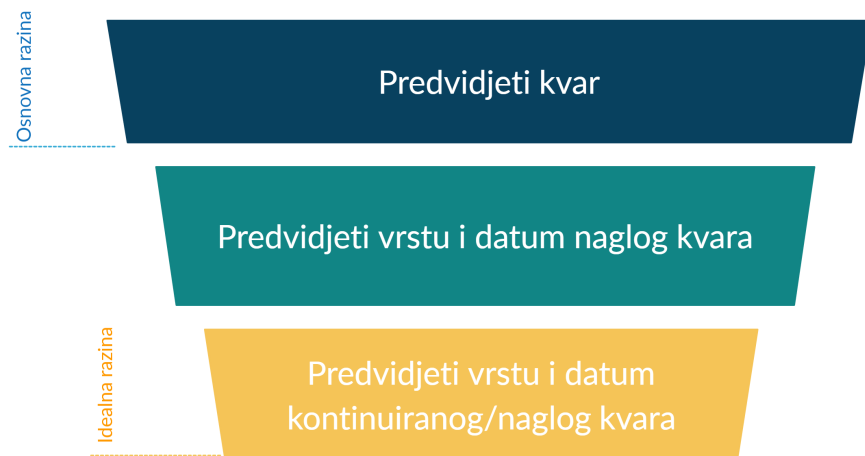


Lumen Data Science 2020: Predictive maintenance

1 Uvod

Predictive maintenance (PdM) je skup tehnologija kojima je namjera periodički ili u realnom vremenu prikupljati podatke kako bi evaluirali stanje uređaja i odredili financijski optimalni trenutak za održavanje. Motivacija takvog modela održavanja je što kvar čak i malih dijelova nekog uređaja može dovesti do velikih troškova koji se ne mjere samo u tome koliko košta uređaj nego i povezanih troškova, primjerice privremeno prekidanje proizvodnje. Isto tako možda nema rezervnih dijelova pa može proteći neko vrijeme do popravka što je dodatni teret. Kako bi se predvidjelo stanje uređaja koriste se mnoge neinvazivne tehnike poput infracrvenog zračenja, akustike, analize ulja te mnoge druge. Mi ćemo u ovom radu proučiti analizu vibracija koja se pokazala korisnom u području predictive maintenance-a. Koristeći dataset koji je dobiven za natjecanje "Lumen Data Science 2020" želimo konstruirati model kojim možemo unaprijed predvidjeti kvar uređaja. Ciljevi ovog projekta su prikazani na slici 1, no mogućnosti ostvarivanja istih ovise o samom modelu i podacima.



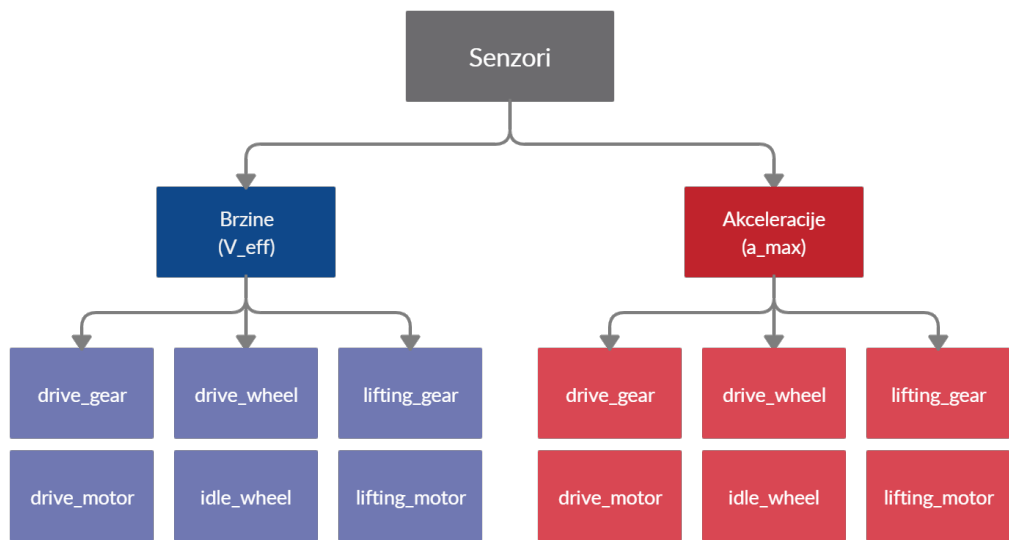
Slika 1: Hijerarhija ciljeva projekta.

2 Eksploratorna analiza

Dataset koji smo dobili sastoji se od sedam stupaca te gotovo 4×10^6 redaka i sadrži mjerenja 12 senzora na svakom od sedam strojeva, kao što je prikazano na slici 2. Postoje dvije skupine senzora - senzori koje mjere efektivne brzine (RMS)

$$V_{\text{RMS}} = \sqrt{\frac{1}{n} [v(t_1)^2 + v(t_2)^2 + \dots + v(t_n)^2]}$$

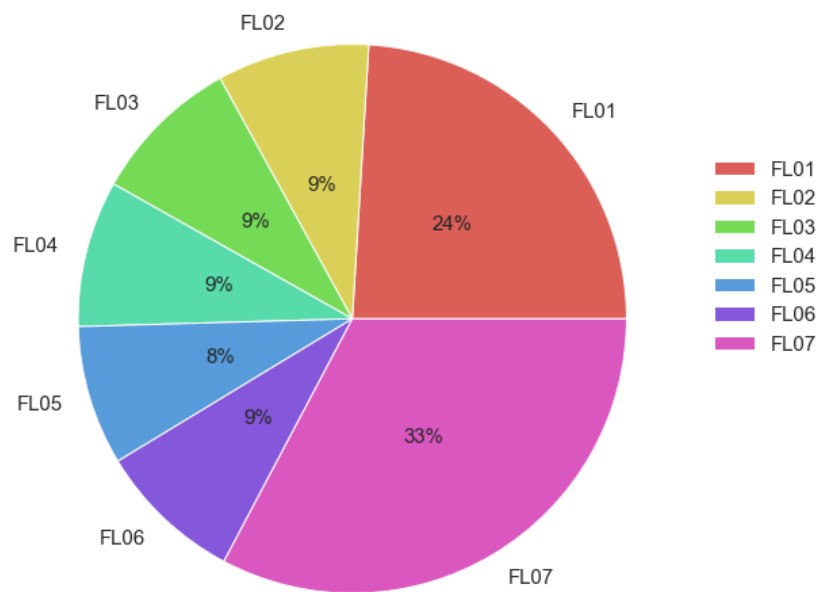
te maksimalne akceleracije (maksimalni iznos akceleracije između dva mjerenja). Naime, uređaji su prije imali puno više senzora, no dugoročnim promatranjem je većina bila izbačena jer nisu bili korisni. Iz tog razloga, dva uređaja (FL01 i FL07) imaju znatno veći vremenski raspon mjerenja od skoro dvije godine, dok ostali strojevi imaju mjerenja za puno kraći period (nešto više od pola godine). Točan udio podataka od strojeva prikazan je na slici 3. Uz podatke smo dobili i datoteku koja nam daje opis sedam ljudskih intervencija nad uređajima te kada i na kojem uređaju se dogodila. Pošto je takva intervencija indikacija kvara na uređaju vidimo da se sami kvarovi događaju vrlo rijetko.



Slika 2: Kategorizacija senzora na strojevima.

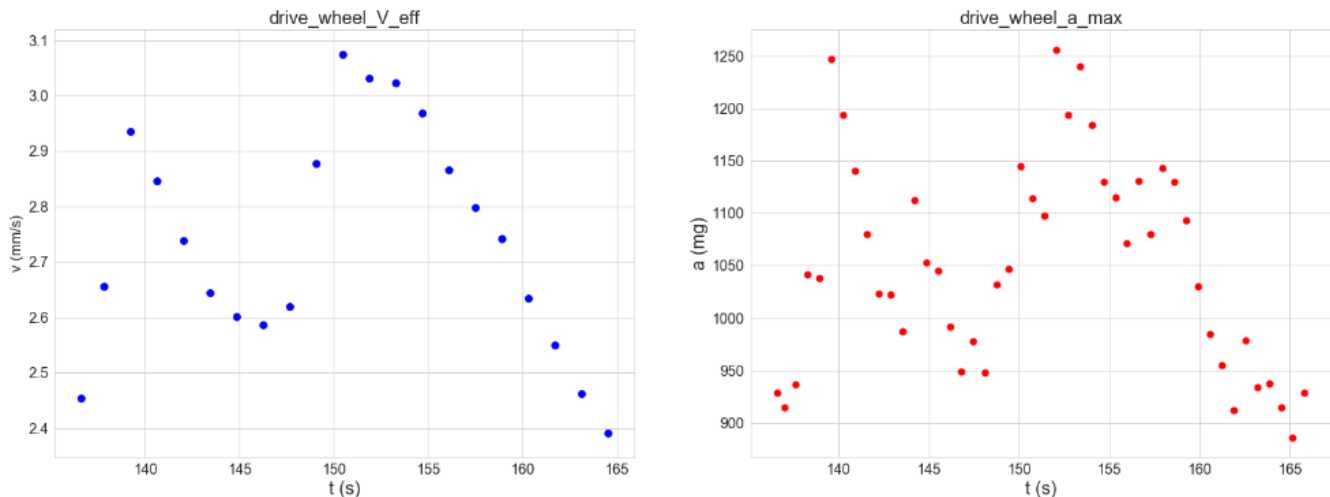
Bitno je naglasiti da mjerenja nisu kontinuirana, već se mjerenja izvršavaju otprilike svakih 8 sati, traju oko minute i generiraju oko 20-30 točaka (dakle svake 2-3 sekunde se dobije novo mjerenje). Ovakav dataset nam stvara komplikacije jer ne možemo točno automatizirano grupirati mjerenja koja su se događala u istom "sessionu". "Session" definiramo kao period od nekoliko minuta u kojem su kontinuirano prikupljeni podaci. Iz tog razloga, kako bismo imali organiziranije podatke napravili smo kod koji se nalazi u bilježnici "Lumen - creating session dataframe". Postupkom koji je detaljnije opisan u bilježnici grupirali smo mjerenja iz istog sessiona, zahtjevajući da mjerenja nisu razmaknuta za više od jedne minute. Dakle, ako je jedno mjerenje udaljeno od drugog za više od minute, drugo mjerenje ćemo grupirati idućem sessionu mjerenja (najčešće su mjerenja razmaknuta barem za nekoliko sati). Dataframe koji sadrži "session" stupac koristimo za sve daljnje analize. Opis 8 stupaca dataset-a nalazi se:

