U ovom dokumentu predstavljamo naše rješenje zadatka za Lumen Data Science 2020. Prateći Crisp-DM metodologiju prikazat ćemo razne metode koje smo koristili za analizu i pripremu podataka, predložiti konačni model koji rješava zadani problem te spomenuti neke ideje za daljnji nastavak rada.

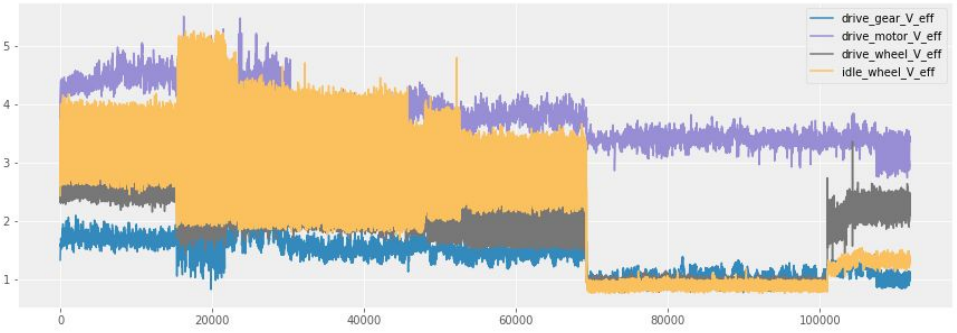
1. BUSINESS UNDERSTANDING

Zadatak je bio stvoriti model koji će omogućiti provedbu prediktivnog održavanja na viličarima koji prenose robu u skladištu. Model bi trebao, koristeći podatke dobivene pomoću IoT senzora na strojevima, u realnom vremenu ukazivati na moguće kvarove. Prednost prediktivnog održavanja u odnosu na klasične metode poput preventivnog, odnosno reaktivnog, održavanja je velika novčana ušteda povezana i s povećanom produktivnošću u proizvodnji. Naime, popravci i održavanja vršili bi se pravovremeno, pritom sprječavajući velike kvarove na strojevima.

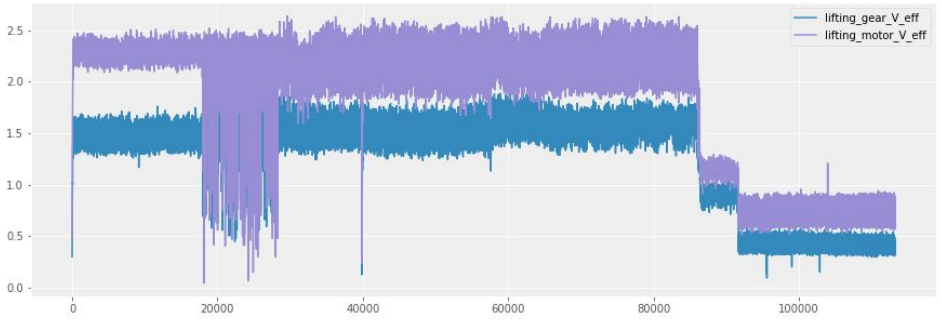
2. DATA UNDERSTANDING

2.1 Describe data

Dobili smo podatke mjerene na 7 viličara, pri čemu na svakom viličaru imamo 6 senzora koji mjere vibracije (akcelerometri), a za svaki senzor imamo 2 izmjerene vrijednosti: V_eff (efektivnu brzinu, mjerna jedinica mm/s) i a_max (maksimalnu akceleraciju, mjerna jedinica mg (mili-g)). Uz to, dobili smo i podatke o izvršenim popravcima u danom razdoblju.



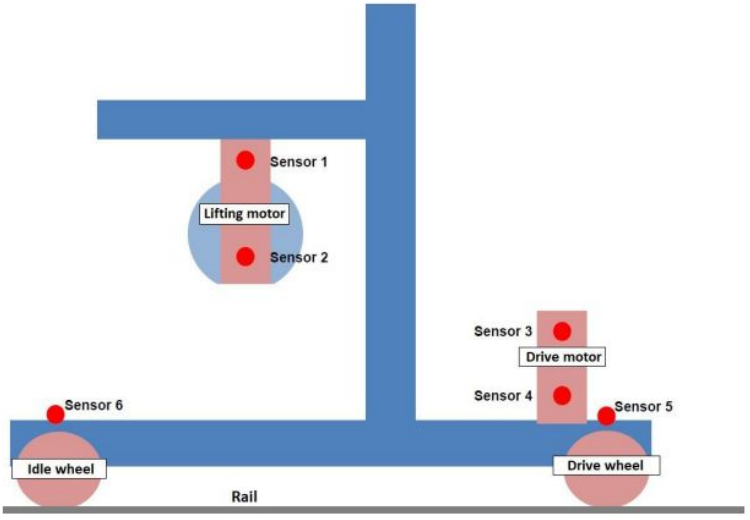
Slika 2.1.1 Efektivne brzine drive senzora za FL01



Slika 2.1.2 Efektivne brzine lifting senzora za FL01

Dva stroja, FL01 i FL07, imaju mjerenja u dužem vremenskom periodu nego preostalih pet. Razlog je to što su ta dva stroja sudjelovala u testnom periodu.

Senzori se dijele na drive senzore (mehanizam za vožnju) i lifting senzore (mehanizam za prenošenje tereta) te idle wheel senzor (kotač koji nije pod pogonom, služi za ostvarivanje ravnoteže) koji smo grupirali s drive mehanizmom.



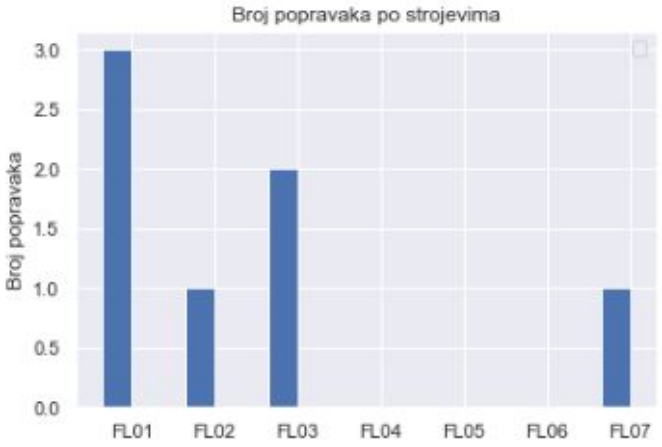
Slika 2.1.3 Shematski prikaz viličara s označenim senzorima.

Podaci su dani u obliku CSV datoteke sa 7 stupaca te 3860434 redaka odvojenih s “;”.

	machine_name(string)	sensor_type(string)	date_measurement(date)	start_timestamp(timestamp)	end_timestamp(timestamp)	realvalue(decimal)	unit(string)
0	FL01	drive_gear_V_eff	2017-09-02	2017-09-02 15:26:42.823	2017-09-02 15:26:42.823	0.395	mm/s
1	FL01	drive_gear_V_eff	2017-09-02	2017-09-02 15:26:45.653	2017-09-02 15:26:45.653	0.577	mm/s
2	FL01	drive_gear_V_eff	2017-09-02	2017-09-02 15:26:48.467	2017-09-02 15:26:48.467	0.717	mm/s
3	FL01	drive_gear_V_eff	2017-09-02	2017-09-02 15:26:51.293	2017-09-02 15:26:51.293	0.832	mm/s
4	FL01	drive_gear_V_eff	2017-09-02	2017-09-02 15:26:54.107	2017-09-02 15:26:54.107	0.941	mm/s

Slika 2.1.4 Početni podaci s pripadajućim tipovima podataka u zagradi

Popravci su navedeni u Excel dokumentu gdje je popravak označen imenom stroja, datumom i opisom popravka. Odnos popravaka je prikazan sljedećim grafom:

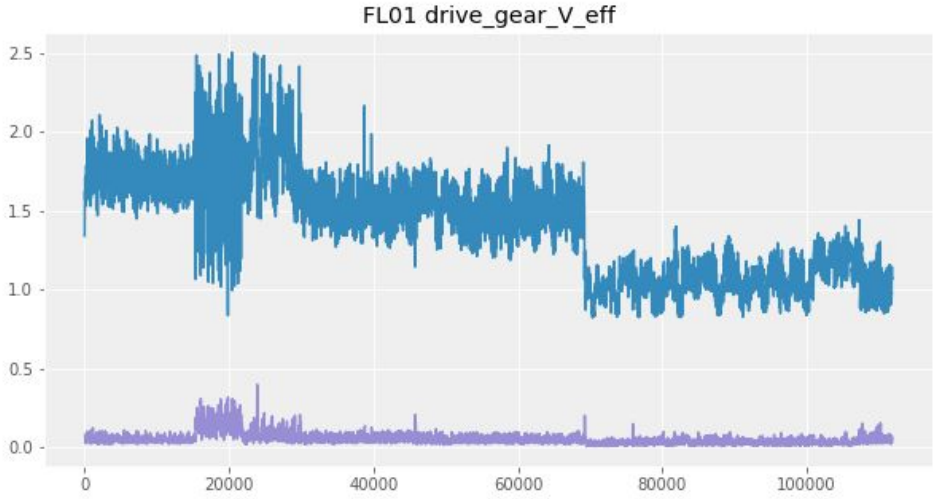


Slika 2.1.5 Distribucija popravaka po strojevima

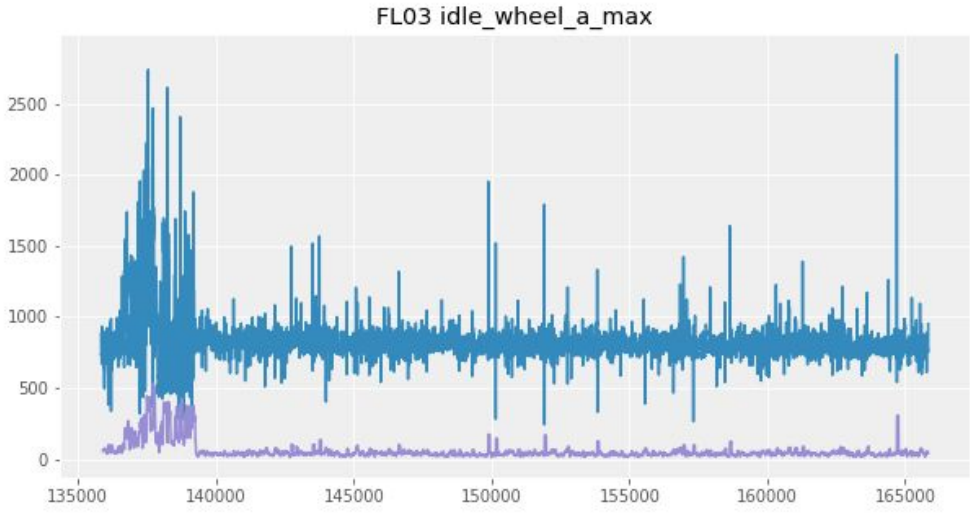
2.2 Explore data

Istraživanje smo započeli osnovnom eksploratornom analizom podataka koristeći Tableau te Matplotlib biblioteku za Python. Mjerenja su vršena triput dnevno (ujutro, poslijepodne i navečer). Uočili smo brojne outliere i neravnomjerne vrijednosti u mjerenjima. Nadalje, uočili smo utjecaj popravaka na vibracije, ali i razne druge promjene ponašanja u podacima.

Na sljedećim grafovima prikazane originalne vibracije i pripadne rolling standardne devijacije. Vidimo da su promjene u ponašanju vibracija popraćene promjenama u standardnoj devijaciji.

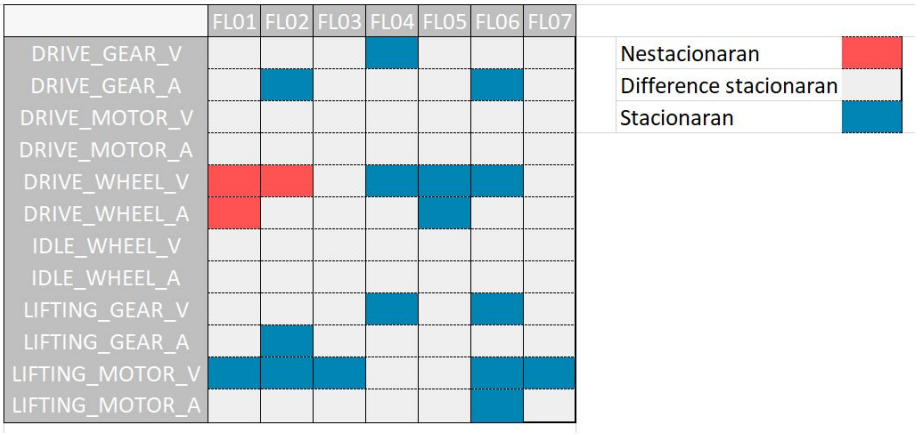


Slika 2.2.1 Plavo: originalni signal, ljubičasto: rolling standardna devijacija s veličinom prozora koja odgovara jednom mjerenju.



Slika 2.2.2 Plavo: originalni signal, ljubičasto: rolling standardna devijacija s veličinom prozora koja odgovara jednom mjerenju.

Stacionaran vremenski niz je onaj u kojem su statistički pokazatelji poput prosjeka, varijance, autokorelacije i dr. konstantni tokom vremena. Alati kojima smo se služili, poput ARIMA modela, kao bitnu pretpostavku imaju stacionarnost niza s kojim rade. U svome rješenju za provjeru stacionarnosti niza uzeli smo kombinaciju dva vrlo poznata testa, Augmented-Dickey-Fullerov test te KPSS (Kwiatkowski–Phillips–Schmidt–Shin). Napominjemo da imaju različite nulte hipoteze. Kombinirajući rezultate oba testa možemo kao povratnu informaciju dobiti 4 slučaja, a to su da je niz: stacionaran, nestacionaran, difference-stacionaran te trend-stacionaran. Difference-stacionarne nizove znamo interpretirati – potrebno je diferencirati niz da bismo dobili stacionaran. Nekada je potrebno više puta ponoviti postupak da bi se postigao željeni rezultat. Trend-stacionarnim nizovima potrebno je reducirati trend, no nismo imali takvih slučajeva. Kod strogo nestacionarnih opet smo jednim diferenciranjem dobili stacionaran niz.



Slika 2.2.3 Podjela po stacionarnosti

Ovakve rezultate smo i očekivali jer naši podaci sadrže promjene ponašanja. S druge strane, promatrano u manjim vremenskim intervalima svi nizovi bili su difference stacionarni, što je za naše primjene bilo važnije.

Promatrajući rad EDM Breakout Detectiona došli smo do sumnje da su senzori drive_gear_a_max te drive_motor_a_max nositelji značajnih informacija u detekciji kvara stroja u odnosu na ostale.

Spomenimo odmah da smo usporednu analizu uglavnom vršili na podskupu svih podataka koje imamo, točnije na skupu podataka u razdoblju od 19.3.2019. do 10.7.2019. To je vremenski period od uvođenja svih 7 strojeva pa sve do zadnjeg zabilježenog mjerenja. Na preostalim podacima (koji uključuju samo FL01 i FL07) mogli smo testirati parametre za korištene algoritme, međutim bili smo oprezni pri donošenju zaključaka uzevši u obzir da je to razdoblje u kojem se izvorno isprobavao rad senzora, stoga nije vjerodostojno kao gore navedeni period.

2.3 Data quality

Dobiveni podaci nisu imali NaN vrijednosti, ali bilo je dana u kojima nisu bila vršena mjerenja. Nadalje, u startu smo primijetili neravnomjeran omjer količine podataka dobivenih mjerenjima te stvarnih etiketiranih popravaka. Nijedan popravak koji smo imali nije se odnosio na lifting senzore. Popravci na strojevima FL02 i FL03 bili su na samom početku praćenog rada, dok na FL04, FL05 i FL06 uopće nije bilo evidentiranih popravaka.

