Alapítva: 2001

INFORMÁCIÓS TÁRSADALOM

SZAKTUDOMÁNYOS FOLYÓIRAT

Gulyás Gábor György: Gépi tanulási módszerek alkalmazása deanonimizálásra

Hivatkozás/reference:

Gulyás Gábor György, „Gépi tanulási módszerek alkalmazása deanonimizálásra”, Információs Társadalom, XVII. évf. (2017) 1. szám, 72-86. old.
http://dx.doi.org/10.22503/inftars.XVII.2017.1.5

Számos olyan adathalmaz áll a rendelkezésünkre, amelyek jelentős üzleti és kutatási potenciált hordoznak. Azonban – gondoljunk például a hordozható eszközök által gyűjtött egészségügyi adatokra – a hasznosítás mellett kiemelkedő kockázati tényező a privátszféra sérülése, amelynek elkerülésére többek között anonimizálási algoritmusokat alkalmaznak. Jelen tanulmányban az anonimizálás „visszafordítására” szakosodott algoritmusokat, az úgynevezett deanonimizációs eljárásokat, illetve azoknak egy speciális és újnak tekinthető szegmensét tekintjük át, amelyeknél gépi tanulási eljárásokat alkalmaznak a robusztusság, illetve a hatékonyság növelése érdekében. A tanulmányban a privátszféra-sértő üzleti célú támadások és a biztonsági alkalmazások hasonlóságára is rámutatunk: ugyanaz az algoritmus hogyan tud biztonsági indokkal a privátszférával szemben dolgozni, kontextustól függően.
Kulcsszavak: anonimitás, deanonimizálás, gépi tanulás, privátszféra védelme

Using machine learning techniques for de-anonymization

Today we have unprecedented access to datasets bearing huge potential in regard to both business and research. However, beside their unquestionable utility, privacy breaches pose a significant risk to the release of these datasets (e.g., datasets originating from healthcare are good examples), thus service providers must use anonymization techniques to minimize the risk of unwanted disclosure. In this study, we focus on de-anonymization attacks, algorithms that are designed to “reverse” the anonymization process. In particular, we focus on a novel segment of these attacks that involve machine learning to improve robustness and efficiency. Furthermore, we highlight and discuss the similarity between de-anonymization and authentication: how can these algorithms, which are generally perceived as unethical, be used legitimately for security reasons under special constraints.
Keywords: anonymity, de-anonymization, machine learning, private sphere protection

Gulyás Gábor György

Gépi tanulási módszerek alkalmazása deanonimizálásra

Bevezető

Az okostelefonok, a különféle szenzorok elterjedése, az internetnek a magánéletbe való szoros integrációja megkönnyíti a mindennapi életet és munkát. Ezekből a forrásokból rendkívüli potenciállal rendelkező adatok származnak, amelyek üzleti, kutatási vagy akár a nyílt adatok elvén működő kormányzati szolgáltatások szempontjából korábban sosem látott lehetőségeket tartogatnak a társadalmak számára.

Az előnyök mellett azonban nem elhanyagolható problémát jelenthet az efféle adatgyűjtésnek a privátszférára gyakorolt hatása sem, hiszen már nem csupán az állami szereplők – köztük a nyomozó szervek – számára könnyíti meg a megfigyelést, új szereplőként megjelennek a technológiát létrehozó cégek és maguk a felhasználók is. Ebben az új rendszerben a cégek adatokat gyűjtenek felhasználóikról, amit által a szolgáltatásaikat fejleszthetik, az adatok értékesítésével további bevételekhez juthatnak, illetve a technológia megkönnyíti, hogy felhasználók egymás után leskelődjenek, sőt kontrollálják is egymást. Méltán nevezhetjük a létrejövő rendszert kukkoló társadalmaknak (Székely 2010).

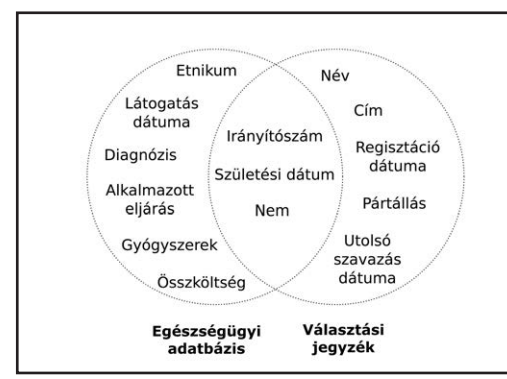
A kölcsönös „kukkolási" probléma enyhítésére a technológiai cégek privátszférát védő beállításokat fejlesztettek ki és tették elérhetővé felhasználóik számára. Például ilyenek a különféle, részletesen szabályozható láthatósági listák, amelyekkel üzenetek, profilok láthatóságát lehet beállítani. Ez azonban a szolgáltató oldaláról történő adatgyűjtés esetére nem oldja meg a privátszféra védelmét, amit indokol az adatokat kezelő cégek önvédelme, illetve megkövetelheti is ezt a jogi környezet. Az előbbire jó példát ad az America Online (AOL) esete, amikor 657 ezer felhasználójának három havi keresési előzményét tettek közzé kutatási célból, és emiatt a cég kénytelen volt perrel és a sajtónyilvánosságból fakadó presztízsveszteséggel szembenézni (Bangeman 2006). Az utóbbira pedig példa lehet az Európai Bizottság 95/46/EC direktívája, amely szerint az anonimizált adatokra már nem vonatkoznak az európai adatvédelmi irányelvek; így például az ilyen adatbázisok megosztása (eladása) elé kevesebb akadály hárul.

A védekezési módszerek legegyszerűbb formája a pszeudonimizálás (angolul *pseudonymization*), amely során megfosztják az adatokat az egyértelmű személyes adatoktól (mint nevek, felhasználónevek és egyéb azonosítók), és pszeudonim azonosítókra cserélik azokat (például véletlen számokra). Ennek célja, hogy az egyes rekordok ne legyenek triviális módon kapcsolatba hozhatóak az eredeti adatalanyokkal. Már több esetben is láthattunk rá „éles" demonstrációt, hogy ezek az eljárások nem megfelelőek (például Narayanan és Shmatikov (2008) és Barbaro (2006)). Az AOL előbb említett esete is ide tartozik: az AOL módosítás nélkül, de pszeudonim formában osztotta meg felhasználóinak körülbelül 20 millió keresésének szövegét. Ennek ellenére a keresések szövege alapján mégis be lehetett azonosítani egyes, a kereséseket végző személyeket, és ez nem volt különösebben bonyolult: a New York Times riporterének sikerült visszakövetnie és meg is szólaltatnia a 4417749-es számú felhasználót (Barbaro 2006).

Az AOL esetében a problémát egyértelműen az okozta, hogy ugyan a keresések nem kötődtek a kereső személyéhez, mégis minden egyes keresés közelebb vitt hozzá. Az anonimizálási eljárások (angolul *anonymization*) célja hasonló a pszeudonimizáláshoz, de az egyértelmű azonosítók eltávolításán túl az is elvárt, hogy az anonimizálás során létrejövő rekordokat ne lehessen hozzákötni az adatot szolgáltató eredeti személyhez, vagy hasonló módon az újságíró módszeréhez „szűkíteni a kört". Ez már érinti az AOL esetében látott információszivárgásokat is, ezért az anonimizálási eljárások az adatbázis nem azonosító jellegű mezőinek értékét is megváltoztatják.

