

Diagnosztikai adatok kinyerése és hasznosítása okosautós környezetben

SIK DÁVID, EKLER PÉTER, LENGYEL LÁSZLÓ

BME Automatizálási és Alkalmazott Informatikai Tanszék
siktdavid@gmail.com, {ekler.peter; lengyel}@aut.bme.hu

Kulcsszavak: OBD, CAN, járműdiagnosztika, IoT, SensorHUB, VehicleICT, Big Data, Hadoop, üzleti intelligencia

A járművek jelenleg már nemcsak közlekedési vagy szállítási funkciókat szolgálnak. Ugyanolyan szenzorrendszerek, adatgyűjtő egységek, mint például az okostelefonok. Megfelelő eszközökkel és okostelefonunkkal kinyerhetjük ezen adatokat, és bekapcsolhatjuk járművünket az Internet of Things (IoT) világába.

A SensorHUB keretrendszert felhasználva, a kinyert adatokra alapozottan közösségi és elemző mobil alkalmazást dolgoztunk ki.

1. Bevezetés

Napjainkban az okos eszközök és a hozzájuk kapcsolódó szolgáltatások egyre jelentősebb szereppel bírnak a legtöbb iparág számára. Az elmúlt évtizedben nagyban nőtt az internetre kapcsolt eszközök száma, folyamatosan épül az Internet of Things (IoT) világa. Ez főként az okostelefonok és a hordozható „kütyük” fejlődésének, az egyszerű, de nagy számban elérhető és alkalmazott szenzoroknak köszönhető.

Ha egy okostelefonra nemcsak telefonként tekintünk, hanem szenzorok halmazaként, akkor azt látjuk, hogy ezen adatok felhasználásával hihetetlen mennyiségű információ nyerhető ki. Gondoljuk csak a földrajzi koordinátákra (mind GPS, mind a bázisállomások alapján), a mikrofonra, a kamerára, a gyorsulásmérőre, a fényérzékelőre, pulzusszámmérőre stb. Ezek csak a legalapvetőbb, szinte minden okostelefonban és okosórában elérhető mérőegységek. Éveken belül egy IP-alapú szenzor átlagos ára nagyságrendileg 500 Ft lesz, 2020-ban pedig 30 milliárd IP-szenzor lesz működésben [1].

Ha kicsit absztraktabban gondolunk egy autóra, vagy bármilyen járműre, akkor könnyen beláthatjuk, hogy az is egy szenzorhalmaz. Annyiban tér el egy okostelefontól, hogy adatai nem töltődnek fel az internetre, de ugyanúgy minden olvasható a műszerfalán, a kijelzőkön, mérhető az egyes rendszerek feszültsége, teljesítménye stb.

1. ábra Bluetooth-os OBD adapter



A szenzoradatokhoz kapcsolódási lehetőség az OBD-II (On-board Diagnostics II) és a CAN bus (Controller Area Network bus) interfészek, amelyek segítségével a járművekből kinyerhetők különféle információk; például a sebesség, fordulatszám, valamint különböző hőmérséklet és nyomásértékek. Ezeket felhasználva elfogadhatóan közelíthető számos hasznos információ, például a jármű aktuális fogyasztása, illetve CO₂-kibocsátása [3].

Az OBD-II port egy egyszerű adapterrel Bluetooth-on vagy USB-n keresztül tud kommunikálni (1. ábra), adatfolyamot küldeni. Ezeket egy okostelefon segítségével már könnyen továbbítani lehet a szerveroldal irányába.

A kinyert és a származtatott adatokra alapozva, majd kiegészítve további funkciókkal különböző okosautó-alapú megoldások készíthetők a jármű használója, de akár közösségek vagy ipari résztvevők (flották, biztosítók, szervizek stb.) számára.

A cikk bemutatja a két legismertebb járműdiagnosztikai technológiát és az adatok kinyerésének módját. A harmadik szakaszban tárgyalásra kerül a SensorHUB keretrendszer és koncepció. A következő két fejezetben egy-egy SensorHUB-ra épülő, a jármű szakterülethez kapcsolódó okostelefonos alkalmazás kerül bemutatásra, a Social Driving közösségi alkalmazás, illetve az ObdCanCompare mérőalkalmazás. A hatodik szakaszban az adatok Big Data alapú feldolgozása, kiértékelési és vizualizációs lehetőségek ismertetésére kerül sor. Az összefoglalásban pedig rövid kitekintést kívánunk mutatni a SensorHUB-ban rejlő jövőbeli lehetőségekre.

2. Járműdiagnosztika (OBD és CAN)

Az Amerikai Egyesült Államok Kaliforniában működő levegőtisztaság-védelmi hatósága (CARB) felismerte a folyamatos állapotfelügyelet jelentőségét, és a gyártók részére előírásban rögzítette a gépjárműemisszió-korlátozó műszaki rendszereinek fedélzeti ellenőrzési kötelezettségét [4]. Az OBD-I néven ismertté vált fedélzeti

diagnosztikai rendszert az 1988-as modellévtől kezdve kötelezővé tették. A szabályozás műszaki előírásait SAE (Society of Automobile Engineers) szabványok és ajánlások rögzítik.