- `machine_name` - redni broj uređaja čija mjerenja promatramo (FL01 - FL07)
- `sensor_type` - vrsta senzora čija mjerenja promatramo (jedan od 12 gore navedenih senzora)
- `date_measurement` - datum mjerenja (YY:MM:DD format)
- `start_timestamp` - vrijeme u koje je započeto mjerenje (YY:MM:DD HH-MM-SS.MS format)
- `end_timestamp` - vrijeme u koje je završeno mjerenje (YY:MM:DD HH-MM-SS.MS format)
- `realvalue` - vrijednost mjerenja
- `unit` - mjerna jedinica mjerenja (za brzine mm/s, a za akceleracije mg)
- `session` - redni broj session-a za dano mjerenje

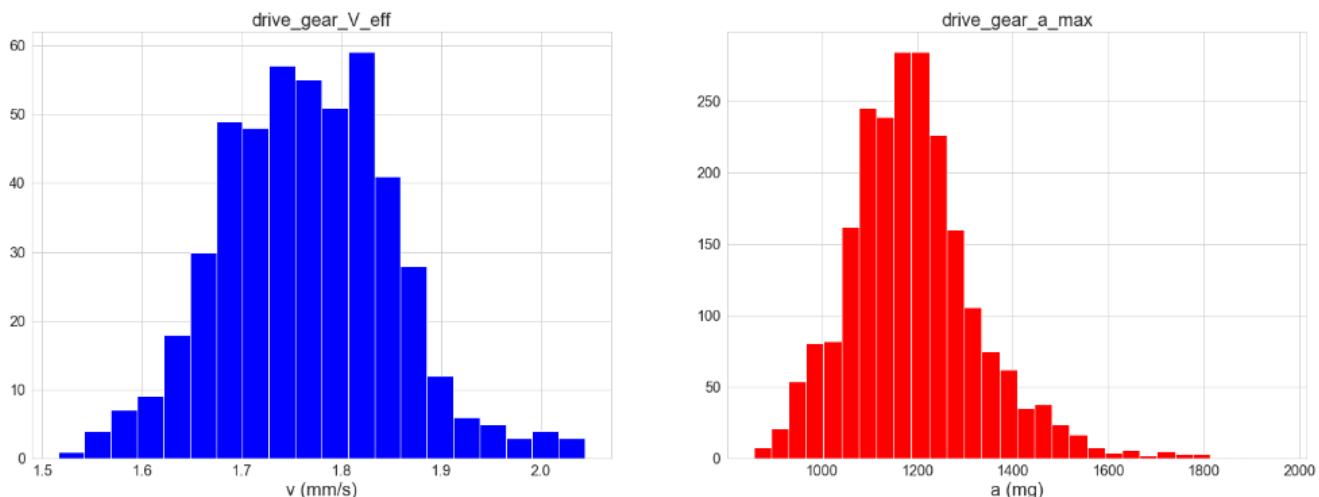


Slika 3: Prikaz udjela podataka po strojevima.

Kako bismo dobili uvid u izgled samih mjerenja, promotrit ćemo ovisnost mjerenih brzina/akceleracija o vremenu. Primjer ovisnosti efektivnih brzina te maksimalne akceleracije o vremenu za jedan odabrani session se nalazi na slici 4. Vidimo da nemamo puno točaka za jedan session mjerenja. Kako bismo dobili bolji uvid u samu distribuciju brzina, prikazali smo raspodjelu brzina u rangi od dvadesetak session-a na slici 5, pod pretpostavkom da se bitne karakteristike (srednja vrijednost, standardna devijacija) ne mijenjaju znatno na skali od nekoliko dana, ako nismo u blizini kvara. Grafovi za ostale senzore nalaze se u bilježnici "Lumen - Introductory Analysis".



Slika 4: Ovisnost efektivne brzine/maksimalne akceleracije o vremenu tijekom jednog sessiona.

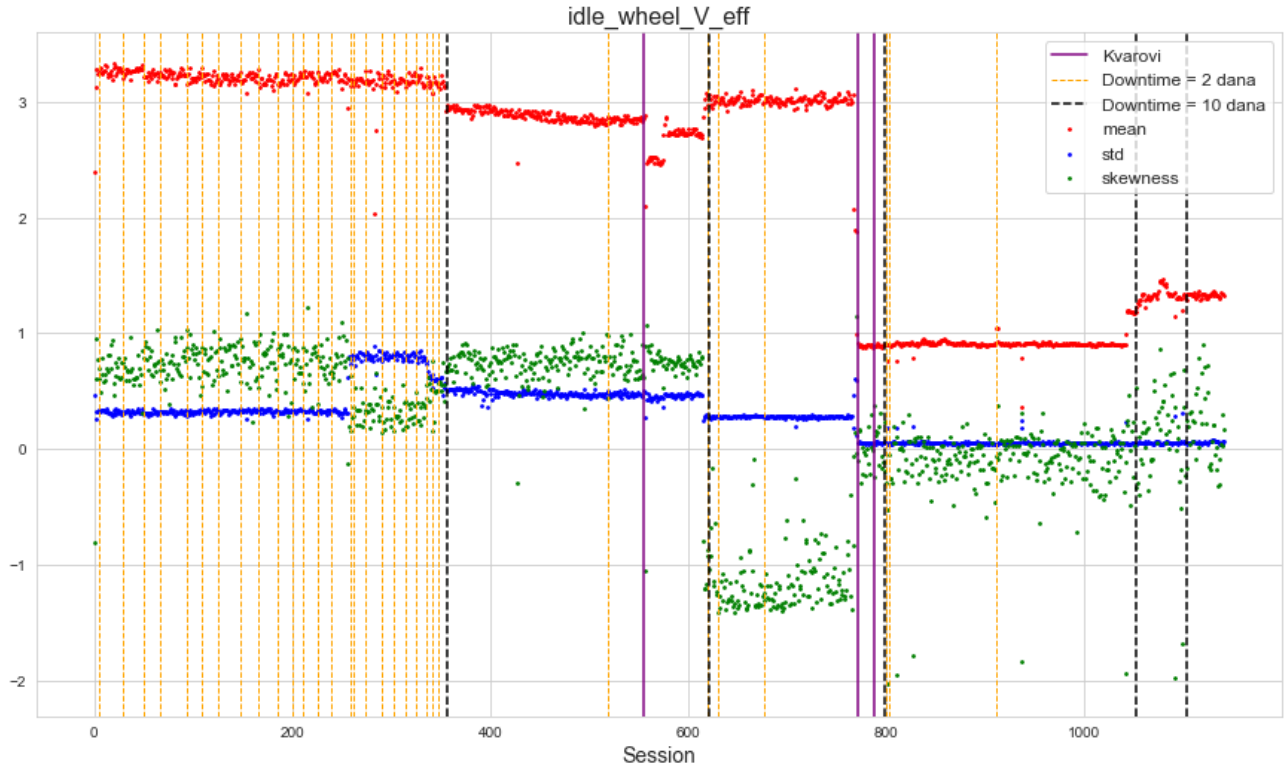


Slika 5: Raspodjele efektivnih brzina (lijevi histogram) i maksimalnih akceleracija (desni histogram) za dva uparena senzora.

Iz distribucije akceleracija na slici 5 uočavamo asimetriju raspodjele, što nam daje motiv za promatranje trećeg centralnog momenta distribucije (*skewness*) kao potencijalni feature.

Nadalje, želimo dobiti uvid u podatke u većem vremenskom intervalu. U tu svrhu, prikazali smo veličine koje opisuju distribuciju brzina na slici 6. Dakle, prikazujemo srednje vrijednosti, standardne devijacije te *skewness* distribucija za svaki session zasebno. Na primjer, svaka crvena točka na ovom scatter plotu označava srednju vrijednost efektivnih brzina iz tog sessiona. Vertikalne ljubičaste linije označavaju session na kojem je došlo do kvara, odnosno kada je započet servis (koji ne traje dulje od par sati, dakle manje od jednog sessiona). Jasno se vidi promjena sva tri parametra nakon servisa. Isto tako, na grafu su prikazani verikalnim narančastim linijama periodi kada nije bilo mjerenja barem 2 dana, a vertikalne crne linije predstavljaju periode kada nije bilo mjerenja barem 10 dana stoga promjene u ponašanju u tim periodima treba uzeti u obzir kao moguću posljedicu velikih razmaka između mjerenja. Vidimo da prvi dio dataset-a nije uporabljiv pošto su mjerenja vrlo nekontinuirana. Valja napomenuti kako je na ovom grafu prikazan i *skewness*, makar je on bezimenzionalna veličina, stoga to treba uzeti u obzir. Naš problem jest predvidjeti kvar prije nego li se dogodi, stoga moramo pronaći značajke koje će što prije i što više promijeniti ponašanje kada se stanje stroja pogorša kako bismo pravovremeno mogli reagirati.

Također, "odokativno" je vidljivo da su parovi senzora korelirani. Te korelacije nismo uzimali u obzir jer se vremenski trenutci ne poklapaju među sensorima, a i senzori u parovima nemaju isti broj mjerenja. Korelacije između ostalih senzora uopće ne promatramo jer mjerenja nisu izvršena istovremeno.



Slika 6: Srednje vrijednosti, standardne devijacije te treći normalizirani centralni moment u ovisnosti o sessionu za uređaj FL01.

