyes billentyűleütések hosszát értjük. Tehát minden egyes gépelésminta egy idősor.
Ahhoz, hogy gépelésminta alapján történő személyazonosítást tudjunk végezni, gépi tanulás kell. A gépi tanuláshoz viszont adatok, tehát jelen esetben gépelésminták szükségesek. A módszer használatát úgy képzeltük el, hogy a felhasználó a regisztráció során 2-3-szor begépel egy szöveget, ugyanis ennél többször nem feltétlenül kérhetjük meg, mert a végén megunja, és ott hagyja az egész rendszert. Az imént begyűjtött pár gépelésminta alapján már egy „kezdeti", még nem kifejezetten pontos, de valamennyire elfogadható pontosságú azonosítást már tudunk végezni. A használat során, amikor újra bejelentkezik a felhasználó, gyűlik a címkézetlen adat (amiről nem tudjuk, hogy tényleg a felhasználó gépelte-e). Amikor van elég sok címkézetlen adat, akkor érdemes újra tanítani a rendszert, de ekkor már egy félig felügyelt eljárást, jelen esetben a SUCCESS-t használva.
Annak ellenére, hogy a SUCCESS ígéretes félig-felügyelt idősor osztályozási eljárás, korábban még nem lett alkalmazva a gépelésdinamika alapján történő személyazonosítás feladatára.

Tehát a kitűzött feladatunkra SUCCESS eljárást alkalmaztuk és megnéztük az eljárás pontosságát.

Az eljárás során felhasznált gépelésmintákat, a www.biointelligence.hu oldalon kiírt online Person Identification Challenge-ről vettük. A felhasznált adathalmazban 12 felhasználótól gyűjtött, összesen 548 darab gépelésminta van. A felhasználóknak minden egyes gépelésminta megadásánál, ugyanazt a rövid szöveget kellett begépelniük. Ez konzisztens azzal a felállással, amikor egy online-rendszerben a felhasználó mindig a saját jelszavát gépeli be, vagy egy előre megadott, az azonosításhoz előírt rövid szöveget.
Az oldalon két féle feladat van kitűzve, az egyik a Person Authentication, ahol a gépelésmintákhoz adott egy – egy hipotetikus címke, és el kell eldönteni, hogy az adott gépelésmintát begépelő felhasználó vajon megegyezik-e az állított felhasználóval, a másik pedig a Person Identification, ahol el kell döntenünk, hogy melyik gépelésminta melyik

felhasználótól származik. Mi a Person Identification feladatát választottuk, mivel úgy gondolkodtunk, hogy ez a felállás nehezebb, itt nincs megadva egy címke, amiről csak el kell döntenünk, hogy helyes-e vagy sem.

Ezek a gépelésdinamikáról gyűjtött adatok feldolgozatlan, egy JavaScript applikáció által rögzített formátumban elérhetők. Így, először fel kellett dolgoznunk a nyers adatokat, amihez Python kódokat írtunk. Az egyes billentyűleütések hosszát tartottuk relevánsnak a gépelésdinamika jellemzői közül.

```
TYPING PATTERN 0
keydown 16 16 0 true 0
keydown 84 84 0 true 28
keypress 84 84 84 true 28
keyup 84 84 0 true 100
keyup 16 16 0 false 122
keydown 72 72 0 false 247
keypress 104 104 104 false 249
```

6. ábra: Egy JavaScript kód által gyűjtött nyers adat

Miután elkészítettük a feldolgozott adatokat, a SUCCESS eljárást implementáltuk a fent említett 3 lépésének megfelelően, szintén Python programozási nyelvet használva.

Az oldalon elérhető volt mind a 12 felhasználótól 5- 5 gépelésminta, tanulóadatként. Mi ebből felhasználónként csak 2-nek illetve 3-nak használtuk fel a megadott címkéjét, a többit a maradék 5-2 illetve 5-3 mintából címkézetlennek vettük, hisz egy olyan rendszerben ahol semi-supervised eljárást használunk, feltételezzük, hogy nem áll rendelkezésünkre sok címkézett adat.

Majd az így keletkezett tanító halmazon tanítottuk a SUCCESS eljárásnak a PyHubs[1] csomagban publikusan elérhető implementációját, aminek az elkészítéséhez én is hozzájárultam.