A több, nem azonosító jellegű mező összevonásából létrejövő, úgynevezett kvázi-azonosítók kulcsfontosságúak a privátszféra védelme szempontjából. A kvázi-azonosítók segítségével összekapcsolhatóak különböző adatbázisok, és így az anonimizálás is visszafordíthatóvá válik, elég hozzá egy anonimizált adatbázisban szereplő rekordokat a kvázi-azonosítók mentén azonosítóval rendelkező rekordokkal párosítani. Ez utóbbit hívjuk deanonimizálásnak vagy újraazonosításnak (angolul rendre *de-anonymization* és *re-identification*).

Latanya Sweeney (2002) tanulmánya volt az egyik első prominens példa, ami felhívta a figyelmet a kvázi-azonosítók és újraazonosítás problémájára. Kutatóként hozzáférést kapott 135 ezer állami dolgozó és családja névtelen egészségügyi adataihoz, majd megvásárolta a Massachusetts-ben regisztrált szavazók listáját (20 dollárért). Az irányítószám, születési dátum és nem mezőkből formált kvázi-azonosítóval össze tudta kötni a két adatbázis rekordjainak egy részét, amelyet Massachusetts kormányzója egészségügyi adatainak kikeresésével demonstrált (1. ábra).



1. ábra: Latanya Sweeney a név nélkül módon publikált egészségügyi adatokat a választási jegyzék segítségével kompromittálta: a két adatbázist az irányítószám, születési dátum és nem mezők segítségével össze lehetett vonni (Sweeney 2002)

Sweeney az úgynevezett k-anonimitás (angolul *k-anonymity*) anonimizálási módszert javasolta az újraazonosítással szemben: a k-anonimitás akkor teljesül egy adatbázisra, ha minden egyes rekordjához tartozik legalább k-1 olyan másik rekord, amelyeknek a kvázi-azonosítója megegyezik. A k-anonimitás célja, hogy ha valaki össze is tudna kötni a kvázi-azonosítóval két adatbázist, akkor is csak legfeljebb 1/k valószínűséggel tudja a rekordokat helyesen összekapcsolni.

Sweeney (2002) munkája után számos tudományos cikk jelent meg, amelyek a módszer hibáit igyekeztek javítani, további adattípusra javasoltak anonimizálási eljárásokat, vagy éppen deanonimizálási algoritmusokat. E tanulmányban a deanonimizálási eljárások egy új típusát vizsgáljuk, nevezetesen azokat, amelyek gépi tanulási módszerekre (angolul *machine learning*) épülnek. A gépi tanulási módszerek automatizált adatelemzési módsze-

rek, amelyek során az adat modellezését az algoritmus a minták alapján maga tanulja meg (például kellő minta esetén fel tudja ismerni addig nem látott képeken is a macskákat). Pontosan ez a tulajdonságuk teszi a gépi tanulási módszereket vonzóvá az újraazonosítási támadásokban is: alkalmazásuk esetén például nem szükséges a tervezőnek pontosan értenie, hogy a két adathalmaz egyes rekordjai a különféle attribútumok alapján miképpen hasonlítanak egymásra (és így hogyan köthetőek össze) – az összerendelést végző függvényt a gépi tanulási módszer automatikusan képes megtalálni.

A deanonimizálási algoritmusok „történelme"

Sweeney demonstrációja a táblázatos adatokra és a k-anonimitás nagy visszhangot váltott ki, több száz művet inspirálva a következő években. A táblázatos adatok azonban csak egy speciális esetét képviselik az új technológiákból származó adatbázisoknak, ugyanis csak az esetek kisebb részében beszélhetünk egyáltalán relációs adatbázis jellegű felépítésről, vagy zárt attribútumhalmazról, amelyek azonosítókat tartalmaznak vagy kvázi-azonosítókként felhasználhatók. Az esetek többségében az egyes rekordokat leíró mezők (vagy attribútumok) száma nem zárt, folyamatosan bővül és nagy számú. Gondoljunk például egy webáruházra, ahol a felhasználók értékelik a termékeket: ez esetben a termékek száma nagy és folyamatosan nő, és a felhasználók jellemzően csak a termékek kis töredékét értékelték valaha. Ugyanilyen elrendezést kapunk, ha mondjuk egy közösségi hálózat kapcsolatrendszerének gráfját szomszédsági mátrixszal írjuk le. Ezeket nagy attribútumszámú vagy nagy dimenziójú adathalmazoknak hívjuk.

Viszonylag korán kiderült, hogy ezekben az esetekben a k-anonimitás nem megfelelő védekezési eljárás és legfeljebb csak kompromisszumot lehet keresni az anonimitás szintje és az adatbázis hasznossága között (Aggarwal 2005). Ennek oka az, hogy ahogy nő az attribútumok száma, úgy nő a potenciális kvázi-azonosító kombinációk száma is, ami miatt a rekordok egyre kevésbé hasonlítanak egymásra, és ez megnehezíti az anonimizálást, hiszen a k-anonimizálás csak sok attribútum törlésével lesz lehetséges.

A táblázatos adatoktól a tetszőleges struktúra felé

A Netflix cég a 2000-es évek elején DVD kölcsönzési tevékenységet folytatott, és rendszerének egyik kulcskomponense a Cinematch elnevezésű ajánlórendszere volt, amely a felhasználó értékeléseit figyelembe véve ajánlott számára további filmeket. A Netflix a Cinematch algoritmus továbbfejlesztésére versenyt indított, amelyhez 2006 októberében közzé tette körülbelül félmillió felhasználójának 1998 október és 2005 december közötti értékeléseit (Bennett és Lanning 2007). A cég közleményében is hangsúlyozta, hogy az adatokat név nélkül, azonosítókkal ellátva tették közzé, és az értékeléseket is kis mértékben módosították, hogy konkrét személyek értékeléseit nehezebb legyen visszakeresni.

A Netflix adatbázis szolgáltatott alapot az első nem táblázatos, nagy attribútumszámú adatbázis deanonimizálás demonstrációjára (Narayanan és Shmatikov 2008). Narayanan és Shmatikov a Netflix adathalmaz inspirációjára javasolta a Scoreboard algoritmust, amely egy általános deanonimizálási sémát követ, így a Netflix-specifikus alkalmazáson túlmenően tetszőleges nagy dimenziójú adatra alkalmazható.

A Scoreboard feltételezi, hogy a támadó rendelkezik olyan D' adatbázissal a deanonimizálás célpontjairól (ez az úgynevezett háttértudás, vagy kiegészítő információ), amely legalább részben megtalálható az anonimizált D adatbázisban, és az sem kritérium, hogy ez pontos információ legyen. A Scoreboard összehasonlítja a háttérinformációban szereplő $r' \in D'$ rekordokat az anonimizált adat $r \in D$ rekordjaival, és pontozza a potenciális (r, r') párosításokat a hasonlóságuk alapján. A pontozásban a kevésbé gyakori jellemzők nagyobb hangsúlyt kapnak, ami a filmes vonatkozásban könnyen értelmezhető, hiszen például az kevesebbet árul el valakiről, hogy látta a Men in Black-et, mint mondjuk a Citizenfour-t. Ezután az algoritmus a legnagyobb pontszámú r'' rekordot jelöli ki a deanonimizálás eredményének, hogyha annak pontszáma a többi potenciális jelölthöz képes kellően kiemelkedő.