3. DATA PREPARATION

3.1 Data Cleaning

Podaci koje smo dobili dobiveni su mjerenjima mjerenjima koja su vršena triput dnevno, otprilike u 6, 14 te 22h. Veliko mjerenje sastoji se od nizova uzastopnih točaka (podataka). Svaki pojedini niz sastoji se od otprilike 25 točaka efektivne brzine te 35 maksimalne akceleracije za drive senzore, dok, s druge strane, lifting senzori sadrže nešto više. Navedeni nizovi podataka unutar jednog velikog mjerenja udaljeni su otprilike 20 sekundi.

Željeli smo uskladiti sve podatke s kojima radimo tako da postoje fiksirane veličine koje se pojavljuju periodično. Na tom putu naišli smo na niz problema koje smo morali otkloniti.

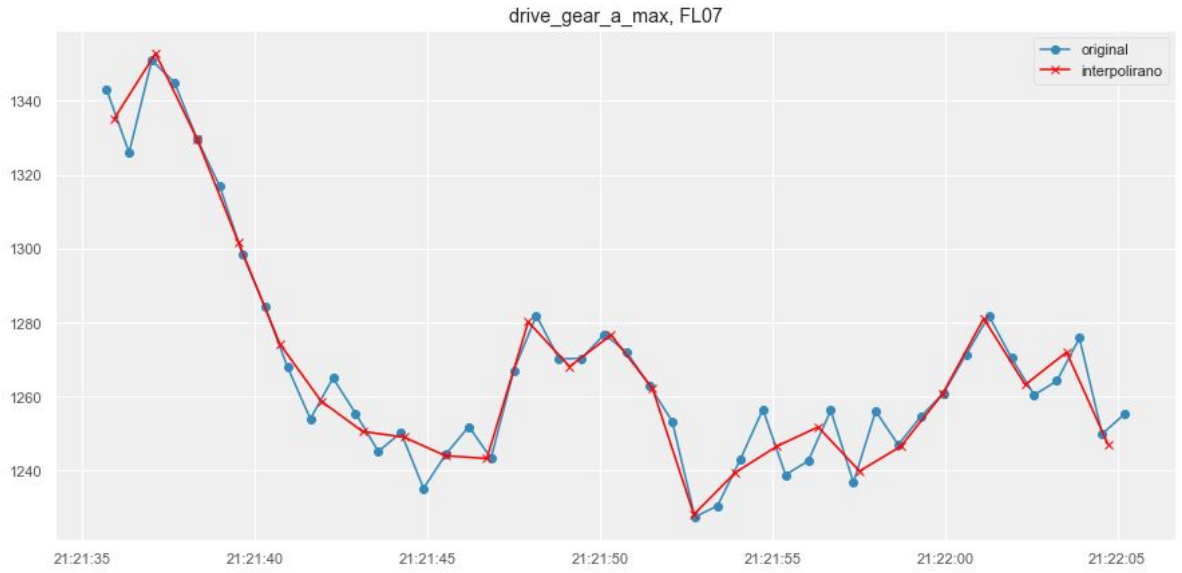
Prvi problem s kojim smo se susreli su izolirane točke mjerenja, točnije one udaljene od prve najbliže za više od 9 sekundi. Najjednostavnije smo ih izbacili zato što ionako ne mogu nositi značajnu informaciju. U istom vremenskom trenutku pojavljivalo se više mjerenja. Takve slučajeve zamijenili smo prosjekom njihovih vrijednosti. Ukoliko su se pojavila mjerenja nastala manje od 20ms nakon prethodnog, također smo ih izbacili.

Nadalje, lifting senzori često imaju svoja mjerenja 10ak sekundi prije mjerenja drive senzora. Osim toga, ponekad u vremenskom intervalu nema mjerenja za sve senzore na našem raspologanju. To nikako nisu osobine koje želimo imati u svom setu podataka, stoga smo za početak razdvojili cijeli dataset na 2 dijela – drive i idle senzore zajedno te lifting odvojeno. Mjerenja koja ne obuhvaćaju svih 8 senzora za drive i idle, odnosno sva 4 za lifting, naprosto smo izbacili. Nismo htjeli sačuvati neravnotežu u podacima.

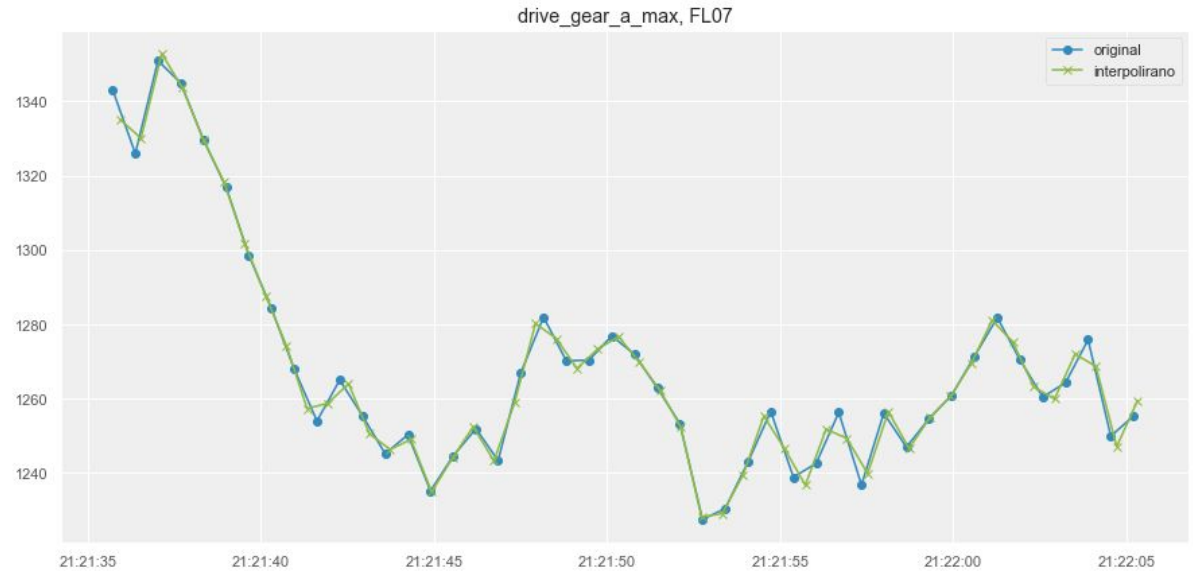
Kako su se nizovi podataka razlikovali u broju točaka, točke u vremenu smo interpolirali kubičnim splajnom s 60 točaka za drive te 70 za lifting senzore.

I dalje se mjerenja nisu potpuno vremenski poklapala, no zaključili smo da su razlike dovoljno male da ih možemo zanemariti. Jednak broj točaka po mjerenju nam je omogućio daljnju analizu, poput change-point

detection-a te ICA-a prvenstveno. Odlučili smo se za 60, odnosno 70, točaka po mjerenju jer se za manje gubi struktura mjerenja.



Slika 3.1.1 Primjer interpolacije s premalo točaka u mjerenju.



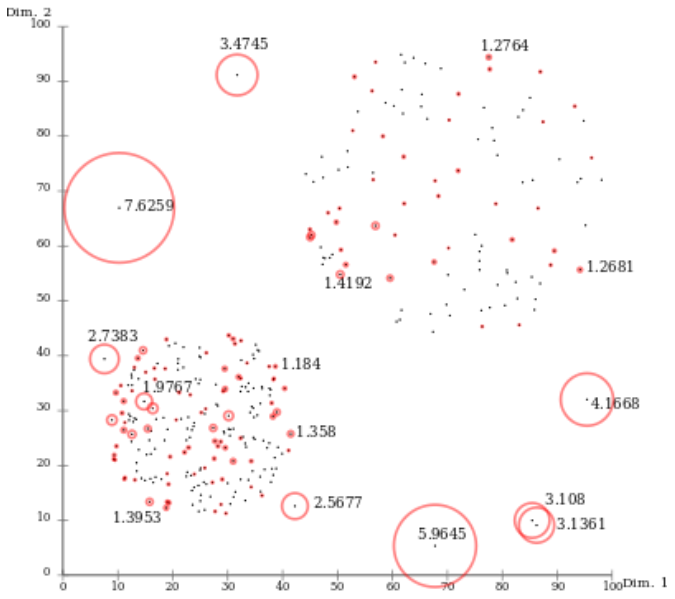
Slika 3.1.2 Primjer konačne interpolacije.

Daljnju analizu su nam otežali outlieri u mjerenjima, kao i mjerenja koja su sama bila abnormalnih vrijednosti. To je najviše do izražaja došlo u spektralnoj analizi, gdje bi cijeli period kojeg smo promatrali davao krive vrijednosti.

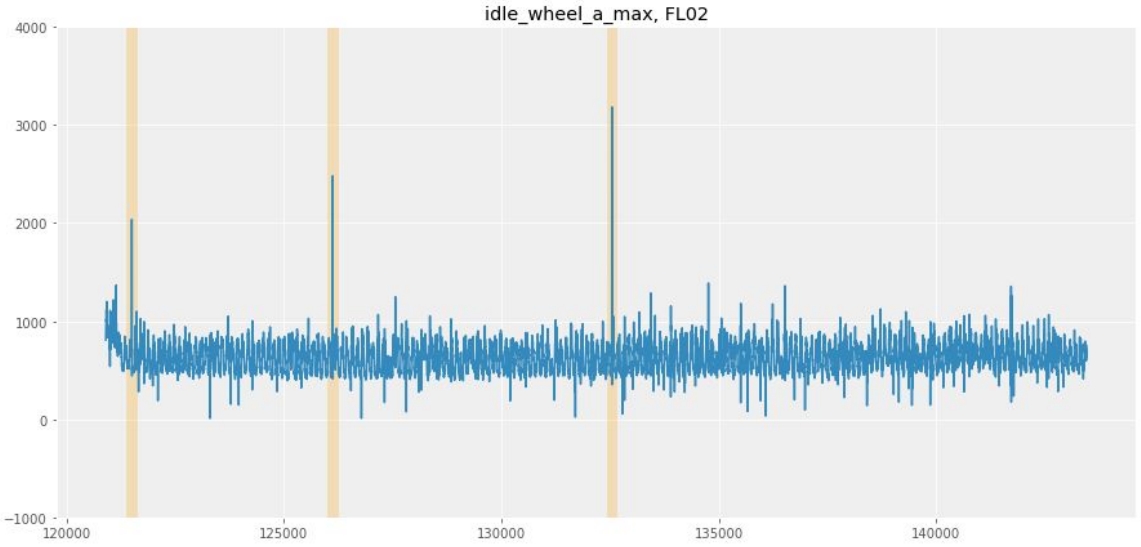
Za detekciju outliera odabrali smo Local Outlier Factor (LOF), algoritam koji uspoređuje lokalne gustoće (local density) objekta s lokalnom gustoćom svojih susjeda. Prednost LOF-a je sposobnost prepoznavanja outliera u datasetu koji, možda, ne bi bili outlieri u drugom području tog istog dataseta. Kod detektiranja bili

smo relativno strogi jer smo odlučili izbaciti cijelo mjerenje na kojem se pojavio outlier, čak i u slučajevima kada bi bilo samo jedno takvo mjerenje. Tako zadržavamo opisanu strukturu podataka, odnosno sva mjerenja imaju konzistentan broj točaka.

Smatrali smo da bi broj outliera u nekom periodu mogao biti jedan od pokazatelja stanja stroja. Ovime smo izbacili oko 7% svih podataka.



Slika 3.1.3 Primjer rezultata LOF-a. Zaokružene točke su outlieri.



Slika 3.1.4 Primjer detekcije outliera

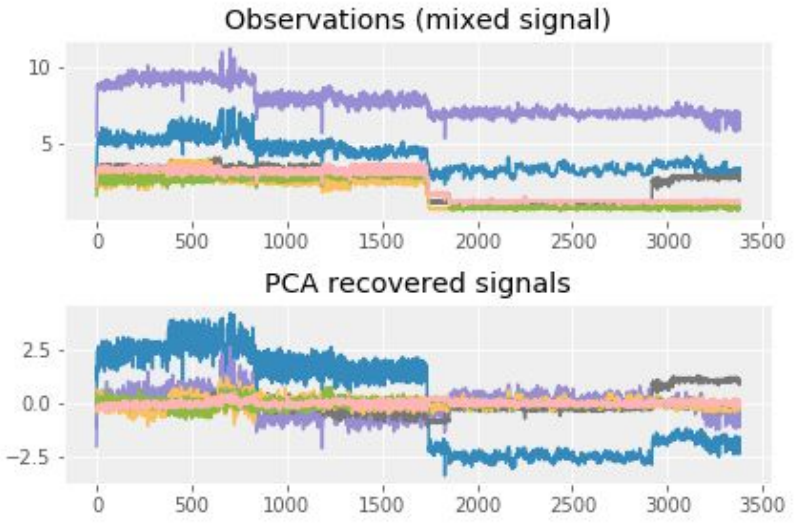
3.2 Data construction

Kako smo za svaki stroj imali relativno velik broj različitih podataka (6 senzora, V_eff i a_max za svaki), počeli smo proučavati tehnike za smanjenje dimenzionalnosti u nadi da će nam smanjenje dimenzije olakšati razumijevanje podataka i modeliranje. Metode koje smo koristili za to bile su PCA (Principal Component Analysis) i ICA (Independent Component Analysis).

PCA projicira podatke u odabranu dimenziju, pri čemu su dobivene komponente odabrane tako da maksimiziraju varijabilnost u podacima.

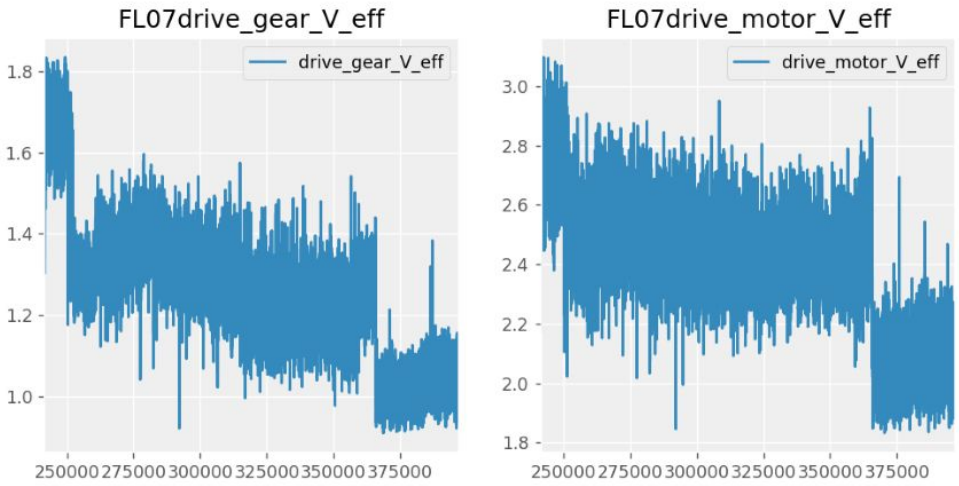
Za primjenu PCA-a podatke smo prvo odvojili po grupama, gdje su jednu grupu činili lifting senzori, a drugu drive senzori te idle wheel. Zatim smo odvojeno promatrali V_eff i a_max te iz svake grupe uzeli samo prvu komponentu. Pokazalo se da ta jedna komponenta čuva većinu varijabilnosti u podacima. Dobivene podatke smo kasnije koristili za SARIMAX model kako bismo smanjili vrijeme potrebno za treniranje modela bez gubitka previše informacija.

Na sljedećem grafu vidimo kako se većina strukture u originalnim signalima prepoznaje u prvoj principalnoj komponenti koja je označena plavom bojom.



