

Program

Digitalna transformacija robotiziranih tovarnih prihodnosti

DIGITOP

Rezultat D1.1:
**Metodološke osnove za napovedovanje in upravljanje
porabe energije**

Projekt/RRP	DIGITOP/RRP1.2
Vrsta dokumenta	Poročilo
Verzija dokumenta	1.0
Sodelujoči partnerji	IJS-E8, IJS-E2, ULFE-LAK
Avtorji	Aljaž Osojnik, Martin Žnidaršič, Bernard Ženko, Matevž Bošnak, Simon Tomažič, Miha Glavan, Žiga Stržinar
Datum priprave	31. 7. 2024

Kazalo

1	POVZETEK	3
2	IDENTIFIKACIJA PROBLEMA.....	3
3	METODOLOŠKI PRISTOPI.....	5
3.1	ARIMA	5
3.2	REKURENTNE NEVRONSKE MREŽE	6
3.3	IZGRADNJA ZNAČILK IN UPORABA KLASIČNIH PRISTOPOV STROJNEGA UČENJA	7
3.4	REKURZIVNO NAPOVEDOVANJE	8
4	VZORČNI PRIMER	8
4.1	PODATKI.....	8
4.2	PARAMETRI NAPOVEDOVANJA	9
4.3	VREDNOTENJE NAPOVEDNIH MODELOV	9
4.4	PRIPRAVA PODATKOV IN UČENJE RNN.....	10
4.5	PRIPRAVA PODATKOV IN UČENJE S KLASIČNIMI PRISTOPI STROJNEGA UČENJA	11
5	ZAKLJUČKI IN PRIPOROČENA METODOLOGIJA	11
	BIBLIOGRAFIJA	12

1 Povzetek

V tem poročilu predstavimo metodološke osnove za napovedovanje in upravljanje porabe električne energije, ki temeljijo na uporabi metod umetne inteligence. Začnemo z identifikacijo problema, kjer orišemo proces modeliranja in napovedovanja porabe električne energije, ki se nato lahko uporablja kot orodje za energetske učinkovitejše načrtovanje in upravljanje trenutnih in bodočih proizvodnih procesov. Pri tem poudarimo, da so podatki o porabi električne energije pogosto zajeti kot časovne vrste. Identificiramo tudi različne parametre učno-napovednega postopka.

V nadaljevanju predstavimo tri različne pogoste pristope, ki se uporabljajo za modeliranje in napovedovanje časovnih vrst, ARIMA metode, rekurentne nevronske mreže ter uporabo klasičnih pristopov strojnega učenja. Za napovedovanje porabe električne energije predlagamo uporabo rekurentnih nevronskih mrež ali klasičnih pristopov strojnega učenja, saj problem napovedovanja porabe ne zadostuje predpostavkam metod ARIMA.

Vzorčno uporabo predlaganih metod prikažemo na podatkih pridobljenih s strani partnerja Kolektor Sisteh. Poleg opisa podatkov prikažemo tudi vzorčni način ocenjevanja napovednih modelov. Opišemo še nastavitve parametrov predlaganih pristopov ter prikažemo eksperimentalne rezultate naučenih modelov. Poročilo zaključimo s predlaganim postopkom za napovedovanje porabe električne energije.

2 Identifikacija problema

Pri prožni proizvodnji se proizvodni načrti pogosto spreminjajo, s tem pa se spremenijo tudi časovni profili porabe energije. Za uspešno energetske načrtovanje proizvodnje je tako pomembno oceniti oziroma napovedati spremembe profilov porabe energije pri morebitnih spremembah v proizvodnji. Napovedovanje porabe energije je mogoče izvesti na več načinov, na primer, na podlagi fizikalnih lastnosti naprav in procesov, ki v proizvodnji nastopajo, ali pa, na primer, na podlagi preteklih meritev porabe električne energije različnih komponent proizvodnega procesa. To poročilo predstavi metodološke pristope, predvsem pristope, ki temeljijo na uporabi metod umetne inteligence (UI), za napovedovanje porabe električne energije.

Metode UI za napovedovanje uporabljajo pristope strojnega učenja in tipično temeljijo na zgodovinskih meritvah, iz katerih naučijo napovedne modele, ki lahko napovedujejo vrednosti opazovanih količin v prihodnosti. Napovedni model naučen na zgodovinskih podatkih o porabi električne energije se nato lahko uporablja kot orodje za ocenjevanje porabe električne energije načrtovanih ali bodočih proizvodnih procesov.

Za uporabo metode UI ni eksplicitno pomembno ali modeliramo porabo električne energije ene proizvodne naprave, večjega dela proizvodnega procesa ali celo celotnega proizvodnega procesa. Vendar pa bodo dinamike večjih delov bolj kompleksne, kar pomeni, da bomo potrebovali več virov (predvsem večje količine podatkov), da bomo dinamiko uspeli uspešno napovedovati.

Pri napovedovanju porabe električne energije se pogosto orientiramo na obračunske intervale, po katerih porabljeno električno energijo upravnik proizvodnega procesa plačuje, saj s tem obenem pridobimo še oceno cene porabljene energije načrtovanega procesa, če poznamo cene električne energije v času poteka procesa.

Pogosto so tudi meritve na podlagi katerih modeliramo merjene natanko ob obračunskih intervalih, vendar to ni nujen predpogoj za uporabo metod UI.

