

Predikcija korektivnih mjera u stabilizaciji sustava proizvodnog procesa

Šverko, Mladen

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:385426>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-09-03**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

MLADEN ŠVERKO

**PREDIKCIJA KOREKTIVNIH MJERA U STABILIZACIJI SUSTAVA
PROIZVODNOG PROCESA**

Diplomski rad

Pula, lipanj, 2021.

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

MLADEN ŠVERKO

**PREDIKCIJA KOREKTIVNIH MJERA U STABILIZACIJI SUSTAVA
PROIZVODNOG PROCESA**

Diplomski rad

JMBAG: 0016076929, izvanredni student

Studijski smjer: Informatika

Predmet: Izrada informatičkih projekata

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informatičke i komunikacijske znanosti

Znanstvena grana: Informatički sustavi i informatologija

Mentor: doc. dr. sc. Nikola Tanković

Komentor: izv. prof. dr. sc. Darko Etinger

Pula, lipanj, 2021.



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani **Mladen Šverko**, kandidat za magistra informatike ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

U Puli, 6. lipnja, 2021 godine



IZJAVA o korištenju autorskog djela

Ja, **Mladen Šverko** dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „Predikcija korektivnih mjera u stabilizaciji sustava proizvodnog procesa“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, 6. lipnja, 2021 godine

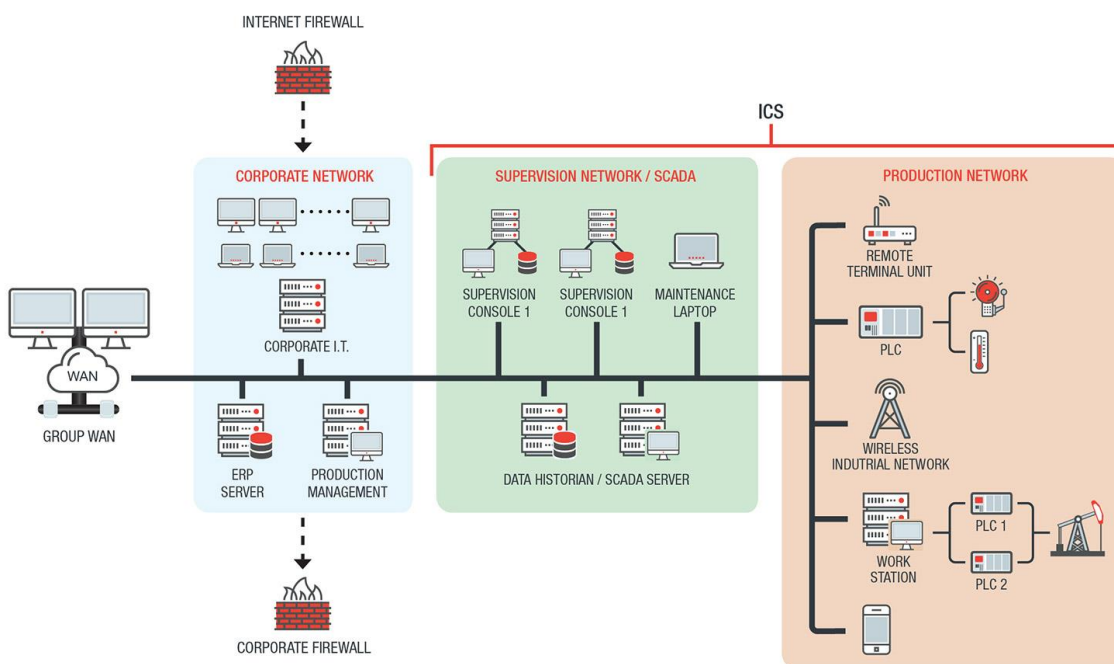
Potpis

Sadržaj

Sadržaj	1
1. Uvod.....	2
2. Definicija problema i uvodna razmatranja	5
2.1 Operateri – ljudski čimbenik.....	5
2.2 HMI – kompleksnost sučelja	6
3. Prethodna istraživanja.....	7
3.1 Usporedba algoritama strojnog učenja primijenjenih u proizvodnji.....	7
3.2 Problemi implementacije algoritama strojnog učenja u realnom vremenu. ...	10
3.3 Dostupna rješenja za ispitne SCADA sustave (<i>testbed</i>).....	12
3.4 Kompleksnost sučelja i performanse operatera	15
4. Metodologija istraživanja	18
4.1 Dizajn istraživanja	18
4.2 Ispitni SCADA sustav.....	20
4.3 Konfiguracija ispitnog SCADA sustava.....	23
4.4 Parametri procesa.....	29
4.5 Simulacija promatranog procesa.....	31
4.6 Korektivne mjere od strane operatera	36
4.7 Skup podataka (dataset) i podatkovni okvir (dataframe).....	39
4.8 Odabir metode strojnog učenja	49
5. Rezultati	53
5.1 Više-izlazna linearna regresija (uz <i>Gradient Boost regressor</i>).....	53
5.2 Više-izlazna klasifikacija (Stroj potpornih vektora).....	56
6. Diskusija.....	59
7. Zaključak.....	64
8. Literatura	65
9. Popis slika	69
10. Popis tablica	69
11. Popis Grafikona.....	70
12. Sažetak	70
13. Abstract.....	71

1. Uvod

Za upravljanje proizvodnim procesima odgovoran je industrijski kontrolni sustav (engl. ICS - Industrial Control System) koji u današnjem obliku objedinjuje domene automatizacije i informatizacije industrijskih postrojenja uz visok stupanj integracije i interoperabilnosti između operacijskih tehnologija (engl. OT - Operational Technology) i informacijskih tehnologija (engl. IT - Information Technology). Na taj se način postiže jedinstvena funkcionalna cjelina usmjerena ka kontroli procesa te akviziciji velike količine podataka generiranih od strane industrijskih procesa. ICS uz neposrednu kontrolu i nadzor procesa, dodatno služi kao izvor podataka prema višim sustavima na razini korporativne mreže. Postoji više definicija ICS-a pri čemu se općenito podrazumijeva termin koji opisuje integraciju hardvera i softvera uz mrežnu komponentu sa ciljem potpore kritične infrastrukture. „Industrijski kontrolni sustav je opći termin koji obuhvaća nekoliko tipova kontrolnih sustava i povezane instrumentacije korištene za industrijsku procesnu kontrolu“ (Wikipedia, 2021). Slika 1 prikazuje gore opisani sustav pri čemu je naglašena mrežna komponenta. U tom smislu ICS objedinjuje proizvodnu mrežu te mrežu za nadzor i kontrolu procesa (engl. SCADA - Supervisory Control and Data Acquisition System) u jedinstveni sustav na koji se dalje nadovezuje korporativna mreža.



Slika 1: Industrijski kontrolni sustav (Trend Micro Inc., 2021)

Tijekom posljednjeg desetljeća rapidno se povećava razina složenosti ICS-a uz višestruke razloge jednoznačno obuhvaćene konceptom industrije 4.0 koji prvenstveno uključuje tehnologije kibernetičko-fizičkih sustava, industrijski internet stvari (engl. IIoT - Industrial Internet of Things) i računalstvo u oblaku (Hermann, Pentek and Otto, 2016; Lu, 2016). U tom smislu Industrija 4.0 predstavlja korak u tehnološkoj evoluciji od uklopljenih sustava do kibernetičko-fizičkih sustava (Leitão, Colombo and Karnouskos, 2016).

Primijenjene novih tehnologija nude višu razinu kontrole i integracije. IIoT i bežični protokoli velikog dometa omogućuju integraciju inače izoliranih proizvodnih pogona. Tehnologije iskorištavanja obnovljivih izvora energije generiraju velik broj umreženih farmi vjetroelektrana i solarnih elektrana. Sve navedeno, s ciljem povećanja produktivnosti i pouzdanosti sustava, rezultira značajnim porastom razine automatizacije, širenjem računalnih mreža, implementacijom naprednih softverskih rješenja za praćenje i upravljanje industrijskim procesima s opsežnim mogućnostima analize, te generiranjem velike količine podataka u oblaku. Slijedom toga, dobiveni ICS postaje sve kompleksnije održavati ali i sve kompleksnije njime učinkovito upravljati u smislu kontrole industrijskog procesa. To ne izgleda nužno tako imajući u vidu trend ulaska umjetne inteligencije u industrijske pogone za što se odlučuje sve više kompanija (Pokorni *et al.*, 2021), visoku razinu implementacije algoritama strojnog učenja na području optimizacije industrijskih procesa kojima je cilj upravo učinkovitije upravljanje i kontrola istog (Goldin *et al.*, 2017; Stanula, Ziegenbein and Metternich, 2018; Rehse, Mehdiyev and Fettke, 2019; Qin and Chiang, 2019; Wuest *et al.*, 2016) te primjena tehnika strojnog učenja za detekciju kvarova u proizvodnoj industriji (Ademujimi, Brundage and Prabhu, 2017), energetici (Kim *et al.*, 2011), te na području prediktivnog održavanja (Settemsdal, 2019).

Sve od navedenog odnosi se na primjenu tehnika i algoritama strojnog učenja izravno na kontrolu industrijskog, odnosno proizvodnog proces. Time se kontrola i nadzor istog može značajno poboljšati te u konačnici podići razina učinkovitosti. Takav je pristup dostatan u potpuno automatiziranim procesima proizvodnje ili procesima koji sadrže takve podsustave, te za udaljene autonomne sustave kao što je slučaj sa geografski izdvojenim sustavima vjetroelektrana. Međutim, postoje brojni proizvodni procesi koji uz određenu razinu industrijske automatizacije tradicionalno uvelike ovise i o čovjeku tj., operateru koji nadzire proces i donosi ključne odluke. To je osobito bitno za procese

kod kojih postoji određena stohastička komponenta koja se može različito manifestirati, a može rezultirati destabilizacijom procesa, te ima rijetku pojavnost. Neki od takvih proizvodnih procesa su oni koji podrazumijevaju kontinuirani tok materijala što je karakteristično za metalnu industriju i pogone kontinuiranog lijevanja, raznih izvedbi galvanizacijskih linija te hladnog i toplog valjanja. U tom slučaju povećanje razine kompleksnosti upravljanja procesom ima izravan utjecaj na učinkovitost s obzirom na ljudska ograničenja i sposobnosti. Postoje istraživanja novijeg datuma koja fokus prebacuju sa procesa na čovjeka. Golan, Cohen and Singer, (2020) predlažu novi koncept *operator-workstation* za industriju 4.0 koji prati stanje operatera i daje ocjenu njegovih performansi u odnosu na stanje promatranog procesa. Problem adekvatne reakcije čovjeka s obzirom na kompleksnost sučelja Reguera-Bakhache *et al.*, (2020) predlaže riješiti adaptivnim sučeljem koje se prilagođava korisniku u realnom vremenu (engl. RT – *Real Time*). Problem kompleksnosti kontrole procesa i preopterećenost korisnika prepoznaje se i u domeni avio-industrije gdje autor predlaže oblik primjene umjetne inteligencije kao potporu pilotima, te u zaključku navod i slijedeće: „Pomoć temeljena na umjetnoj inteligenciji sadržavat će sustave temeljene na umjetnoj inteligenciji u primarnoj ulozi pomaganja ljudima, jer je umjetna inteligencija samo još jedan računalni alat koji treba pravilno koristiti“. (Hourlier, 2021) (samostalni prijevod).

Cilj ovog rada je primijeniti tehnike strojnog učenja kako bi se pružila potpora operateru koji upravlja proizvodnim procesom putem HMI-a, na način da se sugeriraju potrebne korektivne mjere koje je potrebno poduzeti s obzirom na stanje procesa. Navedeno će se postići analizom podataka generiranih iz proizvodnog procesa, kao i prethodno poduzetih korektivnih mjera u dosadašnjim intervencijama operatera. Za generiranje skupa podataka iz proizvodnje koristiti će se HMI simulacija hidrauličkog podsustava tipičnog za pogone kontinuiranog lijevanja u metalnoj industriji, te će biti razmatrani modeli nadziranog učenja (engl. supervised learning). S obzirom da će predviđanje korektivnih mjera uzeti u obzir dosadašnje intervencije operatera, time se podrazumijeva da će se pratiti reakcije više različitih operatera. Kako je u realnoj situaciji za očekivati da postoje razlike u iskustvu operatera, kao i njihovom poznavanju promatranog procesa, time je nužno da svaka zabilježena reakcija operatera sadrži i identifikacijski broj istog. Time će se u konačnici, za određenu

situaciju u realnom vremenu, moći ponuditi one korektivne mjere koje bi odgovarale mjerama poduzetim od strane naj iskusnijeg operatera.

2. Definicija problema i uvodna razmatranja

Kod prethodno spomenutih industrijskih pogona i kontinuiranih procesa u metalnoj industriji postupak zaustavljanja i ponovnog pokretanja procesa kompleksan i skup da bi se dnevno prekidao. Takvi su kontinuirani proizvodni procesi često ulančani u nizu talionica-ljevaonica-valjaonica, pri čemu zastoj na jednome djelu posljedično stvara zastoj i na svakom sljedećem. Osim toga, spomenut ulančani industrijski pogon može biti dug više stotina metara, čime je nužno da se nadzire sa više lokacija, odnosno kontrolne sobe su podijeljene prema sekcijama proizvodnje.

2.1 Operateri – ljudski čimbenik

Da bi se održavao kontinuitet proizvodnog procesa iz opisanog industrijskog pogona logično je u nadzor uključeno i više timova operatera u smjenskom radu. Uzimajući u obzir da se valjaonica tipično sastoji od ulazne, procesne i izlazne sekcije, te sa postoji barem po jedna sekcija za talionicu i ljevaonicu, jednostavnom kalkulacijom dolazimo do najmanje deset operatera da bi se osigurao nadzor u jednoj smjeni uz uvjet od minimalno dva operatera po kontrolnoj sobi. Kada se k tome uzmu o obzir barem tri smjene za neprekidno praćenje proizvodnje, dolazimo do minimalnog broja od trideset operatera. Premda je realno očekivati da je taj broj veći uzimajući u obzir pričuvnu smjenu i zamjensko osoblje za slučajeve izostanaka, već je iz ovog broja očigledno da se mogu očekivati razlike u performansama i sposobnostima pojedinih operatera unutar skupine u istoj smjeni. Naime, čak i kada bi na razini tvornice postojao savršen sustav obuke, i dalje je razumno očekivati da u skupini od trideset osoba na istom radnom mjestu postoji određena fluktuacija kadrova, te u konačnici i smjena generacija. U takvim uvjetima nije realno za očekivati da postoji sustav obuke i praćenja rada operatera koji bi imao takvu razinu učinkovitosti koja bi mogla osigurati vremenski kontinuitet ujednačene kompetentnosti svih operatera.

Osim toga, trenutne potrebe mogu zahtijevati privremeno premještanje operatera između procesnih sekcija, pri čemu nije realno za očekivati da svaki operator podjednako poznaje svaki segment procesa. Sve navedeno neminovno rezultira

očekivanim različitim performansama i odlukama pojedinačnih operatera u reakcijama na različita stanja promatranog proizvodnog procesa.

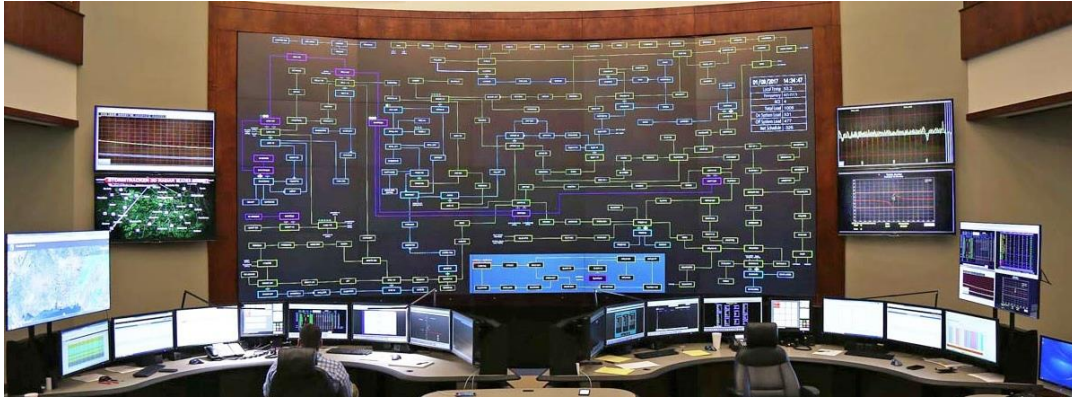
2.2 HMI – kompleksnost sučelja

Kako je u uvodu rečeno, ICS postaje sve kompleksnije održavati ali i sve kompleksnije njime učinkovito upravljati u smislu kontrole industrijskog procesa. Kada je riječ o izravnom upravljanu nadzoru procesa koje vrši operater, ta se kompleksnost prije svega očituje u grafičkom sučelju. Slika 2 prikazuje kontrolnu sobu karakterističnu za proizvodne pogone. Uz kontinuirani rast količine podataka kao posljedica koncepta pametne tvornice (engl. smart factory) u okviru industrije 4.0 posljedično se povećava i količina podataka prikazana na grafičkom sučelju. Takav razvoj situacije sa namjerom što preciznije i sveobuhvatnije kontrole i nadzora procesa nameće fizičko povećanje površine grafičkog sučelja što rezultira dodatnim monitorima (Engl. *multi-window display*) i sustavima sa video zidom (engl. video-wall systems) (LAMB, 2021). Slika 2 daje primjer konfiguracije sučelja sa više monitora u kontrolnoj sobi proizvodnog pogona.



Slika 2: Kontrolna soba za nadzor proizvodnog procesa – HMI (GIC, 2017)

Primjenom sustava sa video zidovima karakterističnim za centralizirane sustave kontrole, postiže se veća gustoća prikaza podataka, ali time i veća vjerojatnost za težim uočavanjem kritičnih detalja od strane operatera. Slika 3 prikazuje HMI sustav temeljen na video zidu koji je karakterističan za nadzor elektro-energetskih sustava distribucije.



Slika 3: Kontrolna soba za nadzor elektro-energetskog sustava (LAMB, 2021)

Uz veću količinu prikazanih podataka i višu razinu kompleksnosti sučelja, logično se povećava i mogućnost pogreške operatera. Premda postoje različiti pristupi problemu kompleksnosti HMI sustava (Shi *et al.*, 2019; Cataldo and De Souza, 2011; Reguera-Bakhache *et al.*, 2020) izostaje mogućnost izravne potpore operateru u donošenju odluke u realnom vremenu koja će biti temeljena na dosadašnjoj dobroj praksi i koja će biti neovisna u odnosu na vrstu, dizajn i kompleksnost sučelja.

3. Prethodna istraživanja

Osvrt na dosadašnja istraživanja koji slijedi u nastavku pruža temeljna saznanja od važnosti za dizajn istraživanja, donošenja odluke o arhitekturi i konfiguraciji SCADA sustava potrebnog za simulaciju proizvodnog procesa, odabiru odgovarajućeg modela strojnog učenja u odnosu na specifičnosti industrijskog okruženja i proizvodnih pogona u domeni industrije 4.0 (Ademujimi, Brundage and Prabhu, 2017), te razumijevanje karakteristika i atributa procesnih podataka koji su jedinstveni u neregularnostima (Qin and Chiang, 2019), nastalim kao posljedica raznih hardverskih utjecaja na senzorske uređaje u proizvodnom polju, a koji su specifični za pojedine industrijske procese (vibracije, elektro-magnetska zračenja, metalna prašina, ekstremno visoke temperature, fizička oštećenja). Osim navedenog, prethodna istraživanja u nastavku teksta sadrže iskustva implementacije algoritama strojnog učenja u realnom vremenu, konceptualne prijedloge rješenja za praćenje performansi operatera, te podizanje učinkovitosti operatera na razini HMI sučelja (Mehta *et al.*, 2012; Reguera-Bakhache *et al.*, 2020).

3.1 Usporedba algoritama strojnog učenja primijenjenih u proizvodnji.

Ademujimi, Brundage i Prabhu (2017) u svom radu pružaju pregled značajnijih algoritama strojnog učenja temeljem dvadeset izdvojenih radova objavljenih u periodu od 2007. god. do 2017. god., te kao relevantne izdvaja slijedeće algoritme strojnog učenja i daje njihovu usporedbu u tablici 1:

- Bayesove mreže (engl. BN - Bayesian Networks).
- Umjetne neuronske mreže (engl. ANN - Artificial Neural Network).
- Stroj potpornih vektora (engl. SVM – Support Vector Machine).
- Skriven Markovljev model (engl. HMM - Hidden Markov Model).

Tablica 1: Prednosti i nedostaci metoda strojnog učenja (Ademujimi, Brundage and Prabhu, 2017) (vlastiti prijevod).

Metoda	Prednosti	Nedostaci
BN	Intuitivno, lako razumljivo, dobro za stohastičko modeliranje. Može se koristiti za modeliranje hijerarhijskih razina višestrukih uzroka i posljedica, te može zaključivati u oba smjera (predviđanje i dijagnoza).	Struktura stabla čini je relativno manje lakom za inicijalizaciju. Izgradnja strukture stabla može biti izazovna.
ANN	Može modelirati nelinearne složene probleme s visokim stupnjem preciznosti. Relativno je lakše inicijalizirati i nema potrebe za određivanjem mrežne strukture kao kod BN.	Model nije lako protumačiti i ne može se nositi s neizvjesnošću u inputima. Intenzivno u zauzeću procesora (računalne snage) što usporava konvergenciju tijekom treninga. Model dobro predviđa u treningu ali loše generalizira (<i>overfitting</i>).
SVM	Izvrstan u modeliranju linearnih i nelinearnih odnosa. Vrijeme izračunavanja relativno je brzo u usporedbi s ANN-om. Teži dobro generalizirati čak i s ograničenim mogućnostima.	Odabir „kernel“ parametara funkcije je zahtijevan. Nije lako ugraditi domensko znanje. Teško je razumjeti naučenu funkciju.
HMM	Izvršno u modeliranju procesa sa stanjima koja se teško opažaju.	Proces treninga obično je računski intenzivan (zauzeće procesora).

Tablica 1 pruža dobru polaznu točku za odabir modela strojnog učenja u odnosu na zadani problem predikcije korektivnih mjera iz podataka generiranih proizvodnim procesom s obzirom da postoje značajna preklapanja u skupu podataka budući oba sadrže vrijednosti dobivene iz uređaja u proizvodnom polju (eng, *production field devices*). Međutim, Ademujimi, Brundage i Prabhu (2017) svoje istraživanje usredotočuju isključivo na poboljšanje dijagnoze i detekciju primarnog uzroka kvarova u proizvodnom sustavu (engl. *fault detection*). U tom smislu, algoritmi strojnog učenja primjenjuju se i na kompleksne podatke koji su specifični za detekciju anomalija u materijalu, konstrukcijama te površini materijala (Correa, Bielza and Pamies-Teixeira, 2009). Navedeno uvjetuje primjenu metoda strojnog učenja poput ANN i HMM koje prema karakteristikama danim u tablici 1 nisu prvi izbor u obradi striktno tabelarnih podataka sa kontinuiranim i diskretnim varijablama koji se očekivano generiraju iz senzorskih uređaja proizvodnog polja, te zahtijevaju značajnu računalnu snagu što može biti prepreka u implementaciji koja mora polučiti rezultat u realnom vremenu. Kang, Catal i Tekinerdogan (2020) proveli su sveobuhvatno istraživanje sa ciljem utvrđivanja *state-of-the-art* na području implementacije strojnog učenja u proizvodnim pogonima. Istraživanje je obuhvatilo 310 dotadašnjih relevantnih radova, te je izdvojilo najčešće probleme u proizvodnim linijama, i primijenjene metode strojnog učenja kako je prikazano u tablici 2.

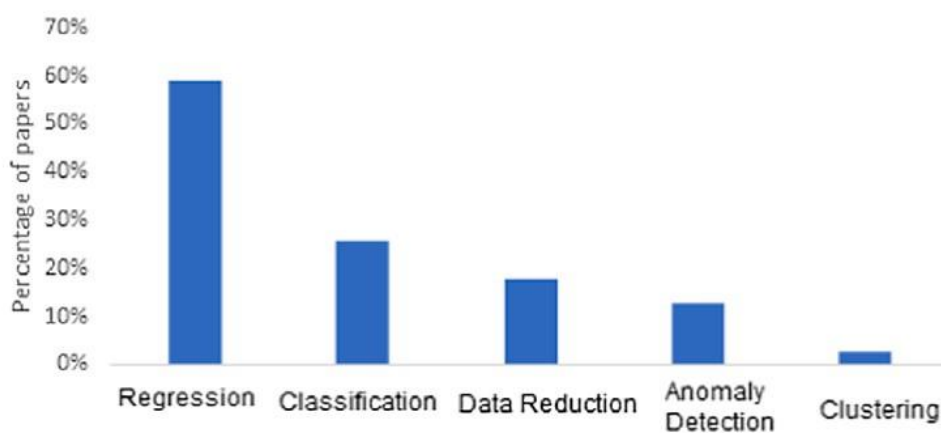
Tablica 2: Problemi, kategorije i primijenjene metode (Kang, Catal and Tekinerdogan, 2020), (vlastiti prijevod).

Realan problem	OEE¹ Kategorija	Metoda strojnog učenja	Zavisne varijable
Optimizacija kvalitete	Kvaliteta / performanse	Regresija	Kontinuirana mjerenja
Dijagnostika kvara	Dostupnost	Klasifikacija / Detekcija anomalija	Kategorijske
Detekcija greške	Kvaliteta / performanse	Klasifikacija / Detekcija anomalija	Kategorijske

¹ OEE = Ukupna učinkovitost opreme (engl. Overall Equipment Effectiveness)

Optimizacija rasporeda	Performanse	Regresija	Vremenski ciklus
Reduciranje otpada	Kvaliteta	Regresija	Kontinuirana mjerenja
Poboljšanje prinosa	Kvaliteta	Regresija	Kontinuirana mjerenja
Preventivno održavanje	Dostupnost	Regresija	Preostalo vrijeme (RUL ²)

U nastavku, slika 4 daje i prikaz distribucije metoda strojnog učenja prema učestalosti pojavljivanja u obuhvaćenim radovima.



Slika 4: Distribucija metoda strojnog učenja (Kang, Catal and Tekinerdogan, 2020).

Osim navedenog, Kang, Catal and Tekinerdogan (2020), analiziraju i zastupljenost industrijskih grana, te uspješnost u primjeni rješenja zasnovanih na strojnom učenju, iz čega se uočava kako se strojno učenje značajno primjenjuje u domeni metalne i poluvodičke industrije. Budući da su ti procesi vrlo složeni, te generiraju više podataka, time je dostupnost i kvaliteta podataka ključ performansi implementiranih modela strojnog učenja. Nadalje, uočava se kako nadzirano učenje (engl. supervised learning) uglavnom postiže bolje rezultate u odnosu na polu-nadzirano (engl. semi-supervised) ili nenadzirano (engl. unsupervised).