Slika 3.2.1 Primjer rezultata PCA-a

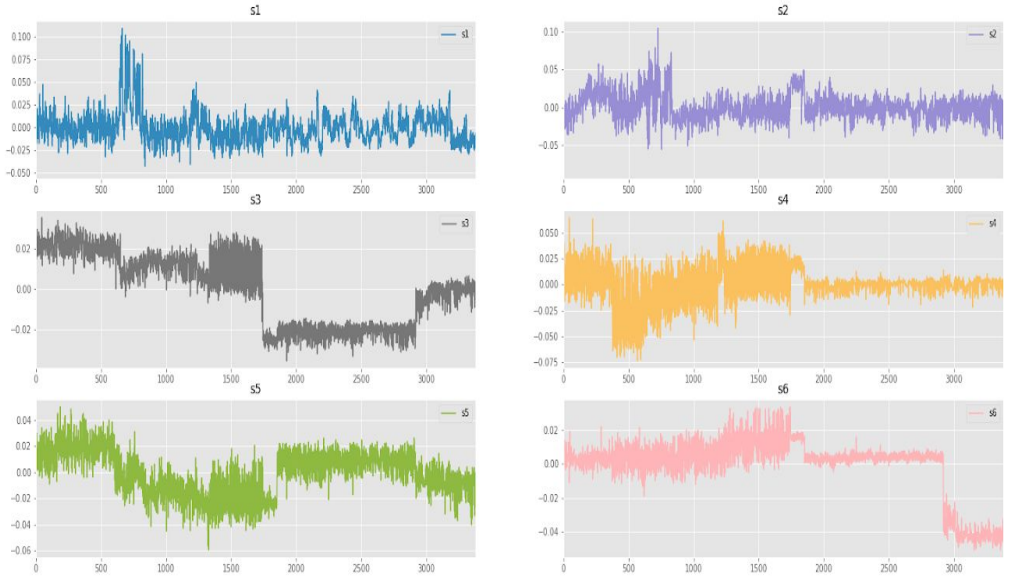
Primjetili smo da drive_gear i drive_motor senzori daju izrazito slične informacije, što je bila dodatna motivacija za primjenu ove metode. Na sljedećem grafu prikazana je ta sličnost na efektivnim brzinama za stroj FL07.



Slika 3.2.2 Slično ponašanje drive_gear i drive_motor senzora.

ICA ima pretpostavku da su naši signali dobiveni kao linearna kombinacija nekih originalnih signala te kao povratnu informaciju vraća te originalne signale. Smatrali smo da je kod nas ta pretpostavka ispunjena jer vibracije na jednom dijelu stroja sigurno utječu na mjerenja vršena na ostalim dijelovima. Metoda rekonstruira originalne signale tako što maksimizira njihovu međusobnu nezavisnost.

Na sljedećem grafu vidimo rezultate ICA-a za stroj FL01. Možemo uočiti da različite komponente čuvaju različite uzorke i da su, u usporedbi s originalnim podacima, dobiveni signali manje korelirani, odnosno više nezavisni.



Slika 3.2.2 Primjer rezultata ICA-a

ICA smo opet primijenili nakon odvajanja drive i lifting senzora zasebno promatrajući V_eff i a_max.

3.3. Format data

Pri korištenju svih alata htjeli smo imati razdvojene vremenske nizove po strojevima i pripadnim senzorima. Zbog toga smo transformirali početni set podataka u novi tako da stupci predstavljaju senzore. Preciznije, postoje 2 nova dataseta razdvojena po drive i idle senzorima, odnosno lifting senzorima. Konačna verzija izgleda ovako:

	machine	start_timestamp	drive_gear_V_eff	drive_gear_a_max	drive_motor_V_eff	drive_motor_a_max	drive_wheel_V_eff	drive_wheel_a_max	idle_wheel_V_eff	idle_wheel_a_max
0	FL01	2017-09-02 15:26:40.683000000	0.221627	166.862000	0.412642	189.187000	0.427227	129.874777	0.285272	203.291599
1	FL01	2017-09-02 15:26:41.171372881	0.264266	280.065835	0.454375	296.118121	0.544546	204.607209	0.867165	378.999661
2	FL01	2017-09-02 15:26:41.659745762	0.305033	347.036077	0.506939	369.389561	0.721272	247.615678	1.306367	406.758835
3	FL01	2017-09-02 15:26:42.148118644	0.343995	473.749828	0.568786	453.217412	0.928983	380.797241	1.623450	555.329540
4	FL01	2017-09-02 15:26:42.636491525	0.381225	630.215678	0.638370	543.167812	1.139253	550.031254	1.838985	783.934411

Slika 3.3.1 Prikaz konačnih podataka za drive i idle senzore

	machine	start_timestamp	lifting_gear_V_eff	lifting_gear_a_max	lifting_motor_V_eff	lifting_motor_a_max
0	FL01	2017-09-02 15:26:41.310000000	0.301401	292.933000	0.460982	398.680000
1	FL01	2017-09-02 15:26:41.718507246	0.365649	349.839834	0.512700	440.348150
2	FL01	2017-09-02 15:26:42.127014492	0.422266	437.485051	0.559318	584.568136
3	FL01	2017-09-02 15:26:42.535521739	0.471835	542.729608	0.601246	725.791486
4	FL01	2017-09-02 15:26:42.944028985	0.514942	585.265274	0.638890	781.086816

Slika 3.3.2 Prikaz konačnih podataka za lifting senzore

Tip podataka za „machine“ je string, timestamp je za „start_timestamp“, a svi ostali su floatovi.

Ukupan broj redaka jest 855170, pri čemu 428520 odgovara drive i idle, a 426650 lifting senzorima.

4.MODELING

Prvo donosimo pregled svih metoda koje ulaze u konačni model, a zatim ćemo objasniti konstrukciju samog modela.

4.1 Fourierova analiza

Kako se mjerenja ne događaju neprekidno cijeli dan, nego nekoliko puta dnevno po par minuta, signal koji dobivamo je isprekidan po vremenu. Da bismo bili sigurni u točnost rezultata koje dobijemo promatranjem domene frekvencija signala (nakon primjene Fourierove transformacije), treba nam duži period od jednog mjerenja. Problem nastaje jer se jedno mjerenje ne nastavlja savršeno na prošlo pa se, kada gledamo podatke bez vremenske komponente, vide skokovi u vrijednostima.

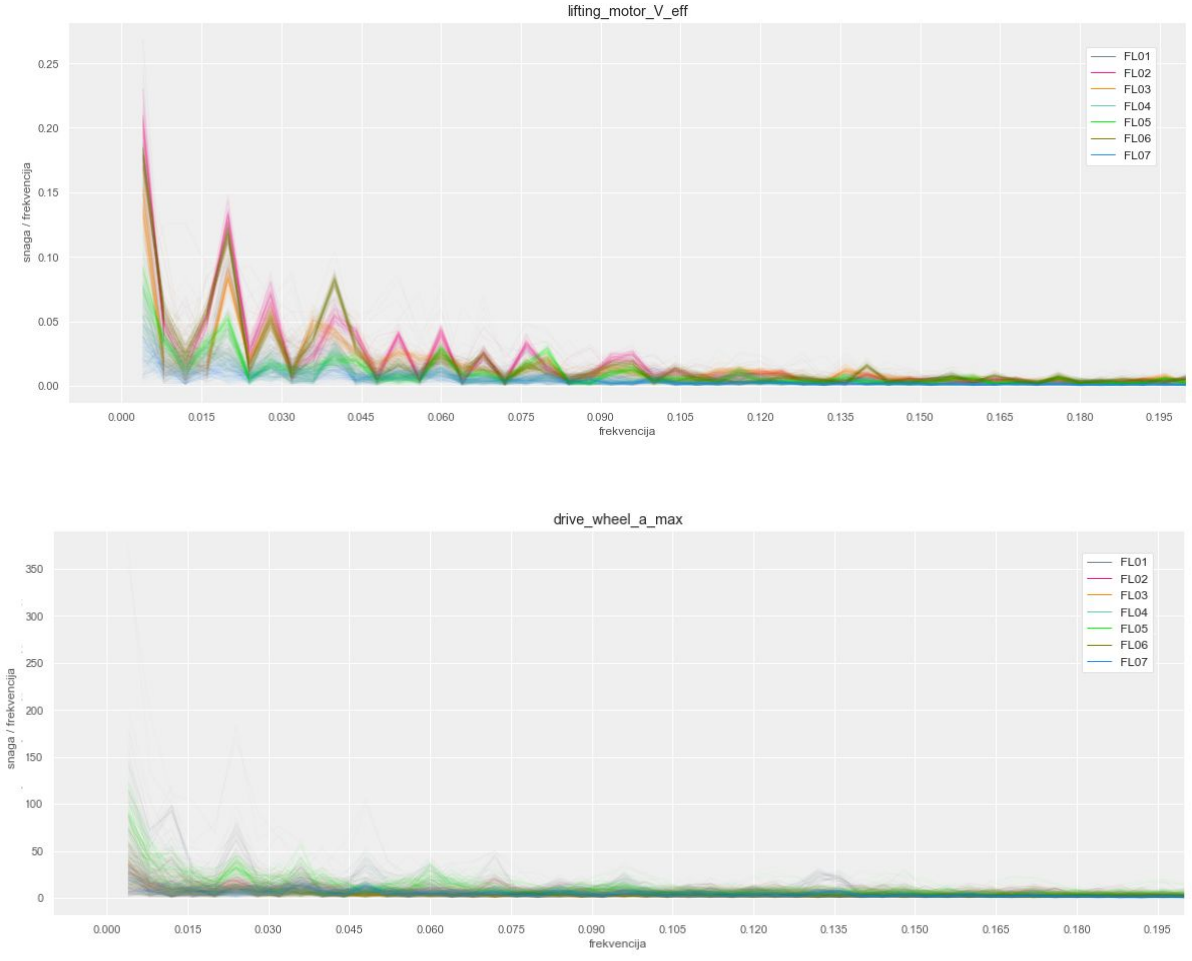
Problem smo riješili tako da smo zanemarili ukupnu vrijednost mjerenja te se koncentrirali na oblik. Mjerenja bi spajali tako da sve vrijednosti mjerenja povećamo ili smanjimo kako bi se lijepo nastavljalo na prošlo.

Sada nam je problem stvaralo što strojevi nekad kreću iz nultog momenta pa su prve vrijednosti u mjerenju puno niže nego ostale. Nakon micanja nekoliko točaka s početka i kraja te micanja trenda smo napokon dobili jednolik signal s kojim se može dalje raditi.

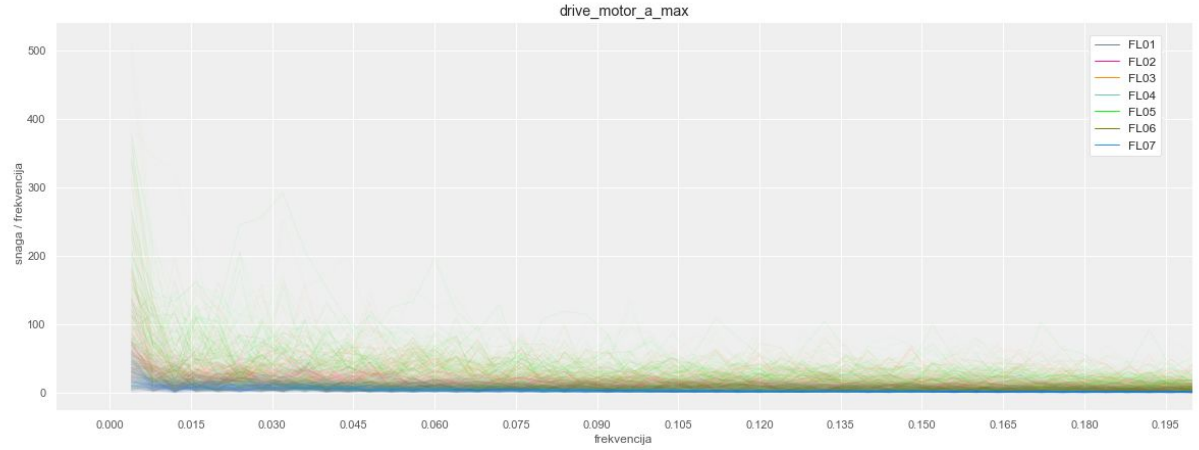
Osnovni graf koji smo promatrali bila je procjena Power Spectral Densityja napravljena Fourierovom analizom. Na mjerenja u nekom periodu smo primijenili Fourierovu transformaciju te na realni i imaginarni dio izlaznih podataka kvadrat norme.

Da provjerimo sličnost ponašanja strojeva međusobno, a i u vremenu, proučavali smo velik broj takvih grafova uhvaćenih u različitim trenucima za različite mašine. Otkrili smo da postoje velike pravilnosti u ponašanju mjerenja efektivne brzine za sve strojeve, dok se izmjerene maksimalne akceleracije ne ponašaju tako pravilno.

Na sljedećim grafovima vidimo pravilno javljanje nekih glavnih frekvencija za svaki senzor (na sljedećem grafu su to frekvencije čije su vrijednosti višekratnici 0.08), te na zadnjem nepravilno ponašanje ‘drive_motor_a_max’ senzora. Grafovi su dobiveni uzimanjem velikog broja mjerenja za različite strojeve i vremenske periode, ali za isti senzor.



Slika 4.1.1 Frekvencije dobivene uzimanjem velikog broja mjerenja istog senzora za različite strojeve

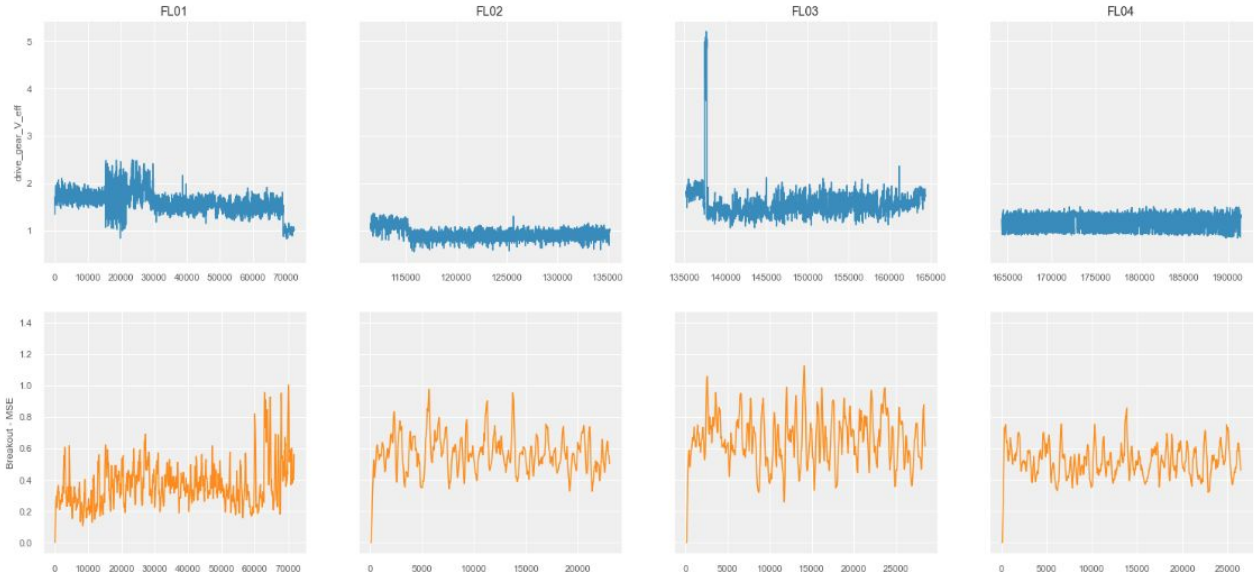


Slika 4.1.2 Frekvencije dobivene uzimanjem velikog broja mjerenja istog senzora za različite strojeve.

