REPÜLÉSTUDOMÁNYI KÖZLEMÉNYEK
33. évfolyam (2021) 3. szám 5–24. • DOI: 10.32560/rk.2021.3.1

Madácsi Richárd

Data science workflow a radaradat-elemzésben

Az adatelemzés a 21. század egyik leginkább felkapott szakterülete. Ennek egyik oka vélhetően a fejlett, neurális hálózatokon alapuló gépi tanulási algoritmusok megjelenése, amivel a lehetőségek látszólag korlátlanok. Ebből következően azon cégeknek, amelyek nem szeretnének a versenyben lemaradni, komoly energiákat kell ebbe a területbe fektetni. Sokszor azonban elsikkadnak a gépi tanuló algoritmus futtatása előtti teendők, az adatok tisztítása, transzformálása, a felhasználni kívánt modellnek megfelelő magyarázó változók előállítása, amelyek sok esetben nagyobb hatást gyakorolnak az eredményre, mint a választott gépi tanuló algoritmus vagy annak paraméterei [1]. *Számos ingyenes eszköz elérhető több különböző programozási nyelvben, amivel ezek a munkafolyamatok hatékonyan végezhetők, de az ATM*[1]*-fókuszú radaradat-elemzésben egyedi elemző eszközök fejlesztése is szükséges lehet. A cikk célja, hogy egy ilyen eszközzel szemben támasztott követelmények egy részhalmazát konkrét példákon keresztül bemutassa.*

Kulcsszavak: *adatelemzés, mesterséges intelligencia, gépi tanulás, ATM, adatvizualizáció, adattisztítás*

1. Bevezetés

Az adatelemzés még viszonylag fiatal terület, így nem alakulhatott ki olyan egyértelmű és hatékony munkafolyamat, mint a szoftverfejlesztésben [2]. Az ATM-területen történő adatbányászat, modellezés és gépi tanuláson alapuló termékfejlesztés pedig még ennél is kevesebb eddig felgyülemlett tapasztalatra építhet, így nem csoda, ha vannak félreértések egy cél elérése érdekében teendő lépéseket illetően. Sokan ugyanis úgy tekintenek a *data science* munkafolyamatra, hogy az pusztán a megfelelő adatok adatbázisból való lekérése és az azokon való gépi tanulási algoritmus futtatása. Ezt a tévhitet tovább erősíti a mesterséges intelligencia tudományának rohamos fejlődése, és az azzal szemben támasztott túlzott elvárások [3], például az, hogy bármilyen nem strukturált, csak alapadatokat tartalmazó információhalmazban lehetséges a rejtett összefüggések felfedése.

A *data science* projektekben érdekes módon gyakran a legtöbb energiát az adatlekérés és gépi tanulási algoritmusok futtatása közé beékelődő egyik fontos teendő, a megfelelő magyarázó változók előállítása igényli. A gépi tanulás könnyű, amennyiben rendelkezésre áll számos egymástól független változó, amelyek korrelálnak a célváltozóval. Ha nem ez a helyzet, akkor meg kell vizsgálni, hogy a rendelkezésre álló alapadatokból származtathatók-e jobb

[1] *Air Traffic Management.*

magyarázóváltozók [4]. A kérdés már csak az, hogy a radaradatok tekintetében ez milyen módon tehető meg.

A válaszhoz a HungaroControl Zrt. környezetirányítási rendszerében meghatározandó egyik környezetvédelmi mérőszám előállítását vesszük górcső alá. Az előállítandó adat – a repült útvonal hossza a T-bar eljárásokhoz képest – azt hivatott megvilágítani, hogy milyen gyakran van szükség a zajkoncentrációs céllal készült, elméleti legrövidebb útvonalként tekinthető eljáráshoz képest hosszabb útvonalra a hatékony forgalom áramoltatása érdekében. Ez az érték az extra repülések környezetvédelmi hatásainak vizsgálatán túl magyarázó változóként is szolgálhat, ha például a futópályaküszöbre érkezés várható idejét szeretnénk megbecsülni, ami megkönnyíthetné az egypályás futópályaüzem mellett a kapacitás maximalizálását. Ezen a területen már most is jelentős kutatások folynak [5].

2. Adatelemzői munkafolyamat

A *data science workflow* a [6] alapján a következőkő fő lépésekből áll.

2.1. Célkitűzés

Mivel semmilyen munka nem cél nélkül indul, az első lépés az elemzés konkrét feladatának meghatározása és annak pontos megértése. Utóbbi kiemelendő, hiszen bár fontos elvárás a szakterület megfelelő szintű ismerete *(domain knowledge),* az adatelemző ismeretei a szakterületi szakértőkével általában nem versenyezhet. A futópálya-elhagyási idő becslése jól definiált, aktívan kutatott terület [7], de a repülőtéri irányítótoronyban a forgalom megfigyelésével töltött idő és az irányítókkal történő beszélgetések olyan egyedi látásmódot adhatnak, amelyek akár a becslési pontosság növekedéséhez is vezethetnek.

2.2. Adatforrás

Felderítő rendszerből származó adatoknál (az egyszerűség kedvéért a továbbiakban radaradatok) a nulladik lépés az adatforrásnak megfelelő, hatékony valós idejű adattovábbítást támogató tömörített bináris állomány kikódolása. Mivel ez egy műszaki standard (például ASTERIX[2]), elvégzése tekinthető adottnak, adatelemző bevonása nem szükséges.

A következő lépés a további szükséges adatok összegyűjtése. Szerencsés esetben ez a már meglévő, mindenre kiterjedő adatbázisból való lekéréssel teljesíthető, de sokszor lehet szükség akár manuális adatelőállításra (például kérdőíves felmérés az adott időszak szektorterhelésének szubjektív megítéléséről), külső adatforrás bevonására (például Eurocontrol – Aircraft Performance Database – géptípus alapján szárnyfesztáv) vagy az adatok egy másik cégtől való megvásárlására (például FlightRadar24 Data Services).

[2] *All-purpose structured Eurocontrol surveillance information exchange.*

2.3. Csoportosítás

A radarjeleknek adatelemzés szempontjából megvan az a sajátosságuk, hogy bizonyos helyzetekben egyedi rekordként vizsgálandók, más esetekben pedig járatonként csoportosítva. Az előbbire példa a kisgépes forgalom ellenőrzése a tekintetben, hogy megsértették-e az ellenőrzött légtér határait. Ennél a feladatnál egy egyszerű, „*point-in-polygon*" algoritmussal meghatározható, hogy egy adott radarjel egy ellenőrzött légtér területén helyezkedik-e el, és a radarjel magasságát összevetve a légtér alsó határával meghatározható a légtérsértés ténye.

