Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Jan Kraljič

Napovedovanje podatkovnega toka porabe električne energije

Diplomsko delo
na univerzitetnem študiju

Mentor: prof. dr. Marko Robnik-Šikonja

Ljubljana, 2011

Original iz fakultete

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Jan Kraljič, z vpisno številko 63010073,

sem avtor diplomskega dela z naslovom:

Napovedovanje podatkovnega toka porabe električne energije

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Marka Robnik-Šikonje,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 01.06.2011 Podpis avtorja:

Zahvala

Za mentorstvo in strokovne nasvete pri izdelavi moje diplomske naloge se zahvaljujem prof. dr. Marku Robnik-Šikonji.

Kazalo

Povzetek

Napovedovanje podatkovnega toka porabe električne energije postaja vse pomembnejše pri obvladovanju poslovnih tveganj distributerjev in trgovcev z električno energijo, saj vpliva na uspešnost poslovanja. Napovedane količine uporabljajo tako za učinkovito trgovanje z električno energijo in distribucijo električne energije, kot tudi za planiranje oddane moči elektrarn ter upravljanje elektrarn v rezervi.

Število merilnih mest se v zadnjem času povečuje, prav tako pogostost pošiljanja podatkov o porabi električne energije. Podatki, ki prihajajo v enakomernih razmikih, predstavljajo podatkovni tok, ki se uporablja za izdelovanje kratkoročnih napovedi porabe. Za napovedovanje vrednosti v podatkovnem toku je potrebno preseči omejitve nekaterih algoritmov strojnega učenja.

Problem napovedovanja porabe električne energije smo razdelili na napovedovanje porabe električne energije za celoten naslednji dan in napovedovanje posameznih urnih vrednosti za naslednji dan. Teste podatkovnega toka smo izvedli na procentu porabe električne energije Ljubljane za obdobje v letih 2005-2008. Upoštevali smo tudi podatke o vremenu in koledarske posebnosti.

Na obeh podproblemih smo primerjali različne kombinacije algoritmov strojnega učenja, diskretizacij in podatkovnih oken. Klasične modele smo uporabili s pomočjo podatkovnega okna. Primerjali smo zlati standard, ARIMA, linearni model, naivni Bayesov klasifikator, k-nn, nevronske mreže in naključne gozdove glede na proporcionalno in ekvidistančno diskretizacijo ter z različno velikimi podatkovnimi okni. Večina uporabljenih algoritmov je bila že večkrat testirana na podobnih primerih, vendar ne skupaj z različnimi okni in z različnimi diskretizacijami napovedane spremenljivke.

Izbrane kombinacije dajo rezultate z relativno napako pod 5%, kar je rezultat, ki ga lahko uporabimo v praksi.

Ključne besede: strojno učenje, napovedovanje porabe električne energije, podatkovni tok, podatkovno okno

Abstract

Forecasting data streams of electricity consumption data is becoming more and more relevant for business risk management of electrical power distributors and traders. The forecasted values are used in electric market, load distribution, power plants load and power plants reserve management.

As the numbers of measurement points are increasing the electricity consumption data is measured in increasingly shorter intervals. The data, read at equal width intervals generates data stream which we use for short term consumption forecast. Data mining of data streams has to be treated specially by machine learning algorithms.

In this work forecasting problem has been split into two subproblems, one day ahead consumption forecast and hourly values for one day ahead. Data stream tests are performed on data for 1% of Ljubljana's electricity consumption between years 2005 and 2008. Additionally, weather data and calender have been taken into account.

Various combinations of data mining algorithms, discretizations and sliding windows are compared for both subproblems. Classical learning algorithms are used with sliding windows. Golden standard, ARIMA, linear model, naive Bayes classifier, k-nn, neural networks and random forest model are used in combination with equal frequency and equal width discretization and different sized sliding windows. Most of the mentioned models are commonly used on these type of data but not in combination with sliding windows and different discretizations.

The error rate of selected combinations is bellow 5%, which is already acceptable for practical use.

Keywords: machine learning, forecasting of electricity consumption, data stream, data window

1 Uvod

V zadnjih tridesetih letih sta se strojno učenje in podatkovno rudarjenje osredotočala na množice podatkov, kjer je celotna učna množica na voljo takoj na začetku učenja. Šele v zadnjem času se začenja raziskovanje podatkovnega rudarjenja tudi na podlagi podatkovnih tokov. Glavni motiv so čedalje večje količine zajetih merilnih podatkov, ki se zajemajo v bolj ali manj enakih časovnih intervalih. To so predvsem TCP/IP promet, strežniški podatki, zajemi podatkov iz senzorjev v industriji in v zadnjem času tudi meritve porabe električne energije. Sčasoma se je pokazalo, da podatkovno rudarjenje v toku potrebuje drugačen pristop.

Napovedovanje porabe električne energije postaja pomemben dejavnik obvladovanja poslovnega tveganja pri trgovanju z električno energijo. Kvaliteta napovedi močno vpliva na poslovno uspešnost trgovcev z električno energijo. Podatke pridobivajo iz velikega števila merilnih mest, ki v kratkih razmakih pošiljajo podatke o porabi električne energije. Iz tega podatkovnega toka se izdelujejo kratkoročne napovedi porabe električne energije.

Na področju napovedovanja porabe električne energije je bilo opravljenih že veliko raziskav, pretežno z uporabo nevronskih mrež, malo pa je takih, ki kot vir podatkov uporabljajo podatkovni tok. Pri proučevanju podatkovnega toka se večinoma uporabi podatkovno okno, ki omeji količino podatkov za algoritme strojnega učenja. Na osnovi podatkovnega okna je mogoča uporaba klasičnih algoritmov strojnega učenja, kot so k-najbližjih sosedov, naključni gozdovi in nevronske mreže, tudi na podatkovnem toku.

V tem delu za napovedovanje porabe uporabljamo preproste algoritme, kot so ARIMA, linearni model, naivni Bayesov klasifikator in k-najbližjih sosedov. Uporabljamo tudi kompleksnejše algoritme, kot so nevronske mreže in naključni gozdovi, vendar ti algoritmi, kljub svoji kompleksnosti ne dajejo dobrih rezultatov. Te učne algoritme smo uporabili za testiranje na podatkovnem toku.

Pri uporabi zveznih vrednosti smo preverili diskretizacijo in vpliv diskretizacije na kakovost napovedi. Uporabili smo proporcionalno in ekvidistančno metodo diskretizacije. Poleg izbire učnega algoritma in diskretizacije je zelo pomembna tudi analiza velikosti in oblikovanja podatkovnega okna.

Za napovedovanje električne energije smo zbrali podatke o porabi električne energije 1% mesta Ljubljane. Pri napovedovanju se je izkazalo, da imata pomembno vlogo vreme in koledar.

Za ocenjevanje različnih kombinacij algoritmov strojnega učenja, podatkovnih oken in diskretizacij smo izvedli različne teste. Testirali smo realne podatke dnevnih in urnih vrednosti za obdobje med 2005 in 2008.

V tem diplomskem delu smo primerjali različne kombinacije algoritmov strojnega učenja, diskretizacij in podatkovnih oken. Primerjavo smo izvedli na realnih podatkih porabe električne energije z uporabo programskega jezika R.

V drugem poglavju podajamo teoretične osnove uporabljenih postopkov strojnega učenja. Tretje poglavje predstavlja napovedovanje porabe električne energije, meritve in podatke o vremenu. V četrtem poglavju predstavljamo teste za dnevne in urne napovedi za naslednji dan. V rezultatih primerjamo različne testirane kombinacije. Zaključujemo s komentarjem rezultatov ter predstavitvijo ideje za nadaljnje delo.

2 Strojno učenje

V tem poglavju zgoščeno povzemamo delovanje uporabljenih algoritmov stojnega učenja, diskretizacije in podatkovnega okna. Na začetku opredelimo umestitev podatkovnega toka v strojno učenje.

Strojno učenje je del raziskav umetne inteligence. V zadnjem času je to področje zelo zanimivo za raziskovalce, kar kažejo številni članki objavljeni v strokovni literaturi. Strojno učenje se uporablja predvsem za analizo in odkrivanje znanja iz podatkov. Raziskav je mnogo, predvsem zaradi možnosti uporabe pridobljenega znanja v aplikacijah.

Strojno učenje opisuje podatke z modeli, ki so lahko pravila, funkcije, drevesa, verjetnosti ali enačbe. Naučeni modeli se uporabljajo za klasifikacijo ali regresijo novih primerov.

2.1 Podatkovni tok

Podatkovni tok opredelimo kot stohastični proces, ki se dogaja neprekinjeno in katerega primeri so medsebojno neodvisni [1]. Pri tem gre za časovno urejen niz podatkov s časovnimi žigi. Pri večini primerov so podatki enkrat brane vrednosti spremenljivk in dostop do njih ni ponovljiv. Primeri podatkovnih tokov so internetni promet, telefonski pogovori, vremenski podatki, poraba zemeljskega plina, poraba električne energije in zajemi podatkov iz senzorjev v realnem času.

Glavne lastnosti podatkovnega toka:

- podatki prihajajo neprekinjeno in v realnem času,
- na zaporedje pridobljenih podatkov ni mogoče vplivati,
- zaporedje podatkov je lahko (neskončno) dolgo.

Statični podatki	**Podatkovni tok**
enkratno pridobljeni podatki	podatki prihajajo neprestano
dostopamo lahko do kateregakoli podatka	dostopamo samo do zaporednega niza podatkov
načrtovano pridobivanje podatkov in vzorcev	podatkovni tok določa zaporedje pridobivanja podatkov in vzorcev

Tabela 2.1 Primerjava med statičnimi podatki in podatkovnim tokom.

V tabeli 2.1 so prikazane glavne razlike med statičnimi podatki in podatkovnim tokom.

Podatkovni tokovi s svojimi lastnostmi prinašajo nekaj specifičnih problemov in omejitev glede delovanja algoritmov. Glavni problem je količina podatkov podatkovnega toka in sposobnost izdelave čimbolj natančne napovedi v zelo kratkem času. Omejitve, ki jih prikazuje tabela 2.2, so predvsem v strojni opremi in časovni zahtevnosti posameznih algoritmov.

	Statično	**Podatkovni tok**
Število poskusov za posamezni primer	Poljubno	En
Čas na voljo za obdelavo podatkov	Poljubno	Omejeno
Poraba spomina	Neomejeno	Omejeno
Rezultati	Natančni	Približni

Tabela 2.2 Primerjava načina delovanja algoritmov strojnega učenja na statičnih podatkih in delovanjem na podatkovnem toku.

2.2 Rudarjenje v podatkovnem toku

Podatkovno rudarjenje v podatkovnem toku je proces pridobivanja znanja iz podatkovnega toka in velja za podpodročje strojnega učenja. Glavni problem je napovedovanje naslednjih (prihodnjih) vrednosti v podatkovnem toku na podlagi modelov naučenih iz predhodnih vrednosti v podatkovnem toku. Napovedujemo lahko zvezne vrednosti (regresija) ali diskretne vrednosti (klasifikacija).

Metode uporabljene nad podatkovnim tokom se razlikujejo od tistih, ki se uporabljajo na statičnih podatkih. Z nekaterimi prilagoditvami, kot je na primer podatkovno okno, je možna uporaba istih metod.

Vrednosti spremenljivk v podatkovnem toku se ves čas spreminjajo, zato se ves čas spreminjajo tudi naučeni modeli. Inkrementalno učenje je v podatkovnem toku neuporabno zaradi neučinkovitosti. V večini primerov znotraj podatkovnega okna modele izdelujemo vsakič znova. Zaradi tega izbiramo algoritme, ki imajo kratek čas učenja. V

primerih, kjer zaradi časovnih omejitev ni mogoče vedno znova graditi celotnega modela, se uporablja kombinacija več modelov. Osnovni model se gradi kolikor je mogoče pogosto (vsak dan, uro), kar je odvisno od količine podatkov in hitrosti uporabljenega algoritma strojnega učenja. Poleg osnovnega modela znanje kombiniramo z modelom, ki se na zelo ozkem podatkovnem oknu uči iz zadnjih podatkov in je dovolj hiter.