Podatke o porabi električne energije tako obravnavamo kot časovno vrsto (Hyndman & Athanasopoulos, 2018), tj., zaporedje parov oblike: (čas meritve, meritev porabe električne energije). Kot vhod za modeliranje torej vzamemo zgodovinske meritve porabe električne energije; kot vhod za posamezno napoved v danem bodočem času pa zgodovinske meritve iz časovnega okna stalne dolžine neposredno pred podanim časom. Na primer: iz zgodovinskih podatkov zadnjih 24 ur želimo napovedati, kakšna bo poraba v naslednji uri. Širina časovnega okna zgodovinskih podatkov iz katerih napovedujemo (v tem primeru 24 ur) je torej eden izmed parametrov modeliranja.

Napovedni model tako praviloma napoveduje le vrednosti, ki neposredno sledijo že izmerjenim vrednostim, tj., če želimo napovedati vrednost ob času t_0 , potrebujemo meritve ob časih $t_{-1}, t_{-2}, \dots, t_{-n}$, ki so razporejene v vhodnem časovnem oknu. Za potrebe načrtovanja proizvodnje take napovedi nosijo nizko informativno vrednost saj želimo načrtovati procese, ki so daljši od enega časovnega (obračunskega) intervala. Za nek načrtovani proces nas morebiti zanima, kakšen bo profil porabe električne energije v naslednjih 24 ali 48 urah, ne le v naslednji uri. Da pridobimo napovedi za daljša časovna obdobja, lahko uporabljamo različne pristope, ki so opisani v nadaljevanju poročila. Širino napovednega intervala zato identificiramo kot enega izmed parametrov modeliranja.

Pomemben faktor pri izbiri metodologije za napovedovanje je tudi količina zgodovinskih (učnih) podatkov na podlagi katerih naučimo model. Nekateri UI pristopi zahtevajo velike količine učnih podatkov, da dosežejo dobre rezultate, medtem ko so drugi manj občutljivi na količino učnih podatkov.

Za izboljšanje napovedi lahko uporabimo tudi informacije oz. meritve o drugih količinah, ki lahko vplivajo na porabo električne energije; v tem primeru imamo opravka z multivariabilnimi časovnimi vrstami. Primeri takih meritev so na primer vremenski oz. okoljski pogoji, kot so temperatura, pritisk ali vlažnost, pri procesih kjer te vplivajo na vhodni material ali sam proizvodni proces; ali pa program, ki ga izvaja določena naprava, če modeliramo njeno porabo električne energije. Take meritve so praviloma specifične za določeno napravo ali proizvodni proces, tako da v splošnem ne moremo pričakovati, da so na voljo. Če pa imamo med procesom modeliranja na voljo dodatne meritve, jih lahko metode UI za napovedovanje upoštevajo in s tem potencialno izboljšajo napovedne modele. Koristnost upoštevanja dodatnih meritev je jasno odvisna od tega, koliko merjene količine vplivajo na napravo/proces in, posledično, na porabo električne energije.

Za napovedovanje porabe električne energije naprave/procesa z metodami UI torej moramo:

- pridobiti zgodovinske meritve porabe električne energije,
- določiti parametre napovedovanja:
- časovno širino okna vhodnih meritev uporabljenih za napovedovanje,
- širino napovednega intervala; in
- (opcijsko) pridobiti zgodovinske meritve dodatnih količin, ki vplivajo na porabo električne energije.

Na podlagi teh informacij se nato odločimo za različne pristope modeliranja z UI, ki so opisani v sledečem razdelku.

3 Metodološki pristopi

Napovedovanje porabe električne energije je problem, ki je že dolgo relevanten. V ta namen so se uporabljali različni pristopi, od eksergijskih pristopov (Duflou, et al., 2012), pristopov na osnovi nominalnih specifikacij naprava v proizvodnem procesu (Thiede, Bogdanski & Herrmann, 2012) in simulacij (Dietmar, Eberspaecher & Verl, 2009, Dietmar & Verl 2009) do uporabe empiričnih metod, kjer se porabo modelira neposredno v odvisnosti parametrov proizvodnega procesa (Kara & Li, 2011, Qureshi, et al. 2012). Obstajajo tudi kompleksnejši pristopi, ki modeliranje porabe zakodirajo v odločitveno drevo, ter na podlagi opisa naprave oz. procesa izberejo ustrezno pod-metodologijo (Schmidt, et al., 2015). Natančnejši pregled preteklih metodologij je na voljo v rezultatu D4.1.

Za napovedovanje časovnih vrst na podlagi podatkov z metodami UI lahko ločimo 2 splošna pristopa: (1) **napovedovanje bodoče vrednosti neposredno iz vhodnih časovnih vrst**, tj., vsako napoved neposredno izračunamo iz dela časovne vrste, katere širina je natanko enaka širini vhodnega okna; in (2) **iz časovnih vrst izračunamo značilke, na podlagi katerih naučimo model**. Pri prvem pristopu so podatki na katerih se učimo izseki časovne vrste, ki jo modeliramo; pri drugem pa te podatke uporabimo za izračun značilk kot so, npr., povprečja, minimumi, maksimumi, kvartili, ipd. Vrednosti teh značilk nato zberemo v tabelarni obliki in jih uporabimo za učenje.