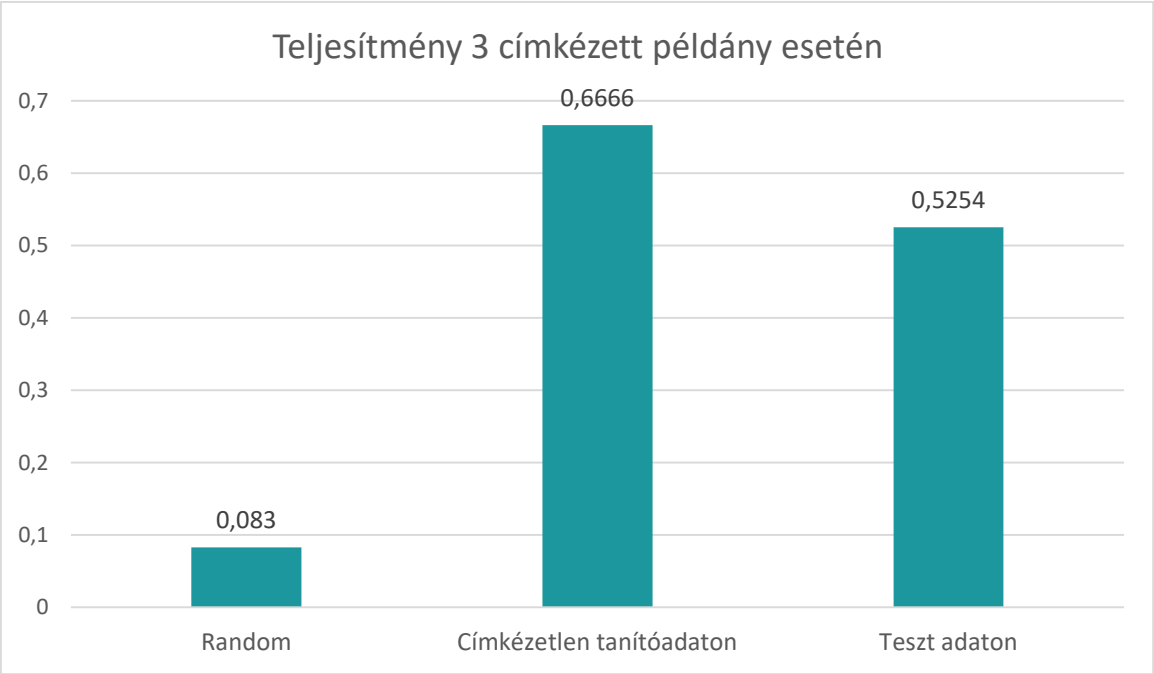
Végül az így tanított modellt alkalmaztuk a teszt adatokra, amik szintén elérhetőek a honlapon.

[1] http://www.biointelligence.hu/pyhubs

4.Kísérleti eredmények és az eredmények diszkussziója

Több féle paraméterezéssel is kipróbáltuk az eljárást. Az egyik eset, amikor a modell tanítóhalmazában 2 címkézett példány van, a másik pedig az amikor 3.

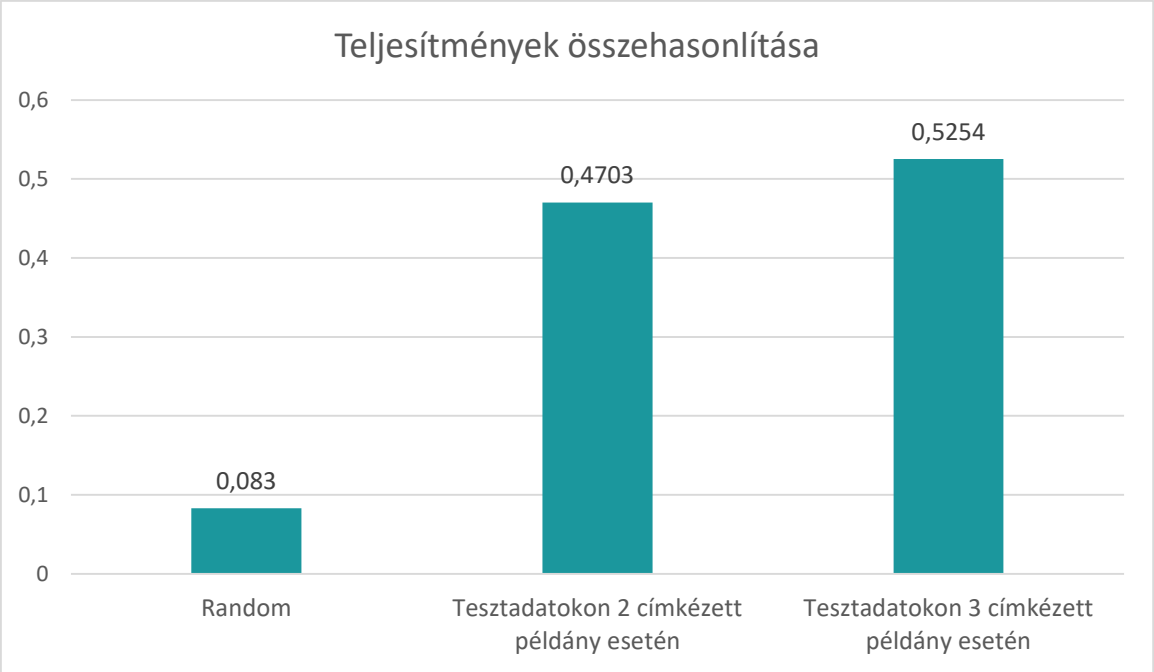
Az alábbi diagramok foglalják össze keletkezett modellek teljesítményét a különböző tanítóadatok mellett, összehasonlítva a baseline-ként választott random találgatással.