2.2.1 Naključne vrednosti podatkovnega toka

Pri podatkovnem toku sestavljenem iz naključnih vrednosti je vsak podatek popolnoma neodvisen od predhodnega. To pomeni, da lahko zavzema celoten spekter vrednosti, ne glede na to, kako kratek je razmik pri zajemanju vrednosti. Na primer, pri internetnem prometu se prenos velikih količin podatkov lahko hipoma začne ali zaključi.

2.2.2 Delno odvisne vrednosti podatkovnega toka

Včasih podatki niso popolnoma neodvisni od predhodnih vrednosti. Tu podatek ne more zavzeti celotnega spektra vrednosti. Do tega prihaja predvsem zaradi fizikalnih omejitev. Na primer, temperatura zraka se v izbranem časovnem intervalu lahko spremeni le za določeno vrednost. Podobno velja za obrate elektro motorja, kjer se tudi v primeru izklopa elektrike vrednost ne spremeni na 0, saj vrtenje omejuje vztrajnost kot fizikalna lastnost.

2.3 Algoritmi strojnega učenja

V tem podpoglavju opisujemo uporabljene algoritme strojnega učenja in primerjalno metodo "Zlati standard".

2.3.1 "Zlati standard"

Pravzaprav ne gre za algoritem, pač pa za metodo napovedovanja, pri kateri napovedujemo z zadnjo poznano vrednostjo v podatkovnem toku. Metoda je merilo uspešnosti za druge metode in algoritme.

2.3.2 ARIMA

ARIMA (Autoregressive integrated moving average) model je splošen algoritem za napovedovanje časovne vrste, ki deluje na avtoregresijskem drsečem povprečju.

ARIMA model je uravnotežena kombinacija naključnega sprehoda (random-walk) in naključnega trenda (random-trend) [2]. Uravnoteženost doseže z dodajanjem diferencialov serije in/ali zamikov od napak napovedi v napovedovalne enačbe pri odstranjevanju zadnje napake v seriji napovedi.

Posebne različice ARIMA modela so naključni sprehod (random-walk), naključni trend (random trend), avtoregresijski modeli in eksponentno glajenje (exponential weighted moving averages).

Nesezonski ARIMA model, enačba 2.1, je opredeljen kot »ARIMA(p,d,q)«:

$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right) (1 - L)^d X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t \tag{2.1}$$

kjer je t kazalec niza, X_t številska vrednost v nizu in L_i časovni zamik.

V modelu:

- prvi del predstavlja avtoregresijsko povprečje, p je število avtoregresijskih izrazov,
- drugi del predstavlja diferenciale in d je število nesezonske razlike,
- tretji del predstavlja zamike od napovedi napak in q je število zamikov napak napovedi v napovedovalni enačbi.

2.3.3 Linearni model

Linearni model (lm) je definiran z linearno enačbo oblike:

$$g(V) = w_1V_1 + w_2V_2 + w_3V_3 + ...w_nV_n \tag{2.2}$$

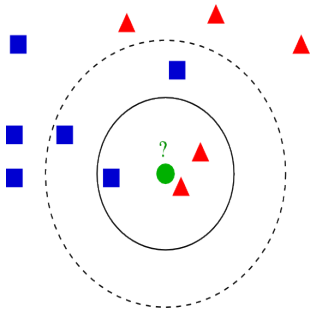
pri čemer je V_i spremenljivka atributov in w_i utež.

Učenje pri linearnem modelu [3] je določitev optimalnih uteži w_i na podlagi učne množice. Napovedovanje vrednosti po tej metodi je računanje linearne funkcije na podlagi vrednosti atributov in uteži.

2.3.4 k-najbližjih sosedov (k-nn)

Metoda k-najbližjih sosedov (v nadaljevanju k-nn) je osnovana na podobnosti testnega primera učnim primerom [3]. K-nn je tipični primer lenega učenja, za katerega velja, da se model ne pripravi vnaprej, ampak se kot znanje uporablja učna množica.

Pri testiranju primera iz učne množice poiščemo najbližji primer. Testni primer uvrstimo v razred, ki mu pripada največ bližnjih sosedov.



Slika 2.1 Slikovni primer delovanje k-nn algoritma.

Slika 2.1 prikazuje delovanje k-nn klasifikacije. Testni vzorec (zeleni krog) razvrščamo v razred modrih kvadratov ali razred rdečih trikotnikov. Če je k = 3, je testni primer uvrščen v razred trikotnikov, ker sta med sosedi 2 trikotnika in le 1 kvadrat. V kolikor je k = 5, je testni primer uvrščen med kvadrate (3 kvadrati proti 2 trikotnikoma znotraj zunanjega kroga)[4].

2.3.4.1 k-najbližjih sosedov z jedrno funkcijo podobnosti (k-nn kernel)

K-nn lahko izboljšamo z uteževanjem najbližjih sosedov z oddaljenostjo. Pri tem se razdalja upošteva z Gaussovim jedrom [5].

2.3.5 Naivni Bayesov klasifikator

Bayesov klasifikator deluje na osnovi pogojnih verjetnosti za vsak razred pri danih vrednostih atributov. Bayesov klasifikator, ki bi natančno izračunal pogojne verjetnosti razredov, bi bil optimalen, saj minimizira pričakovano napako. Takega klasifikatorja v večini primerov ne poznamo, zato je potrebno izračunati približke verjetnosti z vpeljavo predpostavk.

Naivni Bayesov klasifikator je enostaven verjetnostni klasifikator, ki predpostavi pogojno neodvisnost atributov pri danem razredu. S tem učna množica zadošča za zanesljivo oceno vseh potrebnih verjetnosti za izračun končne pogojne verjetnosti vsakega razreda [3].

Naivni Bayesov klasifikator je izpeljan iz Bayesovega pravila:

$$P(r_k|V) = P(r_k) \prod_{i=1}^{a} \frac{P(r_k|v_i)}{P(r_k)} \tag{2.3}$$

kjer je $P(r_k)$ apriorna verjetnost razreda r_k pri danih vrednosti v_i atributa A_i, V vektor vrednosti za vse atribute in a število atributov.

2.3.6 Naključni gozdovi

Naključni gozdovi so ena izmed najuspešnejših metod stojnega učenja z dobrimi rezultati na vseh področjih uporabe.

Algoritem naključnih gozdov (random forest) je predstavil Breiman leta 2001. Model je sestavljen iz množice odločitvenih dreves. Drevesa gradimo tako, da izbiramo najboljši atribut v vsakem vozlišče med naključno izbranimi kandidati. Učna množica primerov je izbrana s stremenskim vzorčenjem (bootstrap sampling).

Množica tako zgrajenih odločitvenih dreves pri klasifikaciji glasuje. Vsako drevo ima en glas, ki ga dodeli razredu v katerega je drevo klasificiralo primer. Število odločitvenih dreves v naključnih gozdovih je tipično večje od 100 [3].

Glavna prednost naključnih gozdov je robustnost, odpornost na šum (noise) in odpornost na prekomerno prilagoditev učni množici (over-fitting). Poleg tega nudi tudi vizualno predstavo modela.

2.3.7 Nevronske mreže

Umetne nevronske mreže (v nadaljevanju: nevronske mreže) so model, ki posnema funkcionalnosti biološkega nevrona s pomočjo matematične funkcije [3].

Model nevrona ima na vhodu x_i atribute z utežmi w_i, izhod pa se izračuna kot produkt vhodnih spremenljivk in uteži ter se transformira s pragovno funkcijo f.

$$x_{out} = f(\sum_i w_i x_i + w_{bias}) \tag{2.4}$$

kjer je pragovna funkcija f določena kot

$$f(X) = \begin{cases} 1 & \text{X} > 0 \\ -1 & \text{X} \leq 0 \end{cases} \tag{2.5}$$

ali kot sigmoidna zvezna in zvezno odvedljiva funkcija:

$$f(X) = \frac{1}{1 + e^{-X}} \tag{2.6}$$

Najpogosteje se uporabljajo usmerjene večnivojske nevronske mreže. Najpomembnejša nivoja sta vhodni in izhodni. Med njima pa je poljubno število skritih nivojev. Nevronske mreže so lahko tudi brez skritih nivojev.

2.4 Diskretizacija vrednosti

Za nekatere algoritme je potrebna diskretizacija podatkov. Diskretizacija je spreminjanje številskih vrednosti v diskretne vrednosti glede na vnaprej določene intervale. V tem poglavju obravnavamo diskretizacijo napovedane spremenljivke (razreda).

Določanje intervalov in števila diskretnih vrednosti razreda vpliva na rezultate algoritmov strojnega učenja, zaradi tega je izbira postopka diskretizacije pomembna. Pri velikem številu diskretnih vrednosti je izguba informacije manjša, vendar je število primerov s posamezno diskretno vrednostjo majhno zato pa tudi slabše delovanje algoritmov strojnega učenja. Pri majhnem številu diskretnih vrednosti pa se velik del informacije izgubi.

V našem primeru smo uporabili proporcionalno in ekvidistančno diskretizacijo [1].

2.4.1 Določanje števila vrednosti

Število diskretnih vrednosti je lahko vnaprej določeno ali pa število izračunamo iz učnih primerov. Na izbiro metode z vnaprej določenim številom vrednosti vpliva izbira algoritma strojnega učenja.

Pri majhni učni množici (do 20 primerov) je primerna razdelitev na dva intervala. Pri uporabi dovolj velike učne množice se v našem primeru izkaže, da je primerna razdelitev primerov na četrtine in na koren števila učnih primerov. Dobra lastnost izbire korena števila učnih primerov je enakomernost razporeditve učnih primerov v diskretne vrednosti, kar v splošnem daje dobre rezultate.

2.4.2 Proporcionalna diskretizacija

Pri proporcionalni razdelitvi primerov [1] v diskretne vrednosti se v vsako vrednost razvrsti enako število primerov.

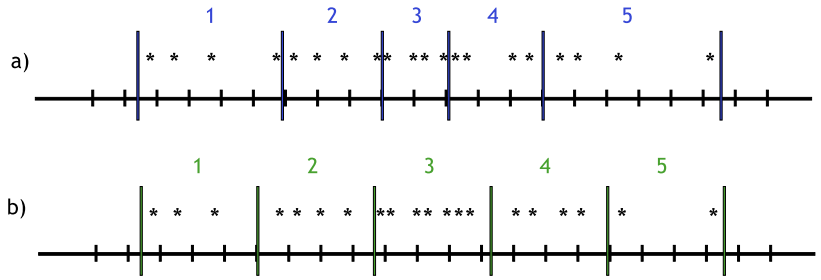
Na podlagi števila primerov in števila vrednosti izračunamo število primerov s posamezno vrednostjo. Razdelitev deluje tako, da številske vrednosti razvrstimo po velikosti, nato pa v vsako vrednost uvrščamo primere po vrsti do zahtevanega števila primerov. Tako ima vsaka vrednost enako število primerov, le zadnja jih ima lahko manj.

Proporcionalna diskretizacija je uspešnejša, kadar je več številskih vrednosti zelo neenakomerno razporejenih.

2.4.3 Ekvidistančna diskretizacija

Ekvidistnačna razdelitev [1] razdeli primere v vrednosti na podlagi enakih dolžin intervalov.

Dolžino intervala izračunamo tako, da razliko med največjo vrednostjo in najmanjšo vrednostjo delimo s številom diskretnih vrednosti. Na podlagi dolžine intervala določimo meje intervalov diskretnih vrednosti. Primerom določimo diskretne vrednosti glede na določene meje intervalov.



Slika 2.2 Primerjava disktretizacije; a) prikazuje proporcionalno diskretizacijo, b) prikazuje ekvidistančno diskretizacijo.

Na sliki 2.2 je prikazana diskretizacija številskih vrednosti s proporcionalno in ekvidistančno diskretizacijo. Na sliki je vidno, da se meje razlikujejo in da se številske vrednosti spreminjajo v različne diskretne vrednosti.