Az algoritmus tesztelésénél először azt ellenőrizték, hogy egy-egy véletlenszerűen választott felhasználót jól be tud-e azonosítani az algoritmus, majd pedig azt, hogy ha törlik az adatbázisból, ezt képes-e jelezni (téves alternatívák ajánlása helyett). Az előbbihez a felhasználótól mindössze 2-8 értékelést választottak ki a Scoreboard számára mint háttérinformációt (a felhasználók túlnyomó többsége 20 vagy több értékeléssel rendelkezik az adatbázisban). Ha a film pontozása pontosan ismert volt (1-5 csillag), a dátum pedig ±3 és ±14 nap pontosságú, akkor az algoritmus már mindössze 5/6 értékelés (6 értékelésből 5 helyes) alapján több mint 80%-os bizonyossággal volt képes a helyes deanonimizálásra, illetve ha az adatbázisban nem szerepelt a rekord, akkor annak elutasítására. 7/8 értékelésnél a deanonimizálás esélye bőven 90% felé nőtt. A kiegészítő információ pontatlanságát jól lehet ellensúlyozni az értékelések számával. Ha a film pontozásánál ±1 csillag eltérés volt megengedett, a dátumnál pedig a ±14 nap pontosság, akkor a 4/8-8/8 értékelések esetén a deanonimizálás valószínűsége körülbelül 60%-95% között mozgott.

A Scoreboard algoritmus akkor hatékony, hogyha megfelelő háttérismerettel rendelkezik a támadó, ennek feltárása azonban nem igényel különösebb nyomozói tevékenységet. Narayanan és Shmatikov munkájukban cáfolták, hogy a Netflix által kiadott adathalmaz jelentősen módosítva lenne: egyrészt két ismerősüknél, akiket megtaláltak az adatbázisban, mindössze legfeljebb 1/306 és 5/229 értékelés tért el az eredetitől, illetve a módosítást az adathalmaz statisztikai jellemzői sem támasztották alá túl meggyőzően. Ez viszont azt jelenti – tekintettel az algoritmus hibatűrési képességére –, hogy akár egy rövidebb munkahelyi csevegés vagy néhány értékelés az IMDb-n[1] is visszakereshetővé teszi azokat, akiknek értékeléseit a Netflix publikálta. (Ha pedig valaki mégsem szerepelne benne, az algoritmus ezt is hatékonyan jelzi.)

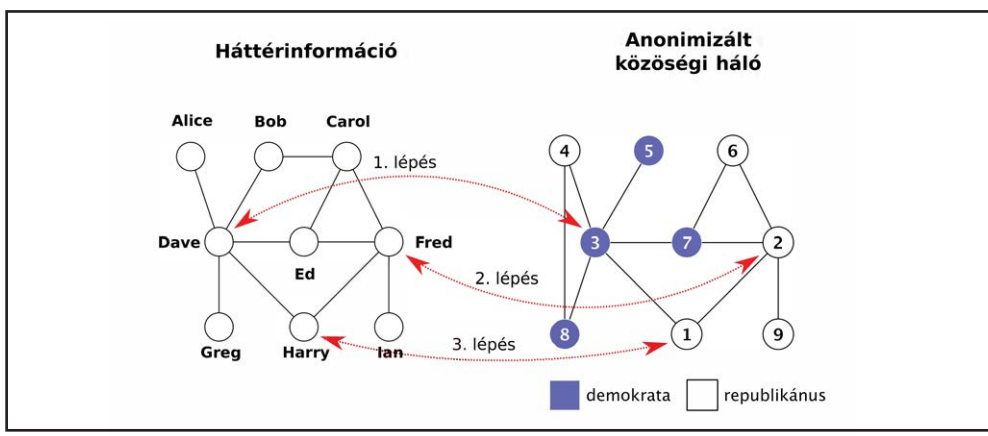
Hatékonyságnövelés: százezres közösségi hálózatok deanonimizálása

Bár a deanonimizálás valószínűsége (és pontossága) elég magas volt a Scoreboard algoritmus esetén, hatékonyságát jelentősen rontotta, hogy a háttérinformáció elemeit a teljes anonim adatbázissal összehasonlította. Ez a keresési séma nem működőképes, ha nem célzott támadásról van szó, hanem tömeges deanonimizálásról: azaz a Scoreboard csak addig hatékony, amíg a háttérinformáció néhány rekordot tartalmaz, de ha két teljes adatbázist kell összehasonlítani, akkor a számításigénye elfogadhatatlanul megnő.

[1] http://www.imdb.com

A másik probléma a fals-pozitív paradoxon, ami akkor áll fenn, ha a téves találatok aránya nagyságrendekkel több, mint a helyes. Ezt a filmes példánál maradva a következőképpen képzelhetjük el. Tegyük fel, hogy van egy a Scoreboardhoz hasonló algoritmus, amelynek ha megmutatunk egy $(r; r')$ felhasználó-párt ($r \in D$ anonim és $r' \in D'$ identitása ismert), akkor ha a párosítás helyes, az algoritmus 99% valószínűséggel ezt megmondja, míg hogyha helytelen, akkor 0,01% valószínűséggel téved csupán. Ha ezen feltételek mellett keresünk egy felhasználót az értékelései alapján egy 100 milliós halmazban, akkor az algoritmus körülbelül 10 000 találatot fog visszaadni, amiből azt az egy darab helyes találatot még ki kell valahogyan szűrni. (Ezért a Scoreboard-ban a legkevésbé egyező értékelés alapján diszkriminálták a téves találatokat, de ez a megközelítés nem mindig alkalmazható.) Éppen e miatt a probléma miatt kételkedhetünk az olyan projektek sikerében, amelyek a különféle bűncselekményeket tömeges megfigyeléssel kívánják megelőzni vagy visszaszorítani (Parra-Arnau és Castelluccia 2015).

Narayanan és Shmatikov (2009) a Scoreboardban alkalmazott alapelvek mentén javasoltak egy olyan algoritmust közösségi hálózatok deanonimizálására, amely ezeket a problémákat kiküszöböli, és emiatt akár két többszázezres (vagy nagyobb) közösségi hálózat deanonimizálását is képes hatékonyan és pontosan elvégezni – csupán a kapcsolatrendszer (gráf struktúra) figyelembevételével. Egyszerű trükköt alkalmaztak: mivel közösségi hálózatról van szó, és az egyes felhasználókat kapcsolatok kötik össze, az algoritmus futása során a már meglévő deanonimizálásokat is figyelembe vették.



2. ábra: Közösségi hálózat deanonimizálásának bemutatása. Először a globálisan kiugró felhasználókat párosítja (1-2. lépés), majd ezt követően a meglévő párok felhasználásával folytatja a többivel (3. lépés)

Az algoritmus működési elvét a 2. ábra segítségével mutatjuk be, amelyen látható egy nevekkel ellátott G' közösségi hálózat mint háttérinformáció, valamint egy G anonimizált közösségi hálózat. Mivel G tartalmaz egy érzékeny információt (politikai preferencia) a felhasználókról, a példában a támadó célja G deanonimizálása G'-vel. Az első lépésben az algoritmus a jellemzőik alapján globálisan kiugró felhasználókat keres, ez az inicializálási fázis. Ilyen például a Dave nevű felhasználó, amelynek összesen öt kapcsolata van (vagyis

az őt reprezentáló csomópontnak a fokszáma öt), ami G'-ben a legtöbb és így ez a jellemző Dave-et egyedivé teszi. A másik hálózatban a 3-as felhasználó szintén öt kapcsolattal rendelkezik és egyedi, ezért a támadó úgy veszi, hogy a 3-as felhasználó Dave-nek felel meg – azaz létrehoz egy párosítást a két közösségi hálózat felhasználói között. Ugyanilyen logika mentén létrehozza a (Fred, 2) párosítást is.

