Biometrikus azonosítás érintőképernyős gesztúrákkal

Touchscreen gestures for biometric identification

ANTAL Margit

Sapientia EMTE, Műszaki és Humántudományok kar, Marosvásárhely
manyi@ms.sapientia.ro

Abstract

In this paper we investigate the suitability of touchscreen gestures for user identification. We designed an experiment for collecting touch screen gestures and collected data from 21 users. The people, who took part in this experiment, had to solve such tasks on Android devices, which required up-down and left-right scrolling (strokes, gestures). 20 distinctive features were extracted from each stroke and different classification algorithms were tested using these data, demonstrating the viability of the features for biometric identification. The clustering classifier achieves 86.13% accuracy using 5 consecutive strokes for testing.

Összefoglaló

A dolgozatban megvizsgáltuk az érintőképernyős gesztúrák alkalmazhatóságát biometrikus azonosításra. Egy olyan adatgyűjtéssel összekötött kísérletet végeztünk, amelyben 21 felhasználó vett részt. A felhasználóknak Android eszközökön olyan feladatokat kellett megoldaniuk, amelyekhez vízszintes és függőleges gesztúrákat kellett használniuk. Az érintési adatokból gesztúrákra vonatkozó jellemzőket vontunk ki, amelyek alapján különböző osztályozási algoritmusokkal bizonyítjuk ezek alkalmazhatóságát biometrikus azonosításra. A legmagasabb pontosság 86,13%-os, amelyet 5 egymást követő gesztúra együttes osztályozása során kaptunk klaszterezést használva.

Kulcsszavak: biztonság, mobil biometria, Android, adatbányászat.

1. Bevezetés

A mobil eszközeinket másképpen használjuk, mint az asztali számítógépeket. Rendszerint egy ilyen eszközt naponta többször, de rövid ideig veszünk igénybe, ezért az asztali számítógépeken elterjedt belépési pontoknál használt felhasználónév/jelszó pároshoz kötött autentikációs sémák nem megfelelőek. A mobil eszközökhöz kifejlesztett képernyőzáró módszerek is nagyon sok felhasználót zavarnak, ezért ezeket is rendszerint kikapcsolják a készülékek tulajdonosai, ezáltal lehetővé téve a telefonon tárolt erőforrások elérését jogosulatlan felhasználók számára is. Sokkal kézenfekvőbb lenne, ha az okos eszközeink képesek lennének folyamatosan figyelni a felhasználót és érzékelni, ha az eszközt nem a tulajdonosa használja. Ezt a folyamatot folytonos autentikációnak nevezzük. Folytonos autentikációra több kísérlet is történt, ezek közül az egyik legelterjedtebb módszer a billentyűzési ritmus figyelése [2][3][4]. A biometrikus módszerek közül nem mindenik alkalmas folyamatos azonosításra. Például az ujjlenyomat vagy retina alapú felismerések nem alkalmasak, de használható az arcfelismerés asztali számítógép esetében, vagy a járásazonosítás mobileszközökön. Érintőképernyővel rendelkező mobileszközökön a képernyő érintési adatokból biometrikus jellemzők vonhatók ki, amelyek alapján szintén megvalósítható a folytonos autentikáció. A dolgozat írása pillanatában egyetlen tanulmányt találtunk, amely érintési adatok alapján folytonos autentikációt valósít meg [1]. Dolgozatunkban követtük a Frank és tsai. [1] által leírt adatgyűjtési kísérletet, azzal a különbséggel, hogy a felhasználók jelen esetben egyetlen munkamenet során szolgáltatták az adatokat, míg a Frank és tsai. által végzett kísérletben több munkamenetben. Amíg a Frank és tsai. dolgozatban az adatgyűjtést csak mobiltelefonon végezték, addig mi egy telefont és egy tabletet is használtunk az adatgyűjtéshez, így lehetővé vált, hogy kipróbáljuk a módszer érvényességét nagyobb képernyőjű eszközökön is.

A dolgozatban először az adatgyűjtést mutatjuk be, amelyet a jellemzők kinyerése követ. A rákövetkező rész a mérési eredményeket szemlélteti, legvégül pedig levonjuk a következtetéseket.