3.2 Problemi implementacije algoritama strojnog učenja u realnom vremenu.

² RUL = Preostao vijek trajanja (engl. Remaining Useful Life)

Albertos et al. (1999) prepoznaju osnovne probleme i specifičnosti kontrole procesa kao i implementacije algoritama strojnog učenja u realnom vremenu na toj razini. Promatrajući kontrolu procesa kao kompleksni sustav podrazumijeva se postojanje više različitih procesa procesa (engl. multiprocess) sa većim brojem varijabli (engl. multivariable) i senzora, te višestrukim kontrolnim petljama (engl. multiloop) kojima se postiže više ciljeva (engl. multigoal) sa različitim vrstama podataka i razinama odluka (engl. multilevel) uz različitim brzinama uzorkovanja (engl. multirate). U takvim okolnostima pravilno i uniformno uzorkovanje teško je izvedivo u realnom vremenu, te se za pojedine režime rada podsustava kontrole razmatra uzorkovanje sa varijabilnim periodom, nesinkrono uzorkovanje i višestruko uzorkovanje (engl. multirate).

S obzirom da se ovo referentno istraživanje također bavi problemom sveobuhvatne kontrole procesa u realnom vremenu, pri čemu se kao primarna poteškoća predstavlja različit raspon zahtjeva koji se postavljaju pred sustav kontrole, predlažu se slijedeći koraci u pristupu izgradnje cjelovitog rješenja kontrole procesa (uključujući i implementaciju modela strojnog učenja) kako bi se odgovorilo na izazove koji proizlaze iz gore spomenutih specifičnosti procesa:

- Pretpostaviti da postoje neovisni problemi kontrole procesa, razviti odgovarajuće upravljačke algoritme za svaki od njih i primijeniti ih u odgovarajućem trenutku, te pokušati minimizirati kašnjenja i interakcije.
- Pretpostaviti stvarno stanje i razviti globalni algoritam upravljanja zasnovan na cjelovitom i detaljnom modelu procesa.
- Uzeti u obzir danu strukturu sustava kontrole procesa i iskoristiti postojeće raspoložive programske alate za izvršenje zadataka. Zadaci se mogu izvršiti s periodičnom shemom (planer zadataka) ili se mogu pokrenuti kada se događaj detektira u kontrolnom sustavu (engl. event-driven).

Albertos et al. (1999) predlažu slijedeću petlju implementacije sa sljedećim redoslijedom:

1. Prikupljanje podataka sukladno vremenskim ograničenjima (sa varijabilnim periodom, nesinkrono uzorkovanje, višestruko uzorkovanje).
2. Procesiranje kontrolne aktivnosti (mora biti izvršeno u što skorijem mogućem roku nakon prikupljanja podataka). Na kraju ovog koraka, osnovna kontrolna

funkcija je spremna za generiranje upravljačkih signala ili drugog rezultata obrade.

3. Poboljšanje rješenje iz prethodnog koraka. U ovom koraku algoritam ili metoda može poboljšati dobiveni rezultat. Ovisno o tipu zadatka i kontrolne aktivnosti, ovaj korak može biti nepotreban.
4. Izvedba kontrolne aktivnosti (engl. Output the control action) mora biti izvršena što ranije ili unutar zadanog fiksnog vremenskog intervala.
5. Procjena stanja i priprema i priprema podataka za slijedeći ciklus. Ovaj korak mora biti završen prije početka idućeg ciklusa.

Rezimirajući gore navedeno, moguće je izvući ključne pretpostavke i redoslijed aktivnosti koje se odnose na razradu i implementaciju predikcije korektivnih mjera promatranog procesa, Primjena algoritama strojnog učenja u industrijskom ambijentu proizvodnog procesa podrazumijeva usvajanje gore navedenih koraka i pretpostavki, kao i petlju implementacije koja u predmetnom istraživanju podrazumijeva da je kontrolna aktivnost isključivo na razini potpore u donošenju odluke operatera koji je u potpunosti samostalan u konačnoj odluci, te sama implementacija predikcije korektivnih mjera ne rezultira konkretnim kontrolnim aktivnostima prema proizvodnom procesu. Time se može dozvoliti i određena fleksibilnost u odnosu na vremenska ograničenja ciklusa koja bi se inače morala striktno poštivati. Također, budući predmet ovog rada nema za neposredni cilj primjenu na cjelokupni proizvodni proces, razina kompleksnosti sustava može se znatno reducirati primjenom na jedan od neovisnih podsustava procesa. Na taj se način može postići jednostavnije uzorkovanje s obzirom na reduciranu količinu podataka, broj kontrolnih petlji, višestruke procese i ostalim navedenim specifičnostima kontrole procesa.

3.3 Dostupna rješenje za ispitne SCADA sustave (*testbed*).

Zhang *et al.* (2015) u svojem istraživanju analiziraju dizajn i konstrukcijske sheme predloženih ispitnih SCADA sustava, te pružaju pregled korištenih komponenti i protokola. Premda je istraživanje prvenstveno usmjereno ka ispitivanju kibernetičkih napada i odgovarajućih strategija odgovora na iste, također pruža usporedbe osnovnih kategorija, te njihove prednosti i nedostatke što je općenito primjenjivo u odabiru temeljnog smjera izgradnje ispitnog SCADA sustava za predmetno istraživanje. Tablica 3 daje usporedbu softverske i hardverske kategorije.

Tablica 3: Kategorije ispitnih SCADA sustava (Zhang *et al.*, 2015) (vlastiti prijevod)

Prednosti	Nedostaci
Softverski ispitni SCADA sustavi	
Prednosti	Nedostaci
Jednostavniji za repliciranje izgrađenog sustava. Lakša nadogradnja i izmjene. Mogućnost zajedničke platforme za više timova (lakša usporedba rezultata).	Eksperimenti koji se odnose na točno određeni hardverski uređaj nisu mogući.
Slika stanja (engl. <i>snapshot</i>) omogućava usporedbu različitih stanja čime je lakše analizirati veću količinu podataka.	Simulirane komponente ne mogu biti odraz realnih situacija u stvarnom SCADA sustavu. Određeni parametri nisu točni (prim. Kašnjenje signala).
Lakša izgradnja kompleksnijih sustava. Minimizira ukupne troškove simuliranjem hardverski komponenti.	
Hardverski ispitni SCADA sustavi	
Eksperimentalni podaci su točniji i uvjerljiviji.	Pojedini hardverski uređaji mogu biti skupi, što znatno utječe na ukupnu cijenu ispitnog sustava većih dimenzija.
Kontrolne procedure su točne i bez negativnog utjecaja ostalog softvera poput operativnog sustava.	Skalabilnost hardverskih sustava je lošija od softverskih. Nije jednostavno mijenjati hardversku strukturu SCADA sustava.
Neovisnost hardverskih uređaja daje mogućnost ciljanog eksperimenta za točno određeni uređaj, bez dodatnih utjecaja na ostale.	Performanse hardverskih uređaja sa samo nekoliko značajki je limitirano, tj. ne mogu pružiti realnu sliku stvarnih uvjeta.

Zhang *et al.* (2015) se također referencira na druga istraživanja koja su predložila različite konfiguracije i sheme za izgradnju ispitnih SCADA sustava (Queiroz *et al.*, 2009; Davis *et al.*, 2006; Reaves and Morris, 2012; Fovino *et al.*, 2010; Reaves and Morris, 2012). Zajedničko za većinu razmatranih istraživanja jeste da su za izgradnju ispitnih SCADA sustava odabrali softversku (virtualizacijsku) opciju, uz eventualne fizičke komponente u proizvodnom polju (engl. field devices). Tablica 4 prikazuje

usporedbu primijenjenih kategorija prema komponentama u obuhvaćenim istraživanjima.

Tablica 4: Usporedba kategorija komponenti ispitnih SCADA sustava (Zhang *et al.*, 2015).

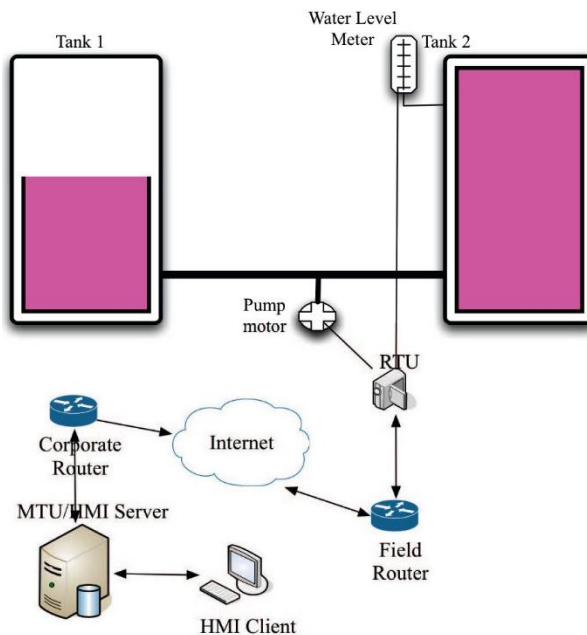
Testbed	HMI	MTU³	RTU⁴	Field devices
Davis et al.(2006)	S	S	S	S
Reaves and Morris (2012)	S	S	S	S
Fovino et al. (2010)	S	S	S	S
Queiroz et al. (2009)	S	S	S	H
Morris et al. (2011)	S	H	H	H

S = softverska izvedba; H = hardverska izvedba

Queiroz *et al.* (2009) daju primjer softverskog ispitnog SCADA sustava uz implementaciju hardverskih uređaja u polju (senzorski uređaji). Sa aspekta predmetnog rada značajno je kako se za simulaciju procesa odabrao *case-study* pojednostavljenog pogona za pročišćivanje voda (engl. WTP - *Water Treatment Plant*). S obzirom da je primarni cilj ovog referentnog istraživanja bila kibernetička sigurnost SCADA sustava, te se ispitni sustav koncipirao sa naglaskom na mrežnu komponentu, time je WTP značajno pojednostavljen kao proces. Međutim, takav pristup daje dobre smjernice za definiranje podsustava proizvodnog procesa koji će se simulirati. Naime, s obzirom na osnovne elemente (pumpa, tankovi, pokazivači razine) i parametre (razina u tanku, status pumpe, upravljački signali) simuliranog WTP-a, u domeni kontinuiranih proizvodnih procesa može se povući određena analogija sa hidrauličkom jedinicom. Ukoliko se simulacija proizvodnog kontinuiranog procesa limitira na podsustav hidrauličke jedinice uz zadržavanje potpune funkcionalnosti iste (bez reduciranja broja signala, komponenti i funkcija) moguće je sa aspekta nadzora i kontrole procesa putem HMI-a (pogled operatera) postići dovoljno realističnu sliku ponašanja sustava u realnom vremenu, a time i realističnu primjenu modela strojnog učenja u realnom vremenu. Slika 5 prikazuje konfiguraciju SCADA sustava sa osnovnim elementima WTP-a.

³ MTU = Master Terminal Unit

⁴ RTU = Remote Terminal Unit



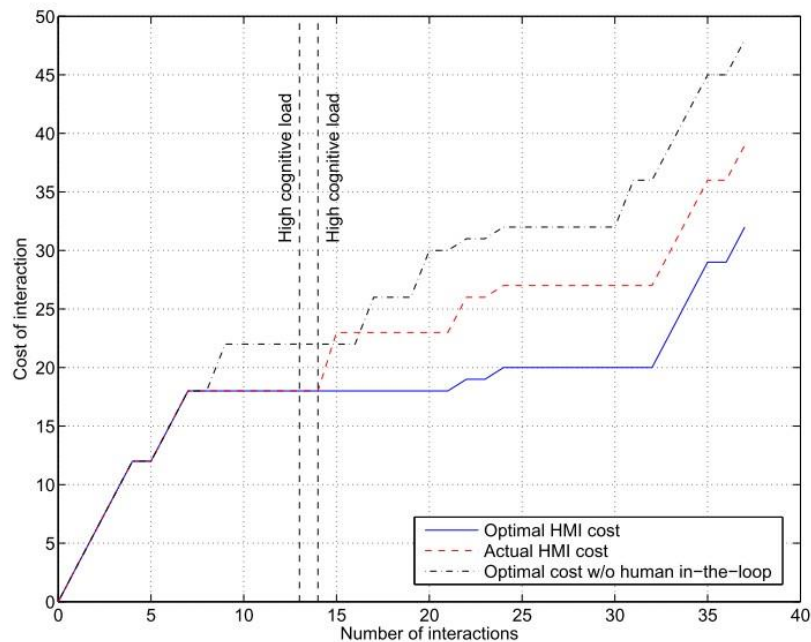
Slika 5: ispitni SCADA sustav za pojednostavljeni WTP proces

Premda gornja slika daje dobre smjernice za odabir promatranog procesa, primarni fokus na mrežnu komponentu nije ključan imajući u vidu kako se predmetni rad odnosi primarno na lokalnu implementaciju modela strojnog učenja i aktivnosti operatera u lokalnoj kontrolnoj sobi. Time se prikazani elementi mogu značajno reducirati i virtualizirati.

3.4 Kompleksnost sučelja i performanse operatera

Premda se u nadzoru i kontroli kompleksnih sustava ne može isključiti čovjeka kao ključnog čimbenika, očigledno je da u upravljanu kompleksnim sustavima postoje pitanja učinkovitosti s obzirom na ljudska ograničenja i kognitivne sposobnosti prema kojima postoje različiti pristupi (Mehta *et al.*, 2012; Reguera-Bakhache *et al.*, 2020; van Rensburg, Kamin and Davis, 2019; Golan, Cohen and Singer, 2020). Osim kompleksnosti sustava automatizacije, a time i sučelja, Mehta *et al.* (2012) u svom radu uočavaju i dodatni utjecaj na performanse operatera koji je u vezi sa neizvjesnošću (eng. uncertainty) u radu sustava, tj., nepouzdanost sustava koje stvara nepovjerenje prema istome. „Nepouzdana automatizacija zajedno s velikim kognitivnim opterećenjem može znatno pogoršati performanse, a performanse operatera su osjetljivije na pogreške automatizacije nego na vlastite pogreške“ (Mehta *et al.*, 2012) (vlastiti prijevod). Predloženo rješenje kombinira metode praćenja aktivnosti i kognitivnog opterećenja operatera (*engl.* CSI - Cognitive State Indicator) sa *Tracking controller-om* koji mjeri razliku između izvršene kontrole operatera i predviđene

optimalne kontrolne aktivnosti. Slika 6 prikazuje rezultat simulacije koja su Mehta *et al.* (2012) dobili u istraživanju.

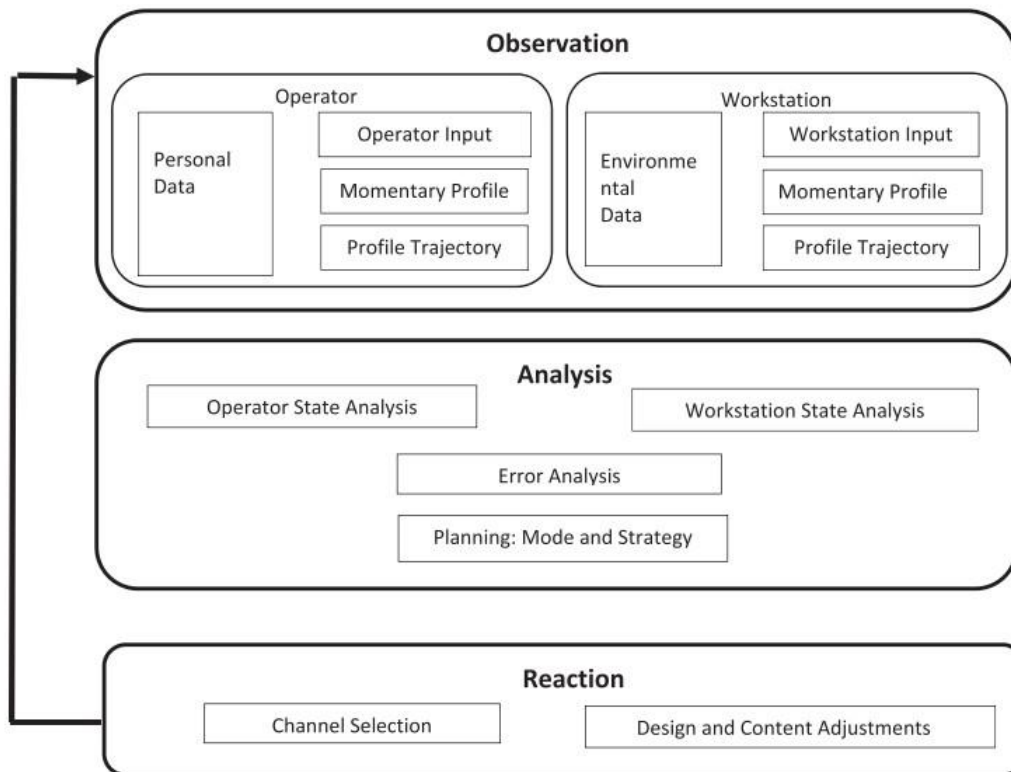


Slika 6: Performanse sustava u odnosu na prisustvo operatera

Iz gornje slike jasno se da uočiti značaj odluke i performansi operatera na učinkovitost sustava, što opravdava daljnji razvoj sustava u smjeru pružanja potpore operateru, odnosno rješenja koja su usmjerena na pružanju potpore i percepciju operatera kao ključnog čimbenika sustava (engl. human in-the-loop).

Golan, Cohen i Singer (2020) prezentirali su novi konceptualni *framework* za oblikovanje interakcije između operatera i HMI radne stanice pod nazivom "Operator – Workstation Interaction 4.0" (OWI 4.0) koji je dizajniran kao potpora operateru, te da inteligentno reagira u odnosu na njegovo kognitivno i fizičko stanje. Primjerice, sustav ima sposobnost odlučiti hoće li i kako pomoći operateru čiju razinu pozornosti treba povećati. *Framework* je također u stanju prepoznati neučinkovite aktivnosti operatera, te evidentirati izgubljene resurse. Cijeli se sustav temelji na kognitivnoj proizvodnji (engl. cognitive manufacturing) kao ključni element industrije 4.0, što se odnosi na integraciju ljudskih kognitivnih sposobnosti, tj, umjetne inteligencije i automatiziranog procesa u proizvodnom polju (Zhao and Xu, 2010), te anticipira očekivani razvoj kao što su HMI sučelja fokusirana na izbjegavanje grešaka operatera (engl. human error avoidance tactics) (Pinto, Melo, and Saldanha, 2015) i koncept sučelja sa emocionalnom inteligencijom (engl. machine emotional intelligence interface) (Picard,

Vyzas, and Healey, 2001). Slika 7 prikazuje strukturu predloženog OWI 4.0 sustava sa ključnim ciklusom aktivnosti promatranja, analize i reakcije.



Slika 7: OWI 4.0 *framework*

U gore prikazanoj strukturi vrši se usporedna i kontinuirana aktivnost prikupljanja podataka o neposrednim aktivnostima ali i stanju operatera te radne stanice (SCADA, HMI) pri čemu opservacije generiraju trenutni (engl. momentary profile) ali i kontinuirani (engl. profil trajectory) profil koji omogućava praćenje trendova. Nastavni proces analize na temelju prikupljenih podataka utvrđuje stanje sustava (učinkovitosti, potreba za održavanjem, opterećenost) te opće stanje operatera. Analiza grešaka (engl. error analysis) te planiranje načina i strategije intervencije (engl. mode and strategy) uključuje primjenu modela strojnog učenja i predikcije korektivnih mjera za preveniranje uočenih grešaka koje će se primijeniti na adekvatan način. Konačno, proces rezultira odgovarajućom reakcijom kroz odgovarajuće komunikacijske kanale sa odgovarajućim sadržajem i na adekvatan način.

Ovako koncipirana struktura OWI 4.0 konceptualnog *framework-a* kojeg su predložili Golan, Cohen i Singer (2020), u velikoj se mjeri može primijeniti na predmetni rad, odnosno slijed aktivnosti u predikciji korektivnih mjera promatranog procesa. Naime, ako se iz gornjeg prikaza isključe elementi prikupljanja podataka o fizičkom i

kognitivnom stanju operatera kao i sustava automatizacije, dolazimo do podataka o trenutnoj aktivnosti operatera, dok se stanje sustava automatizacije može zamijeniti sa podacima iz proizvodnog polja. U daljnjem koraku, na temelju prethodno pohranjenih podataka, analiza takvih podataka odnosi se na primjenu algoritama strojnog učenja u usklađivanju aktivnosti operatera u odnosu na tijek procesa, što u konačnici rezultira predikcijom korektivnih mjera za aktualno stanje procesa. Posljednji korak reakcije ne mora nužno sadržavati odabir komunikacijskog kanala, ali se sadržajno može prilagoditi prema potrebi.

4. Metodologija istraživanja

Proveden je eksperiment temeljen na kvantitativnim metodama uz primjenu *Desk* metode prikupljanja podataka, čime su se kao izvor primarnih podataka koristili simulirani uvjeti procesa, tj., hidraulička jedinica kao jedan od neovisnih pod-procesa koji je u stanju generirati reprezentativan skup podataka. Središnji proces eksperimenta bila je primjena modela strojnog učenja nad vrijednostima koje predstavljaju signale dohvaćene sa polja, sa temeljnim ciljem predikcije upravljačkih signala (naredbi operatera putem HMI sučelja) koje služe kao korektivne mjere u stabilizaciji promatranog kontinuiranog procesa. Takav je eksperiment imao za cilj utvrditi mogućnost realizacije sustava potpore operateru u realnom vremenu na način da se za aktualno stanje praćenog procesa operateru sugeriraju preporučene korektivne mjere kao rezultat analize dosadašnjih poduzetih mjera kumulativno od strane svih operatera. Dodatni zahtjev eksperimenta bila je realizacija korištenjem lokalnih računalnih resursa na razini HMI radne stanice, te utvrđivanje razine učinkovitosti usporednom primijenjenom odabranih algoritama strojnog učenja. Sekundarni cilj bio je izvršiti analizu prikupljenih podataka te izdvojiti operatera čije su aktivnosti povijesno polučile bolje rezultate u smislu učinkovitije kontrole procesa. Time se ostvarila dodatna mogućnost predikcije učinkovitijih mjera od onih koje bi očekivano samostalnom odlukom primijenili manje iskusni operateri.

4.1 Dizajn istraživanja

Primijenjen je longitudinalni panel dizajn (engl. *fixed-sample panel design*) koji je omogućio prikupljanje podataka od istog seta simuliranih uređaja u polju, te u zadanim vremenskim točkama koje su definirane trenutkom generiranja kontrolnog signala, tj., korektivne aktivnosti operatera putem HMI sučelja. Primjena takvog nesinkronog

višestrukog uzorkovanja odgovor je na problem uzorkovanja u realnom vremenu s obzirom na odvijanje i specifičnosti kontinuiranih procesa gdje je pravilno i uniformno uzorkovanje teško izvedivo (Albertos *et al.*, 1999). Negativnost takvog načina formiranja skupa podataka leži u narušenim trendovima kao posljedica nepravilnih vremenskih razmaka u uzorkovanju, čime može doći do oscilacija vrijednosti varijabli u oba smjera između dva mjerenja. Međutim to nije od značaja za predmetno istraživanje kojemu je cilj predikcija korektivnih mjera (preporučene kontrolne aktivnosti od strane operatera) s obzirom na trenutno stanje procesa, što ne uključuje vremensku komponentu u skupu podataka.

Takav je dizajn uzorkovanja podataka u ovom slučaju apriori eliminirao *post-hoc*⁵ pogrešku budući se činjenica da jedna varijabla prethodi drugoj ne razmatra kao moguća uzročno-posljedična veza iz razloga što za rezultati eksperimenta u ovom obliku nije bilo bitno odrediti trendove promjene vrijednosti varijabli. Usko gledano, takav pristup potencijalno isključuje i vremenski slijed kao kriterij za ustanovljavanje kauzalnog odnosa budući da varijacija nezavisne varijable ne mora nužno prethoditi varijaciji zavisne varijable. To nije u potpunosti točno jer bi eliminiralo sekundarni cilj eksperimenta koji se odnosi na spomenuto izdvajanje povijesnih podataka za razdvajanje operatera prema učinkovitosti, a pri čemu vremenski slijed ima ključnu ulogu.

Kako je već prethodno spomenuto, u skladu sa softverskim modelom simulatora procesa WTP postrojenja (Queiroz *et al.*, 2009), simulacija praćenja kontinuiranog proizvodnog procesa izvedena je softverski te je reducirana na pod-sustav hidrauličke jedinice koji u dovoljnoj mjeri zadovoljava osnovne parametre promatranog procesa (simulacija signala sa senzorskih uređaja u proizvodnom polju, te ponašanje uređaja u odnosu na kontrolne signale sa HMI sučelja), i pruža realističnu sliku ponašanja sustava u realnom vremenu (kontinuirane varijable na izlazu iz pod-sustava u zadanim granicama vrijednosti). Istovremeno, rezultati koji su dobiveni u ovakvom obimu lako se mogu skalirati na cjelokupni proizvodni proces na razini postrojenja (P) multiplicirajući utrošak resursa (URes) sa brojem pod-sustava (PS) uz indeks korekcije (IndK) s obzirom na razliku u veličini skupa podataka određenog brojem signala po

⁵ Post hoc (skraćeni oblik od „post hoc, ergo propter hoc“) logička je zabluda koja nalaže da je raniji događaj uzrok kasnijeg događaja jednostavno zato što se dogodio ranije.

svakom pojedinom pod-sustavu (BrS) u odnosu na referentni podsustav sa najmanje signala (RefPS).

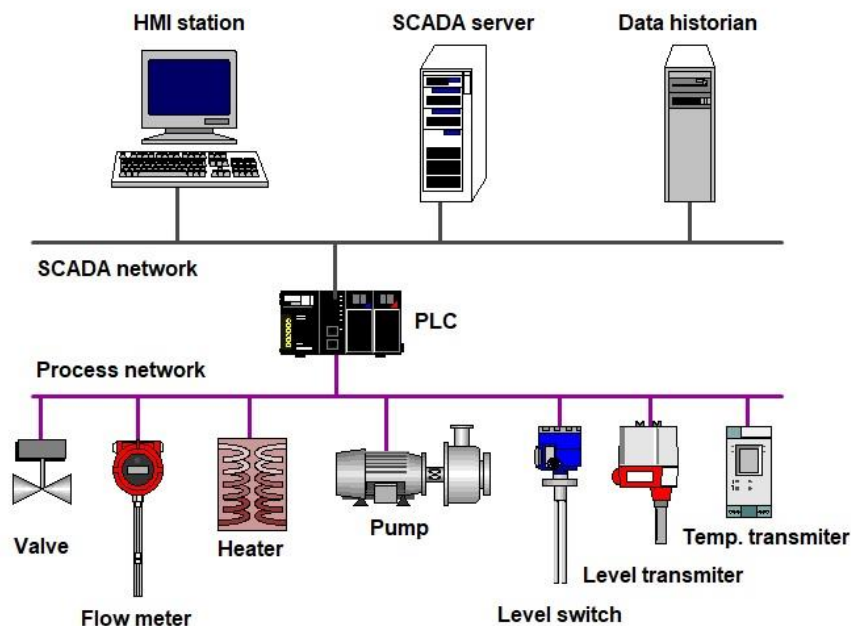
$$IndK_{PS(n)} = BrS_{(RefPS)} \cdot \frac{BrSPS(n)}{BrS_{(RefPS)}} \quad (1)$$

$$URes_{(P)} = \sum_{n=1}^{NumPS} \left(URes_{(RefPS)} \cdot IndK_{PS(n)} \right) \quad (2)$$

Podrazumijeva se da navedene kalkulacije vrijede samo ako se za sve pod-sustave primjenjuju skupovi podataka sa jednakim brojem mjerenja. Budući rast volumena podataka proporcionalno uzrokuje i porast vremenske komponente eksperimenta u generiranju rezultata uz iste modele strojnog učenja, time je dana mogućnost za kvantitativna kauzalna objašnjenja u generalizaciji sa uzorka na populaciju.