V tem poročilu opišemo tri splošne pristope, ki se uporabljajo za modeliranje časovnih vrst: metodo ARIMA (ter njeni razširitvi SARIMA in ARIMAX), ki temelji na statističnem pristopu; rekurentne nevronske mreže (RNN) in uporabo klasičnih metod strojnega učenja. Glede na razporeditev zgoraj, ARIMA in RNN delujeta neposredno na meritvah časovne vrste, medtem ko klasične metode zahtevajo pripravo ustreznih značilk.

Poleg metodologij za učenje napovednih modelov, na koncu izseka predstavimo še način za razširitev napovednega okna napovednih modelov.

3.1 ARIMA

Ena izmed najbolj popularnih metod za napovedovanje časovnih vrst je avtoregresijska metoda ARIMA in njene razširitve SARIMA, ARIMAX, ter SARIMAX (Box & Jenkins, 1970). ARIMA bodoče vrednosti časovne vrste napoveduje neposredno iz preteklih vrednosti in preteklih napak v napovedih.

Metoda ARIMA je sestavljena iz treh komponent: avtoregresije (AR), integracije (I) ter pomičnega povprečja (angleško: *moving average*; MA). Avtoregresijska komponenta zajema povezavo med meritvijo v nekem časovnem intervalu ter meritvami v vnaprej določenem številu preteklih meritev, tj., zgodovinskih meritev. Komponenta pomičnega povprečja modelira povezavo med meritvijo in ostankom napake med pomičnim povprečjem zgodovinskih meritev. Komponenti AR in MA predpostavita, da so podatki stacionarni, tj., da imajo podatki statično povprečje, varianco in avtokorelacije. Integracijska komponenta stacionarnost doseže tako, da meritve diferencira, tj., namesto meritev samih obravnavamo razlike med zaporednimi meritvami. AR in MA komponenti namreč privzemata stacionarnost.

Poznamo še dve razširitvi metode ARIMA: SARIMA, ki modelira še sezonske vplive na vrednosti v časovni vrsti, ter ARIMAX, ki poleg meritev časovne vrste, pri modeliranju upošteva še, t.i., eksogene faktorje (v primeru napovedovanja porabe električne energije bi bile to morebitne dodatne spremenljivke). Možna je tudi uporaba obeh razširitev hkrati – v tem primeru metodo imenujemo SARIMAX.

Da lahko metode iz družine ARIMA uspešno uporabimo za napovedovanje časovnih vrst, morajo imeti modelirani podatki sledeče lastnosti:

- stacionarnost: povprečje, varianca in avtokorelacija časovne vrste se ne spreminja. Stacionarnost lahko do neke mere doseže integracijska komponenta z višjim redom diferenciacije; če časovna vrsta tudi po več korakih diferenciacije ni stacionarna, uporaba metode ARIMA ni priporočljiva.
- avtokorelacija: ker ARIMA napoveduje bodoče vrednosti na podlagi preteklih vrednosti, privzemamo da so bodoče vrednosti avtokorelirane s preteklimi vrednostmi.
- linearnost: metoda ARIMA privzema, da so bodoče vrednosti v linearni odvisnosti od preteklih vrednosti.

Družina metod ARIMA zaradi zgornjih zahtev ni najboljša za napovedovanje porabe električne energije, še posebej v kontekstu planiranja proizvodnih procesov. Z načrtovanimi spremembami v proizvodnem procesu namreč neposredno vplivamo na stacionarnost časovne vrste, še posebej kadar načrtujemo spremembe, ki v preteklosti niso bile zabeležene.

3.2 Rekurentne nevronske mreže

Za modeliranje podatkov, ki si sledijo v zaporedju, na primer časovnem, so zelo primerne tako-imenovane rekurentne nevronske mreže (angleško: *recurrent neural networks*; RNN) (Elman, 1990), med katerimi posebej izpostavljamo nevronske mreže z dolgim kratkoročnim spominom (angleško: *long short-term memory neural networks*; LSTM) (Hochreiter & Schmidhuber, 1997). Pri časovno urejenih podatkih so vrednosti določenih atributov-značilk odvisne tudi od lastnih preteklih vrednosti in ne le od vrednosti drugih značilk istega primera. Rekurentne mreže omogočajo izkoriščanje te informacije zaradi svoje posebne strukture. Za razliko od običajne strukture nevronske mreže, pri katerih so vhodni nivoji povezani s skritimi, ti pa med seboj in z izhodnimi, imajo rekurentne nevronske mreže izhodne nivoje povezane tudi s skritimi nivoji naslednje tovrstne serije nivojev, ki na vходу sprejema vhode naslednjega podatka, in tako naprej. Taka serija nivojev je lahko različno dolga in običajno nas zanima le izhodni nivo zadnje serije, čeprav so na voljo tudi vsi vmesni. Na nek način tako mreža lahko modelira omejen pretekli kontekst ali spomin vhodnih značilk in vmesnih rezultatov.