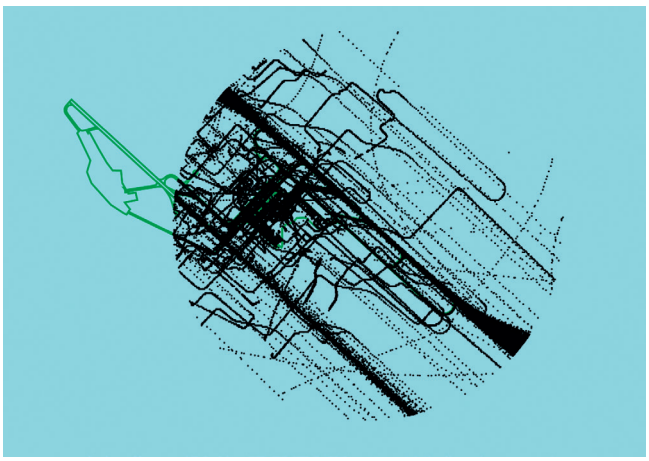
Az érkező légi járművek üzemanyag-hatékonyság miatt fontos folyamatos süllyedésének elemzése értelemszerűen nem végezhető el a radarjelek elkülönülten történő vizsgálatával. Az adatelemzési eszköztár egyik gyakran használt eleme az információk csoportosítása *(grouping),* amelyhez természetesen szükség van egy egyedi azonosítóra, amely alapján a csoportok képezhetők. Légi járművek esetén látszólag triviális ez a feladat, hiszen a légi jármű hívójele alapján könnyen egy járathoz társíthatók a radarjelek. A hívójel (például ABC123) viszont csak egy időpillanatban egyedi, heti, havi elemzések esetén több járat is szerepel az adatcsomagban azonos értékkel. Felmerülhet megoldásként a dátum hozzáadása a hívójelhez (pédául ABC123-2019.10.17), ez viszont sok problémát okozhat például az éjfél előtti tervezett érkezési idővel rendelkező, de ténylegesen éjfél után megérkező járatoknál. Az éjfél utáni pár perces repülést az aznap késő esti radarjelekkel egyként kezelni természetesen nem megfelelő. Következő ötlet lehet az SSR-kód[3] felhasználása a hívójelhez adva (például ABC123-2643), abban bízva, hogy az elemzés céljából származtatható elfogadhatósági kritérium alatt van annak az esélye, hogy egy adott időszakban több azonos hívójelű járat számára is azonos SSR-kód lett kiosztva. A Mode S radarok terjedésével, és az azon alapuló azonosítással az előbb vázolt lehetőség elvethető, hiszen ebben az esetben egységesen az 1000 állítandó be *(Mode S Conspicuity Code).* Gyakorlati tapasztalat alapján elfogadható megoldás lehet a hívójel mellé az SSR-kódot, a radar által kiadott track azonosítót és a Mode S transzponder kódját társítani, amivel az így képzett járatazonosító egyezésének esélye minimalizálható. Utóbbi állítás abban az esetben igaz, ha az egyszerre feldolgozandó radaradatok mennyisége relatíve alacsony, maximum havi bontású. Ez természetesen nem jelenti azt, hogy egy big data projektben nem használható a módszer, hiszen egy több évtizedet felölelő radaradatbázis használata esetén sem lehet az összes adatot egyszerre betölteni a számítógép memóriájába, hanem kisebb részekre vágva kell azt feldolgozni.

Egy szofisztikáltabb, de nagyobb erőforrás-igényű módszer az azonos hívójelű, nagyobb időszakot átölelő plotok csoportosítása, majd azokban az időbeli és térbeli „szakadások" azonosítása, és a szakadások által elválasztott tényleges járatok újracímkézése egy sorszámmal. A legegyszerűbb módszer viszont az adatforrás specifikációjának módosítása úgy, hogy az abból kiexportált járatazonosító tényleg egyedi legyen, például az adott időpillanatban egyedi hívójel kiegészítése egy ténylegesen egyedi UUID[4]-vel.

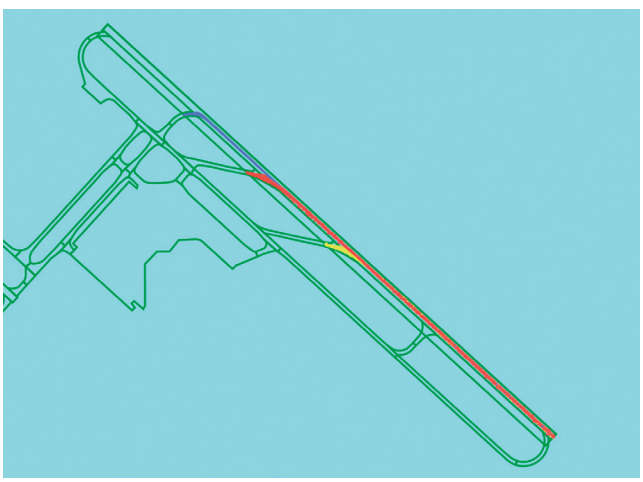
[3] *Secondary Surveillance Radar.*
[4] *Universally Unique Identifier.*

2.4. Adattisztítás

Egy adatbázisból betöltött adathalmaz ritkán lesz azonnal használható az elemzéshez. Az eltávolítandó rekordoknak két fő csoportja van. Az egyik a felderítő berendezésből származó fals jelek és más adattorzulások, valamint a hibát nem tartalmazó, csak egyszerűen az elemzéshez nem szükséges adatok, amelyek szintén eltávolítandók a munka gyors és hatékony végzéséhez. Az utóbbi a következő rész, a szűrés feladata (1. és 2. ábra).



1. ábra
Nyers ASMGCS[5]*-adatok* [a szerző]

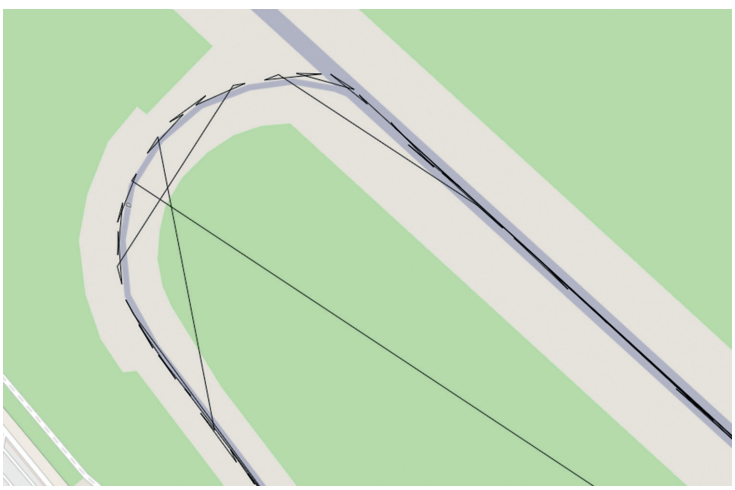


2. ábra
Megtisztított és szűrt ASMGCS-adatok [a szerző]

[5] *Advanced Surface Movement Guidance and Control System.*

Az adattisztításban kiemelendő a vizualizáció szerepe és annak pontos módja. A csoportosításnál említettekhez hasonlóan itt is különbség van abban, hogy radarjelként (pont) vagy járatként tekintünk (vonal) az adatokra. A különböző adattranszformációk végzése a pontok halmazán egyszerűbb, de bizonyos típusú anomáliák a járatok időben egymást követő radarjeleinek vonalakkal való összekötésével könnyebben azonosíthatók.

Az 3. ábrán egy a RWY31R futópályára érkező légi jármű helyzetei láthatók, ami éppen a D jelű gurulóúton hagyja el a pályát. Az összekötött radarjeleknek köszönhetően egyértelműen látszik, hogy az időalapú sorrendezésbe hiba csúszott, ami pontonkénti megjelenítéssel sokkal később, esetlegesen hibás vagy használhatatlan elemzések után derült volna csak ki. A MongoDB adatbázis használata JavaScript nyelven (+Node és Express) a Mongoose keretrendszerrel a legegyszerűbb. Itt a sémadefiníciós részben meg lehet adni, hogy az adatbázisból letöltött adatok változóinál milyen típuskonverzió *(casting)* szükséges. Esetünkben a „*timestamp*" értéke dátum, azaz String => Date átalakítás kell. A szerverről a kliensoldalra továbbítandó adatok nagy mennyisége miatt viszont érdemes a Mongoose által opcionálisként felajánlott „*lean*" funkcióval élni, amivel nem az alapértelmezett „*Document",* hanem sima JavaScript objektumok (POJO – *Plain Old JavaScript Object*) lesznek használva. Emiatt viszont elveszik többek között a sémavalidáció és a típuskonverzió, így a sorrendezés az idők különbsége[6] helyett szövegek különbsége alapján megy végbe, aminek természetesen nincs értelme. A megfelelő vizualizáció miatt a problémára hamar fény derült, és a feldolgozás javítása a fejlesztés korai fázisában megtörtént.