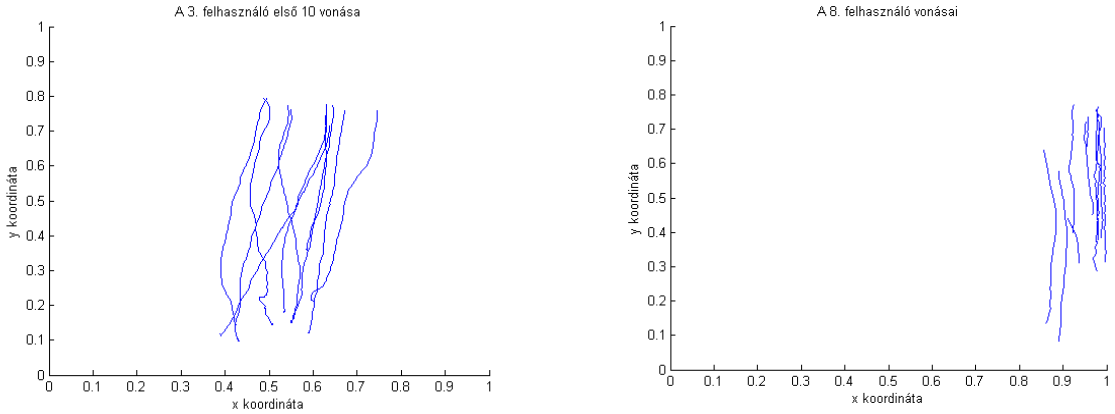
2. Adatgyűjtés

Az adatgyűjtéshez olyan mobil eszközökön elvégzendő feladatokra van szükség, amely során a felhasználónak nagyon sok függőleges, valamint vízszintes görgetési műveletet kell elvégeznie. A függőleges görgetésekhez kitűnő feladat a nagyobb méretű szövegolvasás szövegértéssel egybekötve. Ehhez a feladathoz egy nagyon egyszerű szöveget használtunk.

A vízszintes görgetési feladathoz a Frank és tsai. cikkben használt kép összehasonlítási feladatát használtuk. Ennél a feladatnál a felhasználónak két hasonló kép között kellett megtalálni a különbségeket. Egyszerre csak egy kép volt látható a képernyőn, a másik kép megtekintéséhez vízszintes görgetést kellett végezni.

A kísérletben 21 felhasználó vett részt, ebből 7 férfi és 14 nő. A kísérletben résztvevő személyek átlagéletkora 29.8 év, a szórása pedig 9.08, a legfiatalabb 21, a legidősebb pedig 47 éves volt.

Az 1. ábra két felhasználó első 10 függőleges vonásait szemlélteti.



1.ábra *Két felhasználó első 10 függőleges vonása*

Az adatgyűjtést két Android készülék segítségével végeztük: Google Nexus S telefonkészülék (480 x 800, 4.0 coll), illetve Asus Nexus 7 tablet (800 x 1280, 7.0 coll).

A görgetés során a felhasználó érintési adatokat szolgáltatott. Minden érintési pontról a következő adatokat tároltuk: x koordináta, y koordináta, vízszintes irányú sebesség, függőleges irányú sebesség, eseménykód (lenyomás, húzás, felengedés), nyomás, az ujj által lefedett terület, illetve a készülék helyzete (vízszintes vagy függőleges). Ezeket az adatokat a standard Android API segítségével nyertük ki.

3. Jellemzők kinyerése

A begyűjtött adatokat gesztúrákra (vonásokra) osztottuk. Egy vonáshoz tartoznak azon képernyőpontok, amelyek egy lenyomás és egy felengedés között helyezkednek el. Az érintési képernyőpontok együttesen egy olyan pályát alkotnak, amelynek elemei a következő alakú vektorok:

$$v^k = \left(x_i^k, y_i^k, v_{xi}^k, v_{yi}^k, t_i^k, p_i^k, A_i^k, o_i^k\right), i \in \{1, 2, \ldots, N^k\} \quad (1)$$

ahol v^k a k. vonás, x_i^k, illetve y_i^k az érintési pozíció, v_{xi}^k, v_{yi}^k a sebesség vízszintes, illetve függőleges irányú összetevője, t_i^k az időbélyeg, p_i^k a pontban mért nyomás, A_i^k az ujj által lefedett terület, o_i^k a készülék helyzete (vízszintes vagy függőleges), N^k a vonáshoz tartozó pontok száma. Az egyes vonásokhoz különböző számú pont tartozik.