A következő fázis célja a meglévő párosítások felhasználása új párosítások kereséséhez. Erre szükség is van, hiszen – a példánál maradva – a kapcsolatok számának összehasonlítása nem minden esetben működik: például Harrynek két kapcsolata van, de ez igaz Bobra is. Azonban ha felhasználjuk a már meglévő párosításokat, akkor feltételezhetjük, hogy ahogy Harry Dave és Fred barátja G'-ben, úgy a neki megfelelő jelöltre is igaznak kell lennie, hogy 3 és 2 barátja G-ben és valószínűleg két kapcsolata van. Ez viszont csak az 1-es felhasználóra igaz, így az algoritmus létrehoz egy új párosítást (Harry, 1) között.

A meglévő párosításokat a példához hasonló módon használja fel a javasolt algoritmus a potenciális találatok szűkítésére (a gráf egészéről egy szűkebb körre): az adott G'-beli felhasználónak megfelelő találatról feltételezi, hogy az ismerőseik nagyjából ugyanazok, csak G-ben (Narayanan és Shmatikov 2009). A kisebb keresési tér így lényegében megoldja a hatékonysági és fals-pozitív problémákat is. A továbbiakban az algoritmus egyébként ugyanúgy működik, mint a Scoreboard: a potenciális találatokat pontozza (a koszinusz hasonlósághoz hasonlóan), és ha van kiemelkedő találat, akkor azt választja meg a deanonimizálási párosításhoz.

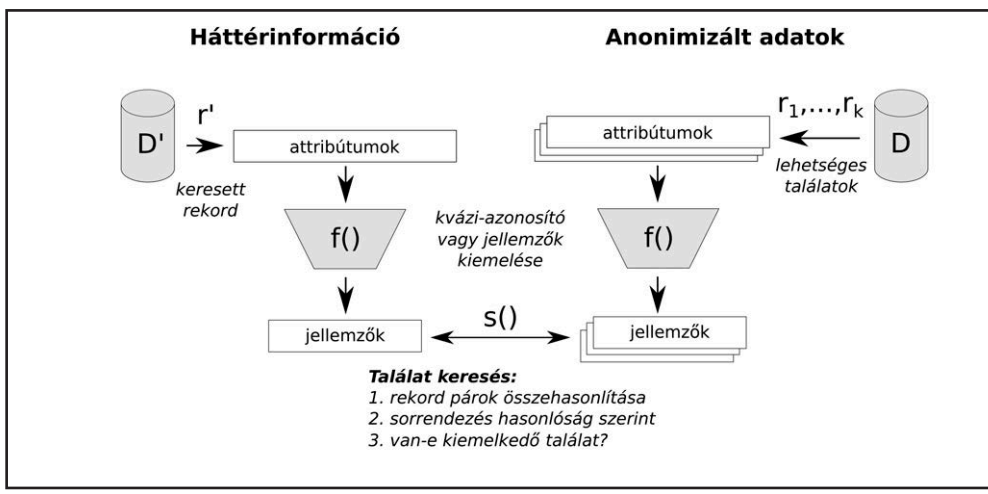
Az algoritmust élesben is tesztelték: egy 224 ezer felhasználóból álló Twitter kivonatot „deanonimizáltak” egy 3,3 millió felhasználóból álló Flickr kivonattal. A cél az volt, hogy az algoritmus párosítsa össze azokat a felhasználókat, amelyek mindkét kivonatban szerepelnek. Ehhez először, hogy az algoritmus eredményét ellenőrizni lehessen, meg kellett keresni valamilyen módszer alapján azokat a felhasználókat, akik ténylegesen mindkettőben szerepeltek. Ehhez felhasználták a felhasználónév, név és hely információkat, s ezekkel végül összesen 27 ezer felhasználót sikerült azonosítani mindkét adathalmazban. Az algoritmus inicializálásához kiválasztottak 150 felhasználót (akik rendelkeztek legalább 80 kapcsolattal). Az algoritmus a 27 ezer felhasználó 30,8%-át sikeresen megtalálta (ez a *felidézés*, a helyes találatok aránya az összes lehetséges találatokhoz képest), és csak 12,1%-nyi hibát vétett (ez a *hibaarány*: a hibás találatok száma az összes lehetséges találathoz képest). A későbbiekben is használni fogjuk a felidézés, precizitás és hibaarány fogalmakat; a precizitás a hibák aránya az összes találat között (angolul *precision*).

A 2009-ben publikált algoritmus az idő próbáját is kiállta. Egy 2015-ös cikkben szimulációs vizsgálatokkal összehasonlították az addig megjelent deanonimizáló támadásokat a legkorszerűbb anonimizálási eljárásokkal szemben (Ji et. al 2015), és eszerint egyetlen időközben publikált támadás sem múlta felül a Narayanan és Shmatikov által megalkotott algoritmus hatékonyságát. Az első, általában jobbnak tekinthető algoritmust Gulyás, Simon és Imre (2016) javasolta, ahol a 2015-ös összehasonlítást identikus módon megismételték, és az új algoritmust összemérték a korábbiak közül a legkiemelkedőbb eredményt nyújtóakkal. Ennek alapján az új algoritmus valamennyinél magasabb deanonimizálási arányt ért el alacsony hibaarány mellett.

Deanonimizálás gépi tanulás segítségével

A gépi tanulási módszereket jól körülhatárolható módon alkalmazzák a deanonimizálási eljárásokban. Az eddig tárgyalt példák alapján a következőképpen vázolhatjuk fel a deanonimizálási algoritmusok működési sémáját (3. ábra):

1. A támadás célja egy D anonimizált adathalmaz, melyből az egyértelmű azonosítók hiányoznak és az adatokat is módosították bizonyos mértékben. A támadó egyD' adathalmazt használ fel a D-beli rekordok deanonimizálására, amelyben a rekordok számossága adattípustól függően változó lehet.
2. A támadó az adat típusa alapján kiválasztja az $f(\cdot)$ és $s(\cdot)$ függvényeket. Az $f(\cdot)$ függvényt használja a rekordokból kvázi-azonosító előállítására, az $s(\cdot)$ függvény pedig két rekord kvázi azonosítóinak a hasonlóságát megadó függvény.
3. A támadó kiválaszt egy $r' \in D'$ rekordot, amelynek az anonimizált $r \in D$ párját keresi.
4. A támadó kiszámítja $f(r')$-et, majd valamennyi potenciális $r'' \in D$ rekordra szintén, és ezután kiszámolja a rekordok közötti $(f(r'), f(r''))$hasonlóságokat.
5. Ha van kiugró hasonlósággal bíró, vagy valamilyen más elfogadási kritériumnak megfelelő rekord, akkor ezt fogadja el $r=r''$ a helyes deanonimizálásnak.



3. ábra: A deanonimizálási eljárások jellemző sémája

A kvázi-azonosító kiválasztás a korábbi példákban bizonyos oszlopok kiválasztása volt (Sweeney 2002), illetve ennek felelt meg az egyes értékelések súlyozása a Netflix adathalmaz deanonimizálásában (Narayanan és Shmatikov 2008). Azonban ez nem mindig triviális feladat, például mit választanánk ki, ha hívásindítás és -fogadás idejéről és helyéről van egy adatbázisunk? Vagy mondjuk a telefon gyorsulás- és sebességmérőjéből származó információk alapján? Illetve az sem mindig egyértelmű, hogy a meglévő támadásokban használt választás a legoptimálisabb. Ezen okok miatt alkalmazták egyes esetekben a gépi tanulási módszereket a kvázi-azonosító, vagyis jellemző kiválasztására (angolul *feature selection*); azaz a $f(\cdot)$ függvényt helyettesítették olyan eljárásokkal, amelyek önállóan képesek azonosítani a releváns jellemzőket az adatban.

