UNIVERZITET SINGIDUNUM

Departman za poslediplomske studije

Predikcija ispravnog očitavanja senzora mase protoka vazduha motora automobila

- MASTER RAD -

Mentori:

Prof. dr. Marina Marjanović Jakovljević

Prof. dr. Miodrag Živković

Kandidat:

Dejan Čugalj (2019/430054)

Beograd, 2021.

Sadržaj

1. Uvod ..... - 1 -

1.1 Predmet i značaj istraživanja ..... - 2 -

1.2 Cilj istraživanja ..... - 2 -

1.3 Hipoteze istraživanja ..... - 4 -

1.4 Metode istraživanja ..... - 4 -

1.5 Struktura master rada ..... - 5 -

2. Teorijske osnove komunikacione mreže automobila ..... - 6 -

2.1 OBD II standard (*SAE J1979*) ..... - 6 -

2.2 CAN protokol (*ISO 11898*) ..... - 8 -

2.3 Poruke CAN protokola ..... - 9 -

2.3.1 Poruka sa podacima (*Data frame*) ..... - 9 -

2.3.2 Poruka sa zahtevom za podatke (*Remote frame*) ..... - 10 -

2.3.3 Poruka o grešci (*Error frame*) ..... - 10 -

2.3.4 Poruka o zauzetosti (*Overload frame*) ..... - 10 -

2.4 Struktura poruke sa „*zahtevom za podatke*“ (*Data frame*) ..... - 11 -

2.5 Konverzija izvornih podataka „*poruka sa zahtevom za podatke*“ ..... - 13 -

3. Senzor mase protoka vazduha (*MAF Senzor*) ..... - 15 -

3.1 Tipovi MAF senzora ..... - 16 -

3.2 Princip rada MAF senzora ..... - 17 -

3.3 Teorijske osnove merenja MAF senzora ..... - 18 -

4. Topologija ciklusa predikcije ispravnosti očitavanja MAF senzora ..... - 20 -

4.1 Multifunkcionalan uređaj „*Infinity SCBA*“ ..... - 21 -

4.1.1 Hardverska osnova sa operativnim sistemom ..... - 22 -

4.1.2 CAN kontroler sa RS485 serijskim protokolom ..... - 22 -

4.1.3 3G/LTE Internet konekcija ..... - 23 -

4.1.4 Softver za komunikaciju sa CAN-*bus* mrežom ..... - 23 -

4.2 Uslovi generisanja i uvid u strukturu skupa podataka ..... - 25 -

4.2.1 Struktura skupa podataka (*dataset*) ..... - 26 -

4.3 Izbor metoda mašinskog učenja ..... - 28 -

4.3.1 Prosta Linearna Regresija (*PLR*) ..... - 29 -

4.3.2 Višestruka Linearna Regresija (*VLR*) ..... - 31 -

4.3.3 „*Random Forest*“ ansambli (*RF*) ..... - 34 -

4.3.4 Pregled dobijenih rezultata i njihova uporedna analiza ..... - 37 -

4.4 Praktična implementacija predikcije ispravnosti MAF senzora ..... - 38 -

5. Zaključna razmatranja ..... - 40 -

6. Literatura ..... - 42 -

7. Dodatak I

Listing 1 I

Listing 2 I

Listing 3 II

Listing 4 III

Listing 5 IV

Listing 6 V

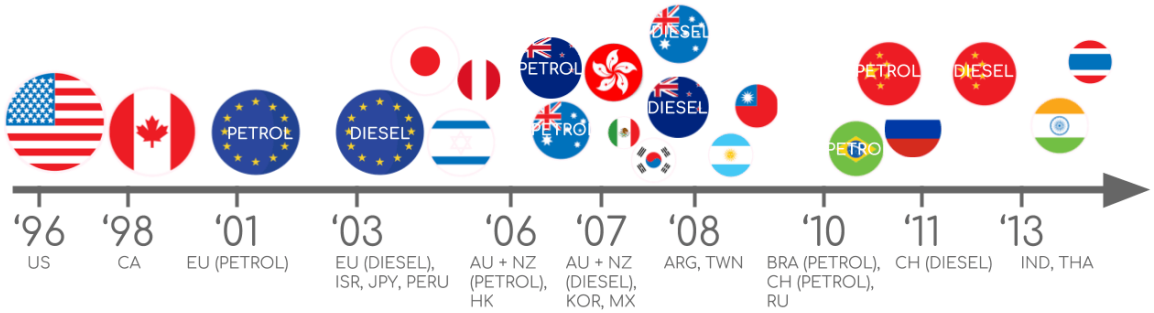
Listing 7 VI

1. Uvod

Usled klimatskih promena efektom staklene bašte, agencije za zaštitu životne sredine ***EPA*** (eng. *U.S **E**nvironmental **P**rotection **A**gency* [1]) i ***CARB*** (eng. ***C**alifornia **A**ir **R**esources **B**oard* [2]), direktivama vlade su dobile mogućnost primene standardizovanih protokola po kojima se prati emisija izduvnih gasova automobila. Tokom veka trajanja automobila, vlasnik je dužan da održava emisiju izduvnih gasova u propisanim parametrima, dok su proizvođači u obavezi da omoguće potrebnu infrastrukturu preko koje se dobija uvid u emisiju ispuštanja štetnih gasova u atmosferu.

Elektronski nadzor i dijagnostika rada motora sa unutrašnjim sagorevanjem automobila i lakih teretnih vozila, predstavljeni su krajem 1970-tih, dok je već početkom 1980-tih godina počeo da se primenjuje ***OBD*** (eng. ***O**n-**B**oard **D**iagnostics*) komunikacioni standard koji daje uvid o ispunjenosti zahteva *EPA* i *CARB* kontrole emisionih standarda. Tokom narednih godina, dijagnostički sistemi postaju sve sofisticiraniji, pa sredinom 1990-tih godina OBD standard dobija svoju nadogradnju pod nazivom OBD II standard za koji su nadležna i odgovorna Udruženje svetskih automobilskih inženjera: ***SAE*** (eng. ***S**ociety of **A**utomotive **E**ngineer* [3]) i Međunarodno udruženje za standardizaciju: ***ISO*** (eng. ***I**nternational **S**tandardization **O**rganization* [4]). Ovaj novi standard pruža gotovo potpunu dijagnostičku kontrolu vozila i emisionih parametara ali isto tako i nadgledanje ostalih sistema kao što su: delovi šasije, dodatne opreme, elektronskog sistema proklizavanja vozila itd. Proizvođači na američkom tržištu automobila su bili u obavezi da implementiraju OBD II standard počevši od januara 1996. godine, dok je evropski ekvivalent EOBD, počeo sa primenom 2001. godine kod svih evropskih automobila [Slika 1].

Automobilska industrija, implementacijom OBD II standarda, a u svrhu kontrolisanja rada svih sistema i podsistema automobila, imala je obavezu da uvede standardizaciju, protokole, po kojima računari komuniciraju između sebe, bez potrebe da se uvodi glavni nod (eng. *Master Host*) kao kontrolna tačka u razmeni informacija. Ta komunikacija, razmena binarnih informacija na krajevima, je inspirisala „*Robert Boš*“ kompaniju (*Robert Bosch GmbH* [5]) da počne razvijanje KAN (u daljem tekstu: CAN) protokola 1983. godine, a koji je zvanično objavljen 1986. godine. na konferenciji svetskih automobilskih inženjera (***SAE***) i već 1991. godine je počela eksperimentalna implementacija CAN protokola u „*Daimler Mercedes-Benz*“ automobilima. Nedugo zatim, prihvaćanje ovog protokola je našlo veliki odziv, drugim rečima, svi proizvođači automobila su bili u obavezi da implementiraju neki od protokola OBD II standarda.



Slika 1. Vremenski tok implementacije OBD2 standarda.
izvor: https://www.csselectronics.com/

1.1 Predmet i značaj istraživanja

Svetska regulatorna tela za kontrolu emisije štetnih gasova, kao i međunarodna organizacija za standarde, uvideli su način smanjenja emisije štetnih gasova koja ispuštaju automobili, dok se sama implementacija kontrole realizovala instrumentima informaciono komunikacionih tehnologija kroz prizmu OBD II standarda.

Kako OBD II standard u svojoj „paleti“ poseduje nekoliko protokola po kojima je standardizovana komunikacija raznih komponenti automobila, primarni predmet istraživanja ovog rada se bazira na ekstrakciji i analizi informacija dobijenih iz CAN-*bus* mreže koja implementira CAN protokol, a sve u svrhu predikcije ispravnosti senzora protoka mase vazduha motora automobila.

Komponente sistema prečišćavanja izduvnih gasova automobila su u tesnoj sprezi jedna sa drugom i ukoliko samo jedna „karika“ zakaže u celom sistemu, može da se kaže da je sistem kontrole izduvnih gasova u kvaru i da kao takav ne zadovoljava Evropske standarde kontrole emisije izduvnih gasova.

Kako je sama priroda komponenti za kontrolu štetnih gasova u tesnoj sprezi jedna sa drugom, isto tako i veoma utiče na ispravan rad motora automobila, što može da dovede na samom kraju i do potpunog zakazivanja motora. Uvid i analiza podataka senzora mase protoka vazduha motora automobila, metodama i alatima mašinskog učenja dobijenih iz CAN-*bus* mreže, ima primaran značaj za rano otkrivanje kvara u sistemu za prečišćavanje izduvnih gasova. Takođe, daje uvid u ispravan rad motora.

Predikcija i analiza ispravnih očitavanja senzora protoka vazduha dobija na značaju, jer ukoliko se rano utvrde neregularnosti očitavanja istog i preventivno se preduzmu akcije defektaže[1], može da se spreči ispuštanje štetnih gasova u atmosferu ali isto tako daje korisniku automobila sigurnost u budući ispravan rad motora svog vozila.

1.2 Cilj istraživanja

Pretpostavimo situaciju u kojoj je potrebno proveriti automobil pre polaska na dalji put, situaciju u kojoj se kvarovi pojavljuju ali isto tako i misteriozno nestaju, situaciju gde je rešavanje problema sa vozilom jedino moguće uvidom u podatke koji su dobijeni očitavanjem senzora, rutinska provera ispravnosti automobila (uvid u neregularnost podataka), „*real-time*“ monitoring stanja automobila, iznenadan dim koji se pojavljuje u izduvnim gasovima, priprema za godišnji tehnički pregled automobila itd.

Pomenuta problematična i preventivna scenarija koja mogu da se dese na vozilu su istovremeno i izazovi na koje može da se deluje iz aspekta uvida u neregularnost očitavanja senzora protoka vazduha, jer je za ispravan rad motora potrebno da svaka komponenta šalje podatke koji su u okvirima ispravnosti, a koje je proizvođač propisao u tehničkoj specifikaciji automobila. Ukoliko dođe do zakazivanja u sistemu prečišćavanja ali isto tako i u sistemu usisa vazduha motora, jasno se pojavljuju neregularnosti podataka u očitavanju senzora mase protoka vazduha, koji će detaljnije biti opisan u radu.

Sagledavajući tržište koje se bavi prodajom specijalizovanih dijagnostičkih uređaja pomoću kojih se dobija uvid u ispravnost automobila, kako profesionalnih, tako i amaterskih, predstavlja sisteme koji omogućavaju samo uvid u greške koje su se desile u toku veka trajanja vozila i kao takve su uskladištene u memoriju nekih od računara automobila.

[1] Defektaža – *Detekcija i ustanovljavanje uzroka kvara.*

Servisni tehničar uvidom u izveštaj, koji daje dijagnostički uređaj, može da locira delove sistema ili podsistema vozila na kojima se desio kvar. Ovo je tipičan scenario prilikom defektaže kvara, a koji primenjuju servisni tehničari u današnje vreme. Ovaj vid otklanjanja kvarova se još naziva i „reaktivno održavanje“, koje se u svojoj praktičnoj primeni obavlja tek kada se kvar na vozilu desi. Najznačajnije strategije održavanja koje mogu da se izdvoje su:

- Reaktivno održavanje (eng. *Reactive Maintenance*).
- Preventivno održavanje (eng. *Preventive Maintenance*).
- Prediktivno održavanje (eng. *Predictive Maintenance*).

Preventivno održavanje se može opisati postupcima i radnjama koje se obavljaju na automobilu kao preventiva pojave kvarova. To može da se opiše radnjama kao što su: zamena motornog ulja, zamene raznih filtera, rutinski pregled vozila, itd.[2]

Osnovni cilj ovog istraživanja, je rano otkrivanje neregularnosti u očitavanju sistema za protok vazduha motora u realnom vremenu i konstantan monitoring rada subjektnog vozila, spregom i upotrebom istreniranih modela, alatima mašinskog učenja veštačke inteligencije uz pomoć „*Infinity*“ [3] uređaja. Ovaj vid predikcije neregularnosti u očitavanju senzora, spada u klasu strategije „prediktivnog održavanja“.

Iako je koncept „prediktivnog održavanja“ prisutan već dugi niz godina, primena u praksi retko može da se vidi, dok se modeli „prediktivnog održavanja“, koji su koncipirani na naučnim istraživanjima, mogu pronaći, a njihov diverzitet se ogleda u raznim teorijskim modelima mašinskog učenja.

Neki od modela se može videti u [6], koji u svojoj osnovi implementira neuronske mreže nad simulacionim modelima sistema usisa vazduha motora automobila. Ovakav model predikcije nije pogodan za ovo istraživanje iz razloga jer neuronske mreže zahtevaju veće računarske kapacitete, koje nisu bile dostupne u „*Infinity*“ uređaju, napravljenim za ovaj istraživački rad.

Takođe, još jedan vid prediktivnog održavanja se može videti u [7], kao vid predikcije ispravnog očitavanja „O_2“ senzora u svrhu određivanja da li senzor ima neki vid začepljenja.

Još jedan od predloženih modela kao u [8], se oslanja princip udaljenog monitoringa vozila, koji iako ima zanimljiv pristup, jedna od najvećih mana je što se prilikom monitoringa koristi „*ELM 327 Bluetooth*“ mikrokontroler, koji po svojoj arhitekturi nikako ne može da zadovolji brzinu koja je potrebna za prikupljanje podataka sa mreže kojoj je brzina definisana sa 1 Mbit/s. Kao takav nije pogodan za očitavanje vrednosti u realnom vremenu.

Primer predikcije ispravnosti kompresora na „*Volvo*“ kamionima je dat u [9], koji se oslanja na dve baze podataka: evidentirani podaci o vozilu (eng. ***L**ogged **V**ehicle **D**ata(**LVD**)*) i bazi podataka o evidenciji servisa vozila (eng. ***V**ehicle **S**ervice **R**ecords* (***VSR***)) u sprezi sa algoritmom klasifikacije prema pravilima (eng. *Rule-based classification algorithm*).

Prediktivno održavanje je predstavljeno i u [10] koje svoje informacione resurse uzima iz predefinisanih baza podataka koje su generisane u pet različitih auto servisa u periodu od 24 meseca. Predstavljena metoda mašinskog učenja je hijerarhijski modifikovana SVM[4] (eng. *hierarchical modified fuzzy support vector machine*). Korišćeni modeli mašinskog učenja su implementirani na voznom parku kompanije za izdavanje vozila.

---

[2] Preventivno održavanje se u praksi još naziva i „Mali servis“, „Veliki servis“.
[3] Infinity uređaj – *Hardver koji je konstruisan i napravljen za prikupljanje podataka iz CAN-bus mreže.*
[4] SVM – *Support Vector Machine.*

1.3 Hipoteze istraživanja

Hipoteze koje bi za ovo istraživanje činile sveobuhvatnu opštu celinu, a poštujući zadate uslove opisane u [4.2], mogu da se definišu kao:

Osnovna hipoteza

H^0: *Induktivnim učenjem kroz aposteriorno dobijeno znanje analizom prikupljenih podatka iz CAN-bus mreže, može da se utvrdi kvar u sistemu prečišćavanja ispuštenih gasova vozila ali isto tako i uvid u ispravnost motora automobila.*

Pomoćne hipoteze

H^1: *Analizom dobijenih podataka se proverava sistem usisa u motoru automobila.*

H^2: *Neregularnost u prikupljenim podacima može da ukaže na ispravnost turbine motora.*

H^3: *Uviđena odstupanja mogu da ukažu na skoro zakazivanje katalizatora / DPF filtera.*

H^4: *Uviđena odstupanja mogu da ukažu na skoro zakazivanje O_2 senzora (lambda sonda).*

1.4 Metode istraživanja

Priroda podataka koja se koristi kroz istraživanje je regresionog tipa i to u vidu vremenskih serija koje kroz sisteme induktivnog učenja na osnovu dobijenih primera vrši predikciju ispravnosti senzora protoka vazduha motora automobila.

Obzirom na spomenutu prirodu podataka, a uzimajući u obzir i kompleksnost teme, u ovom istraživanju su se primenile različite metode istraživanja. Da bi se zadovoljili opšti postulati pouzdanosti, objektivnosti, sistematičnosti, ovo istraživanje je oblikovano osnovama opštih naučnih metoda prikupljanja podataka.

Induktivni modeli posmatranja i deduktivne analize dobijenih podataka iz CAN-*bus* mreže, iskorišćena su u svrhu predikcije ispravnosti senzora ali i kao opšte naučne metode. Prikupljeni podaci su dobijeni sledećim metodama:

- Komparativnom metodom.
- Statističkom metodom.
- Analitičkom metodom.
- Deskriptivnom metodom.

Dobijeni podaci koji su prikupljeni u toku istraživanja će biti obrađeni:

- Kvantitativno: *Odgovarajućim statističko-matematičkim metodama u sprezi alata veštačke inteligencije.*
- Kvalitativno: *Metodama aposteriornih saznanja, sinteze, analize i indukcije.*

Osnovni tok izrade master teze je pokrenut u momentu kada je definisana osnovna hipoteza, u korelaciji sa pomoćnim hipotezama kroz prizmu teorijskih, praktičnih i empirijskih saznanja. Induktivnim metodama učenja se dobio opšti uvid u dobijene podatke, dok se deduktivnom metodom došlo do krajnjih informacija, kao rezultat upotrebe alata mašinskog učenja.

1.5 Struktura master rada

Master teza, pod imenom „*Predikcija ispravnog očitavanja senzora mase protoka vazduha motora automobila*“, je organizovana u sedam poglavlja, od kojih se u prvih četiri poglavlja opisuje detaljno istraživački rad, u petom se daju zaključna razmatranja, u šestom poglavlju je prikazana stručna literatura koja je korišćena, a u dodatku se predstavljaju programerski kodovi koji su implementirani u raznim aspektima ovog istraživačkog rada.

U uvodnom poglavlju je predstavljen pregled inicijalne motivacije za izradu ove master teze. Predstavljen je cilj istraživanja kao i osnovne smernice, hipoteze i alati korišćeni tokom istraživanja.

U drugom poglavlju se detaljno opisuje komunikaciona mreža u automobilima kao i struktura samih poruka koje su upotrebljavane prilikom generisanja skupa podataka. Predstavljeni su različiti tipovi poruka i naglašene one koje su bile najznačajnije za ovo istraživanje. Prikazane su metode konverzija „sirovih“ poruka iz heksadecimalnog u decimalan brojni sistem i opisi metoda matematičke konverzije u metrički SI sistem.

U trećem poglavlju se daje pregled subjektnog senzora nad kojim su obavljane analitičke, matematičke i predikcione radnje. Opisuju se tipovi senzora mase protoka vazduha motora automobila (MAF) i analizira se princip rada kroz prizmu teorijskih osnova merenja.

Četvrto poglavlje predstavlja topologiju ciklusa istraživanja kroz četiri koraka koja su detaljno opisani u narednim tačkama.

1. Prvi korak opisuje komunikacioni sistem koji je korišćen u radu sa opisima hardverskih komponenti koje su to omogućile.
2. Predstavljeni su uslovi pod kojima se generiše skup podataka kao i njihova struktura, kao drugi korak.
3. U nastavku, kao treći korak, detaljno je opisan postupak predikcije ispravnosti očitavanja MAF senzora upotrebom alata mašinskog učenja veštačke inteligencije. Vršene su evaluacije nad dobijenim podacima, kao i uporedna analiza rezultata. Statističkim metodama su merene greške prilikom predikcije i analize uporednih tačnosti dobijenih vrednosti.
4. U četvrtom koraku je prikazana praktična implementacija istraživačkog rada, kroz grafičke prikaze predikcionih vrednosti. Predstavljena je baza podataka namenjena vremenskim serijama i grafičko okruženje za prikaz analiziranih vrednosti.

U petom poglavlju se daje zaključno razmatranje kao i izvedeni zaključci kojima bi se definisao doprinos ove master teze. Takođe, dat je uvid u poboljšanja i preporuke koje bi bile od značaja budućim istraživanjima na ovu temu.

Šestim poglavljem se daje uvid u stručnu literaturu koja je korišćena tokom istraživačkog rada.