Az OBD-I előírásokat az 1994-es modellévtől kezdődően felváltották az OBD-II előírások. Az OBD-II a személygépjárművekre és a könnyű haszongépjárművekre, az 1996-os modellévtől kezdődően pedig a dízelmotorral meghajtott gépjárművekre is hatályos az USA-ban. Az OBD-II európai megfelelője az EOBD, amelynek bevezetését az Európai Unió tagországaiban egy 2001-es irányelv írta elő. Az OBD-II szabvány 9 különböző módot (szintet) definiál a diagnosztikai eszközök számára. Az OBD rendszert nem teljes körűen támogató járműveknél előfordulhat, hogy a jármű vezérlőegysége csak az első néhány módot képes megvalósítani.

A diagnosztika 9 szintje a következő [4, 5]:

- 1. mód: diagnosztikai adatok kiolvasása
- 2. mód: Freeze Frame paraméterek (egy adott hibakód eltávolításakor mért paraméter információk)
- 3. mód: hibakód kiolvasás
- 4. mód: hibakód tároló törlése
- 5. mód: Lambda szonda tesztértékeinek kijelzése
- 6. mód: nem folyamatosan figyelt funkciók mérési értékei, vizsgálati és küszöbértékek kijelzése
- 7. mód: hibakód kiolvasása
- 8. mód: beavatkozó tesztek (Európában nem használatos)
- 9. mód: járműspecifikus adatok és információk

Az 1. módot tudjuk használni a folyamatos, vezetés közbeni adatkiolvasásra, a további módok főleg a szervizelés folyamán kerülnek előtérbe.

Ha az autónkban vezetés közben, vagy a gyújtás ráadása után a műszerfalán felvillan a MIL (Malfunction Indicator Light) indikátorlámpa, akkor abból valamilyen hibára következtethetünk. Ennek és a további hibajelzések feloldásáról elsősorban az autó üzemeltetői kézikönyvből érdemes tájékozódni és megpróbálni elhárítani az okokat. Összetettebb veszélyek és hibák esetén nem biztos, hogy találunk a kézikönyvben konkrét megoldást, ilyenkor célszerű egy szakszervizt felkeresni és ott kiolvasatni a hibakódot.

A hibakód (DTC, Diagnostic Trouble Code) kiolvasása az eddig ismertetett OBD rendszeren keresztül történik (3-as és 7-es mód), a kód jelentésének feloldása után lehetőség van a hibakód tárolásakor mért paraméterek kiolvasására (2-es mód), a hiba elhárítására, valamint a hibakód törlésére is (4-es mód).

Az OBD-II szabvány szerint a hibakódok öt karakterből állnak, amelyből az első karakter betű, a további négy karakter szám, pl. P0506. Az előírások megkövetelik a gyártóktól az OBD adatokhoz történő egységes hozzáférési lehetőséget. Ennek megfelelően a diagnosztikai csatlakozó kivitelét, annak elhelyezését és az adatátvitelhez használható protollokat is szabályok rögzítik.

Az aljzat beépítésére olyan helyet kell választani, ahol a karbantartó személy számára hozzáférhető, de a

normál üzemeltetés során védve van. A gyakorlatban a csatlakozó a műszerfalban vagy alatta, a kormánykerék közelében, egy méteren belül található (2. ábra) [5].



2. ábra OBD csatlakozó helye egy Rover 25-ös modellben

Az OBD csatlakozást sokáig csak a szervizekben használták. 2005-ben az ELM Electronics készítette el az ELM327 mikrovezérlőt, aminek segítségével az OBD-s jeleket a számítógépen elérhető soros RS-232 porton olvashatóvá alakítja [6]. Azóta már megjelentek az USB-s, valamint a Bluetooth-os és Wi-Fi-s átalakítók is.

Az OBD port is a jármű CAN bus rendszerére csatlakozik, de rajta keresztül kevesebb szenzor érhető el, mert az újabbak nincsenek benne az eredeti szabványban [3]. A CAN bus célja, hogy a járművekben található vezérlők és eszközök közvetlenül tudjanak kommunikálni egymással különböző üzenetekkel, egy soros buszrendszeren keresztül. Vagyis nem kell mindenkit mindenkivel összekötni a kommunikációhoz, jelentősen csökkentve a felhasznált vezetékek mennyiségét, tömegét és árát, valamint a kommunikációs hibák felderítésének nehézségét [7].