Prvi od gore prikazanih grafova prikazuje graf s pravilnim ponašanjem, dok su ostala dva primjeri nepravilnog ponašanja.

Nakon što znamo da postoji pravilnost u vibracijama i da će stroj u dobrom stanju jednoliko vibrirati kroz vrijeme, možemo gledati promjene ponašanja u vibracijama.

Prvi, i najjednostavniji, način kako smo modelirali promjene u vibracijama je da smo graf snage i frekvencije (dobiven Foureirovom analizom) u jednom periodu usporedili s takvim grafovima iz nekoliko prošlih perioda istog trajanja. Pretpostavka je bila da će stroj koji se kvari bitno nepravilnije vibrirati u usporedbi s pravilnim.

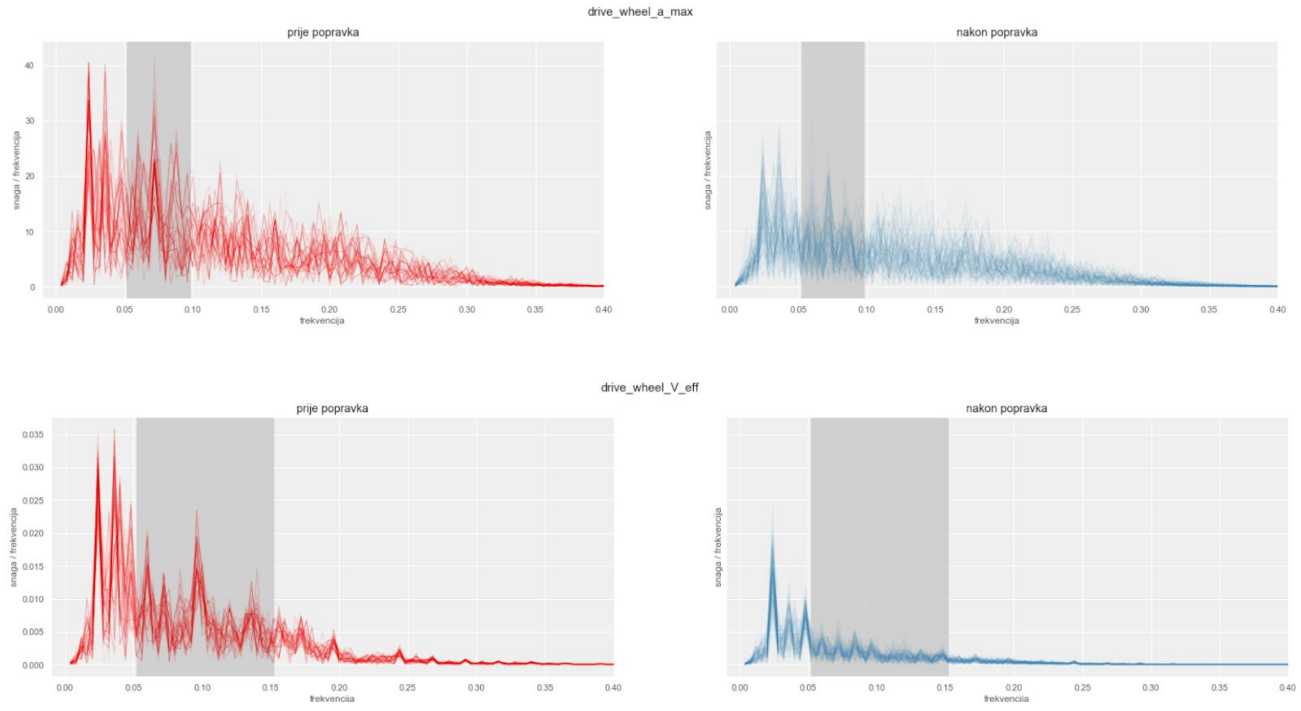


Slika 4.1.3 Graf izvornih mjerenja i razlike u vibracijama od prošlih mjerenja.

Analizom frekvencija vibracija, u periodu prije popravka i nakon popravka, uočili smo da su, osim što su amplitude veće, u periodu prije popravka češće vibracije veće frekvencije.

Budući da su grafovi koje smo promatrali bili procjene power spectral densityja, veće amplitude kod pokvarenog stanja govore da se vibracijama u tom periodu oslobodilo više energije, stoga gledanjem površine ispod grafa možemo dobiti dobru procjenu ispuštene energije. Objašnjenje koje smo uzeli je da će stroj u kojem dolazi do npr. trošenja ležaja ili neravnoteže u radu jače vibrirati od onog gdje sve radi kako treba.

Isto tako, za vibracije manjih frekvencija smo smatrali da bi mogle predstavljati podrhtavanja stroja zbog nekog kvara.



Slika 4.1.4 Razlike u vibracijama prije i poslije popravaka.

4.2 Hvatanje trenda

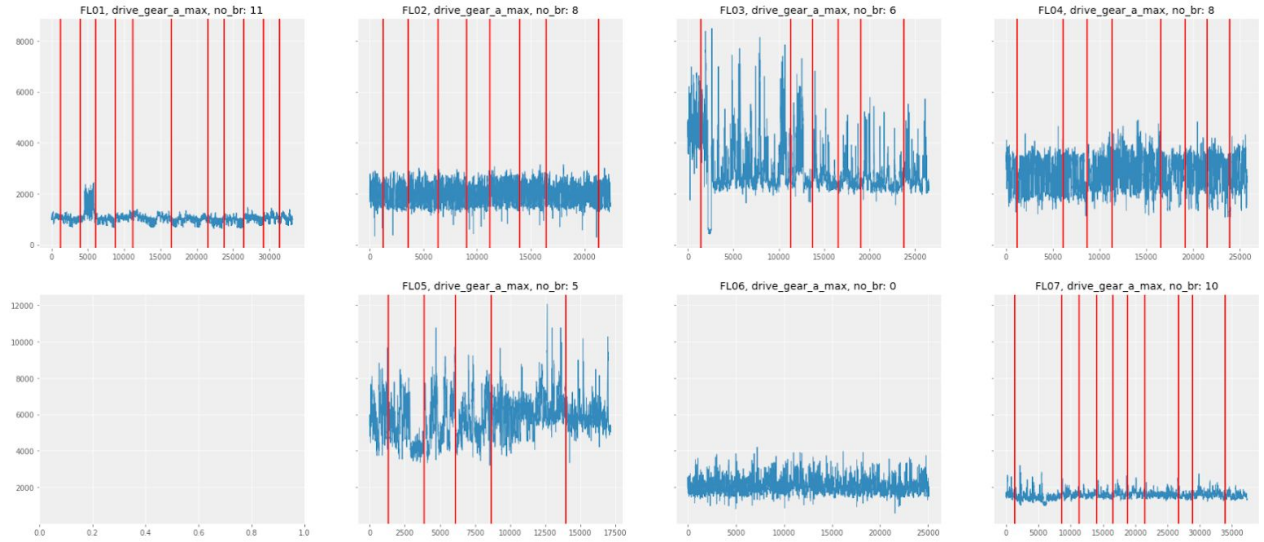
Proučavanjem literature o analizi vibracija saznali smo da je jedan od glavnih pokazatelja neispravnog rada stroja pojava trenda u vrijednostima vibracija. S tim na umu napravili smo sistem za hvatanje trenda u vremenskom nizu koji smo u sklopu modela primjenjivali na originalne podatke, kao i na neke transformirane podatke dobivene iz nekih komponenti modela (FFT, ICA, standardne devijacije).

4.3 EDM Breakout Detection

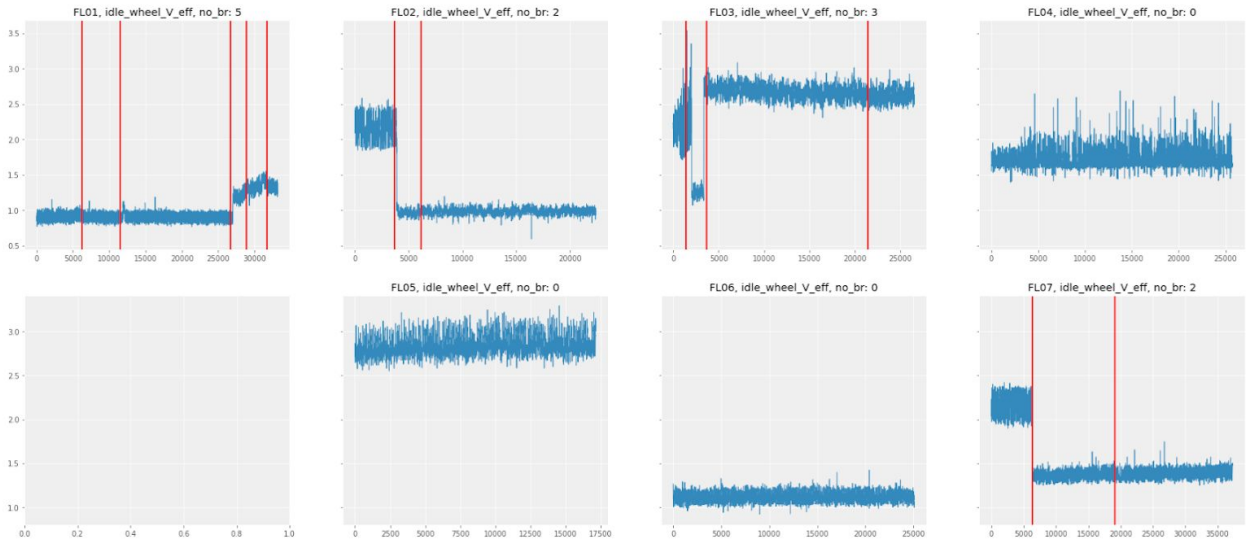
Željeli bismo znati kada senzor značajno promijeni svoje ponašanje. Ta informacija vrlo je važna u otkrivanju grešaka u radu. Međutim, zbog kaotičnosti gibanja, golim okom teško je mnoge promjene uočiti. Ideja je pronaći algoritam kojim bismo na efikasan način identificirali te promjene ponašanja, npr. mean shift, promjene trendova i sl. S druge strane, ne bismo htjeli „slučajne“ skokove okarakterizirati kao bitne. Za to koristimo EDM (E-divisive with medians) Breakout Detection, algoritam razvijen za potrebe Twittera. EDM Breakout Detection korištenjem statističkih metoda vraća položaj breakouta u danom vremenskom nizu koristeći mean shift kao glavnu mjeru. Bitno je da je robustan na anomalije u podacima, primjerice navedene „slučajne“ skokove koji ne promijene distribuciju vremenskog niza. Verzija koju smo koristili sposobna je u vremenskom nizu pronaći višebrojne breakoute. Značajni parametri su „beta“, „degree“ te „min_size“.

Beta i degree u paru predstavljaju razinu penalizacije. Laički rečeno, što je beta veći, to je otporniji pronalazak breakouta. Min_size predstavlja duljinu intervala po x-osi unutar kojeg neće biti pronađen novi breakout. Bitna pretpostavka algoritma je da su vrijednosti u intervalu [0, 1]. Zbog formata u kojem se naši podaci nalaze, odlučili smo se da je realna pretpostavka da bitnije promjene ponašanja budu barem svaka 3 dana (min_size). Napomenimo, to se odnosi na 3 dana u mjerenjima koje imamo, što je u stvarnim danima nešto više. Polazeći od te pretpostavke gledali smo prozore od 7 dana mjerenja te na svakom pojedinačnom pozivali algoritam.

U konačnom modelu smo, međuostalom, grupirali svakih 5 uzastopnih podataka uzimajući njihov prosjek. Analiziranjem smo došli do zaključka da na taj način ne gubimo puno informacija, a znatno ubrzavamo izvršavanje, što je također jedan od prioriteta za upotrebljivost modela. Nakon što smo sveli set parametara na samo jedan za sve senzore, ispostavilo se da drive_motor_a_max te drive_gear_a_max vraćaju najveći broj breakouta u danom periodu, što smo već ranije u radu spomenuli. Bitna napomena je da smo radije dopustili više pronađenih breakouta iz razloga što je bolje da prepoznamo grešku koja ne postoji, nego da ne prepoznamo kvar koji postoji.



Slika 4.3.1 Rezultat EDM-a na drive_gear_a_max po svim strojevima



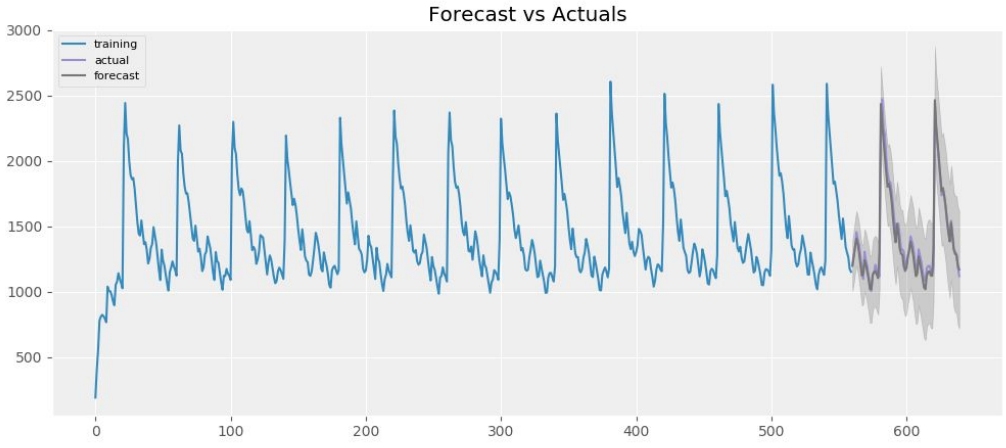
Slika 4.3.2 Rezultat EDM-a na idle_wheel_V_eff po svim strojevima

4.4 Sarimax

Među prvim stvarima koje smo počeli proučavati bile su klasične metode za analizu vremenskih nizova. Jedan od modela koje smo isprobali bio je ARIMA, koji spada u najkorištenije alate za time series forecasting.

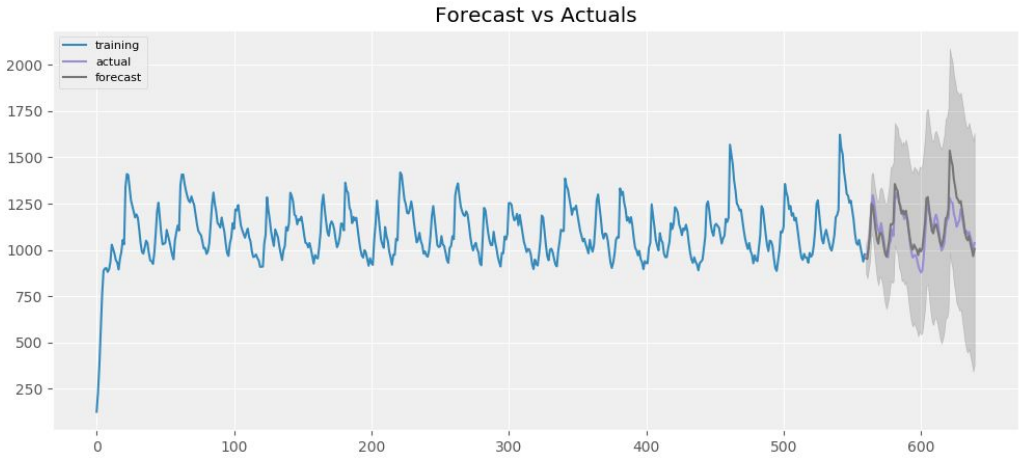
Općenito je dugoročno predviđanje ponašanja vremenskih nizova veoma težak problem, što se pokazalo istinitim i u našem slučaju. S druge strane, uspostavilo se da se lokalno, unutar kraćeg vremenskog perioda, naši signali mogu dobro modelirati ARIMA procesom.