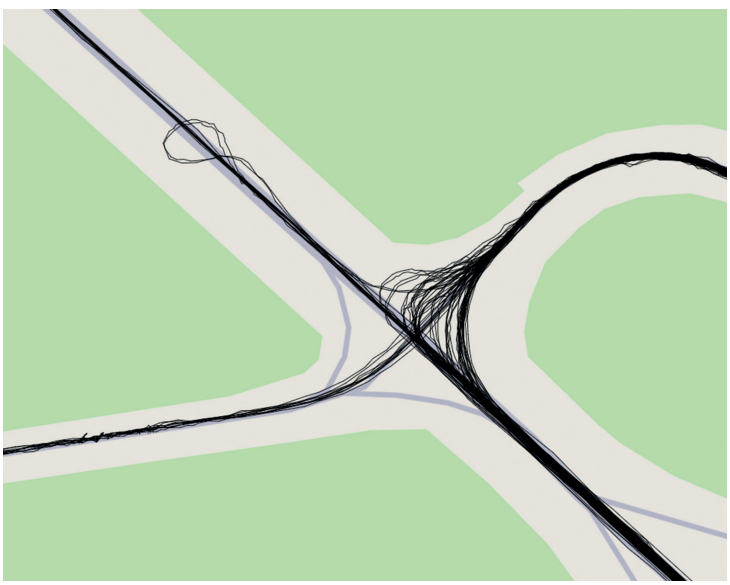


3. ábra
Hibás radarjel sorrendezés [a szerző]

A hamis vagy hibás radarjelek/járatok egyik eltávolítási módja lehet azon trajektóriák szűrése, ahol a radarjelek száma egy adott küszöbértéknél alacsonyabb [8]. A küszöbérték azonban adatforrástól, időszaktól, a radarrendszer beállításaitól is függhet, ezért nem javasolt azt kőbe vésni. Felmerül a kérdés, hogy hogyan határozható meg?

[6] data.sort((a, b) => a.timestamp – b.timestamp).

A *data science* folyamatban gyakori lépés a különböző grafikonok megalkotásával történő feltáró célú adatelemzés *(exploratory data analysis).* A Python és R programozási nyelvek kiváló eszközökkel rendelkeznek ehhez (Matplotlib, ggplot2). Az így készített ábrák statikusak, így ha például a járatokhoz tartozó radarjelek számának hisztogramját kívánjuk megjeleníteni, akkor a szűréshez használt küszöbérték leolvasható, de az azzal való művelet plusz egy lépés. Ráadásul ez a háttérben történik, azaz egy Python vagy R parancs/kódrészlet lefuttatásával. Erre eklatáns példa az *outlier*-ek szűrése, az átlagtól való eltérés nagysága alapján, például [7] 2 szórásnál nagyobb eltérést tekint abnormális pályaelhagyási időnek. Egy hosszabb ideig tartó pályaelhagyásnál viszont nem mindegy, hogy az alacsony sebesség volt az ok, vagy például *backtrack.* Utóbbi egyértelműen kihagyható abból a halmazból, amely alapján végső egyenesen tartandó térköz számítható, hiszen a repülőtéri irányító (TWR) egy nagyobb forgalmú időszakban, ahol kritikus a mielőbbi pályafelszabadítás, ilyen manővert nem engedélyez. Az előbbi azonban ilyen egyszerűen nem ignorálható, gondoljunk csak arra az esetre, ha a légi járművek döntő többsége RWY13R irányból a J4-et használja, de néhány a J4-en való elhagyáshoz való fékezés után mégsem tudja ott a pályát felszabadítani, ezért visszagyorsítás nélkül el kell hogy guruljon a pálya végéig (A1-A2).



4. ábra
Pályaelhagyás backtrackkel [a szerző]

Az adatok grafikonon való megjelenítését egy saját fejlesztésű programban interaktívvá lehet tenni, például úgy, hogy a hisztogram oszlopaira való kattintással kiemelhetők az abban a csoportban szereplő radarjelek (4. ábra). Ezzel a módszerrel a munkafolyamat ténylegesen a szemünk előtt végződik, így az eredmény tekintetében is nagyobb a bizalom, hiszen pontosan tudható, hogy mi és miért törlődött. Ez különösen akkor jelentős, amikor például az ASMGCS adataival végzett pályaelhagyási idők mérésén alapul a végső egyenesen, egymást követő érkező légi

járművek között tartandó elkülönítés számítása. Blackbox jellegű elemzési folyamat után a repülésbiztonsági érvelés megírása a változtatásról és annak lehetséges hatásairól nehézkes.

2.5. Szűrés

Ha az adatok már nem tartalmaznak anomáliákat, a következő lépés az elemzéshez nem szükséges részhalmaz eltávolítása. Pályaelhagyási idők mérésénél például értelemszerűen az induló légi járművekkel nem kell foglalkozni. Ezek szűréséhez a legegyszerűbb módszer a repülési terv (FPL[7]-) adatok felhasználása, ahol, ha az ADEP-mező megegyezik LHBP[8]-vel, akkor egyértelműen indulóról van szó. De a pályaelhagyási idő nem repülőtérre, hanem futópályára vonatkozik, azt viszont nem tartalmazza az FPL.

Az érkező forgalomnál a használatos küszöb meghatározása történhet az utolsó radarjel és a pályaküszöbök (THR[9]) közötti távolság mérésével [9], ahol a legkisebb távolsággal rendelkező küszöb a valós elméletileg. Utóbbi kitétel azért szükséges, mert a föld közelében a radarjelek eltűnhetnek, így előállhat az a helyzet, hogy egy RWY31L LHBP érkező utolsó radarjele a THR31R-hoz van közelebb. A másik problémaforrás, ha a földön lévő légi járműnél is tökéletesen lát a radar, és az utolsó radarjel a pálya felénél túl van, akkor az ellentétes küszöb lesz a járathoz rendelve (THR31R → THR13L).

Univerzális megoldásként a megfelelő szűréshez szintén a vizualizáció interaktív felhasználása a javasolt. A megjelenített radarjeleket manuálisan „körbe rajzolva" 2 dimenzióban egyértelműen meghatározható a valamilyen szempontból fontosnak ítélt radarjelek halmaza. Ezt kiegészítve a radarjelek attribútumaiban való szűrés lehetőségével, megkapjuk a leggyakrabban és legsokoldalúbban használható eszközt a radaradat-elemzésben.

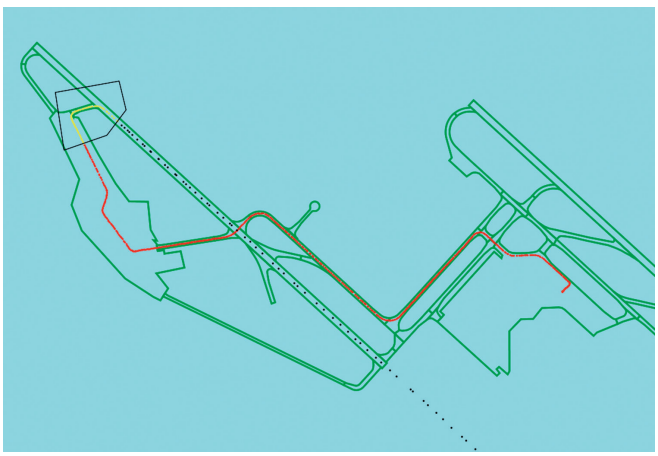
A RWY13R érkezők elválasztása a RWY31L indulóktól történhet ezzel a módszerrel úgy, hogy a végső egyenes területén létrehozott poligonon belül azokat a radarjeleket jelöljük ki, ahol a haladás iránya (track) 110 és 150 fok között van. Ezután szükséges kijelölni az összes radarjelet járatonként, ahol legalább egy már kijelölt van, hogy majd a kijelölést megfordítva, a kijelölt radarjelek törölhetőkké váljanak. Természetesen az utóbbi lépéseket egy saját fejlesztésű elemzőrendszerben össze is lehet vonni egy olyan, szintén gyakran használt parancsban, amely azoknak a járatoknak a radarjeleit tartja meg, ahol van legalább egy kijelölt.