4.2 Ispitni SCADA sustav

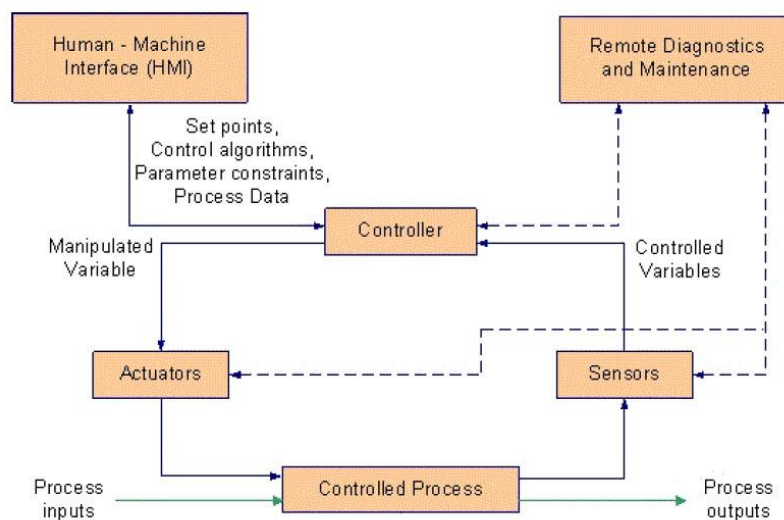
Razvijeni softverski ispitni SCADA sustav (engl. *testbed*) poslužio je kao simulator promatranog pod-procesa hidrauličke jedinice koja se tipično nalazi u već spomenutim kontinuiranim proizvodnim procesima karakterističnim za metalnu industriju i pogone kontinuiranog lijevanja, galvanizacijskih linija te hladnog i toplog valjanja. Slika 8 prikazuje komponente SCADA sustava ograničenog na pod-proces hidrauličke jedinice, a koji je proširen na elemente operacijske tehnologije, odnosno potrebne komponente senzorskih uređaja u proizvodnom polju.



Slika 8: SCADA sustav hidrauličke jedinice

Striktno gledano, te promatrajući podjelu sa slike 1, ovakvo proširenje SCADA sustava može se konceptualno klasificirati kao ICS. Međutim takva je konfiguracija tipična za ispitne SCADA sustave brojnih istraživanja na području *cyber-physical* sustava (Queiroz *et al.*, 2009; Maynard, McLaughlin and Sezer, 2018; Hahn *et al.*, 2010; Yang *et al.*, 2015; Zhang *et al.*, 2015).

Konfiguracija gore prikazanog SCADA sustava pruža sve potrebne komponente za znatno šire područje testiranja od potrebnog za predmetni rad. Time je ispitni SCADA sustav, razvijen na temelju prikazanog SCADA sustava, reduciran na ključne funkcionalnosti i elemente u odnosu na problem koje istraživanje obrađuje. Za odgovarajuće reduciranje sustava nužno je razumijevanje toka podataka i kontrolnih signala između svih elemenata te između procesne i SCADA mreže. Slika 9 pojašnjava pojednostavljen prikaz toka podataka između elemenata prikazanih na slici 8.

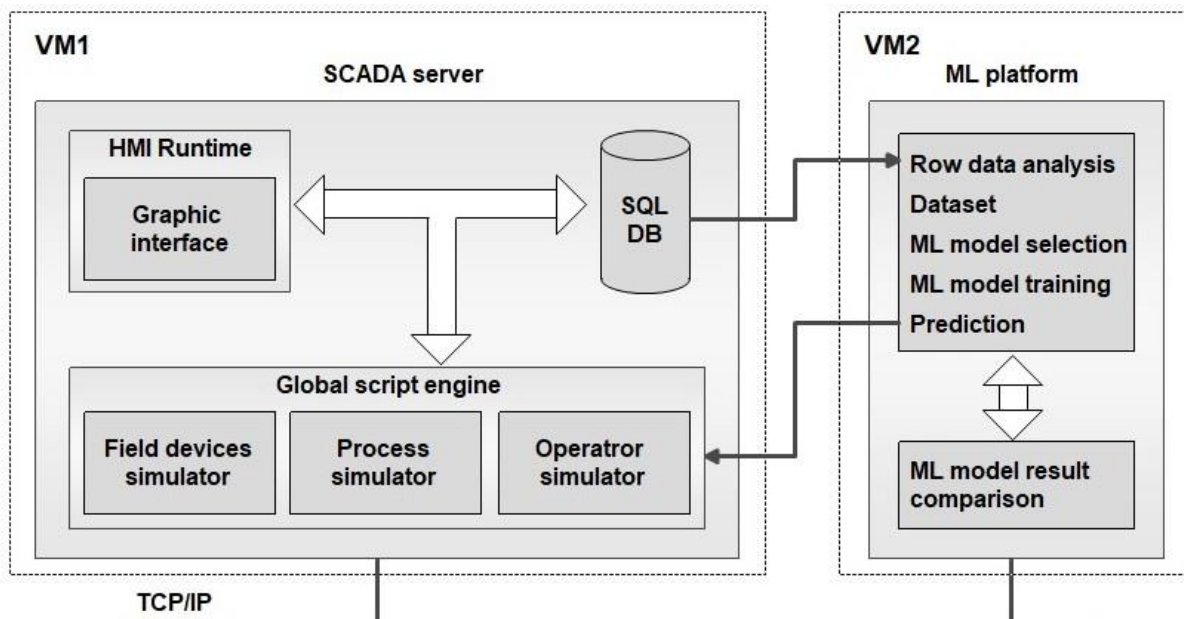


Slika 9: Tok procesnih podataka unutar SCADA sustava (Stouffer *et al.*, 2015)

Kako se da uočiti sa Slike 9., ključni element za prikupljanje procesnih podataka i generiranje kontrolnih signala je PLC kontroler. Međutim, na slici je također vidljivo kako je glavni koncentrador podataka HMI sučelje u varijanti prikazane sheme *stand-alone* konfiguracije SCADA sustava, odnosno SCADA server u distribuiranom sustavu. Time je u ispitnom SCADA sustavu, s obzirom na obim predmetnog eksperimenta, u potpunosti eliminirana procesna mreža, te PLC kontroler (engl. Programmable Logic Controller). Takvo reduciranje komponenti ima smisla jer, osim što se ulazni podaci o trenutnom stanju procesa prikupljaju na razini SCADA servera, svi su uređaji u polju izvedeni softverski. Eventualan razlog odstupanja od takvog načina može biti potreba

da se uštedi na vremenu izravnim čitanjem sa PLC-a. Međutim takvo rješenje imalo bi smisla u hardverskoj izvedbi, te bi zahtijevalo paralelni I/O driver za odgovarajući industrijski protokol korišten na procesnoj mreži, a koji bi morao biti barem jednako brz kao postojeći koji je dio SCADA servera. Međutim, i u tom slučaju dodatno dolazi u pitanje propusnost mreže u uvjetima dvostrukog prometa ako se takvo rješenje primijeni na cijelo postrojenje. Za potrebe predmetnog rada, softverska simulacija senzorskih uređaja (procesa) odvijala se na SCADA serveru, te su se vrijednosti za formiranje skupa podataka pohranjivale na *historian* serveru. Treniranje i usporedba modela strojnog učenja odvijala se na zasebnom virtualnom stroju (engl. VM – *Virtual Machine*) koji nije prikazan na slici 7 budući se ne radi o uobičajenom elementu SCADA sustava. Konačna struktura razvijenog softverskog ispitnog SCADA sustava dana je na slici 10. Takva struktura sukladno zahtjevima eksperimenta omogućava realiziranje slijedećeg:

- Simulacija procesnih vrijednosti varijabli i uzročno-posljedičnih veza (inicijalne vrijednosti senzora i aktuatora, inicijalizacija procesa, definiranje raspona vrijednosti za prihvatljive granice tolerancije stanja procesa).
- Simulacija ponašanja senzorskih uređaja u polju s obzirom na promjene stanja procesa (temperatura, pritisak, protok, razina, radni status uređaja).
- Simulacija aktivnosti poduzetih od strane različitih operatera (odvajanje dva osnovna modela predviđenih korektivnih aktivnosti poduzetih od strane operatera u cilju održavanja kontinuiteta izvođenja procesa).
- Pregled odvijanja promatranog procesa putem grafičkog sučelja.
- Prikupljanje i pohranu podataka u bazu podataka prema definiranim pravilima uzorkovanja.
- Razmjenu podataka na mreži između dva virtualna stroja: VM1 i VM2, odnosno između SCADA servera (ili HMI radne stanice) i ML platforme.
- Simulaciju platforme strojnog učenja (*ML platform*) za formiranje skupa podataka, odabir i trening modela strojnog učenja, te usporedbu rezultata različitih modela.



Slika 10: Struktura ispitnog SCADA sustava

4.3 Konfiguracija ispitnog SCADA sustava

Softverska konfiguracija ispitnog sustava uključuje operativni sustav, virtualizacijski i aplikativni softver, te programske pakete i module navedene u tablici 5.

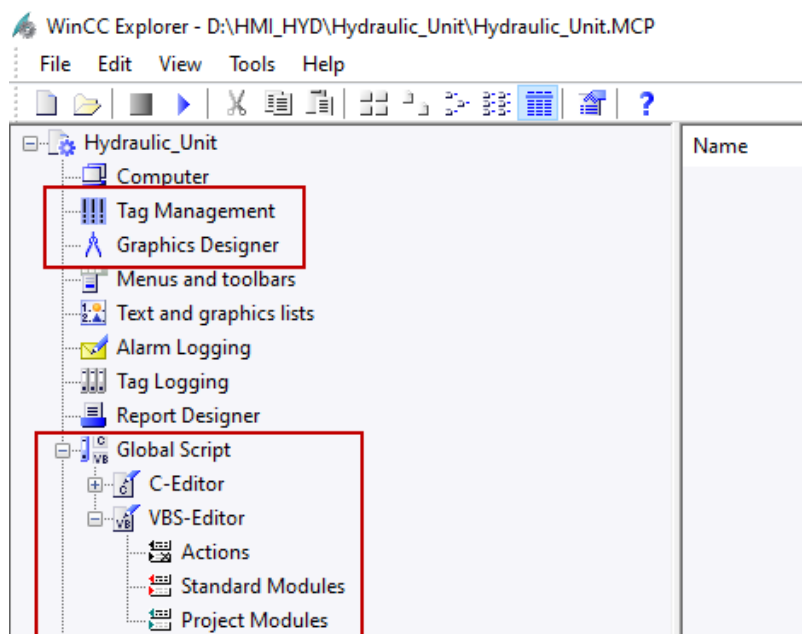
Tablica 5: Softverska konfiguracija ispitnog SCADA sustava

Softverski proizvod	Tip proizvoda	Inačica
Windows	Operativni sustav	10 - 64bit
WinCC	SCADA IDE/RT	7.5 SP1 Upd2
PostgreSQL	Baza podataka	13.1
pgAdmin	PG Administrativni alat	4.28
DbSchema	DB Administrativni alat	8.4.2
VMware Workstation Player	Type 2 Virtualizacija	16.0
Visual studio code	IDE	1.55.2
Python	Programski jezik	3.8.8 (64bit)
Anaconda	<i>Data Sci.</i> platforma	4.10.0
pandas	Python biblioteka	1.1.3
sqlalchemy	Python biblioteka	1.3.20
numpy	Python biblioteka	1.19.2
matplotlib	Python biblioteka	3.3.2
scikit-learn	Python biblioteka	0.32.2

WinCC HMI aplikacija konfigurirana je sa sljedećim komponentama razvojnog sučelja:

- *Tag management*: HMI baza signala/identifikatora (Engl. tag) identifikatora (engl. tag) za komunikaciju u realnom vremenu.
- *Graphic designer*, razvojni alat za razvoj HMI grafičkog sučelja.
- *Global script*: ANSI C i VBS editore sadržane u WinCC IDE (*Integrated Development Environment*).

Na slici 11 naznačene su gore navedene razvojne komponente korištene za razvoj SCADA servera. Osim navedenih vidljive su i komponente *Alarm Logging*, te i što je također dio funkcionalnosti razvijenog ispitnog SCADA sustava, ali se za te funkcionalnosti nisu koristili zadani alati u okviru WinCC razvojnog okružja. Razlog leži u tome što se signali alarma ne prikazuju na grafičkom sučelju, već je dovoljno da se registriraju u bazi podataka što je dio simulirane logike odvijanja procesa koja vrijednosti varijabli/identifikatora pohranjuje u bazu podataka izvan zadanog sustava. Time eksterna baza podataka simulira postojanje *Data historian* servera kao zasebnog čvora na mreži.



Slika 11: Korištene komponente WinCC razvojnog okružja

Komponenta *Tag management* predstavljena slikom 12, zadužena je za pohranu konfiguracije svih internih i ulazno-izlaznih identifikatora/signala. Na desnom panelu vidljiv je dio identifikatora korištenih u simulaciji procesa koji pripadaju „hyd“ grupi

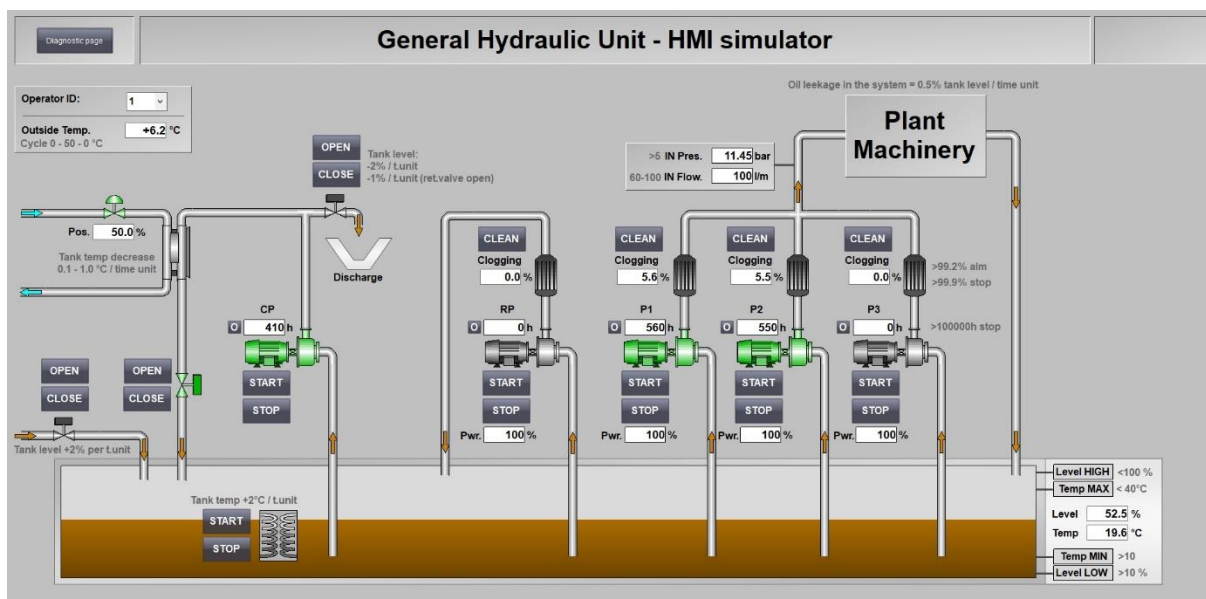
internih identifikatora. Ostale grupe internih identifikatora vidljive na lijevom panelu sadrže raspoložive sistemske identifikatore. Takvom konfiguracijom RT baze identifikatora postigla se mogućnost upisivanja vrijednosti u sve identifikatore što inače ne bi bilo moguće sa su konfigurirani kao ulazno-izlazni identifikatori u slučaju kada postoji hardverski PLC kao zasebni čvor na mreži. To u domeni ovog eksperimenta nije bilo potrebno iz već prethodno navedenih razloga tj., uloge SCADA servera kao primarnog izvora podataka za potrebe implementacije strojnog učenja u predikciji korektivnih mjera.

The screenshot shows the 'Tag Management' window in WinCC Configuration Studio. On the left, a tree view shows the project structure with 'Internal tags' expanded to show a folder named 'hyd'. On the right, a table lists the tags for 'hyd' with columns for Name, Data type, and Length.

	Name	Data type	Length
55	STAT_FLOW_IN	Unsigned 32-bit value	4
56	STAT_HEATER_RUN	Binary Tag	1
57	STAT_INLET_VALVE_OPEN	Binary Tag	1
58	STAT_OPERATOR_ID	Unsigned 32-bit value	4
59	STAT_OUTSIDE_TEMP	Floating-point number 32-bit IEEE 754	4
60	STAT_P1_FILTER_CLOG	Floating-point number 32-bit IEEE 754	4
61	STAT_P1_RUN	Binary Tag	1
62	STAT_P1_WORK_HRS	Unsigned 32-bit value	4
63	STAT_P2_FILTER_CLOG	Floating-point number 32-bit IEEE 754	4
64	STAT_P2_RUN	Binary Tag	1
65	STAT_P2_WORK_HRS	Unsigned 32-bit value	4
66	STAT_P3_FILTER_CLOG	Floating-point number 32-bit IEEE 754	4
67	STAT_P3_RUN	Binary Tag	1

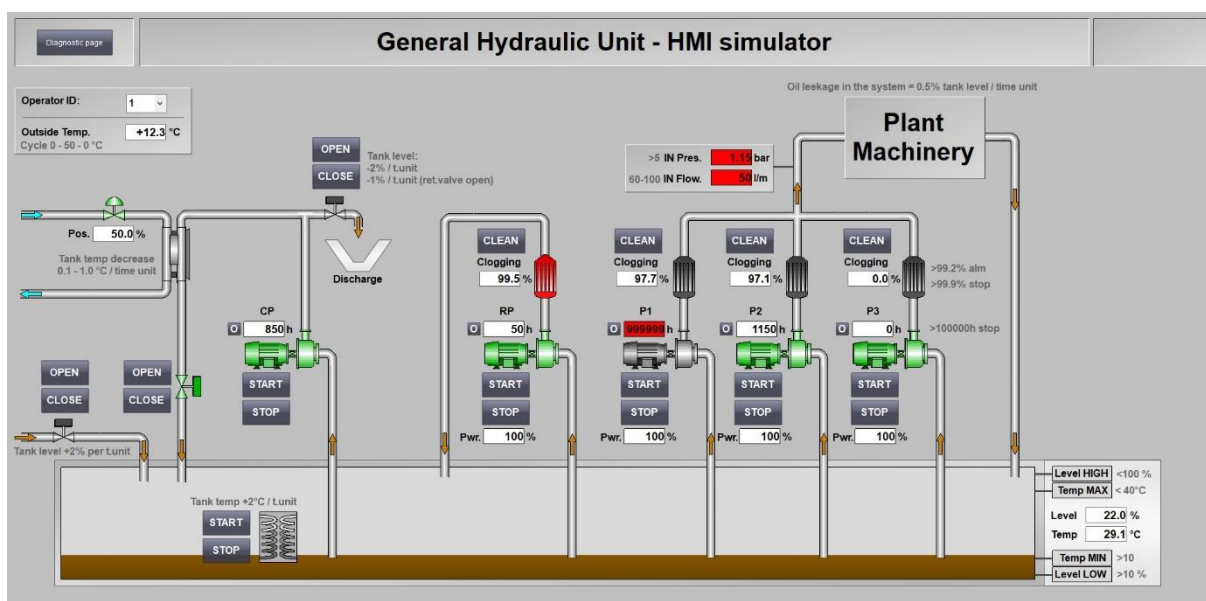
Slika 12: Tag management komponenta WinCC razvojnog okruženja

Razvojni alat *Graphic Designer* korišten je za razvoj grafičke komponente HMI sučelja prema operateru koje pruža sve raspoložive informacije o stanju procesa tj., simulirane vrijednosti signala uređaja sa polja. Osnovne funkcionalnosti grafičkog sučelja omogućavaju praćenje i kontrolu procesa u normalnom režimu rada, kao i prihvat mogućih kontrolnih aktivnosti operatera putem elemenata sučelja. Dodatna opcija sučelja je izbor operatera. S obzirom da se za potrebe generiranja potrebnog skupa podataka simuliraju i reakcije operatera u odnosu na aktualna stanja procesa, sučelje omogućuje izbor jednog od dva simulirana operatera, ili isključivanje oba operatera i preuzimanje manualne kontrole. Sučelje na slici 13 prikazuje simulaciju hidrauličke jedinice u normalnom režimu rada.



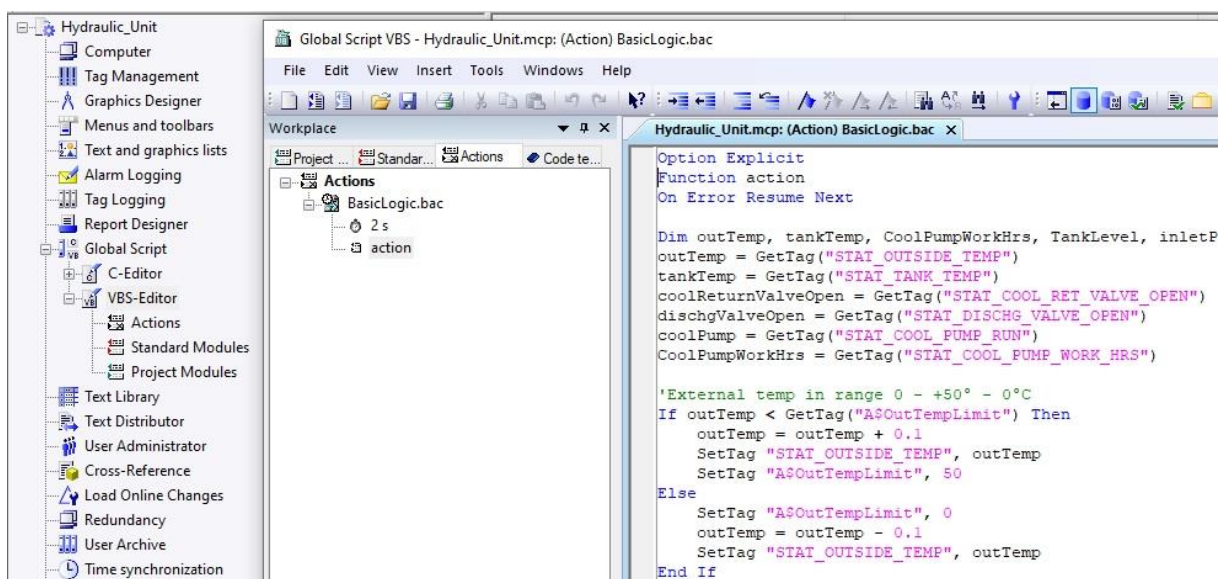
Slika 13: Hidraulička jedinica u normalnom režimu rada

Aktivni uređaji su pumpe P1 i P2, te pumpa sustava hlađenja (CP), proporcionalni ventil rashladnog sustava u poziciji 50% otvorenosti, te ulazni ventil rashladnog sustava. Razina ulja i temperatura u tanku su u granicama normale, kao i ulazni pritisak i protok prema strojevima u pogonu (*engl. plant machinery*). Sukladno stanju sustava, nema vizualizacije aktivnih alarma. Slika 14 prikazuje hidrauličku jedinicu u režimu rada koji je izašao izvan granica tolerancije u vrijednostima ulaznog pritiska i protoka prema strojevima u pogonu, te radnog vijeka (broj radnih sati) pumpe P1, kao i propusnosti filtera recirkulacijske pumpe RP. Sukladno tome aktivirani su pripadni alarmni signali odgovarajuće vizualizirani crvenom bojom.



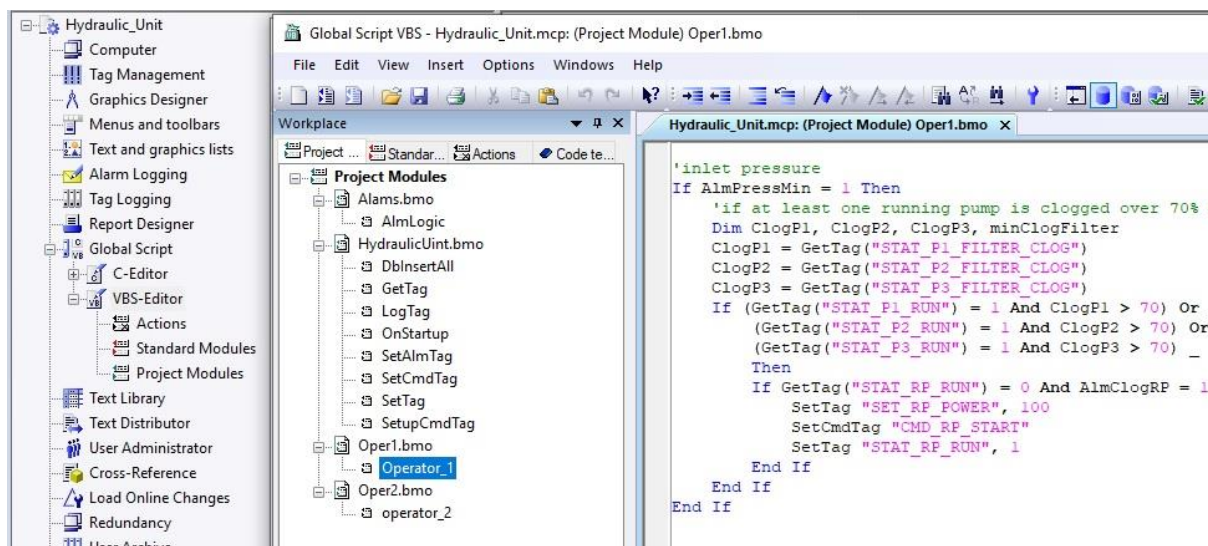
Slika 14: Hidraulička jedinica u režimu rada izvan granica tolerancije

Kako je već prikazano na slici 10, cjelokupna simulacija procesa i uređaja u polju, te aktivnosti operatera razvijena je u globalnim skriptama (VBS editor), te se u realnom vremenu odvija unutra samog SCADA servera, tj., njegove RT komponente. S obzirom na izvršavanje i namjenu, globalne skripte korištene za simulaciju podijeljene su u VBS editoru podijeljene na akcije i projektne module. Osnovna razlika je u tome što se u realnom vremenu akcije pokreću ciklički u zadanom vremenskom ciklusu ili promjenom vrijednosti praćenog identifikatora, dok projektni moduli sadrže funkcije koje se izvršavaju na vanjski poziv iz grafičkog sučelja ili akcija. Slika 15 prikazuje segment modula osnovne logike.