A másik tipikus alkalmazási terület az $s(\cdot)$ függvény helyettesítése gépi tanulási módszerekkel. Ennek az oka ugyanaz, mint az előbb: sok esetben nem ismert (különösen ha a jellemző kiválasztása is gépi módszerekkel történik), illetve előfordulhat, hogy gépi tanulással a szakértők által javasolt hasonlósági metrikánál jobbat lehet találni. A következő fejezetekben megvizsgáljuk, hogy különböző adattípusokra hogyan alkalmaztak gépi tanulási módszereket.

Gépi tanulás közösségi hálózatok deanonimizálásához

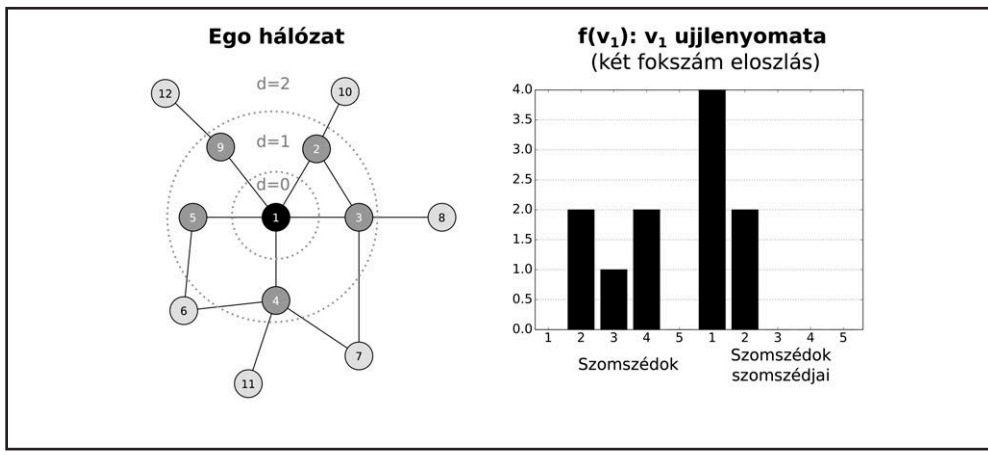
Az első közösségi hálózatokat gépi tanulás segítségével deanonimizáló algoritmust Pedarsani, Figueiredo és Grossglauser (2013) javasolta. Esetükben az algoritmus döntési mechanizmusa a Bayes-tételre épült, és jellemzőként az egyes felhasználók fokszámát és a gráf struktúrában már azonosított szereplőktől vett távolságait használta fel. Ahogy G' és G között nő a deanonimizált felhasználók száma, úgy tud egyre több információt az algoritmus a Bayes-döntés során figyelembe venni és egyre pontosabb döntéseket hozni. Az algoritmus további érdekessége, hogy ellentétben a Narayanan és Shmatikov által javasolt támadással, nem igényel inicializálást. Az algoritmus első lépésben a két közösségi hálózat legnagyobb fokszámú csomópontjai között keres párokat, majd a vizsgált csomópontok számát fokozatosan, iteratív módon kiterjeszti.

Az első teljesen átfogó összehasonlítást a közösségi hálózati deanonimizáló algoritmusokról Ji et al. (2015) készítette, amelyben a hét legkorszerűbb támadás közé beválasztották a Bayes-döntésen alapuló eljárást is. Az összehasonlításuk valós közösségi hálózatokon végzett mérésekre támaszkodott, amelyben az algoritmusokat többféle anonimizálási sémával szemben is alkalmazták. Noha az algoritmus jónak mondható eredményt ért el a helyesen deanonimizált felhasználókat tekintve, Ji et al. (2015) összehasonlító tanulmánya két jelentős részletet figyelmen kívül hagyott (amelyek a munka módszertana szempontjából is kritikusnak tekinthetők): a magas korrekt deanonimizálási arány mellé egészen magas hibaarány is társult (tehát alacsony volt a precizitás), illetve az algoritmus memóriaigénye, amely a meglévő deanonimizációs párosítások számától függően gyorsan növekszik (Gulyás, Simon és Imre 2016). Így az algoritmus eredményei kevésbé imponálóak; vélhetően az utóbbi probléma állhat annak a hátterében is, hogy az eredeti cikkben mindössze kétezer felhasználóból álló gráfon demonstrálták az algoritmus hatékonyságát (Pedarsani, Figueiredo és Grossglauser 2013), szemben a több tízezer csomópontból álló gráfokkal, amelyeket jellemzően alkalmazni szoktak az ilyen jellegű cikkekben.

A Sharad és Danezis (2014) által tervezett deanonimizáló algoritmus már általánosabb (de nem gépi tanulás által előállított) jellemzőket használt és véletlen erdőket (angolul *random forest*) a döntések meghozatalához. Az Orange 2012-es Data for Development (D4D) felhívása során ki akarta adni körülbelül 5 millió elefántcsontparti személy hívásinformációit (ki-kivel kommunikált), és előzetesen felkérte a kutatókat, hogy vizsgálják meg, hogy az adatok kellő mértékben anonimizáltak-e. Ez azt jelentette, hogy a kommunikációból létrejövő hálózatot ego hálózatokra (angolul *ego network*) szabdalták, amely egy felhasználóból és a körülötte lévő közvetlen kapcsolatokból állt (szomszédok és szomszédok szomszédai; lásd a 4. ábra, bal oldalt). Hamar kiderült, hogy ezek a darabok könnyen újra egyesíthetőek, ezért az Orange új, némileg módosított anonimizálási módszerrel állt elő. Hogy a macska-egér játéknak a kutatók elejét vegyék, megalkották a következőkben tárgyalt

algoritmust, amely az efféle apróbb módosítások esetén is működik és nem csupán ego hálózatok egyesítésére alkalmazható, hanem közösségi hálózat deanonimizáló algoritmusként is.

Sharad és Danezis a jellemző kiválasztásához az adott csomópont körüli szomszédok és azok szomszédjainak a fokszámának eloszlását javasolta (lásd a 4. ábra, jobb oldalt). Döntésük mögött az a megfontolás áll, hogy a különféle anonimizálási eljárások ellenére a fokszám eloszlás a hálózatban viszonylag érintetlen kell, hogy maradjon – különben az adatok felhasználhatósága (és üzleti, kutatási értéke) is jelentősen csökkenne.



4. ábra: Az 1-es felhasználó ego hálózata (a körök az egotól vett távolságot jelölik), és az 1-es felhasználóról (csomópontról) készített strukturális ujjlenyomat, amely a szomszédok és szomszédok szomszédjainak fokszámeloszlásának egymás után fűzése