Az adatbázisból letöltött és megjelenített adatok sok helyet foglalnak a memóriában, ezért egy lépéssel tovább is lehet menni, és javasolt azokat az adatokat betölteni már az elején, amelyek adott térbeli és attribútum jellegű feltételeknek megfelelnek.

A radarjelek szűrésénél még számos fontos és sok területen felhasználható módszer azonosítható. A pekingi TMA[10]-ban lévő légi járművek vizsgálatához [8] csak a 25 NM (46,3 km) sugarú körön belül lévő adatokat használták. Mivel a Budapest TMA (és a pekingi) nem koncentrikus kör alapú, ez csak közelítő megoldásként jó. A terület alapú szűrés nemcsak interaktív módon rajzolt poligonnal történhet, hanem már meglévő, adatbázisba betöltöttel is, például a TMA-szektorokkal, gurulóutak körvonalaival, vagy akár a zajvizsgálatoknál használt budapesti kerületek határaival is.

[7] *Flight plan.*
[8] Budapest Liszt Ferenc Nemzetközi Repülőtér.
[9] *Threshold.*
[10] *Terminal Manoeuvring Area.*

Ezenkívül sokszor van szükség egy adott poligonba való belépés utáni első vagy az abból való kilépés előtti utolsó radarjel meghatározására, például a futópálya-elhagyási idők mérésénél. Utóbbinál azonban problémát okozhat az, ha RWY31L érkező C vagy D gurulóúton elhagyja a pályát, de a kettes terminál előterére kell hogy beguruljon a pálya keresztezésével B1 B2 útvonalon. Ebben az esetben a futópálya téglalapját elhagyó járat utolsó radarjele nem az lesz, ami a pályaelhagyási idő méréséhez kell. Ilyenkor jól jön az a funkció, amellyel a kijelölést időben lehet módosítani. A tényleges pályaelhagyás radarjeleinek (sárga) meghatározása után azokat kell eltávolítani, amelyek a legutolsó kijelölt pontnál későbbi időbélyeggel rendelkeznek (piros). Így a pálya téglalapjába eső első és utolsó radarjel *timestamp*jeinek különbsége már a valós, szakmai szempontból releváns pályafoglaltsági időt adja meg (5. ábra).



5. ábra
Duplikált pályaelhagyás szűrése [a szerző]

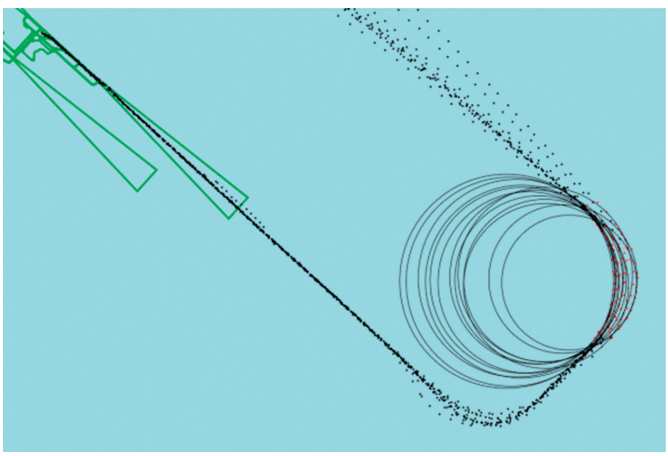
2.6. Feature engineering

A *feature engineering* [10] azt a folyamatot jelenti, amely során megfelelő szakmai háttérismerettel olyan származtatott változók állíthatók elő, amelyek az adott probléma kezelésére felállított prediktív modellt pontosabbá teszik.

Triviális *feature engineering*re példa, ha a járat hívójelének első három karakterét levágjuk, és azt használjuk fel például a küszöbre érkezés várható idejének becslésére használt modellben. A RYR123-ból RYR, amely a járatot jelképezi, nagyon fontos magyarázó változó lehet, hiszen ugyanazt a géptípust másképp repülik a különböző légitársaságok. Kevésbé triviális lehetséges magyarázó változó a fordulóban lévő légi járművek radarjeleire illesztett kör sugarából és a légi jármű sebességéből becsülhető légi jármű bedöntési szöge *(bank angle)* [11] (6. ábra).

A pekingi érkezési idők becslésénél [8] az alapadatokon (radarjel: szélesség, hosszúság, magasság, sebesség, idő) kívül a szerzők felhasználták a radarjelnek egy mesterségesen kijelölt referenciaponttól való távolságát és irányát, valamint egy mesterséges vonatkoztatási rendszer egyik sarokpontját. A küszöbre érkezés ideje pedig a vizsgált járatok utolsó érzékelt radarjeleiből képzett átlagos pozíció (centroid).

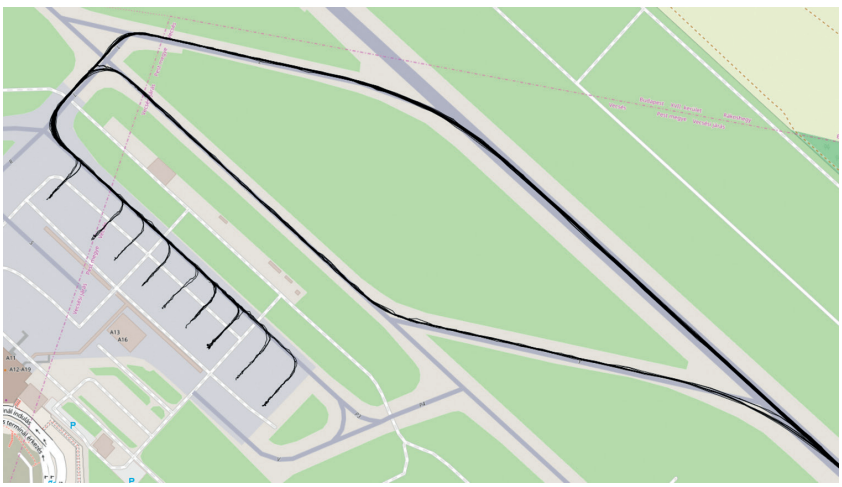
Az említett magyarázó változók előnye, hogy nagy mennyiségű adatokon „látatlanban" is kalkulálhatók. Hátránya viszont, hogy nem jellemzik megfelelően a forgalomkezelés mintázatát, így hiába a profi gépi tanuló algoritmus, ha nincs minta, amit felismerhetne. Miért kellene egy képzeletbeli pályaküszöbre (utolsó plotok átlaga) számolni a tényleges helyett? Miért egy képzeletbeli referenciaponthoz mérik a távolságot és irányt az eljárásnak a járat geometriájától függő *Initial Approach Fix*-e (IAF) helyett? A válasz: mert ez így egyszerűbb. A HungaroControl Zrt. Módszertani és Koordinációs Osztályánál nem engedhető meg az egyszerű megoldás, hiszen itt elsősorban nem egy cikk publikálása a cél, hanem adott esetben az ATM funkcionális rendszer működésének módosítása. Emiatt szükséges a saját elemző rendszer fejlesztése, ahol akár nagyobb munka árán is, de azok a magyarázó változók állíthatók elő, amelyek a legnagyobb előnnyel járnak.