A CAN bus fejlesztése 1983-ban kezdődött meg a Bosch-nál. 1986-ban rögzítette a szabványt a SAE, 1987-ben pedig elkészült az első vezérlő az Intel és a Philips gyárában. 1991-ben jelent meg a CAN 2.0-ás változata, majd ezután is még számos protokollváltozat készült el és ma már az ipar szinte összes területén alkalmazzák. A CAN bus elérése és az adatok kinyerése már sokkal összetettebb és specifikusabb feladat. Ebben nyújt segítséget az Inventure Kft. által gyártott FMS Gateway eszköz, amely az adatokat CAN, RS-232 és Bluetooth-os kimenetre tudja továbbítani, buszokból, kisáruszállítókból, teherautókból és akár személyautókból is (3. ábra) [6].

3. ábra FMS Gateway szimulátor



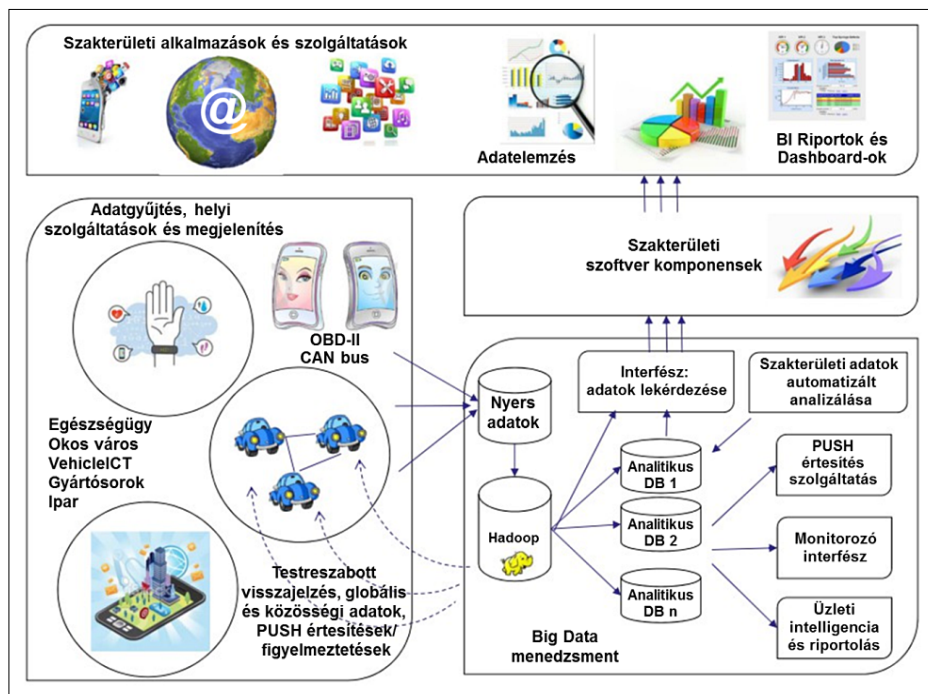
3. A SensorHUB keretrendszer

A Budapesti Műszaki és Gazdaságtudományi Egyetem Automatizálási és Alkalmazott Informatikai Tanszékén fejlesztett SensorHUB keretrendszer különböző szakterületek adatait tárolja, elemzi, dolgozza fel és teszi elérhetővé alkalmazások és szolgáltatások számára. Alkalmazási területei lehetnek például a közlekedés, járművek, okos városok, egészségügy, gyártósortok, valamint további szakterületek. Ezek tipikusan olyan területek, ahol nagy mennyiségű szenzoradatok feldolgozása, tárolása, elemzése és felhasználása a feladat.

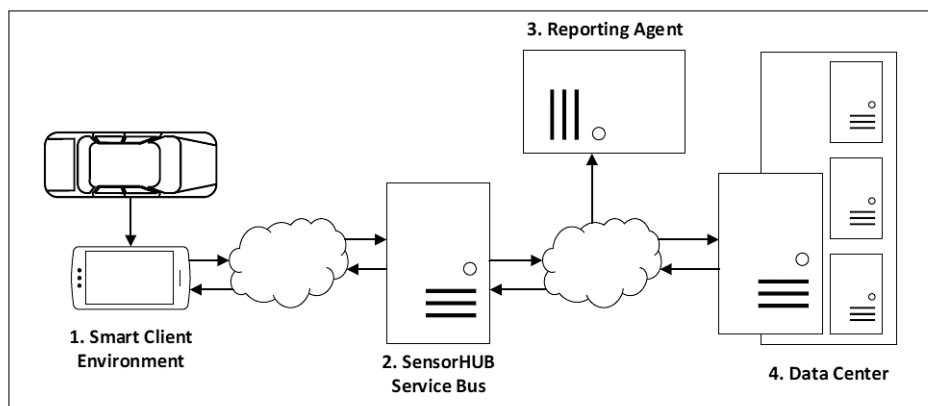
A SensorHUB komponensei (4. ábra) [2]:

- Szenzorok, adatok gyűjtése, helyi szolgáltatások, megjelenítés és adatátvitel.
- Felhő alapú háttérrendszer, Big Data menedzsment funkciókkal.
- Szakterület specifikus szoftver komponensek.
- Alkalmazások, szolgáltatások, üzleti intelligencia jelentések és elemzések.