Umjesto običnog ARIMA modela odlučili smo koristiti SARIMAX koji ima ugrađenu sezonalnu komponentu koja je omogućila modelu da prepoznaje raspored naših podataka u vremenu.

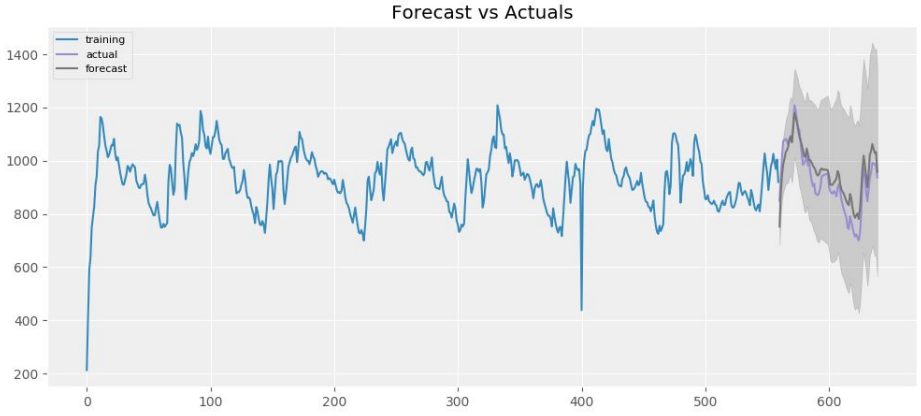


Slika 4.4.1 Primjer oblika signala u vremenu od tjedan dana.

Pokazalo se da najbolje rezultate dobivamo korištenjem prethodnih tjedan dana za predviđanje idućeg dana. U periodima gdje bi ponašanje mjerenja bilo relativno pravilno ova predviđanja bi bila vrlo precizna, dok u periodima nepravilnog i kaotičnog ponašanja dobivali bismo velike greške u usporedbi predviđenih i stvarnih mjerenja. Pretpostavka je bila da česta pojava takvih grešaka ukazuje na nepravilne vibracije koje bi mogle uzrokovati kvar.



Slika 4.4.2 Primjeri pravilnog ponašanja



Slika 4.4.3 Primjer kaotičnijeg ponašanja.

S obzirom da naši podaci uglavnom nisu bili stacionarni, mjerenja smo morali jednom diferencirati što bi ih učinilo stacionarnima. Nakon toga smo analizom autokorelacija i parcijalnih autokorelacija odabrali optimalne parametre za svaki senzor. Na kraju smo, ovisno o senzoru, imali 2 tipa modela, to su bili (1,1,0)x(0,1,1) i (0,1,1)x(0,1,1) SARIMAX modeli. Sezonalnost su nam kod drive senzora predstavljala 2 mjerenja, dok smo kod lifting senzora uzeli jedno mjerenje. Razlog je što su kod drive senzora 2 mjerenja stvarala uzorak koji se periodično ponavljao.

Javilo se par iznimnih slučajeva gdje bi isti senzor na različitim strojevima zahtijevao različite parametre, stoga za buduće korištenje na novim strojevima predlažemo ponovno razmatranje spomenutih autokorelacija pri odabiru parametara.

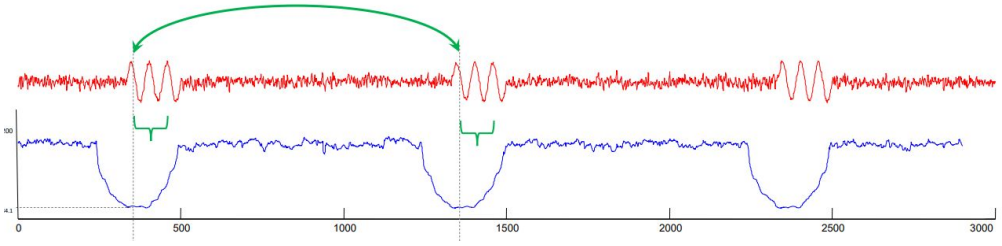
Zbog velikog perioda u sezonalnoj komponenti trebalo nam je previše vremena za treniranje modela na svakom senzoru posebno. Zbog toga smo odlučili primijeniti PCA kako bismo smanjili dimenziju podataka i onda koristiti SARIMAX na pripadnim rezultatima. Pokazalo se da rezultati ostaju slični, a postiže se velika ušteda u vremenu, što nam je bio jedan od prioriteta kako bismo osigurali da naš model bude primjenjiv.

Pri mjerenju grešaka predviđanja modela odlučili smo iskoristiti interval pouzdanosti od 95%, prikazan na gornjim grafovima tamno-sivom bojom, koji nam kao povratnu informaciju daje sam model. Unutar jednog dana mjerili bismo broj točaka koje bi bile van takvog intervala te na osnovu tog broja određivali vjerojatnost nepravilnog ponašanja. Te vjerojatnosti računali smo sigmoid funkcijom koju smo parametrizirali testiranjem na svim senzorima i strojevima koristeći znanje o ponašanju vibracija i popravcima.

U konačnom modelu odlučili smo pratiti prosječnu grešku mjerenja u zadnjih mjesec dana.

4.5 Matrix Profile

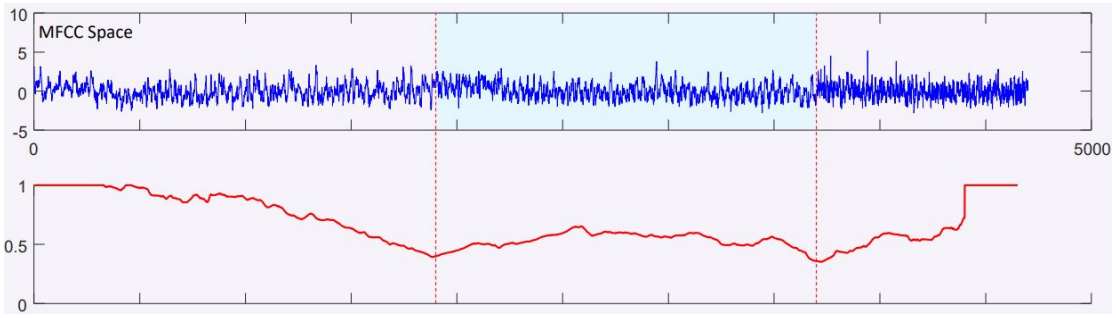
Matrix Profile je metoda koja prepoznaje promjene ponašanja u vremenskom nizu. Za dani vremenski niz i odabranu duljinu intervala prolazi po svim intervalima te duljine i određuje udaljenost do najbližeg sličnog intervala.



Slika 4.5.1 Demonstracija Matrix Profilea

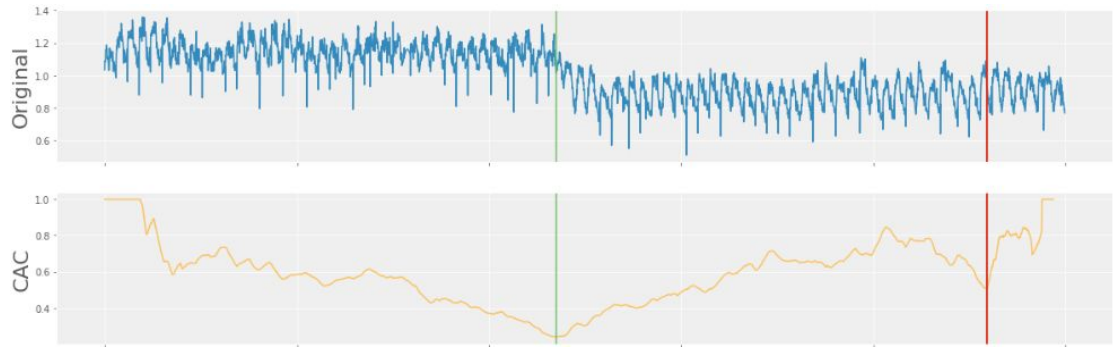
Crvenom bojom je prikazan originalni graf, dok je plavom bojom prikazan pripadni matrix profile. Minimumi u matrix profileu predstavljaju slične intervale koji se ponavljaju u nizu - na slici su isti označeni zelenim strelicama.

Iz podataka dobivenih matrix profileom možemo dobiti corrected arc curve (CAC). CAC se pokazao korisnim za prepoznavanje trenutaka u kojima je došlo do promjene u ponašanju podataka.



Slika 4.5.2 Primjer CAC-a

Mjesta lokalnih minimuma u CAC-u označavaju promjene u ponašanju podataka. Promatrajući CAC na originalnim podacima uočili smo javljanje takvih minimuma netom prije izvršenih popravaka, što nas je navelo da to iskoristimo u modelu. Pokazalo se najboljim promatrati samo najznačajnije lokalne minimume te uzeti samo one nakon kojih se javlja rastući trend uzevši u obzir da takvi trendovi uglavnom signaliziraju neispravno ponašanje.



Slika 4.5.3 Različiti lokalni minimumi u CAC-u

Na gornjoj slici plavom bojom su označeni originalni podaci, dok je žutom bojom označen CAC graf. Možemo vidjeti sa zelenom bojom označeni promjenu ponašanja koju smo dobili iz CAC grafa, no ta promjena ima trend prema dolje, pa nju ne uzimamo u završni model. Crvenom je bojom označena promjena ponašanja gdje postoji lokalni trend prema gore na grafu. Isto tako na grafu mozemo uociti da nisu označeni svi lokalni minimumi, već samo oni koji predstavljaju značajnije promjene.

4.6 Changepoint

Budući da je glavni cilj našeg pristupa bio traženje promjena ponašanja u vibracijama, ključno je bilo precizno određivati trenutke u kojima su se dogodile takve promjene, kako na originalnim podacima, tako i na rezultatima raznih transformacija (FFT, ICA, standardne devijacije). Iako Changepoint radi slično kao već spomenuti Breakout Detection, zapravo koriste različite matematičke alate za analizu, stoga smo ih koristili paralelno. Smatramo da to nije redundantno jer će završni model imati različite težinske vrijednosti povratnih informacija. Dapače, uzimamo to kao prednost zato što se međusobno nadopunjuju.

4.7 Finalni model

Postoje razni pristupi modeliranju koji su primjenjivi na naš problem. Jedan od pristupa je regresijski model koji predviđa RUL (remaining useful lifetime). Zatim, opcija je bio i klasifikacijski model koji predviđa hoće li se stroj pokvariti u nekom vremenskom prozoru. Mogli smo imati i model koji detektira anomalije u novim podacima i na osnovu njih određuje stanje stroja.

Odlučili smo modelirati trenutno stanje stroja, odnosno potrebu za popravkom određenog dijela stroja u narednih 30 dana. Kako se ponašanja različitih dijelova stroja mijenjaju kroz vrijeme, što nekada ne sadrži informacije o krivom radu, te kako se mjerenja nekih senzora jako razlikuju od stroja do stroja, modelu dajemo informacije samo o zadnjih mjesec dana rada stroja. Smatrali smo da je to vremensko razdoblje dovoljno dugo da detektiramo grešku u radu na vrijeme, a opet dovoljno kratko da na rezultat ne utječu predaleki podaci iz prošlosti. Ako je prije više od 30 dana i došlo do promjene ponašanja u radu stroja, a nema

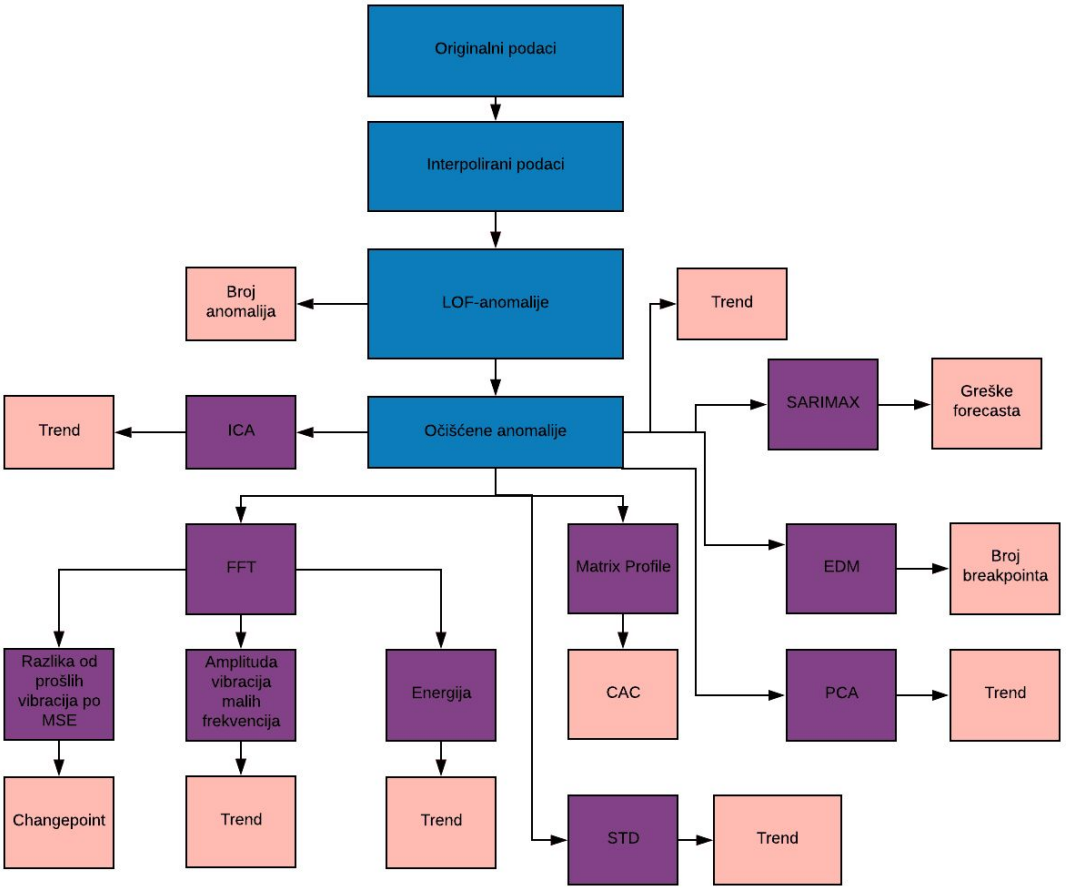
drugih indikatora kvara u zadnjih mjesec dana, smatrali smo da je prošlo dovoljno vremena u kojem je stroj radio normalno da tu promjenu možemo zanemariti.

Zbog malo podataka o popravcima te teže interpretabilnosti, odustali smo od kompliciranih machine learning modela. Kada bismo imali više podataka o popravcima, mogli bismo ih lakše koristiti te bi bilo zanimljivo vidjeti hoće li naći neke veze među podacima koje mi nismo vidjeli.

Odlučili smo da završni rezultat dobivamo kao težinski prosjek faktora za koje smo vidjeli da pozitivno utječu na potrebu za popravkom. Prednost je što u svakom trenutku znamo zbog čega nam model javlja treba li ili ne treba popravak te se kroz vrijeme može automatski, ali i ručno, ponovno podesiti da odgovara novom ponašanju. Isto tako, onaj tko će taj model koristiti može staviti manji ili veći naglasak na neki od faktora, ovisno o onome što želi saznati.

Kako izvorna jačina vibracija nije dobar pokazatelj potrebe za popravkom (npr. FL05), svi faktori koje smo odabrali su neosjetljivi na red veličine vrijednosti mjerenja (rastući trend, detekcija anomalija…). Dodatno smo u svakom koraku dani signal normalizirali.

Prikažimo model:



Slika 4.7.1 Shematski prikaz konačnog modela, zadnja razina (roza) predstavlja faktore koji ulaze u konačnu linearnu regresiju.