Minden egyes vonásból egyetlen jellemzővektort állítottunk elő, amelynek elemeit a következő táblázatban foglaltuk össze:

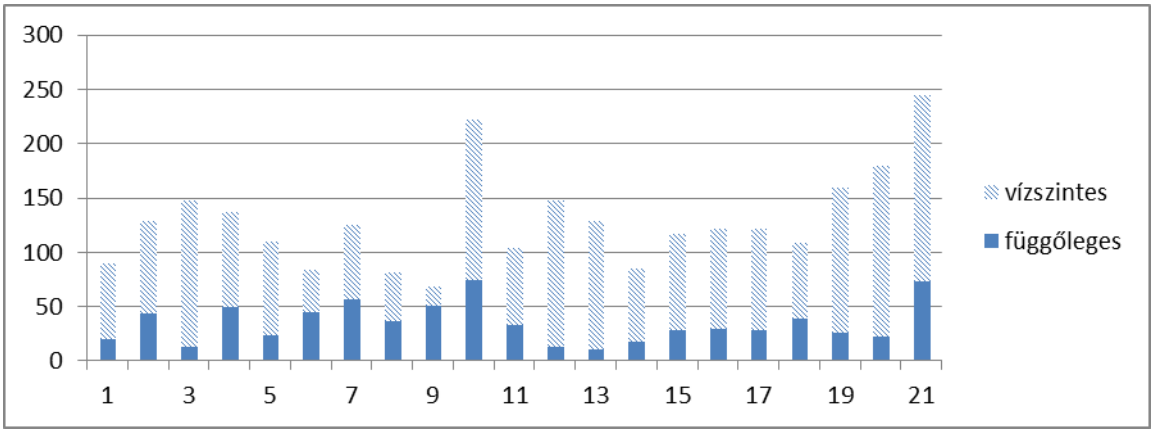
Sor-szám	Attribútum (jellemző)	Magyarázat
	user id	Kihez tartozik a vonás
	doc id	{1, 2} 1- szövegolvasás, 2 - képhasonlítás
1	inter stroke time	Két egymást követő vonás között eltelt idő
2	stroke duration (ms)	Vonás időtartama (ezredmásodpercben)
3	start x	Kezdőpont: x koordináta
4	start y	Kezdőpont: y koordináta
5	stop x	Végpont: x koordináta
6	stop y	Végpont: y koordináta
7	direct-end-to-end distance	A két végpont által meghatározott szakasz hossza
8	R mean resultant length	A vonás egyenessége
9	up/down/left/right flag	A vonás irányítottsága
10	direction of end-to-end line	A két végpont által meghatározott szakasz irányítényezője
11	device id	A készülék azonosítója. 1-telefon, 2-tablet
12	largest deviation from end-to-en-line	A maximális eltérés (elhajlás) a végpontokat összekötő szakasztól
13	average direction	Az útvonalat alkotó szakaszok irányítényezőinek átlaga
14	length of trajectory	A vonás hossza
15	average velocity	A vonás átlagsebessége
16	mid-stroke pressure	A vonás közepének nyomása
17	mid-stroke area covered	A vonás közepén az ujj által lefedett terület
18	average veolcity at first 5 points	A vonáshoz tartozó első öt pont átlagsebessége
19	average velocity at last 5 points	A vonáshoz tartozó utolsó öt pont átlagsebessége
20	phone orientation	A készülék helyzete: 1-vízszintes, 2-függőleges