4. ábra A SensorHUB keretrendszer felépítése



5. ábra A VehicleICT architektúráis felépítése



A SensorHUB koncepciónak előnye, hogy a gyakran előforduló, adatokkal kapcsolatos műveleteket csak egyszer szükséges implementálni és ezek után megfelelő dokumentációkkal támogatva széles körű tud olyan alkalmazást készíteni, ami felhasználva a SensorHUB keretrendszert, képes lesz adatai megjelenítésre, feltöltésre, elemzésre [9]. 2015-ben a SensorHUB koncepció és keretrendszer elnyerte a Pro Progressio Alapítvány Innovációs Díját [10].

A SensorHUB keretrendszer részeként készült VehicleICT szakterületi megoldás ezt a kapcsolatot teremti meg; vagyis az autónk, további néhány eszköz segítségével okosautó lesz, és így az abból nyert adatok bekerülnek a többi okoseszköz által gyűjtött adathalmaz mellé, valamint összehasonlíthatóvá és elemezhetővé válnak [2].

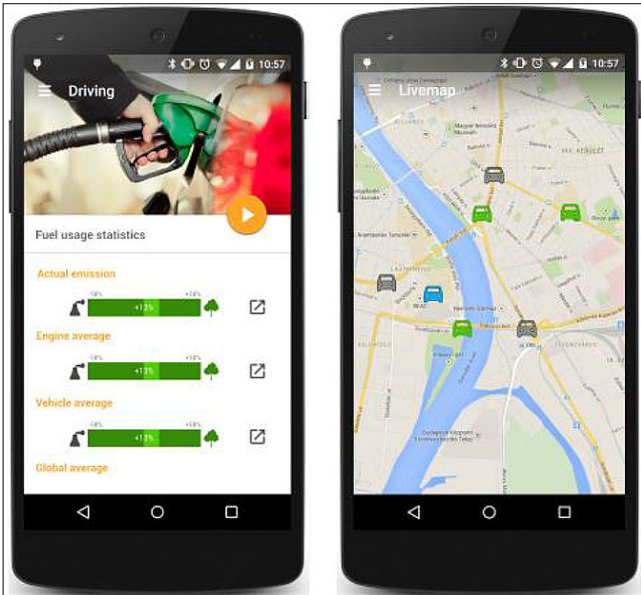
A VehicleICT rendszer a SensorHUB koncepciónak és keretrendszernek a járműipari szakterületre történő alkalmazása. Ezt a rendszert számos alkalmazás felhasználja, mint például a Social Driving és a nemrég elkészült ObdCanCompare alkalmazás.

A rendszer foglalkozik a járművekben lévő adatok kinyerésével, sorosításával, származtatott értékek számításával, kliens oldali megjelenítéssel, szerver oldali feltöltéssel, valamint ezen adatok elemzésével és további felhasználásával (5. ábra) [11]. A felhasználó eszközén futó szolgáltatás végzi a fenti feladatokat. Maga a szolgáltatás nem rendelkezik felhasználói felülettel, egy alkalmazás részeként, osztálykönyvtárként működik.

4. Social Driving alkalmazás

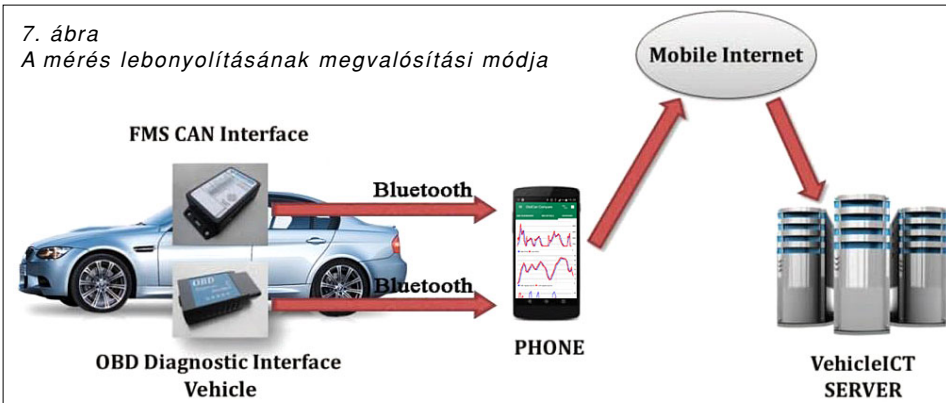
A Social Driving alkalmazás felületén a járműből nyert adatok segítségével össze tudjuk kapcsolni a rendszer felhasználóit, így összehasonlítható lesz vezetési stílusuk, és ha hasonló autót vezetnek, akkor elemezni tudjuk, milyen eltérések mutatkoznak a fogyasztásban a különböző vezetési stílusok eredményeként (6. ábra) [12].