Konačno, u sedmom poglavlju, dodatku, se daju programerski kodovi koji su korišćeni prilikom razvoja hardverskih komponenti, i svi kodovi koji su upotrebljavani prilikom konstruisanja i evaluacija metoda lineranih regresija.

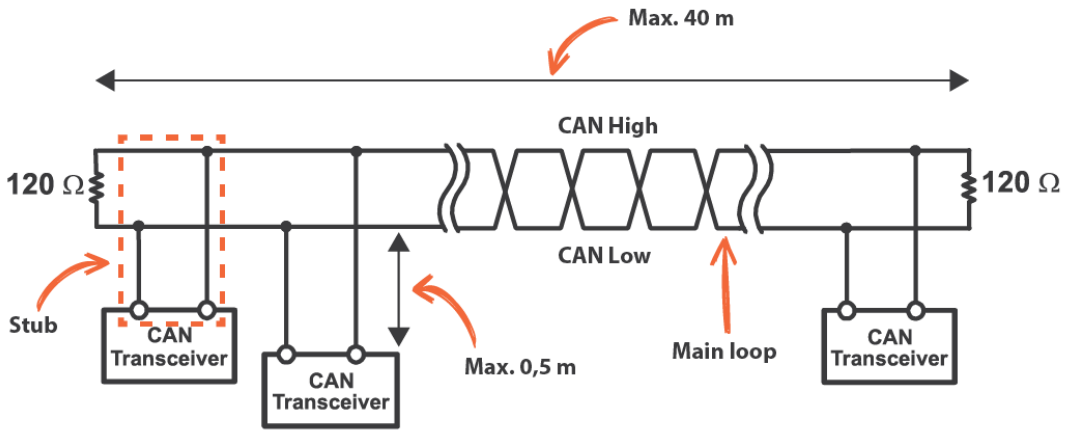
2. Teorijske osnove komunikacione mreže automobila

Komunikacija između mašina se realizuje preko električnih impulsa pretvorenih u binarnu strukturu koja se na kraju nodeva/računara interpretira u obliku informacija koju algoritmi koriste kao parametre po kojima izvršavaju zadatke. Obično, oblici po kojima se šalju i primaju binarne informacije su regulisani protokolima koji nama, ljudskoj vrsti u mnogome olakšavaju konstrukciju i razumevanje istih.

Komunikaciona mreža u automobilima prvenstveno opskrbljuje informacijama računare, koji međusobno interaguju u razmeni svojih trenutnih statusa, stanja, i upravo ta interakcija govori o kompleksnosti sistema koji kontrolišu razne aktuatore[5], senzore i ostale komponente neophodne za ispravna rad svih sistema i podsistema vozila.

Kompleksnost neizmerno vodi do grešaka, koje u automobilskoj industriji moraju da se svedu na minimum, a taj minimum koji u nekim situacijama, kada je reč o automobilskoj industriji, znači i spasavanje ljudskog života, mora biti u okvirima „statističke greške“. Imajući uvid u ozbiljnost ovakve situacije, sistem po kojem se odvija komunikacija, mora biti robustan, otporan na sve vidove grešaka, i na kraju, implementacija mora biti jeftina.

Sve ove zahteve je ispunio OBD II standard koji je implementiran preko „*Robert Bosch GmbH*“ [5] CAN-*bus* infrastrukturne mreže [Slika 2] kroz CAN protokol.



Slika 2. CAN-bus magistrala (sabirnica)
izvor: https://www.kmpdrivetrain.com/paddleshift/practical-tips-can-bus/

2.1 OBD II standard (*SAE J1979*)

Standardizacija je u mnogim oblastima donela eksponencijalan napredak, pa tako i uvođenjem OBD II standarda u automobilsku industriju, dobijena su ogromna poboljšanja.

OBD II standard (eng. ***On-Board Diagnostics***) [11] je predstavljen preko svojih sistemskih funkcija koje često detektuju kvarove na automobilu do kojih je došlo, a da sam korisnik još uvek nije ni svestan da postoje. Takođe, ovaj sistem detekcije u neregularnosti do kojih je došlo u samom sistemu prečišćavanja izduvnih gasova, kao što je ukratko spomenuto u [1.Uvod]-nom delu, sprečava ispuštanje štetnih gasova u atmosferu.

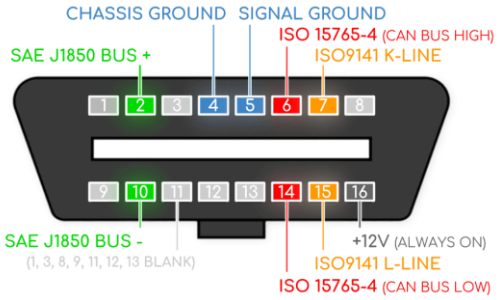
Pristup informacijama koja su uskladištena u internu memoriju nekih od računara se obavlja preko dijagnostičko porta koji je takođe definisan standardom, koji će u daljem radu biti opisan.

[5] Aktuator- U*ređaj kojim se na pobudu upravljačkog signala, pokretni delovi sistema dovode u željeni položaj.*

Kada se postavi pitanje, šta OBD predstavlja, moglo bi se reći da je to ugrađeni samo-dijagnostički sistem, koji ukoliko registruje neispravnost očitavanja senzora, aktuatora, i ostalih elektronskih komponenti, pali indikator lampicu (*MIL*) na instrument tabli vozila, koja sugeriše na to da bi automobil trebao da pregleda stručni tehničar. Takođe, prostim rečima, OBD II standard predstavlja „jezičku strukturu“, dok bi CAN-bus mogao da se opiše kao sredstvo komunikacije, slično telefonskom uređaju.

OBD II standard je definisan OBD konektorom i svojih pet najbitnijih standardnih protokola i to:

SAE J1962: Ovaj standard definiše fizički konektor koji se koristi za OBD II povezivanje, tj. OBD II konektor [Slika 3]. Standard opisuje OBD II konektor vozila i konektor koji koristi spoljašnja oprema za ispitivanje (npr. OBD II skener ili OBD II loger podataka). Standard posebno određuje lokaciju i pristup OBD II konektoru [12].



Slika 3. OBD II konektor
izvor: https://www.csselectronics.com/

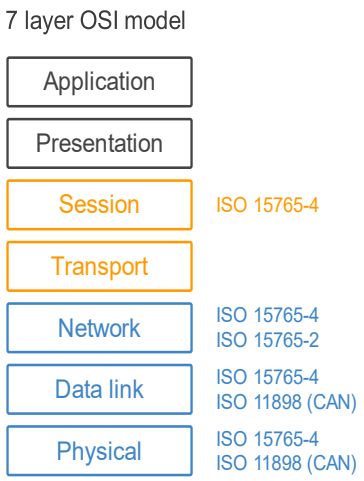
SAE J1979: Standard SAE J1979 opisuje metode za traženje dijagnostičkih informacija putem CAN protokola. Takođe uključuje listu standardizovanih javnih ID-ova OBD II parametara (*OBD II PID*) [13] koje proizvođači automobila mogu primeniti u vozilima (mada od njih to nisu obavezni, sem onih za kontrolu emisije štetnih gasova). OEM[6] proizvođači vozila takođe mogu da odluče da primene dodatne zaštićene OBD II PID-ove, osim onih koji su navedeni standardom SAE J1979 [11].

SAE J1939: Standard J1939 opisuje protokol razmene podataka koji koriste teška vozila. Dok su OBD II PID informacije dostupne samo na zahtev opreme za testiranje. OBD II protokol J1939 koristi se u većini teških vozila kao osnovno sredstvo za komunikaciju CAN saobraćaja, što znači da se podaci emituju kontinuirano [14].

ISO 11898: Ovaj standard opisuje sloj veze podataka CAN-*bus* sabirnice i fizički sloj, koji služi kao osnova za OBD II komunikaciju u većini automobila danas [15].

ISO 15765-4: Standard koji opisuje „Transportni sloj“, tj. kako poslati pakete podataka koji prelaze 8 bajtova preko CAN-*bus* magistrale. Ovaj standard je važan jer čini osnovu za komunikaciju objedinjenih dijagnostičkih usluga (*UDS*), koja se oslanja na slanje „*multiframe*“ CAN paketa podataka [16].

ISO 14229-1: Ovo detaljno opisuje UDS[7] komunikaciju, koja je izvedena iz sada već zastarelog standarda ISO 15765-3 [17].



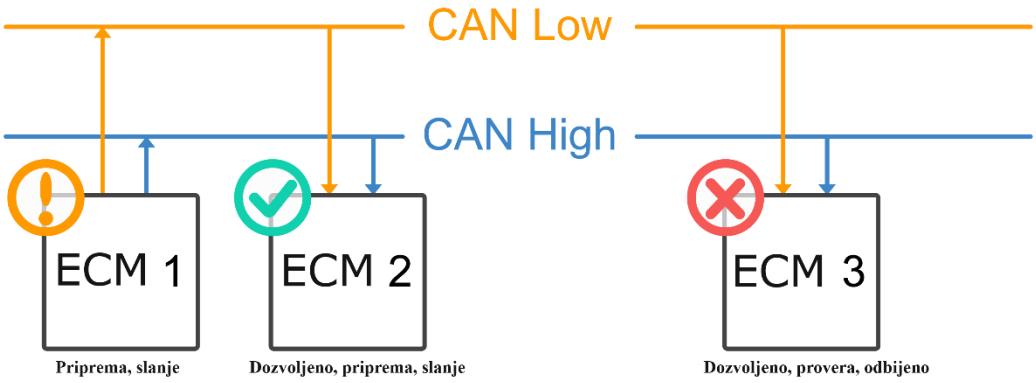
Slika 4. Mrežni OSI model
izvor: https://www.csselectronics.com/

[6] OEM – *Proizvođači Originalne Opreme (eng.* ***O****riginal* ***E****quipment* ***M****anufacturer).*
[7] UDS -Objedinjene dijagnostičke servisi (eng. ***U****nified* ***D****iagnostic* ***S****ervices*).

2.2 CAN protokol (*ISO 11898*)

CAN (eng. ***C**ontroller **A**rea Network*) mrežni protokol je u ovom istraživanju dobio centralnu pažnju jer je upotrebljen kao primarni komunikacioni kanal preko koga su uzorkovani podaci za dalju analizu, a u svrhu predikcije ispravnosti očitavanja senzora protoka vazduha motora u automobilu (MAF[8] senzor). Osnovna prednost ovog protokola je ostvarivanje komunikacione putanje koja na svojim krajevima obezbeđuje siguran prijem u razmeni podataka svih čvorova (eng. *nodes*) koji su u ovom slučaju definisani kao mikrokontroleri. Savremena implementacija mikrokontrolera u automobilskoj industriji podrazumeva visok stepen integracije sa svim definicijama zahteva CAN protokola.

Kada se kaže čvor, posebno se misli na mikrokontrolere-računare koji se još skraćeno nazivaju ECU (eng. ***E**lectronic **C**ontrol **U**nit*), takođe, njihov sinonim je poznat kao ECM (eng. ***E**lectronic **C**ontrol **M**odule*). Skraćenica za glavni motorni računar je takođe ECU, a da ne bi dolazilo do zabune, u ovom istraživačkom radu za globalno sve računare u automobilu će se koristiti skraćenica ECM, dok za glavni motorni računar, skraćenica ECU.



Slika 5. CAN protokol (ISO 11898)
izvor: https://www.csselectronics.com/

Implementacija CAN protokola opisana je CAN-*bus* mrežom i to tako što je deklarisana na fizičkom sloju (eng. *Physical layer*) ISO 11898-1 [15] i na mrežnom sloju veze (eng. *Data link layer*) ISO 11898-2 [18] OSI mrežnog modela [Slika 4].

Standardom ISO 11898-1 su definisane fizičke osobine koje moraju da se zadovolje kao što su: tipovi kablova, jačina električnih signala itd. Ovim standardom su definisana dva različita stanja signala koji su dominantni u CAN-*bus* mreži i njihov naponski nivo je određen načinom implementacije ISO 11898-1 standarda, a najznačajniji su:

- *High-Speed* CAN-*bus* (*CAN-H*) gde je ograničenje brzine od 40 kbit/s pa sve do 1 Mbit/s. Naponski nivo signala ove parice je 3.5v.
- *Low-Speed* CAN-*bus* (*CAN-L*) gde je ograničenje brzine od 40 kbit/s pa sve do 125 kbit/s. Naponski nivo signala ove parice je 1.5v.

Standardom ISO 11898-2 su definisane osobine kao što su: brzina protoka, frekvencijsko semplovanje signala, terminacija signala na krajevima mrežnih parica 120Ω otpornikom itd.

Ovi standardi nisu toliko bitni za ovo istraživanje jer su uzorkovani podaci sa automobila, a ne iz simulacionih hardverskih sistema, te nije bilo potrebe za detaljima implementacije koji su opisani pomenutim ISO standardima.

[8] MAF - ***M**ass **A**ir **F**low sensor – Senzor protoka mase vazduha motora automobila.*

2.3 Poruke CAN protokola

Specifičnost komunikacije u CAN-*bus* mreži se ogleda u samoj realizaciji koja je definisana porukama koje se pakuju u okvire (eng. *frame*). Ova činjenica omogućava da poruke enkapsulirane paketnim okvirima, koje se šalju u magistralu, stižu do svakog čvora (*noda*) koji svojim lokalnim filterom odlučuje da li je informacija u poruci korisna za njegovo dalje delovanje [19].

CAN protokol u svojoj „paleti“ poseduje četiri tipa poruka i to:

- Poruku sa podacima (eng. *Data frame*).
- Poruku sa zahtevom za podatke (eng. *Remote frame*).
- Poruku greške (eng. *Error frame*).
- Poruku o zauzetosti (eng. *Overload frame*).

2.3.1 Poruka sa podacima (*Data frame*)

„Data frame“ tip poruke je najčešći tip koji se šalje u CAN-bus mrežu i ova vrsta poruke se može smatrati kao tip poruke kojom se šalju podaci o stanju i vrednostima koje poseduje krajnji čvor (eng. *transmiter*). Trenutno su izvedene dve vrste poruka CAN protokola i razlika u njima je samo u dužini standardnog identifikatora (ID) poruke koja može da se pošalje u CAN-*bus* mrežu .

- Standardni identifikator paketa CAN2.0A podprotokola je definisan sa 11bita i on ima najčešću primenu u automobilskoj industriji [Slika 6].

Slika 6. Standardni 11bit CAN2.0A paket (eng. frame)
izvor: https://www.csselectronics.com/

- Standardni identifikator paketa CAN2.0B podprotokola je definisan sa 29bita i njegova implementacija se može videti u SAE J1939 [14] protokolu [Slika 7]. koji je rezervisan za teška vozila (kamioni, traktori, građevinska mehanizacija itd.).

Slika 7. Standardni 29bit CAN2.0B paket (eng. frame)
izvor: https://www.csselectronics.com/

2.3.2 Poruka sa zahtevom za podatke (*Remote frame*)

CAN2.0A i CAN2.0B protokoli dozvoljavaju da svi čvorovi-računari u isto vreme šalju podatke u mrežu. Spomenuto je, da ukoliko je nekom od računara potrebna informacija koja je emitovana, dovoljno je samo da je presretne sa mreže i obradi.

„*Remote frame*“ tip poruke se bazira na, kako je i samo ime opisuje, zahtevom za informacijom od određenog računara u automobilu. Ukoliko je prioritet veliki za informacijom određenog noda, za sinhronizovano delovanje drugog, potrebno je da zavistan čvor samo emituje zahtevnu poruku u mrežu sa sadržajem u kojoj je protokolom opisano zaglavlje zahtevne poruke, nodu koji se inicira za slanje poruke.

Pojednostavljen scenario u kojem je glavnom motornom računaru (*ECU*) potrebna informacija o stanju sistema protiv proklizavanja *ABS* (eng. ***A**nti-lock **B**reaking **S**ystem*) računara, u momentu da bi odredio količinu goriva koji je potreban za usklađen rad motora. Tada *ECU* šalje „*Remote Frame*“ poruku sa specifičnim identifikatorom koji prepoznaje samo *ABS* računar. Prilikom prijema zahtevne poruke, *ABS* računar istog momenta vraća „*Data frame*“ informaciju o svom stanju glavnom motornom računaru.

Razlika u porukama za podatke i poruci zahteva za podatke je u polju RTR[9], tj. u bitu dela standardne CAN poruke. Bit koji je u RTR polju, određuje da li je poruka u statusu slanja ili je poruka „zahtevanja“ podatka drugog računara. RTR polje ima još i namenu za označavanje poruke koja u sebi sadrži strukturnu grešku i koja bi trebala da bude propuštena i prepuštena krajnjim terminacionim otpornicima za uništenje.

2.3.3 Poruka o grešci (*Error frame*)

Iako je CAN-*bus* mreža veoma otporna na greške, u nekim situacijama se ipak pojavljuju paketi sa oštećenjem. Pošto je sistem CAN-*bus* mreže robustan, tako je i otkrivanje oštećenih paket na veoma visokom nivou, a jedan od mehanizama je označavanje poruke sa greškom, koji predstavlja još jedan specijalan tip CAN poruke.

Ukoliko neki od računara otkrije paket sa greškom, inicira se specijalan tip poruke koji se šalje u mrežu sa informacijom da i ostali nodovi odbace detektovanu poruku.

Ostali tipovi grešaka su:

- Greška u poslatom bitu (eng. *Error Bit*).
- Greška pri umetanju bita (eng. *Bit Stuffing Error*).
- CRC greška (eng. *CRC error*).
- ACK greška (eng. *ACK error*).

2.3.4 Poruka o zauzetosti (*Overload frame*)

Ukoliko se računar nađe u situaciji da nije u mogućnosti da pošalje poruku u mrežu iz razloga vremenskog procesuiranja ostalih pristiglih poruka, kreira se specijalan tip poruke pod nazivom „*Overload frame*“.

Ovakav vid poruke je u današnjim modernim mikrokontrolerima, sa izuzetnom brzinom, veoma redak, ali usled sve više hakerski napada na otvorene i nedovoljno zaštićene sisteme koji su već integrisani u automobilskoj industriji, ipak moguć. U vreme početka implementacije CAN protokola OBD II standarda, krajem prošlog veka, nije se razmišljalo o neželjenom pristupu mreži automobila, pa je sigurnost stavljena u „drugi plan“. Ovo je otvorilo nestručnim licima da nehotice izazovu preopterećenje mreže, pa samim tim i aktivaciju „*Overload frame*“ tip poruke.

[9] RTR - ***R**emote **T**ransmission **R**equest (Zahtev za transmisiju poruke).*

2.4 Struktura poruke sa „*zahtevom za podatke*“ (*Data frame*)

Ovo istraživanje se koncipira na predikciji očitavanja MAF senzora na osnovu podataka dobijenih iz CAN-*bus* mreže. Opisima u ranijim poglavljima se dobio uvid u informaciono-komunikacionu infrastrukturu koja je u osnovi svakog modernog vozila, dok su za implementaciju predikcije ispravnog očitavanja MAF senzora, potrebne poruke koje su tipa „zahteva za podacima“ (*Data frame*) [20].

Slika 8. Struktura standardne 11bit CAN2.0A poruke
izvor: https://www.csselectronics.com/

Standardna poruka CAN protokola OBD II paketa sadrži osam polja [Slika 8], i to:

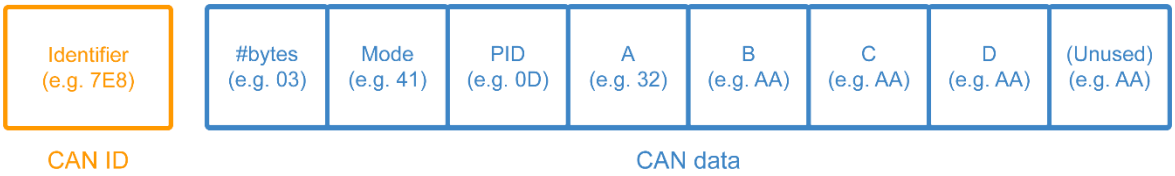
- **SOF** (eng. ***S**tart **O**f **F**rame*): Početak okvira u kom je „dominantna 0“, da bi se ostalim čvorovima reklo da CAN čvor namerava da razgovara.
- **ID** (eng. ***S**tandard **I**dentifier*): Identifikator okvira - niže vrednosti imaju veći prioritet.
- **RTR** (eng. ***R**emote **T**ransmision **R**equest*): Zahtev za daljinski prenos pokazuje, da li čvor šalje podatke ili zahteva namenske podatke iz drugog čvora[10].
- **Kontrola** (eng. *Control*): Sadrži identifikator produženog bita (IDE[11]) za koji je „dominantna 0“ za 11-bit. Takođe sadrži 4-bitni kôd za dužinu podataka (DLC[12]) koji određuje dužinu bajtova podataka koji se prenose (0 do 8 bajtova).
- **Podaci** (eng. *Data*): Sadrže bajtove podataka pod nazivom „koristan teret“ (eng. *payload*), koji uključuju CAN signale, a mogu se izvući i dekodirati radi dobijanja informacija.
- **CRC** (eng. ***C**yclic **R**edundancy **C**heck*): 16-bit (15bita + 1bit za razdvajanje). Provera količine poslatih bitova u svrhu osiguravanja integriteta podataka.
- **ACK** (eng. *Acknowledgement*): ACK indikator predstavlja mehanizam koji daje uvid u integritet poruke ali isto tako i da li je čvor uspešno primio informaciju.
- **EOF** (eng. ***E**nd **O**f **F**rame*): Označava kraj CAN paketa.