Ono što se na prvu čini kao najbolje rješenje automatskog traženja koeficijenata je metoda najmanjih kvadrata. Problem je što već znamo da naši faktori pozitivno utječu na potrebu za popravkom i ne bi smjeli biti negativni. Zato smo se odlučili za metodu nenegativnih najmanjih kvadrata, gdje vraćeni koeficijenti neće biti negativni, kao što smo i htjeli.

Zadnja razina gornje sheme modela predstavlja ulaznu komponentu konačnog modela. Svaki navedeni faktor nam za dani senzor daje numeričku vrijednost koja je proporcionalna potrebi za popravkom. Podatke smo grupirali po danu.

machine	sensor_type	day	n_outliers	n_trends	changepoint	ica_n_trends	ica_changepoint	n_std_trends	energy_n_trends	energy_changepoint
FL03	drive_gear_a_max	2019-05-05	4.0	1.0	4.0	0.0	0.0	3.0	1.0	0.0
FL07	idle_wheel_a_max	2018-05-09	2.0	0.0	0.0	0.0	7.0	1.0	0.0	0.0

Slika 4.7.2 Primjer izgleda podataka

U tako napravljenom datasetu, postoji velika neravnoteža u našoj ciljanoj varijabli (samo 3.2% dana su najviše 30 dana udaljeni od popravka). Odlučili smo balansirati podatke, tako da smanjimo broj negativnih i povećamo broj primjeraka. Za to smo koristili SMOTEENN, koji stvara sintetičke pozitivne primjerke kao kombinacije nekih drugih pozitivnih, te briše negativne primjerke slične pozitivnima. Prednost balansiranja podataka je i povećavanje preciznost i odziva modela, na štetu točnosti, što je odgovaralo našem problemu. Smatrali smo da je bolje češće krivo javljati potrebu za popravkom, nego propustiti neki potreban popravak.

Dobivenim vrijednostima smo metodom nenegativnih najmanjih kvadrata pridružili koeficijente za klasifikaciju strojeva ovisno o potrebi za popravkom u narednih 30 dana, pri čemu smo iskoristili podatke o izvršenim popravcima.

Time smo za svaki senzor dobili vjerojatnost kvara na tom dijelu stroja. Kako bismo odredili konačnu vjerojatnost kvara stroja, na te rezultate ponovno smo primijenili metodu nenegativnih najmanjih kvadrata. Time smo dobili konačnu vjerojatnost potrebe za popravkom odabranog stroja.

Prednosti ovog pristupa je što u svakom trenu znamo točnu vjerojatnost kvara koju javlja svaki senzor, što nam govori o stanju tog dijela stroja, te brzina evaluacije u bilo kojem trenutku. Isto tako, odmah dobijemo informaciju o važnosti senzora – to je pripadni koeficijent uz vrijednost u modelu.

Treća prednost je velika interpretabilnost, jednostavnost korištenja i održavanja modela.

5. EVALUATION

Za treniranje i validaciju modela bile su dvije metoda koje su nam se činile logične: vremenska validacija ili treniranje na jednom skupu strojeva i validacija na drugom.

Prednost svake vremenske validacije je što simulira stvarno korištenje modela. Rezultati validacije o točnosti modela su upravo oni koje bi dobili da smo ga koristili u tom trenutku u prošlosti.

Da su naši podaci drugačije izgledali, vjerojatno bi tako validirali model. Problem je bio što se popravci na FL02, FL03, FL07 javljaju u istom periodu, dva tjedna nakon prvih mjerenja koja nisu bila testna, te nakon tog razdoblja više nema popravaka ni na jednom stroju. Prije toga imamo samo mjerenja na FL01 i FL07. Kada bismo koristili vremensku validaciju, u prvim periodima model bi mogao naučiti uzorke ponašanja prije kvara samo iz podataka mjerenja FL01, koji sadrže nestandardna ponašanja koja se kasnije ne ponavljaju. Isto tako, nakon tog perioda popravaka, kako više nema niti jednog popravka, ne bismo mogli ocijeniti hoće li uspješno označiti kada je potreban popravak.

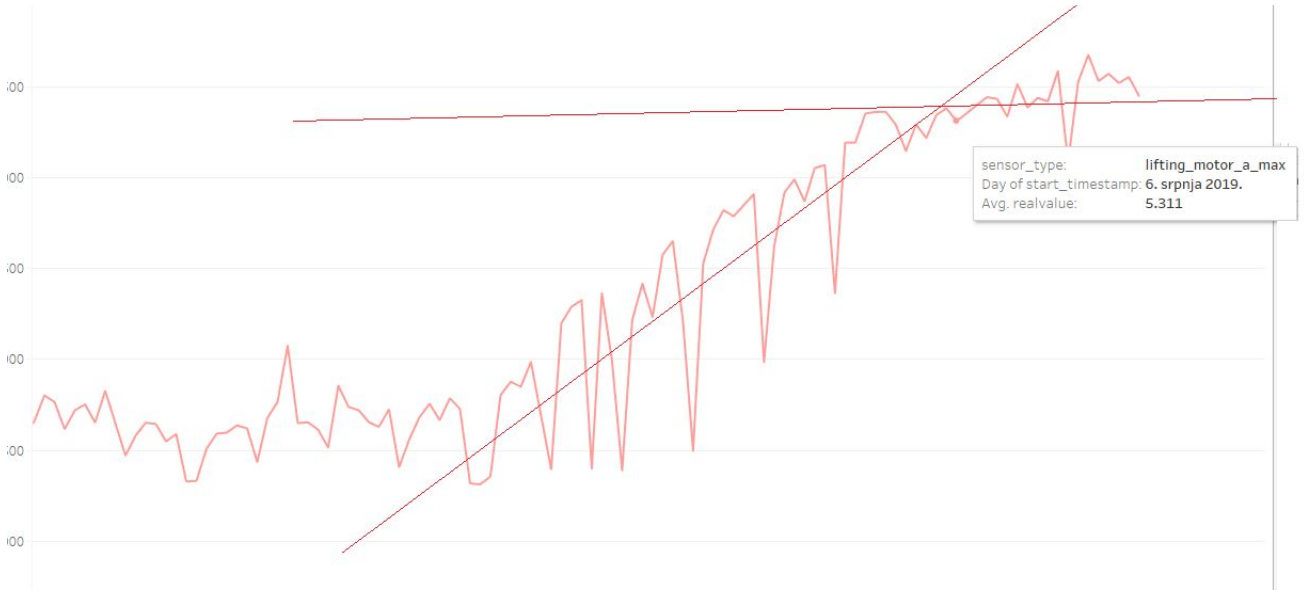
Model smo trenirali na mjerenjima strojeva FL03, FL04, FL05, FL06 i FL07, te validirali na FL01 i FL02. Za FL01 smo gledali samo mjerenja nakon 19.3.2019., jer je prije toga još trajalo testiranje. Zadnjih mjesec dana svih mjerenja smo maknuli iz treninga i validacije jer ne znamo ništa o njihovom popravku u idućih mjesec dana. Bila bi ih greška ostaviti.

	drive_gear_V_eff	drive_gear_a_max	drive_motor_V_eff	drive_motor_a_max	drive_wheel_V_eff
n_outliers	0.053284	0.004500	0.041197	0.004500	0.023547
n_trends	0.004500	0.086433	0.117738	0.004500	0.004500
changepoint	0.004500	0.004500	0.067822	0.004500	0.004500
ica_n_trends	0.004500	0.004500	0.004500	0.004500	0.004500
ica_changepoint	0.043748	0.000653	0.038114	0.004500	0.039266
n_std_trends	0.004500	0.004500	0.004500	0.004500	0.004500
energy_n_trends	0.004226	0.004500	0.004500	0.004500	0.004500
energy_changepoint	0.004500	0.004500	0.004500	0.013253	0.004500
vibChange_n_trends	0.081381	0.004500	0.004500	0.004500	0.079805
vibChange_changepoint	0.004500	0.004500	0.004500	0.004500	0.070483
highFreq_n_trends	0.004500	0.004500	0.004500	0.004500	0.060670
highFreq_changepoint	0.004500	0.004500	0.004500	0.051271	0.004500
EDM	0.004500	0.004500	0.043192	0.004500	0.004500
matrixProfile	0.004500	0.004500	0.004500	0.130716	0.004500
sarimax	0.004500	2.237234	0.004500	2.411146	0.004500
WEIGHT	0.004500	0.450000	0.004500	0.450000	0.004500

Slika 5.1 Izračunati koeficijenti nakon treniranja modela

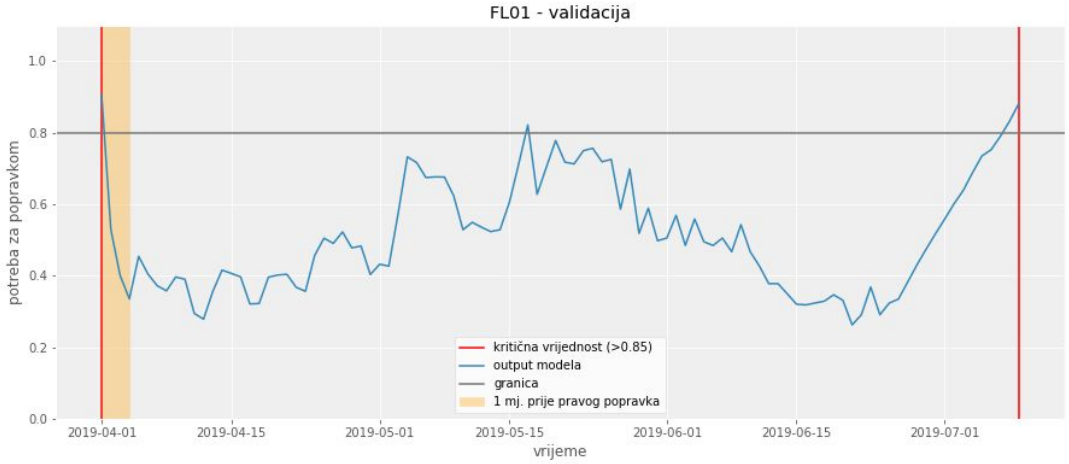
Na početku smo planirali odabrati FL07 za validaciju jer je imala dug period mjerenja i jedan popravak. Problem je bio što tada model nema niti jedan primjerak svih 30 dana prije popravka. Za validaciju smo odabrali FL01 i FL02 jer imaju dva popravka, ali i period normalnog ponašanja. Smatrali smo da ćemo tako dobiti ispravnu sliku o uspješnosti modela. Nadalje, sumnjali smo da je došlo do popravka 24.10.2017. na

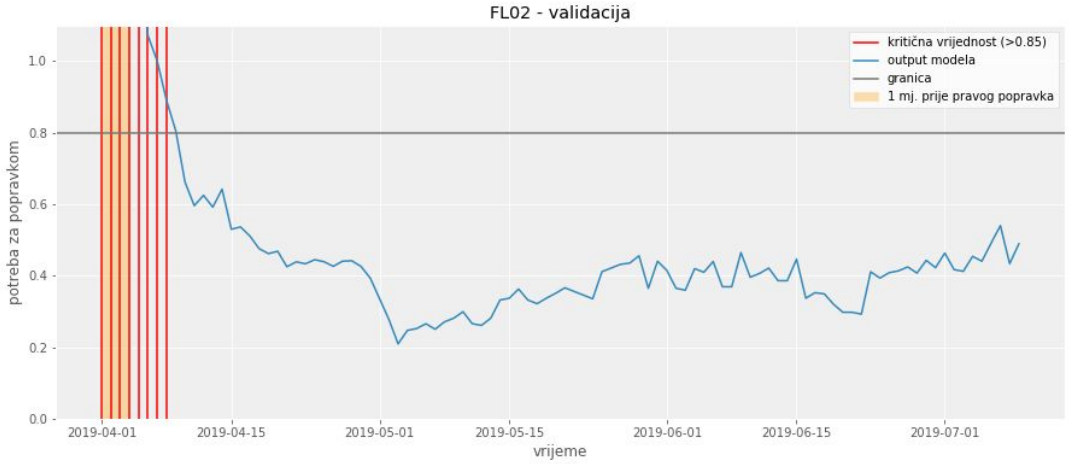
FL07 pa smo ga dodali kao popravak koji se dogodio. Isto tako, kako nije bilo popravaka na lifting senzorima, dodali smo da je došlo do popravka lifting senzora 7.6.2019. na FL05. Može se vidjeti rastući trend prije tog datuma, koji nakon toga nestaje. Čak i da tada nije bilo popravka, zaključili smo da je to ponašanje koje bismo htjeli detektirati.



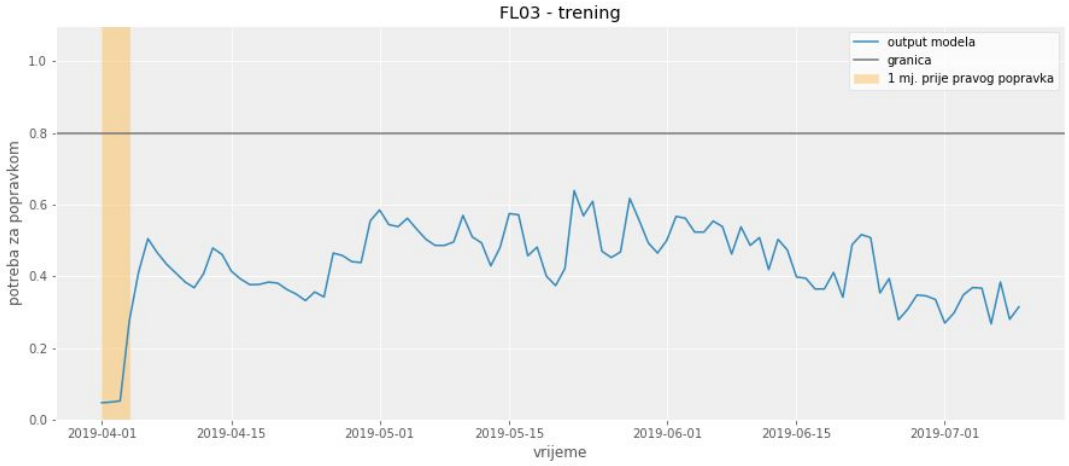
Slika 5.2 Dodani popravak na FL05

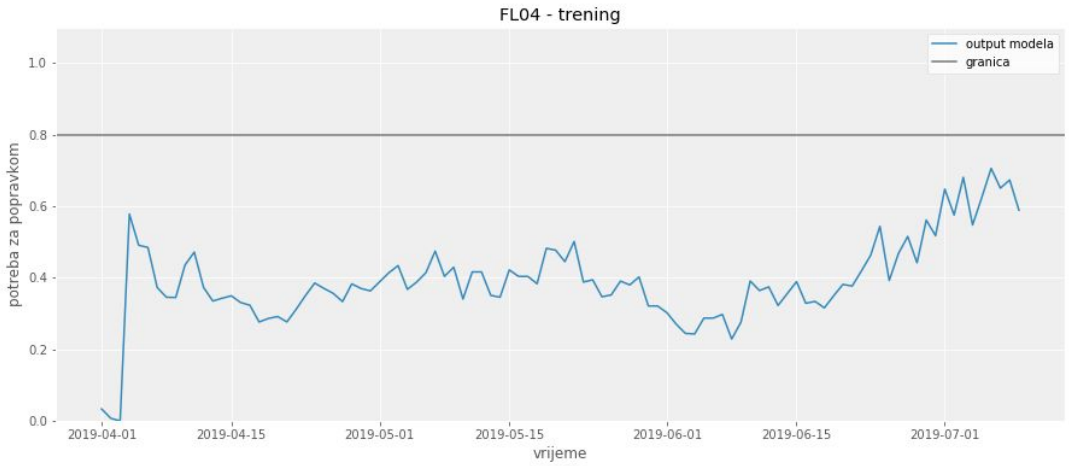
Na donjim slikama prikazana je povratna informacija našeg modela na svim strojevima, podijeljena na validaciju i mjerenja na kojima je model treniran. Plave vrijednosti predstavljaju potrebu za popravkom u određenom danu. Posebno smo crvenom bojom označili rubne vrijednosti koje sugeriraju obavezan popravak. Napomenimo da narančasta linija označava mjesec dana prije stvarnog popravka.