Az adatok felhasználása számos üzleti lehetőséget is rejt. Például, jutalmazni tudjuk azon vezetőket, akiknek autója adott idő alatt a legkevesebb káros anyagot bocsátja ki; ők a töltőállomásokon akár kedvezményt

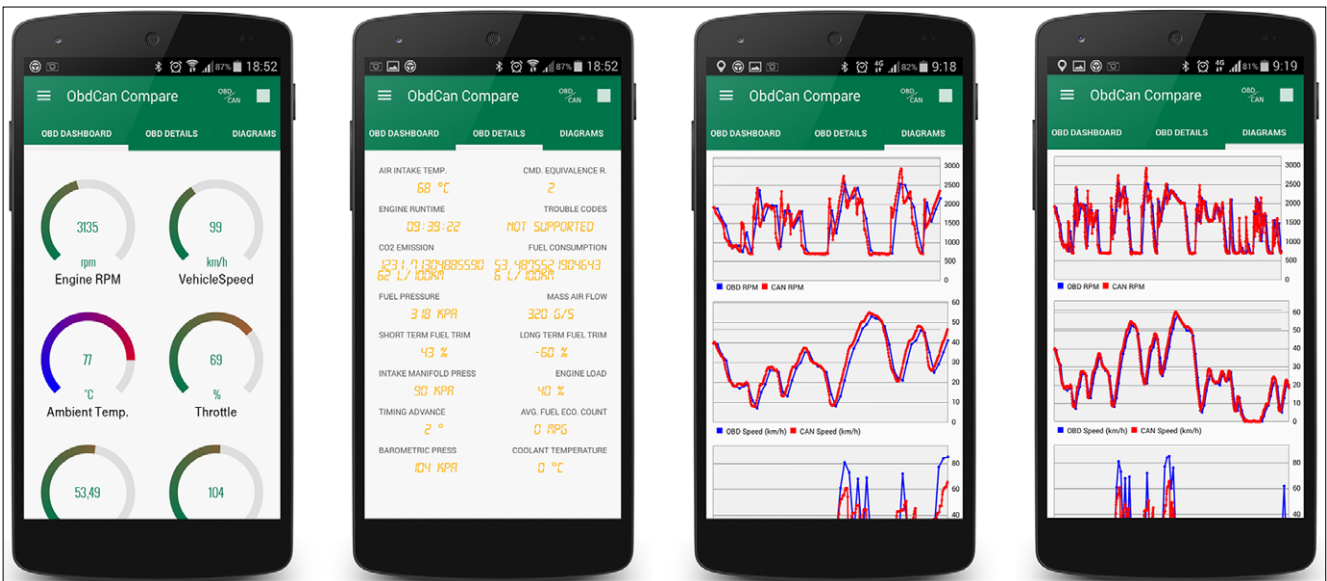


6. ábra A Social Driving alkalmazás felhasználói felülete

is kaphatnának tankoláskor. A vezetőknek arra is van lehetőségük, hogy utánanézzenek, hogy a hasonló autók milyen fogyasztási értékekkel rendelkeznek, vagy milyen további menettulajdonságok jellemzik őket.



8. ábra Az alkalmazás képernyői (Dashboard, Details, Charts)



A hibakódok kiolvasásával és valós idejű értesítéssel, a szervizhálózatok remekül használhatnák ezeket az információkat. Jelentős hiba esetén azonnal értesíteni lehet majd az autó vezetőjét, sőt akár online motor-diagnosztikát is elvégezhetnek, hiszen rendelkeznek a szükséges adatokkal, így már felkészülten fogadhatnak az autóst a szervizben, és akár személyre szabott ajánlattal is megkereshetik a tulajdonosokat. Ha összekapcsoljuk ezt a rendszert az okos város megoldásokkal, akkor valós időben jelezhető az autósoknak, ha már felesleges a gázba taposnia, hiszen a kereszteződési lámpa mindjárt pirosra fog váltani [13].

5. ObdCanCompare alkalmazás

Az ObdCanCompare az előző fejezetekben bemutatott OBD-II és CAN bus mérést egyszerre megvalósító és összehasonlítását biztosító, a SensorHUB keretrendszer felhasználó, Androidra készült alkalmazás (7. ábra).

Az alkalmazásban először a mérések tulajdonságait állíthatjuk be (a Bluetooth engedélyezése után): a két Bluetooth-os eszközt, az üzemanyag típusát, a motor térfogatát és hatásfokát. Ezután indíthatjuk el a mérést. Ekkor mind a Dashboard, mind a Details, mind a Diag-

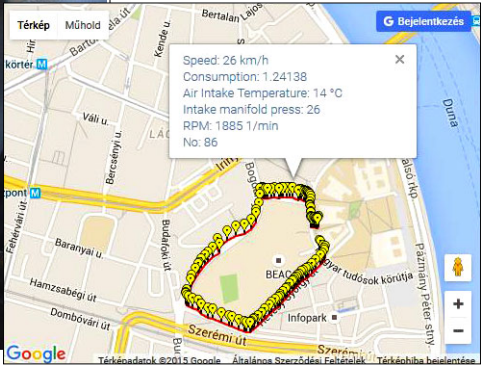
rams fülken olvashatóak az adatok. Az OBD és CAN mérés adatainak megjelenítés között az OBD/CAN gombbal tudunk váltani (8. ábra).