Za dolge časovne kontekste morajo biti rekurentne nevronske mreže zelo velike, kar pomeni precej bolj računsko zahtevno (dolgotrajno) učenje. Še bolj problematična posledica velikih mrež pa je zamiranje ali ojačanje (odvisno od realizacije učenja) vpliva primerov na učne parametre. Posebna vrsta rekurentnih nevronske mreže, ki se tem težavam običajno uspešno izogne, so nevronske mreže z dolgim kratkoročnim spominom ali s tujko na kratko LSTM mreže, ki v strukturi predvidevajo posebna vozlišča, ki deloma ali povsem onemogočajo večje ali manjše podstrukture mreže. S tem se izognejo zamiranju ali ojačanju vpliva primerov na parametre, njihovo učenje pa je kljub temu računsko zelo zahtevno.

Podatki, ki so uporabljeni pri napovedovanju in upravljanju porabe energije, so običajno časovno urejeni in vzorci, ki jih pričakujemo, zajemajo tudi dolg pretekli kontekst ali spomin. V takih problemih je zelo priporočljiva uporaba nevronske mreže z dolgim kratkoročnim spominom, če nam le razpoložljivi računski viri to omogočajo.

3.3 Izgradnja značilk in uporaba klasičnih pristopov strojnega učenja

Čeprav je uporaba nevronske mreže vse pogostejša, predvsem zaradi točnosti naučenih modelov, je preudarno preveriti tudi uporabo klasičnih pristopov strojnega učenja. Nabor algoritmov strojnega učenja namreč ponuja raznolike učne pristope, ki učijo različne vrste modelov z različnimi lastnostmi. Pogosto so, denimo, želene lastnosti sposobnost učenja iz manjšega števila primerov (če modeliramo proizvodnjo butičnih izdelkov), ter interpretabilnosti modelov, kadar želimo razumeti na kakšen način je model napoved izračunal. V primeru napovedovanja porabe električne energije je zanimivo predvsem učenje iz manjšega števila primerov.

Prvi pristop za pripravo vhodnih vrednosti za napovedovanje časovnih vrst je podajanje zakasnjene vrednosti signalov (tako vhodnih kot ciljnih) kot vhodne vrednosti. V primeru preprostih odvisnosti ta pristop lahko zadostuje že za linearni model, kadar so odvisnosti kompleksnejše, pa je ali smiselno nabor vhodnih vrednosti razširiti z izračunanimi značilkami.

Za namene napovedovanja porabe električne energije se večinoma uporablja, t.i., nadzorovano učenje, kjer je med učnim procesom znana prava poraba. Klasični pristopi strojnega učenja praviloma potrebujemo podatke v tabelarni obliki, tj., meritve več spremenljivk ob različnih pogojih (vhodne vrednosti), tem meritvam pa pripišemo ciljno (izhodno) vrednost, ki jo želimo napovedovati.

Ena izmed lastnosti, ki jo metode strojnega učenja izkoriščajo, je namreč neodvisnost spremenljivk, tj., različnih stolpcev v tabeli. Če časovno vrsto pretvorimo na zgoraj opisani način, ta lastnost ne bo zajela kompleksnih odvisnosti, saj so zaporedne meritve praviloma korelirane. Pristop, ki je vsaj delno uspešen pri uporabi podobnega postopka, je pristop zunanje dinamike (Nelles, 2001), vendar se ta opira na dejstvo, da je merjenih več časovnih vrst, na podlagi katerih lahko doseže neodvisnost. Glede na zastavljeno nalogo napovedovanja porabe električne energije, kjer opazujemo le eno časovno vrsto, ta pristop torej ni primeren.

Da pripravimo raznolike vhodne spremenljivke, ki so primerne za učenje, se lahko poslužimo postopka konstrukcije in računanja značilk. Za vsako časovno vrsto, ki jo obravnavamo, tako izračunamo preproste količine (ki na vrsto gledajo le kot nabor števil), kot so povprečna vrednost, varianca, kvartili, itd., kot tudi kompleksnejše vrednosti, ki upoštevajo tudi časovne komponente meritve, na primer, koeficiente Fourier-eve transformacije časovne vrste. Če imamo poleg izračunanih značilk iz časovne vrste na voljo še dodatne spremenljivke, le-te lahko dodamo v izbor značilk. Če so dodatne spremenljivke tudi časovne vrste, lahko tudi iz njih na enak način izračunamo dodatne značilke.

Za vsako časovno obdobje iz katerega se želimo učiti torej najprej identificiramo ustrezno časovno vrsto in iz nje izračunamo vrednosti značilk. Zaporedne vrednosti značilk nato zberemo v tabelo s katero lahko učimo s klasičnimi pristopi strojnega učenja. Podobno storimo tudi, ko želimo za neko časovno obdobje uporabiti

napovedni model, tj., za časovno obdobje za katerega želimo napoved identificiramo časovno vrsto, iz nje izračunamo značilke, in jih uporabimo kot vhod za naučeni napovedni model.

Ko podatke pretvorimo v primerno obliko, imamo na voljo širok razpon klasičnih metod strojnega učenja. Za napovedovanje porabe energije so primerne **regresijske metode**, tj., tiste, ki so namenjene napovedovanju zveznih (numeričnih) vrednosti. Izmed uspešnejših in širše uporabnih izpostavimo, na primer, metodo naključnih gozdov (Breiman, 2001), ter metodo XGBoost (Chen & Guestrin, 2016). Če pričakujemo, da je postopek, ki ga modeliramo časovno spremenljiv, je smiselna tudi uporaba metod, ki se učijo iz podatkovnih tokov, tj., podatkovnih virov, ki neusahljivo generirajo nove podatke. Kot primer takega pristopa, omenimo uporabo mehkih modelov (Ožbot, 2022) za opis dinamičnih procesov.