1. táblázat *Vonásokra vonatkozó attribútumok*

A 8. attribútum meghatározza, hogy mennyire egyenes a vonás. Egyenes vonal esetén a jellemző értéke 0. A vonáshoz tartozó N pont meghatároz N-1 darab szakaszt (x_n, y_n), illetve (x_{n+1}, y_{n+1}) végpontokkal. Minden egyes szakasznak megfeleltetjük a $z_n = e^{i\phi_n}$ irányvektort, majd a (2) képlettel számítjuk a vonás egyenességét:

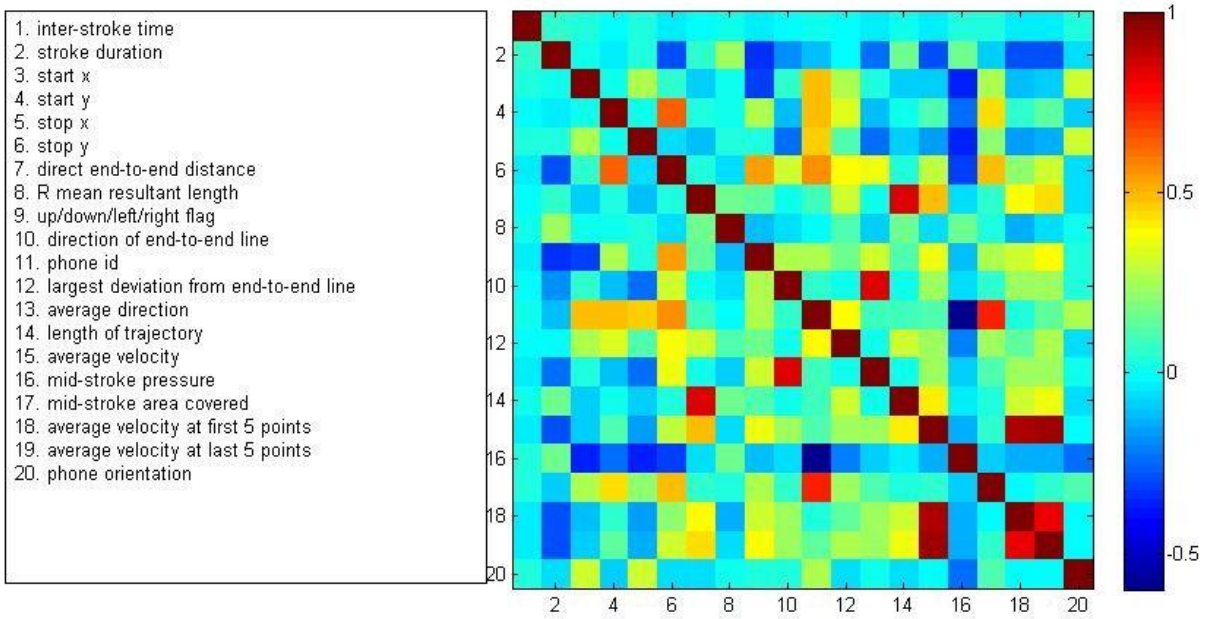
$$R = \frac{|\sum_{n=1}^{N-1} z_n|}{N-1} \tag{2}$$

A 12. attribútumnál kiszámítottuk minden egyes vonáshoz tartozó pont távolságát a vonás két végpontját összekötő szakasztól, majd ezek közül kiválasztottuk a legnagyobbat.
A 2. ábra a vízszintes és függőleges vonások számát szemlélteti minden egyes felhasználóra.