6. ábra
Forduló sugár mérése [11]

Feature engineering esetén felmerülhet kérdésként, hogy a származtatandó változó az elemzési munka során álljon elő, vagy az adatok esetleges előfeldolgozásánál. Elsőre logikusnak tűnhet, hogy például a futópálya-elhagyási időt ne az alap, felderítési berendezésből származó pozícióadatokból kalkuláljuk, hanem maga az ASMGCS-berendezés tegye ezt meg, és egyből az elemzéshez szükséges értéket mentse el az adatbázisba. Alternatív, hasonló eredményre vezető megoldásként a felderítő berendezésből ASTERIX-szabvány formátumban exportálható bináris adatcsomag dekódolásakor, amit a pozícióadatok adatbázisba való feltöltése előtt meg kell tenni, is előállítható a kívánt érték automatizáltan. A *dashboard* jellegű, döntéshozóknak szánt havi statisztika készítésénél ez a módszer hatékony lehet, de a *data science* terület alapvető célja az adattermék [12] előállítása, nem pedig az időszakosan generált riport. Az adattermék lehet egy szoftver (például új légiforgalmi irányítói eszköz), vagy ahhoz egy algoritmus (például küszöbre érkezés várható idejének pontos becslése), egy adatokból kiolvasható szabály, amely végrehajtva biztosítja a rendszer megfelelő működését (például amennyiben képes a pilóta a RWY31R esetén az Y gurulóúton való pályaelhagyásra, akkor csökkentett elkülönítés is elegendő), de akár egy érdekes mintára való figyelemfelhívás (például egy géptípus pályaelhagyási paraméterei szignifikánsan eltérnek a kategóriájának átlagától).

Az előre feldolgozott adatok használata esetén nagymértékben beszűkül az adatelemző mozgástere. A RWY31R futópálya-elhagyási idő mérésénél tegyük fel, hogy a küszöbhöz közelebbi Y gurulóúton elhagyó légi járművek 95%-ának sebessége a gurulóút előtt adott távolságon nagyobb, mint 30 knot (15,43 km/h). Kérdés, hogy ha egy érkező légi jármű a távolabbi Z gurulóúton hagyja el a pályát, mert így könnyebb az állóhelyére beállni, de a sebessége az előbbiekben meghatározott helyen a példaként említett érték alatt van, akkor a relatíve magas pályaelhagyási időt figyelembe vegyük-e a végső egyenesen tartandó térköz meghatározásánál. Ez a járat egyértelműen rontja az AROT[11]-statisztikát, de nem azért mert nem lett volna képes gyorsabb pályafelszabadításra, hanem mert nem volt rá szükség. Előfeldolgozott pályaelhagyási adatok esetén ilyen típusú mélyebb vizsgálatokra nincs mód, ezért egy adatelemző rendszer kialakításakor az adatbázisba való feltöltésnél csak a legszükségesebb adattranszformáció elvégzése javasolt (7. ábra).



7. ábra
31R pályaelhagyás Y és Z gurulóúton [a szerző]

2.7. Exportálás

Természetesen egy saját fejlesztésű elemzőrendszernek nem kell mindent tudnia. A gépi tanuló algoritmusok futtatásához továbbra is az R vagy Python programozási nyelvek megfelelő eszköze javasolt. Ehhez azonban az előállított, megfelelően tisztított, addicionális magyarázó változókkal ellátott adathalmazt a megfelelő formátumba ki kell tudni exportálni. Szoftverfejlesztési szempontból a legegyszerűbb, ajánlott formátum a CSV,[12] amelyet bármilyen keretrendszerbe be lehet olvasni, de akár még szövegként is könnyen értelmezhető, ha gyors ellenőrzés szükséges.

[11] *Arrival Runway Occupancy Time.*
[12] *Comma Separated Values.*

Gyakran van szükség az adatok megjelenítésére az elemzőrendszeren kívül, így az ingyenes Google Earth program KML-formátumában való kiexportálási lehetőség kifejlesztése is javasolt.

2.8. Automatizálás

A háttérben történő, vizualizáció nélküli adattranszformációs műveletekkel kapcsolatban több kritika is megfogalmazódott eddig. A módszer vitathatatlan előnye azonban az automatizálás. Ha egy elemzés egy R vagy Python script futtatásával történik, akkor a vizsgált időszak kibővítése vagy módosítása a dátumparaméterek átírásával könnyen megtehető. Ha az adatokat vizualizálva, poligonok megrajzolásával alakítgatjuk, felmerül a kérdés, hogy egy másik időszakban is újra meg kell-e csinálni minden műveletet. Mivel a számítógép memóriája korlátos, így tetszőleges mennyiségű adatot beolvasni és kezelni nem lehet, tehát muszáj napi, heti, esetleg havi bontásban végezni a munkát. Emiatt javasolt olyan rendszer kidolgozása, ahol a manuális és egyéb műveletek makroprogramként is elmenthetők, így a vizsgált időszak módosítása és az elemzés megismétlése az alternatívához hasonlóan egyszerű.

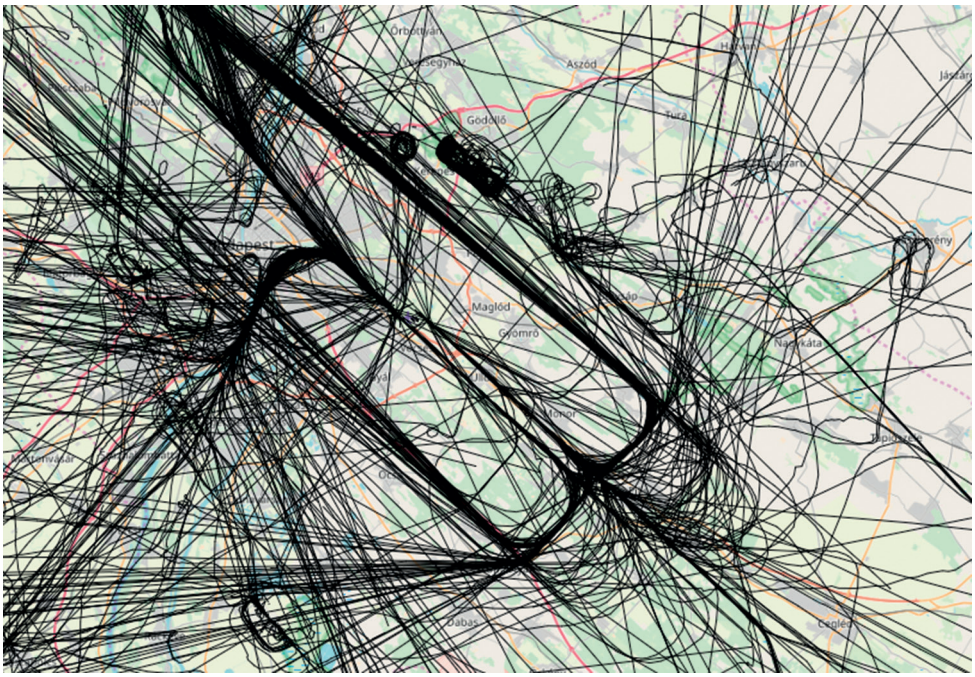
3. Esettanulmány

3.1. Célkitűzés

A bevezetőben említett meghatározandó érték járatonként a Budapest TMA-ban az érkezők által ténylegesen lerepült útvonal és a T-bar kezdőpontjára való közvetlen repüléssel, majd az eljárás követésével előálló elméleti legrövidebb útvonal hosszának különbsége. A futópályaküszöbre érkezés várható idejének becslésénél ez különösen fontos magyarázó változó lehet, hiszen bármilyen pontosan is becsülhető egy légi jármű sebességprofilja, ha közben a lerepülendő távolság nagysága bizonytalan. Szükséges tehát azokat a faktorokat azonosítani, amelyek előre jelzik az elméleti minimum útvonalnál hosszabb repülést és annak nagyságát. Ha ez már megfelelő pontossággal működik, már a lassulással is van értelme foglalkozni.