Slika 15: Programski modul *Actions* VBS editora

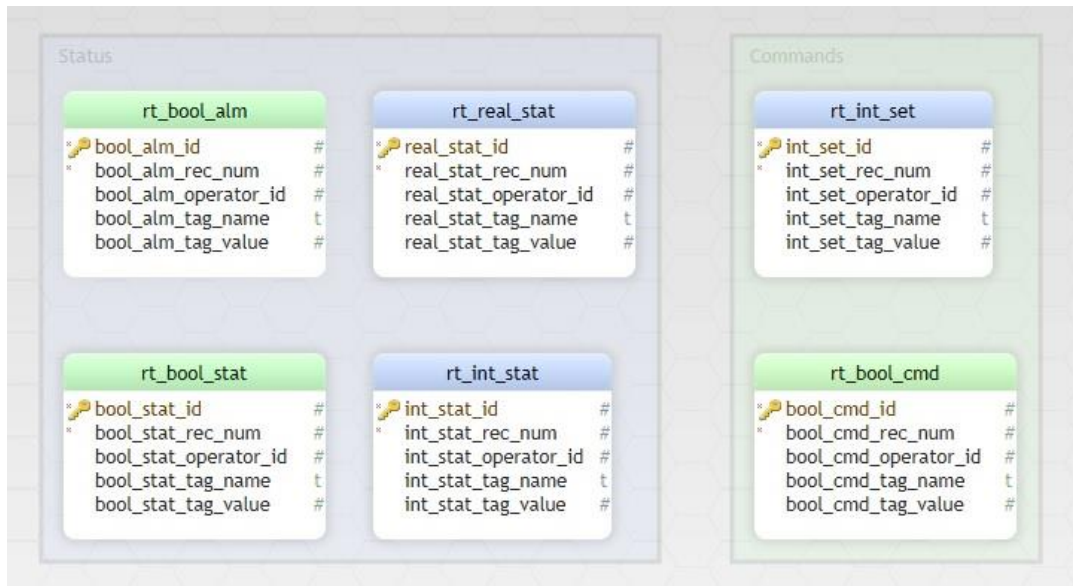
Gore prikazan modul *Basic logic* sadrži funkciju koja se odvija ciklički u intervalima od dvije sekunde i definira utjecaj okoline (vanjska temperatura), početne postavke procesa, te ponašanje uređaja u sastavu hidrauličke jedinice. Nastavno na osnovnu logiku, na slici 16 prikazani su razvijeni projektni moduli simulacije alarma, aktivnosti za dva različita operatera, pomoćne funkcije ključne za odvijanje simulacije, te funkcije *DbInsertAll* kojom se u eksternu bazu upisuju kumulativni podaci za svaki ciklus prikupljanja.



Slika 16: Projektni moduli simulacije

Budući se prikupljeni podaci razlikuju u vrsti i ulozi u procesu, time su u bazi podijeljeni u šest tablica. Logički gledano, postoje dvije skupine koje sve signale/identifikatore korištene za simulaciju dijele na statuse i naredbe. Pod statusima se, iz aspekta ovog ispitnog SCADA sustava, podrazumijeva svaki ulazni podatak iz uređaja u polju, dok se naredbom smatra svaki izlazni signal/identifikator koji vrijednost šalje prema uređaju. Kako je ispitni sustav izveden softverski u potpunosti sa internim identifikatorima, time se eventualna interna logika promatra na isti način. Navedeno se ne odnosi na signale alarma, što ne mora uvijek biti konkretan signal iz polja, već može biti rezultat interne logike. S obzirom na tip podataka, obje skupine mogu sadržavati digitalne i analogne vrijednosti. U tom smislu, izvršena je određena prilagodba digitalnih signala koji se na razini simulacije generiraju kao analogna vrijednost 1 ili 0, te se takvi također upisuju u eksternu bazu gdje su definirani kao cjelobrojni tip podatka (*integer*). Navedeno suštinski ne mijenja mogućnosti daljnje analize i obrade na razini modela strojnog učenja, jer se i dalje mogu klasificirati, ali može biti korisno u slučajevima kada je lakše cijeli skup podataka promatrati kao numeričke vrijednosti.

Slika 17 prikazuje shemu baze podataka u kojoj se pohranjuju vrijednosti svih identifikatora koji sudjeluju u simulaciji. Tablice su podijeljene u spomenute logičke skupine, te su prema tipu podataka analogne vrijednosti u plavim a digitalne u zelenim tablicama.



Slika 17: Shema eksterne baze podataka

Iz sheme je vidljivo kako ne postoje veze između tablica, tj., nema vanjskih ključeva (engl. *foreign key*). U realnom sustavu takva shema predstavljala bi opasnost po integritet baze. Međutim, u konkretnom slučaju ova shema služi isključivo za pohranu vrijednosti signala iz kojih se generira skup podataka za daljnju analizu. U tom smislu, jedino je značajna sinkronizacija između podataka u različitim tablicama s obzirom na vremenski slijed u kojem su bili prikupljeni i upisani. U tu svrhu koristi se broj zapisa (*rec_num*) koji je prisutan kao atribut u svakoj tablici. Budući se taj broj generira od strane prethodno spomenute funkcije *DbInsertAll*, a koja u danom trenutku vrši istovremeni upis svih vrijednosti identifikatora u sve tablice, time je rizik od nepravilnih zapisa isključen. Sve će vrijednosti biti upisane, ili će upis izostati za sve tablice.

4.4 Parametri procesa

Kako je, iz prethodno navedenih razloga, za promatrani pod-proces kontinuiranog proizvodnog procesa odabrana hidraulička jedinica, time se parametri procesa odnose isključivo na rad iste, te nisu povezani sa parametrima ostalih pod-sustava cjelokupnog proizvodnog procesa na razini pogona. U tom je smislu, hidraulička jedinica pod-sustav koji je relativno neovisan. Osnovne međuovisnosti leže u dotoku rashladne tekućine, ulja, te gubitka istog ovisno o stanju strojeva koje opslužuje. Od vanjskih utjecaja koji su u određenoj mjeri neovisni od procesa, značajna je temperatura okoline. Od navedenih međuovisnosti, dotok rashladne tekućine i ulja smatra se raspoloživim u svakom trenutku, te se regulira isključivo odlukom operatera (kontrola ventila). Vanjska

temperatura mijenja se dinamički i potpuno je neovisna o procesu. Tablica 6 daje ključne parametre za odvijanje i kontrolu promatranog procesa u jedinici vremena (j.v.), kao i referentne vrijednosti za održavanje redovnog režima rada hidrauličke jedinice u granicama tolerancije.

Tablica 6: Parametri odvijanja i kontrole procesa

Naziv	Ref. vrijednost	Utjecaj na proces
Razina u tanku	10-100 %	Gubitak u radu je 0,5% razine u j.v.
Temp. u tanku	10-40 °C	+5°C offset u odnosu na vanjsku (bez utjecaja hlađenja/grijanja).
Dovodni ventil (LV)	0-1	Povećava razinu tanka za 2% u j.v.
Regulacijski ventil (RV)	0-100 %	Spušta temperaturu u tanku u rasponu 0,1-1,0 °C u j.v. proporcionalno otvorenosti.
Ispusni ventil (DV)	0-1	Spušta razinu u tanku za 2% u v.j, ili za 1% ako je otvoren ventil hlađenja (CV)
Grijač (HT)	0-1	Podiže temperaturu u tanku za 2°C u j.v.
Začepljenje filtera (P1, P2, P3)	0-100 %	Raste 0,1% u j.v. (1% smanjuje izlazni pritisak 0.05 bar).
Radni sati	0-100000 h	Povećavaju se u koracima od 10 h/j.v., te prekid rada uređaja po doseg limitu.
Snaga pumpe (P1, P2, P3)	0-100 %	Proporcionalni utjecaj na izlazni protok 0-50 l/min
Radni status pumpe (P1, P2, P3)	0-1	Svaka pumpa utječe na izlazni protok do najviše 50 l/min, te postiže najviši izlazni pritisak od 6 bara.
Radni status pumpe (RP)	0-1	Recirkulacijska grana smanjuje razinu začepljenosti filtera pumpi P1, P2, P3 za najviše 0,5% u j.v. (za pumpe u radu)
Začepljenje filtera (RP)	0-100 %	Raste 0,1% u j.v. (100% zaustavlja RP)

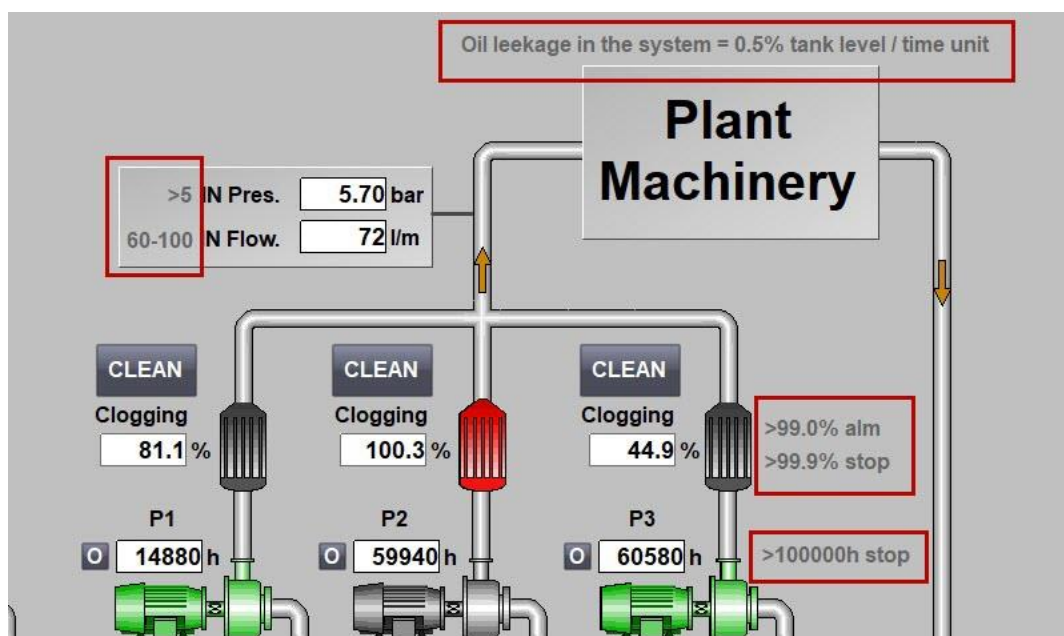
Parametri navedeni u gornjoj tablici određuju odvijanje procesa čija se učinkovitost mjeri izlaznim parametrima u tablici 7. Održavanje parametara u granicama referentnih

vrijednosti osigurava se kontinuirani rad hidrauličke jedinice čime se postiže analogija sa kontinuiranim proizvodnim procesima što je u cilju predmetnog rada.

Tablica 7: Izlazni parametri procesa

Naziv	Ref. vrijednost	Utjecaj na proces
Izlazni pritisak	>5 bar	Pad pritiska ispod dozvoljene razine smatra se kritičnim za odvijanje procesa.
Izlazni protok	60 – 100 l/min	Protok izvan granica danih vrijednosti smatra se kritičnim i potencijalno prekida kontinuitet procesa.

Za lakše praćenje stanja procesa, te razumijevanje parametara odvijanja i kontrole procesa iz perspektive HMI operatera, ključne referentne vrijednosti prikazane su na grafičkom sučelju hidrauličke jedinice. Slika 18 prikazuje detalj ulazne grane sa naznačenim pripadnim vrijednostima koje osim na parametre odvijanja procesa, upozoravaju i na granične vrijednosti za kritične situacije (alarm začepljenja filtera i posljedična blokada pumpe).



Slika 18: Detalj ulazne grane sa parametrima

4.5 Simulacija promatranog procesa

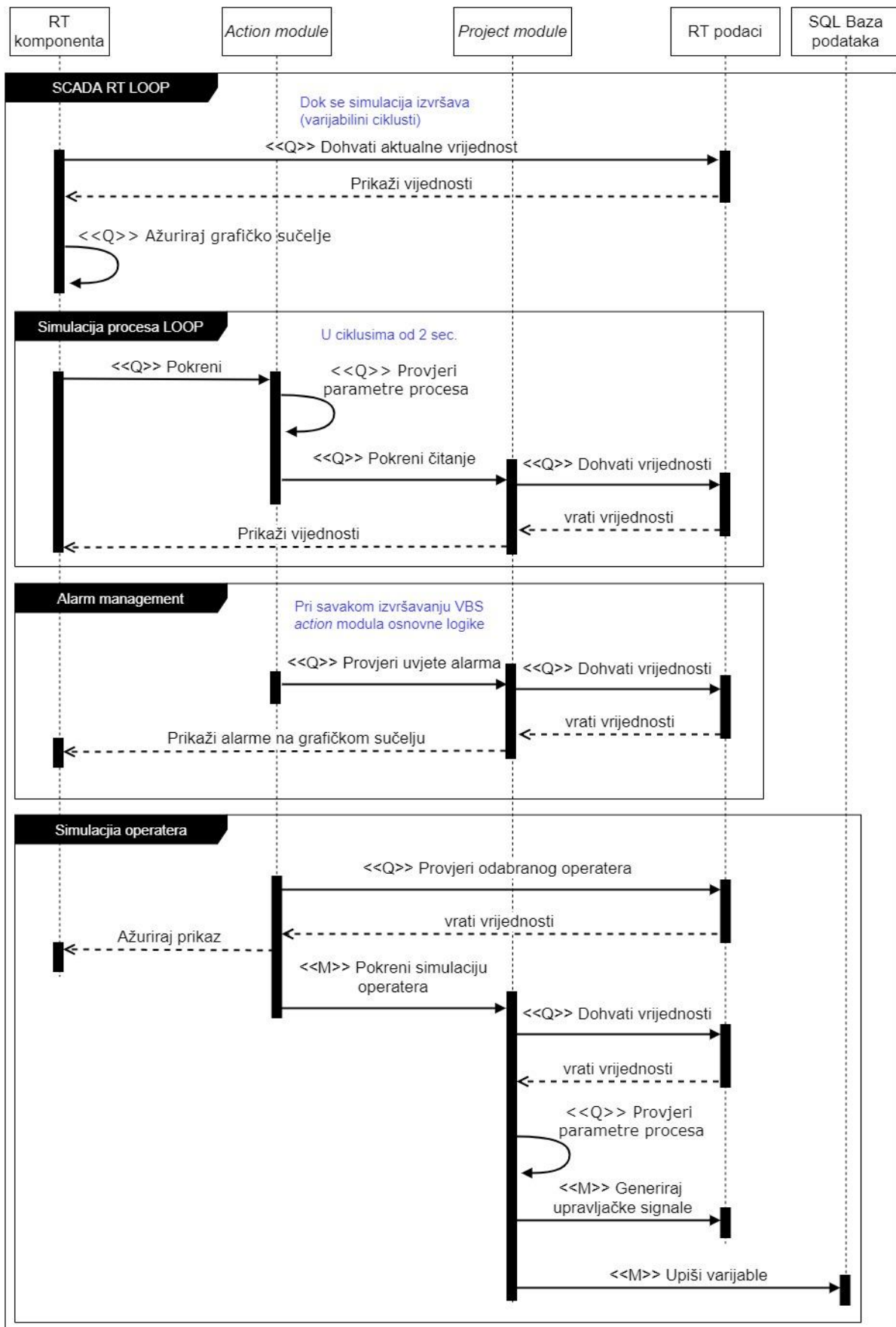
Kako je u prethodnom tekstu već spomenuto, simulacija procesa odnosi se na postizanje kontinuiranog rada hidrauličke jedinice tipične konfiguracije za kontinuirane procese u metalnoj industriji. Cjelokupna simulacija razvijena je korištenjem WinCC SCADA skupa razvojnih alata, te se sukladno tome izvršava u WinCC RT komponenti unutar HMI aplikacije.

Slika 19 prikazuje *sequence* dijagram simulacije procesa iz kojeg se vidi interakcija u vremenskom slijedu između navedenih osnovnih komponenti ispitnog SCADA sustava. HMI aplikacija sa grafičkim sučeljem pokreće se u RT komponenti koja za vrijeme rada kontinuirano ažurira vrijednosti identifikatora u realnom vremenu sa varijabilnim ciklusima. Navedeni ciklusi spadaju u sistemske postavke WinCC RT komponente, te se primarno odnose na kontinuirano ažuriranje vrijednosti u domeni SCADA RT loop sekvence, i nisu dio postavki simulacije. Dva osnovna VBS modula (1) *Action* i (2) *Project module* sadrže algoritme simulacije procesa i ponašanja operatera, te parametre odvijanja procesa. Simulacija se odvija na način da RT komponenta prilikom inicijalizacije dohvaća trenutne (inicijalno zadane) vrijednosti procesa kojima ažurira grafičko HMI sučelje, te ciklički pokreće izvršenje algoritma simulacije procesa sadržanog u sekvenci *BasicLogic* (*Action* modul). Algoritam simulira odvijanje procesa primjenom zadanih parametara kontrole procesa prema tablici 6, na način da budu zadovoljeni izlazni parametri iz tablice 7. U jednom ciklusu izvršenja u trajanju od 2 sekunde, algoritam izvršava slijedeći niz aktivnosti:

- Ažurira vrijednosti identifikatora prema parametrima procesa.
- Provjerava kritične vrijednosti sustava te generira prikaza alarma na grafičkom sučelju animiranjem elemenata prikaza procesa.
- Aktivira izvršenje algoritma simulacije zadanog operatera koji prema potrebi generira upravljačke signale u simulaciji sustava.

Algoritam simulacije aktivnosti operatera dio je *Project* modula koji sadrži dijelove simulatora koji se pozivaju prema potrebi, te izvršava slijedeći niz aktivnosti:

- Dohvaća aktualne vrijednosti procesa.
- Generira upravljačke signale prema vlastitoj logici rada (kontrole procesa).
- Aktivira upis vrijednosti varijabli (skup vrijednosti identifikatora) u bazu podataka.



Slika 19: Sequence dijagram simulacije procesa

Sve vrijednosti procesa u realnom vremenu definirane su i razmjenjuju se putem identifikatora koji u osnovi predstavljaju podatkovnu strukturu koja sadrži vrijednost ulazno-izlaznih signala procesa, te se u tom smislu mogu promatrati kao signali, tj., kao varijable koje se upisuju u eksternu bazu nakon čega formiraju skup podataka sa istoimenim nazivima varijabli. Popis identifikatora definiran je u *Tag management* komponenti WinCC razvojnog sučelja kako je prethodno prikazano na slici 12. Vrijednosti identifikatora uključenih u *Tag management* automatski se ažuriraju tijekom rada RT komponente kao aktualne vrijednosti sustava. U realnom SCADA sustavu ažuriranje aktualnih vrijednosti dolazi iz I/O servera koji izravno komunicira sa PLC-om. Bući se, kako je već ranije rečeno, u ovom slučaju radi o softverskoj simulaciji uređaja i procesa, time se vrijednosti generiraju iz opisanih VBS modula. Dodatna prednost takvog načina generiranja vrijednosti leži u dostupnosti. U ovoj izvedbi ne postoji mogućnost da se bilo koja vrijednost ne generira, tj., da u konačnici postoji varijabla u skupu podataka bez vrijednosti. Radi lakšeg razumijevanja primijenjen je *naming convention*, te su identifikatori podijeljeni u slijedeće skupine prema ulozi u procesu:

- Izlazni - kontrolni identifikatori, kojima se generira digitalni upravljački signal prema sustavu (naziv započinje prefiksom „CMD“).
- Izlazni - kontrolni identifikatori, kojima se generira analogni upravljački signal prema sustavu (naziv započinje prefiksom „SET“).
- Ulazni - statusni identifikatori, koji prikazuju digitalnu ili analognu vrijednost određenog elementarnog stanja/statusa elementa procesa (naziv započinje prefiksom „STAT“).
- Ulazni – identifikatori alarma, koji prikazuju digitalni signal alarma iz određenog elementa sustava, ili rezultat izlaska iz tolerancije parametara procesa (naziv započinje prefiksom „ALM“).

Tablica 8 sadrži identifikatore definirane u *Tag management*-u koji u konačnici predstavljaju varijable skupa podataka za kasniju analizu i implementaciju modela strojnog učenja. Simulacija procesa sadrži ukupno 74 identifikatora, od kojih se 33 odnose na digitalne i analogne izlazne kontrolne identifikatore, te preostalih 41 na digitalne i analogne ulazne identifikatore uključujući i identifikatore alarma. Pri razmatranju daljnjih koraka u cilju ostvarenja predikcije korektivnih mjera, navedena

33 identifikatora treba promatrati kao zavisne varijable čije vrijednosti treba predvidjeti na temelju međuovisnosti sa statusnim varijablama.

Tablica 8: Identifikatori korišteni u simulaciji procesa

Naziv identifikatora	Tip	Naziv identifikatora	Tip
ALM_COOL_PUMP_RUN_COL_RET_VALVE_CLS	Bin	CMD_P3_START	Bin
ALM_COOL_PUMP_WORK_HRS_MAX	Bin	CMD_P3_STOP	Bin
ALM_INLET_FLOW_HIGH	Bin	CMD_P3_WORK_HRS_RESET	Bin
ALM_INLET_FLOW_LOW	Bin	CMD_RP_FILTER_CLEAN	Bin
ALM_INLET_PRESSURE_MIN	Bin	CMD_RP_START	Bin
ALM_P1_FILTER_CLOGGED	Bin	CMD_RP_STOP	Bin
ALM_P1_WORK_HRS_MAX	Bin	CMD_RP_WORK_HRS_RESET	Bin
ALM_P2_FILTER_CLOGGED	Bin	SET_COOL_PUMP_POWER	Int
ALM_P2_WORK_HRS_MAX	Bin	SET_HEAT_EXCHG_VALVE_OPEN	Int
ALM_P3_FILTER_CLOGGED	Bin	SET_P1_POWER	Int
ALM_P3_WORK_HRS_MAX	Bin	SET_P2_POWER	Int
ALM_RP_FILTER_CLOGGED	Bin	SET_P3_POWER	Int
ALM_RP_WORK_HRS_MAX	Bin	SET_RP_POWER	Int
ALM_TANK_LEVEL_HIGH	Bin	STAT_COOL_PUMP_RUN	Bin
ALM_TANK_LEVEL_LOW	Bin	STAT_COOL_PUMP_WORK_HRS	Int
ALM_TANK_TEMP_MAX	Bin	STAT_COOL_RET_VALVE_OPEN	Bin
ALM_TANK_TEMP_MIN	Bin	STAT_DISCHG_VALVE_OPEN	Bin
CMD_COOL_PUMP_START	Bin	STAT_FLOW_IN	Int
CMD_COOL_PUMP_STOP	Bin	STAT_HEATER_RUN	Bin
CMD_COOL_PUMP_WORK_HRS_RESET	Bin	STAT_INLET_VALVE_OPEN	Bin
CMD_COOL_RET_VALVE_CLOSE	Bin	STAT_OPERATOR_ID	Int
CMD_COOL_RET_VALVE_OPEN	Bin	STAT_OUTSIDE_TEMP	Flt
CMD_DISCHG_VALVE_CLOSE	Bin	STAT_P1_FILTER_CLOG	Flt
CMD_DISCHG_VALVE_OPEN	Bin	STAT_P1_RUN	Bin
CMD_HEATER_START	Bin	STAT_P1_WORK_HRS	Int
CMD_HEATER_STOP	Bin	STAT_P2_FILTER_CLOG	Flt
CMD_INLET_VALVE_CLOSE	Bin	STAT_P2_RUN	Bin
CMD_INLET_VALVE_OPEN	Bin	STAT_P2_WORK_HRS	Int
CMD_P1_FILTER_CLEAN	Bin	STAT_P3_FILTER_CLOG	Flt

CMD_P1_START	Bin	STAT_P3_RUN	Bin
CMD_P1_STOP	Bin	STAT_P3_WORK_HRS	Int
CMD_P1_WORK_HRS_RESET	Bin	STAT_PRESSURE_IN	Flt
CMD_P2_FILTER_CLEAN	Bin	STAT_RP_FILTER_CLOG	Flt
CMD_P2_START	Bin	STAT_RP_RUN	Bin
CMD_P2_STOP	Bin	STAT_RP_WORK_HRS	Int
CMD_P2_WORK_HRS_RESET	Bin	STAT_TANK_LEVEL	Flt
CMD_P3_FILTER_CLEAN	Bin	STAT_TANK_TEMP	Flt

Identifikatori iz gornje tablice korišteni su u svim dijelovima kôda za interakciju sa promatranim procesom u realnom vremenu. Algoritam simulacije procesa odvija se u sekvenci *BasicLogic* (*Action* modul) prema prethodno opisanom načinu, te kako je prikazano u *sequence* dijagramu na slici 18.

4.6 Korektivne mjere od strane operatera

Imajući u vidu primarni cilj predikcije korektivnih mjera u stabilizaciji sustava promatranog procesa, te sekundarni cilj mjerenja učinkovitosti poduzetih korektivnih mjera od strane različitih operatera sa ciljem izdvajanja „najiskusnijeg“ u predikciji mjera, očigledna je ključna uloga simulacije ponašanja operatera, odnosno poduzetih korektivnih mjera. Nužno je da u tom smislu postoje barem dva različita operatera koji kontroli procesa pristupaju na različite načine, tj., za isto stanje procesa u određenom trenutku donose različite odluke sa istim ciljem održavanja kontinuiteta odvijanja procesa u granicama tolerancije prema zadanim parametrima. Pri tome su za jednostavniji pristup daljnjoj analizi skupa podataka definirana dva operatera sa očiglednim razlikama u postupcima s obzirom na postignutu učinkovitost u kontroli procesa. Za bolje razumijevanje modela ponašanja operatera 1, dio aktivnosti vezano za kontrolu razine i temperature u tanku prikazan je kroz slijedeće parove okidača (o) u vidu stanja procesa, i pripadnih kontrolnih mjera (m) kao odgovor na okidač.

(o1) Aktivan alarm visoke razine u tanku; Dovodni ventil (LV) otvoren.

(m1) Zatvori dovodni ventil (LV).

(o2) Aktiviran alarm niske razine u tanku; Dovodni ventil (LV) zatvoren.

(m2) Otvori dovodni ventil (LV).

