



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Távközlési és Médiainformatikai Tanszék

Dobreff Gergely

**NAGY PONTOSSÁGÚ POZÍCIÓS
ADATOKON ALAPULÓ ÁLTALÁNOS
KÜLTÉRI ÉS BELTÉRI EMBERI
MOBILITÁSI MODELL**

TDK dolgozat

KONZULENSEK

**Dr. Toka László, Dr. Biczók Gergely,
és Dr. Gódor István**

BUDAPEST, 2018

Tartalomjegyzék

1 Bevezetés	4
2 Az adatok előkészítése.....	6
2.1 Az adathalmaz bemutatása	7
2.2 A campus térbeli felosztása.....	9
2.3 A pozícióadatok helyszínekké történő transzformálása	11
2.4 Pozícióadatok összefűzése mozgássá.....	15
3 Adatleírók létrehozása és értékeik elemzése	20
3.1 A metrikák elkészítése.....	20
3.2 Kiugró mozgások detektálása és szűrése.....	22
3.3 A legfontosabb adatbeli tulajdonságok kiválasztása	26
3.4 Felhasználói csoportok elkülönítése klaszterezéssel.....	29
4 Beltéri mozgási modell építése	34
4.1 Inicializálás.....	35
4.2 Mozgási ciklus.....	37
4.2.1 Következő szoba kiválasztása	38
4.2.2 Állapot átmenetek	41
4.2.3 Attraktorok	45
4.2.4 Látogatási idők	48
4.2.5 Mozgási sebesség és útkeresés	50
4.3 Csoportdinamika	51
5 Alkalmazási lehetőségek	52
6 Összefoglalás	54
7 Irodalomjegyzék	56

Kivonat

Manapság szinte az összes okostelefon és hordozható eszköz képes a globális vagy helyi pozicionálásra, és a navigációs rendszerek a mindennapi életünk részét képezik. Az utóbbi években a beltéri pozicionálásra használt technológiák fejlesztésére koncentráltak az olyan kiaknázatlan lehetőségek miatt, mint a beltéri útkeresés, személyzetfelügyelet, helyfüggő azonosítás és így tovább. Az emberi mozgás jellemzőinek megértése és leírása sok éve a figyelem középpontját képezi különböző kutatói csoportoknak. Távközlésben dolgozó mérnökök, programozók, statisztikai fizikusok és közgazdászok méréseket végeznek, elméleteket gyártanak és hozzáadott-értékkel bíró pozíció alapuló szolgáltatásokat fejlesztenek. A mobiltelefonhálózatok, a globális helymeghatározó rendszer (GPS), a helyspecifikus közösségi platformok (pl.: Foursquare) és az úgynevezett crowdsensing elterjedésével a rendelkezésre álló adathalmazok mennyisége és változatossága, melyekből mobilitási tulajdonságokat lehet kinyerni, illetve amik alapján a modelleket validálni lehet folyamatosan nő. Érdekes módon a beltéri mobilitás vizsgálata messze kevesebb figyelmet kapott eddig, mint a kültérié, vagy a városié.

Jelenleg nincs beltéri emberi mozgásra egy átfogó generatív és adatvezérelt modell, melyet gyakorlati célokra is fel lehetne használni. Nekünk szerencsére egy gyedi, több mint 110 millió rekordból álló adatkészlet használatával lehetőségünk nyílik nagy pontosságú mozgási paraméterek kiszámítására. Olyan emberi mozgási modellt építünk, mely valós nyomkövetési adatokon alapszik, egyszerű matematikai módszerekkel leírható, kültért és beltért magában foglaló környezetre illeszkedik, mint például egy egyetemi campus, iroda park vagy egy városi kerület és lefedi mind az egyéni, mind a csoportos mozgást. A modell szimulációk elvégzésére használható építkezések tervezési fázisában vagy vészhelyzeti kiürítési tervek készítésére.

1 Bevezetés

Az emberi mozgás jellemzőinek megértése és leírása sok éve a figyelem középpontját képi különböző kutatói csoportoknak. Távközlésben dolgozó mérnökök, programozók, statisztikai fizikusok és közgazdászok méréseket végeznek, elméleteket gyártanak és hozzáadott-értékkel bíró pozícion alapuló szolgáltatásokat fejlesztenek. A mobiltelefon hálózatok, a globális helymeghatározó rendszer (GPS), a helyspecifikus közösségi platformok (pl.: Foursquare) és az úgynevezett crowdsensing elterjedésével a rendelkezésre álló adathalmazok mennyisége és változatossága, melyekből mobilitási tulajdonságokat lehet kinyerni, illetve amik alapján a modelleket validálni lehet, folyamatosan nő. Érdekes módon a beltéri mobilitás vizsgálata messze kevesebb figyelmet kapott eddig. A nehézséget az jelenti, hogy a GPS jel épületen belül nem elég erős, hogy pontos pozíciót meg lehessen meghatározni. Azonban az üzleti motiváció jelen van, mivel a pozícion alapuló szolgáltatások használhatósága nem lenne nagy, ha csak kültéri helyszínen működnének. Valójában, egyesek becslése szerint a városi térségben élők az idejük 80-90%-át beltérben töltik [8]. Szerencsére a beltéri helymeghatározó technológiák egyre bővülnek, lehetővé téve az okostelefon-felhasználók, vállalkozások és szoftverfejlesztők számára, hogy meghatározzák emberek és tárgyak helyzetét az épületeken belül.

Sok felfedezés történt az emberi mobilitás mintázatáról makroszkopikus skálán. Néhány tanulmány nagyméretű, mobiltelefonból származó pozíciós adatok alapján adott leíró eredményeket [6], mások pedig analitikus mobilitási modelleket fejlesztettek ki [11] [3] [16], melyek messze túlmutatnak az első leegyszerűsítő és nem igazán realisztikus Random Waypoint modellen [9]. Másrészt a beltéri, illetve a bel- és kültéri (városi) mobilitási modellekből kevés van és azok is egyszerűek. A tanulmány [10] szerzői egy beltéri mobilitási modellt fejlesztettek ki magas épületes környezetre, a mobil cellák közötti adat átvitelének és a bázisállomások lefedettségének tervezésének összefüggésében. Aschenbruk et al. [1] egy speciális beltéri mobilitási modellt hozott létre városi hadviselésre és taktikai kommunikációra. Prentow et al. [15] több esettanulmányt mutatott be egy kórház WiFi hálózatának monitorozásával, az emberi forgalom áramlásának és a beltéri közlekedések dinamikájának vizsgálata kapcsán. Ezen felül Yoshimura et al. [22] részletesen elemezte egy múzeum látogatóinak mozgási mintázatát, mely során Bluetooth technológiát használtak adatgyűjtésre. Bár sok érdekes megfigyelést tettek, a legjobb tudásunk szerint nem létezik teljeskörű,

adatvezérelt, generatív, bel- és külteret magában foglaló mobilitási modell, melyet gyakorlati célokra lehetne felhasználni. Ilyen például a felhasználók mozgásának szimulálása épületek tervezése során az építkezési erőforrások minél jobb felhasználása érdekében. Ennek a hiánynak az egyik magyarázata az, hogy általában nem áll rendelkezésre kellően részletes és nagy adathalmaz.

Ebben a dolgozatban egy egyedi adathalmazt használunk, mely egy norvég egyetemi campus útvonaltervezéssel foglalkozó szolgáltatásából, a MazeMapból [12] származik és létrehozunk a maga nemében egy első emberi mobilitási modellt kül- és beltéri használatra. Célunk egy olyan modell kialakítása, mely

- i) analitikus jellegű
- ii) kevés paraméterrel rendelkezik
- iii) adaptálható egy általános, több-épületes környezethez, és
- iv) magában foglalja az egyéni és a közösségi mobilitási jellemzőket.

A modell legfontosabb jellemzői a következők:

- a) mobilitási gráfon alapul, ahol minden csomópont egy időben változó attraktor
- b) kihasználja az épített környezettel járó hármass hierarchiát (épület, emelet, szoba)
- c) adatvezérelt módon statisztikai eloszlást használva határozza meg az emberi mozgás következő állomását
- d) közvetlenül alkalmazható számos gyakorlati esetre, mint például az épülettervezés, erőforrás-menedzsment, helytől függő vészhelyzeti protokollok kialakítása.

A dolgozat további része az alábbiak szerint van felépítve. A 2. fejezet bemutatja az adathalmazt, amivel dolgoztunk és leírja az adatelőkészítési lépéseket. A 3. fejezet a mozgási metrikák létrehozásának módszerét, az adathalmaz szűrését és a felhasználói csoportok feltárásának folyamatát tárgyalja. A 4. fejezet a beltéri mozgási modell működését és struktúráját vázolja. Végül az 5. fejezetben a modell gyakorlati felhasználási lehetőségeit tárgyalom, majd a 6. fejezet egy összeggel zárja a dolgozatot.

2 Az adatok előkészítése

A MazeMap nevű szolgáltatás egy norvég vállalat és egy norvég egyetem, az NTNU (Norwegian University of Science and Technology) együttműködésével készült. Ezt a szolgáltatást, mint mobil alkalmazást hozták létre és először azért fejlesztették ki, hogy segítsen a hallgatóknak és a látogatóknak eligazodni a campuson, könnyen megtalálják az egyik teremből a másikba vezető legrövidebb utat az egyetemen és a szabadban egyaránt. A campus 60 épületből áll, melyek 13 000 szobát tartalmaznak, ami összesen 350 000 négyzetméter alapterület tesz ki. A MazeMap egy hely-alapú szolgáltatás, mely a MazeMap alkalmazásprogramozási felületét (API) használja a felhasználók pozíciójának és a legrövidebb út kiszámításához az egyes pontok között az egyetemen. A szolgáltatást mind kültérben mind beltérben lehet használni. Kültéri navigáció esetében a szabványos GPS helymeghatározást használja, amennyiben a felhasználó eszköze rendelkezik GPS vevővel. A beltéri helymeghatározás WiFi Access Pointok (AP) segítségével háromszögletes módszerrel történik. Az egyetemi campus több, mint 1 800 darab WiFi AP-vel van felszerelve.

Az NTNU WiFi infrastruktúrája Cisco MSE (Mobility Service Engine)¹ vezérlőt alkalmaz egy helymeghatározó motorral, mely az eszközök pozícióját az AP-okból háromszögletesen kapja meg. A lefedettség a 350 000 négyzetméteren a durván 1 800 AP-n alapul. A nyomkövetés passzív helymeghatározáson alapul, vagyis bármely eszköz, melynek be van kapcsolva a WiFi-je nyomon követhető a szétküldött Probe Request üzenetei segítségével. Ezeket az üzeneteket az eszköz változó időközönként továbbítja, amivel a mintavételezési frekvencia kevesebb, mint 70 másodperc lesz. A rendszer konfigurációjából adódóan főként beltéri lefedettséget biztosít, de kiterjed az épületekhez közeli kültéri területekre is. Ezáltal a beltéri mozgáson kívül a kampusz valamennyi kültéri mozgását is rögzíti.

Az adatgyűjtésben egy absztrakciós és feldolgozási réteget is alkalmaznak. Az összegyűjthető adatok már fel vannak dolgozva és anonimizálva vannak, a nyers adatokhoz nincs hozzáférésünk. A backendben elérhető adatok ki vannak törölve és számunkra nem hozzáférhetők, ilyen például a készülék típusa, modellje vagy a WiFi hálózat forgalma. A feldolgozott adat kimenete, amihez hozzáférésünk van, olyan adatpontokból áll, mely az

¹ <https://www.cisco.com/c/en/us/support/wireless/3365-mobility-services-engine/model.html>

anonimizált eszköz ID-ból, a hosszúsági és szélességi koordinátapárból, a rögzítés helyéből (campus és épület neve, emelet száma), a mérés pontosságából, az időbélyegből és a hash sójának időbélyegéből áll. Az anonimizálás hash függvényt használ, mellyel az eszköz MAC címéből egy véletlen azonosítót hoz létre. Ez naponta változik azáltal, hogy a hash sóját megváltoztatja. Így biztosítja, hogy egyetlen készülék se legyen nyomon követhető egy napnál hosszabb ideig. Emiatt szerepel a só időbélyegzője az adathalmazban. Amíg a só időbélyege nem változik, az azonosító ugyanazt az eszközt azonosítja. Az adatok belső egyetemi használatára ezt a folyamatot az NTNU IRB (Institutional Review Board) jóváhagyta. A személyes adatok védelme különösen fontos az ilyen nagy felbontású adatok esetében. Ennélfogva a mi elemzésünk és modellezésünk nem egyéni, hanem általános mintákat használ.

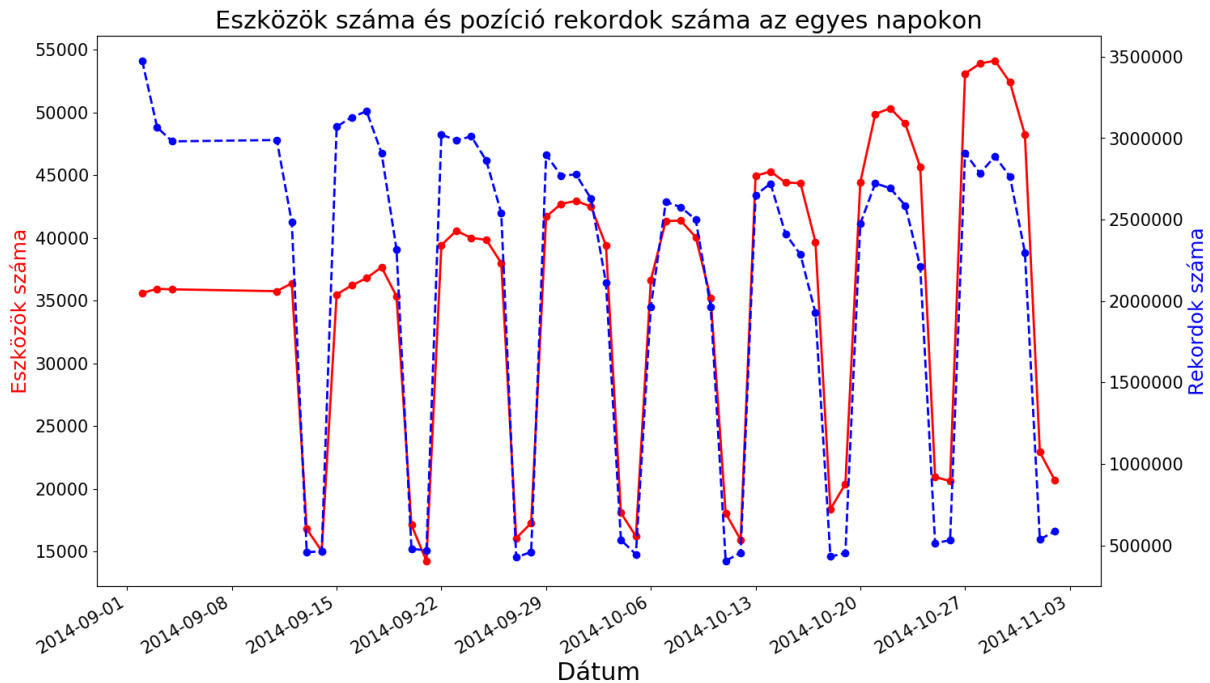
2.1 Az adathalmaz bemutatása

Az adathalmazban a pozíció hierarchia mező (*hierarchy*) három elemet tartalmaz: a campus nevét, az épület nevét és az emelet számát. Fontos megjegyezni, hogy ez a hierarchia a legerősebb WiFi AP helyzetéből lett meghatározva, annak a helyzete (melyik épület, melyik emelet) lett rögzítve. Előfordulhat, hogy az eszköz² egy olyan AP-ra csatlakozik, ami nem azon a szinten van ahol ő, így a hierarchia mező nem a valós emeletet mutatja. További részletezés, mint például, hogy az eszköz melyik szobában van, pontosan nem szerepel az adathalmazban. Ezt a hármas tagozódást külön mezőkben szerepeltettük a könnyebb kezelhetőség érdekében.

Mivel az egész adathalmaz egy campusról származott, így a campus mezőt eldobtuk. Az adathalmazban szereplő pontosság mező (*accuracy*) megadja a koordináták kiszámolásakor lehetséges mérési hiba becsült sugarát minden egyes méréshez. Ebben a dolgozatban a pontosságot nem használjuk a MazeMap fejlesztőinek tanácsára, mivel nem megbízható adatról van szó. A hosszúsági (*longitude*) és szélességi (*latitude*) koordináták szabványos GPS koordináták. Az azonosító (*id*) mező egy eszközt egy egész napon keresztül azonosít, majd minden nap végén új azonosítót kapnak. Az azonosítók újraszámolásának, azaz a hash sójának változásának időbélyegét a *salt_timestamp* mező tartalmazza. Fontos megjegyezni, hogy ez körülbelül 0:00-kor történik UTC időzóna szerint, ami Norvégiában hajnal 2:00-át jelent. Ezen periódusok nem pontosan 24 óra hosszúak, egy kis eltérés mindig van.

² A dolgozatban a felhasználó, ember és eszköz kifejezés ugyanúgy az adathalmazban rögzített WiFi képes eszközökre utal, megelőlegezve azt, hogy ezeket egy ember hordozta magával.

Hétköznaponta átlagosan körülbelül 42 ezer, míg hétvégente 18 ezer darab eszközről készült feljegyzés. Ez összesen hétköznaponta átlagosan több, mint 4,2 millió, míg hétvégente több, mint 470 ezer pozíciórekordot jelent. A 2.1. ábrán látszik, hogy hány felhasználó és hány rekord található az egyes napokon.



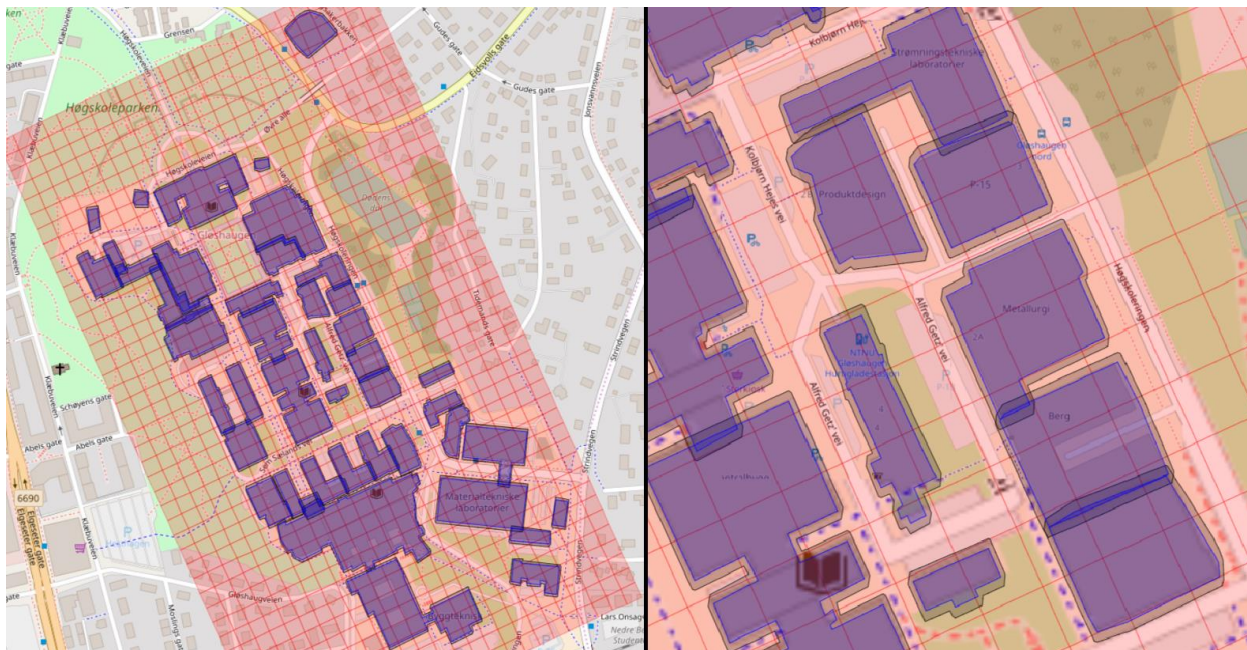
2.1. ábra: Naponta rögzített rekordok és eszközök száma. Hétvégi kis aktivitás jól megfigyelhető az ábrán.