3.4 Rekurzivno napovedovanje

Da razširimo napovedno okno modela, ki napoveduje le za en časovni interval vnaprej, lahko uporabimo postopek rekurzivnega napovedovanja. Model, katerega napovedno okno širimo, za izračun napovedi kot vhod dobi časovno vrsto dolžine vhodnega okna, tj., kot vhodne meritve dobi vrednosti $x_1, \dots, x_n$ (kjer je n enak širini vhodnega okna), napoveduje pa vrednost časovne vrste v naslednjem intervalu $\hat{x}_{n+1}$ (opomba: s strešico $\hat{}$ bomo označili napovedane vrednosti). Model želimo uporabiti še za napovedovanje $\hat{x}_{n+2}, \dots, \hat{x}_{n+k}$, kjer je k širina napovednega okna.

Postopek, ki ga lahko uporabimo za širitev napovednega okna, je rekurzivno napovedovanje. Pri rekurzivnem napovedovanju najprej na podlagi vrednosti $x_1, \dots, x_n$ napovemo vrednost $\hat{x}_{n+1}$. Nato prvo meritev v vhodni časovni vrsti zavržemo, na konec pa pripišemo napovedano vrednost $\hat{x}_{n+1}$. Tako dobimo novo časovno vrsto, $x_2, \dots, x_n, \hat{x}_{n+1}$, ki je enake dolžine kot vhodno okno, ki jo podamo modelu, ki izračuna napoved $\hat{x}_{n+2}$. Ta postopek ponavljamo, dokler ne dobimo napovedi za $\hat{x}_{n+k}$.

Da bo rekurzivno napovedovanje uspešno, potrebujemo modele, ki so (1) natančni ter (2) robustni. Natančnost je pomembna zato, ker napovedi modela v postopku uporabljamo kot vhodne vrednosti. Če model ni natančen, se hitro zgodi, da napovedane vrednosti divergirajo od pravih vrednosti. Robustnost v tem kontekstu razumemo kot dobro pokritje vseh možnih vhodnih časovnih vrst v učni množici, tj., med učenjem je bilo obravnavanih veliko različnih stanj. Robusten model tako ne bo divergiral, tudi če se v napovednem oknu stanje spremeni (v eno izmed stanj na katerih je bil naučen).

4 Vzorčni primer

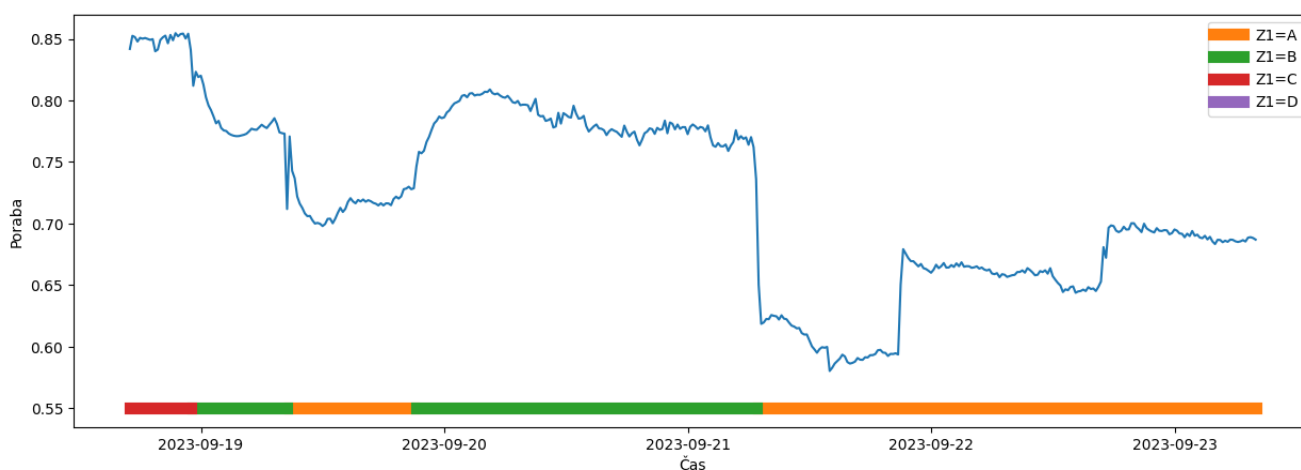
4.1 Podatki

Podatki na katerih bomo prikazali uporabo metod UI za napovedovanje porabe energije so bili pridobljeni s strani projektnega partnerja Kolektor Sisteh. Poleg porabe energije za del proizvodnega procesa vsebujejo še meritve ene dodatne spremenljivke, ki potencialno vpliva na porabo energije. Zaradi varovanja poslovnih skrivnosti so vse porabe normalizirane in izražene kot procent najvišje zabeležene porabe. Dodatno

spremenljivko bomo označili z anonimiziranim imenom "Z1", zavzema pa lahko štiri diskretne vrednosti, ki jih bomo označili kot "A", "B", "C" in "D".