Za generisanje skupa podataka u ovom istraživanju potrebna su samo dva polja identifikatora i ona su označena žutom i plavom bojom na [Slika 8], i to:

- **„ID“** identifikator.
- **„DATA“** identifikator.

[10] Čvor – *Takođe se naziva i mikrokontroler-računar koji se nalazi u automobilu.*
[11] IDE – *Dominantni jedinstveni identifikator ekstenzije (eng. dominant single identifier extension).*
[12] DLC - *Kôd za proveru dužine (eng. **D**ata **L**ength **C**ode).*

Daljim uvidom u strukturu „*ID*“ i „*DATA*“ polja, dobijamo konačnu reprezentaciju poruke koja se šalje u CAN-*bus* mrežu zbog dobijanja traženih informacija od krajnjih nodova-računara u automobilu i to sa identifikatorima [Slika 9]:



Slika 9. Struktura "ID" i "DATA" polja OBD II paketa CAN2.0A 11bit poruke
izvor: https://www.csselectronics.com/

„*ID*“ identifikator:

- Za OBD II CAN2.0A poruke, standardno polje identifikatora je dužine 11-bita i koristi se za razlikovanje „poruka zahteva“ (*ID* $_{0x}$7DF) i „poruka odgovora“ (*ID* $_{0x}$7E8 do $_{0x}$7EF). Vrednost „$_{0x}$7E8“ je u praksi obično PID sa kojim odgovor daje glavni motorni računar (ECU).

„*DATA*“ identifikator:

- ***Dužina***: Ovo određuje dužinu u broju bajtova preostalih podataka ($_{0x}$03 do $_{0x}$06). Kao primer, zahtev u modu $_{0x}$01 za trenutne vrednosti MAF senzora je $_{0x}$02 (jer slede samo $_{0x}$01 i $_{0x}$23), dok je za odgovor $_{0x}$03, jer slede $_{0x}$41, $_{0x}$23 i $_{0x}$32.

- ***Režim*** (eng. *Mode*): Za poruke tipa „zahtevanja podataka“ će ovo biti vrednost između $_{0x}$01-$_{0x}$0A. Odgovori $_{0x}$0 se zamenjuju sa $_{0x}$4 (tj. $_{0x}$41, $_{0x}$42, ..., $_{0x}$4A). Postoji deset režima kako je opisano u OBD II standardu SAE J1979 [Tabela 1]. Režim $_{0x}$01 prikazuje trenutne podatke stanja i očitavanja svih senzora koji su trenutno u operaciji. Režim $_{0x}$02 je koristan, jer ukoliko je došlo do greške koja je uskladištena u memoriju nekog računara, pored opisa greške, beleže se i očitavanja senzora u trenutku pojave same greške. Pozivom režima $_{0x}$02 se dobijaju vrednosti koji se još nazivaju „zamrznut okvir podatka“. Ostali opisi režima se mogu videti takođe u [Tabela 1].

- ***PID*** (eng. ***P**arameter **ID***): Za svaki režim rada postoji lista standardnih OBD II PID [13] identifikacionih zaglavlja. Tako npr. u režimu $_{0x}$01, pozivom PID: $_{0x}$23 se dobijaju očitavanja vrednosti MAF senzora, u režimu $_{0x}$01 pozivom PID: $_{0x}$0C se dobijaju očitavanja senzora brzine motora itd.
  Nakon dobijenih vrednosti, pristupa se konverziji očitavanja jer su dobijene vrednosti u heksadecimalnom brojnom sistemu. Nakon konverzije se radi kalkulacija na osnovu predefinisanih formula, koje su opisane u [2.5 Konverzija izvornih podataka „poruka *sa zahtevom za podatke“*].

- ***A, B, C, D***: Su bajtovi podataka dobijeni u heksadecimalnom zapisu, koje treba pretvoriti u decimalni oblik pre nego što se upotrebe u proračunima konverzione formule. Potrebno je naglasiti da se poslednji bajt podataka (posle polja $D) ne koristi.

2.5 Konverzija izvornih podataka „*poruka sa zahtevom za podatke*“

Poruke, koje se dobijaju kao odziv na zahtev slanja poruka u CAN-*bus* mrežu, su u „izvornom“ (eng. *raw*) formatu. OBD II standard je definisan heksadecimalnim[13] brojnim sistemom, kodirani ASCII[14] karakterima i to tako da nad dobijenim vrednostima mora de se primeni prvo konverzija u decimalan brojni sistem, pa onda formula konverzije, koja dobijene vrednosti pretvara u SI[15] metrički sistem. Tek nakon izvršenih matematičkih operacija konverzije, podaci su spremni za dalju obradu i analizu. SAE J1979 kao deo svog protokola daje tačan pregled standardizovanih identifikacionih parametara (OBD II PID, u daljem tekstu: PID) [13], koje su proizvođači u obavezi da implementiraju u svojim automobilima. Ovi parametri se odnose na identifikatore: „*MODE*“ i „*PID*“ CAN poruke, dok se matematička konverzija odnosi samo na polja identifikatora: „*A, B, C, D*“ [Slika 9].

Predikciji ispravnosti očitavanja MAF senzora, kao predmetu istraživanja ove master teze, identifikator „*MODE*“, je potreban iz: $_{0x}$01 operacija (prikaz trenutnih vrednosti senzora) [Tabela 1], dok je lista identifikatora generisanja podataka u [Tabela 2].

MODE (heksadecimalno)	OPIS
$_{0x}$01	**Prikaz vrednosti u realnom vremenu**
$_{0x}$02	Prikaz „zamrznutog“ paketa (*frame*)
$_{0x}$03	Prikaz memorisanih grešaka u sistemu
$_{0x}$04	Brisanje memorisanih grešaka
$_{0x}$05	Testni rezultati O2 senzora (ne iz CANa)
$_{0x}$06	Testni rezultati O2 senzora (samo iz CANa)
$_{0x}$07	Prikaz detektovanih grešaka u toku zadnje vožnje
$_{0x}$08	Kontrola rada ugrađenih komponenti sistema
$_{0x}$09	**Zahtev za informacijama o vozilu (VIN)**
$_{0x}$0A	Prikaz stalnih grešaka koje ne mogu da se obrišu

Tabela 1. OBD II "MODE" parametri

PID	Dužina(b)	Min	Max	f (x)	Opis
$_{0x}$**00**	4			Bit Encoded	Prikaz podržanih PID [$A7_{16}$ - $D0_{16}$ … 01_{16} - 20_{16}]
$_{0x}$**04**	1	0	100	100/255*A	Izračunato opterećenje motora (%)
$_{0x}$**05**	1	-40	215	A-40	Temperatura motora (°C)
$_{0x}$10	**2**	**0**	**655,35**	**(256A+B)/100**	**MAF senzor (grams/sec)**
$_{0x}$**0A**	1	0	765	3*A	Pritisak goriva (kPa)
$_{0x}$0B	**1**	**0**	**255**	**A**	**MAP pritisak (kPa)**
$_{0x}$0C	**2**	**0**	**16.383,75**	**(256A + B)/4**	**Obrtaji-brzina motora (rpm)**
$_{0x}$**0D**	1	0	255	A	Brzina Vozila (km/h)
$_{0x}$**0F**	2	-40	215	A-40	Temperatura usisa vazduha (°C)
$_{0x}$23	**2**	**0**	**655.350**	**10(256*A + B)**	**FGP - Pritisak merača razvodnika goriva (kPa)**

Tabela 2. OBD II „PID“ standardni identifikacioni parametri

[13] Heksadecimalan brojni sistem – *Brojni sistem kojem je osnova 16.*
[14] ASCII – *Američki standard za razmenu informacija (eng.* ***A****merican* ***S****tandard* ***C****ode for* ***I****nformation* ***I****nterchange).*
[15] SI – *Međunarodni sistem jedinica (franc.* ***S****ystème* ***i****nternational d'unités).*

Primer slanja „*Data frame*“ poruke za informaciju MAF senzora u CAN-*bus* mrežu preko „*Infinity*“ OBD II interfejsa ima oblik [Tabela 3]:

- **[cansend] [can0] [$_{0x}$7DF] # [$_{0x}$02] [$_{0x}$01] [$_{0x}$10] [$_{0x}$00] [$_{0x}$00] [$_{0x}$00] [$_{0x}$00] [$_{0x}$00]**

Komanda	Interfejs	ID	Zahtev	Mod	PID	A	B	C	D	n/a
cansend	can0	$_{0x}$7DF	$_{0x}$**02**	$_{0x}$**01**	$_{0x}$**10**	$_{0x}$00	$_{0x}$00	$_{0x}$00	$_{0x}$00	$_{0x}$00

Tabela 3. Primer slanja CAN poruke za MAF senzor

Dok odgovor koji je dobijen ima oblik [Tabela 4]:

- **[1613135352.451] [can0] [$_{0x}$7E8] # [$_{0x}$04] [$_{0x}$41] [0x10] [$_{0x}$05] [$_{0x}$70] [$_{0x}$AA] [$_{0x}$AA] [$_{0x}$AA]**

Vreme	Inter.	ID	Odgovor	Mod	PID	A	B	C	D	n/a
1613135352	can0	$_{0x}$7E8	$_{0x}$04	$_{0x}$41	$_{0x}$10	$_{0x}$**05**	$_{0x}$**70**	$_{0x}$AA	$_{0x}$AA	$_{0x}$AA

Tabela 4. Primer odgovora CAN poruke za MAF senzor

Kada je dobijen konačan odgovor ECU o informacijama trenutnog očitavanja MAF senzora, potrebno je izvršiti konverziju podataka. Identifikatori koji se uzimaju kao referentni za konverziju su: *A*, *B*, *C* i *D*, dok se u ovom slučaju vidi da samo polja *A*, *B* imaju vrednosti.

- **A ($_{0x}$05)**
- **B ($_{0x}$70)**

Pa iz toga sledi:

$$_{0x}\mathbf{MAF} = \mathbf{AB}\,\frac{grams}{sec} \quad => \quad _{0x}\mathbf{MAF} = {}_{0x}\mathbf{05}\ {}_{0x}\mathbf{70}\,\frac{grams}{sec}$$

Prvi korak u konverziji je prebacivanje iz heksadecimalnog brojnog sistema u decimalni:

$$\mathbf{A_{10}} = 05 = \mathbf{(_{0x}05)} = (0 \times 16^1) + (5 \times 16^0) = \mathbf{(5)_{10}}$$
$$\mathbf{B_{10}} = 70 = \mathbf{(_{0x}70)} = (7 \times 16^1) + (0 \times 16^0) = \mathbf{(112)_{10}}$$

Nakon konverzije iz heksadecimalnog sistema u decimalni, potrebno je upotrebiti formulu za izračunavanje vrednosti MAF senzora i to:

$$\mathbf{MAF} = \frac{(256 * A) + B}{100} \tag{1}$$

$$\mathbf{MAF} = \frac{(256*5) + 112}{100} = \mathbf{13.92}\ \frac{grams}{sec}$$

Ostale formule konverzije bitne za ovo istraživanje su date u (2), (3), (4).

$$\mathbf{RPM} = \mathrm{A}\ (\mathrm{rpm}) \tag{2}$$

$$\mathbf{MAP} = \mathrm{A}\ (\mathrm{kPa}) \tag{3}$$

$$\mathbf{FGP} = 10*(256*\mathrm{A}+\mathrm{B})\ \ (\mathrm{kPa}) \tag{4}$$

3. Senzor mase protoka vazduha (*MAF Senzor*)

Senzor mase protoka vazduha (*MAF* senzor)[16] je veoma važna komponenta u novijim dizel i Otto[17] motorima sa unutrašnjim sagorevanjem u automobilskoj industriji. Glavni motorni računar (ECU) koristi informacije koje daje senzor mase protoka vazduha da bi uskladio isporučenu količinu goriva koja je potrebna za ispravan rad motora [Slika 10].

Slika 10. "Hot-Wire" Senzor mase protoka vazduha (MAF)
izvor: https://www.boschautoparts.com

Ispravna smeša vazduh/gorivo je u tesnoj sprezi sa količinom emisije štetnih gasova koje se ispuštaju u atmosferu. Ukoliko smeša vazduh/gorivo nije u idealnoj proporciji, emisija gasova kao što su: ugljen-dioksid (CO2), ugljen-monoksid (CO), ugljovodonici (HC), oksidi azota (NOx), nisu u propisanim standardima koje su odredile agencija za zaštitu životne sredine [1] [2], pa samim tim dolazi do povećanog zagađenja prirodne okoline.

Pored određivanja potrebne „idealne“ količine goriva, koju je potrebnu isporučiti u cilindre motora, očitavanje protoka vazduha se upotrebljava još i kao parametar za ispravno vreme paljenja smeše u cilindru motora i kao jedan od parametara, kod vozila sa automatskim menjačem, za pravovremenu promenu brzine.

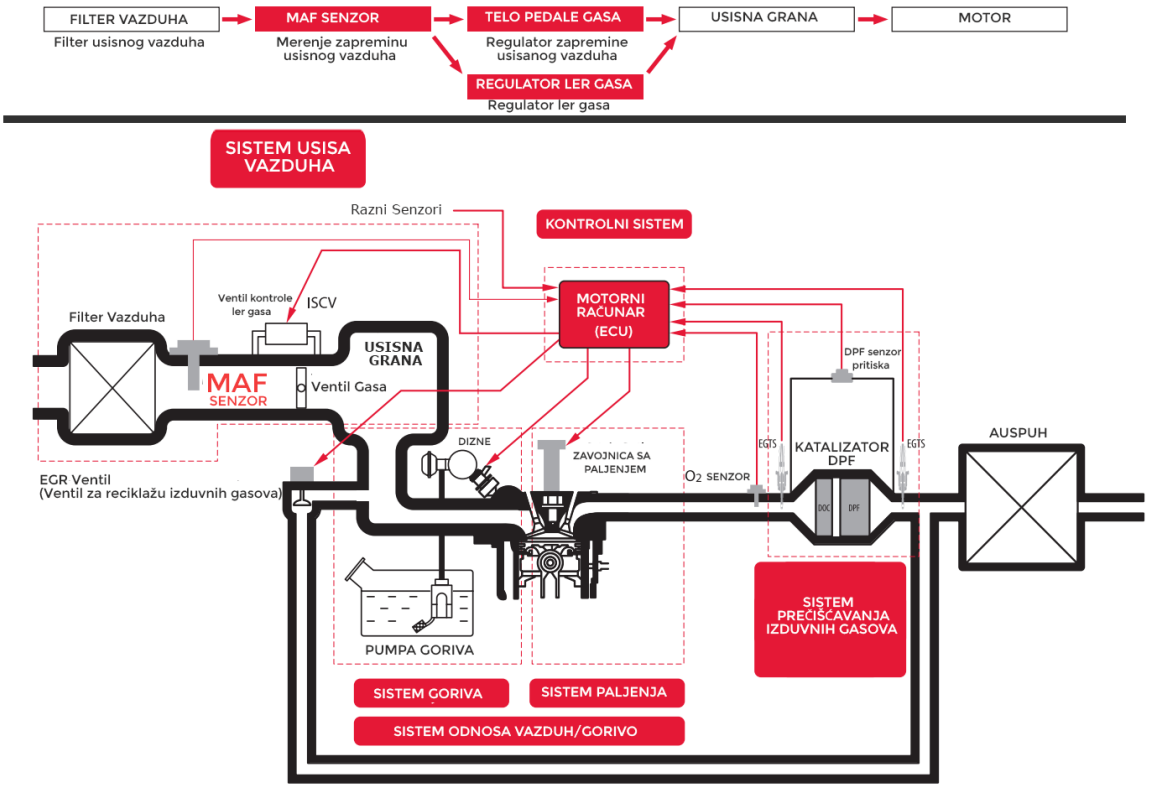
Merenje MAF senzora se uvek kombinuje sa ostalim senzorima, tako u konjunkciji sa senzorom kiseonika (O_2 senzor), može da se kaže da su rezultati izuzetno tačni. Pored O_2 senzora, za tačnost očitavanja ECU koristi i senzor temperature vazduha u usisnoj grani motora (eng. ***I**ntake **A**ir **T**emperature* - ***IAT***).

Istraživanja koja su sprovedena u ovoj master tezi su obavljana u specijalizovanom auto centru „*AC Reno*“ [21] i tokom boravka u „*AC Reno*“ servisu „isplivavale“ su situacije u kojima je neispravno očitavanje MAF senzora ukazivalo na razne kvarove koji su se identifikovali prilikom defektaže vozila, a nikako nisu imale veze sa MAF senzorom. Rezultati defektaže kvarova, prouzrokovane pogrešnim očitavanjem MAF senzora, u raznim situacijama su ukazivala na smanjen pritisak u sistemu ubrizgavanja, na pogrešna očitavanja O_2 senzora, pa čak u nekim situacijama uvid u greške koje su memorisane u ECU računaru putem dijagnostičkog uređaja, nisu davale nikakav rezultat.

Kako MAF senzor radi na principu merenja mase protoka vazduha zagrevanjem tanke žice od platine, jedan od najčešćeg kvara senzora je zaprljanost te žice usled nagomilavanja mikro čestica prašine, pa samim tim može da se desi da automobil radi sasvim normalno ali da uprkos tome očitavanja MAF senzora nisu dobra.

[16] U automobilskoj stručnoj praksi se senzor mase protoka vazduha još naziva i „Protokomer“.
[17] Otto motor – *Prvi konstruisan benzinski motor (Nicolaus August Otto. 1876, Nemačka).*

Senzor protoka vazduha se nalazi u sistemu usisa vazduha, obično posle filtera za vazduh, a pre usisne grane motora. Ova lokacija je u strateškom smislu pozicije, jedna od najbitnijih tačaka u motoru automobila [Slika 11]. Pošto se nalazi na ovako važnom mestu u sistemu upravljanja motorom, ispravnost očitavanja ovog senzora je važna i jedan je od najznačajnijih razloga zašto se za temu ovog istraživačkog rada, „predikciju ispravnog očitavanja“, baš MAF senzor odabrao.



Slika 11. Sistemi upravljanja motorom sa unutrašnjim sagorevanjem
izvor: denso-am.eumedia102385dems14-0001.pdf

3.1 Tipovi MAF senzora

Najčešće zastupljeni tipovi senzora protoka vazduha koji se koriste u automobilskoj industriji, a za merenje mase protoka vazduha u usisnoj grani motora su:

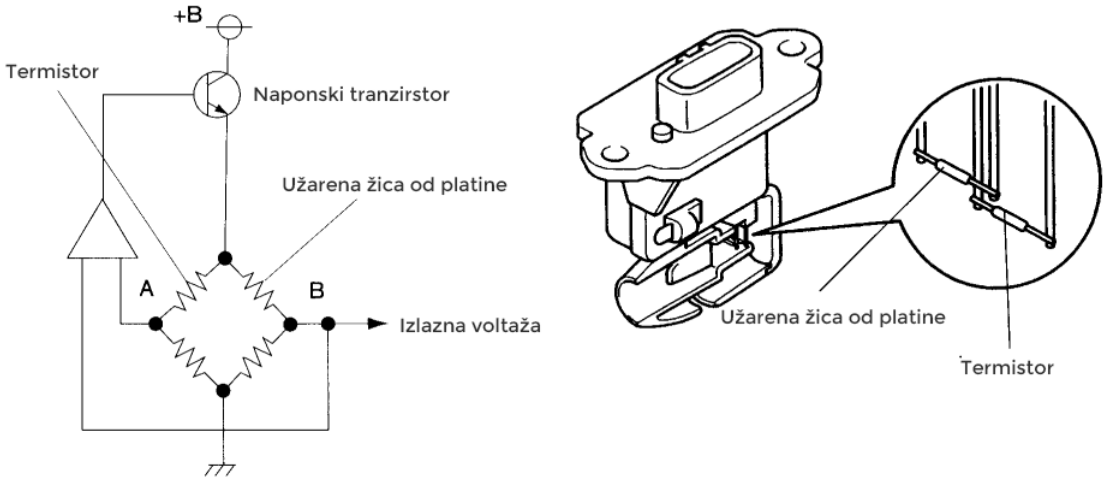
- Zapreminski
  - *VAF- Vane Air Flow* senzor
  - *KV - Karman Vortex* senzor
- Meseni (*MAF – Mass Air Flow*)
  - *Hot-Wire MAF* senzor
  - *Hot Film MAF* senzor

Zapreminski *VAF* senzor je tip senzora koji se koristio u starijim automobilima i njegov osnovni princip rada se oslanja na fizički pritisak vazduha na metalnu ploču koja daje signal protoka vazduha, dok *KV* senzor kontinualno stvara vrtlog koji služi za merenje.