A felhasználók által bejárt terület jól körülhatárolható. Az objektumok egyértelmű hierarchiát mutatnak: a campus épületekből, kültéri és beltéri útvonalakból áll. Az épületek liftekkel és lépcsőházakkal összekötött emeletekből állnak. Magukat az emeleteket egy sor szoba, folyosók és liftek, illetve lépcsőházak alkotják, melyeket alaprajzokon rögzítettek. A modellben a szoba a legkisebb pozícionálás egység amit figyelembe veszünk, ezt mind az intuíció, mind az a tény diktálja, hogy a mérési pontatlanság - ami kb. 5-10m - miatt a nagyobb felbontású elemzés torz modellt eredményezne. Ez a döntés tükrözi az emberi magatartás sajátosságát egy ilyen környezetben, miszerint az embert valami cél mozgatja, ezért megy át egyik szobából a másikba. A modell szempontjából tehát nem fontos, hogy egy szobán belül, hol van az illető. Fontos még azt is megjegyezni, hogy a szoba szintű statisztikákat könnyen lehet összesíteni emelet továbbá, épület szinten, lehetővé téve ezzel a többszintű elemzést.

2.2 A campus térbeli felosztása

Rendelkezésre állt a MazeMap által használt alaprajz nagyjából minden egyes szobáról a campuson, melyeket poligonokként voltak tárolva, ahol minden egyes csúcs egy GPS koordináta volt. Ezen felül meg volt adva, hogy az adott szoba mely épület melyik emeletén helyezkedik el és hogy a szobának mi a típusa. Minden szoba a 87 kategória egyikébe tartozott, például: auditórium, folyosó, számítógépes labor stb. Ezen adathalmaz nagyban megkönnyítette azt a kihívást, hogy a felhasználók mindenkori helyzetét szobákhoz kössük.

A rekordok, ahogy azt fentebb szerepelt a GPS koordináták mellett tartalmazzák az épületet és az emeletet, így ezzel a három információval egy konkrét szobába sorolhatók a rekordok. Azonban a modellnek számolnia kellett azzal, hogy ez egy kevert környezet, ami belteret és külteret is magában foglal. Illetve figyelembe kellett venni a mérési pontatlanságot, mely a koordináták kiszámolásánál ugyan esetenként lehet, hogy csak 5 méter eltérést jelent, de néhol ez a „kint” és a „bent” összetévesztését eredményezheti – például egy ablak mellett ülő oktató esetében. Annak érdekében, hogy ezt tudja a modell kezelni, 3 fő részre osztottuk fel a campust és ezáltal a felhasználók pozícióját. Ez a három kategória a következő: az épületeken belüli *beltéri rész*, az egyes épületek „*vonzáskörzete*” és a campus külterületét lefedő rácsban elhelyezkedő „cellák”. Ez a felosztás látható a 2.2. ábrán. A fő koncepció az volt, hogy azokat a koordinátapontokat, amiket el tudunk helyezni konkrét szobákban arra értelemszerűen azt mondjuk, hogy ezek beltérben vannak. Az olyan koordinátapontok, melyek nagy valószínűséggel egy adott épület adott emeletén belül vannak, de – különböző okok miatt - pontosan nem tudjuk, hogy hol, azokat az adott épület „*vonzáskörzetében*” helyeztük el. Végül azon pontok, amelyek biztosan kültéri környezetben tartózkodnak a campust lefedő rács egy cellájához rendeltük.



2.2. ábra: A campus felosztása két nagyításból. Pirossal a kültéri cellák, feketével az épületek vonzaskörzete, kékkel az épületek vannak jelölve.

Ugyan a campus nagyon sok szobájáról megvolt a koordinátákkal körülírt alaprajz, némely szoba azonban hiányzott ebből az adathalmazból, némelyek viszont rosszul lettek felrajzolva és voltak különleges esetek is. Ezeket a hibákat a kapott alaprajz kirajzolásával és a referencia oldal, a MazeMap manuális összehasonlításával fedeztük fel. Több oka lehetett amiért nem adták meg a szobák alaprajzát. Ilyen lehetett az is, hogy az adott szoba a valóságban egy folyosó aminek a közepén egy galéria van. Ezt lyukas poligonként kellett volna megadniuk, amire nem volt lehetőség, ezt az ilyen esetekben kihagyták az adathalmazból. A különleges esetek az egyetemi környezet sajátosságából adódtak, miszerint a kétemeletes előadótermeknek a második emeleti része a tervrajzon nem volt feltüntetve. Pontosabban csak – az építész terminológiát használva – beépített területként, egy X-el áthúzva volt jelezve a MazeMap oldalán, a mi kapott adathalmazunkban viszont nem szerepelt, mint szoba. Ez akár egy bevásárló központ, vagy egy irodaházban is előfordulhat egy galéria esetében. Ezen hiányosságokat az adathalmazban kézzel pótoltuk, kijelölve a szobákat befoglaló poligonok csúcsait és az ezekből készített sokszögeket az adathalmazhoz adtuk. Így manuálisan összesen 113 szobát vettünk fel az adathalmazba.

Miután pótoltuk ezen hiányosságokat az összes szobákat leíró poligonok száma 10 602 volt. A napi átlagosan 2 millió pont esetében minden egyes poligonra leellenőrizni, hogy tartalmazza-e az adott pontot nagyon számításigényes feladat lett volna. Ezért minden egyes

épületnek manuálisan létrehoztuk a körvonalát. Egy épület körvonalát, az összes emeletének összes szobája által lefedett kétdimenziós területként, mint sokszöggként definiáltuk. Így megkaptuk azt a legkisebb területű poligont, ami az épület minden szobáját magában foglalja.

Megelőzendő, hogy a koordináta pontok kiszámolásakor fellépő akár több méteres pontatlanság miatt egy felhasználóról tévesen mondjuk hogy kint tartózkodik az utcán, a modellnek kezelnie kell ezen tévedéseket, mivel ez nagyban befolyásolja a modell pontosságát. Ezért az épületek körvonalát 3 méterrel megnyújtottuk, ezt egy úgynevezett poligon eltolás módszerrel értük el. Az így kapott sokszög oldalai 3 méterrel távolabb vannak az eredeti körvonal oldalaitól. A három méteres határt azért választottuk, mert sok épület nagyon közel van egymáshoz így, ha nagyobb határral dolgoznánk, akkor egymásra lapolódna ezek a körvonalak, illetve ez a koordináta pontok pontatlanságával is összhangban van. Így kaptunk egy olyan körvonalat, amibe a fent említett hibás mérési pontok bele eshetnek, ezt az épület vonzaskörzetének, *boundary*-nak neveztük.

Természetesen a campus kültéri részén nincsenek szobák, nincs megadva semmilyen adathalmaz koordinátákkal, hogy az egyes parkok, sétálóutca vagy utak hol helyezkednek el. A modell készítésénél célunk volt, hogy a kültéri mozgást is szimulálni tudja, ezért a kinti világot fel kellett osztani. A campus területét 20x20 méteres négyzetekből álló ráccsal fedtük le, ez a rácsszerkezet az egész campust lefedi, tehát minden koordináta pontot, ami nem beltéri, lefed. Ezáltal a kültéri mozgást kisebb cellák közötti mozgássá képeztük le. A méretek kiválasztásánál a campus útjait, az épületek közti távolságot, a mintavételezési frekvenciát és az átlagos emberi gyaloglás sebességét is figyelembe vettük. A túl finom rácsozás azért nem jó, mert például a pontatlanság miatt egy épület bejárata előtt cigiző oktatót hamisan akár mozgó embernek tüntetünk fel miközben csak pár lépést tesz fel-alá. A túl nagy területű négyzetek viszont egy mozgó emberről azt a benyomást kelthetik, hogy nem is mozgott, holott a valóságban igen, csak akkora a cella, hogy a mozgása nem járt cellaváltozással.

2.3 A pozíció adatok helyszínekké történő transzformálása

Miután a campust felosztottuk a fent tárgyalt három részre, elhelyeztük az összes koordináta pontot az egyik rész valamelyikében. Így három eset volt lehetséges.

- A. Az adott pont egy épületen belülre esett. Azaz nagy valószínűséggel azon az épületen belül valamelyik szobában van az eszköz. Rögzítettük, hogy ez az eszköz melyik épületben van.

- B. Az adott pont egy épület vonzáskörzetébe (boundary) esett. Ez lehet amiatt, hogy az eszköz valóban kint van az utcán, de közel az épülethez, vagy csak mérési hiba miatt került oda. Rögzítjük, hogy az eszköz az épületben van ugyan, de az emelet meghatározásán túl nem tudjuk pontosabban meghatározni. A későbbiekben a szűrés során foglalkozunk ezzel.
- C. Az adott pont se az épületbe, se az épület vonzáskörzetébe (boundary) nem illik. Ilyenkor az eszköz kint tartózkodik. Ekkor rögzítjük, hogy melyik kinti cellában van az adott pont.

Az A esetben tovább részletezve a felbontást, a koordinátákat nem csak épületbe, hanem konkrét szobákba soroljuk. Mivel már adott, hogy a koordináta melyik épületbe esik és adott, hogy melyik emeleten van az adott felhasználó, így csak pár szoba poligonjával kellett metszést vizsgálni, ami jóval meggyorsította a folyamatot. Fontos megjegyezni, hogy a nyers adathalmazban szereplő hierarchia mezőből kinyert épület nem feltétlen azonos a koordináta szerinti épülettel. A koordinátapontok szobákba sorolása közben három eset volt lehetséges:

- I. Az adott pont beleillett egy szobába. Megtaláltuk a felhasználó helyét, rögzítjük, hogy az adott rekord melyik szobába tartozik
- II. Az adott pont nem illett egy szobába se az adott épület adott szintjén.
- III. Az adott koordinátapont szerint meghatározott épületnek nincs olyan emelete, ami a rekord hierarchia mezőjében szerepel.

A III. eset úgy fordulhatott elő, hogy a koordináta pont szerint nem abban az épületben van az eszköz, ami a hierarchia mezőben van megadva és a koordinátapont szerinti épületnek nincs a hierarchia mező szerinti emelete. Ez két esetben fordulhat elő, vagy a koordinátapont helyes és az eszköz egy szomszédos épület WiFi AP-jára van felcsatlakozva. Például egy magas épület mellett van egy alacsony és az alacsonyabb épületről a magasabb WiFi-jére csatlakozik egy eszköz. Vagy a hierarchia mező szerinti épület és emelet a helyes, csak a koordinátapont nagyon pontatlanul lett kiszámolva és a szomszédos épületet jelöli. Mivel a hierarchia mező pontosabb képet ad a valódi helyzetről és a koordinátapont szerint nem tudjuk szobába sorolni ezen pontokat, így eldobtuk ezen rekordokat.

A II. eset amiatt van, mert a szobákat meghatározó sokszögek a hasznos alapterületet mutatják. Azaz a falakat nem tartalmazzák, így ha egy emeletet vizsgálunk tulajdonképpen egy lyukas alaprajzot kapunk, mivel a falakat és egyéb berendezések (villanyszekrény, kémény,

szellőzők stb.) nem tartalmazzák a sokszögek. Azonban a mérési pontatlanságok miatt előfordulhat, hogy egy koordinátpont a falban van. Ezt az esetet szemlélteti a 2.3. ábra. Ilyenkor nem tudjuk pontosan eldönteni, hogy az adott eszköz pontosan melyik szobában tartózkodik. Több lehetőség is van az eldöntésre. Lehetne az adott eszköz eddigi pozícióit nézni, hogy hol volt és ez alapján kiszámolni, hogy valószínűleg melyik a tényleges szoba ahol van. Ki lehetne számolni a koordinátponthez legközelebbi falat és azt a szobát választani. Ezek mind nagyon számításigényesek lennének minden pontra. Egyszerűség kedvéért lehetne venni a szobák középpontját és aszerint mérni a távolságot. Ez viszont nagyon torzítana, mivel a szobák mérete és formája nagyon eltérő, így egy nagyon hosszú folyosó középpontja nagyon távol esne a kérdéses ponttól.

Ezeket figyelembe véve egy gyors algoritmus mellett döntöttünk. Minden olyan pontot, ami úgymond a „falban” van, eltolunk egy véletlenszerűen választott irányba 2 méterrel és megnézzük, hogy így már metszenek-e valamilyen szobát, ha nem, akkor az eredeti helyzetükbe visszatéve őket 120 fokkal elforgatva toljuk el, újra leellenőrizzük, hogy van-e találat, ha így sincs, akkor 240 fokkal fordítjuk el és megismételjük a lépést. Ezzel mintha minden egyes ilyen pont körül egy 2 méteres sugarú körben leellenőriztük volna, hogy van-e szoba, és ha van, akkor odasoroltuk volna. Ez a módszer igen durva egyszerűsítésekkel él; nem pontosan körben ellenőriz, nem veszi figyelembe, hogy eddig hol volt az eszköz és nem sorol be minden pontot szobába. Az algoritmus viszont nagyon gyors és a mérések szerint relatíve kevés az a rekord amit nem tud besorolni. Figyelembe vettük azt is, hogy ezen esetekben nem kell túppontosan meghatározni a szobát, mivel a későbbiekben az esetleges zajok kiszűrése során az itt kreált hibákat ki tudjuk szűrni. Azon pontok, melyeket ezzel a módszerrel sem sikerült besorolni valamilyen szobába – tehát olyan pontok, amik körül 2 méteres körben nagy valószínűséggel nincs szoba – a *B* kategóriába soroljuk, ami szerint az adott épület vonzáskörzetében van az adott pont, azaz tudjuk, hogy az adott épület adott emeletén van az eszköz, csak azt nem tudjuk, hogy pontosan melyik szobában.



2.3. ábra: A Kjemiblokk 2 épület 4. emeletén található szobák (balra), illetve az ott rögzített olyan rekordok (fekete pontokkal jobbra), akiket nem sikerült szobába sorolni, mert a “falban” vannak.

Így minden rekordot a koordinátpontok alapján besoroltunk egy szobába, amennyiben nem tudtuk akkor a fentebb vázolt külterületi cellákba soroltuk, hogy majd a modellel a kültéri mozgást is szimulálni tudjuk. Azonban a fenti műveletsornak voltak melléktermékei. Annak következtében, hogy a kapott adathalmaz, mely a szobák alaprajzát tartalmazta nem volt tökéletes, a szobákba sorolás során duplikátumok keletkeztek. Voltak átlapolódások az épületek között, például két egymás melletti épület esetében, az egyik épület pincéje a másik épület alá nyúlt, így az épületek körvonala metszette egymást, ezért aki ebben a metszetben volt úgy tűnt – teljesen jogosan – hogy két épületben van egyszerre. Ugyanez volt megfigyelhető a szobák esetében, mivel volt olyan szoba, ami kétszer szerepelt az adathalmazban más néven, például egy folyosót kétszer tüntettek fel más kódnévvvel. Mivel egy felhasználó egy adott időpontban értelemszerűen csak egy helyen lehet, ezért a következő logika mentén kiszűrtük a duplikált rekordokat:

- a. Mivel az eredeti hierarchia mezőiben szereplő épületnév pontosabb támpontot ad az eszköz pozíciójáról, ezért, ha a duplikátumok között van, amelyik ebben az épületben van, azt választjuk.

- b. Amennyiben egy épületen belül van duplikátum, akkor a több szoba között azt választjuk, amelyiknek az a neve ABC-rendben előbb van. Mivel ilyenkor olyan esetről beszélünk, ahol egy épületben egy adott szinten két szoba fedi egymást, így minden pontot, ami ebbe a metszetbe kerül konzisztensen a kettő közül mindig az egyikbe soroljuk.

Ezen folyamat végeztével az egy napi pozíciórekordok (ami hétköznapi esetében kb. 4,2 millió darab) átlagosan 83%-át sikerült besorolni egy épületen belüli szobába, 9%-át épületbe ugyan sikerült besorolni, de azon belül pontosan szobába nem, tehát őket az adott épület vonzáskörzetébe helyeztük, míg a rekordok 0,01%-át nem sikerült sehova se besorolni pontatlanság miatt. A maradék 8%-át a rekordoknak kültéri cellához soroltuk.

A nyers adathalmaz módosításával egy olyan adathalmazt kaptunk, mely minden napon az összes rögzített eszköz mindenkori pozícióját egy általunk definiált hármas tagozódású rendszerbe sorolja be, ahol a legkisebb egység az épületen belüli szoba. Ez a térbeli felosztás minden ilyen kevert környezetben könnyen felállítható. Ezzel előkészítettük azt, hogy az egyes felhasználók (eszközök) mozgását szobák közötti közlekedéssel írjuk le.

2.4 Pozícióadatok összefűzése mozgássá

Annak érdekében, hogy a mobilitási modellhez az egyes felhasználók mozgását le tudjuk számokkal írni, definiálni kell, hogy mit értünk mozgás alatt. Minden felhasználónak meghatározzuk a napi bejárt útvonalát. Fontos megjegyezni, hogy naponta változik az adathalmazban a felhasználók azonosítója adatvédelmi okok miatt, így egy felhasználót nem tudunk napokon átívelően vizsgálni. A felhasználók által időrendi sorrendben meglátogatott szobák, helyek sorozatát tekintjük a bejárt útvonalnak. Itt lehet akár a meglátogatott szoba kültéri cella, vagy egy épület vonzáskörzete. Ez technikailag azt jelenti, hogy időrendbe téve egy adott felhasználóról szóló rekordokat, összevonjuk azon rekordokat, amelyek egymás után ugyanabba a szobába szólnak. Így ahelyett, hogy azt tárolnánk egy felhasználóról, hogy egy adott időpontban hol volt, azt rögzítjük, hogy egy adott intervallumban folyamatosan mely szobában volt. Ezt mi látogatási listának nevezzük, ami így néz ki: (room_id, enterTimeStamp1, leaveTimeStamp1) -> (room_id2, enterTimeStamp2, leaveTimeStamp2) -> ... -> (room_idN, enterTimeStampN, leaveTimeStampN). Ahol a *room_id* jelöli a szobát, az *enterTimeStamp* azt az időpontot, amikor a felhasználó belépett az adott helyiségbe, illetve a *leaveTimeStamp* azt, amikor azt elhagyta. Ugyan az *enterTimeStamp* megegyezik a megelőző szoba

leaveTimeStamp-jával, de az algoritmusok egyszerűbb működése érdekében mind két időpontot tároljuk az egyes szobáknál. Ebből az következik, hogy a modell feltételezi, hogy a felhasználó mindig van valamilyen szobában és szobaváltások azonnal megtörténnek.

A felhasználókról nyilvántartottuk a látogatási listájukat, ami tulajdonképpen a mozgási mintájuk. Ezen mintákat elemezve kiderült, hogy sok felhasználó természetellenesen viselkedik: nagyon gyorsan váltanak szobákat, emeleteket, sőt épületeket. Ezt életszerűtlenül gyakran teszik. A jelenség oka az, hogy az adathalmazban eredetileg is jelen volt és általunk is bekerült némi pontatlanság a szobák meghatározása során. A jelenség hasonlatos a mobil hálózatok cellahatárain lévő felhasználók egyik cellából a másikba történő ún. ping-pongozásához, ezért itt is ping-pong mozgásként hivatkozunk erre a jelenségre. Ez azt a magatartást takarja, amikor egy felhasználó egy szobából egy szomszédos szobába gyorsan átmegy és hirtelen vissza is tér ugyanoda. Ennek a hátterében az áll, hogy a koordináták kiszámolásánál az pár méteres pontatlanság akár egy szobányit is jelenthetett, ha például a felhasználó a fal mellett tartózkodott. Ráadásul, amikor a szobákba soroltuk be a koordinátapontok alapján a rekordokat és véletlenszerűen eltoltuk a pontokat, hogy szobát találjunk számukra, hibát vittünk a besorolásba. Ezek a hibák itt, a mozgási mintákban jelentkeztek.

Fontos megjegyezni, hogy a modell szempontjából tulajdonképpen lényegtelen, hogy egy A és egy B szomszédos szoba között „ping-pongozó” felhasználó a valóságban melyikben volt. A mozgás elemzéséhez csak az a fontos, hogy rögzítve legyen, hogy a kettő közül valamelyikben ott volt folyamatosan. Ezen magatartás kiszűrése során lényegében korrigáljuk az adott felhasználó pozícióját, tehát egy bizonyos feltétel rendszer alapján áthelyezzük egyik szobából egy másikba, mondván nem reális hogy abban a szobában tartózkodott. A szűrést a hármas hierarchia minden szintjén elvégezzük, így nem csak a hirtelen szobaváltásokat pontosítjuk, hanem az épület és emelet váltásokat is.

Az algoritmus minden hierarchia szinten a hármas szobaváltásokat vizsgálja, és ezekről dönti el, hogy reális-e. Így halad végig a látogatási listán, és a listát hármas szekvenciánkként vizsgálja. Ilyen vélhetően nem valós mozgás az, amikor egy felhasználó az alábbi sorrendben járja be a szobákat: A(10)-B(1)-A(20)-B(2)-C(30)-D(60)-..., ahol a betűk szobákat jelölnek és a zárójelben pedig az eltöltött percek vannak megadva. Látszik, hogy az A szobában sok időt töltött el az illető, de „átugrott” 1-2 percre a B szobába. Ez kiugró magatartás, feltehetően egész végig az A szobában volt, csak pontatlanság miatt szerepel az, hogy 1 percre a szomszédos

szobában volt. Ezt az algoritmus a következőre korrigálja: A(33)-C(30)-D(60)-..., pontosítva a mozgási mintázatát a felhasználónak. Megemlítendő, hogy a lentebb használt 140mp-es időhatárt, ami alapján gyanúsnak titulálunk egy mozgást a mintavételezési frekvenciából számoltuk. Úgy, hogy az átlagos idő két bejegyzés között egy felhasználóról 60-70 mp között van. Beleszámolva azt, hogy kimaradhat egy mintavételezés, így két bejegyzés között akár 140 mp is eltelhet.