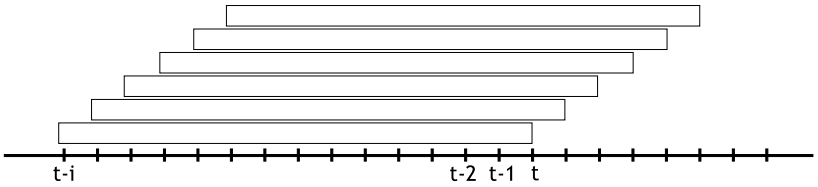
2.5 Podatkovno okno

V večini primerov celotna zgodovina podatkovnega toka ni zanimiva za uporabo v strojnem učenju. Novejši podatki so pomembnejši za napovedovanje kot starejši. Z uporabo podatkovnega okna se problem učenja v časovni sekvenci prevede na klasičen problem podatkovnega učenja. S tem je možna uporaba vseh algoritmov strojnega učenja tudi na toku podatkov.

Najenostavnejša je uporaba podatkovnega okna konstantne dolžine. Literatura [6] navaja dva glavna tipa drsečih podatkovnih oken:

- *Glede na časovno obdobje*, kjer je podatkovno okno definirano s časovnim intervalom. V podatkovno okno se uvrščajo vsi podatki, katerih časovni žig je znotraj časovnega intervala.

- *Glede na število podatkov*, kjer velikost in vsebovane podatke določa število primerov. V podatkovnem oknu je samo omejeno število podatkov iz zadnjega obdobja.



Slika 2.3 Prikaz delovanja drsečega podatkovnega okna.

Za vsako potrebno napoved se zgradi učni model na podlagi podatkovnega okna. Za napovedovanje vrednosti za čas t se za učenje modela uporabi podatke med t-i in t, kjer t ni vključen. Za napovedovanje vrednosti t+1 se celotno okno zamakne za +1. Pri tem je lahko i časovna enota (na primer ura, dan, mesec) ali število, ki opisuje velikost okna. Na sliki 2.3 je prikazano premikanje podatkovnega okna po podatkovnem toku.

Glavni prednosti uporabe podatkovnih oken sta zmanjšanje računske zahtevnosti in odstranitev vpliva starejših podatkov na algoritme.

2.6 Trend

Trend podatkovnega toka je dodatni atribut izračunan iz zgodovinskih vrednosti. Namen izračunavanja trenda je pomoč algoritmom pri upoštevanju lokalnih vrednosti.

Za postopek uporabimo formulo:

$$v'(i) = v(i-1) - \frac{1}{M}\sum_{i=j}^{M} v(j), \tag{2.7}$$

ki zadnji vrednosti iz podatkovnega toka odšteje povprečno vrednost okna. M je velikost podatkovnega okna.

3 Napovedovanje porabe električne energije

V tem poglavju predstavimo motiv za napovedovanje porabe električne energije, opišemo pridobljene podatke o porabi električne energije, pomembnost koledarskih atributov ter atributov o vremenu. Na koncu zgoščeno povzamemo dosedanje raziskovanje tega področja.

Napovedovanje porabe električne energije je ključno pri upravljanju z energetskim sistemom. Kratkoročne napovedi (t.i. short-term) so najpomembnejše za proizvodnjo električne energije in učinkovitost distribucije.

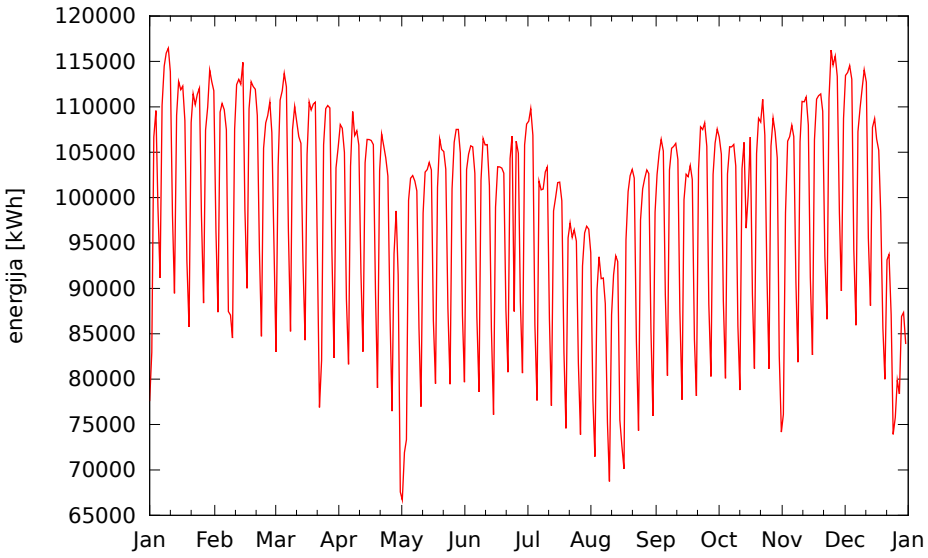
Pri proizvodnji električne energije se napovedovanje uporablja za načrtovanje količine oddane moči v elektrarnah ter za upravljanje rezervnih elektrarn ob napovedanih večjih spremembah porabe.

V distribuciji električne energije se napovedovanje porabe uporablja za zagotovitev kapacitet na omrežjih za prenos električne energije od proizvajalcev do porabnikov.

3.1 Podatki o porabi električne energije

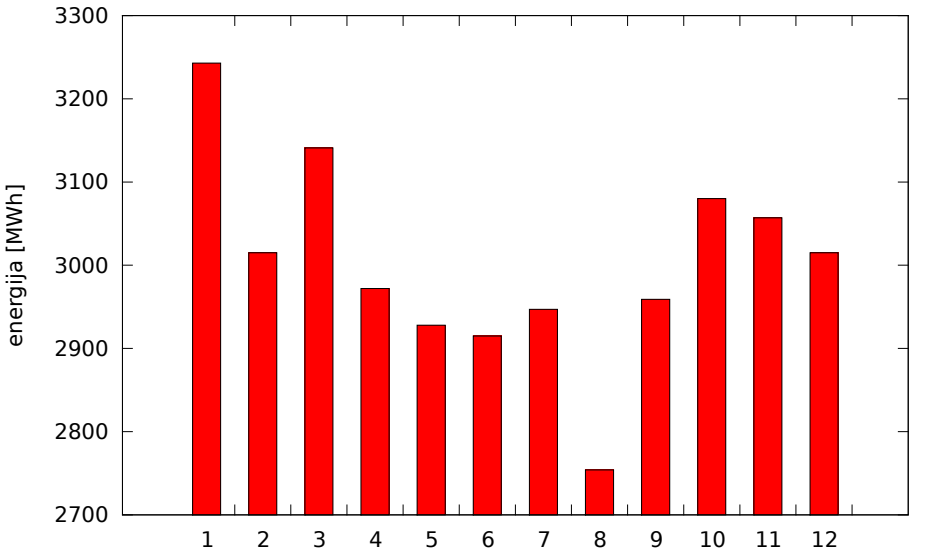
Za primerjave algoritmov strojnega učenja so bili uporabljeni pridobljeni merjeni urni podatki porabe električne energije za obdobje 2005-2008. Pridobljeni podatki predstavljajo 1% porabe električne energije v Ljubljani in so reprezentativni vzorec porabe. Podatki so pridobljeni iz skupka transformatorskih postaj. Reprezentativni vzorec vsebuje tako industrijske odjemalce, poslovne odjemalce kot tudi gospodinjstva.

Slika 3.1 prikazuje urne podatke za leto 2008, ki so bili uporabljeni za testno obdobje. Vidno je tedensko nihanje porabe ter spremembe v letu zaradi letnih časov in družbenih vplivov. Omenimo porabo 2. maja, kjer je najmanjša poraba v celem letu, kar je posledica vpliva praznikov (nedelja sočasno s praznikom in šolskimi počitnicami). Podobno velja tudi za ostale praznike (15.avgust, 1.januar), ki sovpadajo z neko drugo družbeno aktivnostjo.



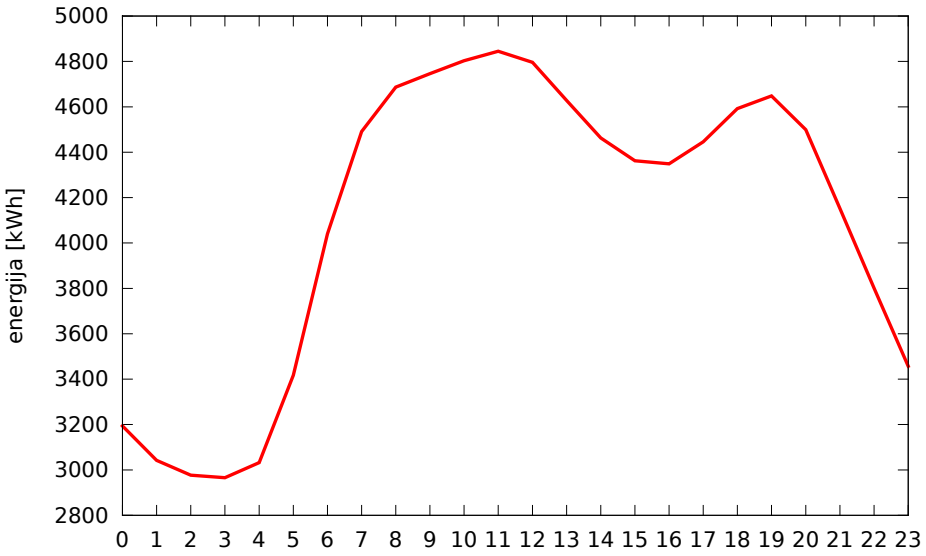
Slika 3.1 Letni urni graf podatkov porabe električne energije za leto 2008.

Slika 3.2 s podatki skupne mesečne porabe povzemajo dnevno porabo. S slike so razvidni letni časi. Poraba je v obratnem sorazmerju povprečne temperature. Opazimo tudi vplive družbe s prazniki, dopusti in šolskimi počitnicami.



Slika 3.2 Letni mesečni graf podatkov porabe električne energije za leto 2008.

Izrazit padec je viden v avgustu, ko je večina odjemalcev na letnem dopustu, kar posledično pomeni tudi nižjo porabo pri poslovnih odjemalcih. Januar ima zaradi nizkih temperatur višjo porabo, saj se del odjemalcev greje z elektriko.



Slika 3.3 Povprečni dnevni graf podatkov porabe električne energije.

Iz analize urnih podatkov na sliki 3.3 je razvidno, da je višja poraba električne energije med delovnim časom in v času okoli 20. ure, ko je večina ljudi doma. Ponoči je poraba bistveno nižja, saj je družbena aktivnost takrat manjša. Najnižja poraba je dosežena okoli tretje ure zjutraj, nato pa začne poraba strmo naraščati.

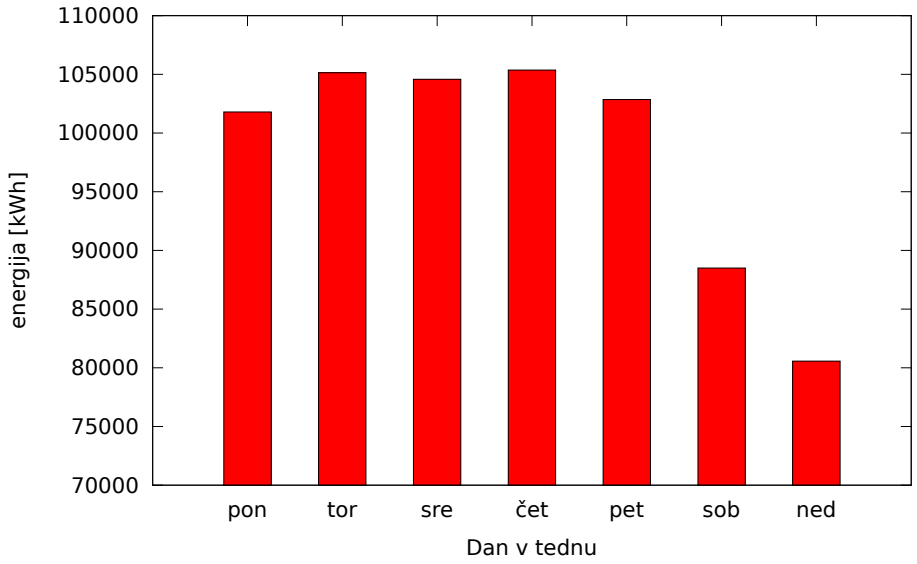
3.2 Koledarski dogodki

Na porabo električne energije močno vpliva koledar. Koledarski vpliv je tako velik, da je eden najpomembnejših atributov pri napovedovanju porabe električne energije. Do tega prihaja predvsem zaradi vpliva koledarja na delo ljudi, kar posledično vpliva na porabo električne energije.