U ovom istraživanju se koristio noviji tip senzora, „*Hot Wire*“ MAF senzor, pa iz tog razloga, u sledećem poglavlju će se dati kratak princip rada samo ove vrste senzora.

3.2 Princip rada MAF senzora

MAF senzori, bez obzira da li je način rada „*Hot-Wire*“ ili „*Hot-Film*“, merenje mase protoka vazduha se oslanja na princip teorije prenosa toplote (eng. *Heat Transfer*) [22], slično anenometru[18], koji u svojoj osnovnoj ideji ima održavanja konstantne temperature užarene žice (eng. *hot-wire*).



Slika 12. MAF „Hot-Wire“ senzor
izvor: https://www.oscium.com/sites/default/files/MAF_autoshop101.pdf

Startovanjem motora automobila, MAF senzor dobija konstantnu voltažu koja zagreva platinsku žicu MAF senzora na temperaturu od ≈100°C iznad izmerene spoljne temperature[19]. Prilikom strujanja vazduha u sistemu usisa motora [Slika 11], gde se nalazi i senzor merenja protoka vazduha, usled zakona termodinamike, platinska žica se hladi. Kao rezultat hlađenja, a tendencija održavanja konstante toplote platinske užarene žice je stalna, dobijamo povećanje napona struje koji je potreban za održavanja ravnoteže.

To diferencijalno stanje u volumenu količine vazduha koji struji preko platinske žice, je srazmerno količini potrebne električne energije da je održi na konstantnih ≈100°C, a koja je interpretirana vrednošću izlaznom voltažom od 0.5V - 0.8V na ler gasu u stanju mirovanja vozila i doseže svoj maksimum, kada je pozicija papučice gasa 100% i iznosi 4.5V – 5V.

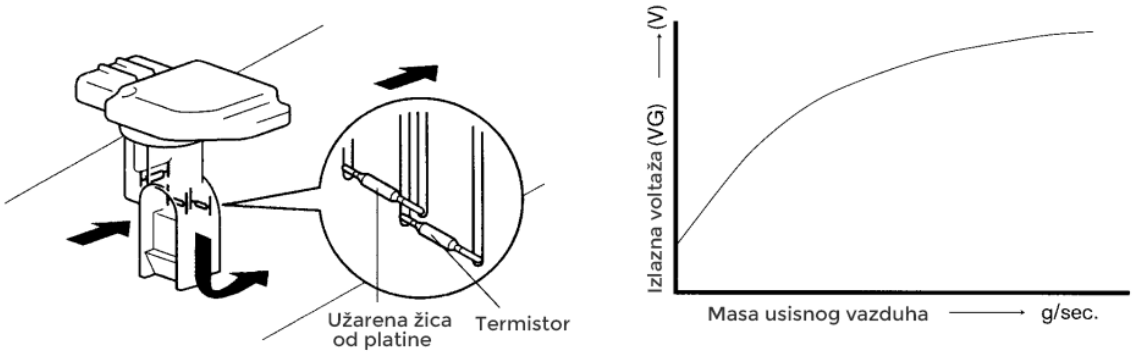
Pomenuto je da je jedan od najčešćeg kvara MAF senzora zaprljanost platinske žice mikro česticama prašine, silikona, motornog ulja, pa je kao jedna od preventivnih mera samočišćenja uvedena procedura koja prilikom gašenja motora automobila podiže temperaturu platinske žice na ≈1000°C. Ovaj električni impuls ima trajanje od 1sekunde, a potrebno je još napomenuti, da je za ovu operaciju napajanja čitavog sistema, implementiran poseban relej, koji se u nekim automobilima nalazi u modulu instrument table.

Izlazna voltaža MAF senzora može biti u obliku analognog ili digitalnog signala. Analogni signal operiše na voltaži od 0.5V – 5V, dok je digitalni signal PWM[20] modulisan na frekvencijama u opsegu od 30Hz – 150Hz. MAF senzori sa digitalnim očitavanjima nemaju mogućnost samočišćenja.

[18] Anenometar – *Merni instrument za merenje brzine strujanja vazduha.*
[19] Kod „*Hot-Film*“ MAF senzora je to temperatura od ≈75°C.
[20] PWM - ***P**ulse-**W**idth **M**odulation.*

3.3 Teorijske osnove merenja MAF senzora

Žica od platine, predstavlja medijum koji je užaren na konstantnih ≈100°C, preko kojeg struji vazduh usisan u usisnu granu motora. Očigledno, da ukoliko je motoru potrebno više snage, utoliko mu je potrebno više i vazduha. Povećana potražnja za vazduhom prouzrokuje i veći protok vazduha preko medijuma za merenje, u ovom slučaju užarene platinske žice, a to prouzrokuje i hlađenje medijuma merenja. Da bi se održala konstantna temperatura, a usled većeg strujanja vazduha se ona smanjuje, povećava se i količina električne energije koja je potrebna da bi se održala konstantna temperatura. Ako se količina električne energije menja da bi se održala konstantna temperatura, može da se kaže da je količina električne energije koja je potrebna za tu ravnotežu ujedno i mera za protok vazduha [Slika 13].



Slika 13. Voltažni signal MAF senzora
izvor: https://www.oscium.com/sites/default/files/MAF_autoshop101.pdf

Matematički model koji bi reprezentovao konvekcioni prenos toplote užarene žice koja je u ravnoteži sa okolinom je jednak količini gubitka ulazne električne energije, srazmerno gubitku temperature usled konvekcionog prenosa toplote [23]:

$$I^2 R_w = \mathrm{h}^* A_w \left(T_w - T_f\right) \tag{5}$$

Gde su:

$\boldsymbol{I}$: Ulazna voltaža.
$\boldsymbol{R_w}$: Otpor platinske žice.
h : Koeficijent prenosa toplote žice.
$\boldsymbol{A_w}$: Izloženost površine žice.
$\boldsymbol{T_w}, \boldsymbol{T_f}$: Temperature žice odnosno fluida.

Ulazna voltaža (I) je konstanta u (5), dok je otpor platinske žice R_w dat u jednačini (6):

$$R_w = R_{Ref}[1+\alpha\left(T_w - T_{Ref}\right)] \tag{6}$$

Gde su:

$\boldsymbol{T_{Ref}}$: Referentna temperatura.
$\boldsymbol{R_{Ref}}$: Referentni otpor platinske žice.
$\boldsymbol{\alpha}$: Temperaturni koeficijent otpora žičanog materijala pri T_{Ref}.

Protok vazduha V_f je funkcija koeficijenta prenosa toplote platinske žice *h*, a prema [24], se može predstaviti kao (7):

$$h = \mathrm{a} + \mathrm{b} * V^c{}_f \tag{7}$$

Gde su *a*, *b* konstante ≈0.5.

Kombinacijom (5), (6), (7) dozvoljava nam da eliminišemo koeficijent toplotnog prenosa *h* kao u (8):

$$\mathrm{a} + \mathrm{b} * V^c{}_f = \frac{I^2 R_w}{A_w(T_w - T_f)} = \frac{I^2 R_{Ref}[1+\alpha(T_w - T_{Ref})]}{A_w(T_w - T_f)} \tag{8}$$

U nastavku (9), konačno dobijamo rešenje V_f protoka fluida (vazduha):

$$V_f = \left\{\left[\frac{I^2 R_{Ref}[1+\alpha(T_w - T_{Ref})]}{A_w(T_w - T_f)} - a\right]/b\right\}^{1/c} \tag{9}$$

U praksi, tehnološka implementacija masenih MAF senzora se razlikuje po samom merenju, a to su merenja konstantnom temperaturom i merenja konstantnim naponom električne energije.

MAF senzor tipa „*Hot-Wire*“, koristi rešenje merenja konstantnom temperaturom i ono se predstavlja kao funkcija ulazne električne energije (*I*) i temperature vazduha T_w i R_w, koje su ujedno i konstante (10):

$$\mathrm{a} + \mathrm{b} * V^c{}_f = \frac{I^2 R_w}{A_w(T_w - T_f)} = \boldsymbol{f}(\boldsymbol{I}, \boldsymbol{T_f}) \tag{10}$$

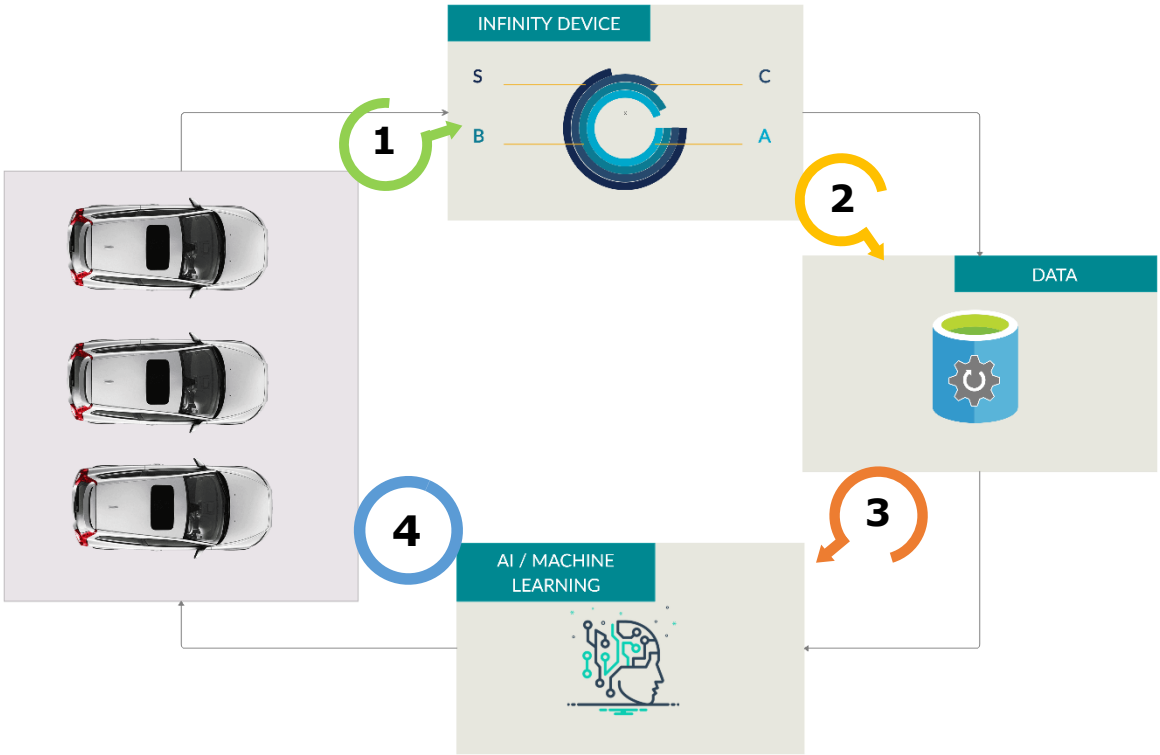
Temperatura protoka T_f se može meriti, pa se brzina fluida redukuje na funkciju samo ulazne struje (*I*).

MAF senzor koji koristi rešenje „Konstantan napon“, kao referenca za merenje protoka vazduha, ima oblik kao u (11).

$$\mathrm{a} + \mathrm{b} * V^c{}_f = \frac{I^2 R_{Ref}[1+\alpha(T_w - T_{Ref})]}{A_w(T_w - T_f)} = \boldsymbol{f}(\boldsymbol{T_w}, \boldsymbol{T_f}) \tag{11}$$

4. Topologija ciklusa predikcije ispravnosti očitavanja MAF senzora

Predložena topologija ciklusa ispitivanja MAF senzora se sastoji od četiri celine, gde će svaka od njih biti opisan u sledećim poglavljima. Predstavljena je u četiri koraka, kao što može da se vidi na [Slika 14].



Slika 14. Topologija ciklusa predikcije ispravnosti očitavanja MAF senzora

Koraci koji su predstavljeni na [Slika 14] su:

- Uređaj za pristup CAN-*bus* mreži, prikupljanje i predikciju.
- Podaci koji su u procesu predikcije senzora.
- Upotreba alata mašinskog učenja veštačke inteligencije.
- Vizuelna prezentacija dobijenih rezultata.

Senzor automobila, kao primarni subjekt ovog istraživanja, predstavlja nultu polaznu tačku u predloženom ciklusu. Zatim sledi „*Infinity - SCBA*“[21], multifunkcionalni hardver koji je konstruisan i napravljen u svrhu pristupa CAN-*bus* mreži, ali takođe njegova funkcionalnost se proteže i u svrhu prikupljanju podataka i predikciju ispravnosti MAF senzora [4.1].

Drugi korak u topologiji ciklusa opslužuje sistem sa podacima, a koji su ranije prikupljeni, i kao takve prosleđuje alatima veštačke inteligencije za dalju analizu i predikciju.

Treći korak je predikcija ispravnosti MAF senzora i njegova svrha je da podatke koji su prošli kroz korak veštačke inteligencije, uporedi ili sačuva za dalju analizu.

Zadnjim korakom se objedinjuju svi prethodni koraci i opisuju sistemi skladištenja, analize i vizuelizacije dobijenih rezultata.

[21] „SCBA“ – ***S**mart **C**an-**B**us **A**nalyser*

4.1 Multifunkcionalan uređaj „*Infinity SCBA*“

Za potrebe ovog istraživanja, pristup CAN-*bus* mreži je imao krucijalnu i centralnu tačku koja je morala da se implementira, tako da mogućnosti uređaja [Slika 15], osim prikupljanja podataka sa mreže, budu:

- Prima CAN poruke sa podacima („*Data frame*“).
- Šalje CAN poruke sa „zahtevom za podatke“ („*Remote frame*“) u CAN-*bus* mrežu.
- Dobijene podatke analizira i obradi predefinisanim alatima mašinskog učenja metodama veštačke inteligencije.
- Izvrši predikciju ispravnih očitavanja MAF senzora na osnovu dobijenih informacija.
- Obrađene informacije internetom pošalje na udaljen server u bazu podataka tipa vremenskih serija (eng. *Time Series database*“).

Slika 15. "Infinity SCBA – Smart Can-bus Analyser "

Prvi problem koji je trebao da se reši je odabir hardverskih komponenti koje će po svojim dimenzijama moći da se uklope u dimenzije kućišta, a da se pri tome ne izgubi ni jedna od gorenavedenih funkcija.

Takođe, prilikom odabira hardverskih komponenti je moralo da se uzme u obzir i operativni sistem koji u svojoj osnovi ima ugrađene mogućnosti pristupa CAN-*bus* mreži preko specijalizovanog interfejsa serijske komunikacije. Problematika bi mogla da se definiše sledećim:

- Hardverska osnova sa operativnim sistemom.
- CAN kontroler sa RS485 serijskim protokolom.
- 3G/LTE Internet konekcija.
- Razvoj softvera za komunikaciju sa CAN-*bus* mrežom.

4.1.1 Hardverska osnova sa operativnim sistemom

Komponenta „*Infinity*“ uređaja koja je zadovoljila sve karakteristike potrebne za CAN komunikaciju je odabrana iz familije mini računara „*Raspberry Pi*“ [Slika 16] i to model „*Zero W*“ [25]. Još jedan od razloga zašto se odabrala ova komponenta kao osnova operativnog sistema je u tome što podržava GNU/Linux operativni sistem, a ujedno i u paleti dodataka ima gotovo rešenje RS485 serijskog kontrolora.

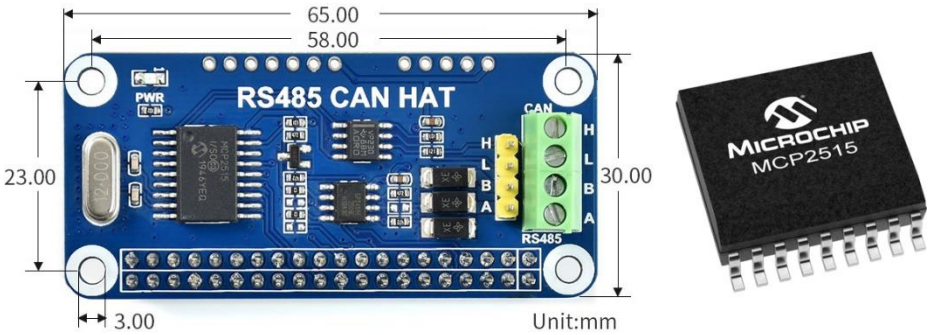
Slika 16. "Raspberry Pi“ model „Zero W“
izvor: https://www.raspberrypi.org/products/raspberry-pi-zero-w/

„*RPI Zero W*“ tehnička specifikacija:

- 802.11 b/g/n wireless LAN.
- Bluetooth Low Energy (*BLE*) 4.1.
- 1GHz, single-core CPU.
- 512MB RAM.
- Mini HDMI i USB „*On-The-Go*“ port.
- Micro USB napajanje.
- HAT-kompatibilno 40-pinsko zaglavlje (eng. *header*).
- CSI konektor za kameru.

4.1.2 CAN kontroler sa RS485 serijskim protokolom

Gotovo rešenje kompanije „*Waveshare*“ [26], omogućava stabilnu komunikaciju sa CAN-*bus* mrežom preko RS485/CAN funkcijama [Slika 17]. Mikrokontroler koji je zadužen za CAN komunikaciju je iz „*Microchip MCP251x*“ familije.



Slika 17. RS485 CAN HAT za Raspberry Pi
izvor: https://www.waveshare.com/rs485-can-hat.htm

4.1.3 3G/LTE Internet konekcija

Iako „*RPI Zero W*“ ima u sebi integrisan *WiFi*, potreba za internetom gde ne postoji bežična infrastruktura prijema interneta, je bila od velikog značaja za slanje prediktovanih podataka na udaljen server. Mogućnost integracije *3G/LTE* interneta je postignuto sa *USB 3G/LTE* uređajem kineske kompanije „*Technological&Computer Online Store*“ [27].

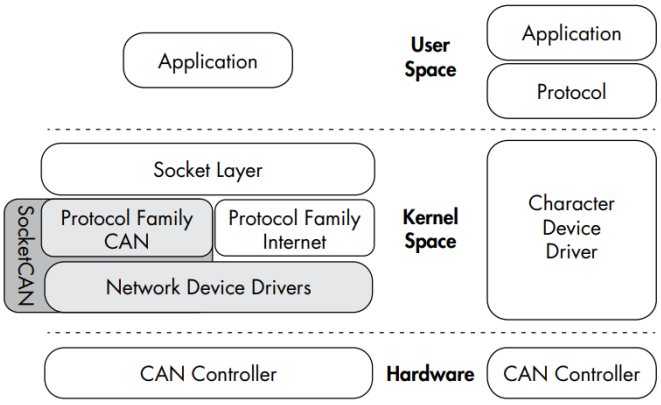
Slika 18. Wireless 3G Mobile Wifi Router Usb Dongle
izvor: https://www.aliexpress.com/item/4001120042109.html

4.1.4 Softver za komunikaciju sa CAN-*bus* mrežom

Operativni sistem pod nazivom „*Raspberry Pi OS Lite*“ [28], koji je instaliran na „*Raspberry Pi Zero W*“ uređaj, je iz porodice GNU/Linux Unix sistema. Obzirom na multifunkcionalnost uređaja, svaki sektor RAM memorije je bio potreban, pa je to jedan od najznačajnijih razloga odabira ovog minimalističkog operativnog sistema.

Posle povezivanja svih neophodnih hardverskih komponenti i instalacije operativnog sistema, bilo je potrebno inicijalizovati RS485/CAN interfejs drajverom otvorenog koda, pod nazivom „*SocketCAN*“[22]. *SocketCAN* drajver je integrisan u GNU/Linux kernel verziji počevši od „v2.6.25“ [Slika 19].

Prvu verziju *SocketCAN* drajvera razvijaju nemački entuzijasti otvorenog koda „*Open Source development BerliOS*[23]“ 2006. godine, a kasnije razvoj i poboljšanja preuzima „*Volkswagen Group Research*“ grupa. Paket iz ambrele alata „*SocketCAN*“ interfejsa koji je korišćen u ovoj master tezi je „*can-utils*“ [29] . Definisan je svojom paletom komandi koje služe za komunikaciju sa CAN-*bus* mrežom automobila.



Slika 19. "SocketCAN" interfejs GNU/Linux kernela

[22] „*SocketCAN*“ - *https://www.kernel.org/doc/Documentation/networking/can.txt*
[23] Open Source development BerliOS - *https://www.berlios.de/*

Programski jezik koji je korišćen u ovom istraživanju je kombinacija „*Python 3*“ [30] i GNU/Linux „*Bash Script*“ programskih jezika.

Osnovne komande „*SocketCAN*“ interfejsa, koje treba da se napomenu su:

- **candump**: Prikaz, filtriranje i evidentiranje CAN podataka u datoteke.
- **canplayer**: Ponavljanje CAN datoteke dnevnika.
- **cansend**: Slanje jednog CAN okvira.
- **cangen**: Generisanje slučajnog (eng. *random*) CAN saobraćaja.
- **cansequence**: Slanje i provera redosleda CAN okvira.
- **cansniffer**: Prikaz razlike u sadržaju CAN podataka.