3 Modeliranje

Sam proces odabira značajki iz prošlog poglavlja nema previše smisla bez modela uz pomoć kojeg ćemo testirati značajke.

Prvo pitanje kod odabira modela jest želimo li klasifikacijski ili regresijski model. Često se u dostupnoj literaturi [1] kao klasifikacijska metoda navodi Fourierov transform (FT). Ideja je da se signal prebaci iz vremenske u frekvencijsku domenu te se analiziraju vrhovi, zatim usporedimo vrijednosti tih vrhova sa standardiziranim vrijednostima frekvencija koje pripadaju kvarovima. Prednost takve metode je što se mogu klasificirati čak i tipovi kvarova jer očekujemo da će tip kvara uvijek imati peak na sličnoj vrijednosti frekvencije. Međutim tu metodu smo odlučili napustiti zbog prirode našeg dataseta. Naime, ne dobivamo vrijednosti brzina kontinuirano nego u već spomenutim session-ima. Čak i kada bismo promatrali FT na jednom sessionu i dalje imamo problem što točke (primjerice efektivne brzine) imaju vremenski razmak od tri sekunde. Uzmemo li u obzir Nyquistov teorem koji govori da ćemo u spektru moći jedino vidjeti frekvencije barem dva puta manje od frekvencije uzorkovanja (koja je za ovaj primjer ≈ 0.33 Hz) vidimo da nam informacija o spektru naprosto neće biti relevantna. U modelima smo se služili brzinama dok smo akceleracije odbacili jer su modeli s njima bili lošiji, a i iz eksploratorne analize smo vidjeli manju osjetljivost senzora na kvarove. Promotrimo li nadalje neke nadzirane klasifikacijske modele poput logističke regresije imat ćemo fundamentalan problem određivanja target vektora. Naime, da bismo istrenirali takav model moramo mu dati vektor koji označava stanje stroja za svaki pojedini session na trening podacima. Kako mi ne znamo točno stanje stroja osim za sedam dana koji su nam dani odlučili smo napustiti taj pristup te predlažemo sljedeća dva modela.

3.1 Neuronska mreža (LSTM)

Glavnu značajku koju ćemo koristiti u ovom pristupu jest rolling average standardnih devijacija session-a (za odabrani senzor i stroj). Navedimo nekoliko argumenata za korištenje ovakve značajke.

- Standardna devijacija je intuitivna veličina čije značenje i ponašanje jest interpretabilno
- Standardna devijacija je mjera raspona vrijednosti danog uzorka. Mala standardna devijacija (na primjer standardna devijacija brzina) znači da su brzine vibracija uskog spektra, odnosno uređaj približno vibrira stalnom brzinom. Uzak spektar/stabilna brzina trebala bi označavati mjeru ispravnosti stroja. Zamislmo da je uređaj pokvaren; u tom slučaju očekujemo da će stohastički procesi u samoj mehanici stroja utjecati na spektar brzina na način da će postojati više brzina vibracija, što implicira povećanje standardne devijacije.
- Rolling average standardne devijacije je korisna, ali i potencijalno loša veličina. Korisnost ove veličine je da izravna šumove za dovoljno veliki time-window, ali za preveliki time-window nagle promjene u ponašanju standardne devijacije bit će teže opazive, stoga je optimizacija nužna.

Glavni problem ovog problema je što podaci nisu labelirani. Pošto se radi o vremenskoj seriji podataka, razmotrili smo korištenje RNN (Recurrent neural network). Konkretno, koristit ćemo LSTM (long short-term memory) mrežu, pošto obična RNN mreža nema long-term pamćenje. Vremenska jedinica će u našem slučaju biti session. Koristeći NN mrežu, pokušat ćemo predvidjeti (forecast-at) **periods** broj session-a u budućnost na temelju zadnjih **length** sessiona. Verbalno izrečen algoritam za detekciju anomalija jest sljedeći:

- Istrenirati NN na temelju **time_window** session-a te na temelju zadnjih **length** session-a predvidjeti vrijednost značajke idućeg session-a (koji će se tek dogoditi)
- Kada se prikupe mjerenja za predviđeni session, izračunati odabranu značajku
- Ako je razlika vrijednosti stvarnog mjerenja i predviđene vrijednosti veća od 3 standardne devijacije¹ značajke u periodu **time_window**, zabilježiti session kao anomalni

Inicijalno smo napravili mrežu sa samo jednim LSTM layerom sa 100 neurona (i jednim output layerom), no ispostavilo se da dodavanje dodatnog **dense** layera sa 30 neurona dobivamo bolje forecast rezultate. U bilježnici "Lumen - RNN" koja se koristi za izvršavanje algoritma nalazi se mnogo hiperparametara te ćemo prodiskutirati najbitnije:

- **rolling_steps** - veličina rolling window-a za izračun rolling značajki. Odabrana vrijednost za većinu izračuna jest 5 jer se pokazala kao efikasna za optimizaciju minimizacije false-positive anomalija, bez dodatka false-negative rezultata. Kada bi senzori više oscilirali između session-a, veličinu bi trebalo povećati.
- **n_features** - broj značajki na kojima radimo predviđanje (forecast). Pošto mi predviđamo na temelju jedne značajke (rolling average standardnih devijacija session-a), ovaj parametar je postavljen na vrijednost 1.
- **time_window** - veličina window-a koji se gleda kao "trenutni skup podataka" (training set). Za danu vrijednost n , forecasting model će se trenirati na zadnjih n podataka. Trenutno je postavljen na vrijednost $3 \times \text{rolling_steps}$. Ova veličina je odabrana da imamo dovoljan skup za trening kako bi neuralna mreža uhvatila trend, no predviđanje našeg algoritma ne ovisi značajno o samom izboru ovog parametra, dok god nije ekstremno malen.
- **epochs** - broj epoha, odnosno broj koliko će se puta trenirati neuralna mreža. Odabran broj je 10. Odabran je tako što smo promatrali loss za svaku epohu te se broj 10 činio plauzibilan. Da imamo test set, mogli bismo implementirati early stopping mehanizam, no pošto u stvarnosti nemamo podatke iz budućnosti (test set) ne možemo tako optimizirati broj epoha. Manji broj epoha doveo bi do lošeg fit-a, a samim time i lošeg forecast-a.
- **periods** - broj koraka u vremenu koje želimo predvidjeti unaprijed. Odabrani broj je 1, jer želimo predvidjeti vrijednost samo za idući session. Kada bismo predviđali više koraka unaprijed, imali bismo robusniji model na outlieri i false-positive, no time gubimo vrijeme jer za evaluaciju algoritma moramo pričekati **periods** session-a, a samim time se kasnije reagira na potencijalni kvar.
- **length** - na temelju zadnjih **length** session-a istreniranih na **time_window** točaka forecast-amo **periods** session-a u budućnost. Odabran broj je 4, makar algoritam ne ovisi jako o izboru ovog parametra, dok god smo u blizini veličine **rolling_steps**.

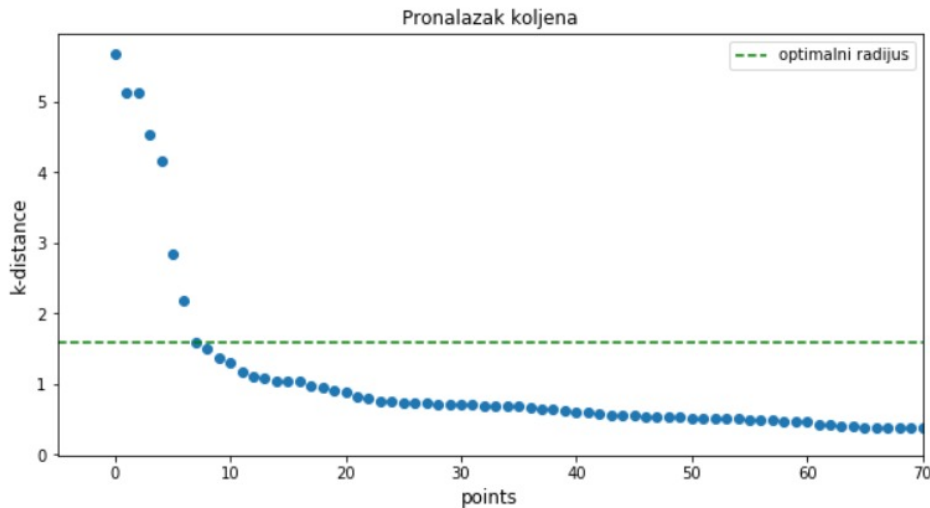
U bilježnici "Lumen - RNN" simuliramo tok vremena (odnosno dostupnost podataka), no to ćemo detaljnije objasniti u poglavlju *Evaluacija*. Inicijalno smo koristili **MinMaxScaler** funkciju za normalizaciju podataka, no pošto su podaci većinom između 0 i 1, sam proces skaliranja podataka nam nije mijenjao predikcije, već je mrvicu usporio sveukupni algoritam te smo ga iz tog razloga odlučili isključiti. Razmišljali smo i o uklanjanju outlier-a, ali jedina prednost toga za ovu metodu jest smanjivanje false-positive rezultata. Glavna mana ovog pristupa je to što je moguće da je sam outlier anomalija, odnosno značajka kvara, što bi dovelo do false-negative rezultata, a to za ovakvu vrstu problema nije poželjno.

3.2 Nenadzirano grupiranje (DBSCAN)

Motivacija ovog pristupa je da uspijemo odrediti sessione prije kvara na način da ih naš algoritam uspije grupirati u prostoru značajki. Kandidate za značajke ćemo ekstrahirati iz modula **tsfresh** [6]. Taj modul nudi preko sedamsto značajki pa sve one koje imaju nesmislene vrijednosti ili jako koreliraju izbacimo te nastavljamo proces.