Először a létrehozott látogatási listákban a gyanús épületváltásokat szűrjük ki minden felhasználó esetében, egyenként. Ebben a három esetben számított valótlanok egy szobahármas (a nagybetűkkel az egyes meglátogatott szobák épületét jelöljük, tehát A-B-A, esetben nem feltétlenül igaz, hogy a két A ugyanazt a szobát jelöli):

- 1) A-B-C sorrendben követték egymást a szobák épületei és B szobában 280mpnél kevesebb időt töltött el és B szoba nem egy kültéri „cellát” jelöl. Ekkor B szobát A-ra cseréljük.
- 2) A-B-A sorrendben követték egymást a szobák épületei és az első A kültéri „cella” és itt csak 140mp-nél kevesebb időt töltött el. Ekkor az első A szobát B-re cseréljük.
- 3) A-B-A sorrendben követték egymást a szobák épületei és B szobában 140mpnél kevesebb időt töltött el. Ekkor B szobát A-ra cseréljük.

A feltételrendszert érintő tervezői döntések magyarázatra szorulnak a könnyebb megérthetőség kedvéért. Az a tény, hogy a felhasználó a 2) és a 3) esetben 140mp-nél kevesebb időt töltött el azt jelenti, hogy ott 1 vagy 2 bejegyzésnél több nem született róla. A 2) esettel a 3) eset azon mellékhatását küszöböljük ki, hogy egy olyan esetben amikor valaki a fal mellett ül és tévesen az utcára van besorolva akkor, ha az A-val kezdődik egy ilyen hibás sorozat pl.: A-B-A-B-A-B, akkor az algoritmus 3) eset miatt az egész időszakra azt mondaná, hogy kint volt az illető. Így egy kicsit a beltér javára dönt az ilyen ping-pongos esetekben. Az 1) esetben pedig azt a mozgásfajta vizsgáljuk amikor valaki hirtelen 3 épület között közlekedik át, fontos itt kikötni, hogy az összekötő B szoba ne legyen kültér, mert az egy teljesen reális eset, ha valaki átsétál A-ból C-be úgy, hogy pár percig kint van B „cellában”.

Az emeletek szintjén az algoritmus hasonló logikával működik. Itt két esetben volt az emeletek közötti váltás módosítva (a nagybetűk emeleteket jelentenek, két azonos betű nem feltétlenül jelenti ugyanazt a szobát, csak az emeletek egyezőségére utal):

- 1) X-Y-X sorrendben követték egymást a szobák emeleti és az Y emelete csak egygyel tér el az X-étől és az Y szoba különleges szoba, akkor nem foglalkozunk vele, ezt a magatartást megengedjük.
- 2) X-Y-X sorrendben követték egymást a szobák emeletei és az Y szobában 140mp-nél kevesebb időt töltött. Ekkor az Y szobát X-re cseréljük

Különleges szobának számít a mosdó és a konyha, ha valaki ilyen szobát látogat meg, azt máshogy kell kezelnie az algoritmusnak. Elképzelhető egy olyan eset, amikor egy emeleten nincs mosdó, ezért valaki lerohan az egygyel lejjebbi szintre majd visszatér a szintjére. Ez nem számít valótlan mozgásnak, ezért az 1) esetben ezt kizárjuk. Amennyiben nem ez történik, az már gyanúsabb, hogy valaki emeletet vált majd 2 perc után vissza is tér ugyanarra a szintre.

Ezek után a szobák szintjén történik meg a kiugró magatartás kiszűrése. Négy esetben történt módosítás (a nagybetűk itt szobákat jelölnek, azonos betű azonos szobát jelent):

- 1) U-W-V sorrendben követik egymást a szobák és a W szobába csak egy darab bejegyzés történt és itt 70mpnél kevesebb időt töltött. Ekkor a W szobát U-ra cseréljük.
- 2) U-W-U sorrendben követik egymást a szobák és U kültéri „cellát” jelöl és itt 140mp-nél kevesebb időt töltött. Ekkor az első U szobát W-re cseréljük.
- 3) U-W-U sorrendben követik egymást a szobák és W szoba különleges szobának minősül, akkor ezt az esetet kihagyjuk, ez a magatartás meg van engedve.
- 4) U-W-U sorrendben követik egymást a szobák és W szobában 140mpnél kevesebb időt töltött el. Ekkor a W szobát U-ra cseréljük.

Azon eseteket, ahol ténylegesen csak egy bejegyzés született egy szobában és az azt követő bejegyzés rövid időn belül (70mp alatt) egy másik szobába mutatnak az 1) esetben szűrjük ki. Az épületek szintjén a második eset hasonló az itt taglalt 2) esethez, ez is a fal mellett ülő véletlenül kintre helyezett felhasználók pozícióját hivatott pontosítani. A 4) eset egy alosa amikor egy felhasználó gyorsan kirohan mosdóba vagy konyhába, ezt nem szabad hirtelen szobaváltásként felfogni, így a 3) esetben ezt kizártuk.

A fentebb taglalt algoritmusnak hála a felhasználók mozgása az adathalmazban élethűen van reprezentálva, hirtelen kiugrásoktól mentesen, folytonos mozgási mintázatot kapunk minden egyes felhasználóról. Átlagosan a szűrésnek köszönhetően a felhasználók

szobaváltásainak száma egy nap folyamán kb. 34%-kal (13,05), az emelet váltásaik száma 20%-kal (2,03), míg az épületváltásaik száma 21%-kal (1,95) csökkent. Eközben a meglátogatott szobák száma csak 4,4%-kal (0,56) csökkent és az átlagosan egy szobában töltött idejük 5%-kal (121 másodperc) nőtt. Ezekből a statisztikákból is látszik, hogy a szűrés hatására a felesleges mozgást sikerült kiszűrni, míg az érdemi mozgást, például az új szoba felfedezése nem csökkent nagymértékben.

3 Adatleírók létrehozása és értékeik elemzése

Szem előtt tartva a végső célt, miszerint egy olyan modellt szeretnénk létrehozni, ami magában foglalja az egyéni és csoportos mozgási magatartási mintázatokat elengedhetetlen, hogy valamilyen módon tulajdonságokkal tudjuk jellemezni és leírni az egyes felhasználók mozgását, úgy, hogy ezen tulajdonságok mentén jól összehasonlíthatóak legyenek. Ahhoz, hogy a csoportos viselkedést is modellezni tudjuk, csoportokat kell tudnunk meghatározni, elkülöníteni.

Ebben a fejezetben először a rendelkezésre álló látogatási listákból, tehát az egyes felhasználók által bejárt utakból metrikákat származtatunk. Arra törekszünk, hogy minél jobb, több és minél relevánsabb, tehát nem redundáns metrikát hozunk létre, amivel minél pontosabban le tudjuk írni egy felhasználó mozgási magatartását.

Majd ezen metrikák segítségével az olyan felhasználókat kiszűrjük, melyek kiugróan viselkednek. Ezen kiugró adatok vagy a mérési pontatlanság miatt, vagy amiatt vannak jelen, mert eleve nem emberi mozgást rögzítettek pl.: egy helyben álló laptop. Ezzel megtisztítjuk az adathalmazt az olyan értékektől melyek mind a felhasználói csoportok, mind a mozgási modell paramétereinek meghatározását negatívan befolyásolnák.

A tisztított adathalmazból kiválasztjuk azt a pár metrikát, melyek a legjobban leírják egy felhasználó mozgását, legjobban elkülöníti azt a többiekétől. Ezen pár metrika segítségével és egy gépi tanuló algoritmust alkalmazva felhasználói csoportokat különítünk el, annak érdekében, hogy a modellünk a csoport dinamikát is magában tudja foglalni.

3.1 A metrikák elkészítése

Annak érdekében, hogy a modell az egyes felhasználók mozgási mintáit tudja vizsgálni, számokkal leírhatóvá kell tenni a felhasználók mozgását. Ezért a látogatási listákból származtatva 50 különböző metrikát hozunk létre, melyekkel minél jobban leírhatók az egyes felhasználók magatartása. Olyan metrikák létrehozására törekedtünk, melyek egyedivé tudják tenni az egyes embercsoportokat. Mivel naponta minden eszköz „új felhasználóként” jelenik meg, ezért nincs rá mód, hogy az egyes emberek mozgását napokon átívelően vizsgáljuk.

Rendelkezésre állt eddig minden felhasználóról egy látogatási lista, mely tartalmazta a szobába belépés és kilépés időpontját, a szoba kódját és típusát, az emeletet és a szintet. Ezekből

az információkból hoztunk létre minden felhasználóra egy „tulajdonság vektort”, melynek az egyes elemei különbözőképp írják le az aznapi mozgását.

Fontos megjegyezni, hogy – ahogy fentebb szerepel is – eredetileg minden pozíciórekordról rendelkezésre álltak a GPS koordináták, azonban távolságokat nem ezzel számoltuk. Ez fals értéket eredményezett volna, mert például egy asztalon hagyott mobil eszköz, a pontatlanság miatt minden egyes mérés alkalmával fél-egy méterekkel arrébb lenne pozícionálva és ezen hibák összeadódtak volna. Ezen felül a modell szempontjából irreleváns, hogy egy szobán belül mennyit mozog az adott felhasználó, a modell csak a szobák közötti mozgást veszi figyelembe. Így az egyes felhasználók által megtett utakat úgy számoltuk ki, hogy az általuk bejárt szobák súlypontját kötöttük össze és ezt tekintettük egy bejárt útvonalnak.

A tulajdonság vektorokat képező metrikák a következők voltak:

- Időbeli metrikák:
 - Összesen eltöltött idő a campuson
 - Első és utolsó bejegyzés ideje (Érkezés ill. távozás)
 - Szobában/Épület körvonalában/Kültéren eltöltött idő aránya az összesen eltöltött időhöz képest
 - Kedvenc szobában eltöltött idő aránya az összesen eltöltött időhöz.
- Helyváltoztatásokat leíró metrikák:
 - Szobaváltások/Szintváltások/Épületváltások száma
 - Szobaváltások/Szintváltások/Épületváltások száma óránként
 - Összesen megmászott emeletek száma
 - Különböző meglátogatott épületek, emeletek és szobák száma
- Látogatásokat leíró metrikák:
 - Látogatások idejeinek statisztikái
 - Egyes látogatások idejének statisztikái
 - Egyes szobákban összesen eltöltött idők statisztikái
 - Egyes szobák látogatásainak statisztikái.
- Távolság alapú metrikák:
 - Összesen megtett méter
 - Átlagsebesség
 - Szobák közötti távolság statisztikái

3.2 Kiugró mozgások detektálása és szűrése

Annak érdekében, hogy modellünk alkalmas legyen az embercsoportok mozgásának pontos leírására, biztosítani kell, hogy valóban csak emberek idő- és térbeli adatait vizsgáljuk. Lévén, hogy egy egyetemen használt WiFi képes eszközökből származó adatokat vizsgálunk, nem biztosított, hogy a kapott adatok részben vagy egészben emberi mozgásból származnak. Például számítógépes laboratóriumokban használt laptopok, vagy egy kutató laborban használt teszt számítógép, nem egy személy sajátos mozgását írja le. Ezt figyelembe véve ki kell szűrni minden olyan felhasználót, aki nem emberre jellemző módon viselkedik.

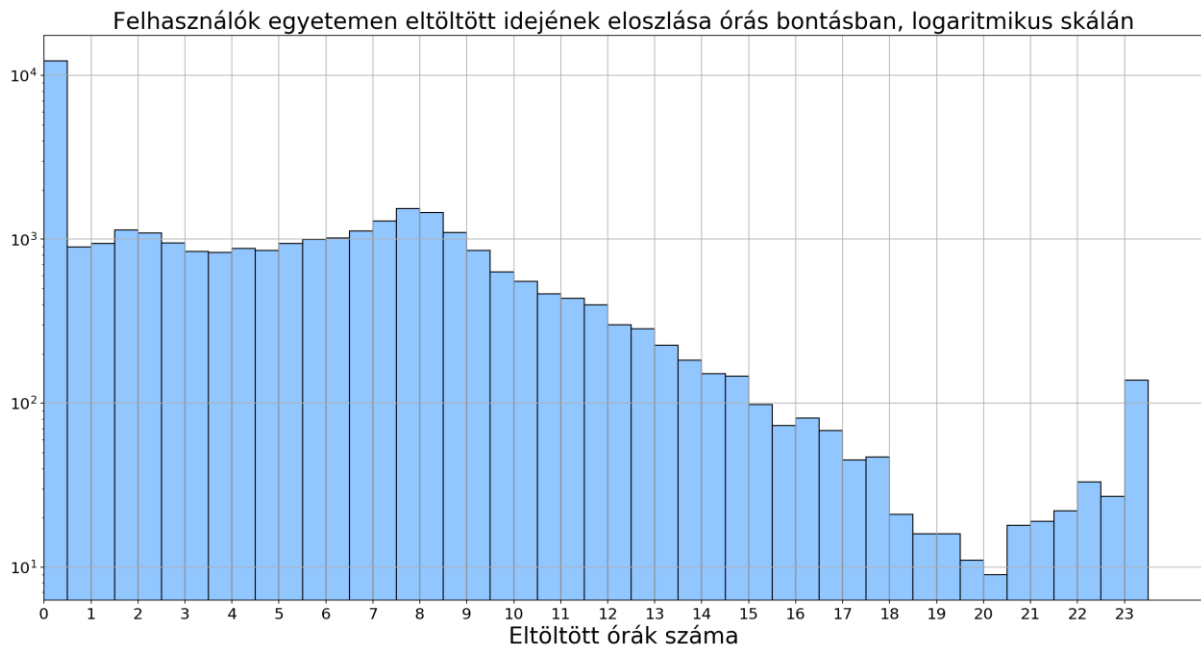
Kiugró viselkedést mutathat egy felhasználó a mozgása során valamikor, például egy hirtelen szobaváltás során, illetve az egész mozgási magatartásának összképe lehet kiugró, például, ha egy asztalon felejtett laptopról van szó. Az előbbi esetet már vizsgáltuk, amikor a látogatási listákat hoztuk létre és kiszűrtük a ping-pong effektust. Mivel létrehoztunk a felhasználókról tulajdonság vektorokat, így ezeknek a vizsgálatával az egész mozgásukról kapunk egy képet, és ez alapján tudjuk eldönteni, hogy emberi mozgásról van-e szó.

Sok algoritmus létezik amik a kiugró értékek feltárására lettek kitalálva, azonban a mi esetünkben nem bízhatjuk ezen viselkedések azonosítását egy algoritmusra, mert az nagy valószínűséggel, olyan viselkedést is kiszűrnie ami valóban kiugró, de teljesen emberi. Például egy PhD-s hallgató akinek határidőre le kell adnia egy munkát lehet, hogy bent van az egyetemen 12 órán keresztül, ez kiugrónak számíthat, de a mi esetünkben ezt nem dobhatjuk el. A szűrés során a felhasználókat 5 kategóriába soroljuk egy előre, kézzel meghatározott kritériumrendszer alapján. Egy adott felhasználó több kategóriába is tartozhat, illetve egybe sem; ők azok, akiket megtartunk, a többi felhasználót majd kidobjuk. Az alábbi kategóriákat határoztuk meg:

1. Olyan emberek, akik nagyon kevés időt töltöttek az egyetemen
 - Nem érdemes magatartást vizsgálni, mert olyan kevés idő alatt nem mozognak sokat
 - A 3.1. ábrán látható, hogy nagyon sok felhasználó legfeljebb 30 percet tölt a campuson.
2. Olyan eszközök, melyek egy helyben vannak
 - Ezen eszközök valószínűleg laborgépek, amik csak egy helyben vannak, nem mozognak.
 - A 3.2. ábrán pirossal láthatóak azon felhasználók, akiket álló eszköznek tekintünk. Ők az idejük nagy részét egyetlen egy szobában töltik, csak 1

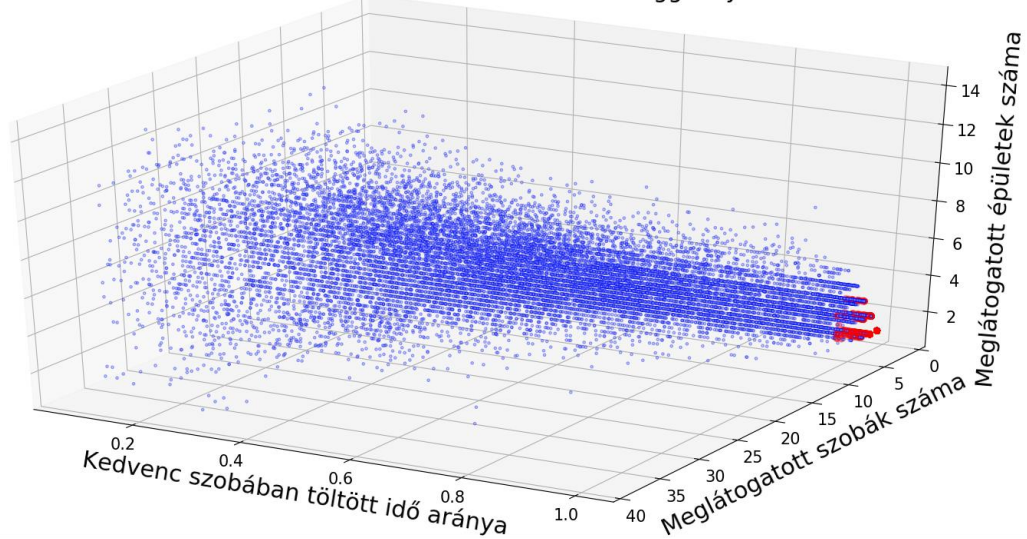
épületben járnak és legfeljebb 4 szobát látogattak. Az utóbbi feltétellel belevesszük azokat az eseteket, amikor véletlenül szomszédos szobába soroljuk az álló eszközt.

3. Olyan emberek, akik nagyon sűrűn váltanak szobát, valószínűleg mérési hibából.
 - Az egyes szobalátogatásaik átlagosan csak pár percesek, és többi felhasználóhoz képest kiugróan sokszor váltanak emeletet, épületet vagy szobát.
 - Mivel a mozgásuk ezen felhasználóknak nagyon szélsőséges a modell pontosságának növelése érdekében ezeket kidobjuk.
4. Olyan emberek, akik túl sok időt töltenek kint vagy egész idejüket valamely épületben töltik, de ott egy ismeretlen szobában tartózkodnak (épület vonzáskörzetében).
 - A 3.3. ábrán látható, hogy kiugróan sok azon felhasználók aránya akik csak kint vagy az épület vonzáskörzetében tartózkodtak.
 - Őket nem érdemes vizsgálni épületen belüli mozgás szempontjából, mert a mozgásuk során nem az a fajsúlyos és a modell elsősorban beltéri mozgást modellez.
5. Olyan emberek, akiknél a helymeghatározás megszakadt vagy félbe maradt
 - Ide tartoznak, akik hajnalig dolgoznak, majd hazamennek és visszajönnek este. Mivel minden nap új felhasználóként számolunk velük, ezért őket többször számolnánk és nem is az egész napi mozgásukat vennénk figyelembe.
 - Azok is ide kerültek, akiknek van kiugróan hosszú ideig tartó látogatásuk. Náluk feltehetően megszakadt az internet, ezért úgy vettük, mintha az utolsó szobában töltöttek volna nagyon sok időt, ez pedig torz képet ad a mozgásukról.
 - A 3.4. ábrán látszik, hogy hajnalban (2 óra körül, az UTC miatt) ugrásszerűen “érkezik” az egyetem területére sok felhasználó. Ők valójában eddig is ott voltak, csak az új nap számítása miatt, úgy érzékeljük, hogy akkor érkeztek. Mivel nekik nem a teljes magatartásuk van rögzítve így a vizsgálatukkal torzítanánk a modellt.