7. ábra: Teljesítmény 3 címkézett példány esetén

Kétféle teljesítményt vizsgáltunk. Az egyik az az, hogy mekkora valószínűséggel címkézte fel az eljárás a címkézetlen tanítóadatokat (Unlabeled train) helyesen, a másik meg az, hogy mekkora valószínűséggel prediktált az eljárás az ismeretlen tesztadatokra helyes értéket. A 7. ábrán láthatjuk, hogy a 3 címkézett példányon tanított modell jelentősen pontosabban prediktál, mint a random találgatás.

A 8. ábrán láthatjuk, hogy a 3 címkézett példányt tartalmazó tanítóadaton tanított modell pontosabban prediktál, mint a 2 címkézett példányon tanított modell.



8. ábra: A teljesítmények összehasonlítása

Mérésünkben alap eljárásnak, amihez hasonlítjuk a mérésünk eredményeit a random tippelést választottuk. A random találgatásnál 1/12 = 0.083 az esélye annak, hogy az eljárás pontosan állapítja meg a példányok osztálycímkéit. Ezzel összevetve a SUCCESS eljárás során kapott eredményeket, kijelenthetjük, hogy lényegesen pontosabban prediktál az eljárásunk, egy random találgatásnál.

5. Kitekintés és összegzés

Jövőbeli kutatások során a gépelésünk dinamikájának figyelembe lehetne több attribútumát, mint például egy billentyűelengedés és a következő billentyűleütés közt eltelt idő, a gépelés során használt törlések száma vagy a billentyűlenyomások erőssége.

Összegzés

Ebben a kutatásban, megvizsgáltam a SUCCESS eljárás teljesítményét egy olyan feladat esetében, amire még nem használták. Konkrétan a gépelésdinamika alapú személyazonosítási feladatra javasoltam egy új, a SUCCESS-re épülő eljárást. Ez a megoldás akár információs rendszerek biztonsága szempontjából is releváns lehet, például jelszó alapú azonosítással történő kombinálás esetén. Továbbá hozzájárultam a publikusan elérhető PyHubs szoftvercsomag fejlesztéséhez, többek között a SUCCESS eljárás implementációjában való részvételemmel.

6. Hivatkozások

[1] K. Marussy, K. Buza (2013): SUCCESS: A New Approach for Semi-Supervised Classification of Time-Series, ICAISC, LNCS Vol. 7894, pp. 437-447, Springer

[2] The 2016 Trustwave Global Security Report https://www2.trustwave.com/GSR2016.html?utm_source=redirect&utm_medium=web&utm_campaign=GSR2016

[3] Bahgat, E. M., Rady, S., & Gad, W. (2016). An e-mail filtering approach using classification techniques. In The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), November 28-30, 2015, Beni Suef, Egypt (pp. 321-331). Springer International Publishing.

[4] Portugal, I., Alencar, P., & Cowan, D. (2015). The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review. arXiv preprint arXiv:1511.05263

[5] Rodica Ioana Lung, Mihai Suciu, Regina Meszlényi, Krisztian Buza, Noémi Gaskó (2016): Community structure detection for the functional connectivity networks of the brain, 14th International Conference on Parallel Problem Solving from Nature

[6] Antal, M., Szabó, L. Z., & László, I. (2014). Keystroke dynamics on android platform. In 8th International Conference Interdisciplinarity in Engineering, INTER-ENG (pp. 9-10).

[7] Pang-Ning, T., Steinbach, M., & Kumar, V. (2006). Introduction to data mining. In Library of congress (Vol. 74).