- (o3) Aktiviran alarm visoke temperature u tanku; Pumpa hlađenja (CP) isključena.
- (m3) Otvori regulacijski ventil (RV) na 50%; Otvori ventil sustava hlađenja (CV); Pokreni pumpu hlađenja (CP).
- (o4) Aktiviran alarm visoke temperature u tanku; Grijač u tanku (HT) uključen.
- (m4) Isključi grijač (HT).
- (o5) Aktiviran alarm niske temperature u tanku; Pumpa hlađenja (CP) uključena.
- (m5) Zatvori regulacijski ventil (RV); Zatvori ventil sustava hlađenja (CV); Zaustavi pumpu hlađenja (CP).
- (o6) Aktiviran alarm niske temperature u tanku; Grijač u tanku (HT) isključen.
- (m6) uključi grijač (HT).

Analogno gornjim aktivnostima operatera 1, slijedeći parovi okidača i kontrolnih mjera vrijede za operatera 2 u istom segmentu kontrole razine i temperature u tanku.

- (o1) Razina u tanku > 85%; Dovodni ventil (LV) otvoren.
- (m1) Zatvori dovodni ventil (LV).
- (o2) Razina u tanku < 15%; Dovodni ventil (LV) zatvoren.
- (m2) Otvori dovodni ventil (LV).
- (o3) Temperatura u tanku > 35°C; Pumpa hlađenja (CP) isključena.
- (m3) Otvori regulacijski ventil (RV) na 100%; Otvori ventil sustava hlađenja (CV); Pokreni pumpu hlađenja (CP).
- (o4) Temperatura u tanku < 20°C; Grijač u tanku (HT) uključen.
- (m4) Isključi grijač (HT).
- (o5) Temperatura u tanku < 30°C; Pumpa hlađenja (CP) uključena.
- (m5) Zatvori regulacijski ventil (RV); Zatvori ventil sustava hlađenja (CV); Zaustavi pumpu hlađenja (CP).
- (o6) Temperatura u tanku < 15°C; Grijač u tanku (HT) isključen.
- (m6) uključi grijač (HT).

Usporedbom gore prikazana dva segmenta modela ponašanja operatera u kontroli procesa vidljiva je temeljna razlika u simuliranim aktivnostima. Operater 1 reagira isključivo na aktivaciju alarma u sustavu promatranog procesa, dok operater 2 primjenjuje preventivne mjere u kontroli kako bi sustav održao u granicama tolerancije bez ulaska u područje kritičnih vrijednosti. Premda se u oba slučaja održava kontinuitet odvijanja procesa čime je ostvaren primarni cilj simulacije, u slučaju aktivnosti poduzetih od strane operatera 1 značajno je povećan utrošak resursa u donosu na operatera 2. Pri kontroli temperature u tanku, operater 1 uključuje grijač tek po aktiviranju alarma niske temperature, te grijač ostaje uključen do aktivacije alarma visoke temperature, čime se dodatno troše resursi koje koristi sustav za hlađenje koji se posljedično mora pokrenuti. U slučaju kontrolnih mjera poduzetih od strane operatera 2, održavanje parametara procesa u zadanim granicama tolerancije postiže se praćenjem vrijednosti temperature i isključivanjem grijača te sustava hlađenja u trenutku postizanja zadovoljavajućeg stanja sustava praćenog procesa. Budući se isti princip primjenjuje na kontrolu ostalih parametara procesa, u konačnici postoji bitna razlika u učinkovitosti između operatera. Cjelokupan algoritam simulacije oba operatera sadržan je unutar *Project* modula, te se poziva iz *Action* modula kako je prethodno prikazano *sequence* dijagramom. Slika 20 prikazuje dio algoritma simulacije operatera 2 u korektivnim mjerama prema izlaznim parametrima procesa (tablica 6) uz postizanje optimalne snage tj., potrošnje pumpi P1,P2,P3 (*Soft control*).

```

'inlet flow - soft control (keep flow between 70 and 80)
If statInletFlow < 70 Then
  If statRunP1 = 1 And setPowerP1 <= 90 Then SetTag "SET_P1_POWER", (setPowerP1 + setPowerP1/10) End If
  If statRunP2 = 1 And setPowerP2 <= 90 Then SetTag "SET_P2_POWER", (setPowerP2 + setPowerP2/10) End If
  If statRunP3 = 1 And setPowerP3 <= 90 Then SetTag "SET_P3_POWER", (setPowerP3 + setPowerP3/10) End If
End If
If statInletFlow >= 80 And (statRunP1 + statRunP2 + statRunP3) > 1 Then
  If statRunP1 = 1 Then SetTag "SET_P1_POWER", (setPowerP1 - setPowerP1/10) End If
  If statRunP2 = 1 Then SetTag "SET_P2_POWER", (setPowerP2 - setPowerP2/10) End If
  If statRunP3 = 1 Then SetTag "SET_P3_POWER", (setPowerP3 - setPowerP3/10) End If
End If

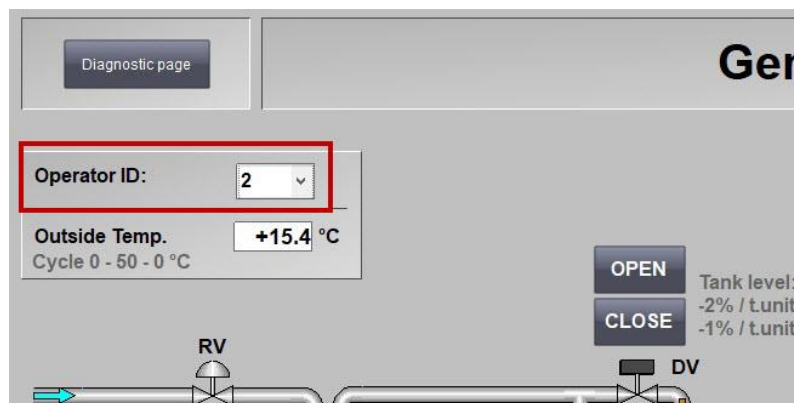
```

Slika 20: Algoritam simulacije operatera 2: *Soft control*

Takvo definiranje različitih modela ponašanja operatera u konačnici omogućuje mjerenje učinkovitosti pojedinog operatera prema parametrima utroška resursa što se može izračunati iz kumulativne snage pumpi u radu (P1,P2,P3), kontinuiranog rada uređaja (grijač, sustav hlađenja), te prema ukupnom broju aktiviranih alarma u odnosu

na broj opažanja (upisa u bazu). Navedeno neće dati preciznu ocjenu učinkovitosti s obzirom na nesinkrono uzorkovanje, ali može biti dovoljno za grubo klasificiranje.

S obzirom da je riječ o praćenju kontinuiranog procesa, time se podrazumijeva da se HMI operateri izmjenjuju u hodu tijekom odvijanja procesa, što je omogućeno je preko grafičkog sučelja kako je prikazano na slici 21.



Slika 21: HMI sučelje: selekcija operatera

4.7 Skup podataka (dataset) i podatkovni okvir (dataframe)

Iz prethodnog teksta poznato je kako skup podataka čine vrijednosti koje predstavljaju simulaciju signala sa senzorskih uređaja u polju proizvodnog pogona, te kako se sve aktualne vrijednosti kumulativno upisuju u bazu prilikom izvršenja svake od upravljačkih aktivnosti kako je to prikazano na *sequence* dijagramu (slika 19). Sukladno tome, svi atributi (zavisne i nezavisne varijable) dani su tablicom 18, pri čemu su nezavisne varijable svi identifikatori koji predstavljaju statute, tj., dolaze iz simuliranog procesa (identifikatori sa prefiksom STAT) i alarmi (prefiks ALM), dok su zavisne varijable svi oni identifikatori koji djeluju prema procesu kao korektivna mjera (prefiks CMD za digitalne, i SET za analogne vrijednosti). Sukladno tome, konačni cilj je odrediti izlazne vrijednosti (korektivne mjere prema procesu) na temelju ulaznih vrijednosti (signali dobiveni iz procesa). Kako se podaci u bazu upisuju u šest različitih tablica (slika 17) grupirani prema gornjoj logičkoj podjeli te tipu podataka, time se za dobivanje inicijalnog podatkovnog okvira, vrijednosti učitane iz baze moraju zajednički grupirati i poravnati prema rednom broju opažanja, te rasporediti prema varijablama predstavljenim grupama identifikatora.

Također, generirani podatkovni okvir mora sadržavati i podatak o operateru (Operator_id) kako bi se moglo razlučiti kojem operateru pripada određeno opažanje

(redak u podatkovnom okviru) što je osnova za reduciranje dobivenog podatkovnog okvira na konačni okvir nad kojim će se primijeniti odabrani model strojnog učenja.

Python kôd u nastavku korišten je za formiranje podatkovnog okvira na način da su prethodno definirana polja sa nazivima atributa usklađenih sa odgovarajućom tablicom u bazi, te su redom dohvaćeni podaci iz svake tablice za pripadno polje naziva atributa. Svako dohvaćanje podataka iz svake slijedeće tablice proširivalo je podatkovni okvir do konačnog unosa svih podataka. Iz prikazanog kôda uočava se logička podjela koja prati attribute u tablicama.

Dodatno, postoji i priprema za podjelu na nezavisne i zavisne varijable na način da se elementi posljednje dvije liste tretiraju kao zavisne varijable, tj., posljednja 33 stupca podatkovnog okvira. Time je olakšana naknadna podjela podatkovnog okvira u primjeni odabranih modela strojnog učenja na različite skupine varijabli. U nastavku kôda, otvara se konekcija na bazu koja se nalazi na drugom virtualnom stroju kao je definirano strukturom ispitnog SCADA sustava na slici 9, te se pri dohvat podataka iz prve tablice koriste polja rednog broja opažanja i šifre operatera da bi se formirala prva dva stupca (atributa) podatkovnog okvira. Nakon toga slijedi učitavanje i konkatencija svih vrijednosti iz svih tablica redom, prema prethodno formiranim poljima naziva varijabli (atributa) čime se podatkovni okvir proširuje do konačnog broja stupaca.

Formirani podatkovni okvir upisuje se u datoteku *FullDataSet.csv*. što nije od značaja za implementaciju konačnog rješenja koje se primjenjuje u realnom vremenu, ali pruža mogućnost da se izdvajanjem podatkovnog skupa generiranog u određenom vremenu, te uzastopnom analizom istog, olakša provođenje eksperimenta, odnosno proces testiranja učinkovitosti odabranih algoritama.

```

# engine to connect to the PostgreSQL database
engine = sqlalchemy.create_engine('postgres://postgres:Pero1!@192.168.144.218:5432/dbHMI')

# dataset with bool alarms
sql_data = pandas.read_sql_table('rt_bool_alm',engine) # get table rt_bool_alm
recColumnLoaded = 0
for tag in arrTagNamesBOOL_ALM:
    df = sql_data.loc[sql_data['bool_alm_tag_name'] == tag] # df with only one tag (matching name in array)
    df1 = pandas.DataFrame({"Record_num": df['bool_alm_rec_num'],
                           "Operator_id": df['bool_alm_operator_id'], tag: df['bool_alm_tag_value']})
    if recColumnLoaded == 0: # first loop only: getting record-number column
        df2 = df1
        recColumnLoaded = 1 # prevent next steps to load record-number column
    else:
        tagVal = df1[tag].to_numpy() # column with tag values in numpy array
        df2[tag] = tagVal # adding tagname column with values from array

# dataset with bool alarms + bool stat
sql_data = pandas.read_sql_table('rt_bool_stat',engine) # get table rt_bool_stat
recColumnLoaded = 0
for tag in arrTagNamesBOOL_STAT:
    df = sql_data.loc[sql_data['bool_stat_tag_name'] == tag] # df with only one tag (matching name in array)
    df1 = pandas.DataFrame({"Record_No": df['bool_stat_rec_num'], tag: df['bool_stat_tag_value']})
    tagVal = df1[tag].to_numpy() # column with tag values in numpy array
    df2[tag] = tagVal # adding tagname column with values from array

# dataset with bool alarms + bool stat + int stat
sql_data = pandas.read_sql_table('rt_int_stat',engine) # get table rt_int_stat
recColumnLoaded = 0
for tag in arrTagNamesINT_STAT:
    df = sql_data.loc[sql_data['int_stat_tag_name'] == tag] # df with only one tag (matching name in array)
    df1 = pandas.DataFrame({"Record_No": df['int_stat_rec_num'], tag: df['int_stat_tag_value']})
    tagVal = df1[tag].to_numpy() # column with tag values in numpy array
    df2[tag] = tagVal # adding tagname column with values from array

# dataset with bool alarms + bool stat + int stat + real stat
sql_data = pandas.read_sql_table('rt_real_stat',engine) # get table rt_real_stat
recColumnLoaded = 0
for tag in arrTagNamesREAL_STAT:
    df = sql_data.loc[sql_data['real_stat_tag_name'] == tag] # df with only one tag (matching name in array)
    df1 = pandas.DataFrame({"Record_No": df['real_stat_rec_num'], tag: df['real_stat_tag_value']})
    tagVal = df1[tag].to_numpy() # column with tag values in numpy array
    df2[tag] = tagVal # adding tagname column with values from array

# dataset with bool alarms + bool stat + int stat + real stat + bool cmd
sql_data = pandas.read_sql_table('rt_bool_cmd',engine) # get table rt_bool_cmd
recColumnLoaded = 0
for tag in arrTagNamesBOOL_CMD:
    df = sql_data.loc[sql_data['bool_cmd_tag_name'] == tag] # df with only one tag (matching name in array)
    df1 = pandas.DataFrame({"Record_No": df['bool_cmd_rec_num'], tag: df['bool_cmd_tag_value']})
    tagVal = df1[tag].to_numpy() # column with tag values in numpy array
    df2[tag] = tagVal # adding tagname column with values from array

# Final dataset with bool alarms + bool stat + int stat + real stat + bool cmd = 73 tag-value columns
# + Record_num. + Oper_id column
sql_data = pandas.read_sql_table('rt_int_set',engine) # get table rt_int_set
recColumnLoaded = 0
for tag in arrTagNamesINT_SET:
    df = sql_data.loc[sql_data['int_set_tag_name'] == tag] # df with only one tag (matching name in array)
    df1 = pandas.DataFrame({"Record_No": df['int_set_rec_num'], tag: df['int_set_tag_value']})
    tagVal = df1[tag].to_numpy() # column with tag values in numpy array
    df2[tag] = tagVal # adding tagname column with values from array

df2.to_csv('FullDataSet.csv', sep='\t', encoding='utf-8')

```

U konačnici, gornji kôd generira podatkovni okvir prikazan na slici 22. sa sljedećim svojstvima:

- 75 stupaca-varijabli, od čega 40 nezavisnih, 33 zavisnih, te dvije varijable za filtriranje operatera i broj opažanja.
- Zavisne varijable u rasponu 42 – 69 (27) predstavljaju binarne vrijednosti (upravljačke signale kao korektivne mjere prema procesu).
- Zavisne varijable u rasponu 69 – 74 (6) predstavljaju cjelobrojne vrijednosti (*set-point* kao korektivne mjere prema procesu).
- Nema null-vrijednosti (u svim opservacijama prikupljene su sve vrijednosti identifikatora kako je objašnjeno ranije u tekstu).
- Sve varijable su numeričke (cjelobrojne ili decimalne).

```
##### Full dataframe info #####
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 76 columns):
#   Column                                     Non-Null Count  Dtype  #   Column                                     Non-Null Count  Dtype
---  ---
0   Unnamed: 0                                15000 non-null  int64  40  STAT_RP_FILTER_CLOG                       15000 non-null  float64
1   Record_num                                15000 non-null  int64  41  STAT_TANK_LEVEL                           15000 non-null  float64
2   Operator_id                               15000 non-null  int64  42  STAT_TANK_TEMP                            15000 non-null  float64
3   ALM_COOL_PUMP_RUN_COL_RET_VALVE_CLS     15000 non-null  int64  43  CMD_COOL_PUMP_START                       15000 non-null  int64
4   ALM_COOL_PUMP_WORK_HRS_MAX              15000 non-null  int64  44  CMD_COOL_PUMP_STOP                       15000 non-null  int64
5   ALM_INLET_FLOW_HIGH                     15000 non-null  int64  45  CMD_COOL_PUMP_WORK_HRS_RESET             15000 non-null  int64
6   ALM_INLET_FLOW_LOW                      15000 non-null  int64  46  CMD_COOL_RET_VALVE_CLOSE                 15000 non-null  int64
7   ALM_INLET_PRESSURE_MIN                   15000 non-null  int64  47  CMD_COOL_RET_VALVE_OPEN                  15000 non-null  int64
8   ALM_P1_FILTER_CLOGGED                    15000 non-null  int64  48  CMD_DISCHG_VALVE_CLOSE                   15000 non-null  int64
9   ALM_P1_WORK_HRS_MAX                      15000 non-null  int64  49  CMD_DISCHG_VALVE_OPEN                    15000 non-null  int64
10  ALM_P2_FILTER_CLOGGED                    15000 non-null  int64  50  CMD_HEATER_START                         15000 non-null  int64
11  ALM_P2_WORK_HRS_MAX                      15000 non-null  int64  51  CMD_HEATER_STOP                          15000 non-null  int64
12  ALM_P3_FILTER_CLOGGED                    15000 non-null  int64  52  CMD_INLET_VALVE_CLOSE                    15000 non-null  int64
13  ALM_P3_WORK_HRS_MAX                      15000 non-null  int64  53  CMD_INLET_VALVE_OPEN                     15000 non-null  int64
14  ALM_RP_FILTER_CLOGGED                    15000 non-null  int64  54  CMD_P1_FILTER_CLEAN                      15000 non-null  int64
15  ALM_RP_WORK_HRS_MAX                      15000 non-null  int64  55  CMD_P1_START                             15000 non-null  int64
16  ALM_TANK_LEVEL_HIGH                      15000 non-null  int64  56  CMD_P1_STOP                              15000 non-null  int64
17  ALM_TANK_LEVEL_LOW                      15000 non-null  int64  57  CMD_P1_WORK_HRS_RESET                    15000 non-null  int64
18  ALM_TANK_TEMP_MAX                        15000 non-null  int64  58  CMD_P2_FILTER_CLEAN                      15000 non-null  int64
19  ALM_TANK_TEMP_MIN                       15000 non-null  int64  59  CMD_P2_START                             15000 non-null  int64
20  STAT_COOL_PUMP_RUN                       15000 non-null  int64  60  CMD_P2_STOP                              15000 non-null  int64
21  STAT_COOL_RET_VALVE_OPEN                 15000 non-null  int64  61  CMD_P2_WORK_HRS_RESET                    15000 non-null  int64
22  STAT_DISCHG_VALVE_OPEN                   15000 non-null  int64  62  CMD_P3_FILTER_CLEAN                      15000 non-null  int64
23  STAT_HEATER_RUN                          15000 non-null  int64  63  CMD_P3_START                             15000 non-null  int64
24  STAT_INLET_VALVE_OPEN                    15000 non-null  int64  64  CMD_P3_STOP                              15000 non-null  int64
25  STAT_P1_RUN                              15000 non-null  int64  65  CMD_P3_WORK_HRS_RESET                    15000 non-null  int64
26  STAT_P2_RUN                              15000 non-null  int64  66  CMD_RP_FILTER_CLEAN                      15000 non-null  int64
27  STAT_P3_RUN                              15000 non-null  int64  67  CMD_RP_START                             15000 non-null  int64
28  STAT_RP_RUN                              15000 non-null  int64  68  CMD_RP_STOP                              15000 non-null  int64
29  STAT_COOL_PUMP_WORK_HRS                  15000 non-null  int64  69  CMD_RP_WORK_HRS_RESET                    15000 non-null  int64
30  STAT_FLOW_IN                             15000 non-null  int64  70  SET_COOL_PUMP_POWER                      15000 non-null  int64
31  STAT_P1_WORK_HRS                         15000 non-null  int64  71  SET_HEAT_EXCHG_VALVE_OPEN               15000 non-null  int64
32  STAT_P2_WORK_HRS                         15000 non-null  int64  72  SET_P1_POWER                             15000 non-null  int64
33  STAT_P3_WORK_HRS                         15000 non-null  int64  73  SET_P2_POWER                             15000 non-null  int64
34  STAT_RP_WORK_HRS                         15000 non-null  int64  74  SET_P3_POWER                             15000 non-null  int64
35  STAT_OUTSIDE_TEMP                        15000 non-null  float64 75  SET_RP_POWER                             15000 non-null  int64
36  STAT_P1_FILTER_CLOG                       15000 non-null  float64 dtypes: float64(8), int64(68)
37  STAT_P2_FILTER_CLOG                       15000 non-null  float64 memory usage: 8.7 MB
38  STAT_P3_FILTER_CLOG                       15000 non-null  float64
39  STAT_PRESSURE_IN                          15000 non-null  float64
```

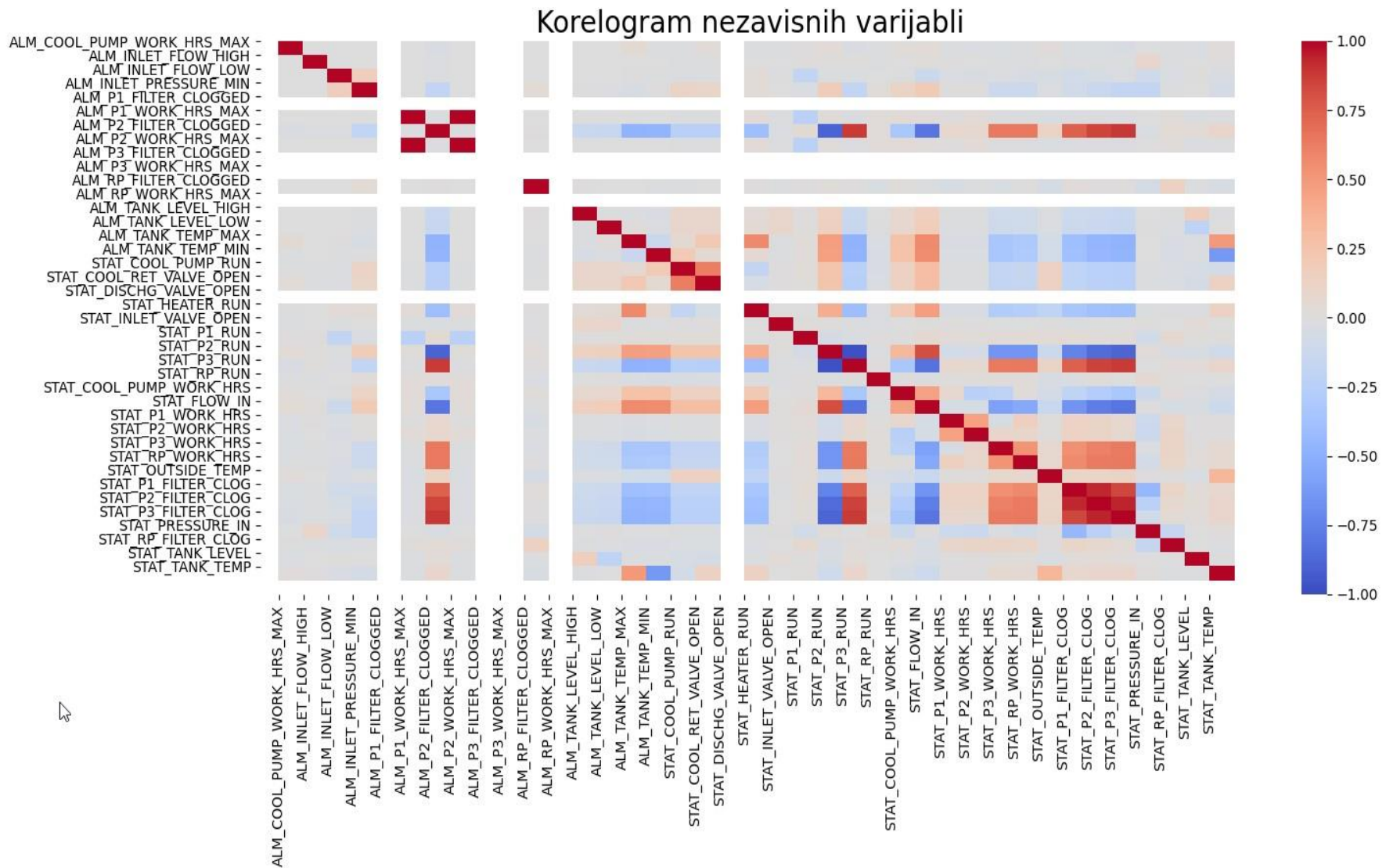
Slika 22: Podatkovni okvir (*dataframe*)

Testni podaci u gornjem podatkovnom okviru sadrže 15000 opservacija. Od toga je 4427 opservacija generirano za vrijeme simuliranja aktivnosti operatera 1, čime se preostalih 10573 opservacija odnosi na aktivnosti operatera 2. Na taj je način obim

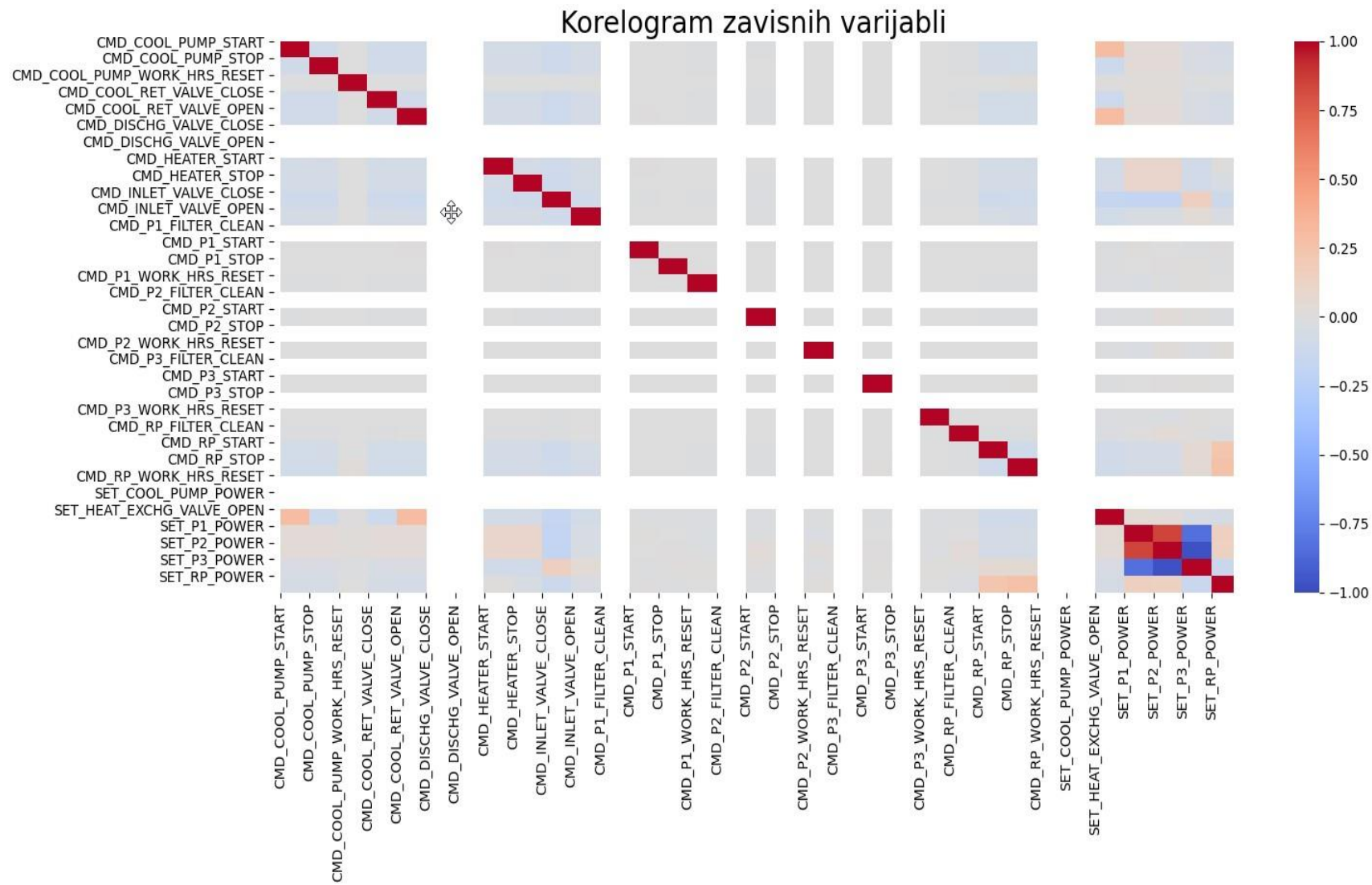
podataka za analizu aktivnosti operatera 2 dvostruko veći u odnosu na podatke prikupljene od aktivnosti operatera 1. Razlog takvom nesrazmjeru leži u temeljnim razlikama u načinu rada između dva simulirana operatera. Sukladno definiranom modelu ponašanja, budući operater 1 reagira prvenstveno po aktivaciji alarma, time su mu je i broj korektivnih mjera u istom vremenskom intervalu reduciran u odnosu na operatera 2 koji provodi više intervencija kako bi održao proces u granicama tolerancije. Navedeno je uočljivo iz slike 20 koja prikazuje *Soft control* način kontinuirane regulacije snage pumpi sa ciljem održavanja izlaznih parametara uz uštedu resursa. Veća količina podataka prikupljenih od operatera 2 također je bitna za uspješnost primijenjenog modela strojnog učenja. Budući se radi od relativno malom setu podataka, a korektivne mjere operatera 2 nisu uniformne kao što je to slučaj sa operaterom 1, time je potrebno i više opažanja za treniranje modela.