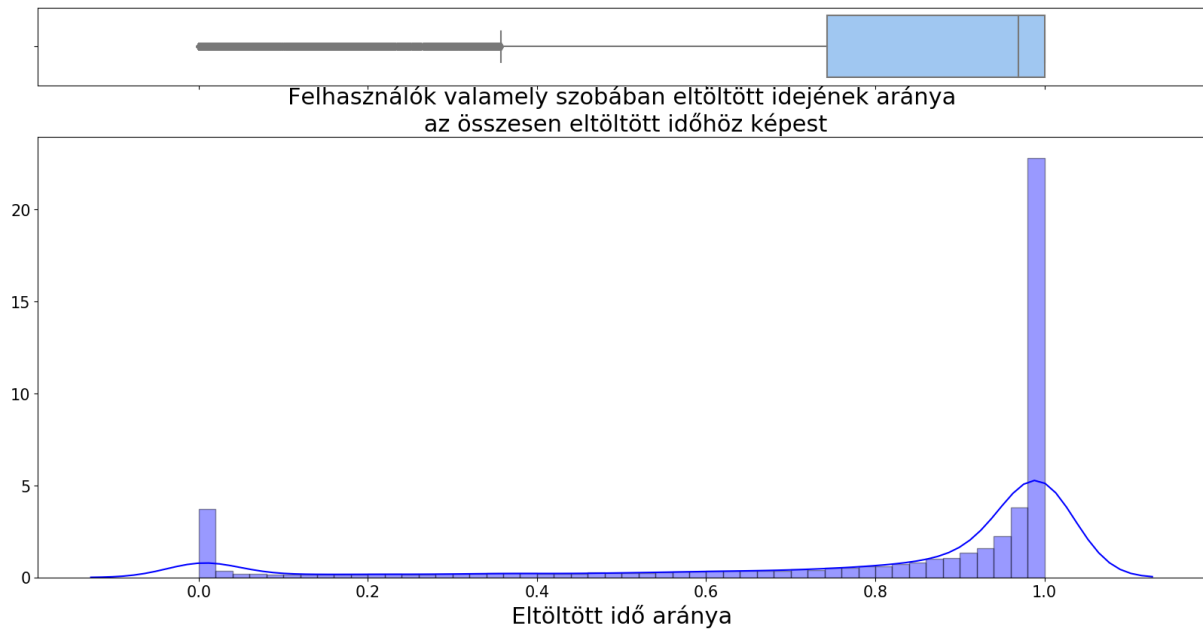


3.1. ábra: A felhasználók összesen a campuson töltött idejének eloszlása fél órás bontásban, logaritmusos skálán. Nagyságrendileg kiugróan sokan vannak azok, akik 30 percnél kevesebb időt töltenek el.

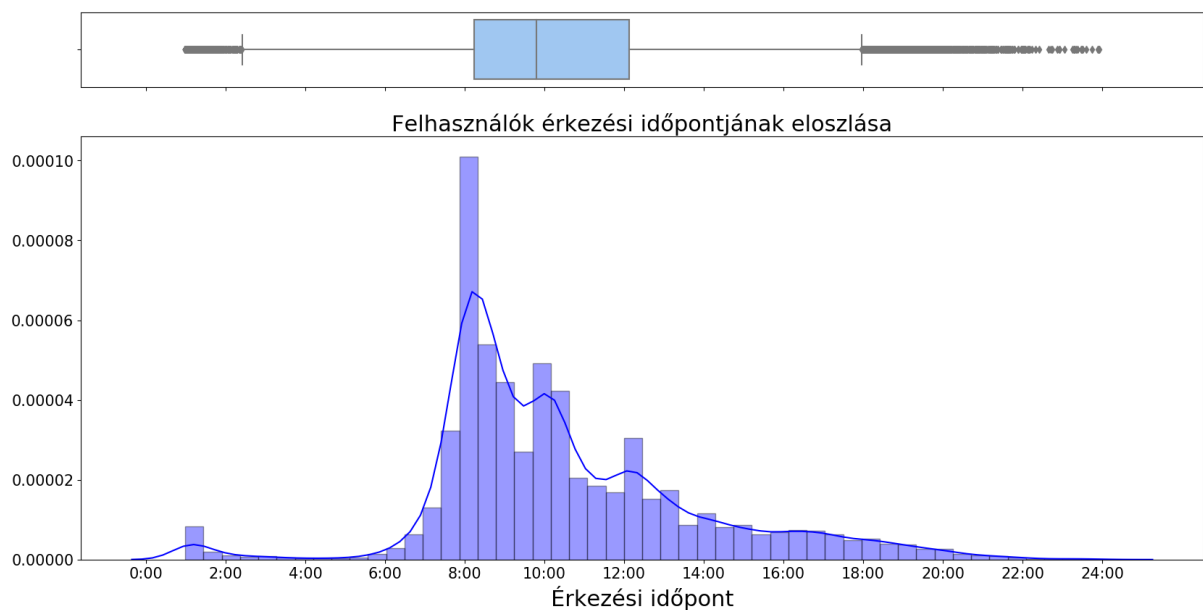
Az egy helyben állók kirajzolása, a meglátogatott épületek és szobák, valamint a kedvenc szobában töltött idő függvényében



3.2. ábra: A felhasználók kedvenc szobában töltött idő aránya, a meglátogatott szobák és a meglátogatott épületek által feszített térben ábrázolva. Pirossal vannak jelölve az álló eszközök.



3.3. ábra: A felhasználók szobákban eltöltött idejének az összesen eltöltött időhöz viszonyított arányának az eloszlása hisztogramon és felette doboz diagramon ábrázolva.



3.4. ábra: A felhasználók egyetemre érkezési idejének időpontjának eloszlása.

Ezzel sikerült megkülönböztetni azokat a felhasználókat, illetve eszközöket, akik valamilyen oknál fogva nem úgy viselkednek, mint egy átlagos ember, vagy csak az adathalmazban nem szerepelnek a viselkedésüknek leírásához szükséges adatok. Így annak fényében, hogy célunk az embereket a viselkedésük alapján csoportosítani, ezen felhasználókat ki kell szűrniük. Ezáltal az emberek mozgási mintái közötti finomabb eltéréseket is fel tudjuk fedezni.

Mivel a kiugró, szélsőséges viselkedésű felhasználók nincsenek az adathalmazban, az egyes tulajdonságoknak az értéktartománya is lecsökken, így a relatív távolság kisebb lesz az egyes felhasználók tulajdonság vektorai között, ami pontosabb csoportosítást eredményez. Ez abból fakad, hogy a későbbiekben használt K-means klaszterező algoritmus érzékeny a zajra. Lévén távolság alapú összehasonlítást használ, azaz azon rekordok, melyek tulajdonság vektorjainak komponensei abszolút értékben közel vannak egymáshoz, valószínűleg egy csoportba tartoznak. Azonban ez a távolság relatív távolság, tehát, ha van pár rekord, ami nagyon kiugrik, akkor a többi rekord közötti távolság ehhez a kiugró rekordhoz képest relatíve lecsökken, ami félrecsoportosításhoz vezethet.

3.3 A legfontosabb adatbeli tulajdonságok kiválasztása

Célunk, hogy az adathalmazban felhasználók csoportjait különböztessünk meg mozgási mintázatuk alapján. Ezt klaszterezéssel tudjuk elérni, ami egy nem felügyelt gépi tanuló módszer. Annak érdekében, hogy minél pontosabb eredményt - azaz minél jobban elkülönülő csoportokat, klasztereket - kapjunk, ki kell választanunk az adathalmaz 50 darab metrikája közül azt a párat, ami a legtöbb információt tartalmazza, a legjobban leírja az adathalmazt és megtartja azt a struktúrát ami az 50 dimenziós térben volt. Ez a folyamat a „Fontos jellemzők kiválasztása”, melyet a szakirodalom Feature Selectionnek hív, ez egy nagyon fontos részét képezi a gépi tanuló modellek betanításának. Ez azért is kardinális folyamat, mert alapjaiban határozza meg a tanuló algoritmus pontosságát. Alapvetően az adathalmaz tartalmazhat olyan metrikákat, amik vagy redundánsak, vagy pedig irrelevánsak, ezért eltávolíthatók az adathalmazból anélkül, hogy nagy információvesztést okoznának.³

Nem felügyelt tanulási algoritmusok esetében az adathalmazon betanított modell az adatokban rejlő összefüggéseket, struktúrákat és mintázatokat hivatott felfedezni. Nem áll rendelkezésre célváltozó, azaz egy olyan attribútum, ami a struktúrát leírná, például egy osztály címke, amit mi, mint eredmény szeretnénk látni a modell kiértékelésekor. Más szóval nem áll rendelkezésre „alap igazság”, így a modell eredményét csak pár mérőszámmal, illetve szubjektíven, a mi megítélésünk alapján – természetesen figyelembe véve a célt és a problémát – tudjuk kiértékelni. Ebből következik, hogy globális optimum nem létezik, csak általunk megfelelőnek fémjelzett megoldás.

³ Forrás: https://en.wikipedia.org/wiki/Feature_selection

A Feature Selection egyebek mellett azért is lényeges, mert azzal, hogy az adathalmazt leíró metrikák számát, azaz dimenzióját csökkentjük egyszerűbbé, értelmezhetőbbé válik majd számunkra a kapott eredmény, így könnyebben tudunk szubjektív ítéletet mondani a modellről. Másrészt a kevesebb dimenzióval elkerüljük az úgynevezett „Dimenzionalitás átkát” [19], ami többek között azt jelenti, hogy magas dimenzióban a legtöbb pont távolsága hasonló (an magas), így nincs különösebb értelme közelségről beszélni. Kisebb dimenzió kevesebb adatot jelent, így az algoritmusok tanítási ideje és memória igénye is kisebb lesz, ami egy ekkora adathalmaznál nem elhanyagolható szempont.

Globális optimum hiányában nem volt egy konkrét algoritmus vagy módszertan, amit alkalmazva, mint Feature Selection eljárást megkaptuk volna a legjobb metrikák halmazát. Erre a problémára nagyon sok publikáció született olyan algoritmusokat bemutatva, melyek különböző statisztikai módszerekkel próbálják megtalálni a legmegfelelőbb részhalmazát a metrikák halmazának. Xiaofei és társai által kifejlesztett eljárás [7], a Laplacian Score-t használja fel erre a célra, mely azon alapszik, hogy az azonos osztályba tartozó adatpontok közel állnak egymáshoz, így az attribútumokat lokalitás megőrző tulajdonságuk alapján értékelik. Az MCFS eljárást Deng és társai [4] közölték, ez L1-regularizáció segítségével azon attribútumokat választja ki, amelyek megtartják az adathalmaz multi-klaszter struktúráját. Illetve a Főkomponens-analízis [14] is használható erre a célra. Az eljárás az eredeti D dimenziós adathalmazt egy P ($\leq D$) dimenziós térbe transzformálja, ahol az egyes dimenziók a fő komponensek, úgy, hogy a dimenzió redukció során az adathalmazban jelenlévő varianciát - ezzel arányosan az információt - megtartja. Azonosítva azokat az eredeti attribútumokat, melyek a legjobban meghatározzák a fő komponenseket, megkapjuk az attribútum halmaznak azon részhalmazát, mely a legjobban megtartja az eredeti adathalmaz struktúráját, a legtöbb információt tartalmazza.

Természetesen az, hogy melyik algoritmus ad jobb attribútumhalmazt akkor válik világossá, ha ugyanazt a klaszterező modellt, ugyanazokkal a paraméterekkel az adathalmazon lefuttatjuk úgy, hogy mindig csak a megfelelő attribútumhalmazt választjuk ki. A kapott eredményeket a szubjektív grafikus ábrázolás mellett két mérőszámmal vizsgáljuk, melyek a klaszterezés pontosságát számszerűsítik: a *Silhouette* [17] és *Calinski-Harabasz* [5] érték.

A k darab klaszterből álló adathalmaz *Silhouette* értéke a következőképp számolható ki. Legyen egy tetszőleges i adatpontra $a(i)$ az i pont és a vele azonos klaszterben lévő adatpontok átlagos távolsága. Ez tulajdonképpen annak a mértéke, hogy mennyire jól van az i pont az adott

klaszterhez rendelve (kisebb érték jobb hozzárendelést jelent). Legyen továbbá $b(i)$ az i pont és a legközelebbi klaszter (amelynek i nem tagja) pontjainak átlagos távolsága. A legközelebbi klaszter itt azt a klasztert jelenti ahol i és a klaszter pontjainak távolságának átlaga a legkisebb. Így egy i pont Silhouette értékét a következőképp definiálhatjuk:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Az egész adathalmaz Silhouette értéke megegyezik az összes i pont $s(i)$ értékének átlagával. Ez a szám -1 és 1 közötti értékeket vehet fel, ahol az 1-hez közeli érték mutatja, hogy az adathalmaz jól lett klaszterezve.

Minél nagyobb a Calinski-Harabasz érték annál sűrűbbek a klaszterek és jobban el vannak szeparálva egymástól, ami a klaszterezés alap koncepciójával van összhangban. Az egész adathalmazra az értéket az alábbi módszerrel lehet kiszámolni:

$$CH = \frac{SSB_k}{SSW_k} * \frac{N - k}{k - 1}, \text{ ahol}$$

$$SSW_k = \sum_{c=1}^k \sum_{x \in S_c} d(x_i, \mu_c)^2$$

$$SSB_k = \sum_{c=1}^k n_c d(\mu_c, \hat{\mu})^2$$

Ahol k a klaszterek száma, N az adatpontok száma, μ_c a c klaszter középpontja, $\hat{\mu}$ az egész adathalmaz középpontja, S_c a c klaszterbe tartozó pontok halmaza, n_c pedig a c klaszter elemeinek számát, d pedig a (Euklideszi) távolságot jelenti.

Az SSB értéke az összes klaszter középpontjának varianciáját méri az egész adathalmaz középpontjától, tehát nagyobb SSB azt mutatja, hogy a klaszterek szét vannak szórva és nem egymáshoz közel helyezkednek el. Az SSW klaszteren belüli varianciát méri, minél kisebb ez a szám, annál sűrűbben helyezkednek el az adott klaszterben a pontok. Így a CH érték az SSB és az SSW aránya miatt minél nagyobb annál jobb klaszterezésre utal.

A fentiek alapján az Laplacian Scoret használva kaptuk a legjobb eredményt. Mely alapján azon attribútumok, amik a legjobban leírják az adathalmazt a következők:

- **maxVisitInOneRoom:** Maximálisan hányszor járt a felhasználó egy szobában.

- **roomChanges:** Szobaváltások száma
- **avgTimeSpentVisit:** Egy látogatás alkalmával átlagosan mennyi időt tölt a felhasználó egy szobában.
- **maxTimeSpentVisit:** Maximálisan mennyi időt töltött egyhuzamban a felhasználó egy szobában.
- **stdTimeSpentVisit:** A szoba látogatási idők szórása.
- **roomsVisited:** Összesen mennyi különböző szobát látogatott meg a felhasználó.
- **stdTimeSpent:** Az egyes szobákban összesen eltöltött idők szórása.

Ezen attribútumok megfelelnek az elvárásainknak is és a valóságnak is, hiszen egy ember mozgása legjobban a szobaváltások gyakoriságával és az egyes látogatások milyenségével írható le. A legjobban az emberek mozgékonyága a meglátogatott különböző szobák és a szobaváltásai alapján írható le. Az, hogy egy felhasználó hányszor tér vissza egy szobába, illetve az egyes látogatásai milyen hosszúak általában és a kedvenc szobájában töltött idő hossza valóban jó leírója a mozgási magatartásának.

3.4 Felhasználói csoportok elkülönítése klaszterezéssel

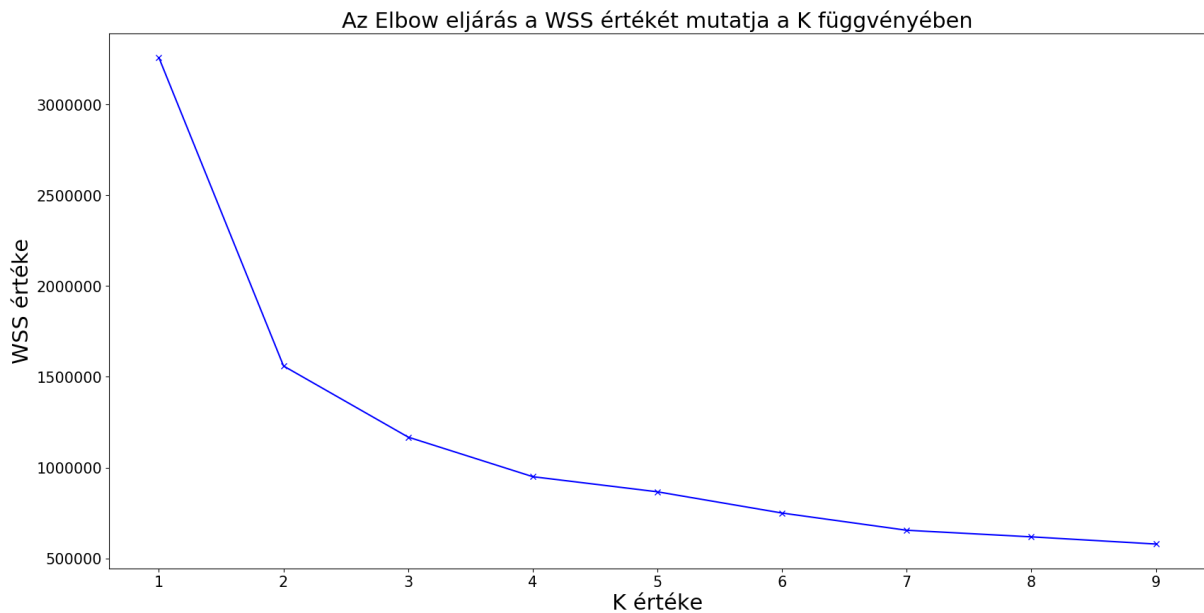
Azzal, hogy kiválasztottuk az 50 darab metrika közül azt a 7 darabot ami a legjobban leírja a felhasználók mozgási mintázatát megalapoztuk, hogy az adathalmazban fellelhető felhasználói csoportokat minél jobban el tudjuk különíteni. A csoportok szegmentációjához a K-means [21] klaszterező eljárást alkalmazzuk. Az algoritmus az adathalmazban megtalálja a K darab olyan klasztert, melyeknek elemei közötti távolság kicsi, miközben a többi klaszter elemei távol vannak tőlük. A K értékét nekünk kell megadni, az algoritmus nem találja ki magától. A K meghatározásához az úgynevezett Elbow eljárást [20] használtuk, mely a WSS-t (within cluster sum of square), mint a K-tól függő függvényt minimalizál.

$$WSS = \sum_{c=1}^K \sum_{x \in S_c} d(x_i, \mu_c)^2$$

Ahol k , a klaszterek számát jelenti, S_i pedig az i . klaszterbe tartozó pontokat, μ_i pedig a klaszterközéppontot jelöli. Ez tulajdonképpen megegyezik az előző fejezetbeli SSW-vel.

Az algoritmus lépései:

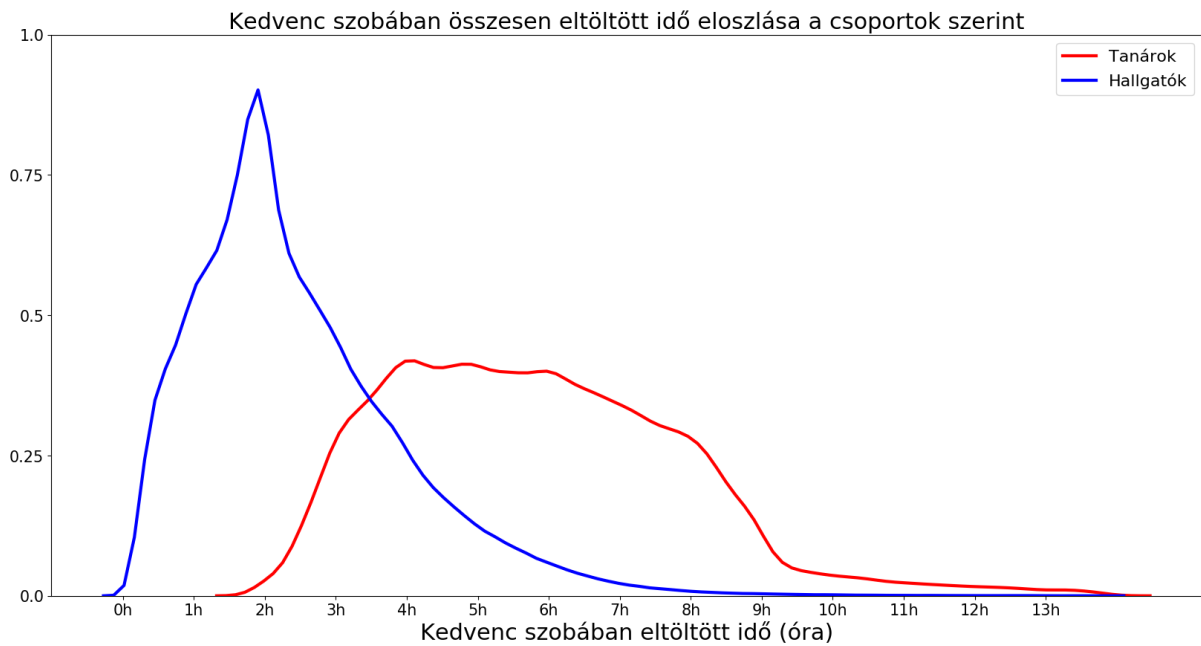
- 1) 1-től 10-ig minden K értékre lefuttatjuk a K-means klaszterezést.
- 2) Minden K érték esetében kiszámoljuk a WSS értékét.
- 3) K függvényében ábrázoljuk a WSS értékeket, ez a 3.5. ábrán látható.
- 4) A görbe könyökpontja tekinthető általában a megfelelő K értéknek



3.5. ábra: Elbow eljárás során kiszámolt WSS értékek a K függvényében.