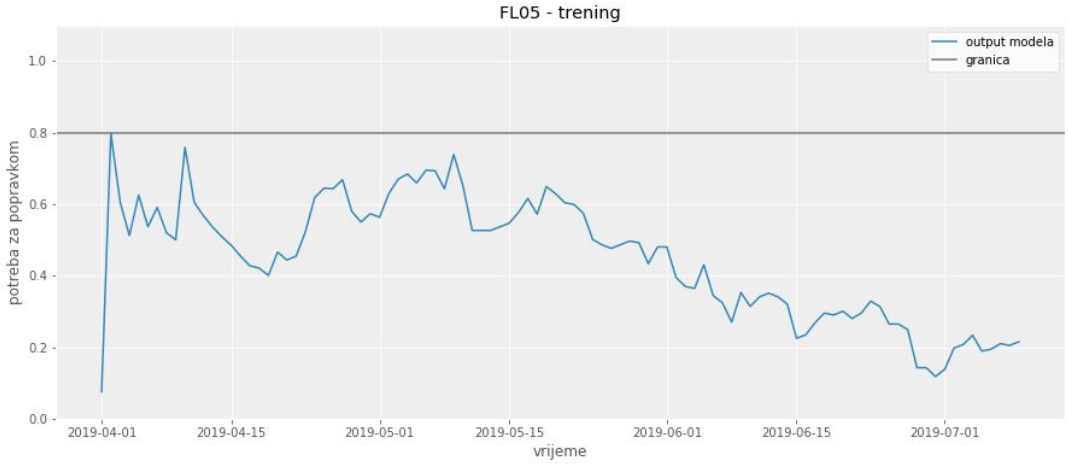


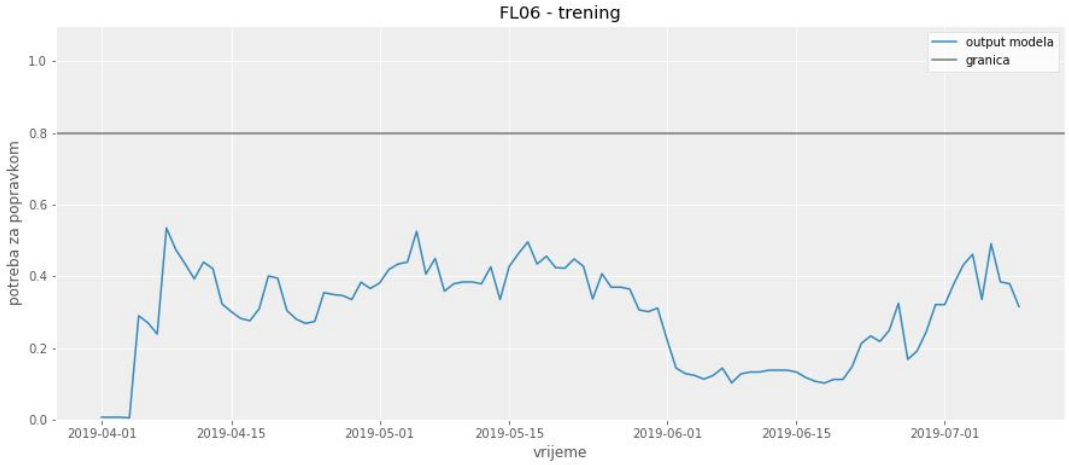


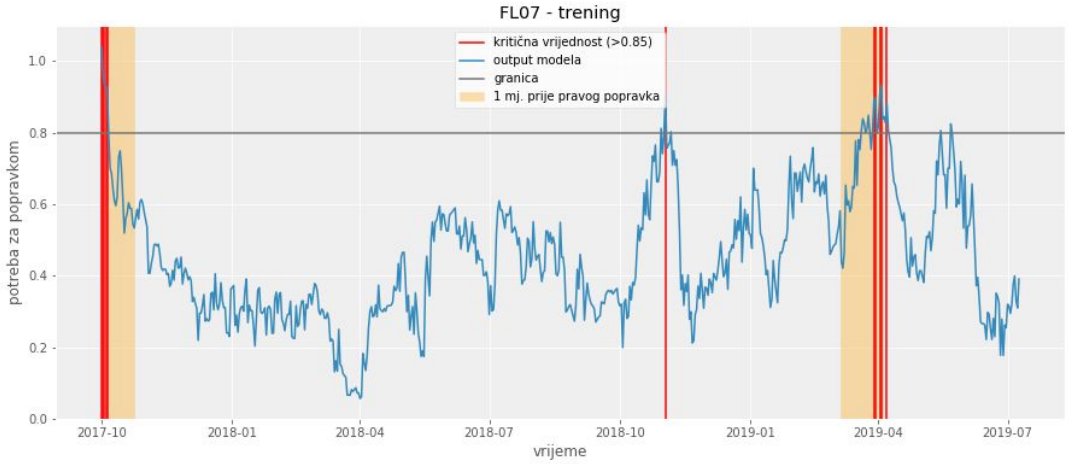
Slika 5.3 Rezultati nakon validacije na FL01 i FL02



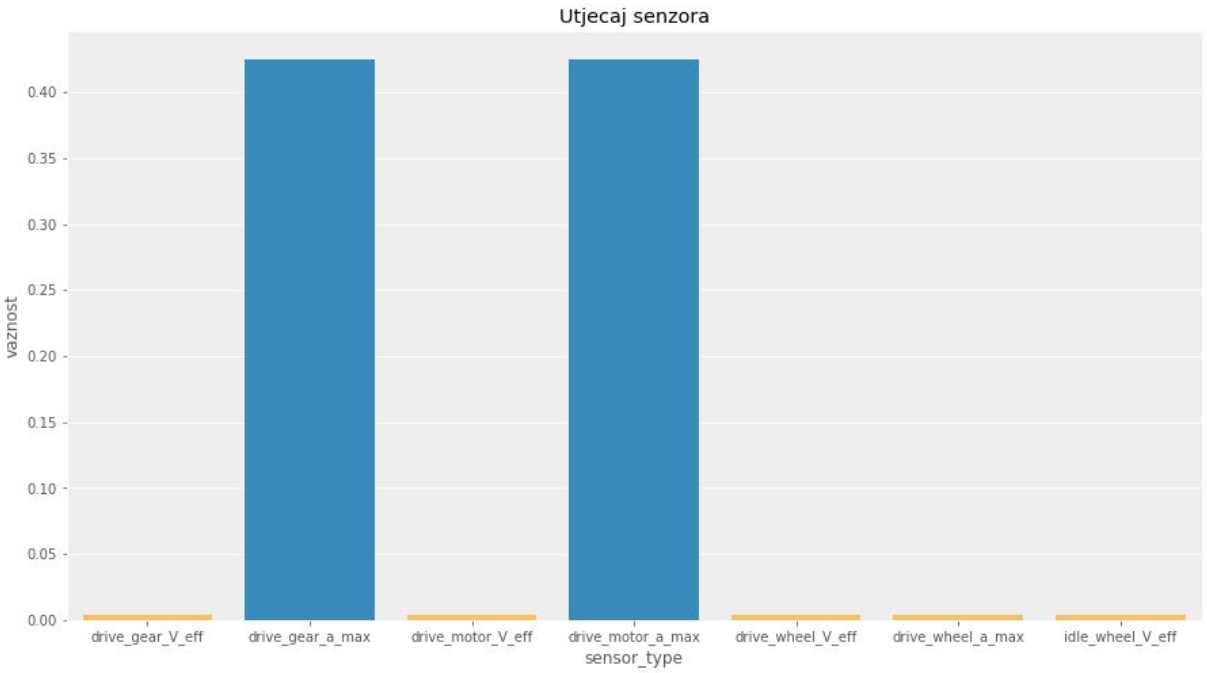




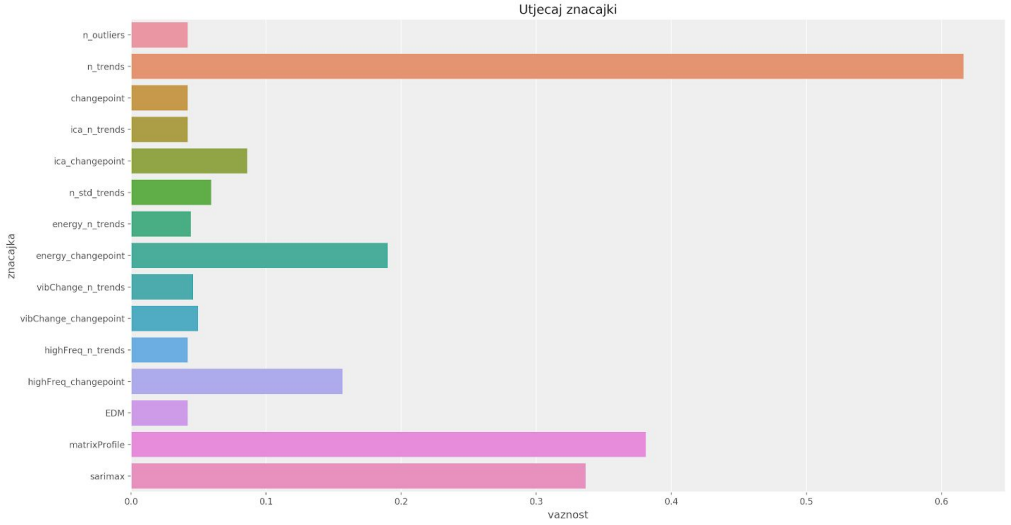




Slika 5.4 Rezultati nakon treninga na FL03, FL04, FL05, FL06, FL07



Slika 5.5 Utjecaj drive i idle senzora na potrebu za popravkom



Slika 5.6 Utjecaj značajki na potrebu za popravkom.

6. DEPLOYMENT

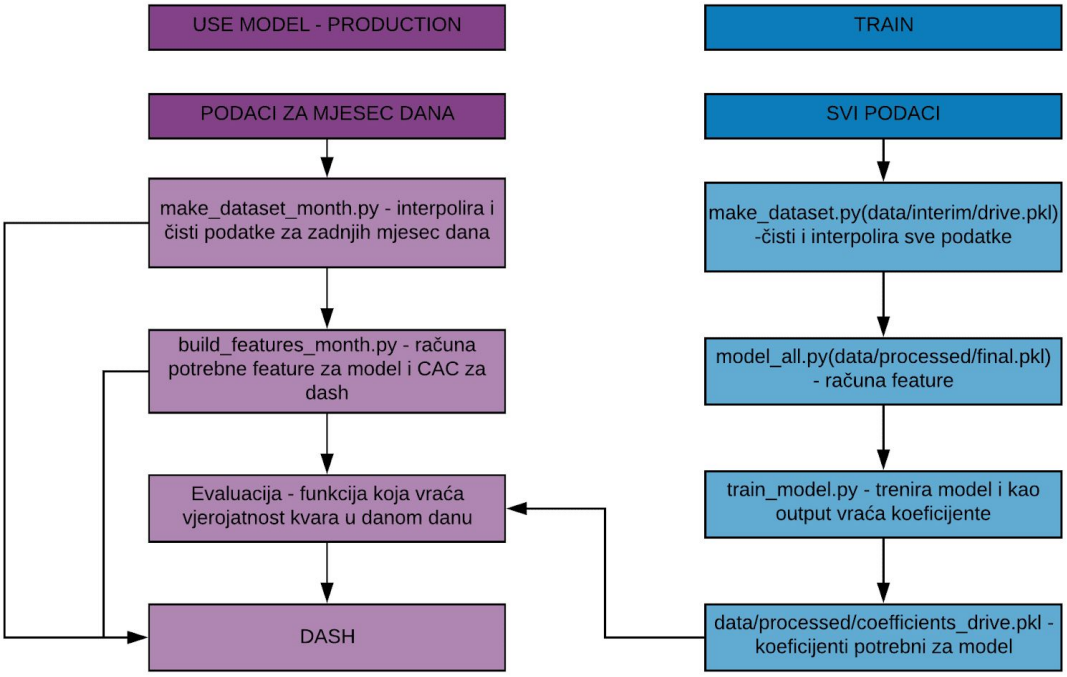
Za pregled rezultata našeg modela dizajnirali smo sučelje napravljeno u Dashu. Podaci potrebni za pravilan rad sučelja izgenerirani su programom “src/update.py”. Detaljnije o radu sa sučeljem možete pročitati u tehničkoj dokumentaciji. Nakon što su svi potrebni podaci izgenerirani sučelje nudi opciju pregledavanja rezultata modela. Isto tako moguće je crtanje orginalnih grafova za svaki senzor svakog stroja, ali i crtanje grafova za neke od metoda koje su upotrebljene u modelu kao što su CAC, graf energije, graf energija visokih frenkvencija. Te sučelje nudi različite metode za detekciju “breakouta” poput EDM-a i metode koju smo opisali ranije u dokumentaciji pomoću CAC-a. Na postojeće sučelje lagano je nadogradi funkcionalnost za preostale metode našeg modela.

Priprema podataka i računanje značajki potrebnih za model traje približno deset minuta, dok je za evaluaciju potrebno nekoliko sekundi. Smatramo da je ta brzina izvršavanja dovoljno dobra jer se ovaj proces treba ponavljati samo jednom dnevno. Brzo i jednostavno se mogu isprobati i neke nove vrijednosti koeficijenata modela.

Uz sve to imamo i spomenutu dash aplikaciju koja nudi vizualni prikaz trenutnih vibracija, što bi omogućilo jednostavnu provjeru trenutnog stanja vibracija kako bi se ustanovilo ima li stvarno potrebe za popravkom. Time bismo povezali primjenu modela i stručno znanje osobe koja bi interpretirala dobivene rezultate.

Lako bi bilo implementirati i jednostavno online učenje; svaki dan treba oduzeti od koeficijenata težinu skaliranu s njihovom vrijednošću za taj dan, te ih povećati kad dođe popravak. Vjerujemo da će naš model biti još uspješniji kroz vrijeme, kada će postojati više označenih popravaka.

Na sljedećem dijagramu prikazan je način na koji obrađujemo podatke za treniranje i evaluaciju modela.



Slika 6.1 Pipeline za obradu podataka i evaluaciju modela

7. APPENDIX

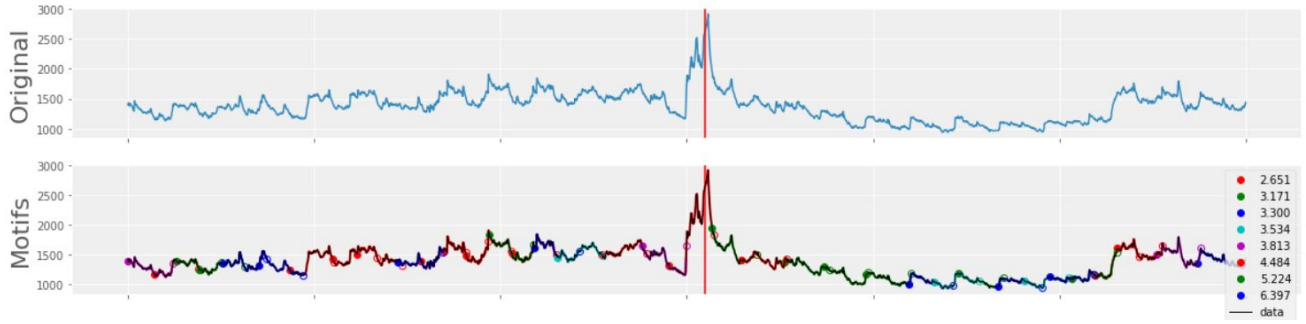
7.1 Neuspjeli pokušaji i ideje za nastavak

7.1.1 State Machine

Jedna od ideja bila je ponašanje stroja modelirati nekim oblikom state machinea, gdje bismo svakom intervalu rada pridružili neko stanje. Namjeravali smo promatrati prelaske iz stanja u stanje i korištenjem teorije Markovljevih lanaca otkriti obrasce koji bi dobro predviđali potrebu za popravkom. U konačnici nismo uspjeli postići neke konkretne rezultate koje bismo iskoristili u modelu.