¹Treba obratiti pažnju na kontekst riječi "standardna devijacija" jer se u prošlom odjeljku javljala u nekoliko različitih dijelova. Na primjer, za značajku koju ćemo mi promatrati (rolling average standardne devijacije raspodjele brzina session-a), u trećem koraku algoritma standardna devijacija se odnosi na standardnu devijaciju raspodjele rolling average-a standardne devijacije.

Nakon što smo pripremili značajke slijedi grupiranje koje se sastoji od filter i wrapper dijelova. Prvo kreiramo matricu sličnosti te na temelju nje svakoj značajki pridajemo **Laplacian score**(LS) [4]. Sortiramo značajke po toj mjeri na način da prva značajka ima najmanji LS što ujedno znači i da je najrelevantnija. Značajke nakon grupiranja jednu po jednu iterativno evaluiramo pomoću WNCH ('Weighted Normalized Calinski Harabasz score'). Ukoliko značajka ostvari bolji WNCH score nego prethodni skup, tu značajku spremamo u skup dobrih značajki, a ako ostvari lošiji WNCH score ju izbacimo. Algoritam grupiranja sessiona na temelju kojeg pridajemo WNCH score je DBSCAN. Njega smo odabrali jer ne zahtjeva broj grupa kao hiperparametar što čini grupiranje generalnijim u strukturama o kojima prethodno nemamo znanja. Bitan hiperparametar DBSCAN-a je **eps**, što je ugrubo radijus koji određuje koliko blizu moraju biti točke u prostoru značajki da tvore grupu. Njega smo nalazili heuristički pomoću metode nalaženja koljena (Slika 7).



Slika 7: Koljeno funkcije udaljenosti k-tog susjeda te pripadajuća vrijednost **eps**(isprekidana linija)

Drugi hiperparametar je **min_samples** kojeg se otprilike može interpretirati kao minimalan broj točaka koji se treba nalaziti unutar radijusa **eps** da bi tvorile grupu. Kao rezultat cijele metode dobivamo labele koje nam govore koji session-i pripadaju kojoj grupi te skup relevantnih značajki koje su pridonjele takvom grupiranju. WNCH metodu možemo opisati i pomoću pseudokoda kao na slici 8, a detaljniji opis rada ovog pristupa može se pronaći u tehničkoj dokumentaciji.

Algorithm 1. Pseudo-code for LS-WNCH-SR method.

Input: X Dataset with m instances and n features
 k_{LS} number of neighbors for the graph construction
 c number of clusters
Output: S_{best} the best feature subset.
// Initializing variables
1: $indRank \leftarrow \emptyset$
2: $\gamma_{best} \leftarrow -\infty$
3: $S_0 \leftarrow \emptyset$
4: Calculate the Laplacian Score L_r for each feature f_r with $r=1$ to n .
5: Sort the L_r values in descending order, and assign the indexes of the features associated to each L_r into the $indRank$ array.
6: **for** $i=1$ to n **do**
7: $S_0 \leftarrow S_0 \cup indRank[i]$
8: Run a clustering algorithm over X_{S_0}
9: $\gamma \leftarrow WNCH(S_0)$ // Clusters evaluation
10: **if** $\gamma > \gamma_{best}$ **then**
11: $\gamma_{best} \leftarrow \gamma$
12: $S_{best} \leftarrow S_0$
13: **end if**
14: **end for**
15: **return** S_{best}

Slika 8: Algoritam za odabir značajki DBSCAN-om

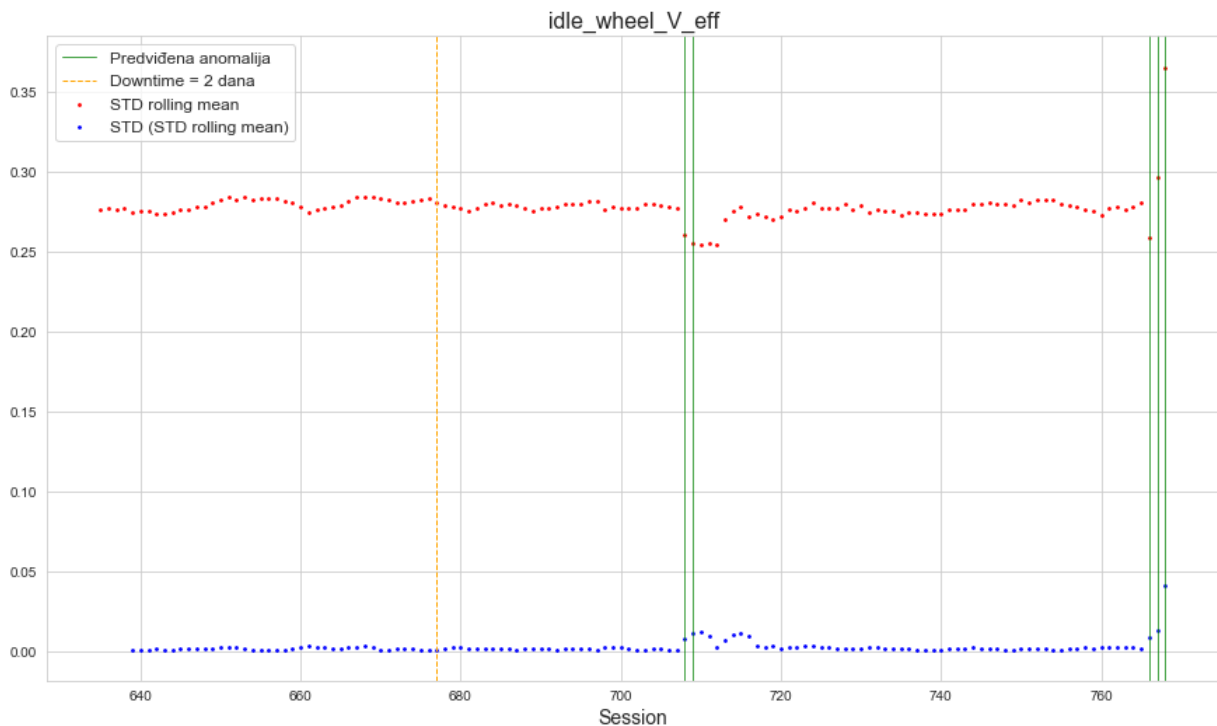
4 Evaluacija

4.1 Evaluacija LSTM modela

U slučaju modela s neuronskom mrežom bitno je ispravno predvidjeti kvarove. Ono što mi možemo detektirati jesu anomalije, a pretpostavka je da će se svaki *nagli* kvar pokazati kao anomalija koju predviđamo.

Objasnimo detaljnije kako pomoću bilježnice "Lumen - RNN" možemo evaluirati model. U bilježnici se mogu birati gore navedeni hiperparametri, a bira se i stroj, senzor te interval datuma koje promatramo. `for` petljom simuliramo "tok vremena" tako što svakom iteracijom dodamo novi podatak iz novog session-a. Na primjer, ako je petlja na i -tom koraku, treniramo LSTM mrežu na zadnjih `time.window` koraka te predviđamo vrijednost značajke za $i + 1$ korak te uspoređujemo poklapa li se to predviđanje sa stvarnim podatkom koji do tada nije bio dostupan (imitiramo da je došao novi session mjerenja te sada uspoređujemo forecast sa stvarnom vrijednošću) te primijenimo gore navedeni algoritam za detekciju anomalije. Isto tako nakon treninga algoritma imamo mogućnost uključivanja `false_positive_filter`-a². Naime, pretpostavka je da će svaki nagli kvar imati porast u standardnoj devijaciji session-a (a samim time i u rolling average-u standardnih devijacija) stoga uključivanjem filtera izbacujemo potencijalne anomalije čiji forecast je 3 standardne devijacije *iznad* vrijednosti koju izmjerimo. Drugim riječima, uključivanjem filtera slučajeve kada standardna devijacija naglo padne ne smatramo anomalijom. Naravno, time možemo (i smanjimo) velik broj false positive slučajeva, no riskiramo prekasno detektiranje anomalija i povećavamo mogućnost false negative slučajeva.

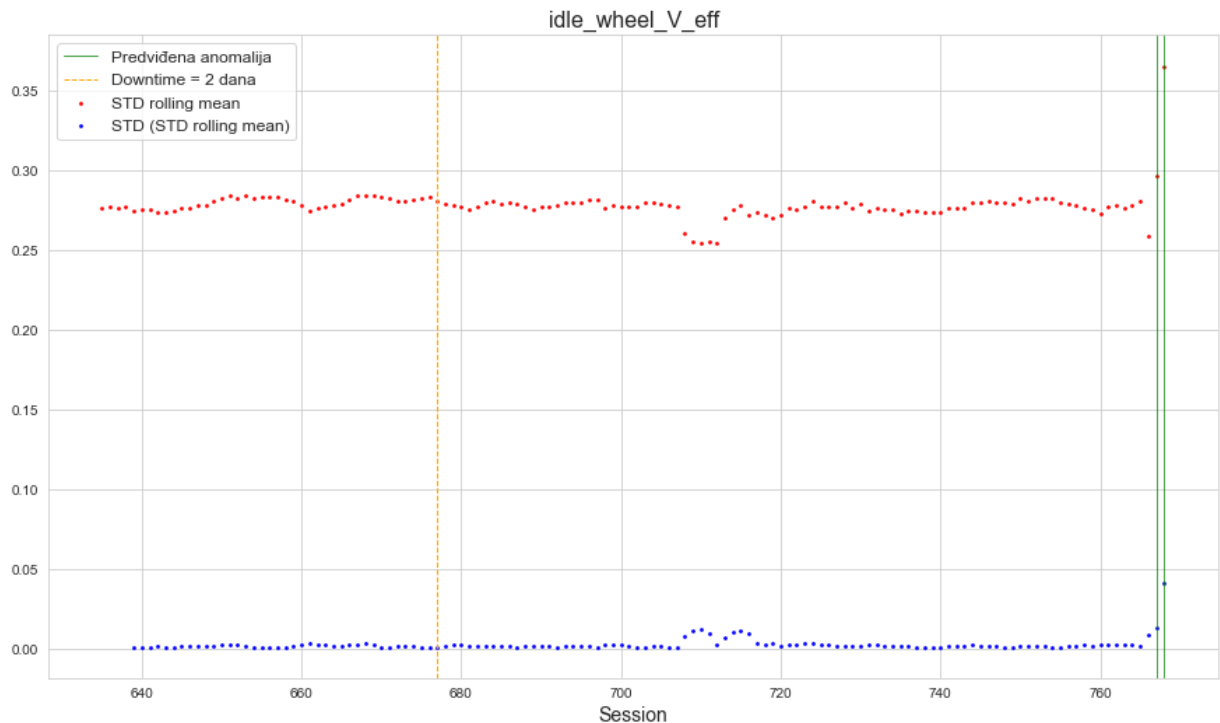
Pogledajmo rezultat (Slika 9) bilježnice za uređaj FL01 te senzor `idle_wheel_V_eff` u razdoblju od početka 2019. godine pa sve do dana *prije* kvara (uključujući i sam dan prije kvara).