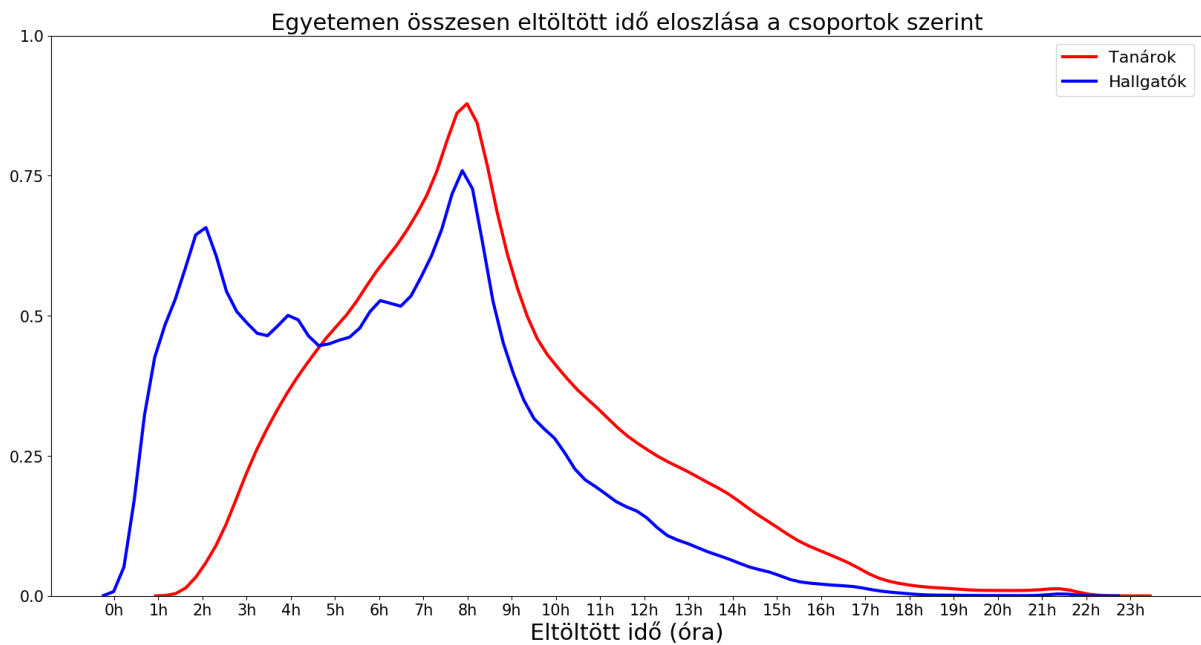
Arra jutottunk, hogy a $K=2$ a legjobb megoldás számunkra. Az algoritmust lefuttatva az adathalmaz kiválasztott részhalmazán, $K=2$ paraméterrel azt találtuk, hogy a két embercsoport a tanárok és a hallgatók csoportja. Ezt a következtetést, a klaszterek számosságából és az egyes metrikák empirikus eloszlásából vontuk le. A klaszterezést az egész adathalmazon elvégezve, ami 651 816 felhasználót jelent összesen (napok között meg vannak különböztetve a felhasználók), a hallgatók csoportjába 526 578 darab felhasználó tartozott, míg a tanárok csoportjába 125 238. Látható, hogy a 4,2-szer annyi diák van, mint tanár, ami nagyjából megfelel a valóságnak.

A 3.6. ábra mutatja, hogy a felhasználó a kedvenc szobájában mennyi időt (órában) töltött összesen. Itt a diákok esetében kb. 2 óránál van egy csúcs, ami egybevág azzal a magatartással, hogy a diákoknak nincs igazán kitüntetett termük, egy előadás hosszáig egy helyben maradnak és utána jellemzően nem térnek vissza oda. Ezzel szemben a tanárok 4 és 8 óra közötti időt töltenek a szobájukban, ami egy munkanapnyit jelent.



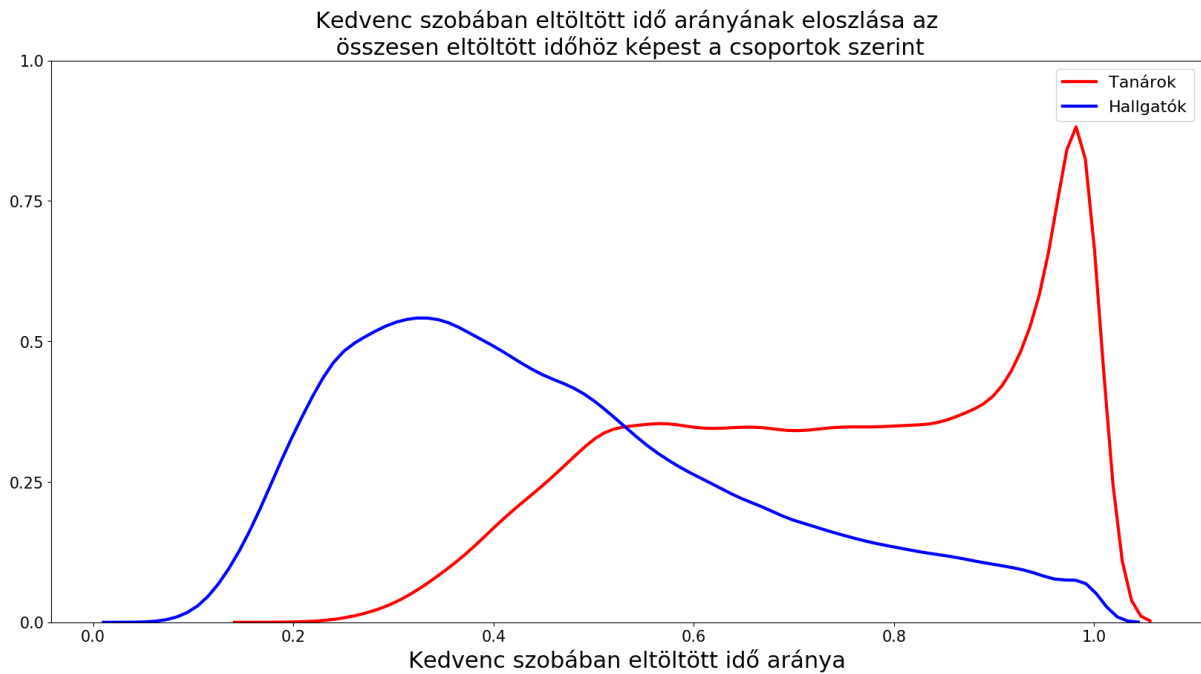
3.6. ábra: A kedvenc szobában összesen eltöltött idő eloszlása a felhasználói csoportok között.

Megfigyelhető a 3.7. ábrán, ami a felhasználók összesen eltöltött idejét mutatja az egyetemen, hogy a tanároknál a 8 órás munkaidő a jellemző, a hallgatók 2-től 8 órát töltenek főként az egyetemen. Figyeljük meg a 2 órás csúcsokat a hallgatók esetében.



3.7. ábra: Egyetemen összesen eltöltött idő eloszlása a felhasználói csoportok között.

Elvállik a két csoport magatartása a 3.8. ábrán látható módon, miszerint a tanárok az idejük nagy részét egy bizonyos szobában töltik. Ezzel szemben a diákok esetében, azon szoba ahol az idejük legnagyobb részét töltötték az egész napjuknak csak töredékét teszi ki.



3.8. ábra: Felhasználói csoportok között annak eloszlása, hogy a felhasználók az idejük hányad részét töltik a kedvenc szobájukba.

A 3.9. ábra mutatja, hogy egy óra alatt hányszor vált szobát egy felhasználó átlagosan. A tanárok esetében jellemzően ez az érték kisebb, mint a diákoknál, ahol az hosszú farkok arra utal, hogy a diákok mozgékonyabbak, sűrűbben váltanak szobát.



3.9. ábra: Óránkénti szobaváltások számának eloszlása a felhasználói csoportok között.

Sikerült eredményesen felhasználói csoportokat meghatározni gépi tanuló algoritmusok használatával. Ezen csoportok a valóságban is léteznek, a klaszterezés nem csak egy virtuális, csak ebben az adathalmazban jelenlévő sajátos mintázatot talált, hanem élethű, ebben az adott környezetben tipikusan jelen lévő felhasználókat sikerült azonosítani. Ezzel megalapoztuk a mozgási modell azon tulajdonságát, hogy a csoport dinamikát is reprezentálja, mivel az egyes csoportokra különböző mozgási paramétereket tudunk meghatározni.

4 Beltéri mozgási modell építése

Eddigiekben az adathalmazt tártuk fel és alakítottuk át úgy, hogy alkalmas legyen egyének mozgásának leírására és csoportok elkülönítésére. Most egy olyan általános modellt építünk, mellyel egyenként lehet felhasználókat generálni az adott külteret és belteret magában foglaló környezetbe, kihasználva az épületek felépítésének hármas hierarchiáját. A paramétereket adatvezérelten határoztuk meg figyelembe véve más tanulmányok megállapításait az egyes törvényszerűségek kapcsán. A modell folytonosnak tekinti az időt, és attraktorokat használ az egyes szobák vonzerejének modellezéséhez. Ezen attraktorok két komponensből állnak, melyek a látogatási népszerűséget és az ott töltött időt befolyásolják. A paramétereket megfigyelésekkel, nyomonkövetésből származó valós adatokkal vagy akár becslések alapján is meg lehet adni. Ilyen bemeneti paraméterek a következők:

- A. Környezet felépítése (szobák listája)
 - a. Szobák befogadóképessége és telítettsége
 - b. Szobák funkcionalitása és átlagos látogatások időtartama
- B. Felhasználói csoportok és rájuk jellemző tulajdonságok
 - a. Érkezési és távozási idejük eloszlása
 - b. Mozgékonyosságuk (milyen gyakran váltanak szobát és mennyi helyen járnak)

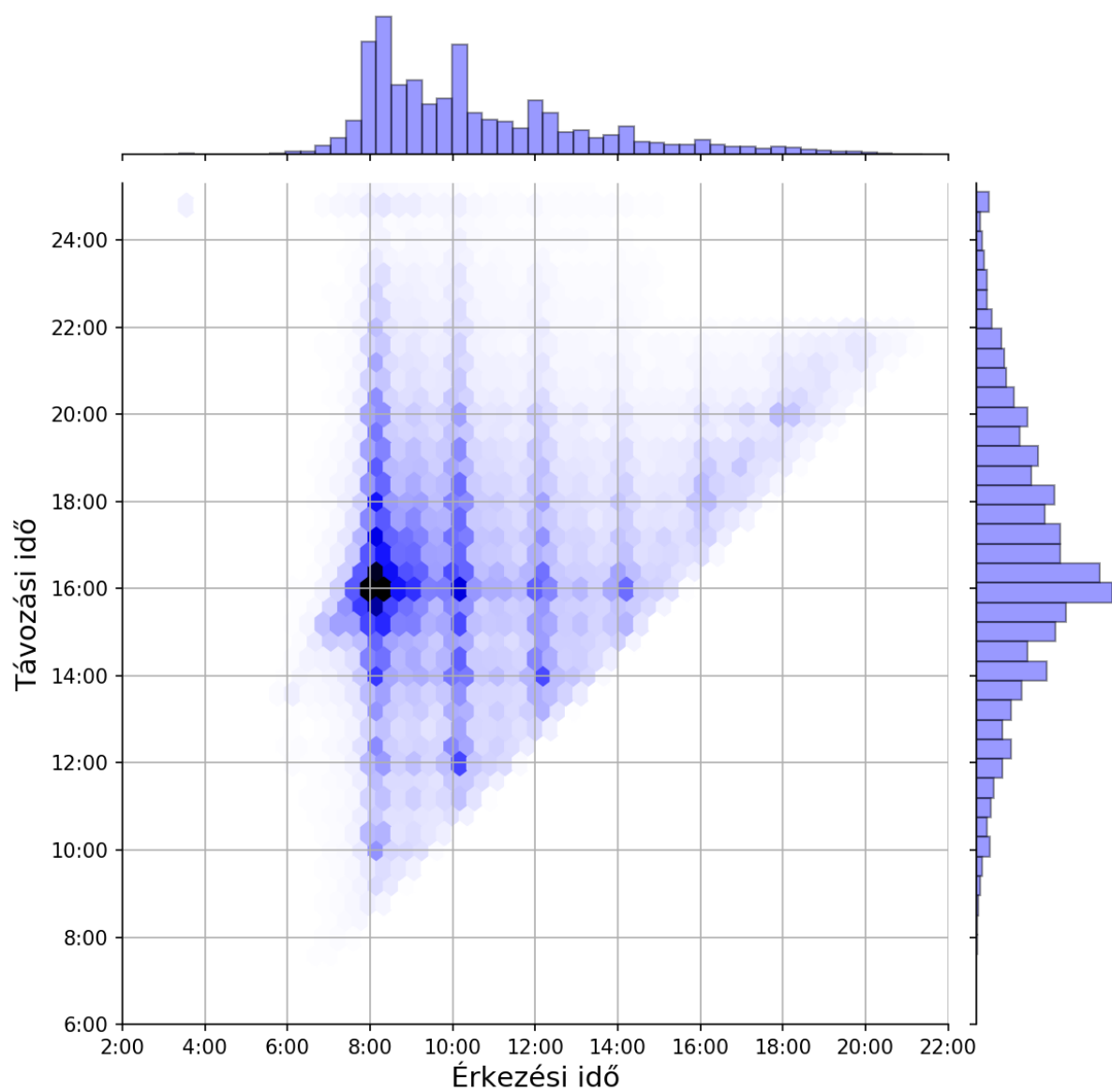
A fenti bemeneti paramétereket könnyű becsülni még tervezési időben is. Például egy szoba telítettségét az órarendek, időbeosztások segítségével könnyen meg lehet adni. Ugyanígy a szobák funkcióját előre lehet tudni, és mérésekből, tapasztalatból és időbeosztásból meg lehet határozni, hogy átlagosan mennyi időt töltenek az egyes helyiségekben. A felhasználói csoportokat teljesen szabadkezűleg testre lehet szabni. A napi útvonalukat és mozgékonyosságukat az egyes szobák attraktorainak változtatásával lehet megszabni.

A paraméterek megadásával a modell kívánt mennyiségű felhasználót generál és megadja azoknak a napi bejárt útvonalát a következőképp. Először felveszi a felhasználót a rendszerbe, azáltal, hogy meghatározza, mikor és hol kezdi a napját, illetve ebben a kezdőszobában eltöltött T időt. Ezzel egyidejűleg előre definiálja a távozás idejét is. Majd amikor lejár a T idő kiválasztja, hogy melyik szobába mozog át, és az ott eltöltendő T időt kiszámolja. Ezt a nap folyamán addig csinálja, míg el nem éri a távozás idejét, amikor is a felhasználó befejezi a napi rutinját és elhagyja a rendszert. Az egyének, mivel valószínűségi

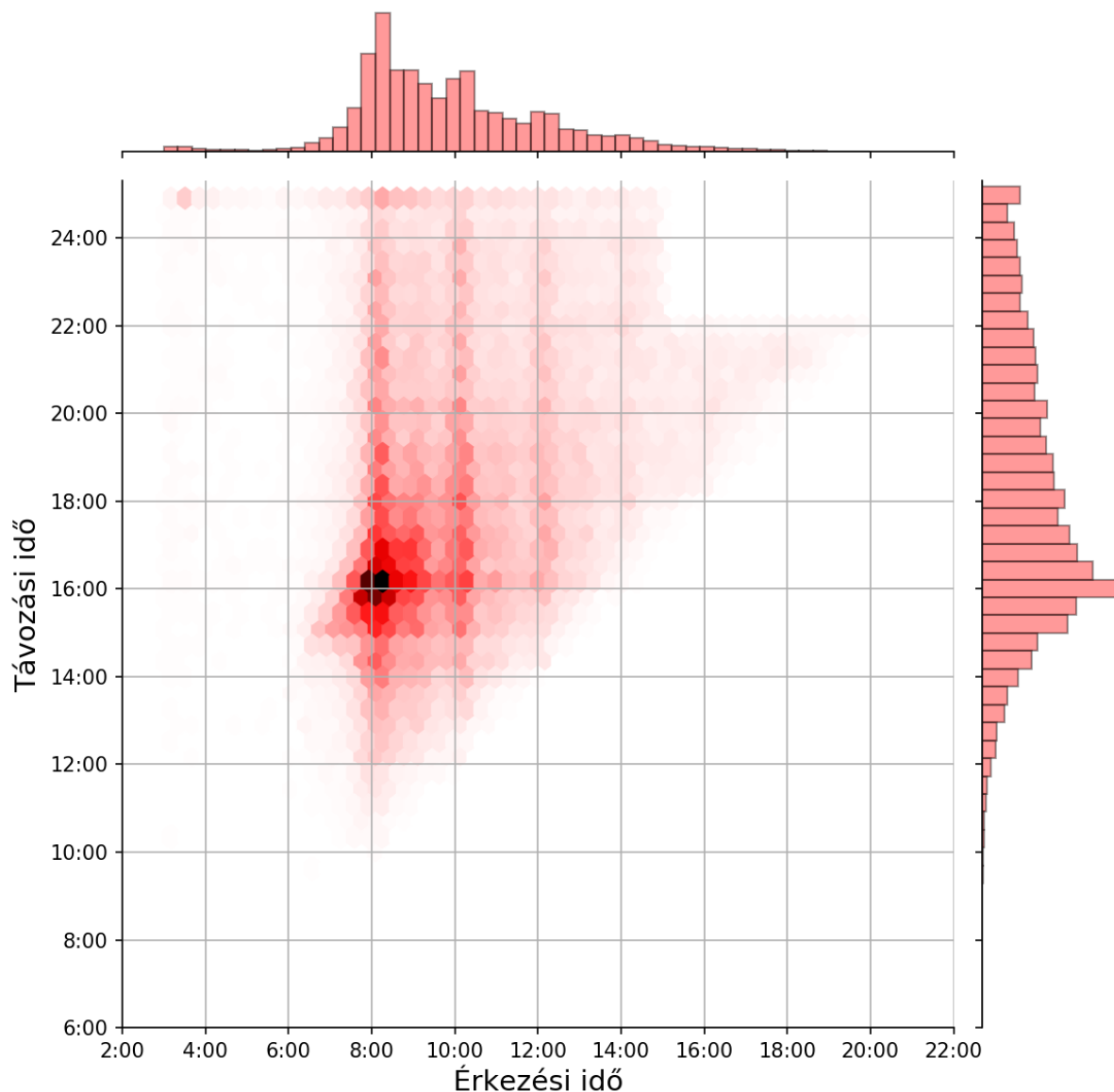
módon lettek létrehozva nem biztos, hogy mind úgy viselkednek ahogy a csoportjuk azt megköveteli. Azonban, ha sokaságként tekintünk rájuk kirajzolódik egy jellegzetesség, egy mozgási mintázat. Így egy tömegként vizsgálva őket, gyakorlati problémák megoldására lehet használni a modellt, például terhelési és kiürítési terveket lehet optimalizálni, illetve bevásárlóközpontokban reklámkampányt indítani.

4.1 Inicializálás

A generálandó felhasználók a modell szerint egy rögzített érkezési és távozási idővel kerülnek feljegyzésre a rendszerben. Adottnak tekintve a felhasználó csoportját, ezt a két időpontot az adott csoport érkezési és távozási időpontjainak együttes empirikus eloszlásából számoljuk ki. Ez a módszer egyidejűleg a felhasználók sokaságát vizsgálva a csoportokra jellemző összesen eltöltött idő eloszlását is visszaadja. A 4.1. és 4.2. ábrán látható a két csoport érkezési és távozási idejének együttes empirikus eloszlása. Figyeljük meg, hogy a hallgatók sokszor érkezési és távozási ideje jobban szór, sokszor kevesebb időt töltenek az egyetemen, ami az alábbi ábrákból is jól kivehető. A meghatározott érkezési idő függvényében a későbbiekben bemutatandó attraktorok alapján rendelünk egy szobát a felhasználóhoz, ahol a napját kezdi.