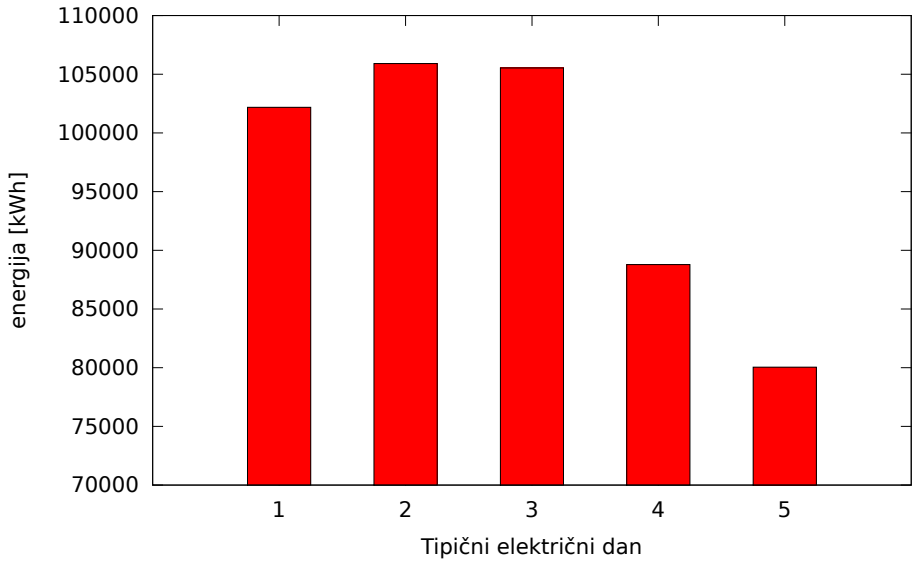
Osnovni koledarski atributi so leto, mesec, dan v mesecu, dan v tednu in ura v kolikor napovedujemo urno porabo.

3.2.1 Dodatni atribut: tipični dan

Dodatni atribut je tipični dan. Za napovedovanje električne energije se uporablja pet različnih tipičnih dni, le-ti pa so določeni glede na empirične ugotovitve strokovnjakov s področja.



Slika 3.4 Poraba električne energije glede na dan v tednu.



Slika 3.5 Poraba električne energije glede na tipične dni.

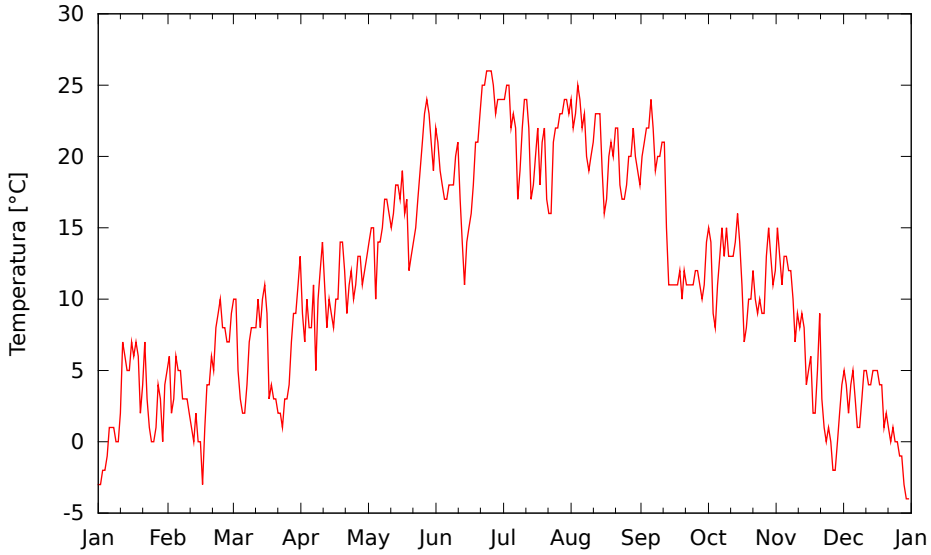
Slika 3.4 in slika 3.5 prikazujeta količino porabe električne energije po dnevih v tednu in tipičnih dneh. Iz slik je razvidna podobnost porabe energije v tednu glede na definicijo tipičnih dni porabe, ki jo podaja tabela 3.1. Te podobnosti so glavni razlog za dodaten atribut.

Tipični dan	**Vsebuje dni**
1	ponedeljek
2	torek, sreda, četrtek
3	petek
4	sobota
5	nedelja, praznik

Tabela 3.1 Definicija tipičnih električnih dni.

3.3 Vremenski podatki

Na porabo električne energije močno vplivajo klimatski dejavniki [7] in vremenski vplivi. Ti vplivajo predvsem na kratkoročne napovedi.



Slika 3.6 Povprečne dnevne temperature.

Uporabili smo urne meritve vremena ter urne napovedi iz sistema ALADIN Agencije Republike Slovenije za okolje (ARSO). ALADIN za napoved vremena uporablja numerični mezo-meteorološki model [8].

Uporabili smo:

- temperaturo,
- padavine,
- sončno obsevanost,
- minimalno in
- maksimalno temperaturo iz sistema ALADIN.

Za dnevne napovedi smo kot atribute uporabili minimalne in maksimalne vrednosti posameznih atributov.

3.4 Podproblemi

Napovedovanje porabe električne energije v tem delu sestavlja več podproblemov. Glavna podproblema sta napovedovanje porabe električne energije za celoten naslednji dan (day ahead forecast) in napovedovanje urnih vrednosti za celoten naslednji dan. Podproblema sta zanimiva zaradi različnih dejavnikov, ki vplivajo na porabo električne energije v celotnem dnevu in na razporeditev po urah znotraj posameznega dne.

Vsakega od obeh glavnih podproblemov sestavljajo trije deli: optimalna izbira diskretizacije, najboljši algoritem in optimalna velikost podatkovnega okna.

Slika 3.7 Slikovni prikaz treh delov posameznega podproblema.

Za vse tri izbire iščemo kombinacijo, ki daje najbolj točno napoved porabe. Pri tem imamo ogromno možnosti in kombinacij. Slika 3.7 prikazuje prostor kombinacij izbir. Potrebna je izbira smiselnih in za uporabo primernih kombinacij.

Pri izbiri podatkovnega okna je potrebno ugotoviti, kolikšna sta najmanjše in največje smiselno okno. Ob tem je potrebno upoštevati, koliko je vseh podatkov in kako pogosto prihajajo novi.

Problem izbire diskretizacije je pomemben. Ugotoviti je potrebno, kolikšna je še dopustna izguba informacije in kolikšno je minimalno še dopustno število primerov na diskretno vrednost.

Najenostavnejše je preverjanje različnih algoritmov stojnega učenja, kajti že pri privzetih nastavitvah algoritmov je mogoče ugotoviti, kako se giba klasifikacijska točnost posameznih algoritmov.

3.5 Obstoječi pristopi za napovedovanje porabe

V zadnjih tridesetih letih je bilo opravljenega veliko raziskovalnega dela na področju strojnega učenja. Na podatkovnih tokovih pa se raziskovanje povečuje v zadnjih desetih letih, kar kaže vse več objavljenih člankov in knjig v zadnjem času. Samo v zadnjih petih letih je bilo v sklopu IEEE objavljenih preko 300 člankov s področja podatkovnih tokov [9] in preko 150 člankov s področja napovedovanja porabe električne energije [9].

Pristopi pri napovedovanju porabe električne energije se nanašajo na uporabo že znanih algoritmov strojnega učenja, predvsem nevronskih mrež.

Leta 2002 je bil objavljen članek »Electric energy demand forecasting with neural networks« [10], kjer je prikazana uporaba nevronskih mrež za napovedovanje porabe električne energije. Rezultati nastavljene trinivojske nevronske mreže so v večini testov dosegali relativno napako do 5% .

Uporaba vremena kot atributa za napovedovanje porabe je predstavljena leta 2003 za primer napovedi porabe v Srbiji [11]. V članku avtor opisuje modela za dnevno in urno napoved na podlagi vremenskih podatkov, vendar ne uporablja podatkovnega toka.

ARIMA model se v povezavi z električno energijo omenja leta 2003 [12] in sicer za napovedovanje cen. Tu je ARIMA model dosegal dobre rezultate.

V zadnjem času so na področju raziskovanja podatkovnih tokov zelo dejavni raziskovalci Univerze v Portu na Portugalskem. Raziskujejo predvsem optimizacijo uporabe oken [13] in uporabe novih ali prilagojenih algoritmov za raziskovanje podatkovnega toka. V [1] so predstavljena zelo hitra odločitvena drevesa in združevanje podatkovnih tokov.

V literaturi ni zaslediti različnih pristopov in analiz s področja diskretizacije podatkov in vplivov le-teh na kakovost napovedi.

Obstoječi pristopi prikazujejo smeri raziskovanja uporabe podatkovnega toka, vendar ne najdemo primerjav med različnimi pristopi.

4 Testiranje različnih pristopov

V tem poglavju predstavljamo rezultate testiranj na podlagi realnih podatkov. Na začetku opisujemo scenarij, na podlagi katerega so bili izvedeni testi, nato pa proces testiranja in parametre testov.

Namen testiranja je primerjati delovanje različnih algoritmov stojnega učenja na podatkovnem toku pri različnih parametrih, kot so podatkovno okno in diskretizacija. Testiranja smo izvajali tako, da je mogoča uporaba obstoječih algoritmov strojnega učenja. Uporabljali smo privzete vrednosti parametrov.

4.1 Scenarij

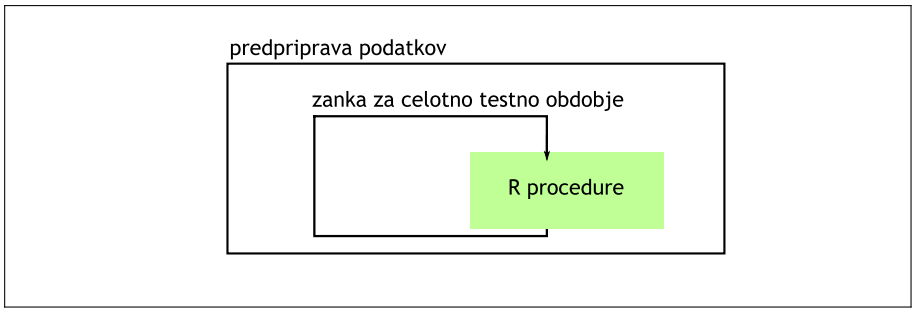
Testiranje je bilo opravljeno po scenariju kratkoročne napovedi porabe električne energije, ki napoveduje porabo električne energije za naslednji dan. Napovedovanje smo izvedli na dnevnem in urnem nivoju.

Scenarij je osnovan na realnem primeru trgovca z električno energijo, ki napoved porabe za svoje odjemalce predaja Borzenu (Borzen je organizator trga z električno energijo) en delovni dan pred odjemom.

4.2 Proces

Proces testiranja (slika 4.1) je sestavljen iz programa za obdelavo podatkov in procedur v programskem jeziku R. Proces testira delovanje na podatkovnem toku preko simulacije na že pridobljenih podatkih.

Simulacija podatkovnega toka je potekala tako, da je bilo za izbran test v času pripravljeno podatkovno okno s podatki. Nato se je okno premaknilo na naslednjo časovno točko. Premik pri dnevnem napovedovanju je za en dan, pri urni napovedi pa za 1 uro.



Slika 4.1 Shema testiranja.

4.2.1 Predobdelava podatkov

Program za predobdelavo podatkov je pripravil podatke za algoritme, s katerimi so bili opravljeni testi v programskem jeziku R. Glavne funkcionalnosti programa so bile izbiranje podatkov glede na podatkovno okno, diskretizacija podatkov, posredovanje podatkov algoritmom v R-ju in urejanje pridobljenih rezultatov.

Program je izvajal simulacijo podatkovnega toka s premikanjem po časovnem nizu podatkov in pripravo le-teh za R, tako da R ni imel direktnega dostopa do podatkov.