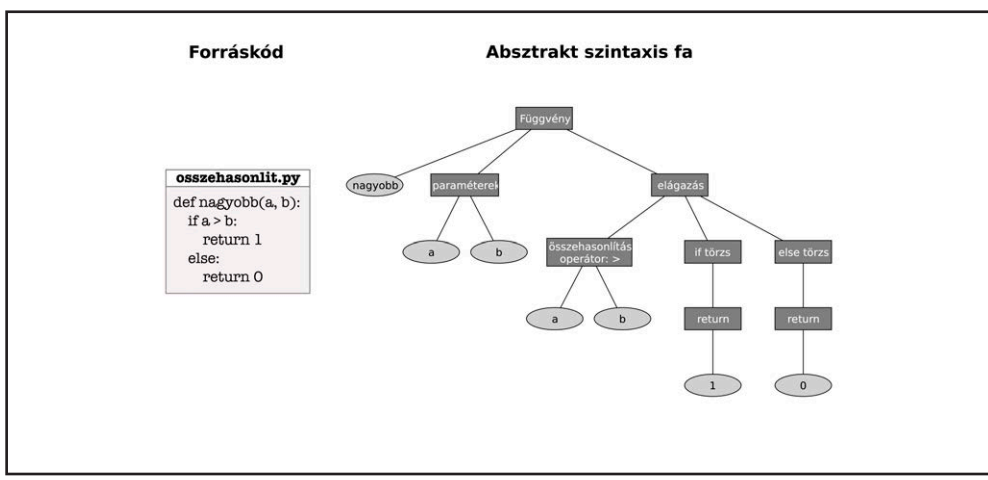
A deanonimizálási döntéshez pedig véletlen fákat használnak, a következő módon. Az algoritmus egy iterációjának bemenete $(v'v'')$ csomópont párok, amelyek egy, a háttérinformációból származó $v' \in G'$ csomópontból, és az ehhez tartozó összes potenciális $v'' \in G$-vel alkotott párosításokból állnak. Ezekhez az algoritmus kiszámolja a $(f(v'), f(v''))$ ujjlenyomatpárt, és ezeket a párokat adják bemenetként a véletlen erdőnek, amelynek kimenetként annyit kell kiadnia, hogy a két csomópont egyezik-e szerinte, vagy sem. Az algoritmus minden párosítást létrehoz, ahol a véletlen erdő egyező csomópontokat jelez. Ebből az is látszik, hogy az algoritmus egyedüli hátránya a Narayanan és Shmatikov (2009) támadásához képest, hogy nem veszi figyelembe a már deanonimizált felhasználó-párokat.

A támadást a Flickr szolgáltatásból származó adathalmazon tesztelték. A betanításhoz a véletlen erdőnek 5000 nem azonos (negatív minta), és változó számú, 10-1250 azonos felhasználó-párt mutattak (pozitív minta), a betanított modellt pedig 10 000 páron tesztelték le. Már 10 pozitív minta esetén is sikerült a felhasználók 16,74%-ának helyes deanonimizálása (felidézés), mindössze 1% hibaarány mellett. 50 pozitív mintával 22,01%-ra emelkedett a felidézés, és ezt érdemben a pozitív minták számával nem tudták növelni. Azonban nagyobb hibaarány tolerálása mellett ez is lehetséges, például 10% hibaarány mellett és 50 pozitív mintával 58,38% felidézést értek el. A szerzők által választott jellemző előállítási mód és a véletlen erdő alkalmazása kellően ellenállónak bizonyult.

Deanonimizálás programozási stílus alapján

Aylin et al. (2015) a programozók forráskód alapján történő deanonimizálását a fentiekhez hasonlóan gépi tanulási problémaként fogalmazták meg, amelyben a forráskód alapján létrehozott profilokról egy osztályozó gépi tanulási eljárás dönt, hogy ugyanattól a szerzőtől származnak-e, vagy sem. A korábbi munkákhoz képest fő újításként egy újszerű jellemzőkészletet dolgoztak ki programozók profilírozásához, illetve a módszerüket egy nagyobb, 250 programozót tartalmazó adathalmazon tesztelték, amely a Google Code Jam (GCJ) nemzetközi kódoló versenyről származik (a vizsgálatot a C++ forráskódú helyes megfejtésekre korlátozták). A támadás modellje a korábbival egyező sémára épül: rendelkezünk néhány forráskód-részlettel (anonim adat), amelynek szerzőjét keressük. Ehhez a rendelkezésünkre álló háttérinformáció egy forráskód-adatbázis, amely a keresett szerzőn kívül további szerzőktől is tartalmaz forráskódot.[2] A deanonimizálásnak ezen alkalmazási módja több pozitív felhasználási lehetőséggel is rendelkezik, például szerzői jog (bíróság előtti) bizonyításában, vagy plágium detektálásban, akkor is, ha konkrét kód-egyezés nincs a vizsgált művek között.

A jellemzők kinyeréséhez egy úgynevezett absztrakt szintaxis fát (angolul *abstract syntax tree,* AST) hoztak létre a forráskódból, erre látható egy egyszerű példa az 5. ábrán. Jól látszik, hogy például a függvények mélysége, az elágazások száma jól tükröződik ezen az ábrázolási módon, és ez összefüggésben áll a programozó absztrakciós képességével, ami máris egy jellemző.



5. ábra: Minta Python kód és a hozzá kapcsolódó absztrakt szintaxis fa

A kódolási profil háromféle jellemző csoportot tartalmazott mint szintaktikai, lexikai és kódelrendezés (vagy struktúra), és ezekből az első kettő tartalmaz több, az AST-ből származtatott jellemzőt is. Szintaktikai volt például az AST maximális mélysége, az egyes fa

[2] A cikk szerzői csak a zárt univerzum modell szerinti eseteket vizsgálták, amelyben a keresett programozó mindig szerepel a háttérinformációban. A nyílt világ modell vizsgálata még megoldandó kutatási feladat.

csomópontok relatív gyakorisága (de például a TF-IDF gyakoriság is (angolul *term frequency–inverse document frequency*), amely megmutatja, hogy egy adott szó a dokumentumon belül mennyire jelentős a korpusz egészéhez viszonyítva), átlagos mélysége, különböző csomópont bigramok relatív gyakoriságai (a bigram két csomópontegyüttes azonos sorrendű előfordulása, és általános formája az n-gram). Lexikai például az átlagos megjegyzéshossz logaritmusa, átlagos függvényszám logaritmusa fájlonként, míg az elrendezési jellemzők közé olyanok tartoznak, mint a szóközök, tabulátorok számának logaritmusa (fájlonként), illetve a tördelési szokások. A különféle n-gramok nélkül valamivel kevesebb, mint 300 jellemzőt határoztak meg, amelyek száma az n-gramokkal nagyságrendileg pár tízezerre nő.

Az így előállított profilokat véletlen erdőkkel hasonlították össze a Sharad és Danezis (2014) munkájában látott módhoz hasonlóan. Mivel a szerzők azonosítóit is közzétették a GCJ által publikált forráskódok mellett, így követni lehet éveken átívelően a munkásságukat. Ez megkönnyítette a modellek betanítását az egyes szerzők munkáinak felismerésére, hiszen nagyobb számú minta állt rendelkezésre.

Az így létrehozott programozó-azonosító rendszer magas találati arányt tudott elérni a különféle helyzetekben. Amikor 250 potenciális szerző közül kellett kiválasztania a megfelelőt, ez az esetek 95%-ában sikerült (plágiumkeresés jellegű alkalmazás), ha azonban csak két szerző között kellett választani, a rendszer az esetek 99%-ban sikeresen teljesítette a feladatot. További érdekesség, hogy megpróbálták egy kereskedelmi forgalomban elérhető kód-obfuszkáló programmal elrejteni a programozó identitását, de ez a találati arányon lényegében nem változtatott: kiderült, hogy a tesztelt alkalmazás a szintaktikai jellemzőket nem módosította, csupán néhány elrendezésbeli és lexikai jellemzőt[3], és ez nem volt elegendő.

Egy későbbi munkában Aylin et al. (2016) hasonló deanonimizálási támadást vizsgált bináris programok esetén, ami például fontos lehet rosszindulatú kódok szerzőjének a felderítésében. Azt találták, hogy bár a forráskódok bináris programmá alakítása (fordítás) során számos dolog visszaállíthatatlanul elvész, mint például a változónevek, illetve a fordító a program struktúráját is módosítja, azonban a szintaktikai jellemzők ez esetben is kevésbé sérülnek. Munkájukban megmutatták, hogy ezek visszanyerhetőek a bináris alkalmazások automatizált visszafordítása (angolul *decompile*) után, ugyanis a visszafordítás által létrehozott forráskódból kinyert absztrakt szintaxis fa továbbra is magán hordozza a programozó kézjegyét. Az AST-ből és további kiegészítésekkel olyan profilt tudtak létrehozni, amely koszinusz hasonlósága az eredeti program profiljához képest 80%-os volt, ami már elegendőnek bizonyult a véletlen erdő számára: 100 programozó esetén a deanonimizálás 78,1%-ban volt sikeres, de még 600 programozó esetén is 51,6%-ban.