Mérés közben, ha van netkapcsolat, akkor a mért adatok feltöltésre kerülnek a SensorHUB rendszerbe (9. ábra). Továbbá, ha a helymeghatározás is engedélyezve van, akkor az adatcsomagokba bekerülnek a mérés közben aktuális földrajzi koordináták is.



9. ábra
Tesztelés

10. ábra
Interaktív térkép megjelenítés
a Pentaho reportban



6. Big Data a szerver oldalon

A szerveroldal alapja az elosztott fájlrendszer, a Cloudera Hadoop rendszere, melynek Data Lake-jében gyűlhetnek a telefon szenzorjaiból az adatok. Az adatak feldolgozásához és elemzéséhez a Pentaho Data Integration és Pentaho Report Designer alkalmazásokat használjuk. A Pentaho egy nyílt forráskódú, Java alapú, Big Data rendszerekkel összeköthető programcsomag [14].

A Data Integration különböző forrásokból származó adatokat tud egységesen kezelni. ETL (Extract-Transform-Load) transzformációkat állíthatunk össze benne a Spoon eszköz grafikus felületén. Ez a rész az adatfolyam manipulálásáról szól, ekkor kell kiválasztani azokat a rekordokat és attribútumokat, melyekre az elemzés során szükségünk lesz [15].

Az így elkészült adattáblát a Report Designer-ben tudjuk felhasználni különböző táblázatok, grafikonok, diagramok és további elemzések, összehasonlítások adatforrásaként. Ennek kimenetei különböző formátu-

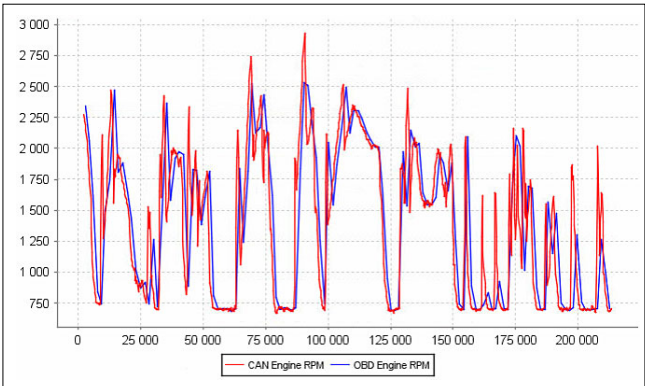
mák lehetnek: PDF, HTML, Excel, RTF, Text, CSV. Továbbá a struktúrák feltölthetők az üzleti intelligencia szerverre, ami generálni tudja a megfelelő formátumú riportot.

A térképes megjelenítés nem a Report Designer-be került beépítésre, hanem egy a riportba belekódolt Java Script kód hozza létre (10. ábra), mely a Google Maps API adatformátumát rendeli össze a lekérdezés eredményének soraival.

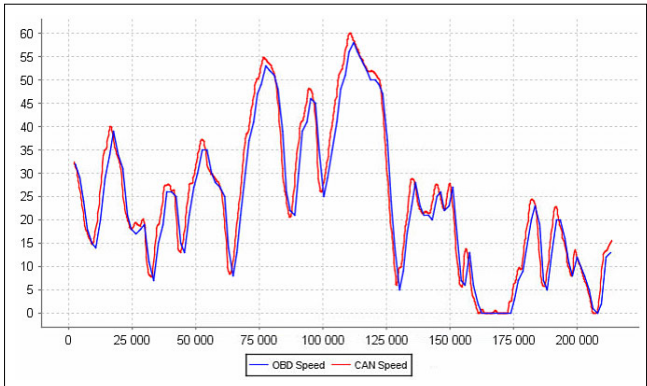
Az előző mobilképernyős, valamint a 11., 12. és 13. ábrákból kitűnik, hogy mennyire hasonlítanak az ObdCanCompare-ben látható diagramok, a Pentaho segítségével készült diagramokhoz. A két megjelenítési forma konzisztens. Az ObdCanCompare-es az azonnali megjelenítéshez hasznos, míg a Pentaho-s a későbbi elemzésekhez szolgál.

A Hadoop klaszter online HUE felülete számos adatvizualizációs lehetőséget biztosít [16]. Az adatfolyam összeállítására az Oozie eszközt használtuk, amiben folyamatábraként állítottuk össze az ETL folyamatot [17]. Mindezt tudjuk ütemezetten végrehajtani.