7.1.2 Motifs

Minimumi na matrix profile grafu predstavljaju uzorke ('motifs') koji se ponavljaju u vremenskom nizu. U našem slučaju ideja je bila vidjeti sugeriraju li takvi uzorci potrebu za popravkom. Nažalost, nismo uspjeli uočiti nikakvu pravilnost koju bismo mogli iskoristiti u modelu. Na donjem grafu crvenom linijom označen je popravak, možemo vidjeti da nema nikakvog uzorka u pojavljivanju motifsa prije i poslije popravka.

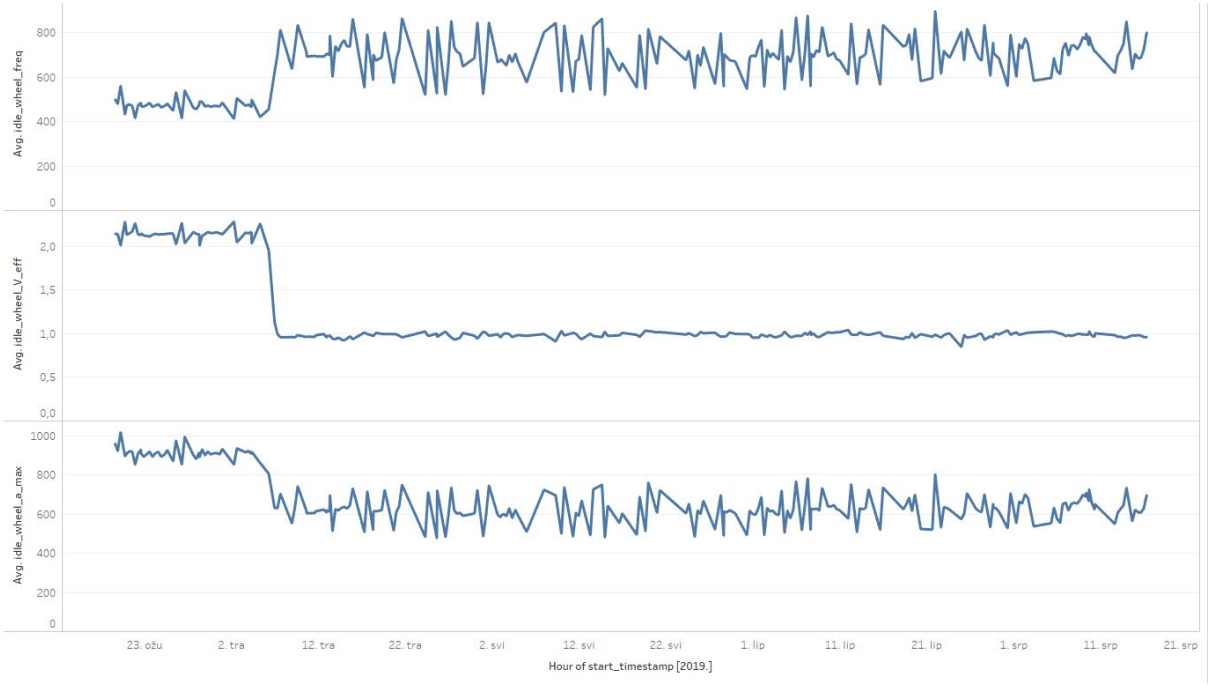


Slika 7.1.2.1 Graf koji prikazuje “motifse”, svaka boja predstavlja različiti uzorak

7.1.3 a_max/V_eff

Proučavanjem literature o analizi vibracija dobili smo ideju da promatramo omjer maksimalne akceleracije i efektivne brzine u nekom trenutku. Naime, kad bi naše vibracije bile najjednostavniji sinusoidni valovi, taj omjer davao bi kutnu brzinu, iz čega se lako izračuna frekvencija naših vibracija.

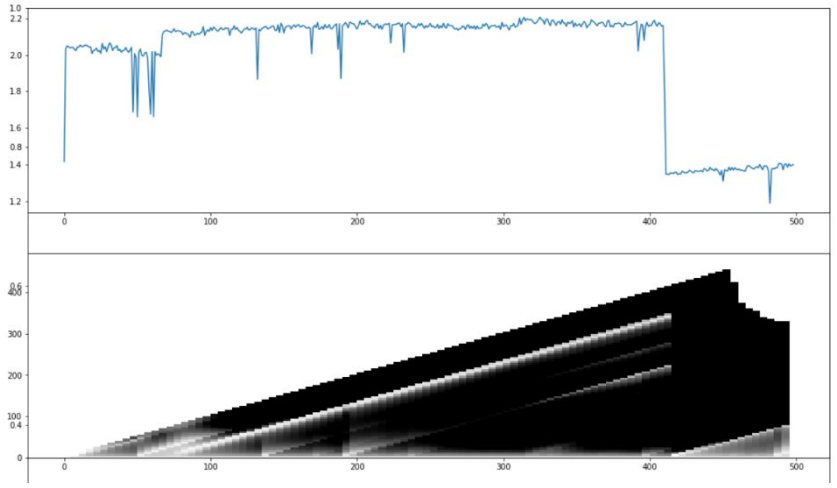
Na našim podacima taj omjer nije davao korisne informacije. Često je efektivna brzina konstantna pa naša 'kutna brzina' samo prati maksimalnu akceleraciju po vrijednostima.



Slika 7.1.3.1 Prikaz “kutne brzine”, V_eff i a_max redom

7.1.4 Bayesian online changepoint detection

Jedan od changepoint algoritama koje smo proučavali bio je temeljen na Bayesovskoj statistici i optimiziran za online rad. Output bi u svakom trenutku davao vjerojatnost da smo ušli u period novog ponašanja.



Slika 7.1.4.1 Bayesian online changepoint – primjer povratne informacije

Na gornjem grafu bijele linije predstavljaju vjerojatnosti da je vremenski niz ušao u novo stanje.

Kako dobiveni rezultati nisu bili posebno bolji od drugih, jednostavnijih algoritama, i zbog nedostatka kvalitetne implementacije spomenutog algoritma, odustali smo od korištenja istog.

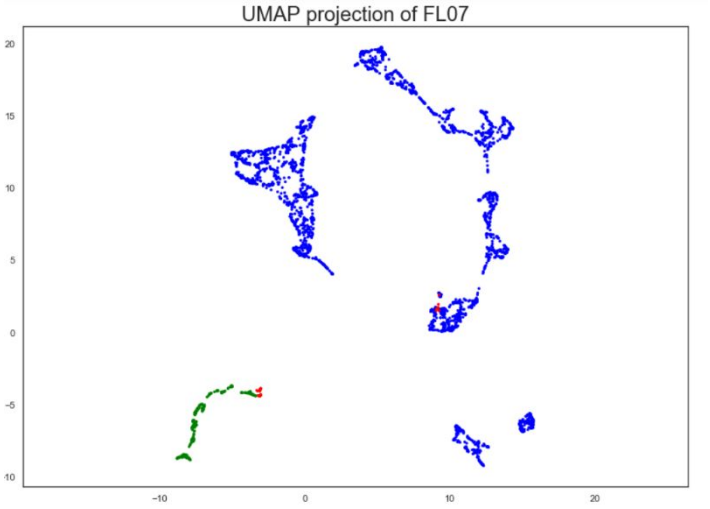
7.1.5 Odvajanje strojeva

Model koji smo predložili može se koristiti za svaki od strojeva. S druge strane, razmatrali smo i mogućnost da svaki stroj modeliramo zasebno, no problem je bio nedostatak podataka o kvarovima za neke strojeve, a i otežano korištenje za nove strojeve u budućnosti.

7.1.6 UMAP

Jedna od metoda smanjenja dimenzionalnosti koju smo isprobali je UMAP (Uniform Manifold Approximation and Projection) koja koristi tehnike iz algebarske topologije kako bi visokodimenzionalne podatke projicirala u 2 ili 3 dimenzije. Dobiveni rezultati su nam demonstrirali da se bitne informacije u našim podacima mogu prikazati pomoću manje dimenzija. Naime, na dobivenim grafovima su različiti periodi ponašanja vibracija jasno razdvojeni.

Na sljedećem grafu plavo su označene točke prije popravka, dok su zeleno označene one nakon, a crvene su točke one koje su u tjedan dana prije popravka. Vidimo da su promjene ponašanja očuvane i da se prepoznaje struktura u podacima koju možemo vidjeti i na originalnim signalima.



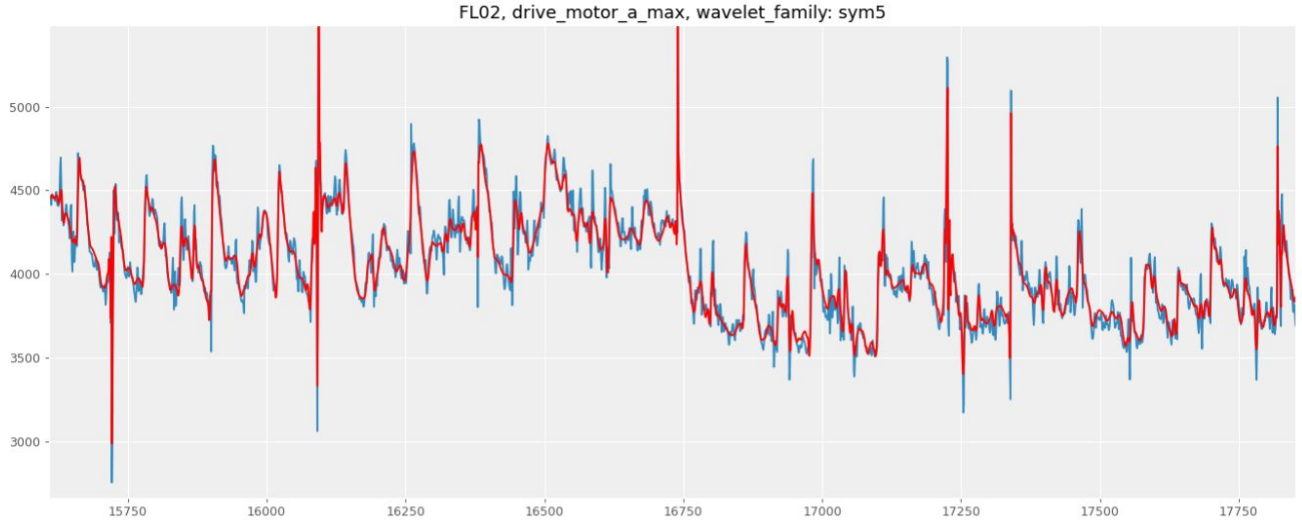
Slika 7.1.6.1 UMAP projekcija svih vibracija za FL07

U konačnom modelu ipak nismo koristili ovu metodu, nego smo se odlučili za klasične metode poput PCA (Principal Component Analysis) i ICA (Independent Component Analysis). Glavni je razlog bio što ove metode daju interpretabilnije rezultate i razdvajaju podatke po određenim kriterijima, što se odlično slagalo s metodama koje smo primjenjivali na rezultatima ovih transformacija.

7.1.7 Wavelet Transformation

Wavelet Transform je još jedna metoda analize signala. Prednost korištenja Wavelet Transforma naspram, primjerice, Fourierove analize jest u tome što se i frekvencija lokalizira u vremenu, točnije ne izgubimo i tu vrlo bitnu vremensku komponentu pojave određene frekvencije. Naše korištenje waveleta svelo se na to da smo htjeli očistiti podatke od šuma (noisea) tako što bismo rekonstruirali graf na temelju tzv. „koeficijanata aproksimacije.“ Bogatstvo wavelet transforma dolazi do izražaja u širokom spektru mogućih familija za

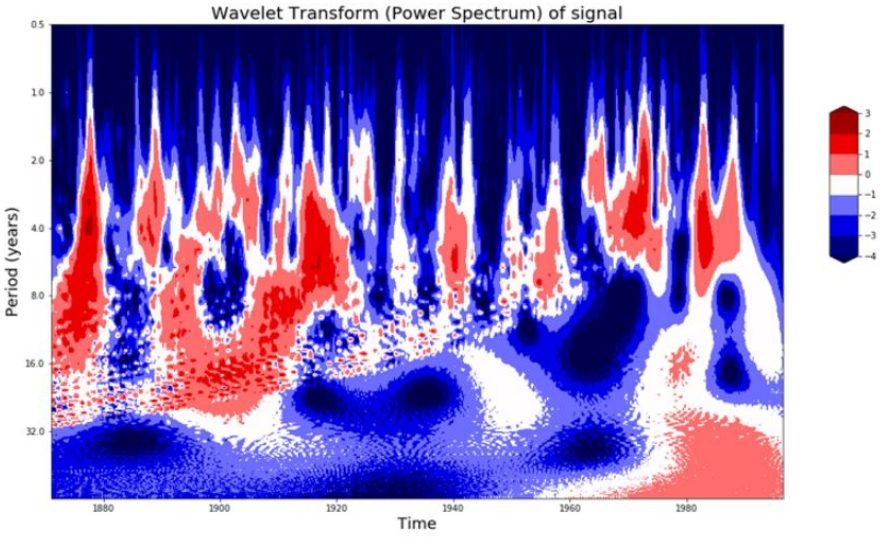
korištenje pri radu. Budući da su grafovi nakon čišćenja bili suviše glatki, odustali smo od te metode da ne izgubimo informacije o stvarnom titranju. Takva greška u modelu previše bi utjecala na krajnje rezultate, koji možda više ne bi odražavali stvarno stanje.



Slika 7.1.7.1 Izbacivanje šuma koristeći wavelet familiju „sym5“

Transformacije waveletima imaju brojne primjene sa signalima koji potječu, kao i naši, od senzora na uređajima.

Neprekidni Wavelet (CWT) ukomponiran s CNN-om (Convolutional Neural Network) tvori izrazito jak alat u klasifikaciji signala. Ideja je da neuronska mreža analizira spektrograme (graf amplitude, frekvencije i vremena) dobivene wavelet transformacijom. Jedna od primjena je u klasifikaciji EKG-a u ovisnosti o tome potječe li od zdrave osobe ili osobe sa srčanom manom. U našem slučaju to nismo mogli iskoristiti jer nemamo podjelu na „dobre“ i „loše“ signale budući da nam podaci sadrže premalo etiketiranih popravaka. Smatramo da neuronska mreža iz tog razloga ne bi bila u stanju točno generalizirati.



Slika 7.1.7.2 Spektrogram nakon transformacije waveletom

7.1.8 Nonequispaced fast Fourier transform

Fourierovu analizu nam je otežavalo što točke nisu uvijek bile jednako vremenski udaljene od mjerenja do mjerenja. Nakon interpolacije 60, odnosno 70 točaka po mjerenju, na dužim mjerenjima bi točke bile raspršenije, dok bi na kraćim bile gušće. Rješenje tog problema bio bi primijeniti NFFT, umjesto FFT jer NFFT ne očekuje jednaku udaljenost među točkama. Od toga smo odustali jer je NFFT bio osjetljiv na red veličine vremenskog podatka. Tako bi npr. davao potpuno drugačiji oblik grafa ako bi vremenske vrijednosti bile skalirane nekim faktorom. Drugi problem bio je dugo vrijeme izvršavanja, što uzima od izvodljivosti našeg modela.