Jövőkép: merre fejlődhetnek a támadások?

Az eddigi munkák eredményeit nem vonhatjuk kétségbe, azonban az is biztos, hogy nem értünk el a lehetőségek végére, több ponton lehet ezeket az eredményeket felülmúlni, vagy legalábbis jó eséllyel kísérletet tenni erre. A két legkézenfekvőbb trükk, amelyet

[3] Ez nem jelenti azt, hogy a programozó identitását ne lehetne anonimizálni ilyen szoftverrel, de azt igen, hogy ez az alkalmazás erre nem volt felkészítve. Lehet, hogy a teljes anonimizálás lehetséges, de ez további vizsgálatokat igényel.

gépi tanulási módszerek teljesítményének növelésénél alkalmazni szoktak, az a rendelkezésre álló adatok mennyiségének növelése, illetve *a döntési mechanizmusban alkalmazott gépi tanulási eljárások cseréje fejlettebbre*. A véletlen erdőket 1995-ben javasolták először (Ho 1995), és bár ma már „kulcskész" termékként elérhetőek különböző gépi tanulási könyvtárakban, nem minden esetben bizonyulnak a leghatékonyabb eszköznek.

Aylin et al. (2015) munkájának módszertanát követve Wisse és Veenman (2015) JavaScript programozókat azonosítottak AST-vel. Azonban munkájukban a véletlen erdőkkel (és néhány további eljárással) szemben lineáris kernelű *support vector machine* (SVM) eljárást alkalmaztak, mivel az jobb eredményt adott. Nem lenne meglepő, ha hamarosan kiderülne, hogy a mesterséges neurális hálózatok (angolul *artificial neural networks*, ANN) ennél is jobb eredményt képesek elérni: az elmúlt években az ANN-ek alkalmazása olyan eredményeket ért el különféle alkalmazásokban (mint például az arcfelismerés, vagy a beszédfelismerés), amely túlmutatott az addigi legjobb eredményen, bizonyos esetekben már az emberi pontosságot is elérve a problémák megoldásában (LeCun, Bengio és Hinton 2015). Az ANN-eket az emberi agy működése inspirálta, és az abban található neurális hálózatokat utánozzák. Bár az ANN-ek már évtizedek óta kutatott terület, az elmúlt évtizedben olyan új tanítási eljárások felfedezése hozott áttörést, amelyek a komplex és mélyebb struktúrájú hálózatok tanítását is hatékonyan el tudják végezni (illetve az ezzel párhuzamosan növekvő számítási kapacitás, amihez a videokártyák elterjedése is hozzájárult).

Szintén továbbfejlesztési lehetőség, ha a kvázi azonosítók helyett *automatizált jellemzőkinyerést* alkalmazunk az azonosítók precíz kézi megtervezése helyett. A komplex mesterséges neurális hálózatok ebben kiemelkedő eredményeket tudnak elérni: a hálózatok csak nyers adatot kapnak feldolgozásra, és maguk végzik el az adatok alapján a jellemzőkinyerést. Például az objektumfelismerésre szakosodott hálózatokban (ez az úgynevezett gépi látás szakterülete) a különböző neurális hálózati rétegek közvetlenül egymásra épülnek, és ahogy az információ feldolgozása történik, egyre komplexebb részletekkel dolgoznak. Arcfelismerés esetén míg az első neuron réteg csak a vonásokat fedezi fel (éldetektálás), a következő neuron réteg az ezekből összeálló részleteket, majd a harmadik pedig már az arc egyes részeit, mint szem, orr, stb. Ez a fajta jellemző-kiemelés hasznos lehet olyan esetekben, amikor a kézileg készített jellemzőkinyerés nem ismert, vagy kétséges, hogy a hatékonysága a legjobb.

Erre kiváló példa McPherson, Shokri és Shmatikov (2016) munkája, amelyben arcfelismerésre használt komplex neurális hálókat használtak elhomályosítással vagy kikockázással védett arcok felismerésére, az esetek 40-97%-ban sikeresen – ebben az esetben a jellemzőkinyerés nyilvánvalóan nem volt egyértelműen meghatározott.

A deanonimizálási probléma továbbá nagyon hasonlít a hitelesítés problémájához: van egy felhasználó (háttérinformáció) és az a kérdés, hogy egy adott felhasználói csoporton belül van-e vele egyező felhasználó (ami az anonimizált adathalmaznak felel meg). A legnagyobb különbség a két probléma között talán a felhasználók számában van: a hitelesítés(i eredményeket bemutató cikkek) jellemzően kisebb számú felhasználóval számol(nak) (lásd az alábbi példát), szemben a deanonimizálásnál látott ezrekkel, vagy többel. Ezért e korlát kiküszöbölése után várható csak, hogy a hitelesítésben alkalmazott áttörések megjelennek majd a deanonimizálási támadásokban is.

Gadelata és Rossi (2016) okostelefonok giroszkóp (sebességmérő) és gyorsulásmérő szenzorjaiból származó adatokat használtak hitelesítésre az okostelefonok felhasználói já-

rási stílusának elemzése alapján. A szenzorokból származó adatokat először is felosztották a járás ciklikussága szerint ablakokra, és ezekből az ablakokból származó mintákból nyertek ki egyéni jellemzőket komplex neurális hálózatokkal (úgynevezett konvolúciós neurális hálózatokkal). Munkájuk jól illusztrálja, hogy a hitelesítési problémában tipikusan mennyivel kisebb adathalmazokkal dolgoznak a kutatók: összesen 50 felhasználótól gyűjtöttek adatokat egy féléves periódus során, majd 35 alany adatait használták fel a jellemzőkinyerés betanítására, 15 felhasználó adatait pedig a hitelesítési mechanizmus tesztelésére.

Összegzés

Tanulmányunkban ismertettük a deanonimizáló támadásokat, amelyek az egyik legfőbb akadályát jelentik a különféle szolgáltatásokban gyűlő adatok publikálásának, hiszen sok esetben hatékonyan lehetővé teszik adathalmazok összekapcsolását, vagy anonimizálás esetén az eredeti identitás visszaállítását. Áttekintettük a korszerű deanonimizáló algoritmusokat, illetve a gépi tanulást alkalmazó támadásokat is. A jelen cikkben megjelenő trendből kitűnik, hogy – a privátszféra helyzetét tekintve – már most is jelentős fölénnyel bíró deanonimizáló támadások további, talán a felhasználói oldalon már egyáltalán nem ellensúlyozható előnyre tehetnek szert a gépi tanulási technikák alkalmazásával.