2.ábra *A 21 adatszolgáltató felhasználók vonásainak száma*

A jellemzőkre kiszámítottuk a korrelációs mátrixot, amely a jellemzők páronkénti korrelációját tartalmazza. Ezt a mátrixot a 3. ábra szemlélteti. Ahogyan várható volt, a legkorreláltabb jellemzők a vonás sebességére vonatkozók (15., 18., 19. attribútumok), tehát a vonás elejének és végének sebessége ugyanúgy változik, mint a vonás átlagsebessége. Magasan korrelált a vonás két végpontját összekötő szakasz iránya (10. attribútum) és a vonás átlagos iránya (13. attribútum - a vonás pontjai által meghatározott szakaszok átlag irányténye zője). Egy másik magas korreláció a vonás átlagos sebessége (15. attribútum) és a készülék azonosító (11. attribútum) között van. Ez azt jelenti, hogy a képernyő méretének növekedése maga után vonja a gyorsabb vonásokat. Ahogyan várható volt, a vonás végpontjait összekötő szakasz hossza (7. attribútum) és a vonás hossza (14. Attribútum - a pontok által meghatározott szakaszok hossza). A vonás kezdő- és végpontjának `y` koordinátája közötti korreláció arra enged következtetni, hogy az azonos irányú (vízszintes vagy függőleges) vonások valamennyire párhuzamosok egymással.



3. ábra *Attribútumok páronkénti korrelációja*

4. Mérési eredmények

A méréseket a Weka [5] programcsomaggal végeztük. Az osztályozási algoritmusok esetében a Weka standard felületét használtuk, a klaszterezéssel kapcsolatos méréseket pedig Java programmal végeztük, amelyben a Weka API-t használtuk.
A méréseket két attribútum halmazra mutatjuk be:

(a) az előző szekcióban bemutatott 20 attribútumra;
(b) a legjobb 14 attribútumra.

A legjobb 14 attribútum kiválasztásához először elkészítettük a 20 attribútumra vonatkozó korrelációs mátrixot (3. ábra). Ebből meghatároztuk, hogy melyek a magasan korrelált attribútumok, ezután a Weka programcsomaggal elvégeztük az attribútumok rangsorolását. Itt az *InfoGain+Ranker* rangsorolást használtuk, amely az attribútumokat aszerint rangsorolja, hogy milyen mennyiségű információval járult hozzá az adott attribútum az osztályozáshoz. A rangsorolás eredményét a 2. táblázatban foglaltuk össze.

1.3887	mid-stroke area covered
1.0345	start x
0.9987	phone id

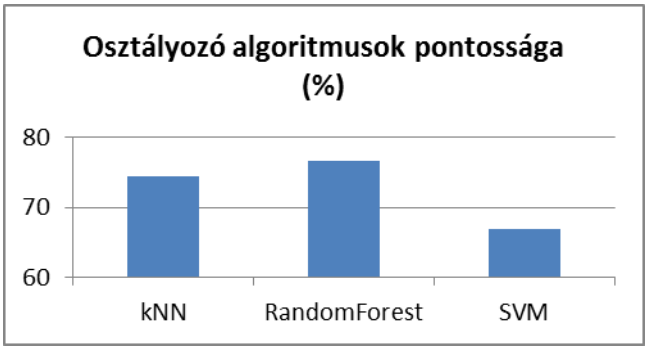
0.9983	stop x
0.8874	start y
0.8595	mid-stroke pressure
0.8529	stop y
0.7655	length of trajectory
0.7554	**direct end-to-end distance**
0.4481	largest deviation from end-to-end line
0.4295	direction of end-to-end line
0.4256	average velocity
0.3724	stroke duration (ms)
0.3356	**average velocity at last 5 points**
0.3213	phone orientation
0.263	**average velocity at first 5 points**
0.1971	inter stroke time
0.1905	**average direction**
0.1556	**up/down/left/right flag**
0.1173	**R mean resultant length**

2. táblázat *Attribútumok rangsorolása*

Figyelembe véve az attribútumok korrelációját és a rangsorolását, a fenti táblázatból a vastagított 6 attribútumot kizártuk. A sebességre vonatkozóan a teljes vonás átlagsebességet hagytuk meg, az első 5, illetve az utolsó 5 pont átlagsebességét kizártuk. A vonás hossza és a vonás két végpontja közötti szakasz hossza is erősen korreláltak, ezért a másodikat kizártuk. Az irányítottságok közül a vonás két végpontját összekötő szakasz irányítottsága teljesített jobban, ezért az átlag irányítottságot kizártuk. Az utolsó két leggyengébben teljesítő attribútumot is kizártuk.