Deo koda koji je implementiran u samom „*Infinity*“ uređaju za inicijalizaciju *RS485 CAN* kontrolora je dat u [Listing 1], dok je ceo kôd dat u [Listing (ceo kod) 1]:

```
os.system("sudo /sbin/ip link set can0 up type can bitrate 500000")
os.system("sudo ifconfig can0 txqueuelen 10000")
os.system("sudo ip link set up can0")

...
```

Listing 1. Inicijalizacija CAN kontroler RS485 i salnje poruka sa zahtevom za podatke

Deo koda koji je odgovoran za proveru ispravnog podizanja operativnog sistema i inicijalizaciju *RS485* je dat u [Listing 2], dok je ceo kod dat u [Listing (ceo kod) 2]. Ukoliko je došlo do greške, bilo u samom operativnom sistemu ili u inicijalizaciji CAN interfejsa, šalje se SMS poruka na broj koji je definisan poljem „*Number*“.

```
os.system("""sudo /opt/umtskeeper/umtskeeper --sakisoperators
"USBINTERFACE='0' OTHER='USBMODEM' USBMODEM='12d1:1001' SIM_PIN = ''
USBDRIVER='option' OTHER='USBMODEM' APN='CUSTOM_APN'
CUSTOM_APN='internet' APN_USER='telenor' APN_PASS='gprs'" --
sakisswitches " --sudo --console" --devicename 'Huawei' --nostats --
silent --monthstart 8 --nat 'no' &""")

message = {
            'Text': 'Infinity SCBA startovan!',
            'SMSC': {'Location': 1},
            'Number': '+38163XXXXXX',
        }

...
```

Listing 2. Provera operativnosti svih elementa i slanje SMS poruke

Deo koda koji je odgovoran za slanje CAN poruka sa zahtevom za podatke, opisan u [2.3.2], je prikazan u [Listing 3], dok je ceo kôd prikazan u [Listing (ceo kod) 3].

```
#!/bin/sh
targetbus="can0"

while :
do
   cansend $targetbus 7DF#02010C0000000000

...
```

Listing 3. Slanje CAN poruke sa zahtevom za podatke

4.2 Uslovi generisanja i uvid u strukturu skupa podataka

Kako je u uvodnom delu kratko opisan OBD II standard i pomenuta „paleta“ protokola pod kojom je implementiran, u ovom radu se uzima u obzir samo CAN protokol, kao primarni vid načina komunikacije u CAN-*bus* mreži.

Svi evropski proizvođači automobila, počevši od 2006. godine, imaju obavezu da implementiraju CAN protokol, pa je to jedan od najbitnijih razloga zašto se odabrao ovaj protokol kao primarni vid komunikacionog kanala za generisanje skupa podataka. Strukturom i smernicama CAN protokola je i konstruisan „*Infinity*“ OBD II interfejs, pomoću koga su prikupljani podaci za analizu iz mreže vozila.

Prilikom prikupljanja podataka iz mreže automobila, a u svrhu predikcije ispravnosti senzora protoka vazduha, uslov koji je morao da se zadovolji je u kom stanju motor vozila mora da bude i način na koji se pristupa generisanju skupa podataka (eng. *dataset*).

Ovim predefinisanim uslovima se i vodio ceo postupak prikupljanja, analize i predikcije MAF senzora, za koji bi trebalo da se zadovolje sledeća stanja:

Osnovni uslov

U[1]: K*omunikaciona mreža automobila, kao subjekat istraživačkog rada, mora da ima implementiran CAN protokol OBD II standarda*.

U[2]: *Automobil je u ispravnom stanju i da se prilikom dijagnostike utvrdi ispravnost motora i sistema za prečišćavanje izduvnih gasova.*

U[3]: *Kontrolna lampica ispravnosti motora (eng.* ***M****alfunction* ***I****ndicator* ***L****amp -* ***MIL****)*[24] *mora da bude ugašena, što predstavlja još jedan indikator ispravnosti motora.*

U[4]: *Sistem usisa vazduha u motoru automobila mora da bude u potpuno ispravnom stanju i da ne postoje otvori koji smanjuju pritisak u sistemu usisa.*

U[5]: *Diferencijalni pritisak lambda sondi u filterima za prečišćavanje (Katalizator, DPF*[25]*) mora da bude u dozvoljenim parametrima, propisanim po tehničkim specifikacijama proizvođača automobila.*

U[6]: *Nadmorska visina ulazi u parametarske osnove prilikom prikupljanja i analize podataka.*[26]

U[7]: *Postupak* p*rilikom prikupljanja podataka sa mreže se obavlja kada je automobil u stanju mirovanja (tzv. ler gas) i postepeno se dodaje gas.*[27]

U[8]: *Alati i metode mašinskog učenja veštačke inteligencije moraju da budu identično parametarizovani prilikom treninga i predikcije senzora protoka vazduha.*

U[9]: *Prilikom prikupljanja podataka i „real-time“ analize, motor mora da ima konstantnu radnu temperaturu od ≈90°C.*

---

[24] **M**alfunction **I**ndicator **L**amp (***MIL***) – *Lampica koja se još naziva i provera motora (eng. Check Engine).*
[25] DPF – *Filter čestica kod dizel motora (*eng. ***D****iesel* ***P****articulate* ***F****ilter).*
[26] Preko 1.500 mnv. usled razređenosti vazduha, podaci iz senzora se koriguju po smernicama ECUa.
[27] Neki automobili imaju sigurnosne mehanizme pa se maksimalan broj obrtaja motora ograničava na 2.500 obrtaja u minuti (eng. *RPM -* ***R****evolutions* ***P****er* ***M****inute*).

4.2.1 Struktura skupa podataka (*dataset*)

CAN protokol, iako standardizovan (SAE 11898) [15], nije otvorio veliku mogućnost za entuzijaste-istraživače uvid u isti, jer je svaki proizvođač automobila interpretirao i implementirao CAN protokol na svoj način, te iz tog razloga se došlo do apsurdne situacije da standardizovan protokol nije bio standardan. Problem se sastojao u tome da zaglavlja u slanju poruka kroz mrežu automobila nisu bila jedinstvena za svaku marku, tip i model određenog proizvođača vozila.

Standardna zaglavlja poruka sa zahtevom za podatke (*OBD-II PIDs*) [13], po kojima se dobija uvid o sistemu za prečišćavanje vazduha kod automobila, otvorila su mogućnost da se ipak standardizuje mali deo CAN protokola i omogući razvijanje raznih aplikacija koje nemaju veze sa kontrolom izduvnih gasova, a koriste standardna zaglavlja zahteva (u daljem tekstu: PID).

CAN-*bus* mreža je „šumovita“, dok je maksimalan protok 1MB/s obično brzina na kojoj se odvija komunikacija ne prelazi 500KB/s. Svi senzori, aktuatori i računari automobila šalju informacije o svom statusu, stanju, vrednostima, sve vreme. Senzori koji po svojoj prirodi generišu linerane vrednosti, diskretizuju se i kao takve šalju u mrežu. Krajnji nodovi/računari u automobilu imaju uvid u sve podatke i samo one koje su esencijalni i potrebni za rad, prioretizuju i „izvlače“ kao informaciju.

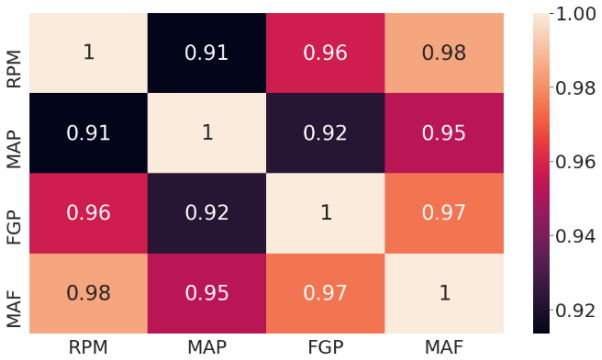
Status informacije podataka koji je glavni motorni računar (u daljem tekstu: ECU) „izvukao“ iz CAN-*bus* mreže je upotrebljen za generisanje skupa podataka, a u svrhu predikcije ispravnosti rada MAF senzora.

Generisan skup ima strukturu informacija od tri nezavisna atributa (eng. *Independent Variable*) [31] [32], dobijenih od ECU računara i to od senzora:

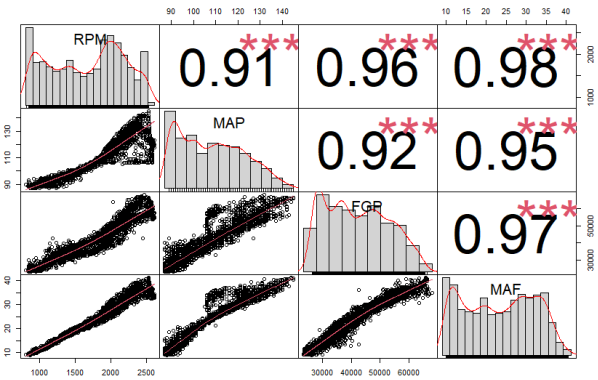
- **RPM** – Senzor obrtaja-brzine motora (eng. ***R**evolutions **P**er **M**inute*).
- **MAP** –Senzor apsolutnog pritiska usisne grane (eng. ***M**anifold **A**bsolute **P**ressure*).
- **FGP** – Pritisak goriva (eng. ***F**uel **G**auge **P**ressure*).

Vrednosti odabranog predikcionog atributa, zavisne promenjive (eng. *Dependent Variable*) su dobijene od MAF senzora koji je zadužen za merenje mase protoka vazduha, a dobijene vrednosti senzora se preslikavaju u količinu „usisanog“ vazduha u motor automobila, kao što je i detaljno opisano u poglavlju [3].

Sprega nezavisnih atributa (*RPM*, *MAP*, *FGP*) i zavisne varijable (*MAF*), se jasno može videti u korelacionoj matrici na [Slika 20], [Slika 21].

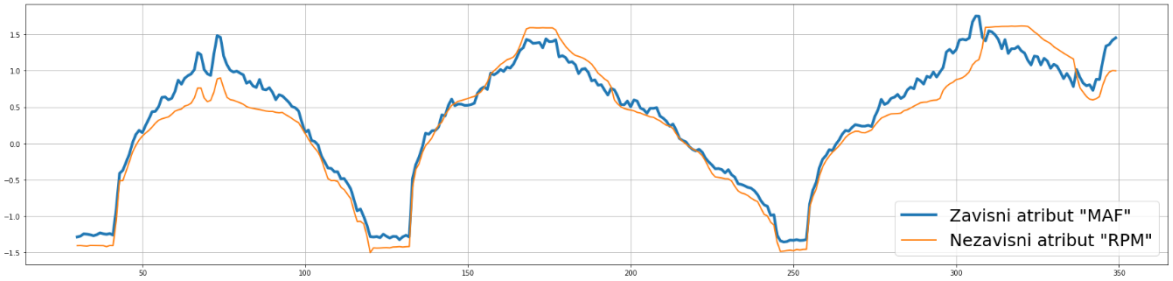


Slika 20. Korelaciona matrica zavisnih i nezavisnog atributa (MAF) - Python-stil

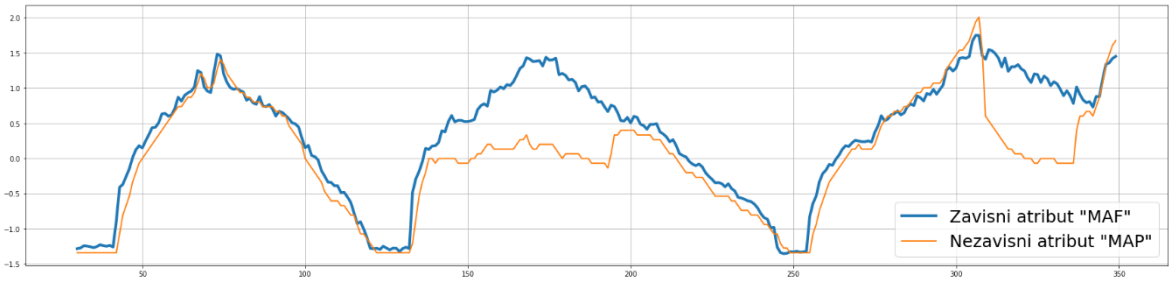


Slika 21. Korelaciona matrica zavisnih i nezavisnog atributa (MAF) - R-stil

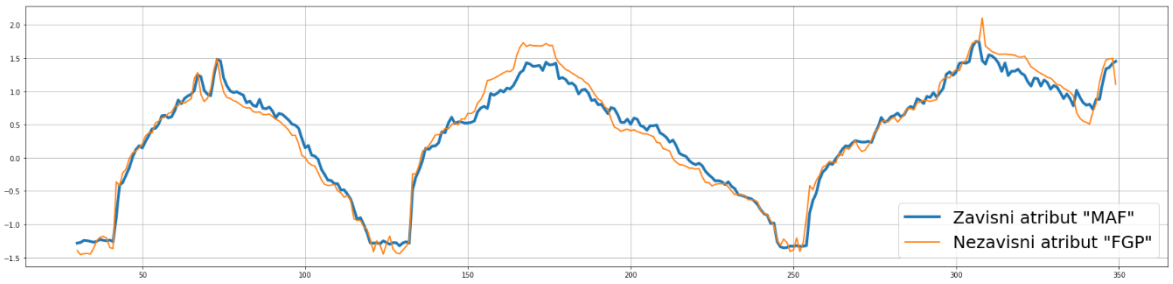
Uporedne analize linearnosti nezavisnih atributa RPM, MAP, FGP i zavisnog MAF atributa, su grafički prikazane na slikama: [Slika 22], [Slika 23], [Slika 24] i [Slika 25].



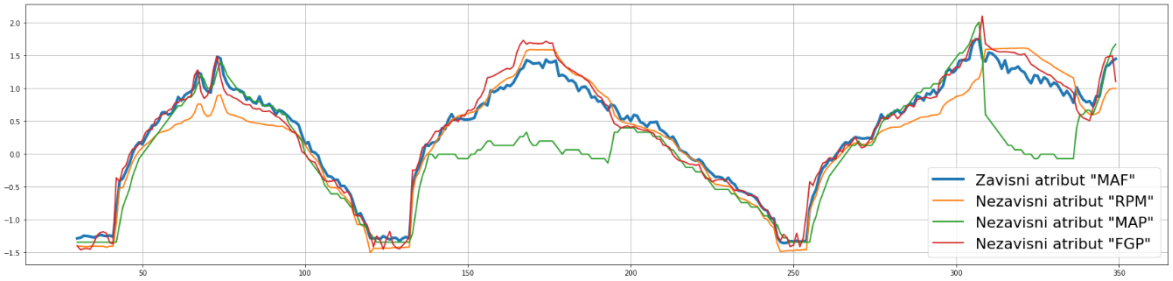
Slika 22. Linearnost zavisnog atributa MAF i nezavisnog atributa RPM



Slika 23. Linearnost zavisnog atributa MAF i nezavisnog atributa MAP



Slika 24. Linearnost zavisnog atributa MAF i nezavisnog atributa FGP



Slika 25. Linearnost zavisnog atributa MAF i nezavisnih atributa RPM, MAP, FGP

4.3 Izbor metoda mašinskog učenja

Problem mašinskog učenja je regresionog tipa i to sa podacima koji su linearno povezani. Potrebno je napomenuti da trerning podaci nisu veliki, jer u predloženoj metodi je dovoljan mali broj uzoraka, pa je trening od deset minuta na ispravnom vozilu, sasvim dovoljan za buduću predikciju neispravnosti očitavanja vrednosti MAF senzora.

Podaci nad kojima se vršio trening i evaluacija u ovom radu sadrže 2.759 uzoraka dobijenih iz CAN-*bus* mreže, od kojih su 80% trening skup (2.207) i 20% test skup (552). Deo skupa podataka je prikazan u [Listing (ceo kod) 4], dok je ceo skup podataka postavljen na repozitorijumu [33].

Mašinsko učenje pravilima regresija vremenskih serija se mogu posmatrati kao interpretacija prepoznavanja zavisnosti nezavisnih atributa u prikupljenim podacima .

Induktivno učenje na osnovu dobijenih primera, posle primene metoda mašinskog učenja, preslikava vremenske sekvence u predikcioni atribut kojim se dalje vrši predviđanje i provera budućih ulaznih vrednosti MAF senzora.

Obzirom na linearnu povezanost zavisnog atributa u ovom istraživačkom radu se primenjuju i evaluiraju metode mašinskog učenja regresione prirode i to:

- ***P***rosta ***L***inearna ***R***egresija (*PLR*)
- ***V***išestruka ***L***inearna ***R***egresija (*VLR*)
- ***R***andom ***F***orest ansambli (*RF*)

Linearna povezanost nam je otvorila mogućnost da se upotrebe alati mašinskog učenja iz palete regresionih predikcionih metoda. Ova činjenica je takođe omogućila implementaciju pomenutih metoda u sam „*Infinity*“ uređaj, koji je ograničenih računarskih resursa.

Dobijeni rezultati regresionih tehnika su evaluirani sa tri osnovne statističke metode:

- Srednje kvadratne greška (eng. ***Mean Squared Error*** – ***MSE***) (12).
- Korena srednje kvadratne greške (eng. ***Root Mean Squared Error*** – ***RMSE***) (13).
- Srednje apsolutne greške (eng. ***Mean Absolute Error*** – ***MAE***) (14).

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2 \tag{12}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2} \tag{13}$$

$$MAE = \frac{1}{N}\sum_{j=1}^{N}|y_j - \hat{y}_j| \tag{14}$$

Gde su:

- $\boldsymbol{N}$: Skup vrednosti predikcija.
- $\boldsymbol{\hat{y}_j}$: Vektor od N vrednosti predikcija.
- $\boldsymbol{y_j}$: Vektor koji predstavlja N tačnih vrednosti za data merenja.

4.3.1 Prosta Linearna Regresija (*PLR*)

Opšti oblik modela Proste Linearne Regresije (PLR) prema [34], možemo predstaviti kao (15):

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \qquad i = 1,2,\ldots,N \tag{15}$$

Gde su:

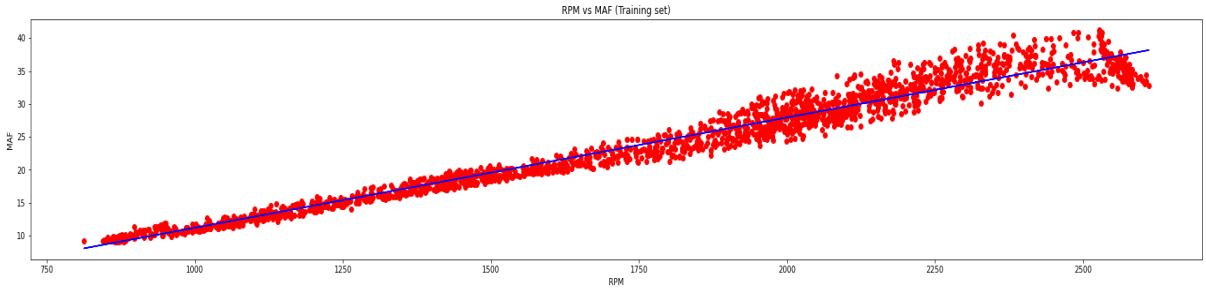
- $\boldsymbol{Y_i}$: Zavisna promenjiva.
- $\boldsymbol{x_i}$: Vrednost nezavisnog atributa.
- $\boldsymbol{\beta_0}, \boldsymbol{\beta_1}$: Nepoznate konstante (regresioni parametri).
- $\boldsymbol{N}$: Veličina osnovnog skupa atributa.

Najbolji kandidat nezavisnog atributa PLR-a se dobija uvidom u korelacionu matricu [Tabela 5]. Najsnažnija korelacija MAF-a su očitavanja RPM senzora.

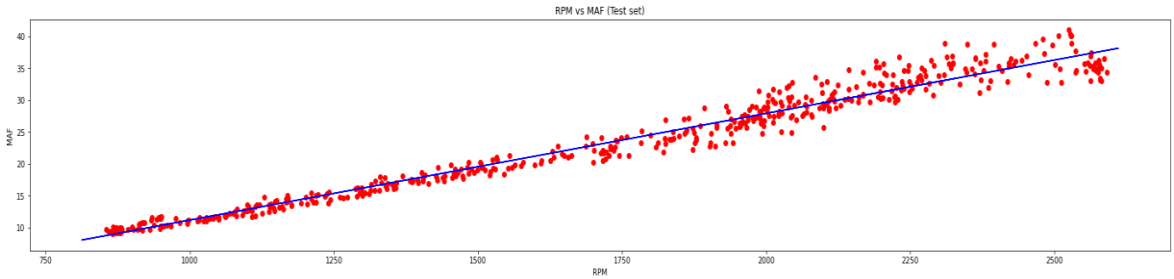
	RPM	**MAP**	**FGP**	**MAF**
RPM	1.000000	0.913430	0.957484	**0.984641**
MAP	0.913430	1.000000	0.922442	0.953109
FGP	0.957484	0.922442	1.000000	0.974711
MAF	**0.984641**	0.953109	0.974711	1.000000

Tabela 5. Korelaciona matrica RPM, MAP, FGP, MAF zavisnosti atributa PLR

Grafički prikaz linearnosti RPM i MAF senzora se vidi na [Slika 26] i [Slika 27].