Ehhez vegyük hozzá, hogy nem egy olyan adattípus létezik (mint például a közösségi hálózatok struktúrája, vagy a telefonos kommunikációból származó helyzetinformáció), amelyre ugyan léteznek privátszféra-védő megoldások, de nincs általánosan elfogadott anonimizálási technika, amely a deanonimizáló támadásokkal szemben is megállja a helyét, és az alkalmazása az adat hasznosságát sem degradálja jelentősen. Emiatt egyelőre rövid és hosszú távon is csupán a jogi szabályozás tűnik megfelelő védelemnek, ami jelentheti azt, hogy az adat kurátora szerződésbe foglalva tiltja a deanonimizálást és az adatok további megosztását, illetve azt is, hogy törvényileg szabályozzák a deanonimizálás lehetőségét, például szigorú kutatási keretek közé szorítva a deanonimizálás lehetőségét. Erre láthatunk példákat, javaslat formájában már fel is merült az ausztrál törvényhozásban a deanonimizálás bűncselekménnyé nyilvánítása (Chirgwin 2016). Valamint az új európai adatvédelmi szabályozás (GDPR) is korlátozza az efféle visszaéléseket, ugyanis személyes adatnak tekinti az adatalanyhoz nem direkt módon köthető információkat is, amely a deanonimizálás célpontja lehet.

Irodalom

Aggarwal, Charu C., „On k-anonymity and the curse of dimensionality", in *Proceedings of the 31st international conference on Very large data bases (VLDB '05, Trondheim, Norway, August 30 - September 02, 2005)*, VLDB Endowment, 2005. pp. 901-909.

Bangeman, Eric, „AOL subscribers sue over data leak", *Ars Technica*, 26 September 2006. https://arstechnica.com/business/2006/09/7835/

Barbaro, Michael, „A Face Is Exposed for AOL Searcher No. 4417749", *The New York Times*, 9 August 2006. http://query.nytimes.com/gst/abstract.html?res=9E0CE3DD1F3FF93AA3575BC0A9609C8B63&legacy=true

Bennett, James and Stan Lanning, „The netflix prize", in *Proceedings of KDD Cup and Workshop 2007, San Jose, California, Aug 12, 2007*, ACM, 2007. https://www.cs.uic.edu/~liub/KDD-cup-2007/NetflixPrize-description.pdf

Caliskan-Islam, Aylin, Richard Harang, Andrew Liu, Arvind Narayanan, Clare Voss, Fabian Yamaguchi and Rachel Greenstadt, „De-anonymizing programmers via code stylometry", in Jaeyeon Jung (Ed.), *Proceedings of the 24th USENIX Conference on Security Symposium (SEC'15)*, USENIX Association, Berkeley, CA, USA, 2015, pp. 255-270.

Caliskan-Islam, Aylin, Fabian Yamaguchi, Edwin Dauber, Richard Harang, Konrad Rieck, Rachel Greenstadt and Arvind Narayanan, „When Coding Style Survives Compilation: De-anonymizing Programmers from Executable Binaries", *ArXiv preprint*, 1 March 2016. https://arxiv.org/abs/1512.08546

Chirgwin, Richard, „Australia wants law to ban de-anonymisation of anonymous data", *The Register*, 28 September 2016. http://www.theregister.co.uk/2016/09/28/oz_wants_to_ban_deanonymisation_ag_brandis/

Gadaleta, Matteo and Michele Rossi „IDNet: Smartphone-based Gait Recognition with Convolutional Neural Networks" *ArXiv preprint*, 19 October 2016. https://arxiv.org/abs/1606.03238

Gulyás Gábor György, Benedek Simon, Sándor Imre, „An Efficient and Robust Social Network De-anonymization Attack" in *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society (WPES '16, Vienna, Austria, 24 October 2016.)*, ACM, New York, NY, USA, 2016. pp. 1-11. https://doi.org/10.1145/2994620.2994632

Ho, Tin Kam, „Random decision forests" in *Proceedings of the Third International Conference on Document Analysis and Recognition Volume 1 (ICDAR '95, 14-15 August 1995.)*, IEEE Computer Society, Washington, DC, USA, 1995. pp. 278-283.

Ji, Shouling, Weiqing Li, Prateek Mittal, Xin Hu and Raheem Beyah, „SecGraph: a uniform and open-source evaluation system for graph data anonymization and de-anonymization", in Jaeyeon Jung (Ed.), *Proceedings of the 24th USENIX Conference on Security Symposium (SEC'15, Washington, D.C., 12-14 August 2015.)*, USENIX Association, Berkeley, CA, USA, pp. 303-318.

LeCun, Yann, Yoshua Bengio and Geoffrey Hinton, „Deep learning", *Nature*, Issue 521. (2015), pp. 436–444. http://dx.doi.org/10.1038/nature14539

McPherson, Richard, Reza Shokri and Vitaly Shmatikov, „Defeating Image Obfuscation with Deep Learning", *ArXiv preprint*, 6 September 2016. https://arxiv.org/abs/1609.00408

Narayanan, Arvind and Vitaly Shmatikov, „Robust De-anonymization of Large Sparse Datasets", in *Proceedings of the 2008 IEEE Symposium on Security and Privacy (SP '08, Oakland, 18-21 May 2008.)*, IEEE Computer Society, Washington, DC, USA, 2008, pp. 111-125. https://doi.org/10.1109/SP.2008.33

Narayanan, Arvind and Vitaly Shmatikov, „De-anonymizing Social Networks", in *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy (SP '09, Oakland, 17-20 May 2009.)*, IEEE Computer Society, Washington, DC, USA, 2009, pp. 173-187. https://doi.org/10.1109/SP.2009.22

Parra-Arnau, Javier and Claude Castelluccia, „Dataveillance and the false-positive paradox", in *Proceedings of the 1st International Workshop on Privacy and Inference (PrInf 2015)*, Dresden, Germany, 2015.

Pedarsani, Pedram, Daniel R. Figueiredo and Matthias Grossglauser, „A Bayesian method for matching two similar graphs without seeds", in *51st Annual Allerton Conference on Communication, Control, and Computing (Monticello, 02-04 October 2013.)*, 2013, pp. 1598-1607. http://dx.doi.org/10.1109/Allerton.2013.6736720

Sharad, Kumar and George Danezis, „An Automated Social Graph De-anonymization Technique", in *Proceedings of the 13th Workshop on Privacy in the Electronic Society (WPES '14, Scottsdale, Arizona, USA — 3 November 2014.)*, ACM, New York, NY, USA, 2014, pp. 47-58. https://doi.org/10.1145/2665943.2665960

Sweeney, Latanya, „k-anonymity: a model for protecting privacy”, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10. (2002) Issue 5., pp. 557-570. http://dx.doi.org/10.1142/S0218488502001648

Székely Iván: „Kukkoló társadalom - avagy van-e még függöny a virtuális ablakunkon?”, in Talyigás Judit (szerk.), *Az Internet a kockázatok és mellékhatások tekintetében*, SCOLAR, Budapest, 2010. 93-120. old.

Wisse, Wilco and Cor Veenman, „Scripting DNA: Identifying the JavaScript programmer”, *Digital Investigation*, Vol. 15. (2015. dec.), pp. 61-71. https://doi.org/10.1016/j.diin.2015.09.001

Gulyás Gábor György, PhD. Budapesten született 1984-ben. 2007-ben a BME Villamosmérnöki és Informatikai Karán szerzett diplomát az Infokommunikációs rendszerek biztonsága szakirányon, majd 2015-ben PhD fokozatot szerzett a BME-n a CrySyS Laboratórium tagjaként. Alapító tagja és rendszeres szerzője a Nemzetközi PET Portál és Blognak (2007-2015). Szervezője és előadója számos privátszféra-védelmet népszerűsítő előadásnak, eseménynek. Jelenleg az INRIA (Institut National de Recherche en Informatique et en Automatique, Franciaország) posztdoktori kutatójaként dolgozik a Privatics csoportban. Főbb kutatási területei a webes privátszféra-védelem (webes látogatók nyomkövetése és megfigyelése) és a (de-)anonimizálás témakörökre esnek.