4.1. Egyedek osztályozása

A méréseket a k-NN, RandomForest [6] valamint SVM osztályozókkal végeztük. Az attribútumok szelekciója nem mutatott javulást az osztályozási rátában, ezért a mérési eredmények az eredeti 20 attribútumra vonatkoznak. A k-NN esetében az 1-NN teljesített a legjobban, az alábbi diagram ezt szemlélteti. A legjobb osztályozási pontosságot a RandomForest algoritmus nyújtotta. Minden mérés esetében 10 keresztvalidációt alkalmaztunk.



4. ábra *Osztályozási algoritmusok pontossága*

4.2. Egyed szekvenciák osztályozása

Az előző mérésekkel ellentétben, ezeket a méréseket saját programmal végeztük. Erre azért volt szükség, mert a Weka a szabványos interfészen keresztül csak egyedek osztályozását teszi lehetővé, nekünk viszont egyed szekvenciák osztályozását is el kellett végezni. A mi esetünkben minden egyed egyetlen vonást jelent, amely lehet függőleges, illetve vízszintes.

Legyen `N` a különböző felhasználók száma. Egy felhasználói profilt az `M` darab klaszter középérték alkotja. A középértékek `D` dimenziós vektorok, ahol `D` az attribútumok száma.

$$\lambda_i = \left(m_i^{(1)}, m_i^{(2)}, ..., m_i^{(M)}\right), \quad i = 1,2,...N \tag{3}$$

Legyen `X, T` darab egymást követő vonások szekvenciájának megfelő egyedek.

$$X = \{x_1, x_2, ..., x_T\}, \quad x_i \in R^D \tag{4}$$

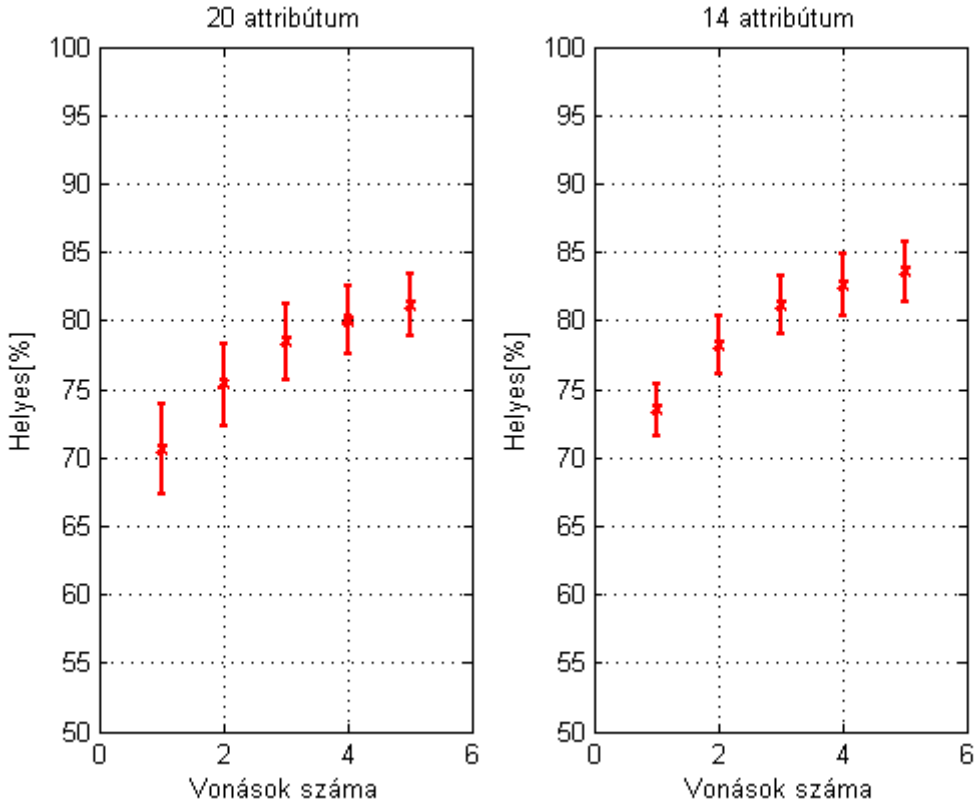
Az `X` szekvencia távolságát az `i.` felhasználó profiljától a következő képlettel számoljuk:

$$d(X, \lambda_i) = \frac{1}{T} \sum_{k=1}^{T} \min_{j=1..M} d_E(x_k, m_i^{(j)}) \tag{5}$$