4.2.2 Algoritmi v R in parametri

Testiranje algoritmov za strojno učenje v sistemu R smo izvajali s privzetimi nastavitvami povsod, kjer je to bilo mogoče.

4.2.2.1 ARIMA

Pri ARIMA modelu smo uporabili kombinacijo avtoregresije in drsečega povprečja.

Klic funkcije:
›arima((ucnamnozica, order=c(1,1,1))‹

4.2.2.2 k-nn, k-nn z jedrom, naivni Bayes

Za k-nn, k-nn z jedrom in naivni Bayes smo uporabili R modul CORElearn [5] s privzetimi nastavitvami.

4.2.2.3 Algoritem naključnih gozdov

Za napovedovanje naključnih gozdov smo uporabili R modul CORElearn [5] za klasifikacijo in modul RandomForest [14] za regresijske teste.

Klic funkcije za klasifikacijo:
›CoreModel(disc ∼ ., elek, model="rf", selectionEstimator="MDL", minNodeWeightRF=2, rfNoTrees=100)‹

Klic funkcije za regresijo:
›randomForest(disc ∼ ., data=elek, na.action=na.roughfix , ntree=100, type=regression)‹

4.2.2.4 Nevronske mreže

Za napovedovanje nevronskih mrež smo uporabili R modul nnet [15].

Klic funkcije:
›nnet (disc ∼ ., data=elek, size=20, rang=0.1, decay=5e-4, maxit=500)‹

4.3 Parametri

Testiranje je bilo opravljeno v treh različnih dimenzijah in dveh podproblemih. Dimenzije v katerih so bili izvedeni testi so:

- različne velikosti podatkovnega okna in statično okno,
- uporaba različnih vrst diskretizacije in brez diskretizacije,
- uporaba različnih algoritmov strojnega učenja.

4.3.1 Statični test

Statični test smo opravili zaradi primerjave med statičnimi modeli in modeli s podatkovnim tokom. V konkretnem primeru smo izdelali model na podlagi učnih podatkov za leto 2005 in 2006 in testirali na letu 2008 za dnevne podatke in januar 2008 za urne podatke. Pri tem se naučeni model ni spreminjal skoz čas. V rezultatih je ta poskus viden kot "static" test.

4.3.2 Podatkovno okno

V testih smo uporabili podatkovna okna velikosti:

- 100 dni,
- 370 dni,
- 100 vrednosti na dnevnem nivoju, 50 vrednosti na urnem nivoju,
- vse vrednosti (poseben primer, ki vsebuje vse podatke).

4.3.3 Diskretizacija

Uporabili smo naslednje diskretizacije:

- ekvidistančna razdelitev, pri čemer je število vrednosti kvadratni koren primerov,
- ekvidistančna razdelitev, pri čemer je število vrednosti četrtina primerov,

- proporcionalno razdelitev, pri čemer se v število vrednosti spremeni kvadratni koren primerov,
- proporcionalno razdelitev, pri čemer se v število vrednosti spremeni četrtina primerov

in posebni primer brez diskretizacije, kjer so algoritmi to dopuščali.

4.3.4 Algoritmi

Na testiranjih smo uporabili algoritme:

- zlati standard,
- linearni model,
- ARIMA,
- k-najbližjih sosedov (tudi z Gaussovim jedrom),
- naivni Bayesov model,
- naključni gozdovi,
- nevronske mreže.

4.4 Kombinacije

Testirali smo 131 kombinacij za napovedovanje dnevnih vrednosti in prav toliko kombinacij za napovedovanje urnih vrednosti. Za algoritma zlati standard in ARIMA ni potrebno preizkušati različnih oken in diskretizacije, medtem ko so bili linearni modeli preizkušeni za različna okna, vendar brez diskretizacije.

4.5 Rezultati

V tem podpoglavju predstavljamo rezultate testiranj. Najprej predstavljamo rezultate dnevne napovedi za nevronske mreže in naključne gozdove, nato pa podatkovno okno in koledar. Med dnevnimi predstavljamo tudi rezultate z dodatnim lokalnim trend atributom. Za dnevnimi napovedmi predstavljamo še rezultate urnih napovedi glede na podatkovno okno in ure v dnevu.

Dobljene napovedi smo primerjali z dejanskimi izmerjenimi vrednostmi. Uporabili smo koren srednje kvadratne napake (RMSE (Root mean squered error)) in relativno napako.

4.5.1 Koren srednje kvadratne napake

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (v_i - t_i)^2}{n - 1}} \tag{4.1}$$

kjer je v_i dejanska vrednost in t_i napovedana vrednost in n število primerov.

S to metodo ocenimo kvaliteto napovedi na podlagi povprečnih kvadratov razlike med napovedjo in dejansko vrednostjo.

Metoda kaznuje večje napake pri napovedih.

4.5.2 Relativna napaka

Relativna napaka je razmerje med absolutno napako in izmerjeno vrednostjo. Absolutna napaka je definirana kot razlika med izmerjeno vrednostjo in napovedano vrednostjo algoritma.

4.5.3 Dnevna napoved

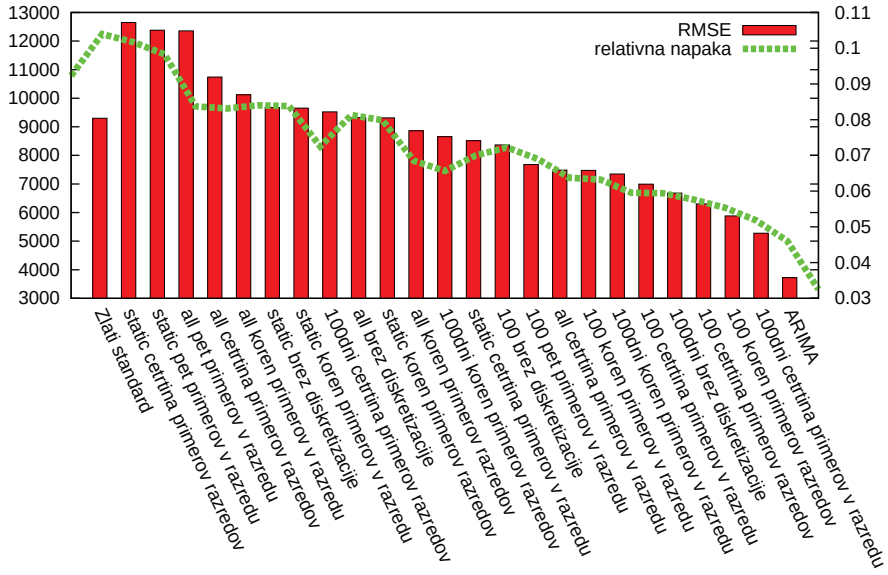
Rezultate za dnevno napoved smo pridobili na podlagi testiranj za leto 2008 in jih primerjali z dejanskimi vrednostmi. Pri primerjavah smo uporabili ARIMA algoritem in napoved z zlatim standardom.

ARIMA algoritem za dnevne napovedi je dajal najboljše rezultate s povprečno RMSE napako 3700kW, kar je 3% relativna napaka. ARIMA je za 0,5% prekosil najboljše rezultate kompleksnejših algoritmov.

4.5.3.1 Znani algoritmi

Nekatere algoritme za strojno učenje smo podrobneje testirali, ker se pogosto uporabljajo v praksi. Izbrana sta bila algoritma nevronske mreže in algoritem naključnih gozdov.

Relativna napaka za nevronske mreže pri testiranju je bila med 5% in 10%, kar je relativno slaba napoved glede na ARIMA s 3% relativno napako. Nekateri izmed testov so dosegli celo slabše rezultate od napovedi z zlatim standardom, katerega relativna napaka dosega 9%.

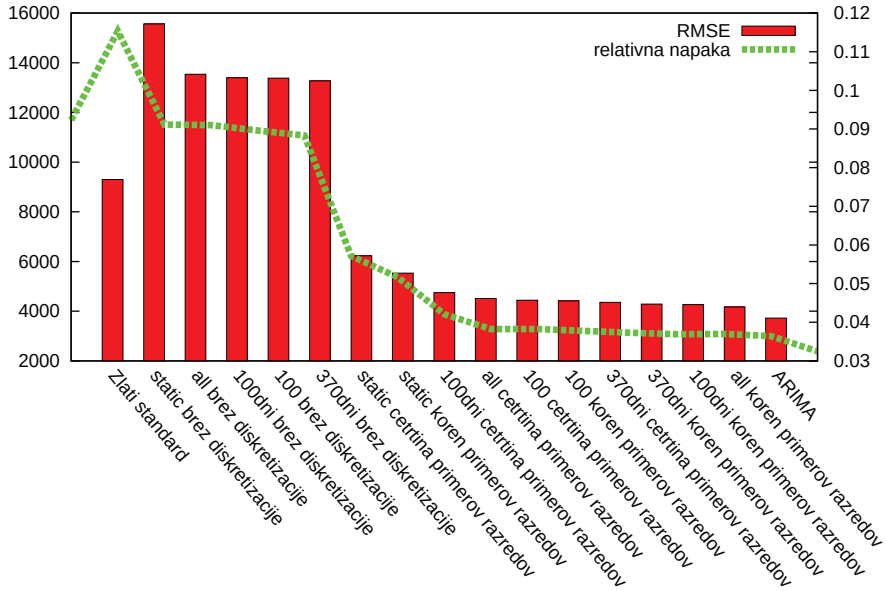


Slika 4.2 Primerjava testiranj nevronskih mrež.

Statični test (static), ki se v praksi največ uporablja, je dosegel slabe rezultate. Na sliki 4.2 so ti rezultati testov večinoma na levi, kjer so večje RMSE vrednosti. Izmed statičnih testov je najboljše napovedi podajala diskretizicija na štiri diskretne vrednosti. Napaka RMSE pri tem testu je bila 8500 kWh, kar je 7% relativna napaka.

Najkakovostnejše napovedi pri uporabi nevronskih mrež smo dobili z manjšimi podatkovnimi okni in diskretizacijo. Najboljši rezultat smo dobili s podatkovnim oknom velikosti 100 dni in diskretizacijo v štiri diskretne vrednosti. Ta test je dosegel 4,7% relativno napako in RMSE 530kWh. Pri diskretizaciji je potrebna previdnost pri zaključkih, kajti pri tem oknu različne diskretizacije dajejo zelo različne kakovosti napovedi.

Pri algoritmu naključnih gozdov lahko glede na rezultate vidimo oblikovane tri dele (slika 4.3). Slabe rezultate napovedi (slabše od napovedi z zlatim standardom) dobimo brez diskretizacije. Pri tem je potrebno poudariti, da algoritem naključnih gozdov ni bil optimiziran, tako da je možno odstopanje z boljšo izbiro parametrov algoritma. Pri testih je bila relativna napaka nad 8%.



Slika 4.3 Testiranje algoritma naključnih gozdov.

Statični test daje pričakovano slabe rezultate, kajti učna množica nima vrednosti, ki bi popravljale trend. Preostali testi dajejo dobre rezultate, ki so podobni ARIMA algoritmu.

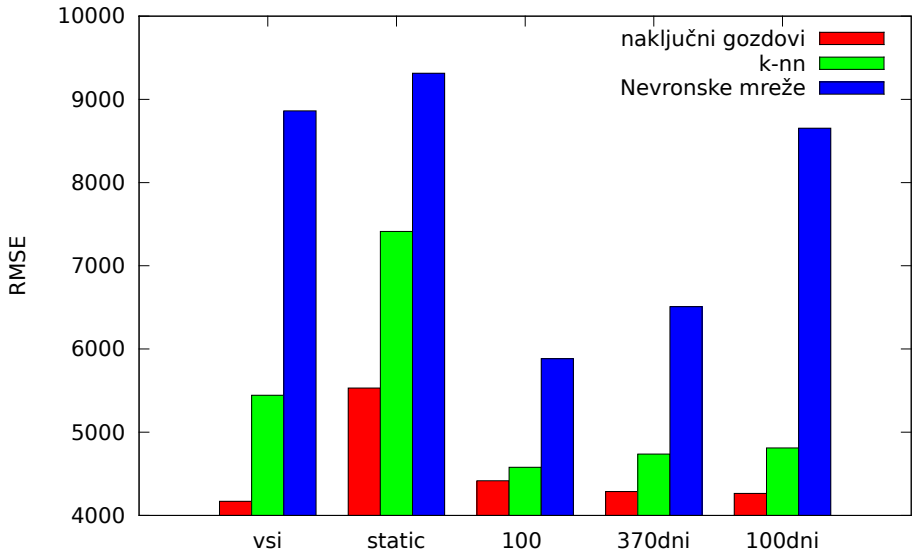
Rezultati uporabe nevronskih mrež in naključnih gozdov kažejo, da v splošnem boljše rezultate daje algoritem naključnih gozdov.