4.1. ábra: Hallgatók érkezési és távozási idejének együttes eloszlása. Sötétebb hatszög jelzi a valószínűbb kombinációt. Figyeljük meg a 2 órás sávokat az érkezési időnél.



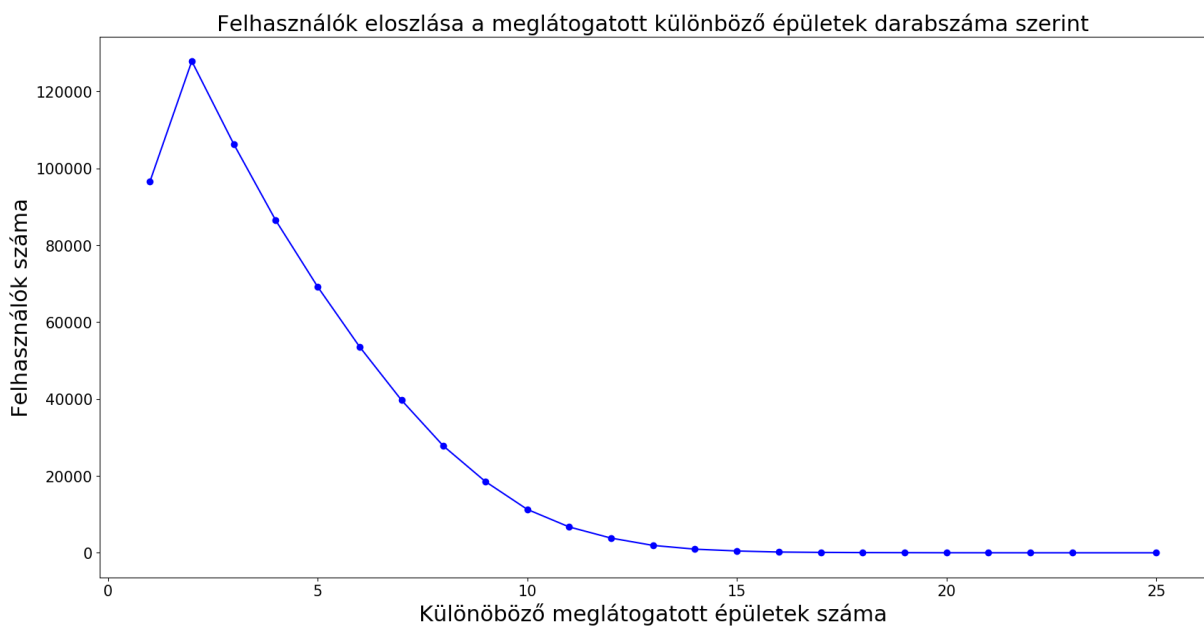
4.2. ábra: Tanárok érkezési és távozási idejének együttes eloszlása. Sötétebb hatszög jelzi a valószínűbb kombinációt. Figyeljük meg a 2 órás sávokat az érkezési időnél.

4.2 Mozgási ciklus

A mobilitási modell fő ciklusa a következő: kiválasztjuk a felhasználók következő meglátogatott szobáját, majd kiszámoljuk a várható T látogatási időt. A felhasználó a kiválasztott szobába megy, és ott eltölt T időt. Ezeket a lépéseket újra és újra megismételjük egy egész napra, amíg el nem érjük az előre meghatározott távozási időpontot, amikor is a felhasználó az adott szobából kilép és elhagyja a rendszert.

4.2.1 Következő szoba kiválasztása

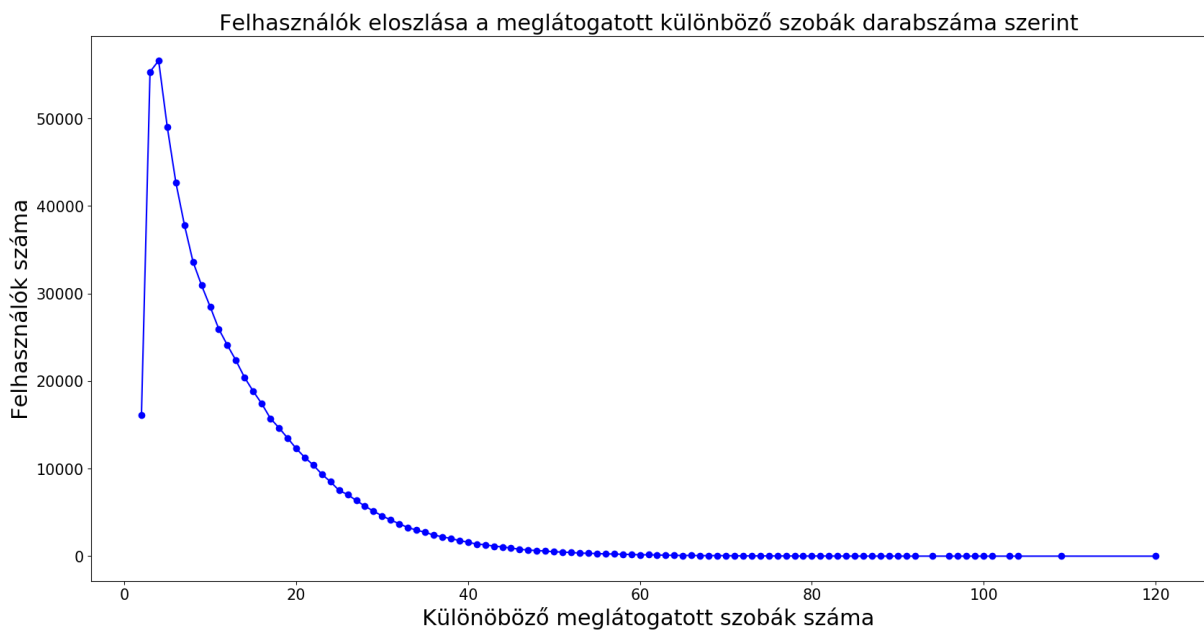
A fő építőeleme a mobilitási modellnek a következő szoba kiválasztása, melybe a felhasználó mozogni fog. Számos megfigyelés található a kapcsolódó tanulmányokban, amiket figyelembe kell venni. Elsőként, az emberek csak néhány szobát látogatnak meg, és a legtöbb idejüket ezekben töltik [6]. Ezt a megállapítást az adathalmazunk is megerősíti. A 4.3. és 4.4. ábra sorrendben mutatja a felhasználók eloszlását a meglátogatott különböző helyszínek darabszámának tekintetében mind épület és emelet szinten. Szoba szinten az adatok azt mutatják, hogy a felhasználók által meglátogatott szobák száma a 4.5. ábrán látható empirikus eloszlást követi, melynek a várható értéke 13 a mediánja pedig 10. Épület szinten a várható érték 4, míg a medián 3. Emelet szinten 6 a várható érték és 5 a medián. Modellalkotási szándékunk kifejezetten a mozgó emberekre irányul, ezért az egy helyben megfigyelt eszközöket ($x=1$) kiszűrtük az adathalmazból. Azok jelenlétével a tapasztalati eloszlások hiperbolikus alakot mutatna.



4.3. ábra: Felhasználók eloszlása a meglátogatott különböző épületek száma szerint.



4.4. ábra: Felhasználók eloszlása a meglátogatott különböző emeletek száma szerint.



4.5. ábra: Felhasználók eloszlása a meglátogatott különböző szobák száma szerint.

Másodsorban, az emberek a rövidebb útvonalakat preferálják a hosszabbakkal szemben: ez jelen esetben azt jelenti, hogy többet mozognak emeleten belül, mint emeletek között, illetve épületen belül, mint épületek között. Ezt támasztja alá ugyanezen a campuson végzett tanulmány eredménye is [16].

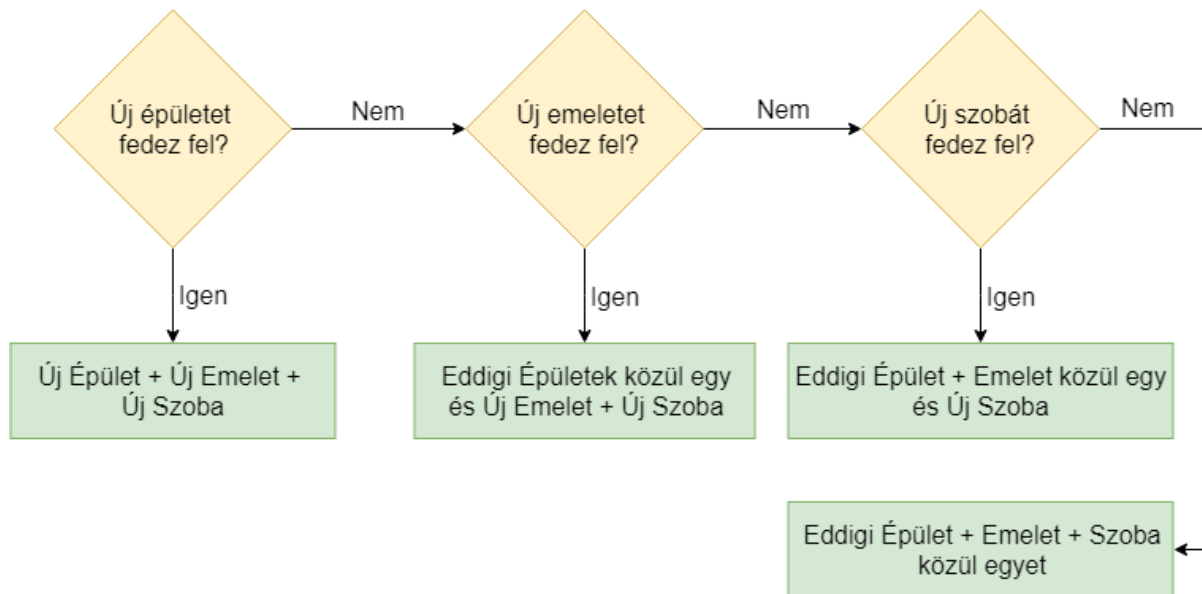
Harmadsorban, egy helyszín – legyen az szoba, emelet vagy épület – vonzerejét attraktorok segítségével modellezzük. Minél erősebb az attraktor annál valószínűbb, hogy a

felhasználó meglátogatja az adott helyszínt. Az attraktorok használata rugalmassá teszi a modellt, és jól illeszkedik egy ilyen térben korlátozott környezethez.

Miután egy adott felhasználó eltöltött T időt egy szobában, egy másik szobába megy át. A következő szobájának kiválasztási folyamata három fő lépésből áll, melyet az 4.6. ábra is mutat:

1. Kiszámoljuk, hogy a váltás során egy új (eddig nem látogatott) épületbe megy-e át.
 - a. Amennyiben **igen**: a következő meglátogatott épületet épület szintű attraktorok határozzák meg. Ezen épületen belüli emeletet és szobát ugyancsak attraktorok adják meg.
2. Ha **nem**, kiszámoljuk, hogy a váltás során új emeletet látogat-e meg.
 - a. Amennyiben **igen**: Az eddig meglátogatott épületek közül az attraktorok alapján megadjuk, hogy melyik épületbe megy. Ezen épületen belül egy eddig meg nem látogatott emeletet és azon belül egy szobát határozunk meg az emelet szintű attraktorok alapján.
3. Ha **nem**, kiszámoljuk, hogy a szoba váltás során új szobát látogat-e meg.
 - a. Amennyiben **igen**: Az eddig meglátogatott épületek és emeletek közül meghatározzuk, hogy hova megy. Majd az emeleten belül egy eddig meg nem látogatott szobát határozunk meg szoba szintű attraktorok alapján.
 - b. Amennyiben **nem**: A következő meglátogatott szoba az eddig meglátogatottak (melyek lehetnek akár külön emeleten és épületben is) közül kerül ki, melyet szoba szintű attraktorok határoznak meg.

Tekintettel a fentebb vázolt folyamatra a modell a működése során, egy listában tárolja az eddig bejárt helyszíneket, hogy megfelelően ki tudja választani a következő helyszínt (pl.: ha a felhasználó már meglátogatott két épületet és a következő lépése a modell szerint egy új épület meglátogatása, akkor a lépésből ki kell zárni az eddig meglátogatott 2 épületet).



4.6. ábra: A következő szoba kiválasztásának folyamatábrája.

4.2.2 Állapot átmenetek

Korábbi tanulmányok arra a megállapításra jutottak, hogy az emberek gyakran újra meglátogatnak olyan helyszíneket, ahol már korábban jártak [3]. Pontosabban, annak a tapasztalati valószínűsége, hogy a felhasználó egy új, eddig még meg nem látogatott helyszínrre megy a következőképp számolható ki az adathalmazból. Megvizsgáljuk, hogy az egymást követő szobaváltások során hanyadik szobaváltással kerül a felhasználó olyan szobába, amiben eddig még nem járt, feltéve, hogy eddig D_n különböző szobában járt. Ha ezen látogatások sorozatát vesszük, egy geometriai eloszlást kapunk. A geometriai eloszlásban a sikertelen eseményeket itt az olyan szobaváltás jelenti, mely nem új szobába vezet. Míg a sikeres eseményt az a szobaváltás jelenti, amikor új szobába megy a felhasználó. Így ahhoz, hogy megkapjuk annak a valószínűségét, hogy T idő lejárta után a felhasználó új szobát fog-e felfedezni - azaz bekövetkezik a sikeres esemény - feltéve, hogy eddig D_n darab különböző szobában volt, ki kell számolnunk a (geometriai eloszlás) p paraméter értékét minden D_n -re.

$$P(X = n) = pq^n \quad (n = 0,1,2 \dots)$$

A fentebbi geometriai eloszlás egyenletében p paramétert - tehát annak a valószínűségét, hogy az esemény sikeres lesz - a sikertelen események számának várható értékéből kaphatjuk meg. Az adathalmazunkon nagy számosságú mintákon ki tudjuk számolni a sikeres esemény előtti sikertelen események átlagos számát, tehát a várható értéket ($E(X)$), így a p értékét a következőképp kaphatjuk meg:

$$p = \frac{1}{E(X) + 1}$$

A 4.7. ábrán láthatóak a p paraméter értékei különböző **roomsVisited** (összesen meglátogatott különböző szobák száma) értékekre. A p paraméter tehát függ attól, hogy a felhasználó aznap összesen mennyi szobát látogatott meg. Ez logikus, hiszen aki egy nap csak pár szobát látogat meg, az egy szobaváltás során kisebb valószínűséggel látogat meg egy új szobát, mint az, aki egy nap több tíz szobában van. Ezen görbék jól közelíthetők egy hatványfüggvénnyel. Az egyszerűség érdekében minden felhasználói csoportra és minden hierarchiára számoljuk csak ki a közelítő függvényeket. Ehhez a különböző roomsVisited értékek görbéjét a felhasználók száma szerint súlyozzuk és így illesztjük rá a hatványfüggvényt, így tartva meg részben az egész populációt reprezentáló eloszlást. Ezáltal a következő paraméteres függvényt kapjuk, mely adott felhasználói csoportra és adott hierarchiára megmondja, hogy mennyi annak a valószínűsége, hogy a következő szobaváltással új helyszínt fedez fel a felhasználó.

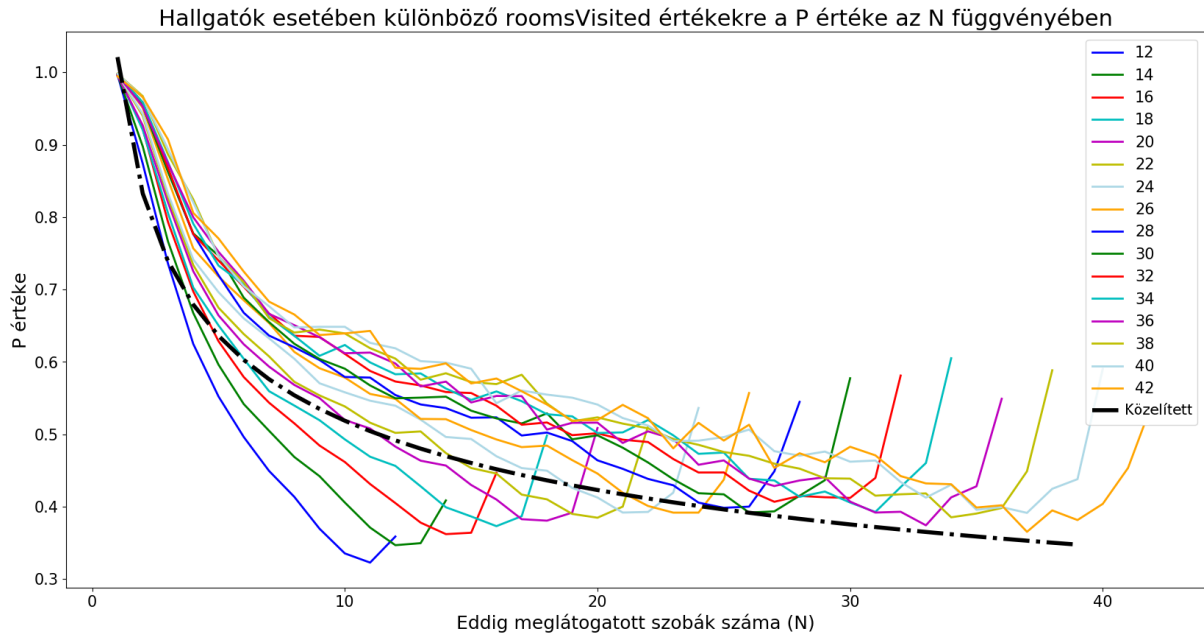
$$P_{felfedez}(D_n) = a * D_n^{-b} \quad (1)$$

ahol a D_n az eddig meglátogatott különböző helyszínek (hierarchiától függően: épület/emelet/szoba) száma. Annak a valószínűsége, hogy egy már meglátogatott helyszínt látogat meg a felhasználó ($1 - P_{felfedez}$). A fentebbi megállapítások a többi hierarchia szinten is igazak, ez látszik a 4.8. és 4.9. ábrán. Az ábrákon fekete szaggatott vonallal van ábrázolva az általunk közelített függvény. Így hat darab függvényt kapunk, mely a két csoportra, minden egyes hierarchia szinten megmondja, annak a valószínűségét, hogy a következő szobaváltással új helyszínt (épület/emelet/szoba) fedez-e fel a felhasználó, függően attól, hogy eddig mennyi különböző helyszínt látogatott meg. A 4.6. ábrán látható folyamatábra döntési helyzeteiben ezen függvényeket értékeli ki a modell, ahol az eddig meglátogatott helyszínek listájából számolja ki a D_n értéket. Így megkapva minden elágazásnál az Igaz ág valószínűségét.