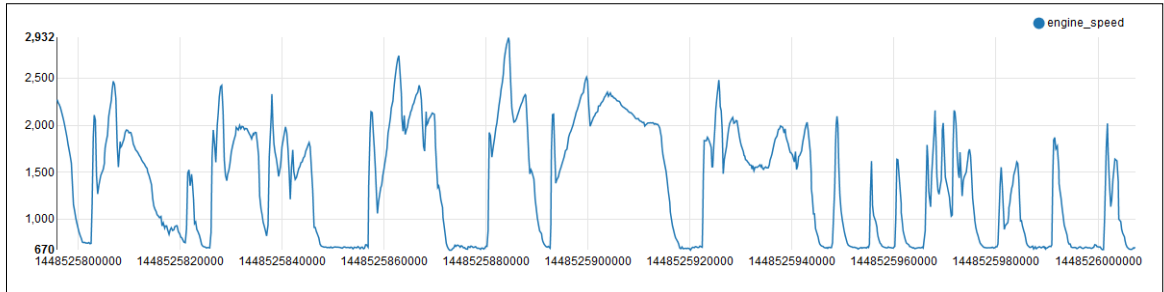
11. ábra
Fordulatszám összehasonlítása a Pentaho reportban



12. ábra
Sebesség összehasonlítása a Pentaho reportban



13. ábra
Fordulatszám
a HUE felület
diagramján



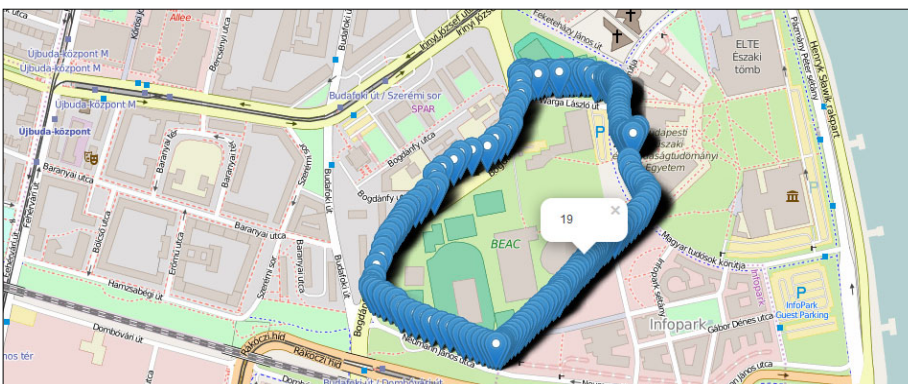
A térképes megjelenítés az online felületen került összeállításra. Itt beállítható, hogy a térképen kattintva mely adatok jelenjenek meg az útvonal adott pontjához. Végül az eredmény megjelenik egy OpenStreetMap alapú térképen (14. ábra).

7. Összefoglalás

Az elkészült alkalmazás, riportok és diagramok alapján látható, hogy az OBD-II mérés és a CAN bus mérés eredményei számos jellemzőben megegyeznek egymással. Ilyen például a fordulatszám és a sebesség értéke. Ezen esetekben apró késleltetés látható az OBD-s adatokban a CAN bus adataihoz képest. Ugyanakkor más, meghatározóan a származtatott (aggregát) értékek már eltéréseket mutattak. Itt az eltérés oka, hogy különböző összetett formulák léteznek például a fogyasztás vagy a CO₂-kibocsátás meghatározására, amikből eltérő eredmények születhetnek az OBD-s és CAN bus-os mérés összehasonlításában.

A fentiek alapján elmondható, hogy az alapvető okosautó-alapú megoldásokhoz elegendő az OBD-II port használata, például a Social Driving alkalmazás esetében, de fontos, hogy a származtatott értékek formulái mindenütt egységesen kerüljenek definiálásra. Amennyiben pedig professzionális megoldásra van szükségünk, például flottamenedzsment esetén, akkor a CAN bus-alapú megoldás a javasolt választás.

A bemutatott mérések, az adatgyűjtés és a következtetések egyszerű, a működést szemléltető példák. Célnk a jövőben, hogy a beérkező adathalmaz alapján összetettebb üzleti intelligencia módszerekkel további érdekes és hasznos összefüggéseket nyerjünk ki, valamint a tendenciák alapján a jövőbeli értékeket jelezzük előre. Ezekkel az adatokkal akár a gépjármű vezetőjét is tudjuk értesítésekkel támogatni, figyelmét felhívni.



14. ábra
HUE Map a sebesség kijelzésével

Köszönetnyilvánítás

A kutatás folytatása és a publikáció az Emberi Erőforrások Minisztériuma ÚNKP-16-2-I. kódszámú Új Nemzeti Kiválóság Programjának támogatásával, az Emberi Erőforrások Minisztériuma ÚNKP-16-4-III. kódszámú Új Nemzeti Kiválóság Programjának támogatásával, valamint a Magyar Tudományos Akadémia Bolyai János Kutatási Ösztöndíj támogatásával készült.

A szerzőkről



SIK DÁVID 2016-ban szerezte meg mérnökinformatikus BSc diplomáját a Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Karán, majd folytatta tanulmányait az MSc fokozatért alkalmazott informatika fősPECIALIZÁCIÓN és mobilsoftver-fejlesztés melléKSPECIALIZÁCIÓN az Automatizálási és Alkalmazott Informatikai Tanszéken. Kutatási témája: Big Data alapú elemzési és előrejelzési lehetőségek vizsgálata.