Kako je već u prethodnim istraživanjima naglašen značaj i razumijevanje podataka za uspješnost modela (Kang, Catal and Tekinerdogan, 2020), time se u nastavku analiziraju osnovne karakteristike skupa podataka. Pri tome je podatkovni okvir podijeljen na nezavisne i zavisne varijable. Grafikon 1 prikazuje korelogram nezavisnih varijabli, nakon čega je korelogram zavisnih varijabli prikazan grafikonom 2. Svrha korelograma jeste utvrditi međuovisnost (korelaciju) zavisnih, tj., nezavisnih varijabli. U tom smislu, korelacija može biti pozitivna tj., proporcionalna (prikazano crvenom bojom), te negativna tj., obrnuto proporcionalna (prikazano plavom bojom). Vrijednost korelacije odgovara intenzitetu boje na način da viši intenzitet znači i veći stupanj korelacije.

Iz prikazanih korelograma vidljivo je kako postoji značajna pozitivna međusobna korelacija između varijabli koje mjere postotnu razinu začepljenosti filtera na usisnim granama pumpi P1,P2,P3, te daljnja pozitivna korelacija istih sa statusom rada pumpe P2, i negativna sa P3. Također, postoji najviši stupanj negativne korelacije između signala statusa rada pumpi P2 i P3, te najviši stupanj pozitivne korelacije između signala alarma dosega maksimalnih radnih sati za P1 i P2. Takav je odnos prediktorskih (ulaznih) varijabli razumljiv imajući u vidu parametre procesa, te modele ponašanja operatera. Primjerice, pri inicijalizaciji simulacije, najprije se pokreću P1 i P2 koje zajednički održavaju razinu protoka do začepljenja filtera. U takvom režimu rada posljedično postoji korelacija dosega njihovih maksimalnih radnih sati.



Grafikon 1: Korelogram nezavisnih varijabli



Grafikon 2: Korelogram zavisnih varijabli

Kada je riječ o korelaciji kriterijskih (izlaznih) varijabli, korelogram sa grafikona 2 prikazuje znatno manji broj korelacija. Naglašene su jedino pozitivne i negativne korelacije visokog stupnja između analognih upravljačkih signala za pumpe P1,P2,P3 kojim se regulira snaga tijekom rada pojedine pumpe. Takve pozitivne i negativne korelacije logična su posljedica već spomenutog *Soft control* načina kontinuirane regulacije snage pumpi sa ciljem održavanja izlaznih parametara uz uštedu resursa.

Iz korelograma ulaznih varijabli da se uočiti kako su viši stupnjevi korelacija u donjem desnom djelu podatkovnog okvira. Budući se skup podataka sastoji od analognih vrijednosti, pri čemu su i kategorijske varijable prikazane brojčano (0,1), kontinuirane varijable (cjelobrojne i decimalne vrijednosti) nalaze se u donjem djelu podatkovnog okvira (posljednjih 14 varijabli). Time se da zaključiti kako najveći broj i stupanj korelacija postoji pretežno između kontinuiranih varijabli. Navedeno je značajno zbog multikolinearnosti, čije je nepostojanje jedna od pretpostavki za implementaciju modela linearne regresije u predikciji kontinuiranih varijabli.

Multikolinearnost je slučaj međusobne linearne povezanosti dvije ili više prediktorskih varijabli. Time je vrijednost jedne nezavisne varijable moguće dovoljno precizno predvidjeti na temelju jedne (savršena kolinearnost) ili na temelju kombinacije više drugih prediktorskih varijabli. Formalno, multikolinearnost je definirana slijedećom jednadžbom (Graham, 2003), pri čemu je β_i koeficijent regresije, X_i su kolinearne varijable, te ε_i predstavlja varijancu odstupanja.

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_i X_i + \varepsilon_i = 0 \quad (3)$$

Multikolinearnost može biti posljedica stvarnih korelacija, kako je to slučaj ovdje, ili može biti samo prividna kao posljedica tehničkih specifičnosti u sirovim podacima. Utjecaj na naglašenost multikolinearnosti ima veličina podatkovnog okvira, odnosno omjer redova i stupaca. Što je veći broj stupaca i manji broj redova, time je multikolinearnost više naglašena. Značaj kolinearnosti leži u tome što pojedine ulazne varijable mogu izgledati redundantne i time biti isključene iz modela višestruke regresije. Osim korelograma, dobar način uočavanja multikolinearnosti je faktor inflacije varijance VIF (engl. Variance Inflation Factor). U slučaju kada postoji linearna međuovisnost između prediktorskih varijabli, standardna pogreška (engl. SE - standard error) koeficijenata prediktorske varijable će porasti, te posljedično uzrokovati rast

varijance kojeg mjeri VIF na temelju koeficijenta determinacije (R^2) kao omjera varijance (σ) zavisne varijable tj., rezultata predikcije ne temelju jedne ili više nezavisnih varijabli (Daoud, 2018).

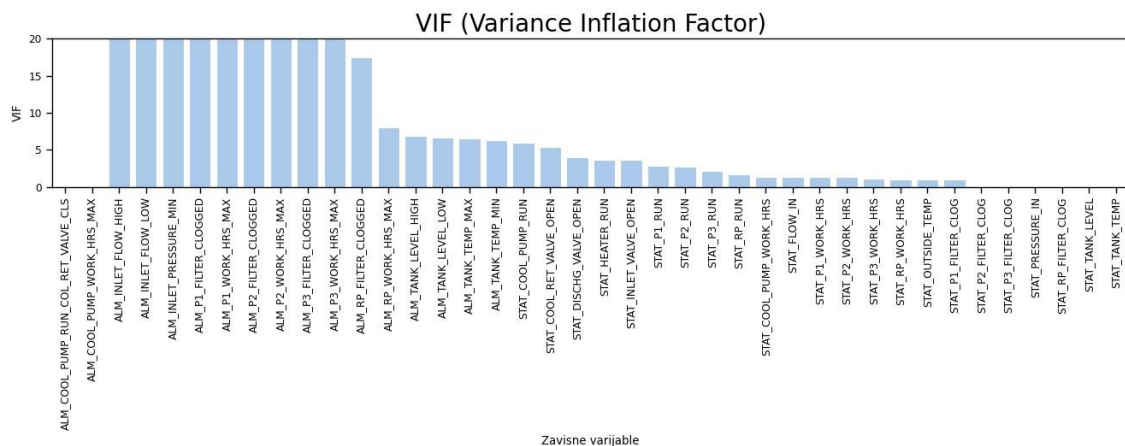
$$VIF = \frac{1}{1-R_t^2} \quad (4)$$

Na ovaj način VIF daje vrijednost koju možemo tumačiti kao procjenu linearne povezanosti za svaku nezavisnu varijablu prema tablici 9.

Tablica 9: VIF interpretacija (Daoud, 2018)

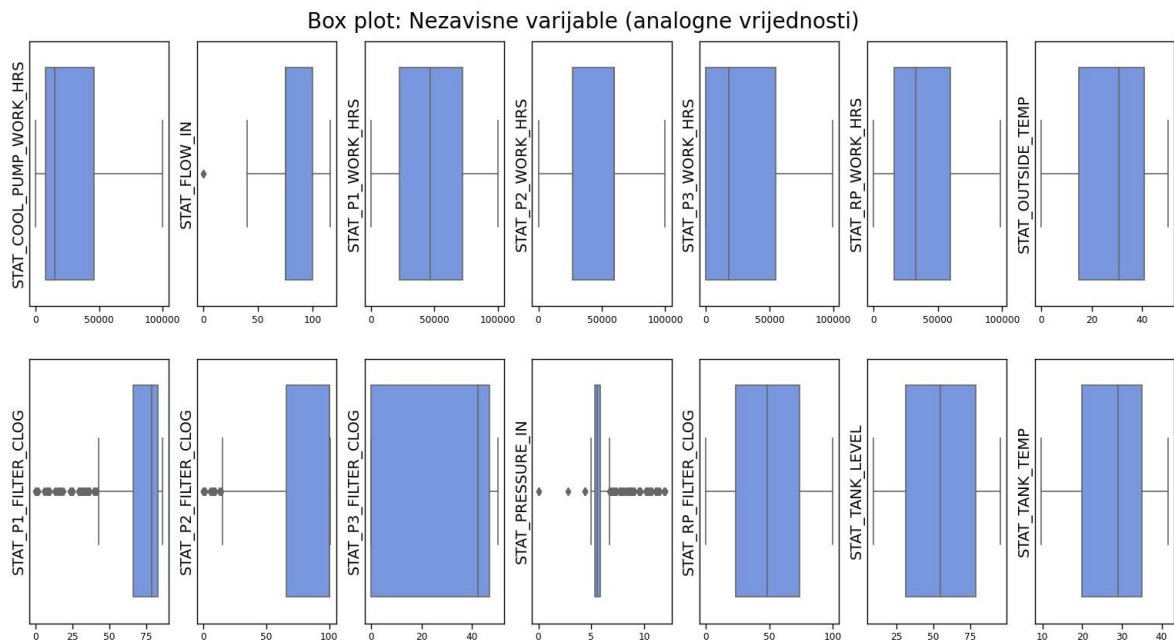
VIF = 1	Ne postoji korelacija
$1 < VIF < 5$	Umjerena korelacija
$VIF > 5$	Visok stupanj korelacije

Grafikon 3 prikazuje VIF vrijednosti nezavisnih varijabli pri čemu se uočava kako jedino vrijednosti alarma izlaze izvan vrijednosti umjerene korelacije prema tablici 9. Takav rezultat je očekivan s obzirom da se alarmi u procesu aktiviraju isključivo na temelju očigledne uzročno-posljedične veze koja postoji u skupu ulaznih varijabli.



Grafikon 3: Faktor inflacije varijance

Dodatne informacije koje upotpunjuju sliku skupa podataka u zadanom podatkovnom okviru prikazane su kroz *Box-plot* na grafikonu 4. Ovakav prikaz se naziva i karakteristična petorka uzorka (engl. five-number summary) X_1, m, qL, qU, X_n budući kroz raspon vrijednosti varijabli $X_{(1...n)}$ prikazuje pet ključnih vrijednosti za razumijevanje distribucije uzorka kroz mjere centralne tendencije (mod) i mjere rasipanja (standardna devijacija, interkvartilni rang).



Grafikon 4: *Box-plot* kontinuiranih nezavisnih varijabli

Iz gornjeg prikaza razabire se relativno normalna distribucija varijabli temperature i razine u tanku, te radnih sati pumpi P1, P2, RP. Zakrivljenost normalne distribucije u slučaju radnih sati pumpe P3, CP, te začepljenosti filtera pumpe P1, P2 i P3 izravna je posljedica režima rada i asinkronog načina prikupljanja podataka (isključivo u trenutku aktivacije upravljačkog signala kao primjenu korektivne mjere). Slično je i u slučaju ekstremno odstupajućih vrijednosti (engl. outlier) koji su uočljivi kod vrijednosti izlaznog pritiska (jedan od izlaznih parametara procesa), te začepljenosti filtera pumpi P1 i P2. U slučaju izlaznih parametara procesa, varijabla protoka zadržava relativno normalnu distribuciju zahvaljujući već spomenutoj *Soft control* regulaciji rada pumpi kojoj se izlazni protok održava u granicama tolerancije (60 od 100 l/min) prema tablici 7. U slučaju izlaznog pritiska takve regulacije nema, te je simulacijom definirana samo donja granica tolerancije (5 bara) na koju utječe začepljenost filtera. Time se uz minimalnu začepljenosti filtera i aktivaciju dvije pumpe vrijednost pritiska može često pojaviti kao *outlier* kako je prikazano na *box-plot*-u.

4.8 Odabir metode strojnog učenja

U odabiru metode strojnog učenja ključnu ulogu imaju tipovi podataka i vremenska zahtjevnost praćenog procesa. Kako je vidljivo iz podatkovnog okvira sa slike 21, te kako je već prethodno rečeno, sve su varijable prikupljene kao numeričke na način da su digitalni signali iz procesa pohranjeni u bazu kao cjelobrojne vrijednosti (0,1) i tako prikazani u podatkovnom okviru. Time se izboru metode može pristupiti dvojako:

- i. Sve se varijable mogu tretirati kao kontinuirane, te se u tom slučaju može primijeniti model višestruke linearne regresije ali i SVM.
- ii. Digitalni signali (varijable sa vrijednostima 0,1) mogu se izdvojiti, te se na njih može primijeniti model klasifikacije.

Dakle u oba gornja slučaja (i, ii) na kontinuirane varijable primjenjuje se linearna regresija. S obzirom na referentne radove (Ademujimi, Brundage and Prabhu, 2017); (Kang, Catal and Tekinerdogan, 2020) kojima su obuhvaćeni najčešće primijenjene metode i algoritmi strojnog učenja, te imajući u vidu karakteristike podatkovnog okvira, SVM može biti logičan izbor algoritma za model klasifikacije. Prema tablici 1, prednosti SVM-a su u modeliranju linearnih i nelinearnih odnosa, uz relativno brzo vrijeme izračunavanja. Uz to, navodi se kao algoritam koji teži dobro generalizirati čak i s ograničenim mogućnostima kakve pruža skup podataka u ovom slučaju imajući u vidu relativno malen uzorak, sa naglašenim međuovisnostima prediktorskih varijabli.

Drugi preduvjet odnosi se na problem vremenske zahtjevnosti koja proizlazi iz praćenja proizvodnog procesa u realnom vremenu. U slučaju izdvojenog pod sustava hidrauličke jedinice, vremenska komponenta nije od ključnog značaja budući u kontinuiranim procesima postoje pod sustavi u sklopu valjaonica koji su znatno kritičniji. Međutim, valja imati u vidu da implementirani model mora generirati rezultate koji mogu pratiti proizvodni proces u realnom vremenu. U tom smislu opravdano je uzeti u obzir više-izlaznu linearnu regresiju (engl. multioutput / multivariate linear regression) koja vrši istovremenu predikciju više izlaznih varijabli na temelju više prediktora. Alternativni način je primjena višestruke regresije (engl. multiple regression) koja bi se morala primjenjivati iterativno prema broju zavisnih varijabli. Odnosi između navedenih modela linearne regresije vidljivi su iz slijedećih formula jednostruke (5), višestruke (6) i više-izlazne linearne regresije (7) (Hidalgo and Goodman, 2013)

$$y = \alpha + x\beta + \varepsilon \quad (5)$$

$$y = \alpha + x_1\beta_1 + x_2\beta_2 + \dots + x_k\beta_k + \varepsilon \quad (6)$$

$$Y_{n \times p} = X_{n \times (k+1)}\beta_{n \times (k+1)} + \varepsilon \quad (7)$$

Uočljivo je kako se u slučaju modela višestruke linearne regresije (6) u suštini odvija postupak jednostavne linearne regresije (5) primijenjen na vektor nezavisnih varijabli sa pripadnim koeficijentima regresije $x_k\beta_k$. U slučaju više-izlazne linearne regresije ne postoji iterativni postupak u tom smislu, već MvLR (Multivariate Linear Regression) model daje istovremeno predikciju cijelog seta zavisnih varijabli što se može promatrati kao matrica $m \times n$ prema slici 23 koja daje matrični prikaz MvLR.

$$\begin{pmatrix} y_{11} & \dots & y_{1m} \\ y_{21} & \dots & y_{2m} \\ y_{31} & \dots & y_{3m} \\ \vdots & \ddots & \vdots \\ y_{n1} & \dots & y_{nm} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ 1 & x_{31} & x_{32} & \dots & x_{3p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} \begin{pmatrix} b_{01} & \dots & b_{0m} \\ b_{11} & \dots & b_{1m} \\ b_{21} & \dots & b_{2m} \\ \vdots & \ddots & \vdots \\ b_{p1} & \dots & b_{pm} \end{pmatrix} + \begin{pmatrix} e_{11} & \dots & e_{1m} \\ e_{21} & \dots & e_{2m} \\ e_{31} & \dots & e_{3m} \\ \vdots & \ddots & \vdots \\ e_{n1} & \dots & e_{nm} \end{pmatrix}$$

Slika 23: MvLR (*Multivariate Linear Regression*) model (Helwig, 2017)

Uz prednost istovremenog generiranja rezultata, više-izlazna linearna regresija dodatno uzima u obzir međuzavisnost izlaznih varijabli, tj., pri predikciji varijable y_i uzimaju se u obzir vektor prediktorskih varijabli X ali i zavisne varijable $y_1 \dots y_i$. Navedeni je model moguće primijeniti kao klasifikaciju (MvLC) i kao regresiju (MvLR) na skup podataka odgovarajuće strukture kako je prikazano na slici 24.

- Training data: $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, $y_i \in \mathcal{Y} = \{0, 1\}^m$
- **Predict** the vector $y = (y_1, y_2, \dots, y_m)$ for a given x .
- Training data: $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, $y_i \in \mathcal{Y} = \mathbb{R}^m$
- **Predict** the vector $y = (y_1, y_2, \dots, y_m)$ for a given x .

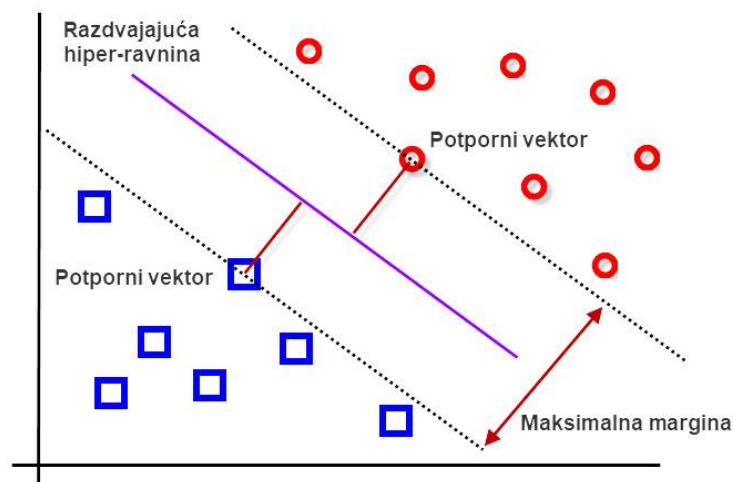
	X_1	X_2	Y_1	Y_2	...	Y_m
x_1	5.0	4.5	1	1	1	0
x_2	2.0	2.5	0	1		0
\vdots	\vdots	\vdots	\vdots	\vdots		\vdots
x_n	3.0	3.5	0	1		1
x	4.0	2.5	1	1		0

	X_1	X_2	Y_1	Y_2	...	Y_m
x_1	5.0	4.5	14	0.3		9
x_2	2.0	2.5	15	1.1		4.5
\vdots	\vdots	\vdots	\vdots	\vdots		\vdots
x_n	3.0	3.5	19	0.9		2
x	4.0	2.5	?	?		?

Slika 24: Primjena MvLR i MvLC na skup podataka (Dembczy, 2013)

Imajući u vidu tipove podataka zavisnih varijabli od kojih se većina (27 od ukupno 33) odnosi na digitalne upravljačke signale prema procesu, prema varijanti (ii) kako je već rečeno, logičan je izbor SVM, odnosno SVC (engl. Support Vector Classifier). To ne

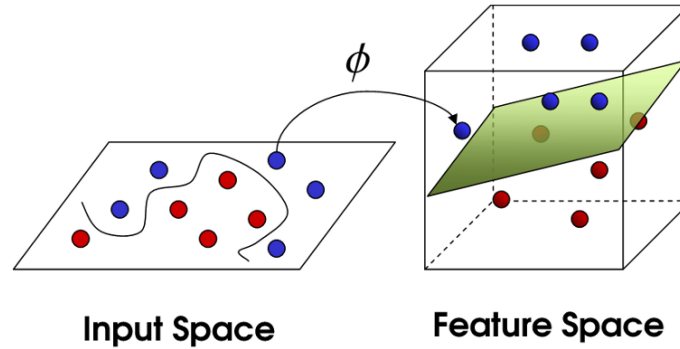
isključuje mogućnost primjene SVM-a na cijeli skup podataka s obzirom da je riječ o modelu nadziranog učenja koji se može primijeniti na probleme regresije kao i klasifikacije. Cilj modela jeste pronaći separirajuću hiper-ravninu (engl. hyperplane) u multi dimenzionalnom prostoru uz pomoću koje se postiže najviši stupanj razdvajanja dviju klasa na način da se maksimizira margina kao razmak između najbližih točaka dviju različitih klasa koje se nazivaju potporni vektori (engl. support vectors). Grafikon 5 prikazuje jedan primjer razdvajanja dviju klasa u dvodimenzionalnom prostoru, pri čemu su klase razdvojene pravcem.



Grafikon 5: Stroj potpornih vektora: dvodimenzionalni prostor

Dimenzija prostora određena je brojem prediktorskih varijabli, dok je hiper-ravnina uvijek jednu dimenziju manja od dimenzije prostora. Prema gornjoj slici prostor je dvodimenzionalan, čime je ravnina jednodimenzionalna tj., pravac.

U slučajevima kada je nemoguće linearno razdvajanje klasa, postoji tehnika *Kernel trick*. Riječ je o funkcijama koje ulaze nižih dimenzija konvertiraju u prostor više dimenzije, čime se klase koje nije bilo moguće razdvojiti u prostoru niže dimenzije pretvaraju u rješiv problem hiper-ravninom više dimenzije. Slika 25 prikazuje primjer takvog načina pronalaska rešenja klasifikacije prelaskom sa dvodimenzionalnog prostora u trodimenzionalni.



Slika 25: Stroj potpornih vektora: *Kernel trick* (PRIMO.ai, 2019)

Sa lijeve strane vide se dvije skupine podataka u 2D prostoru koje se ne mogu odvojiti pravcem, međutim prelaskom na višu dimenziju (3D prostor), pravac može postati 2D ravnina (jedna dimenzija niže od dimenzije prostora) te uspješno odvojiti klase. Osim toga, ova tehnika omogućava znatnu uštedu vremena s obzirom da dozvoljava računanje sa prediktorima u originalnom prostoru, bez računanja koordinata u prostoru više dimenzije kako je prikazano na primjeru prema (Zhang, 2018) sa dvije podatkovne vrijednosti u 3-dimezionalnom prostoru (8) koje se mapiraju u 9-dimenzionalni prostor za koji je potrebna kalkulacija (9), a koja u konačnici rezultira samo skalarnom vrijednošću.

$$x = (x_1, x_2, x_3)^T; y = (y_1, y_2, y_3)^T \quad (8)$$

$$\begin{aligned} \phi(x) &= (x_1^2, x_1x_2, x_1x_3, x_2x_1, x_2^2, x_2x_3, x_3x_1, x_3x_2, x_3^2)^T \\ \phi(y) &= (y_1^2, y_1y_2, y_1y_3, y_2y_1, y_2^2, y_2y_3, y_3y_1, y_3y_2, y_3^2)^T \end{aligned} \quad (9)$$

Vremenska složenost ovakvog računanja je $O(n^2)$. Međutim, ako se primijeni *kernel* funkcija $k(x,y)$ kao umnožak vektora koji predstavlja sumu umnoška svih parova ulaznih vrijednosti (linearni kernel), do rezultata se može doći unutar inicijalnog 3-dimenzionalnog prostora kao sume umnoška $x^T \cdot y$ (10) čija je vremenska složenost $O(n)$ znatno manja.

$$\begin{aligned} k(x,y) &= (x^T y)^2 \\ &= (x_1y_1 + x_2y_2 + x_3y_3)^2 \\ &= \sum_{i,j=1}^3 x_i x_j y_i y_j \end{aligned} \quad (10)$$

5. Rezultati

S obzirom na postavljeni cilj istraživanja u predikciji korektivnih mjera kao potpore operateru u kontroli procesa na temelju pozitivnih rezultata prethodnih aktivnosti različitih operatera, za početak je najučinkovitiji operater izdvojen na temelju optimalne iskoristivosti uređaja. Navedeno je dobiveno iz prosječne vrijednosti zabilježene trenutne postotne snage svih pumpi u radu čime su dobivene slijedeće vrijednosti:

- Operater 1: prosječna snaga po aktivnoj pumpi kroz sve opservacije = 45,97%
- Operater 2: prosječna snaga po aktivnoj pumpi kroz sve opservacije = 33,65%

Prema navedenom rezultatu, operater 2 ostvario je bolji rezultat, odnosno održavao je kontinuitet proizvodnog procesa uz optimalniju iskoristivost uređaja. Navedeno se nikako ne može koristiti kao realna procjena utrška resursa s obzirom na asinkrona opažanja čime je izostavljen vremenski kontinuitet, ali je dovoljno za aproksimaciju učinkovitosti rada operatera. Dakle u nastavku se podatkovni okvir ograničava na one opservacije koje su zabilježene u vrijeme rada operatera 2, tj., pratile su njegove korektivne mjere prema sustavu, čime je broj opažanja limitiran na 10573 od ukupnih 15000 koji čine podatkovni okvir⁶.