Podatki vsebujejo precej meritev pri katerih proizvodni proces ni tekel, tj., podatkov o porabi energije ni oz. je enaka 0. Ti podatki so za učenje neuporabni ali celo moteči, zato smo identificirali najdaljši odsek podatkov, kjer proces neprekinjeno teče. Izbrani odsek podatkov pokriva časovni razpon slabih 5 dni, med 2023-09-18 17:00:00 in 2023-09-23 8:00:00.

Tako meritve o porabi energije, kot tudi meritve dodatne spremenljivke, smo pretvorili na 15 minutne časovne intervale, ki sovpadajo z obračunskimi intervali. Da smo meritve pretvorili na to časovno resolucijo smo izvedli povprečenje čez ustrezne meritve. Izbrani podatki so torej sestavljeni iz 445 15-minutnih intervalov, ki so prikazani na sliki spodaj.



4.2 Parametri napovedovanja

Glede na izbrane podatke za časovno širino okna vhodnih meritev uporabljenih za napovedovanje izberemo vrednost 24 ur, oz., $24 * 4 = 96$ meritev na 15 minutnih intervalih. Ker je časovni razpon izbranih podatkov kratek, ne pričakujemo, da bodo naučeni modeli dovolj robustni za rekurzivno napovedovanje. Zato za širino napovednega intervala izberemo en 15 minutni interval.

4.3 Vrednotenje napovednih modelov

Vrednotenje napovednih modelov mora biti vedno prilagojeno problemu, ki ga želimo reševati. Namen tega primera je prikazati uporabo različnih pristopov, bolj kot doseči visoko napovedno natančnost, čeprav bi bil v praktični uporabi to eden izmed glavnih kriterijev, ki bi jih želeli optimirati.

V splošnem pri uporabi metod UI za učenje napovednih modelov uporabljamo postopek prečnega preverjanja. Ker pa gre pri napovedovanju porabe energije za napovedovanje časovnih vrst ta mehanizem ni primeren, saj so podatki časovno urejeni, postopek prečnega preverjanja pa podatke premeša. Zato bomo za vrednotenje

naučenih modelov uporabili preprostejši postopek in sicer bomo del podatkov zadržali za vrednotenje. Zadnji dan izbranih meritev bomo tako označili kot testno množico na katerih bomo ovrednotili napovedi modelov, preostanek podatkov pa bomo imenovali učna množica. Testna množica je torej sestavljena iz 96 15-minutnih intervalov.

Za vrednotenje napovedi bomo uporabili koren povprečne kvadratne napake (angleško: *root mean squared error*; RMSE), definiran kot:

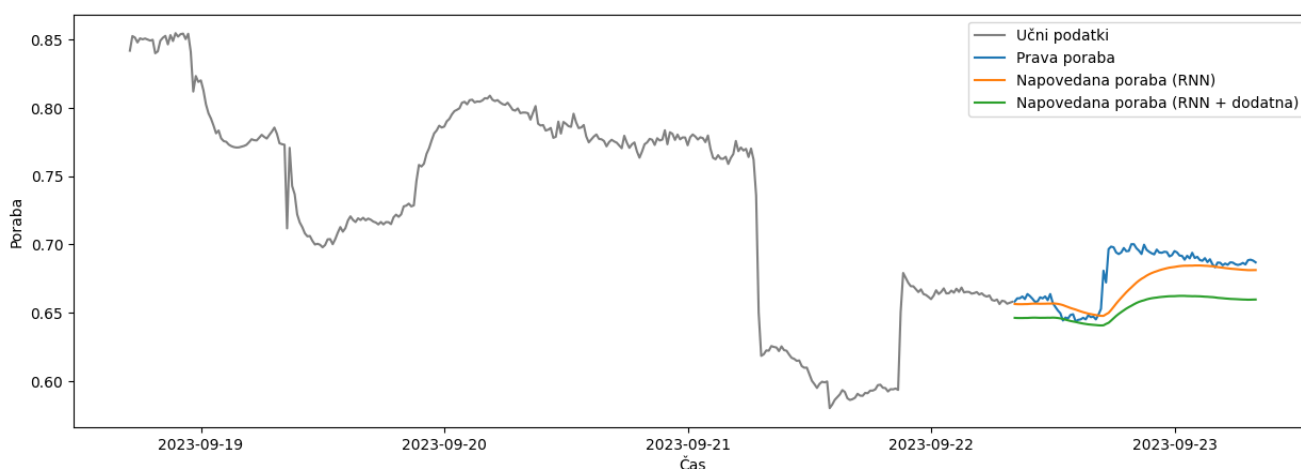
$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{x}_i - x_i)^2},$$

kjer je n velikost testne množice, i indeksira vrednosti v testni množici, x_i je prava vrednost, $\hat{x}_i$ pa napovedana vrednost.