3.2. Adatforrás

A felhasznált adatok a *back-up* radarirányítási rendszer archív állományából származnak, amelyeket a könnyebb felhasználás érdekében a HungaroControl Zrt. Módszertani Csoportján fejlesztett adatelemző rendszer MongoDB adatbázisába töltik át. Ebből egy JavaScript nyelven, Express keretrendszerben készített szerverből egy OpenStreetMap térképet használó webes applikáción keresztül jut el az információ a felhasználóhoz. Az adott időszak kiválasztása után, a példánkban az áttekinthetőség miatt ez egy nap (2019. 08. 01), a 8. ábrát kapjuk.

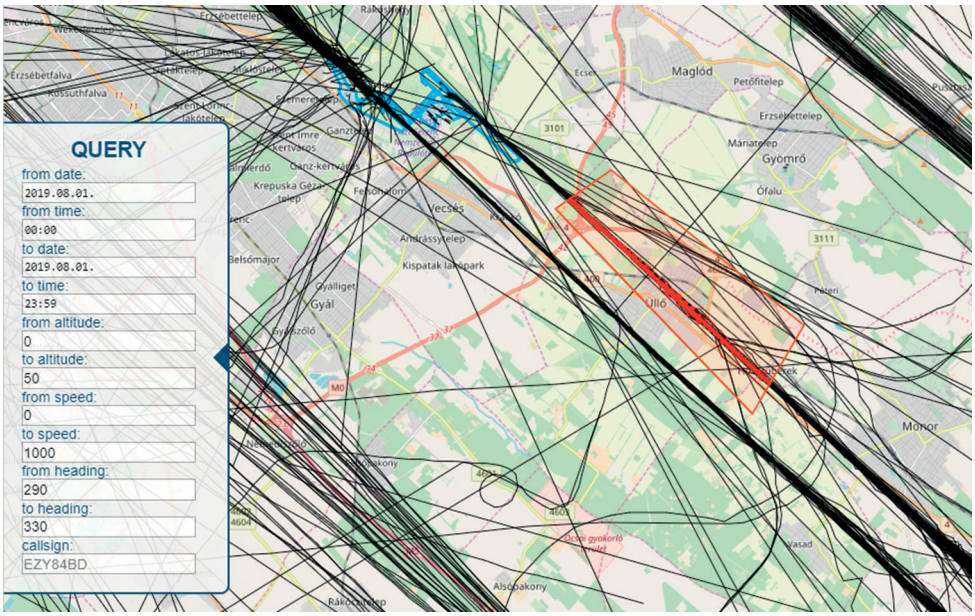


8. ábra
Nyers radaradatok [a szerző]

Látható, hogy a letöltött adatokban induló, kisgépes forgalom és anomáliák is találhatók, így adattisztítás és szűrés szükséges.

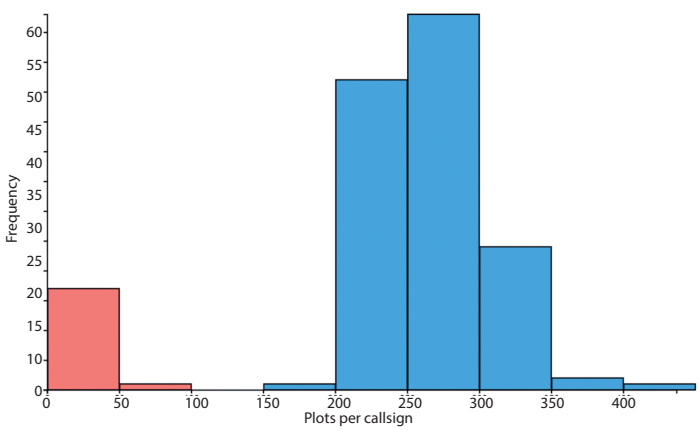
3.3. Adattisztítás és szűrés

A probléma jellege miatt érdemes előre venni a szűrésfázist. Az RWY31R végső egyenes környékén rajzolt poligonon 5000 láb (1524 m) alatt 290° és 330° közötti irányon áthaladó járművek az RWY31R érkezők, amelyeket meg kell tartani. A többi a program memóriájából és a megjelenítésből törölhető. A 9. ábrán a kijelölésnél látszik, hogy nem váltak pirossá a RWY13L indulók, amelyek a felszállás után 13°-kal balra fordulnak. Ezek a radarjelek szintén áthaladnak a kijelölt területen, de a beállított paramétereknek *(track)* nem felelnek meg. Az azonos területen a TMA átrepülő forgalma pedig a magassági korlát miatt nem kerül a kiválasztottak közé.



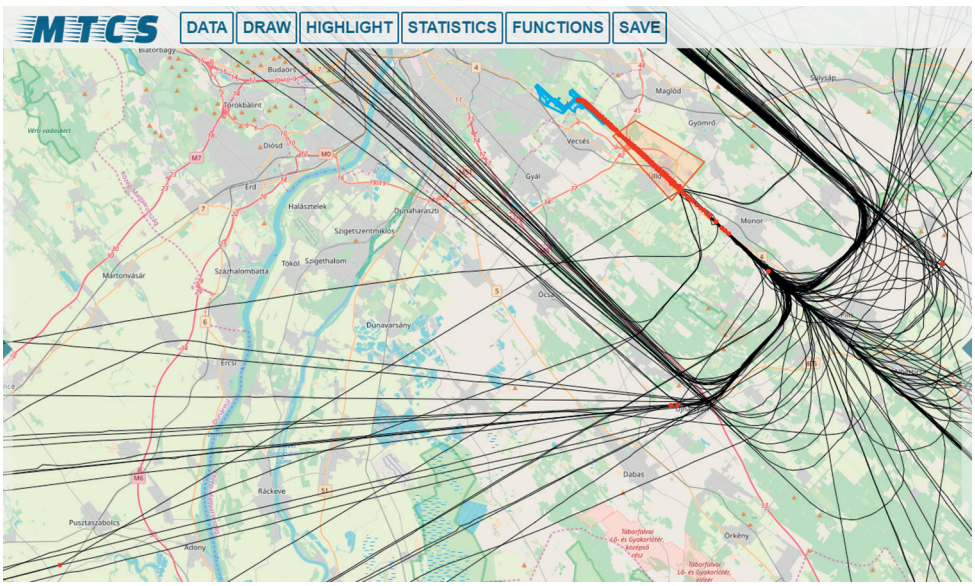
9. ábra
Érkezők és indulók leválasztása [a szerző]

Az első szűrés után a vizualizációnak hála rögtön nyilvánvalóvá válik, hogy egy pár radaranomália is került a rendszerbe. A kiegészített hívójel alapján azonosított járatokhoz tartozó radarjelek számából készített gyakoriságeloszlásból látszik, hogy vannak olyanok, amelyek kilógnak a többségből (10. ábra).



10. ábra
Radarjelek járatonkénti számának gyakoriságeloszlása [a szerző]

Ezeket a hisztogramon kijelölve ellenőrizhetővé válnak a térképes nézeten, hogy valóban törölhető fals adatok, vagy valamilyen előre nem várt, de szakmai szempontból igazolható furcsaság eredménye. Jelen esetben az alacsony radarjelszámmal rendelkező járatok fő csoportja az éjfél után érkezők végső egyenesre eső része, a másik pedig a valódi hívójel, de azonos radarazonosítóval rendelkező fals jelek. Utóbbira lehet példa a 11. ábrán a bal alsó sarokban lévő piros kör, amely egy végső egyenesen lévővel van direktben összekötve, ami nem lehet valós járat, vagy ha igen, hiányos a végső egyenesig való eljutás útvonala, így törölhető.