5.1 Više-izlazna linearna regresija (uz *Gradient Boost regressor*)

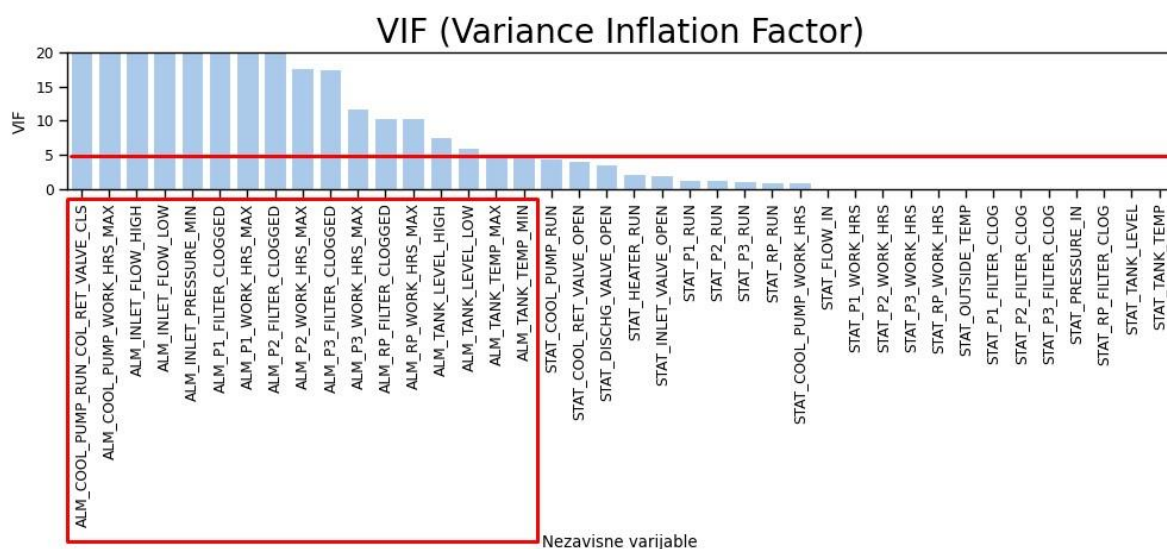
S obzirom na prethodno spomenute izbore primijenjenih metoda, model više-izlazne linearne regresije primijenjen je na cjelokupan podatkovni okvir, kao i na reducirani okvir isključivo kontinuiranih tipova zavisnih i nezavisnih varijabli. Takav je pristup moguć jer su digitalni signali (statusi generirani iz procesa, te upravljački signali generirani iz HMI-a) tretirani kao analogne vrijednosti kako je već spomenuto u prethodnom tekstu. Tablica 10 prikazuje rezultate primjene modela višestruke više-izlazne regresije na različite pod skupove podataka u podatkovnom okviru, bez dodatne obrade skupa podataka (odstupajuće vrijednosti, multikolinearnost, značajnost prediktora, balansiranje). Omjer skupova podataka za treniranje i testiranje je 80:20.

⁶ Nesrazmjernost broja opažanja između različitih operatera (10573:4427) rezultat je razlike u simulaciji aktivnosti pojedinog operatera, te nije odraz razlike u ukupnom vremenu uzorkovanja za pojedinog operatera.

Tablica 10: Rezultati modela više-izlazne linearne regresije

Varijable	R ²	MSE ⁷	MAE ⁸	Trening (sec.)	Predikcija (sec.)
Sve varijable u cijelom skupu podataka (<i>raw dataset</i>)	0,87	32,34	0,93	85,87	0,08
Kontinuirane nezavisne i sve zavisne	0,78	38,62	1,16	48,99	0,08
Kontinuirane nezavisne i kontinuirane zavisne	0,72	212,45	6,28	11,43	0,02

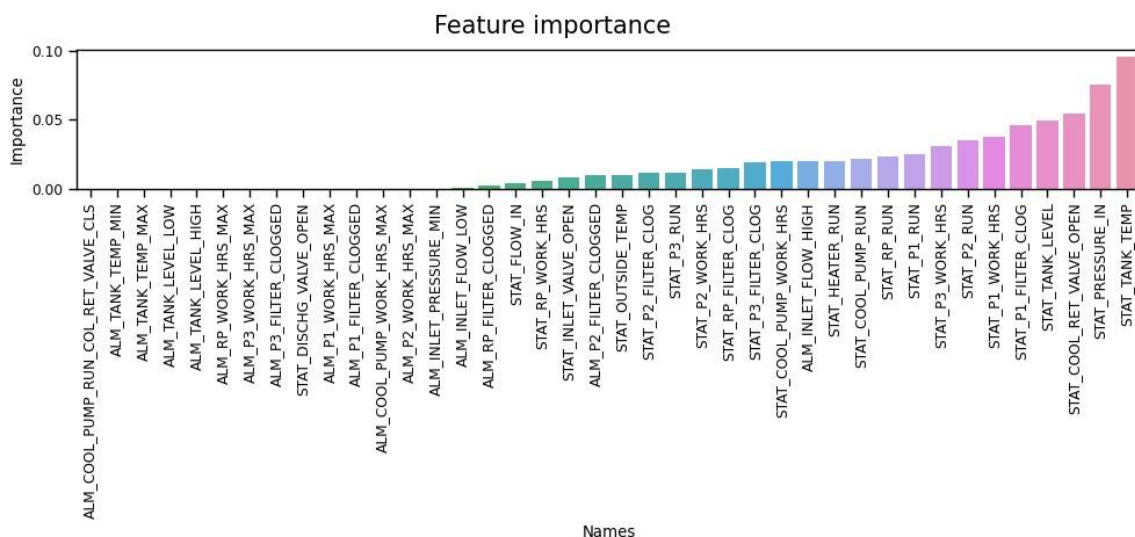
Premda vrijeme učenja (treniranja) modela od 11.43 sekundi (u slučaju primjene modela nad isključivo kontinuiranim varijablama) potencijalno može zadovoljiti primjenu u realnom vremenu za neke podsustave kontrole procesa, evidentno je da se radi o rezultatima koji ne mogu zadovoljiti planiranu primjenu s obzirom na koeficijent determinacije (R²), odnosno točnost predviđanja od najviše 87% (postotak varijance objašnjen modelom). Ovakav rezultat može se popraviti eliminiranjem nezavisnih varijabli sa visokim VIF vrijednostima (tablica 9) prema grafikonu 6, te prediktora koji imaju minoran značaj za model regresije prema grafikonu 7.



Grafikon 6: : Izdvojene nezavisne varijable sa visokim VIF vrijednostima

⁷ MSE (Mean Square Error) = srednja kvadratna pogreška.

⁸ MAE (Mean Absolute Error) = srednja apsolutna pogreška.



Grafikon 7: : Relativna važnost prediktora za skup podataka operatera 2

Tablica 11 prikazuje rezultate nakon navedenog reduciranja broja varijabli prilikom čega su isključeni svi signali alarma osim dva koja imaju višu relativnu važnost prediktora (grafikon 7).

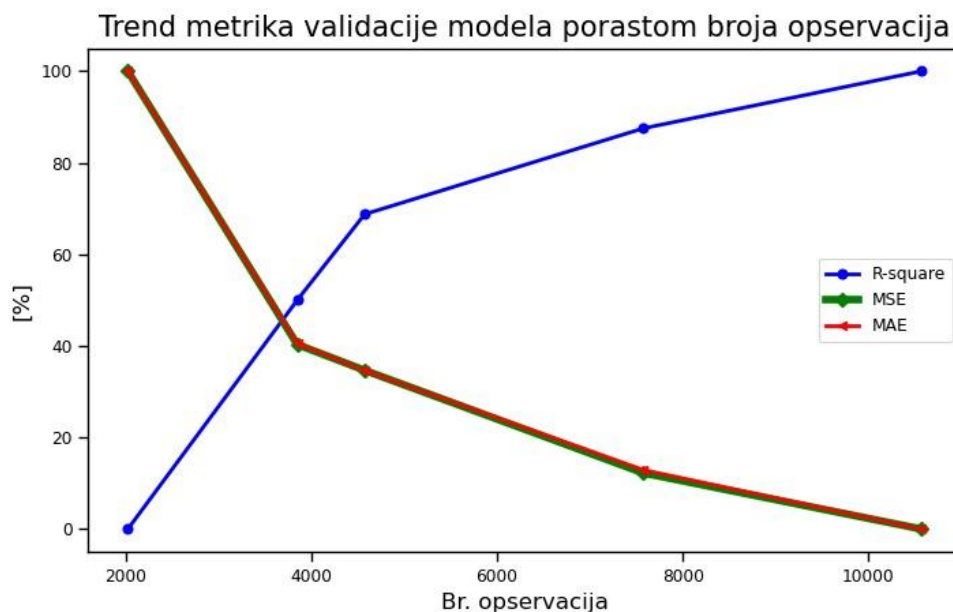
Tablica 11: Rezultati modela više-izlazne linearne regresije na reduciranom setu

Varijable	R ²	MSE ⁹	MAE ¹⁰	Trening (sec.)	Predikcija (sec.)
Sve varijable u reduciranom skupu podataka	0,87	32,65	0,94	52,0	0,08
Kontinuirane nezavisne i sve zavisne	0,78	38,62	1,16	45,93	0,07
Kontinuirane nezavisne i kontinuirane zavisne	0,72	212,56	6,28	11,26	0,01

Reduciranjem podatkovnog okvira na prediktore sa višim značajem za predikciju (engl. feature importance) postiglo se kraće vrijeme treninga, međutim točnost predviđanja i dalje u najboljem slučaju iznosi 87% uz vrlo visoku srednju kvadratnu i apsolutnu pogrešku. Ukoliko se model iterativno primijeni uz uzastopno povećavanje postojećih broja opservacija u podatkovnom okviru, uočava se trend kretanja vrijednosti u metrici evaluacije modela prema grafikonu 8.

⁹ MSE (Mean Square Error) = srednja kvadratna pogreška.

¹⁰ MAE (Mean Absolute Error) = srednja apsolutna pogreška.

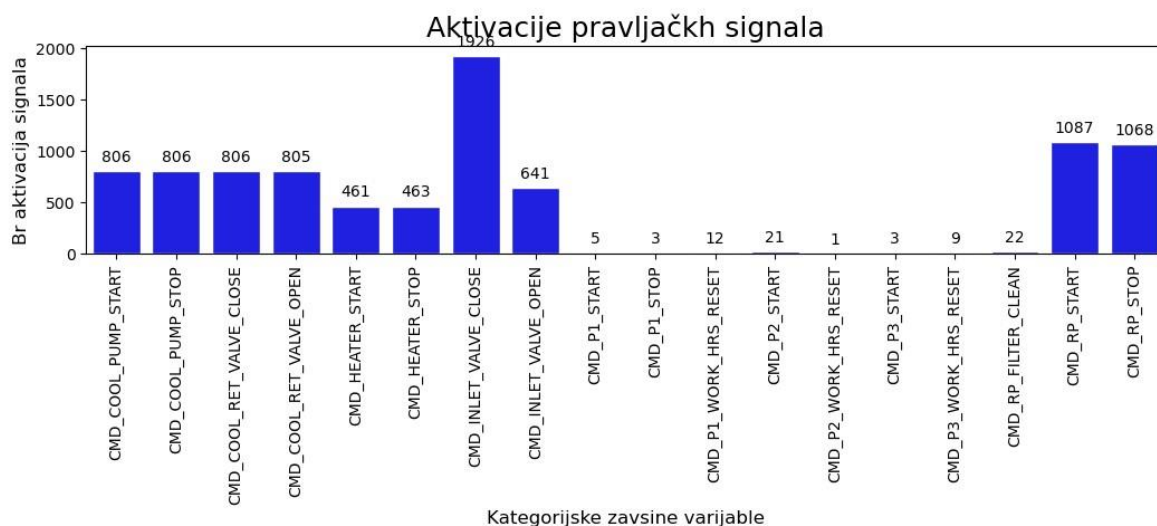


Grafikon 8: Trend vrijednosti regresijskih metrika porastom broja opservacija

Prikazane vrijednosti rezultat su primjene modela na uzorku od 2022, 3847, 4573, 7573 i 10573 opservacija, i ukazuju na kontinuirani porast vrijednosti koeficijenta determinacije (R^2), te pad vrijednosti srednje apsolutne pogreške (MAE) i kvadratne pogreške (MSE). Premda kretanje nije linearno, ono je kontinuirano i ukazuje na porast preciznosti primijenjenog modela sa porastom veličine uzorka.

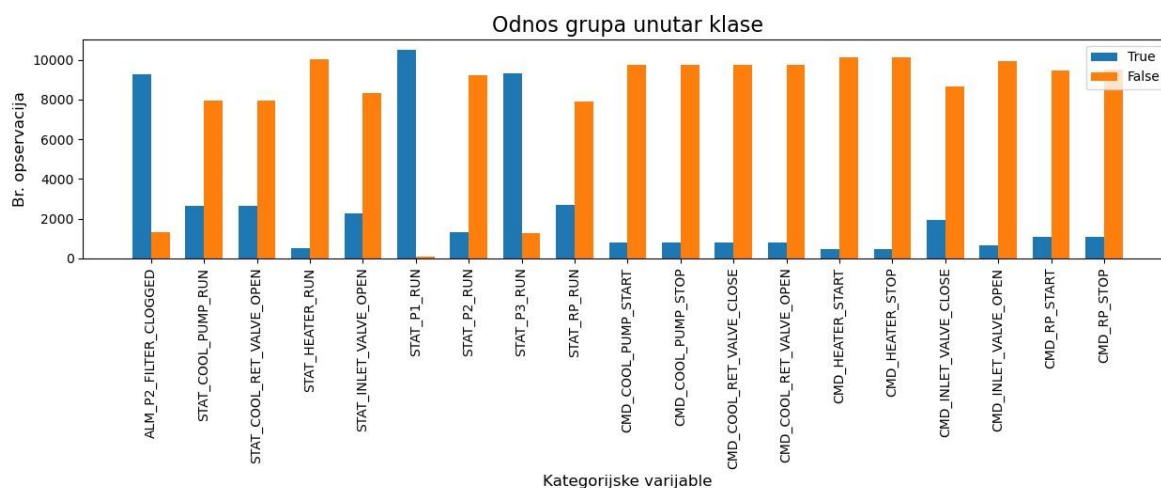
5.2 Više-izlazna klasifikacija (Stroj potpornih vektora)

Primjena SVM-a odnosi se na mogućnost predikcije kategorijskih varijabli, koje su prevladavajuće među zavisnim varijablama budući se odnose na digitalne upravljačke signale prema procesu. Kako se neki od tih signala, ali i neki od statusa procesa vrlo rijetko aktiviraju, time se eliminacijom varijabli sa samo jednom klasom početni podatkovni okvir sa 73 varijable reducirao na 49. Međutim, i dalje za neke varijable postoji suviše nepovoljan omjer klasa kako se vidi iz grafikona 9, što dovodi do problema prilikom odvajanja skupa za treniranje i testiranje. Time su iz podatkovnog okvira, uz analogne, eliminirane i sve varijable čija je pojavnost jedne od klase manja od 100 puta ($< 1\%$), te , čime u konačnici dolazimo do reduciranog podatkovnog okvira sa 19 varijabli, odnosno 10 zavisnih i 9 nezavisnih.



Grafikon 9: Broj aktivacija upravljačkih signala (zavisnih varijabli)

Nastavno na odnos klasa kategorijskih varijabli, i među preostalim varijablama podatkovnog okvira može se uočiti znatan nesrazmjer između klasa, što je osobito naglašeno među zavisnim varijablama kako je prikazano na grafikonu 10.



Grafikon 10: Nebalansirani podatkovni okvir

Prikazane su vrijednosti dobivene iz podatkovnog okvira sa 10573 opservacija, te se da zaključiti kako se radi o izrazito nebalansiranom skupu podataka gdje je „True“ manjinska (pozitivna) klasa. Rješenja koja se mogu primijeniti odnose se na promjenu klasifikatora, te promjene na razini skupa podataka ponovnim uzorkovanjem ili transformacijom postojećeg skupa podataka (engl. resampling dataset), pri čemu se mogu primijeniti tehnike reduciranja većinske klase (engl. undersampling; downsampling) ili dupliciranje opservacija manjinske klase (engl. oversampling; upsampling). Budući je u ovom slučaju klasa od interesa ona sa izrazito malom pojavnošću, time se pristupa nasumičnom dupliciranju iste, što je i preporučljivo u situacijama sa manjim skupom podataka (Jayawardena, 2020), kao što je slučaj u

ovom primjeru. Tablica 12 prikazuje rezultate predikcije primjenom SVM klasifikatora nad nebalansiranom podatkovnom skupom sa 10573 opservacija, uz omjer skupova za trening i testiranje 85:15.

Tablica 12: Rezultati više-izlaznog SVM modela za reducirani podatkovni okvir

Izlazne kategorijske varijable (reducirani podatkovni okvir)	ROC ¹¹	TP/FP ¹²	TN/FN ¹³	T. Pred. (0) ¹⁴ T. Pred. (1)	Točnost modela
CMD_COOL_PUMP_START	0,89	12 92	1458 24	0,98 0,88	0,98
CMD_COOL_PUMP_STOP	0,93	11 108	1451 16	0,99 0,91	0,98
CMD_COOL_RET_VALVE_CLOSE	0,5	0 0	1461 125	0,92 0	0,92
CMD_COOL_RET_VALVE_OPEN	0,5	0 0	1480 106	0,93 0	0,93
CMD_HEATER_START	0,5	0 0	1531 55	0,91 0	0,91
CMD_HEATER_STOP	0,91	6 58	1510 12	0,99 0,9	0,98
CMD_INLET_VALVE_CLOSE	0,5	0 0	1307 279	0,82 0	0,82
CMD_INLET_VALVE_OPEN	0,5	0 0	1488 98	0,94 0	0,94
CMD_RP_START	0,5	0 0	1461 170	0,89 0	0,89
CMD_RP_STOP	0,56	21 24	1406 135	0,91 0,53	0,90

Premda se točnost predikcije prema podacima iz tablice kreće u rasponu od 0,89 do 0,98 pojedinačno po zavisnim varijablama, što se može smatrati relativno dobrim rezultatom, iz konfuzijske matrice (stupci TP/FP i TN/FN) vidljivo je kako za više od 50% varijabli u testnom skupu podataka postoji samo jedna klasa kao posljedica nebalansiranog skupa podataka (grafikon 10). Tablica 13, u nastavku prikazuje rezultate predikcije primjenom klasifikatora nad reduciranim skupom nakon primjene *oversampling*-a, čime je podatkovni okvir uvećan na 19010 opservacija.

¹¹ ROC-AUC = Receiver Characteristic Operator (ROC) Area Under the Curve (AUC)

¹² TP/FP (True positive/False positive) = Osjetljivost modela (istinito i lažno predviđena klasa 1-„True“).

¹³ TN/FN (True negative/False negative) = Osjetljivost modela (istinito i lažno predviđena klasa 0-„False“).

¹⁴ Točnost predikcije klase 1 („True“) i 0 („False“).

Tablica 13: Rezultati SVM modela za reducirani podatkovni okvir uz *oversampling*

Izlazne kategorijske varijable (uz <i>oversampling</i> pod. okvira)	ROC ¹⁵	TP/ FP ¹⁶	TN/ FN ¹⁷	T. Pred. (0) ¹⁸ T. Pred. (1)	Točnost modela
CMD_COOL_PUMP_START	0,83	15 94	2696 47	0,98 0,86	0,97
CMD_COOL_PUMP_STOP	0,85	22 104	2685 41	0,98 0,83	90,5
CMD_COOL_RET_VALVE_CLOSE	0,5	0 0	2720 135	0,95 0	0,95
CMD_COOL_RET_VALVE_OPEN	0,5	0 0	2730 122	0,96 0	0,96
CMD_HEATER_START	0,5	0 0	2764 88	0,97 0	0,97
CMD_HEATER_STOP	0,75	20 39	2756 37	0,99 0,66	0,83
CMD_INLET_VALVE_CLOSE	0,5	1 1	2532 3181	0,89 0,5	0,69
CMD_INLET_VALVE_OPEN	0,5	0 0	2758 94	0,97 0	0,97
CMD_RP_START	0,5	0 0	2690 162	0,94 0	0,94
CMD_RP_STOP	0,87	224 1291	1185 152	0,89 0,85	0,87

S obzirom na raspon vrijednosti točnosti predikcije od 0,69 do 0,97 može se zaključiti da povećanje skupa podataka uz *oversampling* nije polučio rezultate, odnosno promatrajući konfuzijsku matricu, došlo je tek do minornog pomaka u odnosu na omjer pojavnosti klasa.

6. Diskusija

Premda su implementirani modeli strojnog učenja u skladu sa primjenom u praksi, odnosno prema dosadašnjim primjerima istraživanja (Ademujimi, Brundage and Prabhu, 2017; Kang, Catal and Tekinerdogan, 2020), dobiveni rezultati ukazuju na slijedeće ključne nedostatke kod primjene modela više-izlazne linearne regresije:

¹⁵ ROC-AUC = Receiver Characteristic Operator (ROC) Area Under the Curve (AUC) – predstavlja općenitu dijagnostičku točnost modela u rasponu vrijednosti od 0 do 1.

¹⁶ TP/FP (True positive/False positive) = Osjetljivost modela (istinito i lažno predviđena klasa 1-„True“).

¹⁷ TN/FN (True negative/False negative) = Osjetljivost modela (istinito i lažno predviđena klasa 0-„False“).

¹⁸ Točnost predikcije klase 1(„True“) i 0(„False“).

- Niske vrijednosti koeficijenta determinacije (R^2), osobito za kontinuirane varijable.
- Izuzetno visoka srednja kvadratna pogreška (MSE) u odnosu na srednju apsolutnu pogrešku (MAE).
- Znatan dio prediktora sa visokom VIF vrijednošću (>5), tj., svi signali alarma.
- Mali udio prediktora sa značajnom vrijednošću relativne važnosti prediktora (*feature importance*).

Nadalje, promatrajući rezultate implementacije modela više-izlazne klasifikacije uz SVM klasifikator, uočavaju se slijedeći nedostaci:

- Nebalansiran skup podataka (23 varijable sa pojavnošću samo jedne klase).
- Znatan nesrazmjer klasa među zavisnim varijablama (digitalnim upravljačkim signalima).
- Samo tri od ukupno deset zavisnih varijabli iz reduciranog podatkovnog okvira ima prihvatljivu ROC vrijednost iznad 0.7 (Mandrekar, 2010)
- U testnom podatkovnom setu (15% ukupnih opservacija) postoji šest varijabli sa pojavnošću samo jedne klase.
- Primjena tehnike *oversampling*-a uz povećanje broja opservacija za 79.7% daje tek minorna poboljšanja pojedinih parametara (konfuzijska matrica), te rezultira sa nižim ROC vrijednostima.

Na dobivene rezultate moglo se djelomično utjecati kroz parametre primijenjenih algoritama, međutim objašnjenje i uzrok za navedene nedostatke leži u skupu podataka, odnosno količini prikupljenih podataka, parametrima simulacije ponašanja operatera, te asinkronom načinu uzorkovanja.

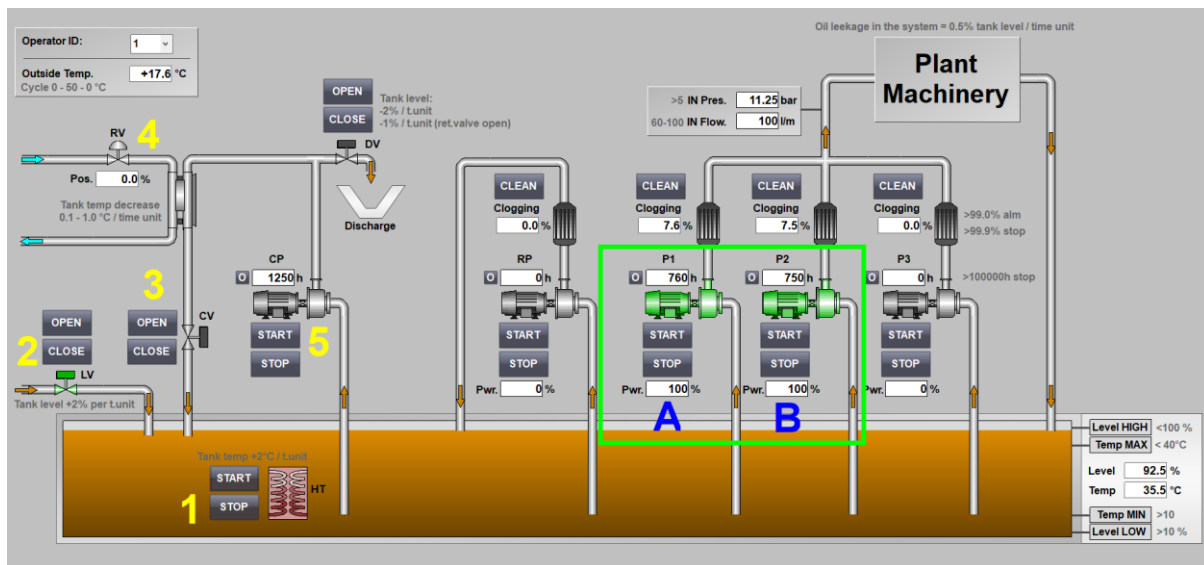
U primjeni modela više-izlazne linearne regresije pokazano je kako se povećanjem skupa podataka postiže veća točnost (grafikon 8). Dakle, samo nastavak uzorkovanja u dužem periodu vremena polučio bi prihvatljivu preciznost modela, no to ne bi riješilo problem značajno viših MSE vrijednosti u odnosu na MAE. Imajući u vidu specifičnost skupa podataka koji sadrži znatan broj ekstremno odstupajućih vrijednosti među kontinuiranim varijablama kako se vidi sa *box-plota* (grafikon 4), vidljivo je kako upravo treće mjerenje (prema tablici 11) ima najvišu MSE vrijednost, a koje se odnosi na isključivo kontinuirane varijable. Navedeno samo potvrđuje slijedeću činjenicu: „Učinak kvadrata u MSE jednadžbi je naj očitiji uz prisustvo ekstremno odstupajućih vrijednosti

(*outliers*) u podacima. Dok svaki rezidual u MAE proporcionalno doprinosi ukupnoj pogrešci, pogreška u MSE raste kvadratno. To u konačnici znači da ekstremno odstupajuće vrijednosti u podacima doprinose znatno većoj ukupnoj pogrešci u slučaju MSE nego što je to slučaj u MAE“ (Pascual, 2018) (vlastiti prijevod). Kako odstupajuće vrijednosti spadaju u regularno ponašanje pojedinih segmenata sustava, kao što je to slučaj sa postotnim začepljenjem filtera svake od pumpi, te vrijednošću izlaznog pritiska, time je kao mjeru rezidualne pogreške dovoljno uzeti MAE, čime se navedeno odstupanje može zanemariti.