4.5.3.2 Podatkovno okno

Za testiranje velikosti podatkovnega okna smo izbrali metodo diskretizacije, pri kateri je bilo število diskretnih vrednosti določeno s kvadratnim korenom števila primerov in se je na testih pokazala za najboljšo pri več algoritmih.

Za primerjavo smo izbrali nevronske mreže, k-nn in algoritem naključnih gozdov. ARIMA algoritma in zlatega standarda tu nismo primerjali.

Rezultati z različnimi okni kažejo, kot je vidno na sliki 4.4, da izbira podatkovnega okna do določene mere vpliva na kakovost napovedi. Statični test je pričakovano najslabši.



Slika 4.4 Primerjava RMSE glede na podatkovno okno in število vrednosti določenih s kvadratnim korenom števila primerov.

Z različnimi podatkovnimi okni smo testirali vpliv zgodovine na delovanje algoritma. Tu se izkaže, da je vpliv podatkovnega okna na naključne gozdove manjši od drugih testiranih algoritmov. Na testih je bila razlika med najboljšim podatkovnim oknom in najslabšim 1,5MWh pri naključnih gozdovih, 4,5MWh pri nevronskih mrežah in 3MWh pri k-nn. Vrednosti RMSE za predstavitev na sliki 4.4 prikazuje tabela 4.1.

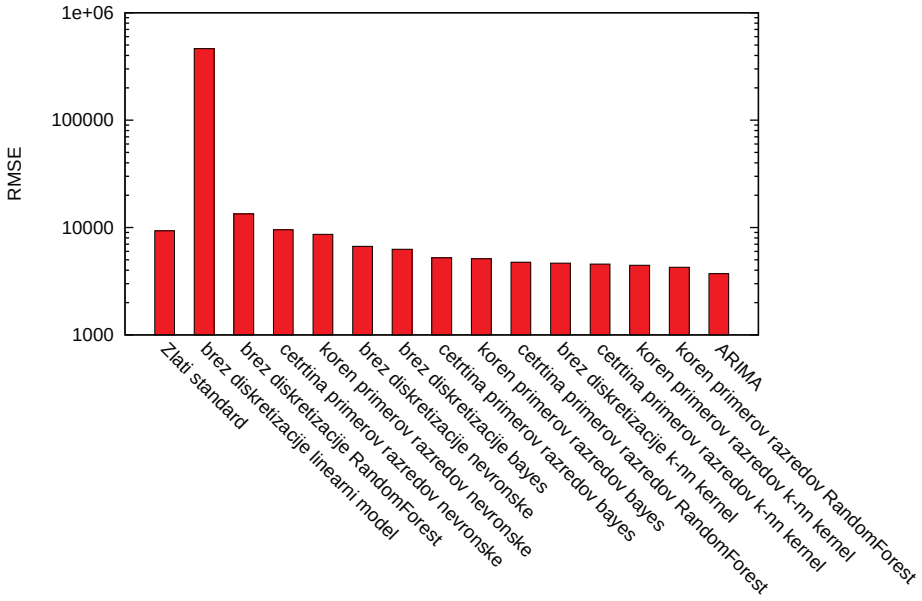
okno / algoritem	**naključni gozdovi**	**k-nn**	**Nevronske mreže**
vsi podatki	4169	5444	8862
static	5530	7412	9313
100	4417	4579	5883
100 dni	4264	4810	8654
370 dni	4287	4737	6508

Tabela 4.1 RMSE za različna podatkovna okna.

Z različnimi podatkovnimi okni (slika 4.5) smo ugotavljali, kje so napovedi najbolj natančne in kolikšna je najmanjša količina podatkov, ki je potrebna za kakovostno napovedovanje.

Rezultati kažejo, da je minimalna količina podatkov za kvalitetno napoved naslednjega dne zadnjih 100 dni zgodovinskih podatkov za algoritem naključnih gozdov, medtem ko potrebujejo nevronske mreže zadnjih 100 vrednosti. Nevronske mreže pri premajhni ali preveliki količini podatkov zaradi prevelike prilagoditve (over-fitting) delujejo slabše.

Primerjava rezultatov s 100 dnevi zgodovine nam pokaže, da poleg algoritma naključnih gozdov kvalitetne napovedi daje tudi k-nn jedrni algoritem.



Slika 4.5 Primerjava različnih algoritmov pri oknu 100 dni.

Na sliki 4.5 je prikazan tudi linearni model, ki vrača nekvalitetne napovedi, vendar je prikazan zaradi primerjave z urnim nivojem.

Slabše napovedi od zlatega standarda so dajali naslednji testi: linearni model, in naključni gozdovi brez diskretizacije, četrtina primerov vrednosti nevronske mreže.

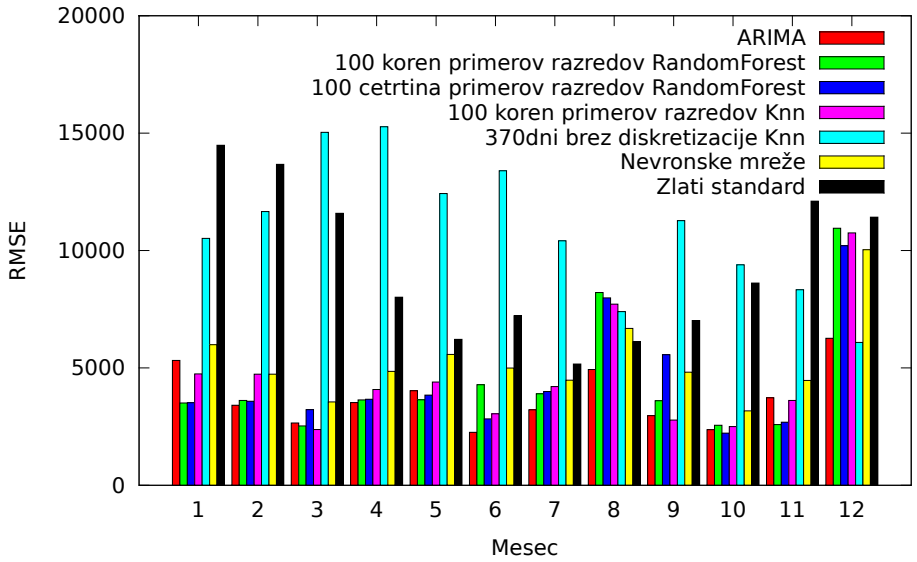
4.5.3.3 Koledarski učinki

Rezultati gledani skozi mesece v letu kažejo, da se kakovost napovedi skozi leto spreminja. Slika 4.6 prikazuje povprečno mesečno RMSE za leto 2008 izbranih algoritmov in diskretizacij.

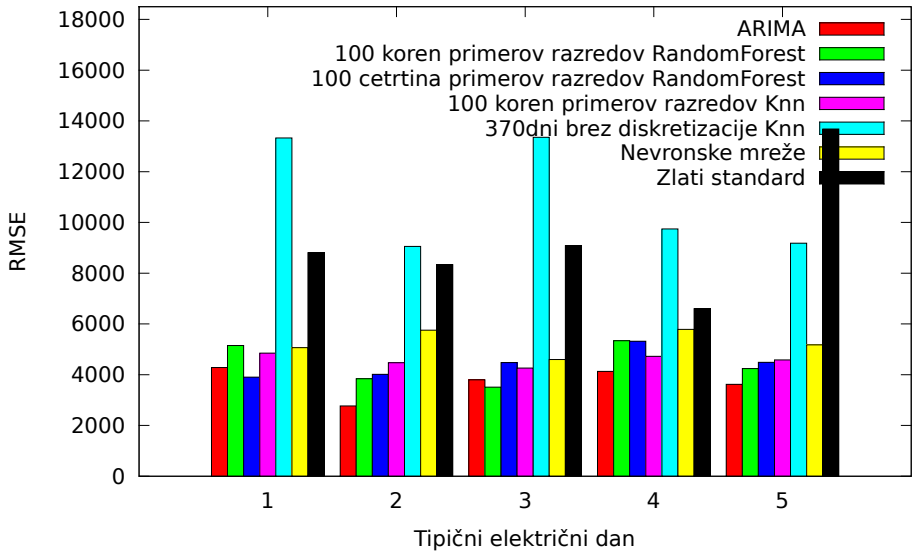
Slabo kakovost napovedovanja so vsi algoritmi dosegli v mesecu decembru in avgustu, kar je posledica družbenega vpliva. December je čas praznikov in s tem netipičnimi družbenimi vzorci porabe električne energije. Na porabo v avgustu vpliva odsotnost z dela in kolektivni dopusti podjetij.

V splošnem je ARIMA dosegala zelo dobre napovedi, vendar so v nekaterih mesecih drugi testi dosegali boljše. Testi z naključnimi gozdovi, v nekaterih primerih pa tudi k-nn, so dosegali dobre rezultate v mesecih, kjer ni večjega družbenega vpliva na porabo električne energije.

Slika 4.7 prikazuje primerjavo rezultatov napovedi izbranih algoritmov po posameznih tipičnih električnih dneh.



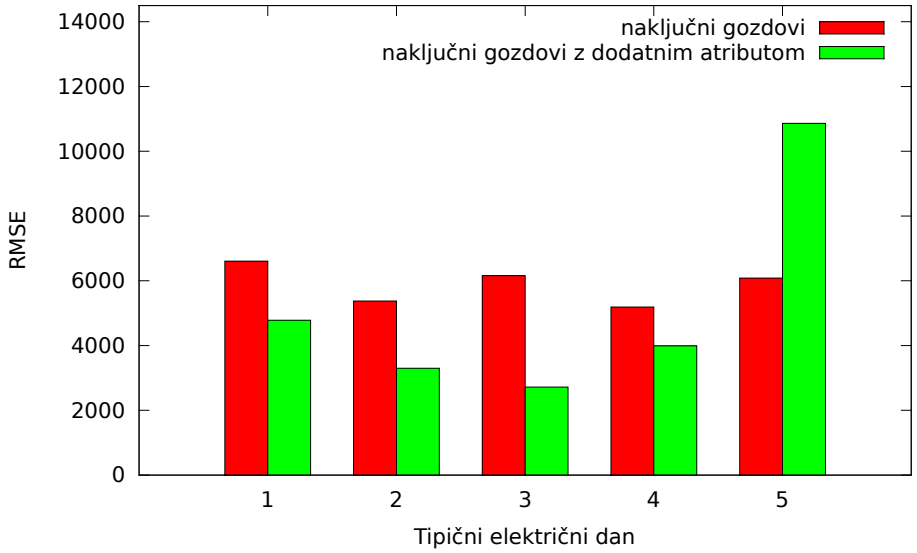
Slika 4.6 Primerjava RMSE izbranih testov po mesecih.



Slika 4.7 RMSE izbranih testov po tipičnih dnevih.

Napaka RMSE je enakomerno razporejena preko vseh tipičnih električnih dni, kar kaže na stabilnost napovedi v vseh dneh.

4.5.3.4 Z dodatnim atributom



Slika 4.8 RMSE naključnih gozdov s 100 koren primerov vrednosti.

Z vpeljavo dodatnega atributa lokalni trend so napovedi algoritmov naključnih gozdov boljše v večini tipičnih dni. Slabše so pri tipičnem dnevu 5, kamor sodijo nedelje in prazniki (slika 4.8). Ker je odstopanje v tipičnem dnevu 5 veliko, dva do trikratno povečanje RMSE, je posledično ocena RMSE višja, kar pomeni slabšo napoved. Pri tipičnem dnevu 5 je prišlo do velikih odstopanj pri rezultatih za praznike, kjer je bila relativna napaka višja od 50%.