11. ábra
Azonosított anomáliák [a szerző]

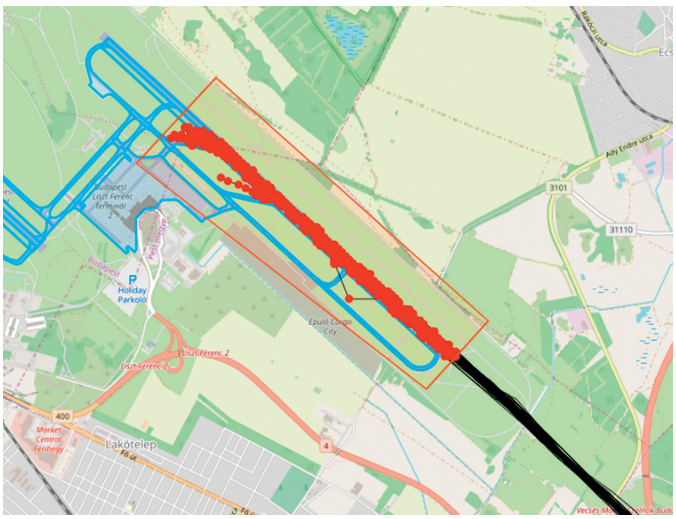
Az érkezők szűréséhez hasonlóan törölhetők a RNAV[13] Y eljárást követő légi járművek is, mivel ezek nem az alapértelmezett legrövidebb útvonalat repülik.

3.4. Feature engineering

A kitűzött feladat megoldásához a legfontosabb teendő a ténylegesen lerepült útvonal meghatározása járatonként. Ez úgy történik, hogy a kibővített hívójel alapján csoportosított és időrendbe tett radarjelek közti távolságot kumuláljuk. Itt két kérdés merülhet fel. Mi legyen a nulla pont, és időben előre vagy hátrafelé történjen a távolságok összeadása. Mivel érkező forgalom a vizsgálat tárgya, ezért az a megoldás a jó, ha a végső egyenesen, azonos pontban azonos a küszöbig lerepült távolság értéke. Ennek két folyománya van. Az első, hogy időben hátrafelé kell az összesítést megtenni, hiszen ellenkező esetben attól függően, hogy hol lépett be a járat a TMA-ba, különböző eredményt kapnánk. A másik, hogy a küszöb átrepülése utáni

[13] *Area Navigation.*

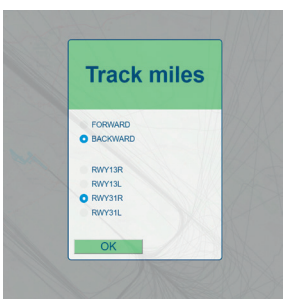
radarjeleket el kell távolítani, mert ha csak egyszerűen az utolsó érzékelt radarjeltől, amely járatonként eltérő, történik visszafelé az összesítés, akkor szintén nem egymással összevethető eredményt kapunk (12. ábra).



12. ábra
A számításhoz nem szükséges radarjelek törlése [a szerző]

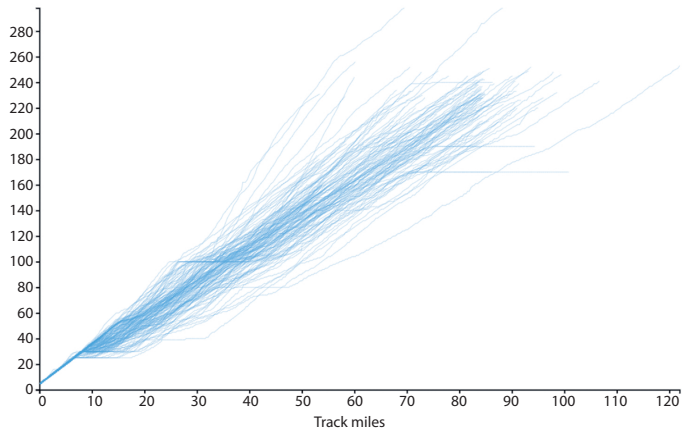
A pálya felett lévő radarjelek kijelölése nem feltétlen kell hogy hajszálpontosan történjen, az adatbázisban szereplő futópálya-koordináták alapján, mert a küszöb átrepülése előtti radarjel biztosan nem esik a pályaküszöbre. Ezért némi pontatlanság, amely a későbbiekben korrigálva lesz, a radarjelek frissítési gyakoriságából elkerülhetetlen.

Az említett pontatlanságot az „*add track miles*" funkció kezeli, ami egy input ablakon keresztül bekéri a távolságok összesítésének irányát, valamint a referenciapontként kezelhető pályaküszöböt. Ezután az időrendben utolsó radarjel lerepült távolság értéke a referenciaponttól való távolság lesz, az összes többinél pedig az előző távolság plusz az előző radarjel és az aktuális távolsága (13. ábra).



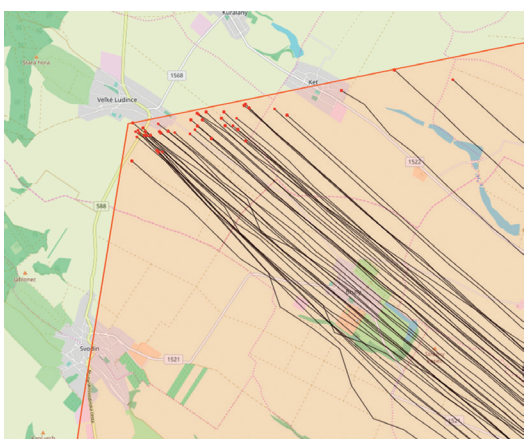
13. ábra
Többcélú funkció kezelőpanelje [a szerző]

Ez a funkció szintén nagyon sokoldalú, hiszen például a süllyedési profilok elemzése, kirajzolása e nélkül nem tehető meg. A 14. ábránál, ha nincs távolságérték, akkor nincs X tengely, és ha a futópálya területén lévő radarjelek nincsenek kiszűrve, akkor az eltérő utolsó radarjel-pozíció miatt a várttal ellentétben nem rajzolódna ki a fix 3 fokos siklópálya az utolsó 6 NM (11,1 km). Azaz járatonként különböző lenne az X tengelyen a nulla pont.



14. ábra
Érkező légi járművek süllyedési profilja [a szerző]

Az elméleti legrövidebb útvonal számítása a MergeStrip koncepciónak [13] megfelelő módon történik. Ehhez először az adatbázisban szereplő TMA körvonala alapján meg kell határozni azokat a radarjeleket, amelyek éppen beléptek az APP[14]-irányítók illetékességi légterébe (15. ábra).



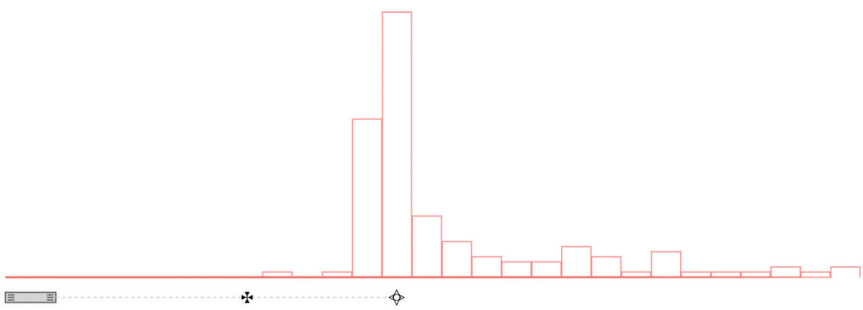
15. ábra
Adott területen a járatonkénti első radarjel [a szerző]

[14] *Approach.*

A kijelölt radarjelekhez már társítható a megfelelő pályaküszöb kiválasztása után a TMA-ba való belépés helyétől függő T-bar eljárás IAF-jéig terjedő távolság és az eljárás hosszának összege. Mivel minden egyes radarjelre meg lett határozva a küszöbig lerepült távolság, az előbbi fázisban kijelölt radarjeleknél a két számított érték különbsége az előállítandó potenciális magyarázó változó.