4.4 Priprava podatkov in učenje RNN

Ker smo za dolžino okna vhodnih meritev izbrali 24 ur oz. 96 meritev, 349 meritev učne množice razporedimo v delne časovne vrste dolžine 96. Natančneje, prva delna časovna vrsta zajema natanko meritve 1 do 96, druga delna časovna vrsta zajema meritve 2 do 97, tretja 3 do 98, itd. S tem postopkom **drsečega okna** pripravimo 253 delnih časovnih vrst s katerih bomo naučili rekurentno nevronske mreže. Ko bomo upoštevali še dodatno spremenljivko, bomo vsaki izmed meritev dodali še ustrezne soležne meritve dodatne spremenljivke

Za učenje RNN bomo uporabili knjižnico **keras** (Chollet, 2015) v programskem jeziku Python 3. Nevronske mreže bomo sestavili iz 3 nivojev: (1) vhodnega, ki bo prejel delno časovno vrsto (ter morebitne dodatne spremenljivke); (2) nivoja LSTM z 200 notranjimi vozlišči; ter (3) izhodnega nivoja, ki bo napovedoval naslednjo vrednost. Prilagajanje števila notranjih vozlišč LSTM nivoja mreže omogoča boljše prilagajanje podatkom na račun potrebe po večjem številu učnih podatkov.



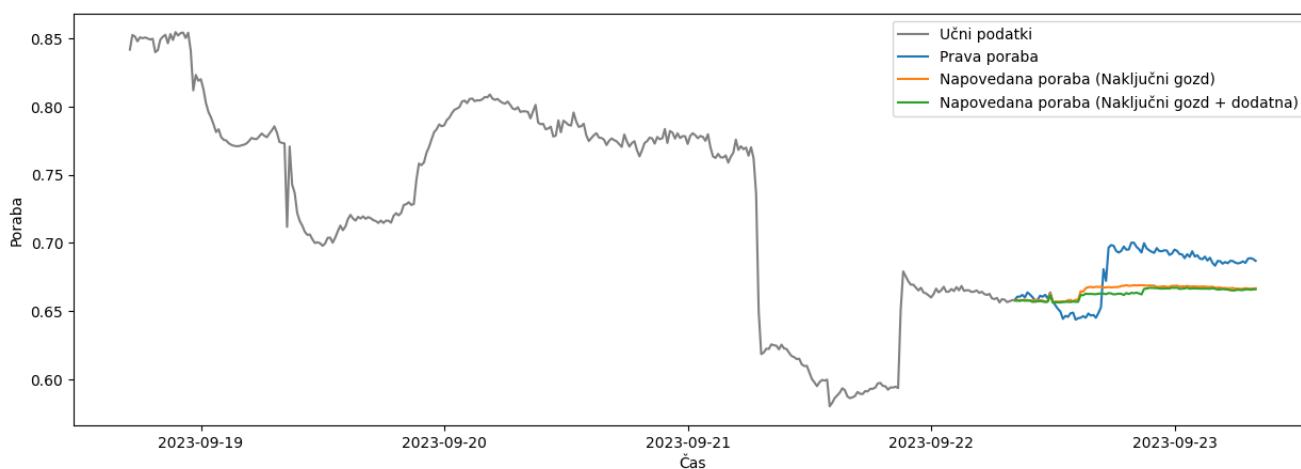
Rezultati napovednega modela so prikazani na sliki zgoraj. Vidimo, da RNN bolje ujame dinamiko porabe, kot RNN, ki uporablja še dodatno spremenljivko. RNN doseže napako RMSE enako 0.0153, RNN z dodatno spremenljivko pa 0.0275.

4.5 Priprava podatkov in učenje s klasičnimi pristopi strojnega učenja

Podatke podobno kot pri učenju RNN najprej razdelimo na 253 delnih časovnih vrst. Tokrat jih za učenje ne uporabimo neposredno, temveč iz njih izračunamo značilke. V ta namen uporabimo knjižnico **tsfresh** (Christ, et al., 2018) v programskem jeziku Python 3. Ta avtomatsko izračuna širok nabor značilk (1566 značilk) na podlagi delnih časovnih vrst. Prav tako nam na podlagi statističnih testov omogoči, da ta nabor skrčimo le na tiste, ki so relevantni za napovedovanje porabe. Takih značilk ostane 648.

Vsakemu 24 urnemu intervalu oz. delni časovni vrsti sedaj pripišemo vrednosti teh 648 značilk. V primeru, ko upoštevamo še dodatno spremenljivko, njene meritve dodamo k izbranim značilkam.

Ti podatki so tedaj primerni za uporabo klasičnih pristopov strojnega učenja. Za prikaz uporabe tokrat izberemo metodo naključnih gozdov, implementirano v knjižnici **scikit-learn** (Pedregosa, et al., 2011) v okolju Python 3. Za velikost modela izberemo 100 naključnih dreves.



Naključni gozd na testni množici doseže napako RMSE 0.0201, z upoštevanjem dodatne spremenljivke pa 0.0216. Kot tudi pri RNN, upoštevanje dodatne spremenljivke nekoliko zmanjša točnost napovednega modela.

5 Zaključki in priporočena metodologija

Napovedovanje porabe električne energije dela ali celotnega proizvodnega procesa je lahko uporabno orodje za načrtovanje novih proizvodnih procesov ali za načrtovanje sprememb v obstoječih proizvodnih procesih. V preteklosti je bila naloga napovedovanja porabe električne energije pogosto naslovljena z fizikalnimi modeli, v tem poročilu pa predstavimo nekaj izbranih metodologij umetne inteligence, tj., pristop z rekurentnimi nevronskimi mrežami ter pristop s klasičnimi metodami strojnega učenja. V tem kontekstu identificiramo še

metodo ARIMA in njene razširitve, ki se pogosto uporablja za napovedovanje podatkov, ki so v obliki časovnih vrst kot so meritve porabe električne energije, a zaradi zahtev metode ta ni primerna za uporabo v procesu načrtovanja.