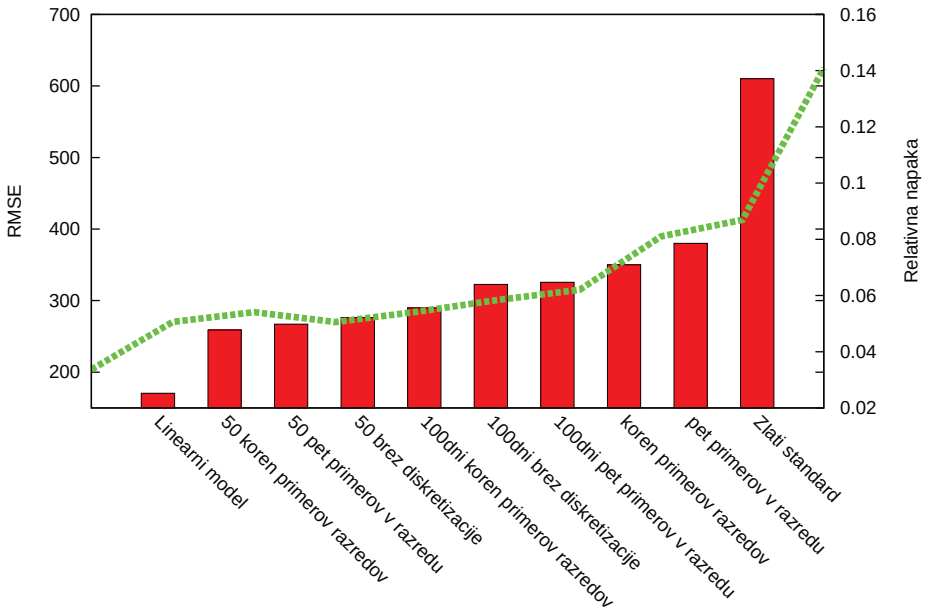
4.5.4 Urna napoved

Urne napoved smo pridobili na podlagi testiranj za mesec januar 2008 in jih nato primerjali z urnimi vrednostmi. Za primerjavo smo uporabili linearni model in napoved z zlatim standardom.

Z linearnim modelom smo primerjali zaradi zelo dobre napovedi na testiranju, kjer je dosegal relativno napako pod 3,5%.

4.5.4.1 Znani algoritmi

Nevronske mreže so na urnem nivoju dosegale relativno napako pod 9%, pri čemer je RMSE napaka 400kWh.

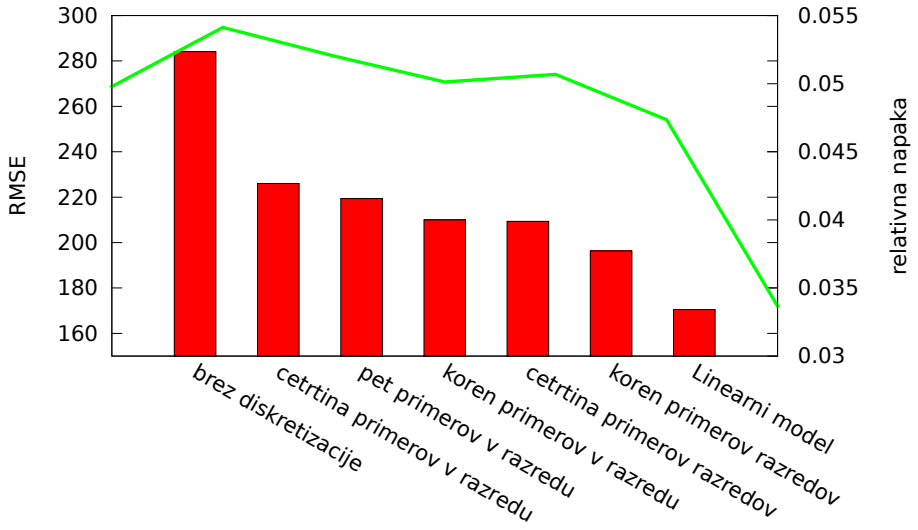


Slika 4.9 Primerjava 8 najkakovostnejših konfiguracij nevronskih mrež, linearnih modelov in zlatega standarda za urno napoved.

Na nevronske mreže vpliva količina podatkov v podatkovnem oknu. Na testih je najboljše rezultate pokazalo podatkovno okno z zadnjimi 50 vrednostmi. Pri večjih podatkovnih oknih kakovost napovedi pada. Podatkovno okno z zadnjimi 100 dnevi podatkov je doseglo boljše rezultate v primerjavi z zlatim standardom, tako kot kaže slika 4.9.

Različni načini diskretizacije pri nevronskih mrežah manj vplivajo na kakovost rezultatov v primerjavi z izbiro okna.

Naključni gozdovi na urnem nivoju pokažejo, da dobimo dobre rezultate ob podatkovnem oknu z vso zgodovino, kar nam pove, da ne pride do prekomernih prilagoditev učni množici.



Slika 4.10 Vpliv diskretizacije na naključne gozdove pri uporabi vseh podatkov.

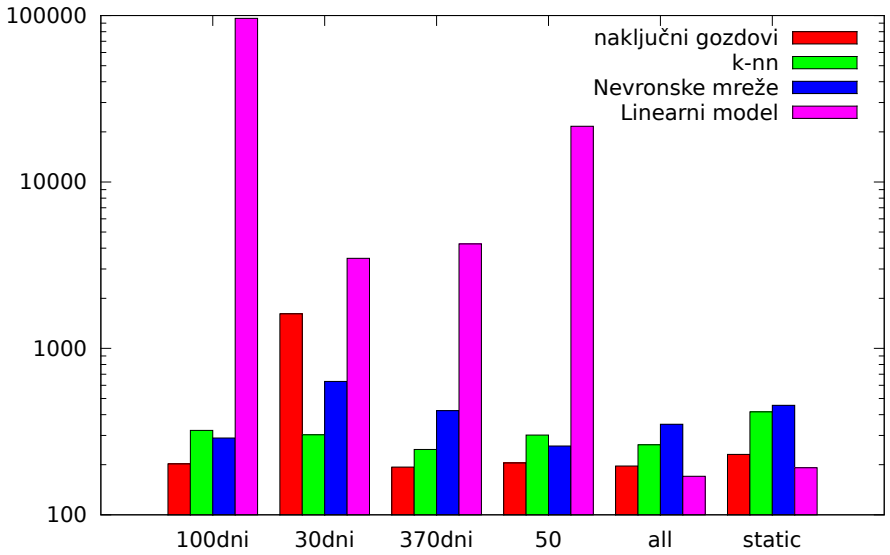
Pri primerjavi naključnih gozdov (slika 4.10) z linearnem modelom je razlika v relativni napaki manj kot 1%, torej so rezultati kakovostno primerljivi z najboljšim pristopom.

Pri diskretizacijah prihaja do majhnih razlik v kakovosti rezultatov. Veliko bolj je na kakovost rezultatov vplivala uporaba brez diskretizacije. Pri testu brez diskretizacije je bil uporabljen tudi drugi R modul, kar lahko vpliva na same rezultate.

4.5.4.2 Podatkovno okno

Pri različnih podatkovnih oknih prihaja do velikih nihanj v kakovosti rezultatov posameznih algoritmov.

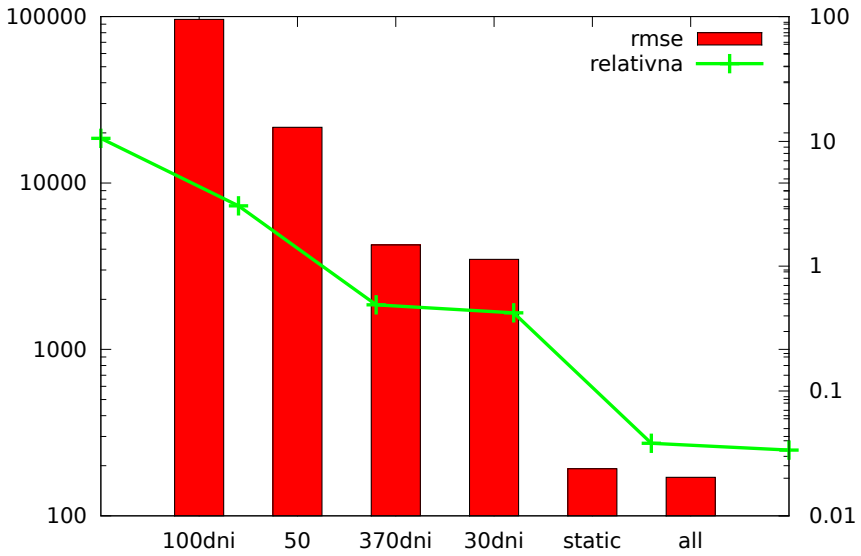
Najbolj stabilne rezultate je podajal algoritem naključnih gozdov, na katerega je podatkovno okno le malo vplivalo. Le pri zelo kratkem podatkovnem oknu se kakovost nekoliko zmanjša.



Slika 4.11 RMSE izbranih testov glede na podatkovno okno pri kvadratnem korenu primerov diskretnih vrednosti.

Algoritma k-nn in nevronske mreže stabilno delujeta s srednje velikimi okni, kar je razvidno iz RMSE (slika 4.11). Pri obeh algoritmih se pri statičnem testu in testu z vsemi podatki nekoliko zmanjša kakovost rezultatov, kar nakazuje na možnost prekomerne prilagoditve učne množice (over-fitting).

Izbira okna močno vpliva na linearni model (slika 4.12) v primerjavi z ostalimi izbranimi algoritmi (slika 4.11). Pri manjših oknih se kakovost rezultatov zelo zmanjša.

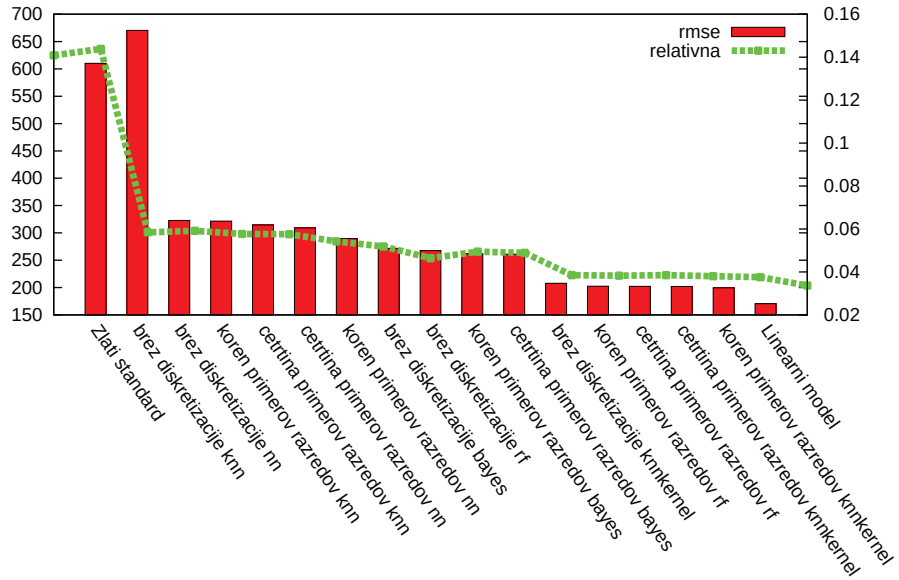


Slika 4.12 Linearni modeli glede na podatkovno okno.

Z izbiro različnih oken kakovost rezultatov linearnega modela niha med 0,5% in 10% relativne napake, pri tem RMSE niha med 150kWh in 10000kWh. Tako veliko nihanje

nas opozarja na nestabilnost algoritma ter zmanjša uporabnost linearnega modela in možnost izbire algoritma za praktično uporabo.

Rezultati testov pri konstantnem podatkovnem oknu 100 dni kažejo, da lahko rezultate testiranj razdelimo na tri dele. Najmanj kakovostni del je tam, kjer so rezultati podobni rezultatom zlatega standarda. To je na sliki 4.13 prikazano kot algoritem k-nn brez diskretizacije.