3.5. Eredmények prezentálása

A Python és R programozási nyelvekhez több kiváló vizualizációs eszköz létezik, amelyekkel könnyen lehet a leggyakrabban használatos grafikonokat jó minőségben előállítani. Ezek azonban nem adnak lehetőséget az egyedi elgondolások megvalósítására. Példánkban a cél annak megmutatása, hogy a publikált T-bar eljáráshoz képest milyen arányban repülnek a járatok hosszabb, illetve rövidebb útvonalon. Ezt átfogalmazva azt is mondhatjuk, hogy a kérdés az, hogy a végső egyenes meghosszabbított vonalára az érkezők a közbülső megközelítési pont (IF[15]) előtt vagy után kerülnek és mennyivel. Ebből kiindulva már egy kifejezőbb ábra is alkotható (16. ábra).



16. ábra
T-bar eljáráshoz képesti repülések hossza [a szerző]

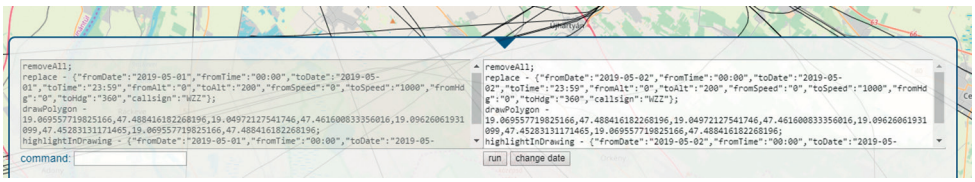
Az X tengelyen számok nélkül is egyértelmű, hogy a vizsgált adatoknál a végső egyenes kezdetét jelölő kereszt előtt senki sem került a futópálya középvonalára, és a leggyakoribb útvonal az eljárás közbülső megközelítési pontját jelölő ponton, azaz a T-bar eljáráson keresztül vezetett, azaz a lerepült útvonal és az elméleti legrövidebb útvonal különbsége 0.

3.6. Automatizáció

A HungaroControl Zrt. Módszertani Csoportjában fejlesztés alatt álló elemző rendszerben minden manuálisan kiadott parancsnak van egy szöveges megfelelője. Ezt és az ehhez esetlegesen

[15] *Intermediate Fix.*

tartozó paramétereket (például rajzolt poligon koordinátái) a rendszer visszajelzi a felhasználónak a munkavégzés során, így az elmenthető makróként, vagy módosítható. A leggyakoribb módosítási igény a dátum, azaz ugyanazt az elemzést kell lefuttatni más időszakra.



17. ábra
Makró futtatásának kezelőfelülete [a szerző]

A makró inputként bevihető a rendszerbe, a 17. ábra jobb oldali paneljébe, így egy elemzés egyszeri elvégzése tetszőleges számban és időszakra újra futtatható. Ez nemcsak a nagyobb adatmennyiségen történő adatelemzést könnyíti meg, de az elemzés eredményéből adódó szakvélemény is pontosan és bárki által reprodukálhatóvá válik, hiszen dokumentáció mellékleteként csak az elmentett parancsok sorozatát kell csatolni. Így az elemzési folyamat nemcsak hatékony, de transzparens is.

4. Következtetések

A felderítő rendszerekből származó adatok elemzése során elengedhetetlen a nyers adatok és a származtatott eredmények vizualizációja. Amennyiben az elemzési munka hatással lehet az ATM funkcionális rendszerre, nem javasolt a munkafolyamatok leegyszerűsítése, a lehető legpontosabb végeredményre kell törekedni, ami azonban sok munkával jár. A munkafolyamatok automatizálhatósága ezt megkönnyíti, továbbá biztosítja az eredmények reprodukálhatóságát.

Felhasznált irodalom

[1] The Official Blog of Kaggle.com, *Q&A with Xavier Conort.* Online: http://blog.kaggle.com/2013/04/10/qa-with-xavier-conort/

[2] C. Byrne, *Development Workflows for Data Scientists.* Sebastopol, California, O'Reilly Media, 2017.

[3] Szarvas D., Tichy R., Rohács D., „Mesterséges intelligencia alkalmazása az aviatikában," *Repüléstudományi Közlemények,* 31. évf. 1. sz. pp. 183–204. 2019. Online: https://doi.org/10.32560/rk.2019.1.15

[4] P. Domingos, „A Few Useful Things to Know about Machine Learning," *Communications of the ACM,* Vol. 55 No. 10. pp. 78–87. 2012. Online: https://doi.org/10.1145/2347736.2347755

[5] Z. Wang, M. Liang, D. Delahaye, „Short-Term 4D Trajectory Prediction Using Machine Learning Methods," In *SID 2017, 7th SESAR Innovation Days,* 2017. pp. 1–9. Online: www.sesarju.eu/sites/default/files/documents/sid/2017/SIDs_2017_paper_11.pdf

[6] D. Cielen, A. D. B. Meysman, M. Ali, "The Data Science Process," In *Introducing Data Science.* New York, Manning Publications, 2016.
[7] F. Herrema, et al., „A Novel Machine Learning Model to Predict Abnormal Runway Occupancy Times and Observe Related Precursors," In *12th USA/Europe Air Traffic Management Research and Development Seminar* (ATM2017) 2017. pp. 1–11. Online: https://pure.tudelft.nl/ws/portalfiles/portal/31444878/12th_ATM_RD_Seminar_paper_107.pdf
[8] Z. Wang, M. Liang, D. Delahaye, Automated Data-Driven Prediction on Aircraft Estimated Time of Arrival, *SID 2018, 8th SESAR Innovation Days*, 2018. pp. 1–8.
[9] V. Kumar, L. Sherry, R. Kicinger, „Runway Occupancy Time Extraction and Analysis Using Surface Track Data," In *Transportation Research Board Annual Meeting, Transportation Research Board Paper,* 10-3676, Washington, D.C., Jan. 2010.
[10] S. Ayhan, P. Costas, H. Samet, „Predicting Estimated Time of Arrival for Commercial Flights," In *KDD '18 Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* London, 2018. pp. 33–42. Online: https://doi.org/10.1145/3219819.3219874
[11] R. Madacsi, R. Markovits-Somogyi, „Bank Angle Estimation Using Radar Data," *Periodica Polytechnica Transportation Engineering,* Vol. 47, No. 1. pp. 1–5, 2019. Online: https://doi.org/10.3311/PPtr.11653
[12] C. O'Neil, R. Schutt, *Doing Data Science: Straight Talk from the Frontline.* Sebastopol, California, O'Reilly Media, 2013.
[13] Madácsi R., Baráth M., Sándor Zs., *A speciális térgeometriára támaszkodó „PointMerge" légiforgalmi irányítási módszer továbbfejlesztése.* Budapest, IFFK, 2015.

Data Science Workflow in Radar Data Analysis

Data science is one of the hottest topics in the 21st century. The reason for this is probably the emergence of advanced machine learning algorithms based on neural networks, with which the possibilities seem to be endless. Therefore, those companies which do not want to be left behind, have to invest in this field heavily. However, most of the time the tasks that need to be done before applying machine learning algorithms do not get enough attention. These are data cleaning, filtering, transforming, feature engineering, which can affect the accuracy of the model more than the selection of the algorithm or its parameters. Quite a few tools are available for free, which makes the data science workflow efficient, although in data analysis focusing on ATM developing bespoke software is often necessary. The article aims to present the most common requirements of that through examples and a small case study.

Keywords: *data analysis, machine learning, artificial intelligence, ATM, data visualisation, data cleaning*

Madácsi Richárd Szenior légiforgalmi eljárástervező és adatelemző HungaroControl Magyar Légiforgalmi Szolgálat Zártkörűen Működő Részvénytársaság Módszertani és Koordinációs Osztály richard.madacsi@hungarocontrol.hu orcid.org/0000-0002-3132-4679	Richárd Madácsi Senior Flight Procedure Designer and Data Analyst HungaroControl Hungarian Air Navigation Services Pte. Ltd. Co. ATS Operational Planning and Airport Coordination Department richard.madacsi@hungarocontrol.hu orcid.org/0000-0002-3132-4679