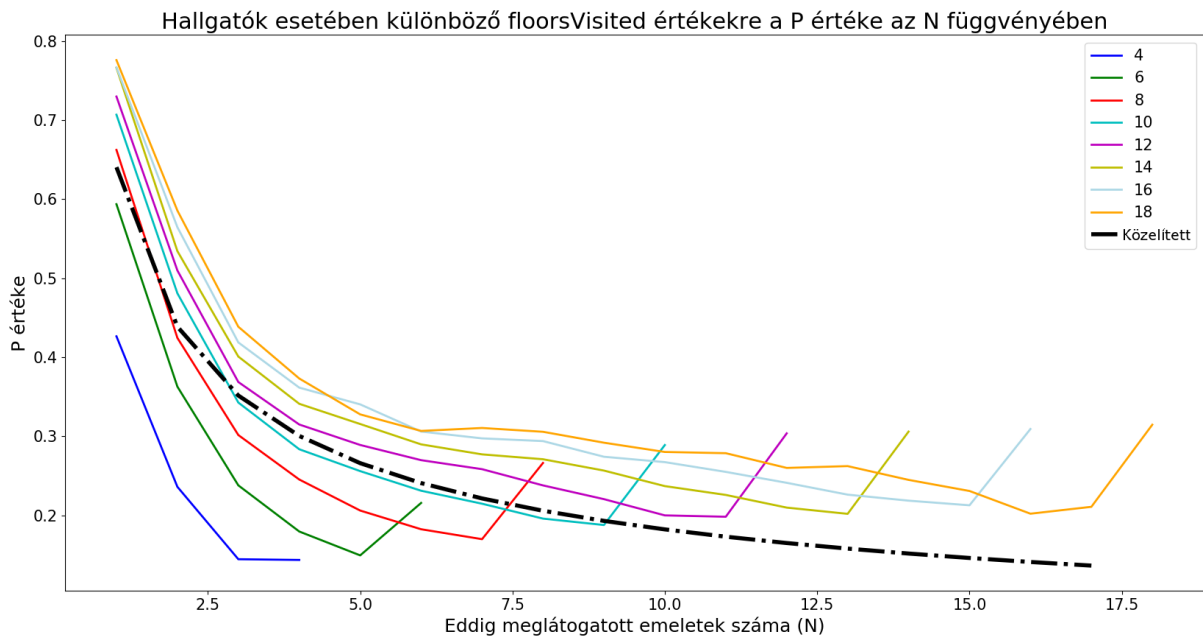
A közelítést egy SciPy Python csomagban implementált függvénnyel végeztük⁴, mely a legkisebb négyzetek módszerével a valós adatpontok és a közelítő függvény által adott adatpontok közötti különbségek négyzetösszegét minimalizálja. Az implementáció a

⁴ https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html

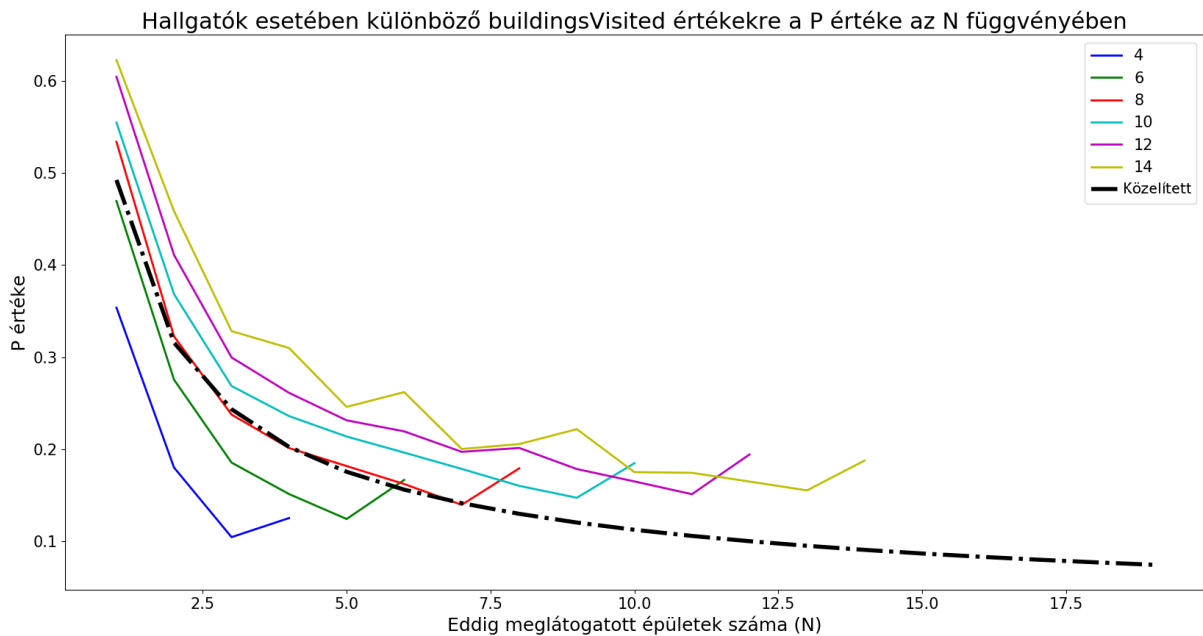
Levenberg-Marquardt algoritmust [13] használja a minimalizálás során, hogy a megfelelő paramétereket ki tudja számolni. Az ábrákon látható adatpontokra lineáris, polinomiális (másodfokú), logaritmusos, exponenciális és hatványos függvényt próbáltunk illeszteni. A felsoroltak közül a hatványfüggvény illeszkedett a legjobban, melyet az közelítések R^2 értékének [18] összehasonlításával döntöttök el.



4.7. ábra: A p értéke az eddig meglátogatott szobák függvényében a hallgatókat tekintve. A különböző függvények különböző roomsVisited (összesen meglátogatott szoba) értékek mellett lettek kiszámítva, ezen értékek láthatók a jelmagyarázatban. Feketével az általunk közelített függvény látható.



4.8. ábra: A p értéke az eddig meglátogatott emeletek függvényében a hallgatókat tekintve. A különböző függvények különböző floorsVisited (összesen meglátogatott emelet) értékek mellett lettek kiszámítva, ezen értékek láthatók a jelmagyarázatban. Feketével az általunk közelített függvény látható.



4.9. ábra: A p értéke az eddig meglátogatott épületek függvényében a hallgatókat tekintve. A különböző függvények különböző buildingsVisited (összesen meglátogatott épület) értékek mellett lettek kiszámítva, ezen értékek láthatók a jelmagyarázatban. Feketével az általunk közelített függvény látható.

Az 1. táblázat tartalmazza az (1) egyenlet „a” és „b” paraméterinek közelített értékeit a különböző felhasználói csoportokra és hierarchia szintekre. Az értékekből több megállapítást is tehetünk. Először is mindkét csoportban a felhasználók gyakrabban látogatnak új szobákat,

mint emeleteket, ugyanakkor gyakrabban mennek addig ismeretlen emeletre, mint épületbe. Ez azt igazolja, hogy az emberek főként emeleten belül mozognak, nem pedig épületek között. Amennyiben csak a paramétereket nézzük, azt kapjuk, hogy a tanárok mozgékonyabbak, azaz gyakrabban látogatnak meg új helyszíneket. Ez viszont ellentmondana az eddigi tapasztalatainknak. Fontos megjegyezni, hogy az itt szereplő értékeket a látogatások számából, pontosabban azok sorozatából számítottuk ki. Mivel a tanárok több időt töltenek az egyes szobákban, így a szobaváltásaik száma kisebb. Így, ha szobát váltanak, akkor nagyobb valószínűséggel fedeznek fel új szobát. Ez azonban nem jelenti azt, hogy a nap végén több különböző szobát látogattak meg, mint a hallgatók, pont a látogatási számuk kisebb értéke miatt. Ezt a modellünk úgy kezeli, hogy a látogatási idő eloszlása is függ a csoporttól, illetve az eloszlásból az A_t attraktorral súlyozva - mely szintén csoportfüggő - generál a modell látogatási időt.

	Hallgatók			Tanárok		
	Épület	Emelet	Szoba	Épület	Emelet	Szoba
a	0.4923	0.6404	1.021	0.6156	0.7486	0.9958
b	0.6411	0.5461	0.2941	0.4131	0.3439	0.2304

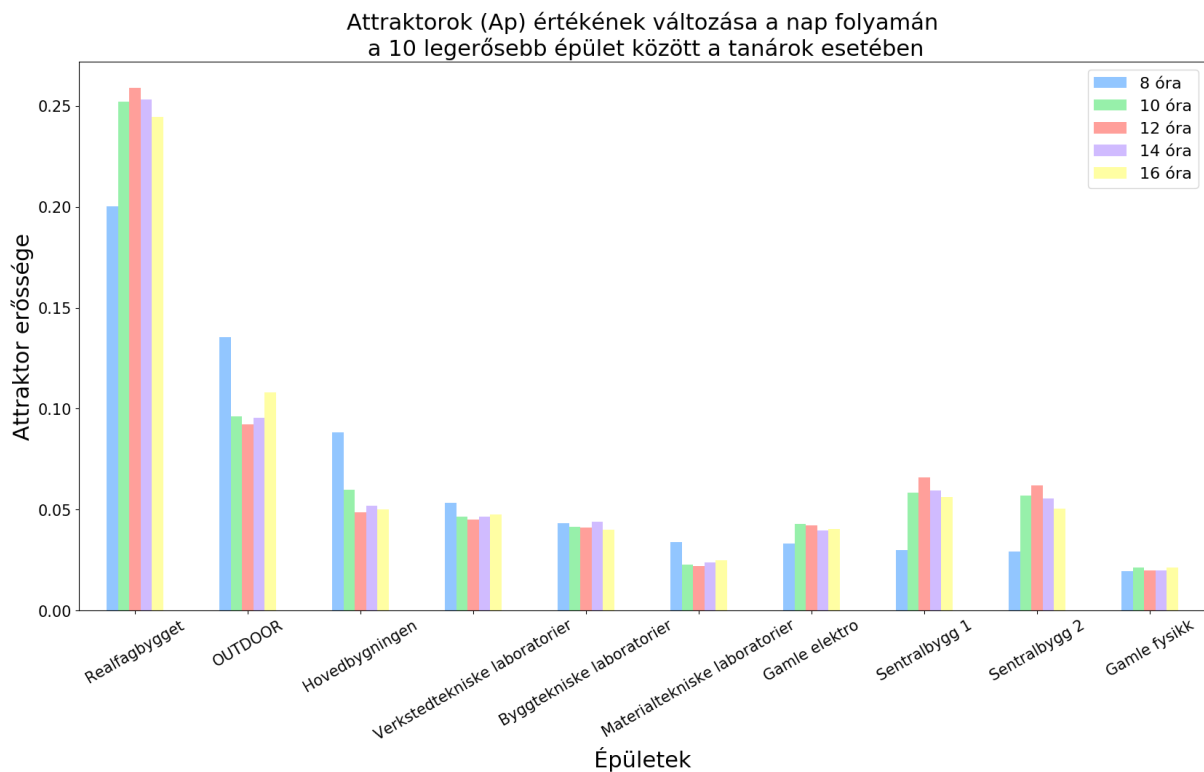
1. táblázat: A p valószínűség hatványfüggvényének paraméterei tanárok és hallgatók esetében, különböző hierarchiaszinteken

4.2.3 Attraktorok

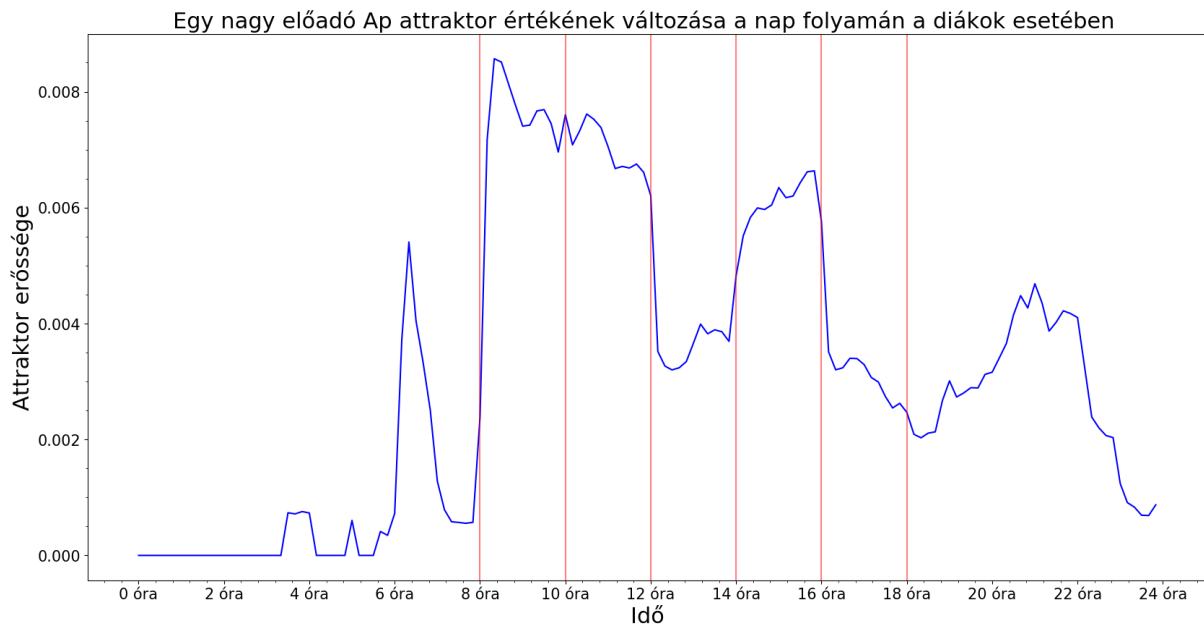
Minden egyes szobához egy attraktor van rendelve, ami az adott szoba „vonzerejét” jelenti. Egy attraktor erősségét $\langle A_p, A_t \rangle$ párosként definiáltuk, ahol A_p -t az adott szobán áthaladó látogatások számából, míg az A_t -t az ott eltöltött látogatási időkből származtattuk. Számolva azzal, hogy egy szoba „vonzereje” időben változik az órarendek és az egyéb ritka események miatt (például ebédidő, előadások kezdete és vége), az attraktorok A_p erőssége 10 perces periodicitással kerül kiszámításra egy egész nap folyamán. Azonban az A_t értékek csak délelőtti és délutáni időszakban kerülnek kiszámításra, mivel egy szoba állandó funkcionalitása miatt, az oda látogató felhasználók nagyjából ugyanannyi ideig tartózkodnak ott. Az A_p értékek normalizálva vannak, úgy hogy az adott időben az összes attraktor értékének összege 1-et ad.

Tekintve, hogy a természetben ezek az értékek additívak, az attraktorokat egyszerűen össze lehet adni, hogy emelet, illetve épület szintű attraktorokat kapjunk. Mivel egy tanár és egy diák egy adott időpontban nagy valószínűséggel más szobába tartózkodik és más szobába fog menni, ezért külön ki van számolva a két embercsoportra az attraktorok értéke.

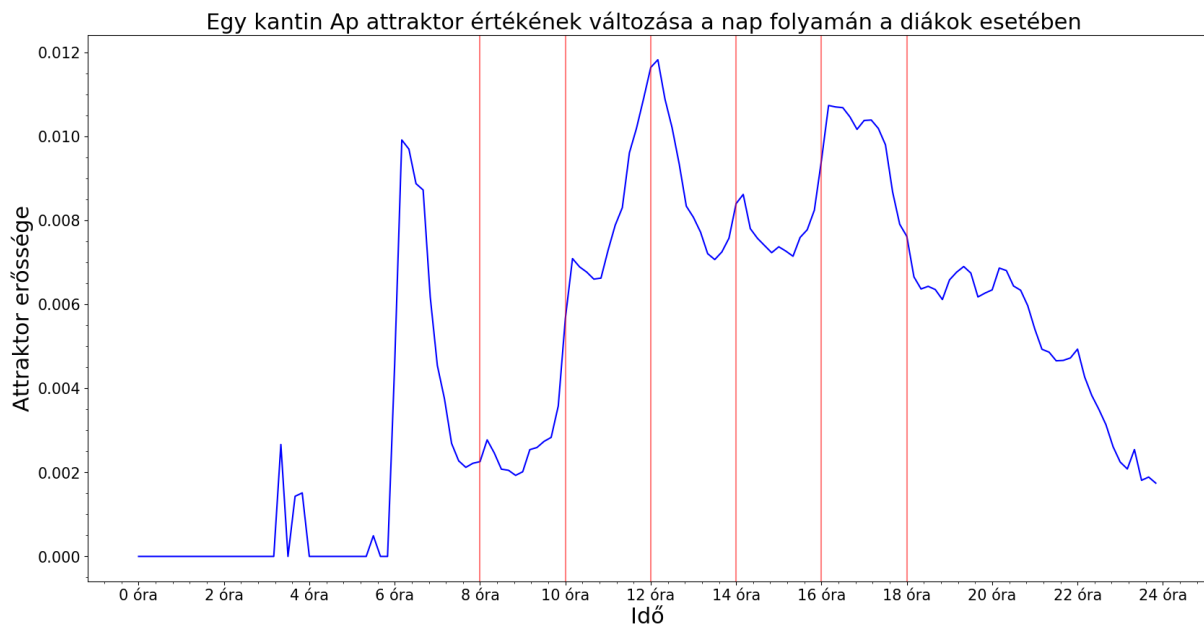
Az A_p értéke tehát egy adott szobára egy adott időben: az ott tartózkodó emberek száma, normálva az összes egyetemen tartózkodó emberek számával. Így lehetséges az, hogy egy szoba attraktor erőssége időben változik, annak ellenére, hogy az ott tartózkodó emberek száma konstans. A 4.10. ábra mutatja, hogy épület szinten, hogy változik az A_p értéke a nap folyamán a tanárok között. A 4.11. ábra és 4.12. ábra mutatja, hogy egy nagyelőadó, illetve egy ebédlő A_p attraktorának értéke hogyan változik a nap során a diákok esetében. Figyeljük meg a két órás periódusokat, amiket a piros függőleges vonal jelez.



4.10. ábra: A 10 legnagyobb épület szintű A_p értékkel rendelkező épület esetében az attraktor változása a nap folyamán a tanárok esetében.



4.11. ábra: A Gamle elektro épület 1. emeletén az EL5-ös nagyelőadó A_p attraktorának változása a nap folyamán. A függőleges piros vonalak 8-18 óráig két óránként vannak behúzva.



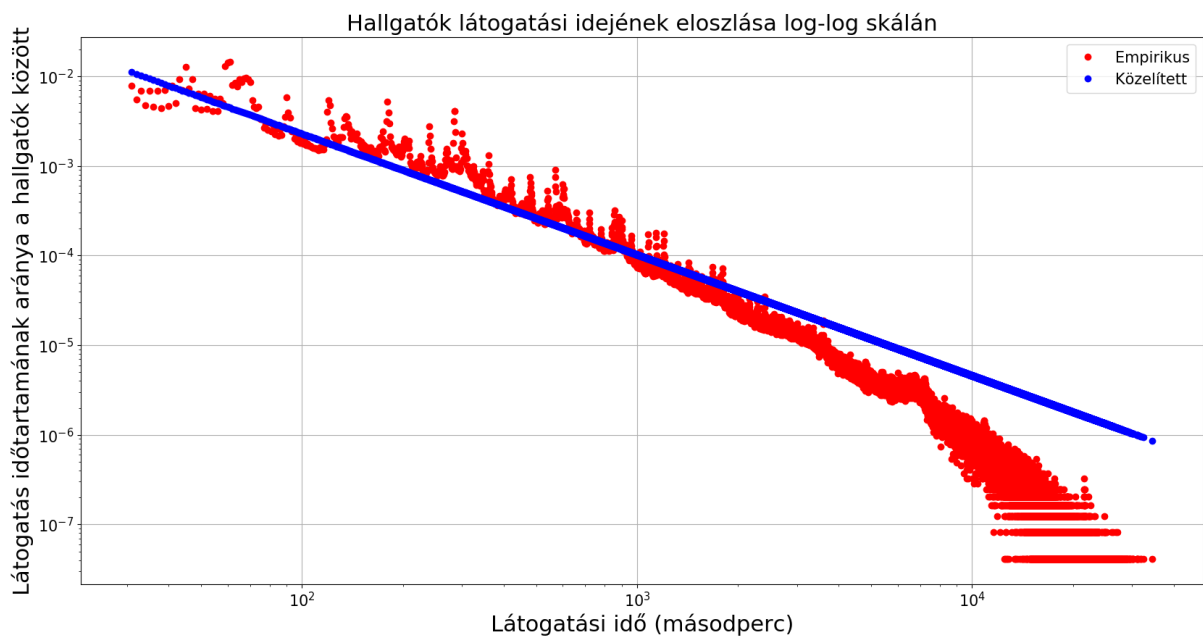
4.12. ábra: A Realfagybgyget épület 1. emeletén a Sit Kafe ebédlő A_p attraktorának változása a nap folyamán. A függőleges piros vonalak 8-18 óráig két óránként vannak behúzva.

Az A_t értéke egy adott szobára: az egész adathalmazt tekintve megvizsgáljuk a délelőtti és délutáni látogatások hosszát. Az ott töltött idők eloszlásából kiszámoljuk az IQR, azaz az interquartilis terjedelmet, ami a 75. percentilis és a 25. percentilis különbsége. Meghatározunk egy alsó és egy felső határértéket. Az alsó: $H_{\text{alsó}} = 25. \text{ percentilis} - 1,5 \cdot \text{IQR}$. A felső $H_{\text{felső}} = 75. \text{ percentilis} + 1,5 \cdot \text{IQR}$. Természetesen az alsó határérték nem lehet negatív. A két határérték

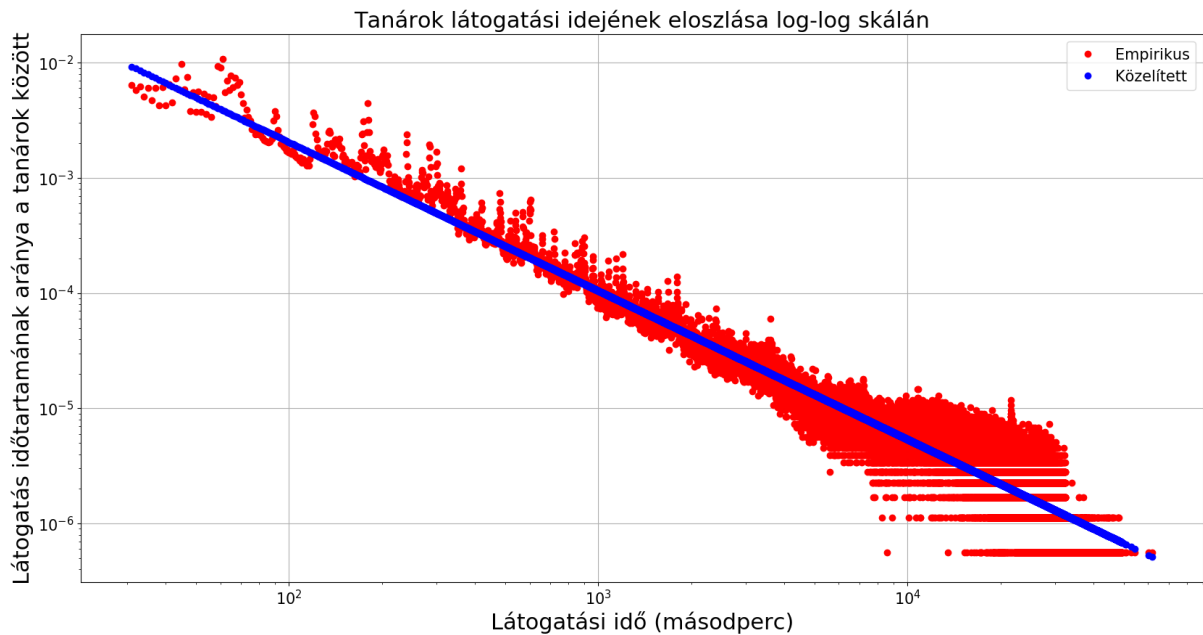
alkotja az A_i -t. Ezzel fogja a modell súlyozottan kiválasztani a látogatási időket, ahogy az a következő fejezet végén részletezésre kerül.