Glavni pogoj za uporabo metod UI za napovedovanje električne energije je zajem zgodovinskih podatkov porabe električne energije. Modeliranje porabe z metodami UI bo bolj uspešno, če je zgodovinskih podatkov več in če bodo le ti visoke kvalitete, tj., brez manjkajočih vrednosti, s čim manjšim šumom, itd. Če želimo v modeliranje vključiti še dodatne spremenljivke, moramo zajeti zgodovinske meritve tudi teh količin.

Glede na zahteve procesa načrtovanja moramo izbrati tudi določene parametre modeliranja, predvsem širino vhodnega okna, tj., na podlagi kako velikega časovnega obdobja želimo napovedovati porabo električne energije, ter širino napovednega okna, tj., za koliko vnaprej želimo porabo napovedati. Zajete podatke nato ustrezno razdelimo glede na izbrani vrednosti teh parametrov.

Da ustrezno ocenimo uspešnost napovedovanja, je priporočljivo, da nek del podatkov zadržimo za vrednotenje. Klasični pristopi za vrednotenje metod UI, kot so prečno preverjanje, v tem kontekstu niso primerni, saj se pri napovedovanju porabe električne energije srečujemo s časovno urejenimi podatki. V poročilu prikažemo en možen primer vrednotenja, kjer del podatkov zadržimo in na njem testiramo naučene napovedne modele z izbrano mero napake. Čeprav ta primer postopka vrednotenja lahko uporabimo kot izhodišče za vrednotenje splošnega napovednega modela, je priporočljivo, da celoten postopek vrednotenja prilagodimo specifičnemu procesu, ki ga modeliramo, predvsem na podlagi ciljev procesa načrtovanja. Tako lahko z načrtovanjem optimiziramo ceno porabljene električne energije ali pa konice v moči električne energije, ki bo v prihodnje vedno bolj pomemben faktor pri obračunavanju električne energije.

Za bolj robustno napovedovanje porabe električne energije priporočamo uporabo večih metodologij umetne inteligence, predvsem z metodami opisanimi v tem poročilu. S primerjavo na zajetih podatkih se lahko nato odločimo za uporabo ene ali več metodologij v procesu načrtovanja.

Bibliografija

Box, G., & Jenkins, G. (1970). Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco.

Breiman, L. (2001). Random forests. Machine Learning, 45, 5-32. <https://doi.org/10.1023/A:1010933404324>

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). <https://doi.org/10.1145/2939672.2939785>

Chollet, F. (2015). Keras: The Python Deep Learning library. <https://keras.io>

Christ, M., Braun, N., Neuffer, J., & Kempa-Liehr, A. W. (2018). Time Series Feature Extraction on basis of Scalable Hypothesis tests (tsfresh—A Python package). Neurocomputing, 307, 72-77. <https://doi.org/10.1016/j.neucom.2018.03.067>

Dietmair, A., & Verl, A. (2009). A generic energy consumption model for decision making and energy efficiency optimisation in manufacturing. *International Journal of Sustainable Engineering*, 2(2), 123-133.

<https://doi.org/10.1080/19397030902953027>

Dietmair, A., Verl, A., & Eberspaecher, P. (2011). Model-based energy consumption optimisation in manufacturing system and machine control. *International Journal of Manufacturing Research*, 6(4), 380-401.

<https://doi.org/10.1504/IJMR.2011.041430>

Duflou, J. R., Kellens, K., Guo, Y., & Dewulf, W. (2012). Critical comparison of methods to determine the energy input for discrete manufacturing processes. *CIRP Annals*, 61(1), 63-66.

<https://doi.org/10.1016/j.cirp.2012.03.048>

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179-211.

https://doi.org/10.1207/s15516709cog1402_1

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.

<https://doi.org/10.1162/neco.1997.9.8.1735>

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Kara, S., & Li, W. (2011). Unit process energy consumption models for material removal processes. *CIRP Annals*, 60(1), 37-40. <https://doi.org/10.1016/j.cirp.2011.03.018>

Nelles, O. (2001). *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Springer, Berlin.

Ožbot, M., Lughofer, E., & Skrjanc, I. (2022). Evolving Neuro-Fuzzy Systems based Design of Experiments in Process Identification. *IEEE Transactions on Fuzzy Systems*, 1–11.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

<http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>

Qureshi, F., Li, W., Kara, S., & Herrmann, C. (2012). Unit process energy consumption models for material addition processes: a case of the injection molding process. In *Leveraging Technology for a Sustainable World: Proceedings of the 19th CIRP Conference on Life Cycle Engineering*, University of California at Berkeley, Berkeley, USA, May 23-25, 2012 (pp. 269-274). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-29069-5_47

Schmidt, C., Li, W., Thiede, S., Kara, S., & Herrmann, C. (2015). A methodology for customized prediction of energy consumption in manufacturing industries. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 2, 163-172. <https://doi.org/10.1007/s40684-015-0019-5>

Thiede, S., Bogdanski, G., & Herrmann, C. (2012). A systematic method for increasing the energy and resource efficiency in manufacturing companies. *Procedia CIRP*, 2, 28-33. <https://doi.org/10.1016/j.procir.2012.05.037>