Slika 9: Predviđene anomalije za uređaj FL01 te `false_positive_filter=0` u periodu od 2019-01-01 do 2019-02-08

Zelene linije su session-i na kojima je naš algoritam predvidio kvar (dakle na tom session-u se forecast nije poklapao sa izmjerenom veličinom iz podataka). Prva zelena linija se pojavljuje 3 session-a prije samog kvara, što znači da smo kvar uspjeli predvidjeti barem 1 dan unaprijed. Iscrtkana narančasta linija je session prije kojeg nije bilo mjerenja barem 2 dana, što nam može služiti za opravdavanje nekih false positive rezultata (što bi bio rezultat nekontinuiranosti mjerenja, a ne lošeg forecasting-a). Prve dvije linije se mogu protumačiti kao false positive rezultati. Dakle, u 5 tjedana imali smo 2 false positive rezultata (jedan za drugim su, stoga se može gledati kao jedan serijal false positive rezultata). Promotrimo što se dogodi ako uključimo `false_positive_filter` (slika 10).

²Spremaju se obje vrste anomalija, filter se može naknadno uključiti za vizualizaciju/dokumentaciju željene vrste (filter/no filter)



Slika 10: Predviđene anomalije za uređaj FL01 te `false_positive_filter=1` u periodu od 2019-01-01 do 2019-02-08.

Confusion matrix 9.		
	F(measure)	W(measure)
F(forecast)	3	2
W(forecast)	0	125

Confusion matrix 10.		
	F(measure)	W(measure)
F(forecast)	2	0
W(forecast)	1	127

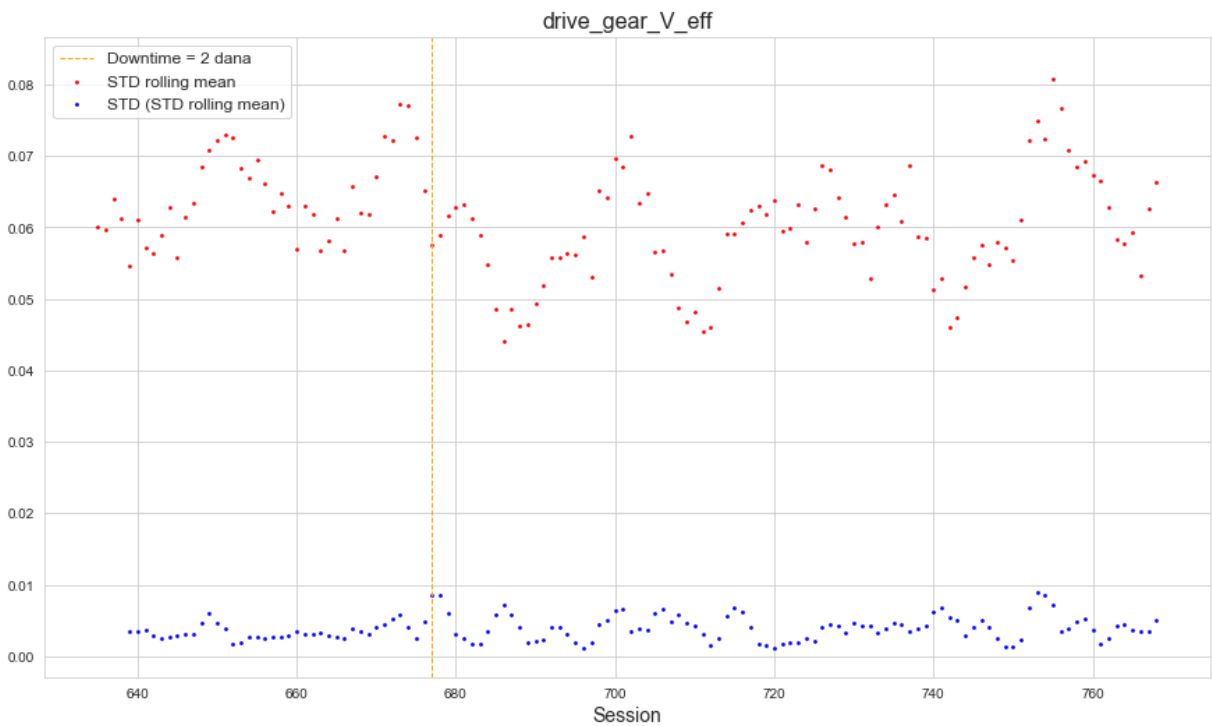
U confusion matrix prikazima koristit ćemo iduće oznake:

- F(forecast) - kvar predviđen algoritmom (fault forecast)
- W(forecast) - ispravan rad uređaja predviđen algoritmom (working forecast)
- F(measure) - potvrđen kvar na uređaju (measured fault)
- W(measure) - nije opažen kvar na uređaju (measured working)

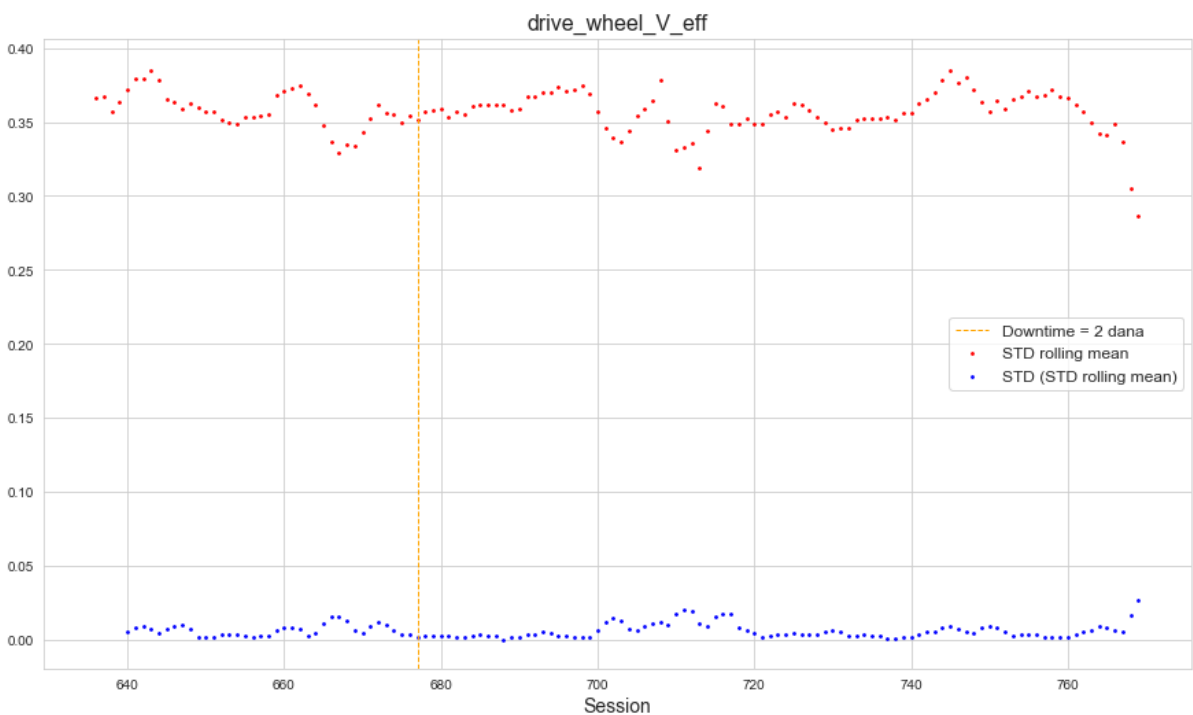
Primjetimo kako su svi false-positive rezultati izbačeni. Isto tako, vidimo kako smo "platili" tako što smo jedan session kasnije predvidjeli kvar (nije uočen prvi pad kao anomalija) te smo to označili kao false negative. Unaprijed nije jasno što bismo točno trebali označiti kao false positive, a što kao false negative, stoga prije daljnje rasprave o evaulaciji bitno je uopće definirati što u ovakvom modelu znače ti pojmovi. Odgovor na to pitanje nije jednostavan pošto podaci nisu labelirani (nije nam poznato točno stanje stroja u svakom trenutku). Intuitivno, iz gornjih grafova zaključilo bi se da su zelene linije prije kvarova true positive, pošto je ponašanje značajke drugačije, a i u neposrednoj smo blizini kvara. Pitanje je, ako bi naš algoritam javio prve dvije anomalije sa slike 8 te bi stručnjak otišao provjeriti uređaj, bi li uočio kvar na `idle_wheel` dijelu? Odgovor na to pitanje nam je nepoznat jer nam tadašnje stanje uređaja nije dostupno. Koristit ćemo "odokativnu" procjenu, odnosno ako je predviđena anomalija 10 dana (ekvivalent otprilike 30 session-a) prije kvara, tretirat ćemo to kao false positive. Napomenimo još jednom da x -os svih grafova (grafove u kojima predviđamo kvar) završava na dan prije kvara (uključujući i sam dan prije kvara).