Slika 26. RPM i MAF linearnosti atributa (trerning set)



Slika 27. RPM i MAF linearnosti atributa (test set)

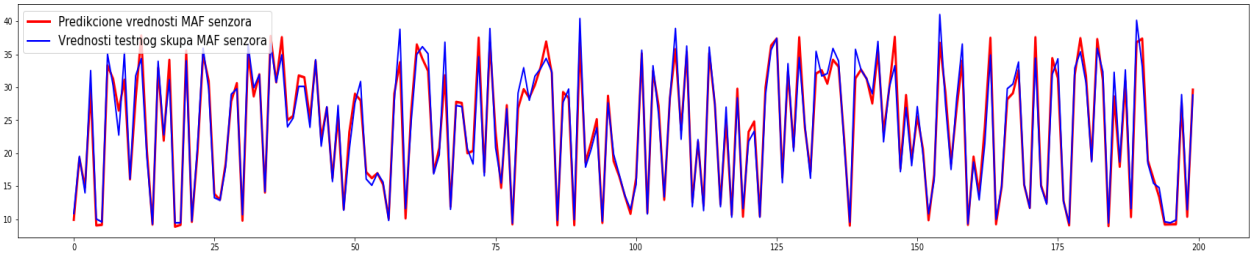
Evaluacija i grafički uporedni prikaz dobijenih rezultata proste linearne regresije, nakon primene metoda mašinskog učenja PLR-a, dobijeni su sledeći rezultati pomenutim metodama ispitivanja grešaka nad testnim skupom (552 uzorka) [Slika 28]:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.971
Model:                            OLS   Adj. R-squared:                  0.970
Method:                 Least Squares   F-statistic:                 1.811e+04
Date:                Tue, 25 May 2021   Prob (F-statistic):               0.00
Time:                        20:25:25   Log-Likelihood:                -1033.9
No. Observations:                 552   AIC:                             2072.
Df Residuals:                     550   BIC:                             2080.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0412      0.183     -0.224      0.822      -0.401       0.319
x1             1.0014      0.007    134.583      0.000       0.987       1.016
==============================================================================
Omnibus:                       30.209   Durbin-Watson:                   2.119
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               50.143
Skew:                           0.393   Prob(JB):                     1.29e-11
Kurtosis:                       4.249   Cond. No.                         67.4
==============================================================================
```

Slika 28. OLS Evaluacija naučenog modela Proste Linearne Regresije (Testni skup)

- Srednja kvadratna greška nad testnim skupom (***MSE***) = **2.48**
- Koren srednje kvadratne greške nad testnim skupom (***RMSE***) = **1.57**
- Srednja apsolutna greška (***MAE***) = **1.15**
- R-squared ($\boldsymbol{R^2}$) = **0.971**

Vizualizacija rezultata PLR nad testnim setom predikcione i stvarne vrednosti MAF senzora je prikazan na [Slika 29]. Uzorak je prikazan nad 200 okvira testnog skupa podataka.



Slika 29. Vizualizacija prediktovanih rezultata PLR nad testnim setom MAF senzora

Sadržaj programskog koda, koji implementira prostu linearnu regresiju, se nalazi u [Listing (ceo kod) 5. *Prosta Linearna Regresija - Python*], dok se kompletan kôd nalazi na „*Github*“ repozitorijumu [35].

4.3.2 Višestruka Linearna Regresija (*VLR*)

Opšti oblik modela višestruke linearne regresije prema [36] (u daljem tekstu: VLR) možemo predstaviti kao (16):

$$Y_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_1 x_{p-1} x_{i,p-1} + \varepsilon_i \qquad \acute{\imath} = 1,2,\ldots,N \tag{16}$$

Gde su:

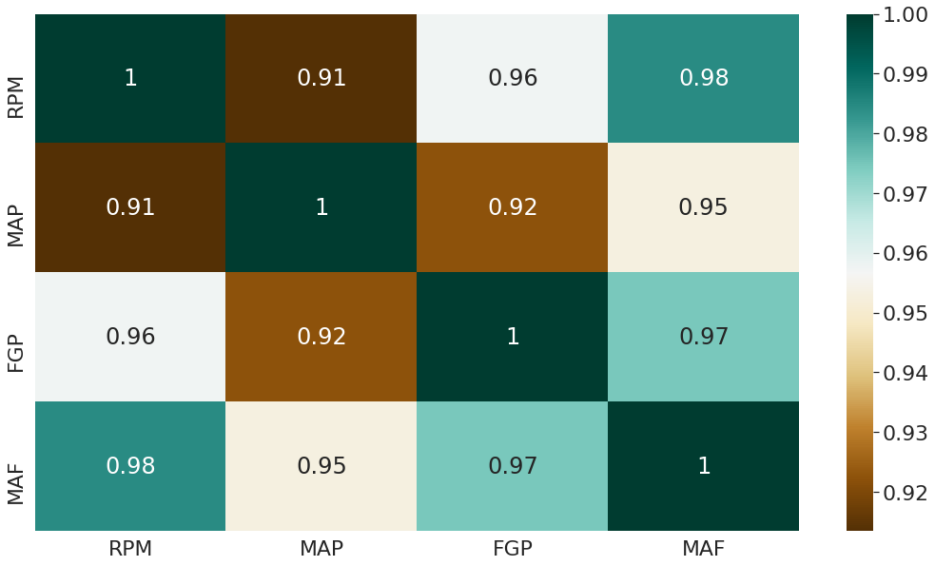
- $\boldsymbol{Y_i}$: Zavisna promenjiva.
- $\boldsymbol{x_i}$: Vrednost nezavisnog atributa.
- $\boldsymbol{\beta_0}, \boldsymbol{\beta_1}$: Nepoznate konstante (regresioni parametri).
- $\boldsymbol{x_{x,k}}$: Predstavlja í-tu vrednost k-tog prediktora za í=1,...,n
- $\boldsymbol{N}$: Veličina osnovnog skupa atributa.

VLR po prirodi uzima više nezavisnih atributa, koji se mogu videti u korelacionoj matrici urađenoj nad celokupnim skupom podataka. Najbolji kandidati nezavisnih atributa VLR-a se dobijaju uvidom u [Tabela 6] korelacionu matricu.

	RPM	**MAP**	**FGP**	**MAF**
RPM	1.000000	0.913430	0.957484	**0.984641**
MAP	0.913430	1.000000	0.922442	**0.953109**
FGP	0.957484	0.922442	1.000000	**0.974711**
MAF	**0.984641**	**0.953109**	**0.974711**	1.000000

Tabela 6. Korelaciona matrica RPM, MAP, FGP, MAF zavisnosti atributa VLR

Kao ulazni nezavisni atributi VLR-a, uzete su u obzir vrednosti očitavanja senzora RPM, MAP i FGP.



Slika 30. Korelaciona matrica RPM, MAP, FGP, MAF atributa

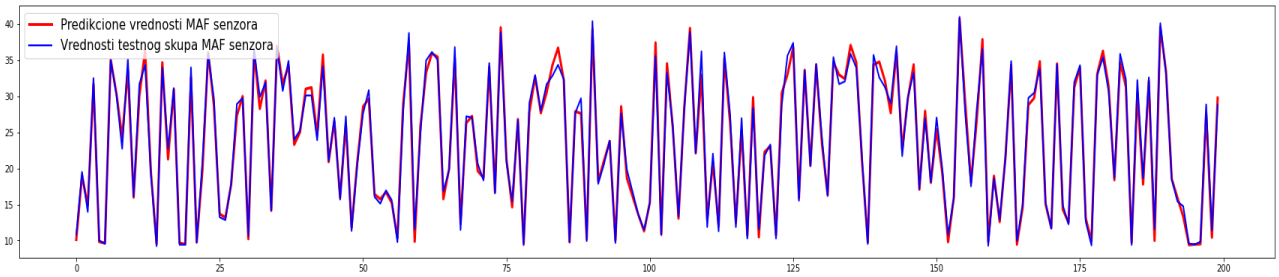
Evaluacija i grafički uporedni prikaz dobijenih rezultata višestruke linearne regresije, nakon primene metoda mašinskog učenja VLR-a, dobijeni su sledeći rezultati pomenutim metodama ispitivanja grešaka nad testnim skupom (552 uzorka) [Slika 31].

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.991
Model:                            OLS   Adj. R-squared:                  0.991
Method:                 Least Squares   F-statistic:                 6.137e+04
Date:                Tue, 25 May 2021   Prob (F-statistic):               0.00
Time:                        15:38:11   Log-Likelihood:                -702.93
No. Observations:                 552   AIC:                             1410.
Df Residuals:                     550   BIC:                             1418.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0015      0.100      0.015      0.988      -0.194       0.197
x1             1.0003      0.004    247.727      0.000       0.992       1.008
==============================================================================
Omnibus:                        7.446   Durbin-Watson:                   2.044
Prob(Omnibus):                  0.024   Jarque-Bera (JB):                7.591
Skew:                           0.232   Prob(JB):                       0.0225
Kurtosis:                       3.340   Cond. No.                         66.7
==============================================================================
```

Slika 31.OLS Evaluacija naučenog modela Višestruke Linearne Regresije (Testni skup)

- Srednja kvadratna greška nad testnim skupom (***MSE***) = **0.75**
- Koren srednje kvadratne greške nad testnim skupom (***RMSE***) = **0.86**
- Srednja apsolutna greška (***MAE***) = **0.67**
- R-squared ($\boldsymbol{R^2}$) = **0.991**

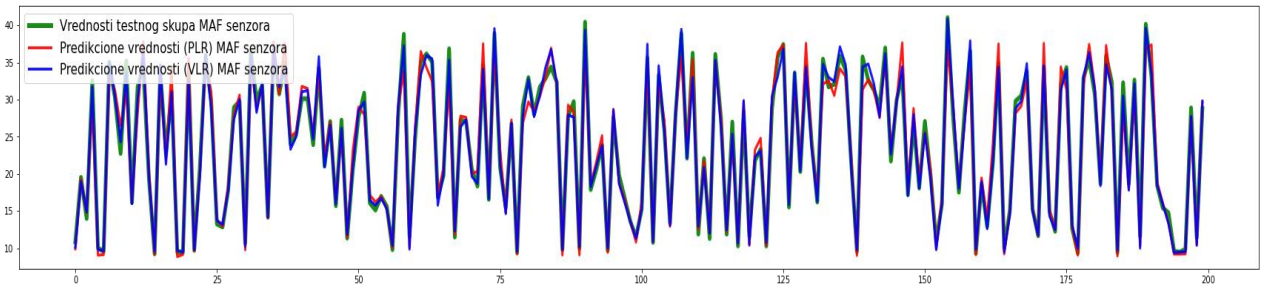
Vizualizacija rezultata VLR-a nad testnim setom predikcione i stvarne vrednosti MAF senzora je prikazan na [Slika 32]. Uzorak je prikazan nad 200 okvira testnog skupa podataka.



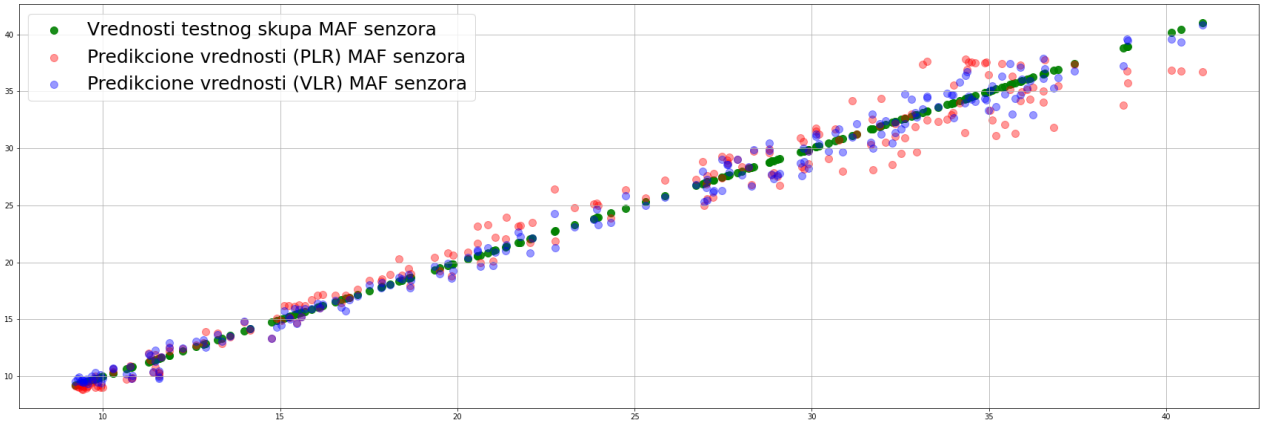
Slika 32. Vizualizacija prediktovanih rezultata VLR nad testnim setom MAF senzora

Sadržaj programskog koda koji implementira višestruku linearnu regresiju se nalazi u [Listing (ceo kod) 6. *Višestruka Linearna Regresija - Python*], dok se kompletan kôd nalazi na „*Github*“ repozitorijumu [35].

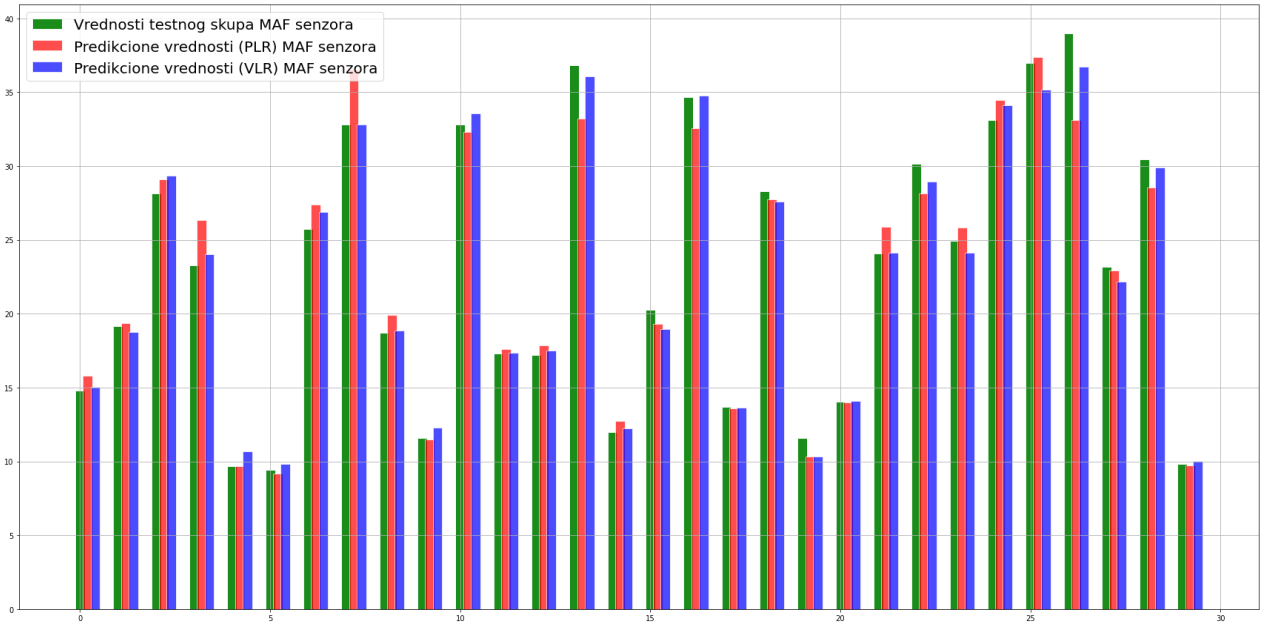
Grafička uporedna analiza PLR i VLR modela jasno ukazuje na poboljšanja predikcije koristeći VLR metode mašinskog učenja, nad metodom proste linearne regresije: [Slika 33], [Slika 34] i [Slika 35].



Slika 33. Grafička uporedna analiza metoda PLR i VLR sa testnim MAF podacima („Graf plot“)



Slika 34. Grafička uporedna analiza metoda PLR i VLR sa testnim MAF podacima ("Scater plot ")



Slika 35. Grafička uporedna analiza metoda PLR i VLR sa testnim MAF podacima ("Bar plot")

4.3.3 „*Random Forest*“ ansambli (*RF*)

RF metoda mašinskog učenja [37] je u ovom radu implementirana iz biblioteke „*Scikit-learn*“ [38], koja u svojoj implementaciji koristi „Gini značaj“ (eng. *Gini Importance*) i može da se predstavi u svom osnovnom obliku binarnog stabla kao (20) izvedenog iz (17), (18) (19):

$$ni_j = w_j C_j - w_{levo(j)} C_{levo(j)} - w_{desno(j)} C_{desno(j)} \tag{17}$$

Gde su:

- $\boldsymbol{ni_j}$: važnost čvora (eng. *nod*);
- $\boldsymbol{w_j}$: težinski broj samplova koji je dostigao čvor j;
- $\boldsymbol{C_j}$: vrednosti šuma tačke j;
- $\boldsymbol{levo(j)}$: levo dete u granama drveta;
- $\boldsymbol{desno(j)}$: desno dete u granama drveta.

Važnost atributa obeležija u svakoj grani se onda može predstaviti kao:

$$fi_i = \frac{\sum_{j:čvora\ j\ deljenje\ obeležija\ i} ni_j}{\sum_{k \in svim\ čvorovima} ni_k} \tag{18}$$

Gde se članovi mogu opisati kao:

- $\boldsymbol{fi_i}$: važnost obeležija i;
- $\boldsymbol{ni_j}$: važnost čvora j.

Normalizacija važnosti obeležija u granicama 0 … 1:

$$normfi_i = \frac{fi_i}{\sum_{k \in svim\ obeležijima} fi_j} \tag{19}$$

Konačna važnost obeležija se onda može predstaviti ka prosek svih grana stabala.

$$RFfi_i = \frac{\sum_{k \in svim\ granama} normfi_{ij}}{T} \tag{20}$$

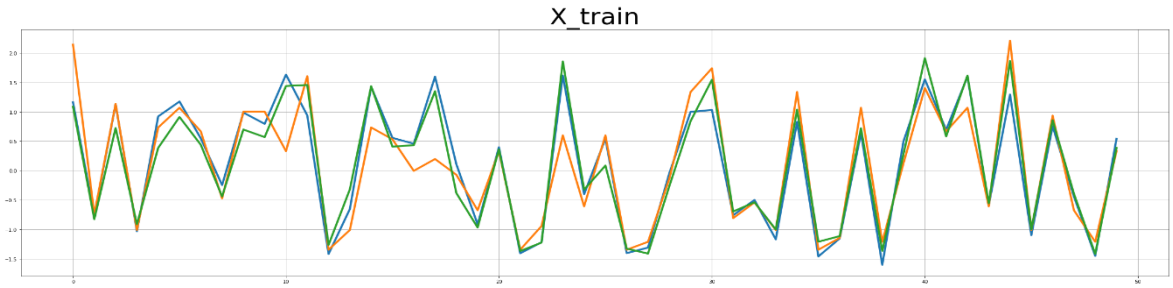
Gde su:

- $\boldsymbol{RFfi_i}$: važnost obeležija u random forest anasamblu;
- $\boldsymbol{normfi_{ij}}$: normalizovana vrednost obeležija za i, za stablo j;
- $\boldsymbol{T}$: ukupan broj stabala.

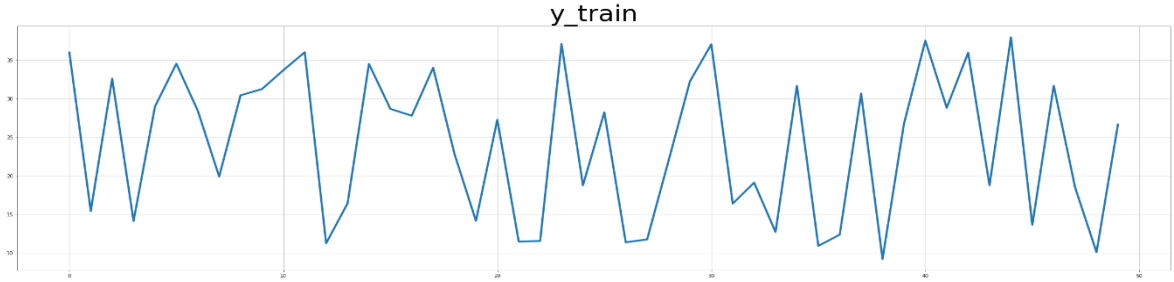
Detalji skupa su dati u [Tabela 7], dok su grafički prikazi trening i test skupa podataka dati na slikama: [Slika 36], [Slika 37], [Slika 38] i [Slika 39].