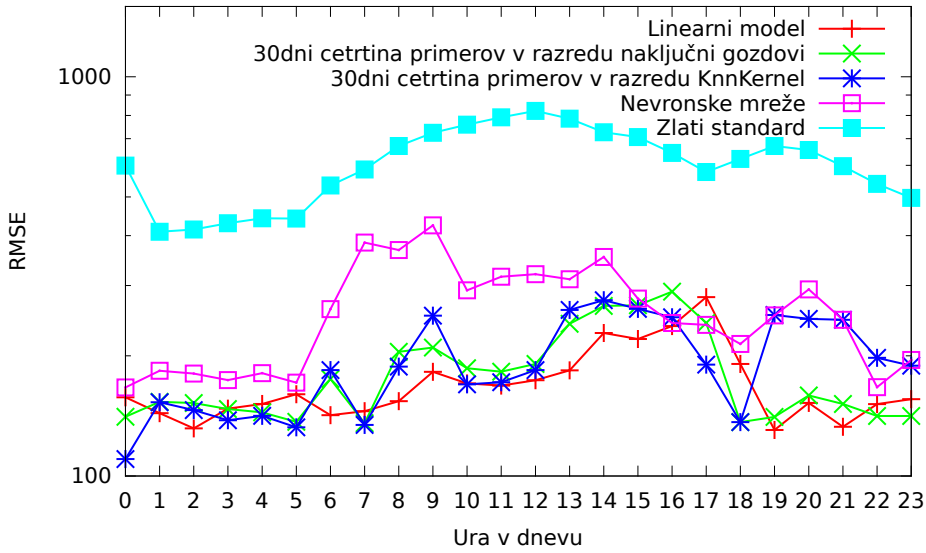


Slika 4.13 Primerjava različnih testov pri oknu 100 dni.

V osrednjem delu slike 4.13 so rezultati algoritmov s približno 5% relativno napako. Tretji del razdelitve pa so algoritmi z manj kot 4% relativno napako. Sem se uvršča algoritem k-nn z jedrom in naključni gozdovi.

4.5.4.3 Ure v dnevu

Rezultati urnih napovedi kažejo večjo kakovost napovedi v nočnem času za testirane algoritme. Naključni gozdovi, k-nn z jedrom in linearni model imajo podobne vrednosti RMSE.



Slika 4.14 RMSE po urah dneva.

Nevronske mreže imajo v celotnem dnevu slabšo kakovost napovedi. Po šesti uri zjutraj se RMSE več kot podvoji. Podrobne vrednosti RMSE so prikazane v tabeli A.1 v dodatku.

RMSE po urah kaže slabše napovedi v jutranjem času med šesto in deseto uro ter v popoldanskem času po štirinajsti uri (slika 4.14). Med dnevom je nižja RMSE v času med deseto in dvanajsto uro, kar kaže na stalnost v tem časovnem obdobju.

4.6 Povzetek in analiza rezultatov

Po pregledu rezultatov je analiza pokazala, katere izmed testiranih kombinacij so primerne za splošno uporabo in katere le za eksperimentalne primerjave. Za podobne primere je na podlagi analiz možno izbrati izhodiščne vrednosti za nadaljnje testiranje.

4.6.1 Algoritmi strojnega učenja

Analize rezultatov kažejo, da se v splošnem najbolje obnaša algoritem naključnih gozdov, ne glede na podatkovno okno in diskretizacijo.

Pri večjem številu podatkov pri nekaterih algoritmih prihaja do prekomernega prilaganja modela učni množici, po drugi strani pa imajo algoritmi težave pri majhnih podatkovnih oknih z malo podatki.

Linearni model se je izkazal pri urni napovedi, vendar je potrebno rezultate jemati z rezervo. Analiza pokaže, da je deloval dobro le pri veliki učni množici, medtem ko je pri manjši množici popolnoma odpovedal. Prav tako ni bil uspešen na dnevnem nivoju. To kaže, da je možnost uporabe tega algoritma omejena na veliko število podatkov, vendar pa vedno ni na voljo informacija o tem, koliko podatkov se bo uporabljalo.

Pri testiranju nevronskih mrež je prišlo do težav s preveliko časovno zahtevnostjo pri večjem številu podatkov, kar povzroča težave pri napovedovanju podatkovnega toka s kratkim razmikom med prihajajočimi podatki.

Na podlagi opravljenih testov in analiz je za podatkovne tokove najbolj priporočljiva uporaba algoritma naključnih gozdov.

4.6.2 Podatkovno okno

Velikost podatkovnega okna zelo vpliva na linearni model. Premajhno okno, v našem primeru 30 dni, vsem algoritmom močno zmanjša kakovost napovedi.

Analiza kaže, da uporaba statične podatkovne množice ni priporočljiva. Glede na teste je priporočljiva uporaba podatkovnega okna velikosti 100 dni ali 100 podatkovnih vrednosti.

4.6.3 Diskretizacija

Analiza je pokazala da je diskretizacija upravičena, kajti s tem pridobimo na kakovosti napovedi pri vseh testiranih algoritmih. Problem pri nediskretiziranih podatkih je velik razpon in razpršenost podatkov.

Primerjava obeh tipov testiranih diskretizacij pokaže, da je bolj kakovostne napovedi pri večini algoritmov dajala ekvidistančna diskretizacija. Pri tem sta, tako četrtina, kot kvadratni koren primerov različnih diskretnih vrednosti, dajala dobre rezultate.

Izjema so nevronske mreže, kjer sta dobre rezultate dajali tako proporcionalna diskretizacija kot tudi ekvidistančna. Pri nevronskih mrežah sta lahko uporabljeni obe diskretizaciji.

4.6.4 Atributi

Z dodajanjem atributa za določanje trenda je mogoče izboljšati napovedi v podatkovnem toku. Z naključnimi gozdovi (pri napovedovanju dnevih vrednosti) se napoved izboljša v vseh tipičnih dneh, razen pri tipičnem dnevu 5, kar je vodilo do skupno slabih rezultatov. Pri urni napovedi z dodanim atributom ni bilo izboljšanja.

5 Zaključek

V diplomi je bilo predstavljeno testiranje algoritmov strojnega učenja za kratkoročne napovedi porabe električne energije v podatkovnem toku. Večina uporabljenih algoritmov je bila uporabljena že večkrat, vendar ne skupaj z različnimi okni in z različnimi diskretizacijami napovedane spremenljivke. Uporabe algoritma naključnih gozdov, ki je na testih dajal odlične rezultate, še nismo zasledili, kajti večinoma se za ogromne količine podatkov uporablja algoritme hitrih odločitvenih dreves.

S testiranjem smo pokazali, da je za dobre napovedi potrebna skrbno izbrana kombinacija algoritma strojnega učenja, diskretizacije in podatkovnega okna. Uporaba podatkovnih oken je priporočena zaradi zmanjšanja časovne zahtevnosti gradnje modelov. Ob tem je pomembno, da se model strojnega učenja gradi čim pogosteje, saj je od tega odvisno upoštevanje zadnjih sprememb v podatkovnem toku. Pri nekaterih algoritmih za napovedovanje dnevne porabe so možne izboljšave rezultatov napovedi s pomočjo atributa za določanje trenda.

Na podlagi opravljenih testiranj algoritmov strojnega učenja, v kombinaciji s podatkovnim oknom in diskretizacijo, je možno kratkoročno napovedovanje porabe električne energije z natančnostjo pod 5% relativne napake. Takšna relativna napaka je sprejemljiva tudi v praksi, tako da je smiseln razvoj nekaterih boljših kombinacij za uporabo v realnih sistemih napovedovanja porabe električne energije. Priporočljiva je uporaba algoritma naključnih gozdov v kombinaciji z podatkovnim oknom 100 dni in diskretizacijo z kvadratnim korenom primerov različnih diskretnih vrednosti.

Nadaljnje delo je smiselno peljati v dve smeri: obdelavo podatkov in prilagajanje algoritmov. Pri podatkih so možne nove razlage podatkov in predobdelave. Za testiranje smo uporabljali privzete nastavitve algoritmov strojnega učenja, zato je tu še prostor za izboljšave.

Pri definiciji tipičnih električnih dni je prostor za izboljšavo pri dneh v bližini praznikov in šolskih počitnic. Predvsem v primerih, ko je dan pred praznikom tipični električni dan 2, je napaka vseh algoritmov napovedovanja velika, ker ne vključuje priprav industrije na začasno zaustavitev.

Pri izbiri velikosti podatkovnega okna in diskretizacije smo preverili le nekatere možnosti, ki niso nujno optimalne, ampak nakazujejo v kateri smeri bi bilo možno nadaljnje delo. Tako je možno bolj natančno določiti izbiro optimalne kombinacije velikosti podatkovnega okna in vrste diskretizacije v kombinaciji z algoritmi strojnega učenja.

6 Literatura

[1] J. Gama, Knowledge Discovery from Data Streams, New York: Taylor and Francis Group, LLC, 2010.

[2] Autoregressive integrated moving average, Dostopno na: http://en.wikipedia.org /wiki/Autoregressive_integrated_moving_average.

[3] I. Kononenko, Strojno učenje, Ljubljana: Založba FE in FRI, 2005.

[4] k-nearest neighbor algorithm, Dostopno na: http://en.wikipedia.org/wiki/K-nearest _neighbor_algorithm.

[5] M. Robnik-Šikonja, P. Savicky, COREIearn - classification, regression, feature evaluation and ordinal evaluation, Dostopno na: http://lkm.fri.uni-lj.si/rmarko/software /.

[6] M. Datar, A. Gionis, P.Indyk, R. Motwani, Maintainig stream statistics over sliding winsows , In proceeding of Annual ACM-SIAM Symposioum on Discrete Algoritms, str. 635-644, 2002.

[7] S. T. Chen, D. C. Yu, A. R. Moghaddamjo, Neural Network based Sohrt-Term Load Forecasting Using Weather Compensation , Power Systems, IEEE Transactions on, št.7, zv. 3, str. 1098 - 1105, 1992., Dostopno na: http://ieeexplore.ieee.org/xpl /freeabs_all.jsp?arnumber=207323.

[8] ALADIN Numerical Weather Prediction Project, Dostopno na: http://www.cnrm .meteo.fr/aladin/.

[9] IEEE Xplore, Dostopno na: http://ieeexplore.ieee.org/.

[10] D. Carmona, E. González, M. A. Jaramillo, J. A. Álvarez, Electric energy demand forecasting with neural networks , IEEE, št.3, str. 1860-1865, 2002.

[11] S. Ružić, A. Vučković, N. Nikolić, Weather sensitive method for short term load forecasting in Electric Power Utility of Serbia , Power Systems, IEEE Transactions on, št.18, zv. 4, str. 1581-1586, 2002.

[12] J. Contreras, R. Espínola, F. J. Nogales, A. J. Conejo, ARIMA models to predict next-day electricity prices , Power Systems, IEEE Transactions on, št.18, zv. 3, str. 1014-1020, 2003.

[13] P. P. Rodrigues, J. Gama, R. Sebastião, Memory fading windows in ubiquitous setting, 2009.

[14] A. Liaw, M. Wiener, Classification and regression by RandomForest , R News, št.2, zv. 3, str. 18-22, 2002., Dostopno na: http://CRAN.R-project.org/doc/Rnews/.

[15] W. N. Venables, B. D. Ripley, Modern Applied Statistics with S, New York: Springer, 2002.

Dodatek A RMSE

ura	linearni model	30 dni štiri vrednosti naključni gozd	30 dni štiri vrednosti k-nn jedro	nevronske mreže	zlati standard
1	157	141	110	167	599
2	144	153	153	184	409
3	132	152	146	181	415
4	147	147	138	174	430
5	151	144	142	181	442
6	160	137	132	171	441
7	142	175	184	262	534
8	145	136	134	384	586
9	154	205	188	368	671
10	182	210	252	424	724
11	171	186	170	292	759
12	168	183	172	316	791
13	174	191	184	320	821
14	184	241	261	311	786
15	228	267	276	354	726
16	220	268	262	278	706
17	237	290	250	242	644
18	281	242	190	239	578
19	191	136	136	214	623
20	130	141	253	253	671
21	152	159	247	293	655
22	133	152	246	246	597
23	151	141	198	167	539
24	156	141	189	195	497

Tabela A.1 RMSE ur.