Uspješno smo predvidjeli kvar `idle_wheel` na uređaju FL01, no sada je pitanje kako naš algoritam reagira na ostale senzore u ovom periodu? Na slikama 11 i 12 vidimo predviđene anomalije za senzore `drive_gear_V_eff` `drive_wheel_V_eff`.

Confusion matrix 11 i 12.		
	F(measure)	W(measure)
F(forecast)	0	0
W(forecast)	0	137



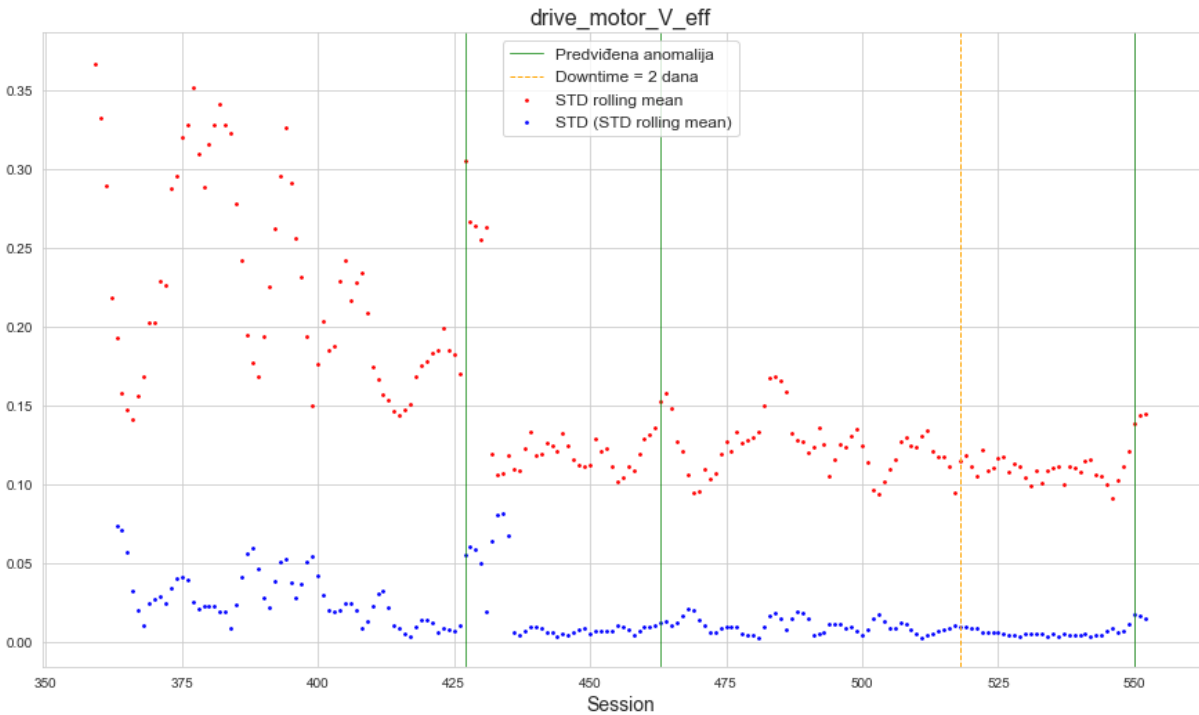
Slika 11: Predviđene anomalije za uređaj FL01 te `false_positive_filter=0` u periodu od 2019-01-01 do 2019-02-08.



Slika 12: Predviđene anomalije za uređaj FL01 te `false_positive_filter=0` u periodu od 2019-01-01 do 2019-02-08.

Na ovim senzorima nisu primjećene nikakve anomalije, što je i očekivano, makar je moguće i da ostali senzori osjete kvar drugog dijela uređaja, samo manjim intenzitetom. Ako je to i slučaj, naš algoritam to nije prepoznao kao anomaliju, što nam daje nadu da se ovakvom metodom može predvidjeti i *vrsta* kvara. Samo je nekolicina senzora osjetila ovaj kvar, no većina nije imala zabilježenu niti jednu anomaliju u ovom poeriodu.

Promotrimo možemo li predvidjeti kvar u 11. mjesecu na uređaju FL01. Pošto ovaj kvar ne pripada niti jednom senzoru specifično (ne znamo točnu prirodu kvara), probali smo predvidjeti kvar na nekoliko različitih senzora te analizirati predviđene anomalije. Promotrimo rezultate za senzore `drive_gear.V_eff`, `drive_wheel.V_eff` te `idle_wheel.V_eff` na slikama 13, 14 i 15.



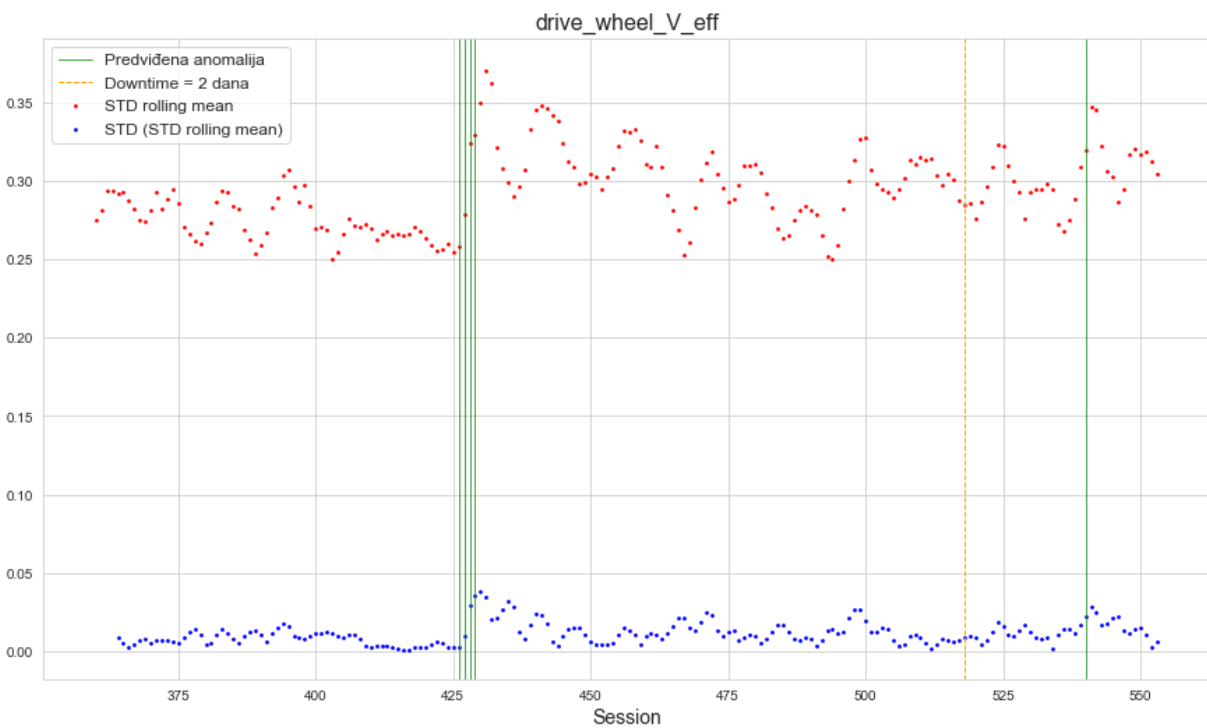
Slika 13: Predviđene anomalije za uređaj FL01 te `false_positive_filter=1` u periodu od 2018-09-01 do 2018-11-13.

Confusion matrix 13.

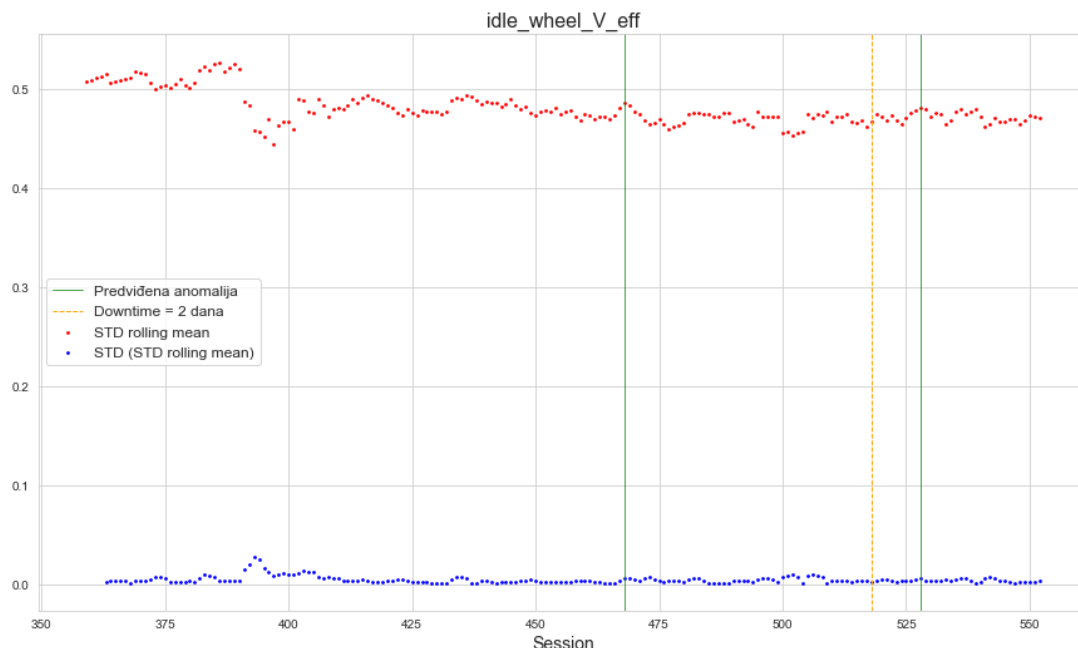
	F(measure)	W(measure)
F(forecast)	1	2
W(forecast)	0	184

Confusion matrix 14.

	F(measure)	W(measure)
F(forecast)	1	4
W(forecast)	0	182



Slika 14: Predviđene anomalije za uređaj FL01 te `false_positive_filter=1` u periodu od 2018-09-01 do 2018-11-13.