EKLER PÉTER 2007-ben szerezte mérnök informatikus diplomáját a BME Automatizálási és Alkalmazott Informatikai Tanszéken. PhD fokozatát 2011-ben szerezte, majd ezt követően adjunktusként folytatta oktatói, kutatói és fejlesztői munkáját az egyetemen. A kutatási és fejlesztői munkák mellett jelentős szerepet vállalt az oktatásban és több tantárgy kidolgozásában. Mobil softverfejlesztés területén két könyv társszerzője és szerkesztője: Bevezetés a mobilprogramozásba, Android-alapú softverfejlesztés. Rendszeres résztvevője és feladatvállalója a témákhoz kapcsolódó hazai és nemzetközi konferenciáknak és projekteknek. Jelentősebb kitüntetések: Bolyai János kutatói ösztöndíj (2016-), NJSZT Kemény János-díj (2015), BME Innovációs Díj (2015), Műegyetem kiváló oktatója díj (2015), Év Informatika Oktatója (2014).



LENGYEL LÁSZLÓ 2006-ban kapta meg PhD oklevelét. Egyetemi docensként dolgozik a BME Villamosmérnöki és Informatikai Kar, Automatizálási és Alkalmazott Informatikai Tanszéken. Kutatási területei: softvermodellezés, szakterület-specifikus modellezés, metamodellezés, modellalapú fejlesztés, validált modelltranszformáció, IoT rendszerek és felhő-alapú módszerek. Számos nemzetközi konferencián és folyóiratban megjelent publikáció szerzője. Jelentősebb kitüntetések: Bolyai János kutatói ösztöndíj (2007–2010 és 2015–2018), Siemens Excellence Award (2008), és NJSZT Kemény János-díj (2012), BME Innovációs Díj (A SensorHUB koncepció és keretrendszer) (2015).

Irodalomjegyzék

- [1] Charaf Hassan: Az infoszféra tudást közvetítő szerepe a mai társadalomban,
<http://www.matud.iif.hu/2015/02/03.htm>
(2016. október).
- [2] SensorHUB keretrendszer,
<https://www.aut.bme.hu/Pages/Research/SensorHUB>
(2016. október).
- [3] VehicleICT keretrendszer,
<https://www.aut.bme.hu/Pages/Research/vehicleict>
(2016. október).
- [4] Lakatos I., Nagyszokolyai I.: Gépjárművek üzeme I., Typotex Kiadó, 2012.
http://www.tankonyvtar.hu/hu/tartalom/tamop412A/0018_Gepjarmuvek_uzeme_1/adatok.html
(2016. október).
- [5] Menedzsment és motordiagnosztika,
<http://www.geree.hu/motormenedzsment-es-diagnosztika/>
(2016. október).
- [6] ELM327,
<http://www.elmelectronics.com/obdic.html#ELM327>
(2016. október).
- [7] Fodor Dénes, Szalay Zsolt: Autóipari kommunikációs rendszerek,
http://moodle.autolab.uni-pannon.hu/Mecha_tananyag/autoipari_kommunikacios_rendszerek/ch03.html
(2016. november).
- [8] CAN bus communication,
<http://fmsgateway.com/glossary/can-bus-communication>
(2016. november).
- [9] László Lengyel, Péter Ekler, Tamás Ujj, Tamás Balogh and Hassan Charaf: SensorHUB: An IoT Driver Framework for Supporting Sensor Networks and Data Analysis, Intern. Journal of Distributed Sensor Networks, 2015.
<http://www.hindawi.com/journals/ijdsn/2015/454379/>
(2016. október).
- [10] Innovációs díjban részesült a SensorHUB
<https://www.vik.bme.hu/hir/874-bme-innovacios-dijban-reszesult-a-sensorhub-fejlesztes>
(2016. október).
- [11] Jereb László, Lengyel László: Jármű ICT fejlesztési irányok és kihívások, Híradástechnika, HTE Infokom 2014 – LXIX. évfolyam, 2014.
- [12] László Lengyel, Péter Ekler, Tamás Ujj, Tamás Balogh and Hassan Charaf: Social Driving in Connected Car Environment, European Wireless 2015.
- [13] A SensorHUB,
Mérnök újság, XXII. évf. 10. szám, 2015. október.
- [14] Pentaho Community,
<http://community.pentaho.com/>
(2016. november).
- [15] Dudás Ákos, Ekler Péter, Tömösvári Imre: Üzleti intelligencia tantárgyi jegyzetek,
<https://www.aut.bme.hu/Course/BMEVIAUMA02>
(2016. november).
- [16] HUE – The Hadoop UI,
<http://gethue.com/>
(2016. november).
- [17] Oozie Workflow Scheduler,
<http://oozie.apache.org/>
(2016. november).