Glede prediktora sa visokom vrijednošću faktora inflacije varijance, isti ukazuju na linearnu međuovisnost između prediktorskih varijabli svih signala alarma, čime standardna pogreška koeficijenata prediktorskih varijabli raste kako je već rečeno prethodno u tekstu. Međutim, isto nije od značaja za implementaciju modela na razmatrani podatkovni skup koji je generiran korektivnim mjerama operatera 2. S obzirom na parametre simulacije korektivnih mjera koje poduzima operater 2 prema parovima okidača i kontrolnih mjera kako je navedeno prethodno u tekstu, cilj je optimalan utrošak resursa, čime se kontinuitet procesa održava unutar granica tolerancije, tj., korektivne mjere se poduzimaju prije aktivacije alarmnih signala. U skladu sa time, grafikon relativne važnosti prediktora (grafikon 7) izdvaja signale alarma kao varijable koje nisu od važnosti za primijenjeni model. Dakle, u slučaju primjena modela više-izlazne regresije može se zaključiti kako se već samim povećanjem broja opservacija dostižu zadovoljavajući rezultati, dok se problem manjeg broja prediktora sa niskim *feature importance* potencijalno može riješiti promjenom načina uzorkovanja, što pak daleko više utječe na rezultate klasifikacije.

U primjeni modela više-izlazne klasifikacije, svi gore uočeni nedostaci mogu se pripisati nebalansiranom podatkovnom skupu nastalom kao posljedica karakteristika praćenog procesa, te parametara simulacije aktivnosti operatera. Navedeno se može lako uočiti na primjeru prema slici 26, koja bilježi stanje procesa uz visoku razinu i temperaturu u tanku, te aktivne pumpe P1 i P2 sa 100% snage, uz izlazni protok od 100 lit./min. U takvim okolnostima, a prema parametrima simulacije definiranim za operatera 2, poduzimaju se korektivne mjere za normalizaciju temperature i razine u tanku, te reduciranje snage pumpi u cilju optimizacije potrošnje uz održavanje izlaznog protoka u granicama tolerancije (60-100 lit./min.). Poduzete mjere mogu imati slijedeći

slijed: isključivanje grijača u tanku (1), zatvaranje dovodnog ventila (2), pokretanje sustava hlađenja (3,4,5), te regulaciju snage pumpi (A i/ili B).



Slika 26: Trenutno stanje procesa

Budući redoslijed ne mora nužno biti takav, prilikom korekcije snage pumpi P1 i P2 moguće su neke od kombinacija vrijednosti signala statusa uređaja prikazane na tablici 14. Ovisno o trenutnim vrijednostima temperature i razina u tanku, moguće je da se sve korektivne mjere provedu u istom vremenskom ciklusu simulacije, ili da se problem visoke razine i visoke temperature rješava u dva ciklusa. U oba slučaja, moguće je da vrijednosti razine i temperature u tanku ostanu nepromijenjene do slijedećeg ciklusa simulacije vrijednosti.

Tablica 14: Primjer mogućih vrijednost signala trenutnog stanja procesa

HT(1)	LV(2)	CV(3)	RV(4)	CP(5)	Razina	Temp.
1	1	0	0.0	0	95.0	36.0
0	0	0	0.0	0	95.0	36.0
0	0	1	50.0	1	95.0	36.0
1	1	1	50.0	1	95.0	36.0
0	1	0	0.0	0	95.0	36.0
1	0	0	0.0	0	95.0	36.0

Iz gornjeg primjera razumljivo je kako su statusni signali uređaja HT, LV, CV, RV, CP relativno loši prediktori. Premda je naveden primjer odraz realnog odvijanja procesa, jasno je da postoji određena stohastička komponenta sustava koja utječe na važnost prediktora (grafikon 7) a time i na rezultate predikcije.

Drugi dio problema povezan sa gornjim primjerom, a koji rezultira sa nebalansiranim skupom podataka leži u asinkronom uzorkovanju. Kako je za predmetni rad ključna predikcija korektivnih mjera prema sustavu, time je primijenjeno asinkrono uzorkovanje koje kao okidač za registriranje vrijednosti varijabli koristi aktivaciju upravljačkog signala prema sustavu (postavljanje vrijednosti bilo koje od zavisnih varijabli). Premda takav pristup ima smisla s obzirom na implementaciju algoritama strojnog učenja u realnom vremenu (Albertos *et al.*, 1999), isti rezultira sa nebalansiranim skupom podataka što se može uočiti na već spomenutom primjeru iz tablice 14.

Ukoliko bi se statusni signali (prediktori) uređaja HT, LV, CV, RV, CP iz tablice zamijenili sa upravljačkim signalima tih istih uređaja, dobili bi iste (aktivacijske) vrijednosti koje su neminovno prethodile statusnim signalima, a predstavljale bi vrijednosti korektivnih mjera, tj., zavisne varijable. U tom slučaju, imajući u vidu upravljački signal kao okidač opservacije, prilikom *setiranja* vrijednosti bilo kojeg upravljačkog signala, registriraju se i svi ostali upravljački signali koji u tom trenutku imaju vrijednost nula (za zavisne kategorijske varijable). Kada bi, primjerice postojalo 10 digitalnih upravljačkih signala i svi se *setirali* u podjednakim omjerima, dobili bi skup podataka u kojemu je omjer klasa svake varijable 1:10. U realnosti je taj omjer daleko nepovoljniji imajući u vidu da se pojedini upravljački signali vrlo rijetko aktiviraju (resetiranje radnih sati uređaja, čišćenje filtera, kontrole recirkulacijske pumpe). Navedeno pruža objašnjenje minornih promjena rezultata nakon primjene *oversampling* tehnike kojom su se selektivno multiplicirale opservacije podatkovnog okvira kako bi se izjednačile klase svake od varijabli. U konačnici povećanjem broja opservacija nije bilo moguće postići bolji omjer klasa između kategorijskih zavisnih varijabli, budući su se multipliciranjem opservacija sa manjinskom klasom (pojedinačno za svaku zavisnu varijablu), također multiplicirale i većinske klase preostalih zavisnih varijabli.

Da bi se postigli bolji rezultati za oba modela (regresija i klasifikacija) nužno je, kao što je već rečeno, povećati broj opservacija. Problem nebalansiranog skupa podataka se time neće riješiti, te ga značajnije neće riješiti ni sinkroni način uzorkovanja koji može čak dodatno povećati nesrazmjer klasa uz dodatne cikličke opservacije u trenutcima kada ne postoji niti jedan aktivan upravljački signal (manjinska klasa zavisnih varijabli). Moguće rješenje je uvođenje trendova kretanja vrijednosti prediktora kao novih

nezavisnih varijabli, čime se može postići dodatna značajnija korelacija između prediktorskih i kriterijskih varijabli.

U konačnici, uz veći broj opservacija, te primjenom višestruke linearne regresije za pojedinačnu predikciju zavisnih varijabli (i ekvivalentnog modela SVM klasifikacije) umjesto više-izlaznih modela, moguće je primijeniti *undersampling* čime bi se klase izjednačile bez dupliciranja manjinske klase uz održavanje dovoljnog broja opservacija.

7. Zaključak

Temeljna ideja predmetnog rada bila je pružanje potpore operateru u kontroli proizvodnog procesa na način da se prati rad svih operatera, te izdvoji onaj operater čije su aktivnosti (korektivne mjere) polučile željene rezultate u donosu na definirane metrike kontrole sustava. Takve mjere su u simuliranom procesu izdvojene kao skup podataka na temelju kojeg je izvršena predikcija potrebnih korektivnih mjera s obzirom na stanje sustava u realnom vremenu. Time svaki od operatera, bez obzira na vlastito iskustvo i poznavanje praćenog procesa, ima mogućnost donijeti odluku koja je u skladu s najboljom praksom. Postignuti rezultati primjenom modela više-izlazne linearne regresije nadziranog strojnog učenja ukazuju da bi za postizanje željene primjene bio nužan je višestruko veći skup podataka od 15000 opservacija koliko je bilo na raspolaganju za testiranje. U slučaju primjene modela više-izlaznog Stroja potpornih vektora, konfuzijska je matrica prikazala nebalansiran skup podataka koji nije bilo moguće korigirati tehnikom *oversampling*-a. Uočeno je kako za postizanje potrebne točnosti predikcije postoji nužnost cikličkog uzorkovanja uz dodatne trendove kretanja vrijednosti prediktora, te primjena *undersampling*-a na znatno veći skup podataka.

Glede praćenja procesa u realnom vremenu, primijenjena metoda više-izlazne linearne regresije na podatkovni okvir od ukupno 73 varijabli i 10573 opservacija utrošila je 85,87 sec. za treniranje i 0.08 sec. za predikciju kriterijskih varijabli. Uzimajući u obzir potrebu za višestruko većim skupom podataka u cilju postizanja ženjenih rezultata, implementacija testiranih modela korištenjem resursa lokalnog ICS-a moguća je jedino na razini konačne predikcije, dok se za trening modela moraju uzeti u obzir dodatni računalni resursi.

8. Literatura

- Ademujimi, T. T., Brundage, M. P. and Prabhu, V. V. (2017) 'A Review of Current Machine Learning Techniques Used in Manufacturing Diagnosis', *IFIP Advances in Information and Communication Technology*, 513, pp. 407–415. doi: 10.1007/978-3-319-66923-6_48.
- Albertos, P. *et al.* (1999) 'Some issues on AI techniques in RT process control', *Annual Reviews in Control*, 23, pp. 125–137. doi: 10.1016/S1367-5788(99)00014-0.
- Cataldo, M. and De Souza, C. R. B. (2011) 'The Impact of API Complexity on Failures: An Empirical Analysis of Proprietary and Open Source Software Systems'.
- Correa, M., Bielza, C. and Pamies-Teixeira, J. (2009) 'Comparison of Bayesian networks and artificial neural networks for quality detection in a machining process', *Expert Systems with Applications*, 36(3 PART 2), pp. 7270–7279. doi: 10.1016/j.eswa.2008.09.024.
- Daoud, J. I. (2018) 'Multicollinearity and Regression Analysis', *Journal of Physics: Conference Series*, 949(1). doi: 10.1088/1742-6596/949/1/012009.
- Davis, C. M. *et al.* (2006) 'SCADA cyber security testbed development', in *2006 38th Annual North American Power Symposium, NAPS-2006 Proceedings*, pp. 483–488. doi: 10.1109/NAPS.2006.359615.
- Dembczy, K. (2013) 'Multi-Target Prediction'.
- Fovino, I. N. *et al.* (2010) 'An experimental platform for assessing SCADA vulnerabilities and countermeasures in power plants', in *3rd International Conference on Human System Interaction, HSI'2010 - Conference Proceedings*, pp. 679–686. doi: 10.1109/HSI.2010.5514494.
- GIC (2017) *Control room*. Available at: http://www.gich.com.hk/values/high_quality.aspx (Accessed: 2 May 2021).
- Golan, M., Cohen, Y. and Singer, G. (2020) 'A framework for operator– workstation interaction in Industry 4.0', *International Journal of Production Research*, 58(8), pp. 2421–2432. doi: 10.1080/00207543.2019.1639842.
- Goldin, E. *et al.* (2017) 'Cloud computing for big data analytics in the Process Control Industry', *2017 25th Mediterranean Conference on Control and Automation, MED 2017*, pp. 1373–1378. doi: 10.1109/MED.2017.7984310.

- Graham, M. H. (2003) 'Confronting Multicollinearity in Ecological', *Ecology*, 84(11), pp. 2809–2815.
- Hahn, A. *et al.* (2010) 'Development of the PowerCyber SCADA security testbed', in *ACM International Conference Proceeding Series*. New York, New York, USA: ACM Press, p. 1. doi: 10.1145/1852666.1852690.
- Helwig, N. E. (2017) 'Nonparametric Independence Tests Copyright c 2017 by Nathaniel E . Helwig', pp. 1–43.
- Hermann, M., Pentek, T. and Otto, B. (2016) 'Design principles for industrie 4.0 scenarios', in *Proceedings of the Annual Hawaii International Conference on System Sciences*. IEEE Computer Society, pp. 3928–3937. doi: 10.1109/HICSS.2016.488.
- Hidalgo, B. and Goodman, M. (2013) 'Multivariate or multivariable regression?', *American Journal of Public Health*. American Public Health Association Inc., pp. 39–40. doi: 10.2105/AJPH.2012.300897.
- Hourlier, S. (2021) 'For Our Complex Future, Don't Give Us AI, Give Us Intelligent Assistance (IA): The Case for Avionics', in *Advances in Intelligent Systems and Computing*. Springer, pp. 16–21. doi: 10.1007/978-3-030-51328-3_3.
- Jayawardena, N. (2020) *How to Deal with Imbalanced Data* . Available at: <https://towardsdatascience.com/how-to-deal-with-imbalanced-data-34ab7db9b100> (Accessed: 1 June 2021).
- Kang, Z., Catal, C. and Tekinerdogan, B. (2020) 'Machine learning applications in production lines: A systematic literature review', *Computers and Industrial Engineering*, 149(August). doi: 10.1016/j.cie.2020.106773.
- Kim, K. *et al.* (2011) *Use of SCADA Data for Failure Detection in Wind Turbines*. Available at: <http://www.osti.gov/bridge> (Accessed: 2 May 2021).
- LAMB (2021) *Control room electric utility*. Available at: https://line.17qq.com/articles/aqwurcqpx_p3.html (Accessed: 3 May 2021).
- Leitão, P., Colombo, A. W. and Karnouskos, S. (2016) 'Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges', *Computers in Industry*, 81, pp. 11–25. doi: 10.1016/j.compind.2015.08.004.
- Lu, Y. (2016) 'Industrial Integration: A Literature Review', *Journal of Industrial Integration and Management*, 01(02), p. 1650007. doi: 10.1142/s242486221650007x.

- Mandrekar, J. N. (2010) 'Receiver operating characteristic curve in diagnostic test assessment', *Journal of Thoracic Oncology*, 5(9), pp. 1315–1316. doi: 10.1097/JTO.0b013e3181ec173d.
- Maynard, P., McLaughlin, K. and Sezer, S. (2018) 'An Open Framework for Deploying Experimental SCADA Testbed Networks', (2016), pp. 92–101. doi: 10.14236/ewic/ics2018.11.
- Mehta, S. S. *et al.* (2012) 'A control architecture for human-machine interaction in the presence of unreliable automation and operator cognitive limitations', *AIAA Guidance, Navigation, and Control Conference 2012*, (August), pp. 1–12. doi: 10.2514/6.2012-4543.
- Pascual, C. (2018) *Understanding Linear Regression and Regression Error Metrics*. Available at: <https://www.dataquest.io/blog/understanding-regression-error-metrics/> (Accessed: 2 June 2021).
- Pokorni, B. *et al.* (2021) 'Development of a Holistic Method to Implement Artificial Intelligence in Manufacturing Areas', in *Advances in Intelligent Systems and Computing*. Springer, pp. 3–8. doi: 10.1007/978-3-030-51328-3_1.
- PRIMO.ai (2019) *Kernel Trick - PRIMO.ai*. Available at: http://primo.ai/index.php?title=Kernel_Trick (Accessed: 21 May 2021).
- Qin, S. J. and Chiang, L. H. (2019) 'Advances and opportunities in machine learning for process data analytics', *Computers and Chemical Engineering*, 126, pp. 465–473. doi: 10.1016/j.compchemeng.2019.04.003.
- Queiroz, C. *et al.* (2009) 'Building a SCADA security testbed', *NSS 2009 - Network and System Security*, pp. 357–364. doi: 10.1109/NSS.2009.82.
- Reaves, B. and Morris, T. (2012) 'An open virtual testbed for industrial control system security research', *International Journal of Information Security*, 11(4), pp. 215–229. doi: 10.1007/s10207-012-0164-7.
- Reguera-Bakhache, D. *et al.* (2020) 'Data-Driven Industrial Human-Machine Interface Temporal Adaptation for Process Optimization', *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2020-Septe, pp. 518–525. doi: 10.1109/ETFA46521.2020.9211930.
- Rehse, J. R., Mehdiyev, N. and Fettke, P. (2019) 'Towards Explainable Process Predictions for Industry 4.0 in the DFKI-Smart-Lego-Factory', *KI - Kunstliche Intelligenz*, 33(2), pp. 181–187. doi: 10.1007/s13218-019-00586-1.
- van Rensburg, N. J., Kamin, L. and Davis, S. (2019) 'Using machine learning-based

- predictive models to enable preventative maintenance and prevent ESP downtime', *Society of Petroleum Engineers - Abu Dhabi International Petroleum Exhibition and Conference 2019, ADIP 2019*. doi: 10.2118/197146-ms.
- Settemsdal, S. (2019) 'Machine learning and artificial intelligence as a complement to condition monitoring in a predictive maintenance setting', *Society of Petroleum Engineers - SPE Oil and Gas India Conference and Exhibition 2019, OGIC 2019*. doi: 10.2118/194590-ms.
- Shi, Q. *et al.* (2019) 'Minimalist and multi-functional human machine interface (HMI) using a flexible wearable triboelectric patch', *Nano Energy*, 62, pp. 355–366. doi: 10.1016/j.nanoen.2019.05.033.
- Stanula, P., Ziegenbein, A. and Metternich, J. (2018) 'Machine learning algorithms in production: A guideline for efficient data source selection', *Procedia CIRP*, 78, pp. 261–266. doi: 10.1016/j.procir.2018.08.177.
- Stouffer, K. *et al.* (2015) 'Guide to Industrial Control Systems (ICS) Security NIST Special Publication 800-82 Revision 2', *NIST Special Publication 800-82 rev 2*, pp. 1–157. Available at: [http://industryconsulting.org/pdfFiles/NIST Draft-SP800-82.pdf](http://industryconsulting.org/pdfFiles/NIST_Draft-SP800-82.pdf).
- Trend Micro Inc. (2021) *Industrial Control System - Definition*. Available at: <https://www.trendmicro.com/vinfo/us/security/definition/industrial-control-system> (Accessed: 1 May 2021).
- Wikipedia (2021) *Industrial control system - Wikipedia*. Available at: https://en.wikipedia.org/wiki/Industrial_control_system (Accessed: 1 May 2021).
- Wuest, T. *et al.* (2016) 'Machine learning in manufacturing: Advantages, challenges, and applications', *Production and Manufacturing Research*, 4(1), pp. 23–45. doi: 10.1080/21693277.2016.1192517.
- Yang, Y. *et al.* (2015) 'Cybersecurity test-bed for IEC 61850 based smart substations', *IEEE Power and Energy Society General Meeting, 2015-Septe*, pp. 1–5. doi: 10.1109/PESGM.2015.7286357.
- Zhang, G. (2018) *What is the kernel trick*. Available at: <https://medium.com/@zxr.nju/what-is-the-kernel-trick-why-is-it-important-98a98db0961d> (Accessed: 21 May 2021).
- Zhang, Z. J. *et al.* (2015) 'A survey of SCADA test bed', *International Journal of Wireless and Mobile Computing*, 8(1), pp. 9–14. doi: 10.1504/IJWMC.2015.066754.

9. Popis slika

Slika 1: Industrijski kontrolni sustav (Trend Micro Inc., 2021)	2
Slika 2: Kontrolna soba za nadzor proizvodnog procesa – HMI (GIC, 2017)	6
Slika 3: Kontrolna soba za nadzor elektro-energetskog sustava (LAMB, 2021)	7
Slika 4: Distribucija metoda strojnog učenja (Kang, Catal and Tekinerdogan, 2020)	10
Slika 5: Ispitni SCADA sustav za pojednostavljeni WTP proces	15
Slika 6: Performanse sustava u odnosu na prisustvo operatera	16
Slika 7: OWI 4.0 <i>framework</i>	17
Slika 8: SCADA sustav hidrauličke jedinice	20
Slika 9: Tok procesnih podataka unutar SCADA sustava (Stouffer <i>et al.</i> , 2015)	21
Slika 10: Struktura ispitnog SCADA sustava	23
Slika 11: Korištene komponente WinCC razvojnog okružja	24
Slika 12: <i>Tag management</i> komponenta WinCC razvojnog okružja	25
Slika 13: Hidraulička jedinica u normalnom režimu rada	26
Slika 14: Hidraulička jedinica u režimu rada izvan granica tolerancije	26
Slika 15: Programski modul <i>Actions</i> VBS editora	27
Slika 16: Projektni moduli simulacije	28
Slika 17: Shema eksterne baze podataka	29
Slika 18: Detalj ulazne grane sa parametrima	31
Slika 19: <i>Sequence</i> dijagram simulacije procesa	33
Slika 20: Algoritam simulacije operatera 2: <i>Soft control</i>	38
Slika 21: HMI sučelje: selekcija operatera	39
Slika 22: Podatkovni okvir (<i>dataframe</i>)	42
Slika 23: MvLR (<i>Multivariate Linear Regression</i>) model (Helwig, 2017)	50
Slika 24: Primjena MvLR i MvLC na skup podataka (Dembczy, 2013)	50
Slika 25: Stroj potpornih vektora: <i>Kernel trick</i> (PRIMO.ai, 2019)	52
Slika 26: Trenutno stanje procesa	62

10. Popis tablica

Tablica 1: Prednosti i nedostaci metoda strojnog učenja (Ademujimi, Brundage and Prabhu, 2017) (vlastiti prijevod)	8
Tablica 2: Problemi, kategorije i primijenjene metode (Kang, Catal and Tekinerdogan, 2020), (vlastiti prijevod)	9

Tablica 3: Kategorije ispitnih SCADA sustava (Zhang <i>et al.</i> , 2015) (vlastiti prijevod)	13
Tablica 4: Usporedba kategorija komponenti ispitnih SCADA sustava.....	14
Tablica 5: Softverska konfiguracija ispitnog SCADA sustava	23
Tablica 6: Parametri odvijanja i kontrole procesa	30
Tablica 7: Izlazni parametri procesa	31
Tablica 8: Identifikatori korišteni u simulaciji procesa	35
Tablica 9: VIF interpretacija (Daoud, 2018)	47
Tablica 10: Rezultati modela više-izlazne linearne regresije	54
Tablica 11: Rezultati modela više-izlazne linearne regresije na reduciranom setu...	55
Tablica 12: Rezultati više-izlaznog SVM modela za reducirani podatkovni okvir	58
Tablica 13: Rezultati SVM modela za reducirani podatkovni okvir uz <i>oversampling</i> ...	59
Tablica 14: Primjer mogućih vrijednost signala trenutnog stanja procesa	62

11. Popis Grafikona

Grafikon 1: Korelogram nezavisnih varijabli.....	44
Grafikon 2: Korelogram zavisnih varijabli.....	45
Grafikon 3: Faktor inflacije varijance.....	47
Grafikon 4: <i>Box-plot</i> kontinuiranih nezavisnih varijabli.....	48
Grafikon 5: Stroj potpornih vektora: dvodimenzionalni prostor	51
Grafikon 6: : Izdvojene nezavisne varijable sa visokim VIF vrijednostima	54
Grafikon 7: : Relativna važnost prediktora za skup podataka operatera 2.....	55
Grafikon 8: Trend vrijednosti regresijskih metrika porastom broja opservacija	56
Grafikon 9: Broj aktivacija upravljačkih signala (zavisnih varijabli)	57
Grafikon 10: Nebalansirani podatkovni okvir	57

12. Sažetak

Razina kompleksnost industrijskih kontrolnih sustava neprestano se povećava tijekom posljednjih desetljeća. To je uglavnom slučaj zbog primjene novih tehnologija s krajnjim ciljem postizanja boljih rezultata produktivnosti poboljšanjem kontrole procesa, dostupnosti sustava, pouzdanosti i fleksibilnosti. Sa tim ciljem, u eri koncepta Industrije 4.0, granice su pomaknute prema *Big data*, industrijskom Internetu stvari (IIoT), strojnom učenju i heterogenim mrežama. Kao posljedica navedenog, sustavi nadzora,

kontrola i prikupljanja podataka (SCADA) upravljaju s više podataka, imaju više funkcija i postaju složeniji za upotrebu. To možda nije problem za industrijske grane u kojima razina automatizacije omogućuje kontrolu procesa bez ljudi. Ali u slučaju industrijskih grana s kontinuiranim procesima poput industrije čelika s ulančanim pogonima kontinuiranog lijevanja, valjaonica, galvanizacijskih linija i valjačkih stanova, gdje se očekuje određena stohastička komponenta, operateri su i dalje nezamjenjivi. Za takva postrojenja kompetencija i iskustvo operatera presudni su u održavanju stabilnosti i kontinuiteta procesa u vremenu. Ovaj se rad fokusira na podršku operateru primjenom nadziranih modela strojnog učenja više-izlazne linearne regresije i modela potpornih vektora koji se koriste za predikciju najučinkovitijeg odgovora operatera za određeno stanje procesa u stvarnom vremenu, putem analize skupa podataka sa aktivnostima svih operatera u odnosu na kontrolirani proizvodni proces te izdvajanja najučinkovitijeg operatera koji će se koristiti kao referenca.

Ključne riječi: scada, hmi, klasifikacija, regresija, stroj potpornih vektora, više-izlazna, kontinuirani, proizvodni, proces, kontrola, industrija, predikcija, operater

13. Abstract

Complexity of industrial control systems has been increasing constantly over past decades. This is mostly the case due to implementation of new technologies with the ultimate goal of achieving higher productivity results by improving process control, system availability, reliability and flexibility. Following this goal, in the era of Industry 4.0 concept, boundaries have been pushed toward Big data, Industrial internet of things, machine learning and heterogeneous networks. As a consequence, supervisory control and data acquisition (SCADA) systems manage more data, have more functions and become more complex to use. This might not be the issue for some industries where level of automation allows unmanned process control. But in the case of industries with continuous stream processes like the steel industry with chained processes of continuous casting, rolling mills, galvanizing lines, and wire rods mills where a certain stochastic component is expected, operators are still irreplaceable. For those plants, an operator's competence and experience is crucial in maintaining process stability and continuity over time. This work focuses on operator decision support by implementing supervised machine learning models of multi-output

linear regression and support vector machine used for predicting the most efficient operator's response for a given process state in real time, achieved by analysing dataset with all operator activities to the controlled production process, and allocating the most efficient operator to be used as a reference.

Keywords: scada, hmi, classification, regression, svm, multi-output, stream, process, control, production, industry, prediction, operator