Ahol d_E az Euklideszi távolság. A szekvenciát abba az osztályba soroljuk be, amelytől a szekvencia távolsága a legkisebb:

$$Id = \arg \min_{i=1..N} \{d(X, \lambda_i)\} \tag{6}$$

A klaszterezés esetében az adatok 2/3-át tanításra, 1/3-át pedig tesztelésre használtuk. Arra is vigyáztunk, hogy ez az arány érvényes legyen a függőleges és a vízszintes vonásokra is. A 3. ábra tartalmazza az osztályozási eredményeket, ahol a klaszterek száma rendre 8, 16, 32, 64.



4. ábra *Osztályozási pontosságok különböző vonás-, illetve klaszterszámok mellett*

A vonások számát 1-től 5-ig változtattuk. Látható, hogy az attribútumok szelekciója (14 attribútum) megemelte az osztályozási pontosságot. A legjobb eredményt (86,13%) 5 darab vonás osztályozásával kaptuk 64 klasztert használva. Minden esetben a k-Means klaszterezési algoritmust használtuk.

5. Következtetések

Dolgozatunkban megvizsgáltuk mobil érintőképernyőkről gyűjtött érintési adatok (gesztúrák, vonások) alkalmazhatóságát biometrikus személyazonosításra. Elkészítettünk egy olyan adatgyűjtő programot, amely vízszintes és függőleges vonásokat (gesztúrákat) gyűjt be a felhasználóktól, miközben ezek feladatokat oldanak meg a mobil eszközöket használva. A gesztúrákból 20 féle jellemzőt nyertünk ki, mint például a gesztúra (vonás) kezdeti, illeti végpontja, a vonás egyenessége, vagy például a nyomás erőssége. A felépített jellemzővektorokat a Weka adatbányászati programcsomag különböző algoritmusai segítségével osztályoztuk, megállapítva a kinyert jellemzők alkalmazhatóságát folyamatos biometrikus azonosításra. Az osztályozáshoz k-NN, RandomForest, SVM, illetve klaszterezést használtunk. A Frank és társai eredményeit kiegészítettük azzal a

következtetéssel, hogy az érintési adatok alapján történő azonosítás nagyobb méretű képernyők esetén is jól működik, illetve azzal is, hogy az osztályozási pontosság növelhető több vonás együttes osztályozásával.

Köszönetnyilvánítás

Az adatgyűjtő programot *Marton-Miklós László* a Sapientia EMTE, Marosvásárhely-i Kar III. éves Informatika szakos hallgatója készítette, aki az adatgyűjtésben is részt vett.

Hivatkozások

[1] Frank, M., Biedert, R., Ma, E., Martinovic, I., Song, D. **(2013).** Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. IEEE Transactions on Information Forensics and Security, 8(1), 136-148.

[2] Killourhy, K. S., Maxion, R. A. **(2009).** Comparing anomaly-detection algorithms for keystroke dynamics. In IEEE/IFIP International Conference on Dependable Systems and Networks (DSN2009), (June 29–July 2, 2009, Estoril, Lisbon, Portugal), 125–134, IEEE Computer Society, Los Alamitos, CA.

[3] Monrose, F., Rubin, A. **(1997).** Authentication via Keystroke Dynamics. In Proceedings of the Fourth ACM Conference on Computer and Communication Security, Zurich, Switzerland.

[4] Bours, H., Barghouthi, P. **(2009).** Continuous Authentication using Biometric Keystroke Dynamics, 1-12. In *Norwegian Information Security Conference*.

[5] Hall,M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H. **(2009).** The WEKA Data Mining Software: An Update; SIGKDD Explorations, 11(1), 10-18.

[6] Breiman, L. **(2001)**. Random Forests. Machine Learning 45 (1).