4.2.4 Látogatási idők

Több tanulmány szerint a különböző szobákban eltöltött idő TPL (Truncated Power Law) eloszlást követ [11] [16]. Ezt az állítást úgy bizonyítjuk, hogy kiszámoljuk a látogatási idők eloszlását az adathalmazunkon. Mivel a felhasználók mindig egy szobához vannak rendelve, ezért csak szoba szinten számoljuk ki az látogatási idő eloszlást. Az egyes látogatások időtartamát egy mindkét tengelyen logaritmikus skála segítségével ábrázolva kapjuk a 4.13. és 4.14. ábrán pirossal kirajzolt empirikus eloszlást, a hallgatók és a tanárok esetében.



4.13. ábra: A hallgatók egyes szobában tett látogatási idejének eloszlása log-log skálán ábrázolva. Pirossal a megfigyelt adatok, késsel a közelített Pareto I eloszlás van ábrázolva.



4.14. ábra: A tanárok egyes szobában tett látogatási idejének eloszlása log-log skálán ábrázolva. Pirossal a megfigyelt adatok, kékkel a közelített Pareto I eloszlás van ábrázolva.

Az empirikus eloszlás ténylegesen Pareto I eloszlással közelíthető. A modellünkben a látogatási idők hosszát 30 másodpercben minimalizáltuk, ezért is szerepel az alábbi függvényben a 30-as konstans, mert az x az legalább 30. Legyen x a látogatási idő másodpercekben mérve, az eloszlás sűrűségfüggvénye az alábbi:

$$P(\text{látogatási idő} = x) = \alpha * 30^\alpha * x^{-(1+\alpha)}$$

Az adatokból a következő eloszlásokat kapjuk, melyek a 4.13. és 4.14. ábrán kékkel szerepelnek:

- Hallgatók esetében, alfa = **0,3495**. Így a látogatási idők eloszlásai a szobák szintjén, így írható le:

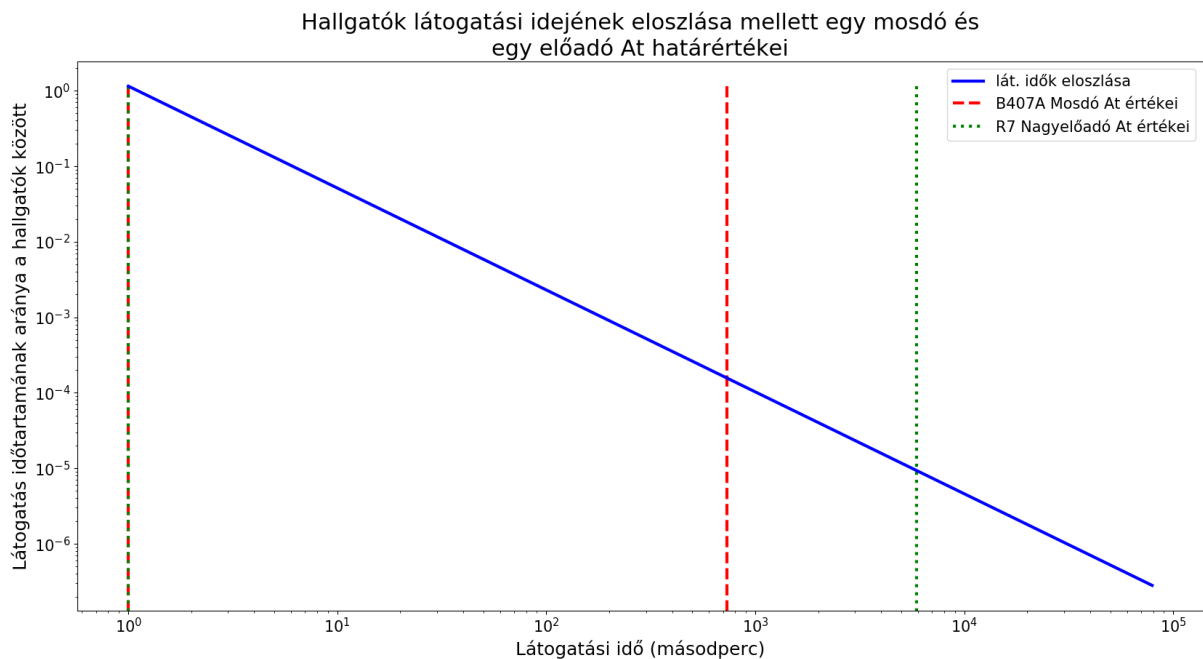
$$\circ P(\text{látogatási idő} = x \text{ sec}) = 1.1474 * x^{-1.3459}$$

- Tanárok esetében, alfa = **0,2894**. Így a látogatási idők eloszlásai a szobák szintjén, így írható le:

$$\circ P(\text{látogatási idő} = x \text{ sec}) = 0.7744 * x^{-1.2894}$$

Annak érdekében, hogy a látogatási idők az egyes szobákban valóságghűen visszaadják az adott szoba jellegzetességét a látogatási idő sűrűségfüggvényéből az adott hely időfüggő A_i attraktora alapján határozzuk meg a pontos T látogatási időt. A fentebb látható és kiszámolt

eloszlások értékét, tehát a látogatási idő értékkészletét bekorlátozzuk az A_t felső és alsó határával. Így a kapott eloszlás ugyancsak Truncated Power Law marad, de megadjuk, hogy milyen értékeket vehet fel. A 4.15. ábrán látható a diákok esetében egy mosdó és egy nagy előadó A_t attraktorának felső és alsó határa.



4.15. ábra: A hallgatók becsült látogatási idejének eloszlása (kék) mellett a B407A mosdó és az R7 nagyelőadó A_t határértékei. Az alsó határértékek egybe esnek.

4.2.5 Mozgási sebesség és útkeresés

Átlagosan a gyaloglási sebességet 4 km/óráként határoztuk meg. Tovább fejlesztési lehetőség van abban, hogy még pontosabb sebesség eloszlást nyerünk ki a felhasználók napi mozgási mintázatából, így precízebb modellt kaphatunk. Jegyezzük meg, hogy az egyetemen keresztül haladó forgalom biciklistákat is magában foglalhat – ugyan, ez nem jellemző. Illetve a lépcsők és liftek egy másik problémát is felvetnek ahol további pontosítás végezhető.

Továbbá feltételeztük, hogy a legrövidebb útvonalakat járják be a felhasználók a campuson. Ezt a modellezés egyszerűsége és a MazeMap útvonaltervező alkalmazás jelenléte indokolta a campuson, mivel ez a legrövidebb utat mutatja meg a felhasználóknak. Bizonyított, hogy a hallgatók inkább a szemeszter elején használják ezt az alkalmazást, amíg meg nem tanulják a campuson a mindennap bejárt útvonalait [16].

4.3 Csoportdinamika

A csoportdinamikát a modell az időben változó attraktorok segítségével veszi figyelembe. A felhasználók az előadótermekben és ebédlőkben összegyűlnek meghatározott időkből a megnövekedett attraktor erősség miatt. Illetve a saját csoportjukra jellemző időt töltenek az egyes szobákban. A felhasználók mozgékonyasága, az, hogy mennyi különböző helyszínen jártak a 4.2.2 szakaszban ismertetett függvényről függ. Mely, ahogy ott említésre is került, a két csoport esetében különböző paraméterezéssel bír. Ezzel a módszerrel a csoportok sajátosságát adatvezérelten, az egyes paraméterek csoporthoz igazításával építettük a modellbe. Így egy adott helyen és egy adott időben az embertömeg nagysága valóságos lesz. A felhasználókat a modell nem címkézi fel, nem tartja őket számon. Ez a személyes adatok védelme miatt, és a GDPR rendelet miatt is fontos, azonban így is megfelelő számos felhasználási területre.

Amennyiben az egyéni felhasználók az érdekesek, pontosan definiálni kell az embercsoportokat és a rendszeres csoportszintű eseményeket (pl.: előadás kezdete), amik hirtelen szuper erős attraktorokat eredményeznek. Ezt minden csoporthoz létrehozott FIFO esemény sorral lehet megvalósítani, ahol az esemény helye is meg van adva. A csoportok szerkezete vagy előre megadott információ (campus, irodapark) vagy a felhasználók nyomon követésével meghatározható [10].

5 Alkalmazási lehetőségek

A 4. fejezetben prezentált általános kültéri és beltéri mozgási modellt nagyon sok szektorban és helyzetben lehet alkalmazni. Az emberek egy kevert kül- és beltéri környezetben az idejük nagy részét az épületeken belül töltik, a modellünk pont az ilyen adottságú helyszíneket szimulálja. A mindennapi életben számos ilyen közeg van: irodaparkok, kórházak, bevásárlóközpontok, lakóparkok. Így pozíción alapuló szolgáltatásoknak igen nagy piaca van, amit még korántsem telítettek be. Számítalan olyan szolgáltatás születhet, mely az emberi mozgásra épít, és ezen tudást felhasználva nyújt cél specifikus megoldásokat.

Beltéri útvonaltervező alkalmazások - mint például az említett MazeMap - pontosabb és optimálisabb utat mutatnának, ha figyelembe vennék az emberi mozgás sajátosságát az adott helyszínen. Ha az útvonalkeresésnél számolnának azzal, hogy az adott időben hol, mennyi ember tartózkodik, akkor el tudnák kerülni az olyan lépcsőfordulókat vagy folyosókat ahol tömegek vannak, feltárva a szűk keresztmetszeteket az adott környezetben. Így alternatív - esetleg hosszabb - útvonal ajánlásával gyorsabban tudnák elérni a felhasználók a céljukat, mely jobb felhasználói élményt biztosít. Ez ugyanúgy alkalmazható kórházak és bevásárlóközpontok esetében, mint egy kampuszon.

Manapság a reklámpia az informatikával összefonódva arra törekszik, hogy minél személyre szabottabb, ezáltal minél hatásosabb reklámokat tudjanak a felhasználóknak mutatni. Az emberi mozgás elemzésével a bevásárlóközpontok pontosabban tudnák felmérni az egyes bolthelyiségek értékét és ezt akár az idő függvényében is meg tudnák határozni. Így egy képet kapnának, hogy mely helységekben érdemes üzletet nyitni az egyes boltosoknak, vagy melyekre kell főleg koncentrálni. Az egyes boltok bérleti díjának látogatási számhoz való illesztésével extraprofitra tehetnének szert. Ezen felül a folyosókon a látogatók egyes csoportjaihoz igazíthatnák a reklámokat, napszaktól függően. Emellett az egyes boltok logikai kapcsolatát, az egymásra gyakorolt hatását, vonzerejét is meg lehetne figyelni, amellyel a nem egyértelmű vásárlói magatartási mintákat lehetne felderíteni. Ezt felhasználva több boltos átívelő reklámkampányokat lehetne indítani termékkapcsolásokkal vagy kuponakciókkal.

Nem csak a már meglévő épületek és környezetük esetében használható a fentebb kidolgozott modell. Az épületeket a jövőben használó emberek mozgását szimulálva már tervezési fázisban be lehet azonosítani a szűk keresztmetszeteket, az egyes folyosók tényleges

áteresztő képességét. A kiürítési tervek megszerkesztésekor is nagy segítséget jelenthet, ha rendelkezésre áll az emberek valószínűsíthető útvonala, mellyel még biztonságosabb épületeket lehetne megalkotni. Üzemeltetési szempontból is használható ez a megközelítés. Ha egy kórházat veszünk példának, fontos számon tartani, hogy a betegek, a személyzet és a felszerelések hol vannak. Egy esetleges váratlan helyzetben minden egyes másodperc kritikus tud lenni egy beteg életének szempontjából. Így az, hogy milyen gyorsan tudnak a kórterembe érni a nővérek és az orvosok és mennyi idő alatt tudják áttolni a beteget a műtőbe kulcsfontosságú. Figyelembe véve az emberek mozgását ezt is optimalizálni lehetne egy kórház tervezése során[2].

6 Összefoglalás

A dolgozat fő célja egy olyan, eddig még nem létező adatvezérelt emberi mobilitási modell létrehozása, mely analitikus jellegű, tetszőleges több-épületes környezetbe adaptálható és mind az egyéni, mind a csoportos magatartást magában foglalja. A rendelkezésre álló kivételesen nagy adathalmazt lépésekben dolgoztuk fel. Létrehoztunk egy olyan térbeli hármas tagozódást, mely más hasonló környezetben is könnyedén felállítható és kezeli az esetleges kis mértékben jelen lévő hiányzó adatot (hiányzó szoba alaprajzokat) és a mérési pontatlanságot. Majd az adathalmazból egy hatékony saját algoritmus által kiszűrtük a pozíció meghatározása során keletkezett pontatlanságokat. A egyes felhasználóknak meghatároztuk bejárt útvonalát, melyet látogatások szekvenciájaként tároltunk. Ezekből numerikus metrikákat származtattunk, melyekkel jól le tudjuk írni a felhasználók mozgási magatartását. Gépi tanuló algoritmus segítségével a felhasználók halmazát két fő csoportra bontottuk fel. Az adatokból egyértelműen kitűnt, hogy ez a tanárok és a hallgatók csoportja.

Korábbi tanulmányokat vizsgálva, és mások az emberi mozgás sajátosságaira tett megállapításait figyelembe véve megalkottuk a mozgási modellünket. Erőssége, hogy figyelembe veszi az épületek hierarchiáját, a kevert kül- és beltéri környezetet és kevés paraméterrel rendelkezik. A két felhasználói csoport merőben eltérő magatartását magában foglalja azáltal, hogy a belső paraméterek különbözőek a csoportokra.

A modell számos gyakorlati problémára alkalmazható, mint például épületek tervezési fázisában terheléselosztás optimalizálására, vészhelyzeti kiürítési tervek készítésére vagy bevásárlóközpontok bérleti díjainak kiszámolásában profitmaximalizálás céljából.

A modellt több szempontból is lehetne finomítani, pontosítani. Először ez a modell csak gyalogos forgalmat vesz figyelembe, holott - az adatokból is látszott, - a kampuszon sokan járnak biciklivel és, mivel egy forgalmas út is keresztezi a területet a buszos forgalom is jelen van. Ezeket meg lehetne figyelni, külön kitérve, hogy mikor és miért használják az emberek ezeket a közlekedési formákat és be lehetne építeni a modellbe. Másodsorban az attraktorok koncepcióját is lehetne finomítani. Igaz az egész munka lényege az volt, hogy egy általános modellt dolgozzunk ki, az attraktorokat egy 1-mély döntésifával lehetne helyettesíteni úgy, hogy az értékük függene attól, hogy az adott személy honnan jött. Így tulajdonképpen a modell már nem emlékezet nélküli lesz, és pontosabban visszaadná az egyének mozgási mintázatát.

Ugyanakkor ehhez a módosításhoz sokkal nagyobb adathalmazra lenne szükség ilyen sok szoba esetében. Továbbá a modell azon lépését is fejleszteni lehetne ahol a felhasználó következő meglátogatandó szobáját választjuk ki. A mostani megoldást egy Véletlen Sétával (Random Walk) lehetne helyettesíteni és elemezni azt, hogy mennyire illeszkednek az adatokra ezen modellek. Ugyan, több tanulmány [16] [9] használ hasonló matematikai modelleket, mint például a Lévy-folyamat vagy Markov-folyamat, de ezen munkák során sokkal kevésbé részletesebb és kültéri adatokat használnak. Érdekes kutató munka lenne bizonyítani, hogy ezen modellek egy ilyen beltéri környezetben is megállják-e a helyüket.

7 Irodalomjegyzék

- [1] Aschenbruck, N., Ernst, R., Martini, P.: *Indoor mobility modelling*. In: GLOBECOM Workshops (GC Wkshps), 2010 IEEE. pp. 1264–1269. IEEE (2010)
- [2] Biczok, G., Diez Martinez, S., Jelle, T., Krogstie, J.: *Navigating mazemap: indoor human mobility, spatio-logical ties and future potential*. In: Pervasive Computing and Communications Workshops (PERCOMWorkshops), 2014 IEEE International Conference on. pp. 266–271. IEEE (2014)
- [3] Boldrini, C., Passarella, A.: *Hcmm: Modelling spatial and temporal properties of human mobility driven by users' social relationships*. *Computer Communications* 33(9), 1056–1074 (2010)
- [4] Cai, D., Zhang, C., & He, X. (2010, July). *Unsupervised feature selection for multi-cluster data*. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 333-342). ACM.
- [5] Caliński, T., & Harabasz, J. (1974). *A dendrite method for cluster analysis*. *Communications in Statistics-theory and Methods*, 3(1), 1-27.
- [6] Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.L.: *Understanding individual human mobility patterns*. *Nature* 453(7196), 779–782 (2008)
- [7] He, X., Cai, D., & Niyogi, P. (2006). *Laplacian score for feature selection*. In Advances in neural information processing systems (pp. 507-514).
- [8] Jenkins, P.L., Phillips, T.J., Mulberg, E.J., Hui, S.P.: *Activity patterns of californians: use of and proximity to indoor pollutant sources*. *Atmospheric Environment. Part A. General Topics* 26(12), 2141–2148 (1992)
- [9] Johnson, D.B., Maltz, D.A.: *Dynamic source routing in ad hoc wireless networks*. In: *Mobile computing*, pp. 153–181. Springer (1996)
- [10] Kim, T.S., Kwon, J.K., Sung, D.K.: *Mobility modeling and traffic analysis in three-dimensional high-rise building environments*. *Vehicular Technology, IEEE Transactions on* 49(5), 1633–1640 (2000)
- [11] Lee, K., Hong, S., Kim, S.J., Rhee, I., Chong, S.: *Slaw: A new mobility model for human walks*. In: *INFOCOM 2009, IEEE*. pp. 855–863. IEEE (2009)
- [12] MazeMap, *Indoor Maps & Wayfinding* <https://www.mazemap.com/> (2018.10.28)
- [13] Moré, J. J. (1978). *The Levenberg-Marquardt algorithm: implementation and theory*. In *Numerical analysis* (pp. 105-116). Springer, Berlin, Heidelberg.
- [14] Pearson, K. (1901). *LIII. On lines and planes of closest fit to systems of points in space*. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11), 559-572.

- [15] Prentow, T.S., Ruiz-Ruiz, A.J., Blunck, H., Stisen, A., Kjærgaard, M.B.: *Spatio-temporal facility utilization analysis from exhaustive wifi monitoring*. Pervasive and Mobile Computing 16, 305–316 (2015)
- [16] Rhee, I., Shin, M., Hong, S., Lee, K., Kim, S.J., Chong, S.: *On the levy-walk nature of human mobility*. IEEE/ACM transactions on networking (TON) 19(3), 630–643 (2011)
- [17] Rousseeuw, P. J. (1987). *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*. Journal of computational and applied mathematics, 20, 53-65.
- [18] Wikipedia: *Coefficient of determination*, https://en.wikipedia.org/wiki/Coefficient_of_determination (2018.10.27)
- [19] Wikipedia: *Curse of dimensionality*, https://en.wikipedia.org/wiki/Curse_of_dimensionality (2018. 10. 24.)
- [20] Wikipedia: *Elbow-method*, [https://en.wikipedia.org/wiki/Elbow_method_\(clustering\)](https://en.wikipedia.org/wiki/Elbow_method_(clustering)) (2018. 10. 24.)
- [21] Wikipedia: *k-means clustering*, https://en.wikipedia.org/wiki/K-means_clustering (2018. 10. 24.)
- [22] Yoshimura, Y., Sobolevsky, S., Ratti, C., Girardin, F., Carrascal, J.P., Blat, J., Sinatra, R.: *An analysis of visitors' behavior in the louvre museum: a study using bluetooth data*. Environment and Planning B: Planning and Design 41(6), 1113–1131 (2014)