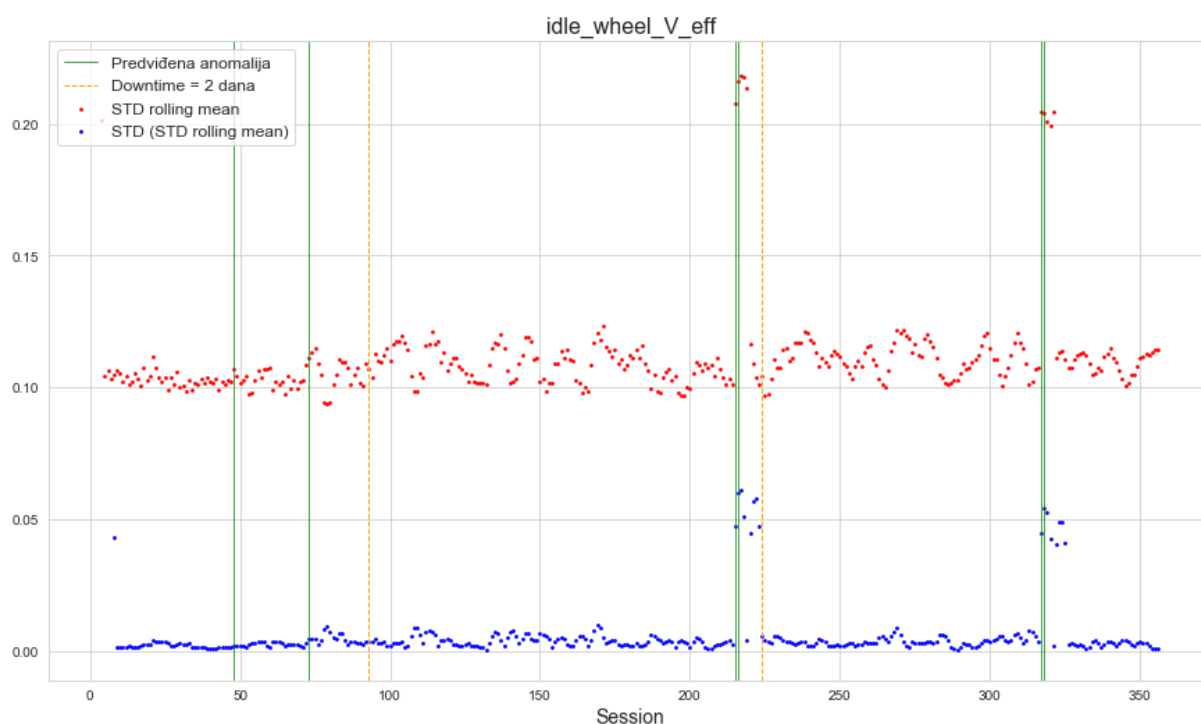
Slika 15: Predviđene anomalije za uređaj FL01 te `false_positive_filter=1` u periodu od 2018-09-01 do 2018-11-13.

Confusion matrix 15.

	F(measure)	W(measure)
F(forecast)	1	1
W(forecast)	0	185

Vidimo malen broj false positive rezultata, a najbitnije, predviđamo kvar na uređaju na svim senzorima. Ostali senzori su pokazali sličan trend, odnosno osjetili su 10-ak session-a prije kvara anomaliju uz nekoliko false positive rezultata u ovom periodu.

Treba provjeriti kako naš algoritam radi na ispravnom uređaju. Za primjer uzimamo uređaj FL05 te senzor `idle_wheel_V_eff`, na kojem nisu zabilježeni kvarovi. Na slici 16. vidimo sva mjerenja (svi sessioni) dostupna za taj uređaj te predviđene anomalije. Imamo 6 false positive rezultata u otprilike 350 session-a, što je odličan rezultat (to odgovara skoro 4 mjeseca mjerenja). Ostali senzori za ovaj stroj imaju otprilike jednak broj false positive rezultata, što može zvučati kao zabrinjavajuć ukupni broj, ali s obzirom da svi senzori daju false positive istovremeno, vjerujemo kako je ovdje bila riječ o globalnoj anomaliji na stroju. Na slici 16 je vidljivo zašto ih je algoritam označio kao takve (nagli skok u rolling average-u). Nužno je naglasiti da anomalija ne znači nužno kvar, ali nagli kvar bi trebao uzrokovati anomaliju.



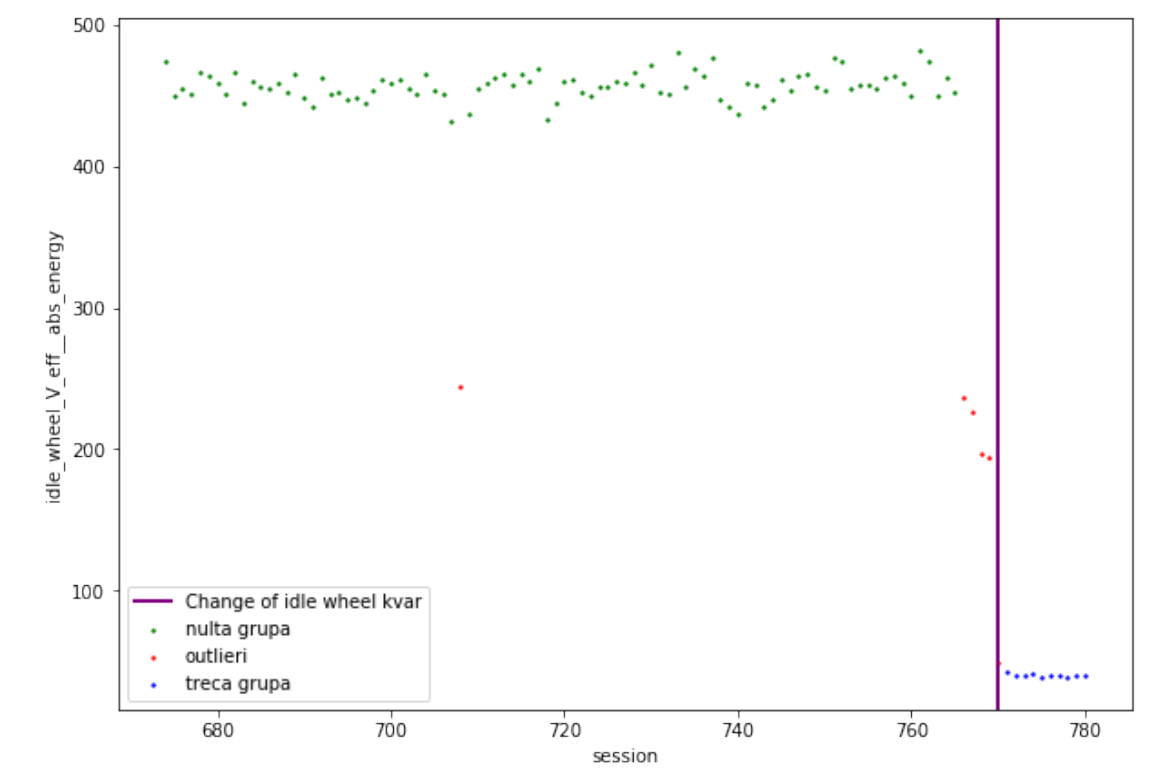
Slika 16: Predviđene anomalije za uređaj FL05 te `false_positive_filter=0`

Confusion matrix 16.		
	F(measure)	W(measure)
F(forecast)	1	2
W(forecast)	0	351

Naš model na ostalim uređajima (u ispravnim periodima) pokazuje vrlo slično ponašanje: malen broj false positive rezultata (otprilike 1 u 30 dana), a većina false positive rezultata zaista izgleda kao neki oblik anomalije kada promotrimo ponašanje značajki za takav period.

4.2 Evaluacija DBSCAN modela

Pristup nenadziranog grupiranja može se evaluirati matricom konfuzije pa ćemo sada promotriti jedan primjer kvara koji se nalazi u bilježnici **Lumen** - DBSCAN. Radi se o kvaru na stroju **FL01** pod oznakom **Change of Idle wheel**. Primijetimo (Slika 17) da nam je algoritam ponudio 3 grupe podataka. Prvu grupu možemo interpretirat kao grupa u kojoj stroj radi ispravno, zatim grupa session-a gdje bismo trebali napraviti servis te posljednja grupa nakon samog servisa. Prikazana je i ljubičasta vertikalna linija na dan kada se dogodio servis koja nam pomaže interpretirati dobivene grupe podataka. Ovdje je još bitno naglasiti da algoritam najčešće izabere više od jedne značajke za grupiranje, no radi lakše vizualizacije ovdje smo prikazali session-e samo u prostoru jedne značajke koju je algoritam izabrao za ovaj servis (značajku apsolutne energije).