	RPM	MAP	FGP	MAF
Broj uzoraka	2759	2759	2759	2759
Srednja vrednost	1704.69	107.03	41018.61	22.99
Minimum	812.50	87.00	23310.00	9.00
25%	1203.62	93.00	30320.00	14.38
50%	1785.50	106.00	40190.00	23.30
75%	2146.75	119.00	50630.00	30.97
Maksimum	2611.50	145.00	68150.00	41.19

Tabela 7. Numerički detaljan opis skupa podataka: RPM, MAP, FGP i MAF



Slika 36. Trening set nezvisnih atributa skupa podataka: RPM, MAP i FGP



Slika 37. Trening set zvisnog atributa skupa podataka: MAF

Slika 38. Test set nezvisnih atributa skupa podataka: RPM, MAP i FGP

Slika 39. Test set zvisnog atributa skupa podataka: MAF

Kao i kod VLR metode mašinskog učenja, uzete su u obzir RPM, MAP i FGP nezavisni atributi kao ulazni parametri učenja RF modela.

Parametri RF regresora koji su uzeti u obzir na trening setu su:

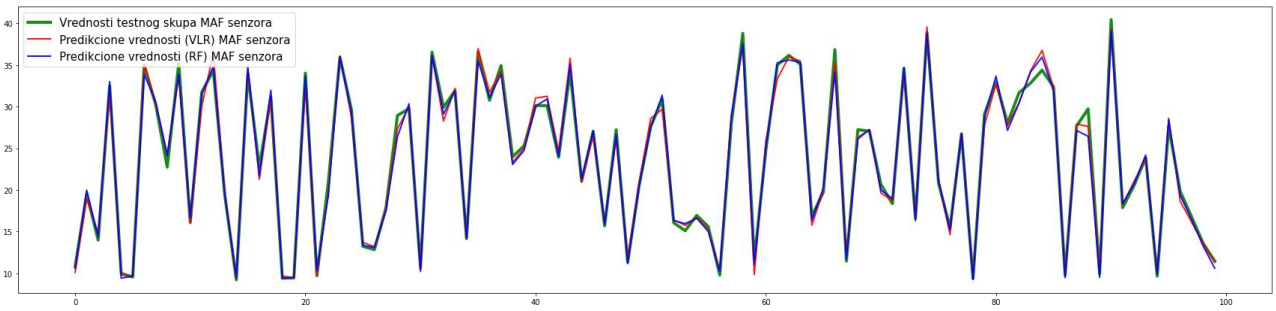
- *bootstrap = True*
- *n_estimators = 100*

Evaluacija [Slika 40] i grafički uporedni prikaz [Slika 41] dobijenih rezultata, nakon primene metoda mašinskog učenja „*Random Forest*“ ansambla, dobijeni su sledeći rezultati pomenutim metodama ispitivanja grešaka nad testnim skupom od 552 uzorka.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.994
Model:                            OLS   Adj. R-squared:                  0.994
Method:                 Least Squares   F-statistic:                 8.581e+04
Date:                Tue, 25 May 2021   Prob (F-statistic):               0.00
Time:                        23:22:32   Log-Likelihood:                -611.09
No. Observations:                 552   AIC:                             1226.
Df Residuals:                     550   BIC:                             1235.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0306      0.084     -0.363      0.717      -0.196       0.135
x1             1.0008      0.003    292.939      0.000       0.994       1.007
==============================================================================
Omnibus:                      165.522   Durbin-Watson:                   1.919
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1192.348
Skew:                           1.116   Prob(JB):                    1.22e-259
Kurtosis:                       9.845   Cond. No.                         66.8
==============================================================================
```

Slika 40. OLS Evaluacija naučenog modela „Random Forest“ ansambla (Testni skup)

- Srednja kvadratna greška nad testnim skupom (***MSE***) = **0.53**
- Koren srednje kvadratne greške nad testnim skupom (***RMSE***) = **0.73**
- Srednja apsolutna greška (***MAE***) = **0.52**
- R-squared ($\boldsymbol{R^2}$) = **0.994**



Slika 41. Grafička uporedna analiza metoda VLR i RF sa testnim MAF podacima („Graf“)

Sadržaj programskog koda koji implementira „*Random Forest*“ ansamble se nalazi u [Listing (ceo kod) 7], dok se kompletan kôd nalazi na „*Github*“ repozitorijumu [35].

4.3.4 Pregled dobijenih rezultata i njihova uporedna analiza

Analize i predikcije modela vremenskih regresija su vršene metodama: „proste linearne regresije“, „višestrukih linearnih regresija“ i metodom „*Random Forest*“ ansambla. Za merenje performansi prediktora su se uzimali matematičko statistički parametri: srednje kvadratne greške (*MSE*), korena srednje kvadratne greške (*RMSE*) i parametar srednje apsolutne greške (*MAE*). Takođe se upotrebljavala analiza OLS[28] statističkom metodom.

Svi rezultati su dobijeni pomoću programskog jezika „*Python 3*“, implementirani u „*Google*“ kolaborativnom okruženju (eng. *Google colab*[29]). Programski kôd je pohranjen na udaljeni repozitorijum za menadžment kontrole verzija (*Github*), na kom je, za potrebe ovog istraživanja, napravljen poseban repozitorijum [35].

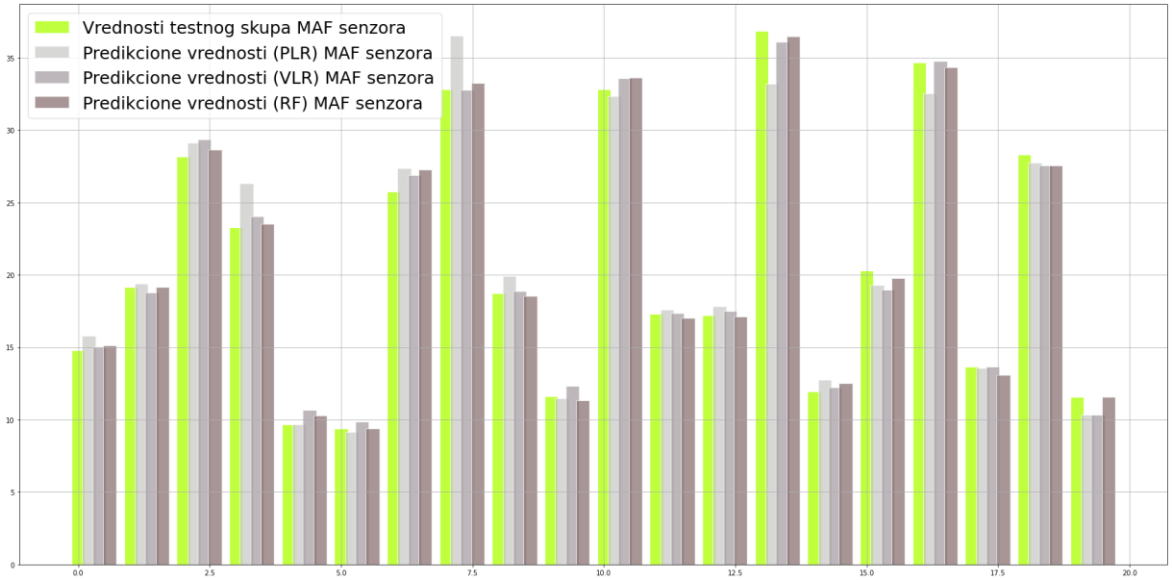
Generisani skup podataka nad kojim se vršila analiza i predikcija ispravnosti MAF senzora, je dobijena iz CAN-*bus* mreže automobila marke „*Volkswagen GOLF VI*“, sa motorom 1.6 dizel, proizvedenog 2012. godine. Prilikom generisanja skupa podataka, potrebno je zadovoljavanje predefinisanih uslova o stanju ispravnosti automobila. Ovi uslovi su detaljno opisani u poglavlju [4.2 „*Uslovi generisanja i uvid u strukturu skupa podataka*“]. Ceo skup podataka je takođe sačuvan na „*Github*“ repozitorijumu [33].

Analizom upotrebljenih metoda, najlošiji rezultati su dobijeni implementacijom proste linearne regresije, dok su najbolji rezultati dobijeni upotrebom „*Random Forest*“ ansambala.

Uporedni rezultati performansi koji su dobijeni u ovom istraživanju, predočeni su tabelarno u [Tabela 8] i grafički na [Slika 42].

	MSE	RMSE	MAE	R^2
PLR	2.48	1.57	1.15	0.971
VLR	0.75	0.86	0.67	0.991
RF	**0.53**	**0.73**	**0.52**	**0.994**

Tabela 8. Tabelarno uporedni evaluacioni rezultati regresija



Slika 42. Grafičko uporedni evaluacioni rezultati regresija

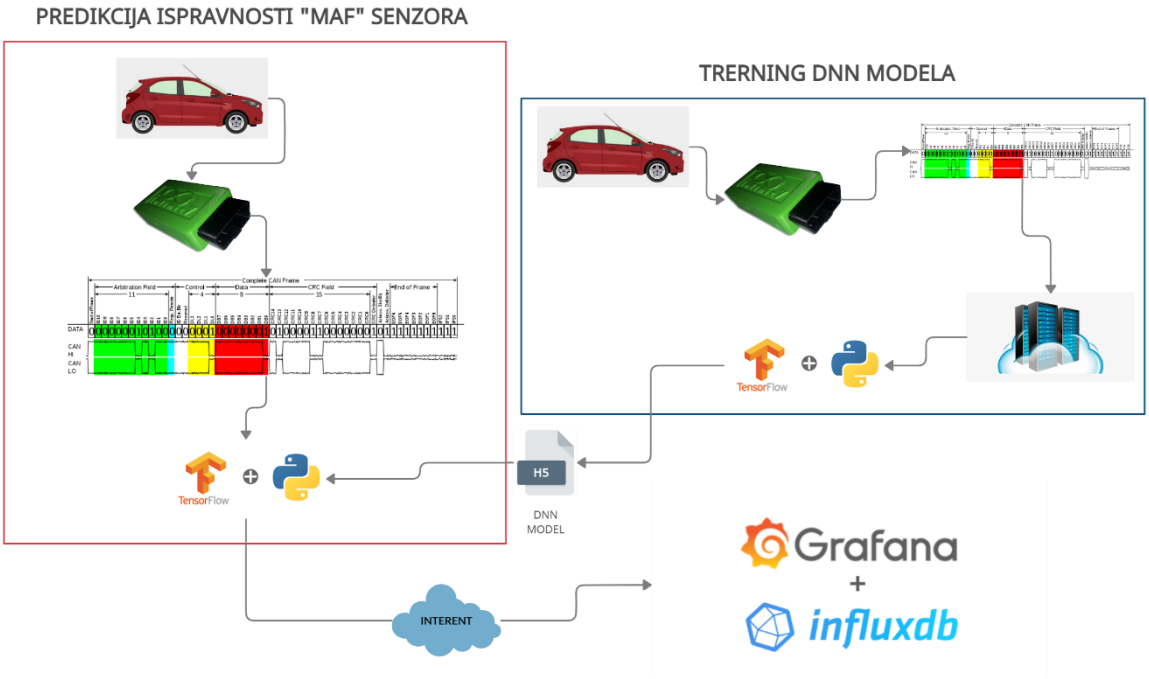
[28] OLS stats model - *https://www.statsmodels.org/stable/index.html*
[29] Google koloborativno okruženje - *https://research.google*

4.4 Praktična implementacija predikcije ispravnosti MAF senzora

4

Praktična primena ovog istraživanja je data kao četvrta tačka u topologiji ciklusa predikcije ispravnosti očitavanja MAF senzora. Završni deo predstavljenog ciklusa, koji kao takav opisuje integraciju sve tri ostale tačke u grafički uporedni prikaz dobijenih vrednosti [Slika 14].

Opšti pregled četvrte, završe tačke ciklusa, bi se mogao predstaviti kao na dijagramu sistema koji je prikazan na [Slika 43].



Slika 43. Dijagram grafičkog prikaza sistema za predikciju MAF senzora

Integracijom baze podataka vremenskih serija („*InfluxDB*“[30]) u sprezi sa grafičkim interfejsom otvorenog koda („*Grafana*“[31]), omogućena je prezentacija dobijenih komparativnih rezultata grafičkim putem, a ujedno je postignuta i globalnost, mogućnošću pristupa i pregleda dobijenih vrednosti preko interneta. Slanjem podataka multifunkcionalnim uređajem „*Infinity*“, se dobija i mogućnost monitoringa senzorskih sistema motora u realnom vremenu. Ovaj vid rešenja dobija na značaju u primerima kada servisni tehničar nije u mogućnosti da fizičkim prisustvom dobije uvid u ispravnost automobila. Slanje prediktovanih informacija, zajedno sa realnim (stvarnim) podacima, na „*Cloud*“ rešenje sistema predikcije MAF senzora, pruža upravo tu značajnu osobinu dobijanja uvida u ispravna očitavanja MAF senzora.

Za potrebe ove master teze implementirano je praktično rešenje, koje u realnom vremenu vrši predikciju ispravnosti MAF senzora.

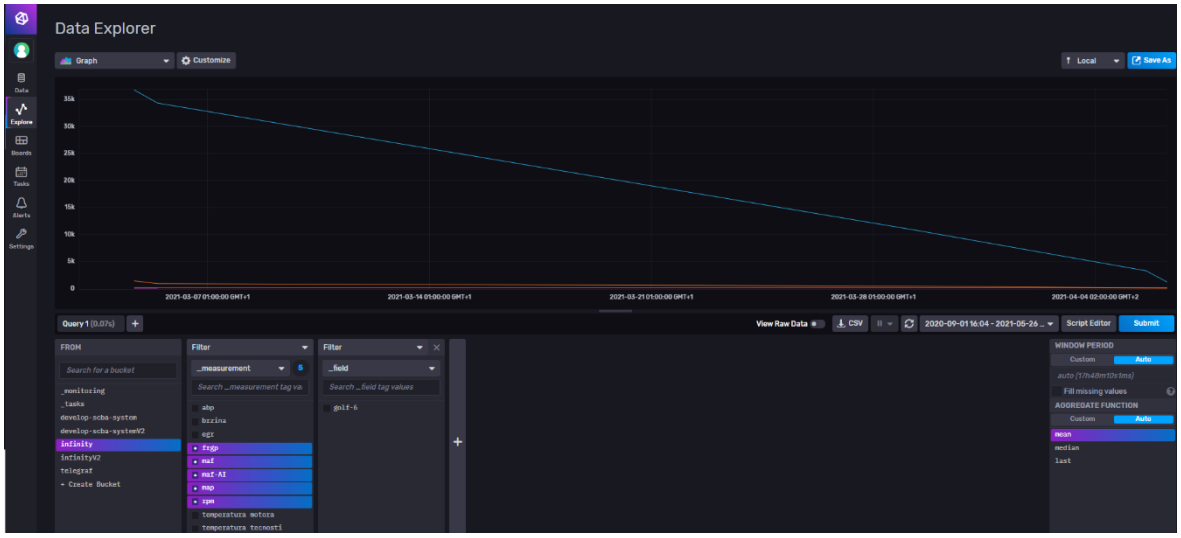
Inicijalizovan je server na kojem su instalirane obe komponente:

- Baza podataka „*InfulxDB*“.
- Grafički korisnički interfejs „*Grafana*“.

[30] „InfluxDB“ - *https://www.influxdata.com*
[31] „Grafana“ - *https://grafana.com*

Podaci koji se generišu su u obliku vremenskih serija su kao takvi u velikoj meri doprineli izboru baze podataka koja se koristila u ovom istraživanju. Takođe, status „otvorenog koda“ je dao smernicu koja je uperena ka bazi podataka vremenskih serija „*InfluxDB*“. Kreirana je 2013. godine od strane „*InfluxData*“ razvojnog tima i pisana je u „GO“[32] programskom jeziku. Neke od najznačajnijih osobina su:

- Visoke performanse (namenski za podatke oblika vremenskih serija).
- Visoke performanse HTTP(S) API-a.
- Upiti se pišu u obliku SQL upita.
- Omogućavanje indeksiranja podataka *Tag*-ovima.
- Manipulacija nad podacima je preko grafičkog interfejsa [Slika 44].



Slika 44. Prikaz grafičkog interfejsa: "InfluDB"

Analiza prediktovanih vrednosti, u četvrtom koraku predloženog ciklusa, je projektovana preko „*Grafana*“ grafičkog interfejsa [Slika 45]. Objavljena je pod licencom otvorenog koda, 2014. godine. Povezana sa „*InfluxDB*“ bazom podataka, predstavlja jedno od najprihvatljivijih rešenja grafičkog uvida i analize podataka.

Slika 45. Prikaz grafičkog interfejsa: "Grafana"

[32] „GO“ programski jezik - *https://golang.org*

5. Zaključna razmatranja

Eksponencijalni napredak u oblasti veštačke inteligencije daje ogromna poboljšanja u raznim aspektima kvaliteta ljudskih života, iako se u nekim trenucima nije toga ni svesno. Ova činjenica je doprinela da se kroz ovaj istraživački rad upotrebe alati mašinskog učenja za predikciju senzora mase protoka vazduha (MAF) u motoru automobila. Ukoliko se upotrebom opisanih tehnologija u ovom radu, ustanovi nepravilnost očitavanja MAF senzora, dolazi se do zaključka da nešto nije u redu sa nekim od sistema ili podsistema koji su povezani sa usisnim komponentama vazduha ili sistemom za prečišćavanje izduvnih gasova. Epilog ove nepravilnosti može da dovede do katastrofalnih kvarova na motoru automobila ali isto tako i na povećanu emisiju ispuštanja štetnih gasova. Ukoliko se zaključi da su prisutna odstupanja realnih podataka od prediktovanih vrednosti očitavanja MAF senzora, preventivnim akcijama servisnog tehničara, mogu da se spreče veći problemi sa motorom vozila, ali ujedno i da donekle dá sigurnost vlasniku u ispravnost svog vozila.

Tokom istraživanja i same implementacije predloženog sistema predikcije se došlo do zaključaka da je sistem veoma lako proširiti i na druge senzore, koji su takođe bitni za ispravan rad motora. Posebne zaključne indikacije se mogu usmeriti ka realizovanju dodatnih metoda autonomnosti, koje bi u svom obliku imale potpuno automatizovano obaveštavanje o neregularnosti očitanih podataka, bez intervencije servisnog tehničara.

Odluka da se upotrebe alati iz palete regresionih metoda mašinskog učenja je doprinela multifunkcionalnosti „*Infinity*“ uređaja koji je projektovan i izrađen namenski za ovo istraživanje. Funkcionalnosti poput, pristupa CAN-*bus* mreži, matematičkim konverzijama podataka, predikciji i slanju informacija putem interneta, se nisu mogle postići da su se upotrebljavale npr. duboke neuronske mreže, jer je prostor računarskih kapaciteta bio minimalan.

Podaci nad kojima se vršio trening i evaluacija u ovom radu sadrže 2.759 uzoraka dobijenih iz CAN-*bus* mreže, od kojih su 80% trening skup (2.207) i 20% test skup (552). Za generisanje ovog skupa podataka iz automobila je bilo potrebno nekoliko minuta, dok je regresiona linearnost generisanog skupa podataka nezavisnih atributa sa zavisnim, omogućila upotrebu tri regresione metode:

- ***P***rosta ***L***inearna ***R***egresija (*PLR*)
- ***V***išestruka ***L***inearna ***R***egresija (*VLR*)
- ***R***andom ***F***orest ansambli (*RF*)

Evaluacija grešaka nad dobijenim predikcionim vrednostima se vršila upotrebom matematičko statističkih metoda, i to:

- Srednje kvadratne greške (eng. ***M****ean* ***S****quared* ***E****rror* – ***MSE***).
- Korena srednje kvadratne greške (eng. ***R****oot* ***M****ean* ***S****quared* ***E****rror* – ***RMSE***).
- Srednje apsolutne greške (eng. ***M****ean* ***A****bsolute* ***E****rror* – ***MAE***).

Dobijeni rezultati, kao pokazatelji grešaka usled predikcije ispravnog očitavanja MAF senzora, ukazuju da je PLR imao najslabije rezultate, dok su predikcione vrednosti metode „*Random Forest*“ ansambala, dale najbolje rezultate.

Dodatna istraživanja bi trebalo da uključe integraciju sistema koji mogu da predvide, koji je tačno kvar uzrokovao pogrešna očitavanja. Takođe, u daljem radu bi trebalo isključiti posebne uslove koji treba da se zadovolje prilikom generisanje podataka opisane u poglavlju 4.2 „*Uslovi generisanja i uvid u strukturu skupa podataka*“.

Rezultati dobijeni evaluacijom grešaka nad dobijenim predikcionim vrednostima, ukazuje na činjenicu da se induktivnim učenjem nad trening skupom podataka dobijaju vrednosti koje mogu da ukažu na neispravna očitavanja MAF senzora. Ovo činjenica da dobijene vrednosti predstavljaju preslikavanje stvarnih očitavanja senzora, dokazuje osnovnu hipotezu „*H0*“, kojom, kroz pregled samih rezultata, može da se utvrdi ispravnost subjektnog automobila, uvid u ispravnost sistema prečišćavanja ispuštenih gasova vozila ali isto tako i uvid u ispravnost motora automobila.

Sklop usisa vazduha u motor automobila je u snažnoj korelaciji sa MAF senzorom, jer je upravo pozicioniran na samom sistemu usisa vazduha. Ukoliko su odstupanja prevelika, to dokazuje i pomoćnu hipotezu „*H1*“, da je sistem usisa „korumpiran“ i da je potrebno izvršiti tehnički pregled vozila.

Bitnom strateškom pozicijom i merenjem grešaka očitavanja MAF senzora se dobija informacija da li je turbina motora u ispravnom stanju. Disproporcija količine kompresije turbine dovodi do fluktacije u sistemu usisa, a samim tim i do neregularnosti u vrednostima dobijenih od MAF senzora. Ovim se dokazuje i pomoćna hipoteza „*H2*“.

Pomoćna hipoteza „*H3*“, je u korelaciji sa hipotezom „*H4*“, jer merenje diferencijalnog pritiska DPF filtera (kod dizel motora) i katalizatora (kod benzinskih motora), na ulazu i izlazu, se utvrđuje O_2 senzorom (lambda sonda). O_2 senzor se koristi kao jedan od parametara koji ECU koristi u svom algoritmu za poboljšanje dobijenih vrednosti MAF senzora, pa samim tim se i dokazuju pomoćne teoreme „*H3*“ i „*H4*“.

6. Literatura

[1] EPA, „U.S. Environmental Protection Agency," [Na mreži]. Dostpuno na: https://www.epa.gov/. [Poslednji pristup 5. 5. 2021.].

[2] CARB, „California Air Resources Board," [Na mreži]. Dostpuno na: https://ww2.arb.ca.gov/. [Poslednji pristup 5. 5. 2021.].

[3] „Society of Automotive Engineer," [Na mreži]. Dostpuno na: https://www.sae.org/. [Poslednji pristup 5. 5. 2021.].

[4] „International Organization for Standardization," [Na mreži]. Dostpuno na: https://www.iso.org/home.html. [Poslednji pristup 6. 5. 2021.].

[5] „Robert Bosch GmbH," [Na mreži]. Dostpuno na: https://www.bosch.com/. [Poslednji pristup 12. 5. 2021.].

[6] N. P. Srivastava, R. K. Srivastava i P. K. Vashishtha, „Fault Detection and Isolation (Fdi) Via Neural Networks," *Journal of Engineering Research and Applications,* t. 4, br. 1, pp. 81-86, 2014.

[7] F. Giobergia, „Mining Sensor Data for Predictive Maintenance in the Automotive Industry," u *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA),* Turin, 2018.

[8] U. Shafi, A. Safi, A. R. Shahid i S. Ziauddin, „Vehicle Remote Health Monitoring and Prognostic Maintenance System," *Journal of Advanced Transportation,* t. 2018, br. https://doi.org/10.1155/2018/8061514, 2018.

[9] S. Nowaczyk, R. Prytz, T. Rögnvaldsson i S. Byttner, „Towards a Machine Learning Algorithm for Predicting Truck Compressor Failures Using Logged Vehicle Data," *Twelfth Scandinavian Conference on Artificial Intelligence,* p. 205–214, 2013.

[10] A. Chaudhuri i S. R, „Predictive Maintenance for Industrial IoT of Vehicle Fleets using," D Institute Delhi, 2018.

[11] „SAE J1979," [Na mreži]. Dostpuno na: https://www.sae.org/standards/content/j1979_201702/. [Poslednji pristup 15. 5. 2021.].

[12] „SAE J1962," [Na mreži]. Dostpuno na: https://www.sae.org/standards/content/j1962_201607/. [Poslednji pristup 15. 5. 2021.].

[13] „OBD-II PIDs," [Na mreži]. Dostpuno na: https://en.wikipedia.org/wiki/OBD-II_PIDs. [Poslednji pristup 15. 5. 2021.].

[14] „SAE J1939," [Na mreži]. Dostpuno na: https://www.sae.org/standards/content/j1939_201808/. [Poslednji pristup 16. 5. 2021.].

[15] „ISO 11898," [Na mreži]. Dostpuno na: https://www.iso.org/standard/63648.html. [Poslednji pristup 17. 5. 2021.].

[16] „ISO 15765-4," [Na mreži]. Dostpuno na: https://www.iso.org/standard/67245.html. [Poslednji pristup 17. 5. 2021.].

[17] „ISO 14229-1," [Na mreži]. Dostpuno na: https://www.iso.org/standard/72439.html. [Poslednji pristup 17. 5. 2021.].

[18] „ISO 11898-2," [Na mreži]. Dostpuno na: https://www.iso.org/standard/67244.html. [Poslednji pristup 17. 5. 2021.].

[19] D. Vasiljević, „CAN PROTOKOL I NJEGOVA PRIMJENA," u *INFOTEH-JAHORINA Vol. 10, Ref. F-20, p. 988-992,* JAHORINA, 2011.

[20] T. I. Incorporated, „Introduction to the Controller Area Network (CAN),“ u *SLOA101B–August 2002–Revised May 2016*.

[21] „AC Reno Auto Servis,“ [Na mreži]. Dostpuno na: https//www.acreno.rs. [Poslednji pristup 20. 5. 2021.].

[22] W. J.H., A High-Performance Constant-Temperature Hot-Wire Anemometer, NASA Contractor Report 177645, 1994.

[23] Bruun i H.H., Principles and Signal Analysis, New York: Oxford University Press Inc., 2002.

[24] K. L.V, „Determination of the Convection Constants of Small Platinum Wires with Applications to Hot-Wire Anemometry,“ McGill University, 1914, pp. 373-432.

[25] „Raspberry Pi Zero W,“ [Na mreži]. Dostpuno na: https://www.raspberrypi.org/products/raspberry-pi-zero-w/. [Poslednji pristup 23. 5. 2021.].

[26] „RS485 CAN HAT For Raspberry Pi,“ [Na mreži]. Dostpuno na: https://www.waveshare.com/rs485-can-hat.htm. [Poslednji pristup 23. 5. 2021.].

[27] „Wireless 3G Mobile Wifi Router Usb Dongle,“ [Na mreži]. Dostpuno na: https://www.aliexpress.com/item/4001120042109.html. [Poslednji pristup 23. 5. 2021.].

[28] „Raspberry Pi OS Lite,“ [Na mreži]. Dostpuno na: https://www.raspberrypi.org/software/operating-systems/. [Poslednji pristup 23. 5. 2021.].

[29] „Can-Utils,“ [Na mreži]. Dostpuno na: https://github.com/linux-can/can-utils.

[30] „Python,“ [Na mreži]. Dostpuno na: https://www.python.org/. [Poslednji pristup 24. 5. 2021.].

[31] Guyon, Gunn i Z. Nikravesh, „Feature extraction, foundations and applications,“ 2006.

[32] Cherkassky i Mulier, Learning from Data: Concepts, Theory, and Methods, 2nd Ed, John Wiley - IEEE Press, 2007.

[33] „Dataset CAN poruka sa podacima,“ [Na mreži]. Dostpuno na: https://github.com/Dejan-Cugalj/CANBUS_dataset/blob/main/golf-master-canbus_dataset.csv. [Poslednji pristup 24. 5. 2021.].

[34] Altman i Krzywinski, „Simple linear regression. Nat Methods 12,“ 2015, p. 999–1000.

[35] „Master Teza "Github" repozitorijum - Python kod,“ [Na mreži]. Dostpuno na: https://github.com/Dejan-Cugalj/MASTER-TEZA. [Poslednji pristup 25. 5. 2021.].

[36] „Višestruka Linearna Regresija,“ [Na mreži]. Dostpuno na: http://www.matf.bg.ac.rs/p/files/69-Visestruka1.html. [Poslednji pristup 25. 5. 2021.].

[37] „Random Forests. Machine Learning. vol 45,“ doi:10.1023/A:1010933404324, 2001, p. 5–32.

[38] „Sklearn-ensemble RandomForestClassifier,“ [Na mreži]. Dostpuno na: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Poslednji pristup 26. 05. 2021.].

7. Dodatak

Listing 1.