Slika 17: Implementacija nenadziranog grupiranja pomoću DBSCAN metode na kvaru **Change of Idle wheel**

Confusion matrix prikazana je sljedećom tablicom

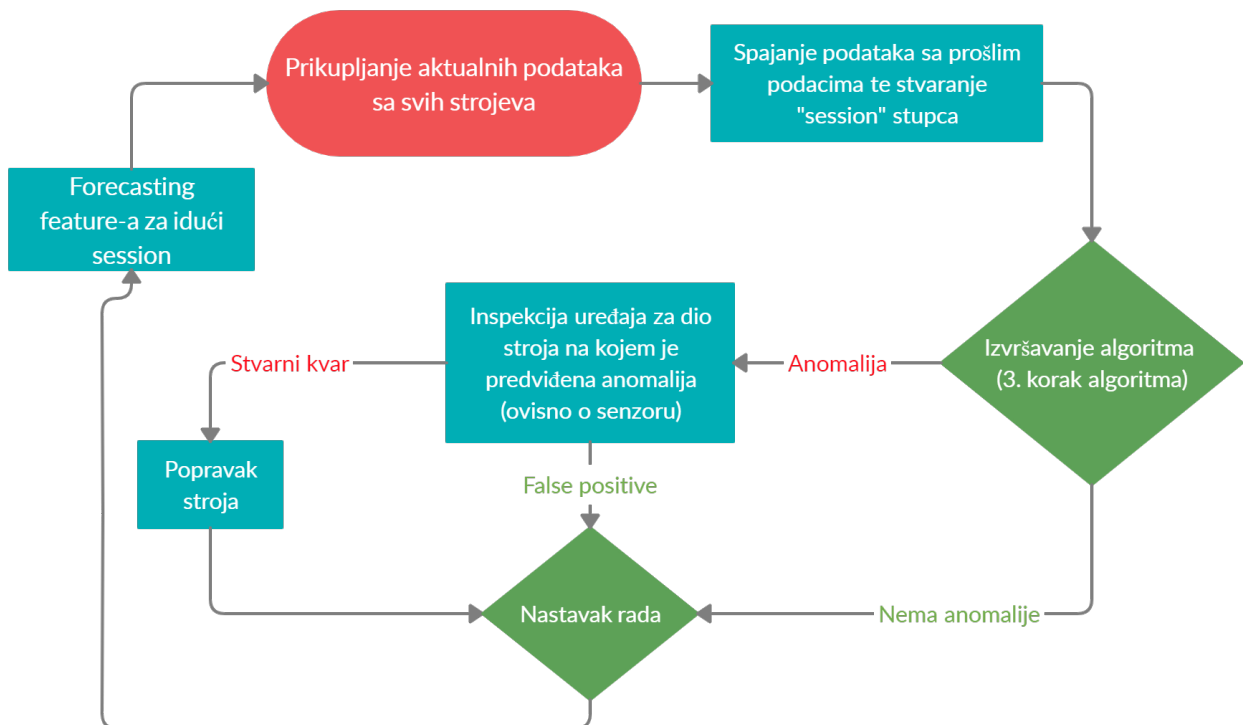
	F(measure)	W(measure)
F(forecast)	4	1
W(forecast)	0	90

Opet se postavlja pitanje kako definirati false negative predviđanja u podacima u kojima nemamo oznake stanja stroja. Smatramo da je jedan dan dovoljno vremena za reakciju i sprečavanje težih kvarova stroja, stoga ćemo za false negative označiti one točke koje se nalaze u periodu jednog dana prije kvara (odnosno tri sessiona), a naš algoritam ih nije označio kao outlieri. Isto tako je važno reći da u tablicu nisu ulazile točke nakon servisa jer nas zapravo te točke ne zanimaju ako nam je cilj predvidjeti kvar. One su samo ovdje kao potvrda da algoritam radi smislene podjele. Također je bitno naglasiti da i ostali senzori primijete kvar u ovoj metodi što je ujedno i prednost i mana. Prednost u kontekstu da smo sigurniji da je kvar blizu, a mana jer nam miče potencijalni uvid u konkretniju klasifikaciju samog tipa kvara. Važno je naglasiti da algoritam za stroj **FL04** te primjerice senzor **lifting_gear_V_eff** javlja da nije našao više od jedne grupe što i očekujemo budući da na tom stroju nisu provedeni servisi.

Kod ostalih kvarova algoritam nije uspio uspješno grupirati sessione prije servisa, a u nastavku teksta dat ćemo neke argumente zašto mislimo da je tome tako.

5 Deployment

Temeljna zadaća modela je detektirati anomaliju te upozoriti osoblje zaduženo za održavanje uređaja. Shematski prikaz implementacija same LSTM metode nalazi se na slici 18.



Slika 18: Plan implementacije LSTM algoritma za primjenu u stvarnosti

Za deployment može se koristiti i sama bilježnica "Lumen - RNN", makar je poželjna jednostavna modifikacija koja ne uzima u obzir cijeli dataset (što smo mi radili radi simulacije dostupnosti podataka), već samo željeni period (`time_window + rolling_steps` session-a prije trenutnog) što znatno ubrzava cijeli proces. Kada bi algoritam označio trenutni `session` kao anomalni (objašnjeno u poglavlju 3.1), stručnjaci bi trebali izvršiti inspekciju uređaja i provjeriti kvar. Ako je stvarno u pitanju kvar, obustavlja se rad uređaja te se uređaj servisira. U finalnom koraku, algoritam predviđa vrijednost značajke za idući session te se proces ponavlja.

Isto tako, bitna je optimizacija hiperparametara kako bi se smanjio udio false positive rezultata. Idealno, optimizacija bi se trebala izvršiti za svaki uređaj i svaki senzor u trenu kada imamo dovoljan broj mjerenja (u stvarnosti ne bi trebalo više od 3-4 tjedna podataka). Trenutno je naš algoritam u fazi ručne optimizacije hiperparametara te kao budući cilj želimo implementirati automatiziranu optimizaciju radi jednostavnijeg korištenja i boljih rezultata.

6 Zaključak

Cilj ovog projekta bio je da predložimo, realiziramo te implementiramo metodu predictive maintenance-a za predviđanje kvara na temelju analize vibracija. Za to smo koristili dva različita pristupa: LSTM neuralnu mrežu te nenadzirano grupiranje. Prodiskutirajmo rezultate ovih dviju metoda.

LSTM model je uspješno predvidio sve kvarove koje kategoriziramo kao *nagle*, ali nismo mogli uspješno evaluirati naš model na stroju FL03 jer nema mjerenja neposredno prije kvara (kraj 3. mjeseca) te je odrađen servis "grinding of the rail" što utječe na naša predviđanja pošto interferira sa samim potencijalnim signalom kvara drive motora. Glavni cilj nam i je predvidjeti takve kvarove (*nagle* kvarove), jer popravci poput "grinding of the rail" su greške koje se redovno servisiraju, što zaključujemo iz činjenice da su 3 uređaja u istom periodu imala takav servis (u travnju 2019. godine). LSTM model ima malen udio false positive rezultata, što je isto poželjno jer smanjuje resurse i potrebu za ljudskom intervencijom. Sam postotak false positive rezultata može se dodatno smanjiti uključivanjem `false_positive_filter`, no hoće li taj filter biti uključen je odabir samog klijenta ovisno o željenom balansu ljudske intervencije i mogućih false negative anomalija (odnosno manjak sposobnosti detektiranja anomalija) ili late detection (primjer na slikama 9 i 10). Radi konzistentnosti, svi gornji rezultati su dobiveni za isti izbor hiperparametara jer kada bismo adaptirali hiperparametre našim podacima bismo zapravo "varali" jer imamo pregled svih podataka (najbolja optimizacija svakog senzora i svakog stroja nije za isti set hiperparametara), a u stvarnosti imamo podatke samo do neke vremenske točke. Ako dugo skupljamo podatke, onda je optimizacija hiperparametara poželjna radi daljnjeg smanjivanja false positive anomalija i poboljšanja prediktivne sposobnosti same mreže. Na primjer,

stroj FL04 izbacuje nešto više false positive rezultata, a samom inspekcijom (kroz korištenje bilježnice "Lumen - RNN") vidimo kako podaci više osciliraju, što nam daje naznaku da bismo trebali promijeniti parametar `rolling_window` (na veću vrijednost od trenutne). Potencijalna mana ovog pristupa jest nemogućnost detekcije bliskih kvarova, odnosno vjerojatno nećemo biti u mogućnosti detektirati novi kvar dok je uređaj u periodu *relaksacije* od prošlog servisa (period stabilizacije uređaja), no smatramo da su takvi slučajevi rijetki.

DBSCAN metoda polučila je izvrstan rezultat kod detektiranja promjene ponašanja te je počela upozoravati više od jednog dana prije samog servisa kao što smo mogli vidjeti na slici 17. Kao dodatnu potvrdu da algoritam "zna" razlikovati različita stanja zdravlja stroja uspio je grupirati i stanje stroja nakon servisa što se može interpretirati kao efektivno "novi" stroj jer smo dio prošlog zamijenili, dakle algoritam očito razlikuje različite modove rada ovog stroja. Isto tako, metoda nam daje smislene rezultate kada na uređaju nisu bile prijavljene ljudske intervencije, na primjer na uređaju FL04 te senzoru `lifting_gear_V_eff` tako što ne nade više od jedne grupe podataka. Međutim postoje servisi koje ova metoda nije uspjela uspješno detektirati. Počnimo sa **Correction of skewed movement of the machine on the rail**. Detaljnijom analizom može se utvrditi da nijedna značajka (odnosno skup značajki) iz modula `tsfresh` nije uspjela prepoznati sessione prije kvara kao outliers. Bilo bi zanimljivo u budućem radu probat generirat još neke specifične značajke prilagođene našem datasetu. Sljedeći servis kojeg bismo htjeli prokomentirati je **Change of bearings of drive motor** na stroju FL03. Postoje dva problema sa predviđanjem ovog servisa za koja mislim da su ključna te ih dijele obje metode i opisani su u prošlom paragrafu kao i zaključci o servisima tipa **Grinding of the rail**.

Konačno, zaključili bismo da je LSTM model robusniji na ovom datasetu te je bolje određivao

Literatura

- [1] Cornelius Scheffer, Paresh Girdhar *Practical Machinery Vibration Analysis and Predictive Maintenance* 1st Edition ISBN: 9780750662758 2004
- [2] T.H. Guo, J. Nurre *Sensor failure detection and recovery by neural networks* 1991
- [3] S.Solorio-Fernández, et al. *A new hybrid filter-wrapper feature selection method for clustering based on ranking*, Neurocomputing, 2016, <http://dx.doi.org/10.1016/j.neucom.2016.07.026>
- [4] Xiaofei He, et al., *Laplacian Score for Feature Selection*, Department of Computer Science, University of Illinois at Urbana-Champaign, 2006
- [5] M. Ester, et al., *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, Institute for Computer Science, University of Munich, 1996
- [6] <https://tsfresh.readthedocs.io/en/latest/index.html>
- [7] <https://github.com/gm-spacagna/deep-ttf/>