```
import os
import time
from datetime import datetime

os.system("sudo /sbin/ip link set can0 up type can bitrate 500000")
os.system("sudo ifconfig can0 txqueuelen 10000")
os.system("sudo ip link set up can0")

now = datetime.now()

dt_string = now.strftime("%d-%m-%Y:%H:%M:%S")

print("*******INICIJALIZIVAN INFINITY: " + dt_string)
os.system("echo INFINITY STARTED: `date ""+%d.%m.%y:%H:%M:%S`"" >>
/home/pi/logs/system_logs/infinity_start.log 2>&1")


def getMAC(interface='wlan0'):
    # Vrati MAC adresu CAN mrežnog interfejsa
    try:
        str = open('/sys/class/net/%s/address' % interface).read()
    except:
        str = "00:00:00:00:00:00"
    return str[0:17]


os.system("candump -ta -L can0,7E8:7FF > /home/pi/logs/can_logs/log-dump." + dt_string +
'":"' + getMAC('wlan0') + " &")
```

Listing (ceo kod) 1. Inicijalizacija CAN kontrolora RS485 i salnje poruka sa zahtevom za podatke

Listing 2.

```
def sendSms():
    try:
        time.sleep(25)
        sm = gammu.StateMachine()
        sm.ReadConfig()
        sm.Init()

        message = {
            'Text': 'Infinity DEVELOP SCBA startovan!',
            'SMSC': {'Location': 1},
            'Number': '+38163xxxxxx',
        }
        sm.SendSMS(message)
        os.system(
            "echo  :**************SMS PORUKA POSLATA**************: " + dt_string + " >>
/home/pi/logs/system_logs/sms.log 2>&1")
    except gammu.GSMError:
        print('Failed at %s using %s' % (self.device, connection))
        os.system(
            "echo  :**************GRESKA U SLANJU SMS PORUKE**************"" >>
/home/pi/logs/system_logs/sms.log 2>&1")


t = Thread(target=sendSms)
t.start()
```

Listing (ceo kod) 2. Provera operativnosti svih elementa i slanje izveštaja SMS porukom

Listing 3.

```
#!/bin/sh

targetbus="can0"

while :
do
   cansend $targetbus 7DF#02010C0000000000 #RPM
   sleep 0.05
   cansend $targetbus 7DF#0201100000000000 #MAF
   sleep 0.05
   cansend $targetbus 7DF#02010B0000000000 #MAP
   sleep 0.05
   cansend $targetbus 7DF#0201230000000000 #FGP
   sleep 0.05
   cansend $targetbus 7DF#0201240000000000
   sleep 0.05
   cansend $targetbus 7DF#0201490000000000
   sleep 0.05
   cansend $targetbus 7DF#02012C0000000000
   sleep 0.05
   cansend $targetbus 7DF#02012D0000000000
   sleep 0.05
   cansend $targetbus 7DF#0201040000000000
   sleep 0.05
   cansend $targetbus 7DF#02010F0000000000
   sleep 0.05
   cansend $targetbus 7DF#0201050000000000
   sleep 0.05
   cansend $targetbus 7DF#0201330000000000
   sleep 0.05
   cansend $targetbus 7DF#0201460000000000
   sleep 0.05
   cansend $targetbus 7DF#02010D0000000000
   sleep 0.05
   cansend $targetbus 7DF#0201420000000000
   sleep 0.05
   cansend $targetbus 7DF#0201310000000000
   sleep 0.05
   cansend $targetbus 7DF#02011F0000000000
   sleep 0.05

done
```

Listing (ceo kod) 3. Slanje CAN poruke sa zahtevom za podatke

Listing 4.

RPM	MAP	FGP	MAF
782.75	100	24680	**5.47**
784.5	100	26660	**5.77**
782.5	100	26480	**5.72**
780	100	25240	**5.54**
776.75	100	24360	**5.74**
780.5	100	24410	**5.6**
775.75	100	24530	**5.89**
779.25	100	26480	**5.61**
776	100	26550	**5.67**
785.5	100	25020	**5.68**
776.5	100	24970	**5.68**
776	100	26110	**5.66**
782.5	100	26340	**5.69**
779	100	25900	**5.65**
786.25	100	26680	**5.75**
779.25	101	25540	**5.62**
776	100	25020	**5.78**
785.5	101	24440	**5.72**
778.75	100	24560	**5.8**
776	100	26480	**5.5**
778	100	26540	**5.58**
783.25	100	24530	**5.66**
779.5	100	26730	**5.72**
775.5	100	24850	**5.59**
779.25	100	25750	**5.65**
781	100	26390	**5.72**
776.75	100	25730	**5.6**
785	100	25260	**5.53**
782.25	100	24410	**5.74**
784	100	26800	**5.58**
783.5	100	24600	**5.67**
779	100	24380	**5.62**
777	100	26460	**5.71**
782.5	100	26670	**5.66**
784.25	100	26180	**5.57**
778.75	100	25870	**5.74**
784	100	25600	**5.8**
...			

Listing (ceo kod) 4. Deo skupa podataka dobijenih iz CAN-bus mreže

Listing 5.

```
# Importovanje dataseta
dataset_lr = pd.read_csv(url, usecols = ['RPM', 'MAF'])
dataset_lr.columns=['RPM','MAF']

print("\n*********** DATASET isnull()********************")
print(dataset_lr.isnull().sum())
print("\n*********** DATASET shape()********************")
print(dataset_lr.shape)
print("\n*********** DATASET info()********************")
print(dataset_lr.info())
corrMatrix_lr = dataset_lr.corr()
print (corrMatrix_lr) # Numrerički prikaz korelacione matrice
sns.heatmap(corrMatrix_lr, annot=True)

X_lr = dataset_lr.iloc[:, :-1].values
y_lr = dataset_lr.iloc[:, -1].values

# Razdvajanje na trerning i testni skup
from sklearn.model_selection import train_test_split
X_train_lr, X_test_lr, y_train_lr, y_test_lr = train_test_split(X_lr,
y_lr, test_size = 0.2, random_state = 0)

# Normalizacija obležija
#sc = StandardScaler()
#X_train = sc.fit_transform(X_train)
#X_test = sc.transform(X_test)

# Linearna Regresija trerning nad testnim setom
regressor_lr = LinearRegression()
regressor_lr.fit(X_train_lr, y_train_lr)

# Predikcija Test set rezultata
y_pred_lr = regressor_lr.predict(X_test_lr)

# Evaluacija naučenog modela Linearne Regresije nad testnim skupom
est_lr_with_constant = sm.add_constant(y_pred_lr)

est_lr = sm.OLS(y_test_lr, est_lr_with_constant)
est2_lr  = est_lr.fit()
print(est2_lr.summary())
# Vizualizacija rezultata PLR nad testnim setom prediktovane i stvarne
vrednosti MAF senzora
y_pred_lr_small = y_pred_lr[50:250]
y_test_lr_small = y_test_lr[50:250]

np.set_printoptions(precision=2)

# Vizualizacija rezultata nad testnim setom
plt.figure(figsize=(30,5))
plt.plot(y_pred_lr_small, label='Predikcione vrednosti MAF senzora',
linewidth=3, color = 'red')
plt.plot(y_test_lr_small, label='Vrednosti testnog skupa MAF senzora',
linewidth=2, color = 'blue')
#plt.title("Uporedna analiza testiranja predikcije Proste Linearne
Regresije modela nad testnim setom", fontsize=20)
plt.legend(loc="upper left", fontsize=15)
plt.show()
```

Listing (ceo kod) 5. Prosta Linearna Regresija - Python

Listing 6.

```
# Importovanje dataseta VLR
dataset_mlr = pd.read_csv(url, sep=',', usecols=['RPM', 'MAP', 'FGP',
'MAF'])
dataset_mlr.columns = ['RPM', 'MAP', 'FGP', 'MAF']

print("*********** DATASET ********************\n")
print(dataset_mlr.head(10))

print("\n*********** DATASET keys()********************")
print(dataset_mlr.keys())
print("\n*********** DATASET isnull()********************")
print(dataset_mlr.isnull().sum())
print("\n*********** DATASET shape()********************")
print(dataset_mlr.shape)
print("\n*********** DATASET info()********************")
print(dataset_mlr.info())

# Alokacija Matrice Obeležija RPM, MAP, FGP, MAF sernzora automobila
X_mlr = dataset_mlr.iloc[:, :-1].values  #Nezavisnih Promenjivih (eng.
Independent variables)
y_mlr = dataset_mlr.iloc[:, -1].values # Alokacija Zavisne Promenjive
(eng. Dependent variable)
# Razdvajanje na trerning i testni skup
from sklearn.model_selection import train_test_split
X_train_mlr, X_test_mlr, y_train_mlr, y_test_mlr =
train_test_split(X_mlr, y_mlr, test_size = 0.2, random_state = 0)
# Linearna Regresija trerning nad testnim setom
regressor_mlr = LinearRegression()
regressor_mlr.fit(X_train_mlr, y_train_mlr)
# Predikcija Test set rezultata
y_pred_mlr = regressor_mlr.predict(X_test_mlr)
# Evaluacija naučenog modela Linearne Regresije nad testnim skupom
est_mlr_with_constant = sm.add_constant(y_pred_mlr)

est_mlr = sm.OLS(y_test_mlr, est_mlr_with_constant)
est2_mlr  = est_mlr.fit()
print(est2_mlr.summary())

print('*************************************************************')
print('*************************************************************')

MSE_mlr = mean_squared_error(y_test_mlr, y_pred_mlr, squared=True)
RMSE_mlr = mean_squared_error(y_test_mlr, y_pred_mlr, squared=False)
MAE_mlr = mean_absolute_error(y_test_mlr, y_pred_mlr)
R2_mlr = r2_score(y_test_mlr, y_pred_mlr)

print('Srednja kvadratna greška nad testnim skupom (MSE) = ',
round(MSE_mlr, 2))
print('Koren srednje kvadratne greške nad testnim skupom (RMSE) =',
round(RMSE_mlr, 2))
print("Srednja apsolutne greške (MAE) =", round(MAE_mlr, 2))
print("R-squared (R^2) =", round(R2_mlr, 3))
```

Listing (ceo kod) 6. Višestruka Linearna Regresija - Python

Listing 7.

```
# Importovanje dataseta RF
dataset_rf = pd.read_csv(url, sep=',', usecols=['RPM', 'MAP', 'FGP',
'MAF'])
dataset_rf.columns = ['RPM', 'MAP', 'FGP', 'MAF']

print("*********** DATASET ********************\n")
print(dataset_rf.head(10))

print("\n*********** DATASET keys()********************")
print(dataset_rf.keys())
print("\n*********** DATASET isnull()********************")
print(dataset_rf.isnull().sum())
print("\n*********** DATASET shape()********************")
print(dataset_rf.shape)
print("\n*********** DATASET info()********************")
print(dataset_rf.info())
print("\n*********** DATASET detaljan opis********************")
print(dataset_rf.describe())

# Alokacija Matrice Obeležija RPM, MAP, FGP, MAF sernzora automobila
#Nezavisnih Promenjivih (eng. Independent variables)
X_mlr = dataset_mlr.iloc[:, :-1].values
# Alokacija Zavisne Promenjive (eng. Dependent variable)
y_mlr = dataset_mlr.iloc[:, -1].values
# Razdvajanje na trerning i testni skup
from sklearn.model_selection import train_test_split
X_train_mlr, X_test_mlr, y_train_mlr, y_test_mlr =
train_test_split(X_mlr, y_mlr, test_size = 0.2, random_state = 0)
# Random Forest trerning nad testnim setom
regressor_rf = RandomForestRegressor(n_estimators=100, random_state =
0, oob_score= True)
regressor_rf.fit(X_train_rf, y_train_rf)
y_pred_rf = regressor_rf.predict(X_test_rf)
# Evaluacija naučenog modela Linearne Regresije nad testnim skupom
est_rf_with_constant = sm.add_constant(y_pred_rf)

est_rf = sm.OLS(y_test_rf, est_rf_with_constant)
est2_rf  = est_rf.fit()
print(est2_rf.summary())

print('*************************************************************')
print('*************************************************************')

MSE_rf = mean_squared_error(y_test_rf, y_pred_rf, squared=True)
RMSE_rf = mean_squared_error(y_test_rf, y_pred_rf, squared=False)
MAE_rf = mean_absolute_error(y_test_rf, y_pred_rf)
R2_rf = r2_score(y_test_rf, y_pred_rf)

print('Srednja kvadratna greška nad testnim skupom (MSE) = ',
round(MSE_rf, 2))
print('Koren srednje kvadratne greške nad testnim skupom (RMSE) =',
round(RMSE_rf, 2))
print("Srednja apsolutne greške (MAE) =", round(MAE_rf, 2))
print("R-squared (R^2) =", round(R2_rf, 3))
```

Listing (ceo kod) 7. "Random Forest" ansambli - Python