Modeli za procjenu potrošnje prirodnog plina

Matić, Mislav

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mining, Geology and Petroleum Engineering / Sveučilište u Zagrebu, Rudarsko-geološko-naftni fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:169:070642

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-26**

Repository / Repozitorij:

Faculty of Mining, Geology and Petroleum Engineering Repository, University of Zagreb

SVEUČILIŠTE U ZAGREBU

RUDARSKO-GEOLOŠKO-NAFTNI FAKULTET

Diplomski studij naftnog rudarstva

MODELI ZA PROCJENU POTROŠNJE PRIRODNOG PLINA

Diplomski rad

Mislav Matić

N-245

Zagreb, 2018.

Sveučilište u Zagrebu
Rudarsko-geološko-naftni fakultet

Diplomski rad

MODELI ZA PROCJENU POTROŠNJE PRIRODNOG PLINA

MISLAV MATIĆ

Diplomski rad izrađen: Sveučilište u Zagrebu
Rudarsko-geološko-naftni fakultet
Zavod za naftno inženjerstvo
Pierottijeva 6, 10 000 Zagreb

Sažetak

Uredba Komisije (EU) br. 312/2014 o uspostavljanju mrežnih pravila o uravnoteženju plina transportnih sustava propisuje pravila uravnoteženja plinskog sustava u EU te ju je Hrvatska ulaskom u EU dužna implementirati. Što boljim i preciznijim modelima za procjenu potrošnje plina je prognoza točnija te je manja potreba za uravnoteženjem sustava i plaćaju će se manje naknade za odstupanje. U ovom radu prikupljeni su ulazni podaci za izradu modela te su konstruirani modeli pomoću platforme Spyder i programskog jezika Python. Potrošnje dobivene modelima uspoređivane su sa stvarnim potrošnjama te se na temelju toga odredila preciznost i točnost svakog modela.

Ključne riječi: prirodni plin, model, prognoza, temperatura, potrošnja plina, koeficijent determinacije

Diplomski rad sadrži: 53 stranice, 5 tablica, 16 slika, 8 priloga i 21 referencu.

Jezik izvornika: hrvatski

Diplomski rad pohranjen: Knjižnica Rudarsko-geološko-naftnog fakulteta
Pierottijeva 6, Zagreb

Voditelj: Dr. sc. Luka Perković, docent RGNF

Ocjenjivači: Dr. sc. Luka Perković, docent RGNF
Dr. sc. Domagoj Vulin, izvanredni profesor RGNF
Dr. sc. Tomislav Kurevija, izvanredni profesor RGNF

Datum obrane: 14.12.2018., Rudarsko-geološko-naftni fakultet, Sveučilište u Zagrebu

University of Zagreb
Faculty of Mining, Geology
and Petroleum Engineering

Master's Thesis

MODELS FOR NATURAL GAS FORECASTING

MISLAV MATIĆ

Thesis completed at: University of Zagreb
Faculty of Mining, Geology and Petroleum Engineering
Department of Petroleum Engineering
Pierottijeva 6, 10 000 Zagreb

Abstract

The Commission Regulation (EU) No.312/2014 regarding the establishing a Network Code on Gas Balancing of Transmission Networks has been presribed the EU and, as such, Croatia is obligated to implement it. Thanks to the increasingly improved accuracy of models for forecasting gas utilization, the need for the balancing of the system is reduced and penalties for deviation are becoming smaller This thesis gathers the input data for the creation of models, which are then created using the Spyder platform and Python programming language. Natural gas consumption calculated using the aforementioned models is then compared to historical gas consumptions and, based on the comparison results, the accuracy and the precision of each model is determined.

Keywords: natural gas, model, forecast, temperature, natural gas consumption, coefficient of determination

Thesis contains: 53 pages, 5 tables, 16 figures, 8 extensions and 21 references.

Original in: Croatian

Thesis deposited at: The Library of Faculty of Mining, Geology and Petroleum Engineering,
Pierottijeva 6, Zagreb

Supervisor: Assistant Professor Luka Perković, PhD

Reviewers: Assistant Professor Luka Perković, PhD
Associate Professor Domagoj Vulin, PhD
Associate Professor Tomislav Kurevija, PhD

Date of defense 14.12.2018: Faculty of Mining, Geology and Petroleum Engineering, University of Zagreb

SADRŽAJ

SADRŽAJ i

POPIS SLIKA iii

POPIS TABLICA v

POPIS OZNAKA vi

1 UVOD 1

2 ENERGETSKA BILANCA PLINA U RH 3

3 PREGLED MODELA ZA PREDVIĐANJE POTROŠNJE PRIRODNOG PLINA 5

3.1 Opći prikaz matematičkih modela 5

3.1.1 Usporedni modeli 5

3.1.2 Linearni modeli 5

3.1.3 Nelinearni modeli 7

3.2 Konstruirani modeli za procjenu potrošnje plina 9

3.2.1 Spyder platforma i programski jezik Python 9

3.2.2 Tipovi podataka koji se koriste pri konstruiranju modela za procjenu potrošnje plina 9

3.2.3 Grafički prikaz skupina podataka s odgovarajućim korelacijama 14

3.2.4 Linearni model s 2 temperaturne zone 15

3.2.5 Linearni model s 3 temperaturne zone 17

3.2.6 Model umjetne neuronske mreže s 1 temperaturnom zonom (ANN s 1 zonom) 20

3.2.7 Model umjetne neuronske mreže s 2 temperaturne zonom (ANN s 2 zone) 22

3.2.8 Model umjetne neuronske mreže s 3 temperaturne zonom (ANN s 3 zone) 24

3.2.9 Model temperaturne korelacije 26

4 ANALIZA REZULTATA I DISKUSIJA 28
5 ZAKLJUČAK 30
6 LITERATURA 31
PRILOZI 34

POPIS SLIKA

Slika 2-1 Energetska bilanca plina RH za 2016. godinu 4

Slika 3-1 Prikaz zemalja članica ENTSO-E grupacije 13

Slika 3-2 Međuodnos (korelacija) potrošnje plina i potrošnje električne energije od 1.4.2017. do 31.3.2018. godine te s pripadajućom satnom temperaturom zraka za svaku točku 14

Slika 3-3 Međuodnos (korelacija) potrošnje plina i temperatura zraka od 1.4.2017. do 31.3.2018. godine te s pripadajućom satnom potrošnjom električne energije za svaku točku 15

Slika 3-4 Ulazni set podataka za linearni 2Z model s podjelom na zonu niže i više temperature 16

Slika 3-5 Korelacija potrošnji dobivenih linearnim 2Z modelom i stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2 17

Slika 3-6 Ulazni set podataka za linearni 3Z model s podjelom na zonu niske,srednje i visoke temperature 18

Slika 3-7 Korelacija potrošnji dobivenih linearnim 3Z modelom i stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2 19

Slika 3-8 Shema izgleda jednog seta podataka sa slojevima i pripadajućim formulama.. 20

Slika 3-9 Korelacija normaliziranih potrošnji dobivenih ANN modelom s 1 zonom i pravih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2 21

Slika 3-10 Korelacija potrošnji dobivenih s ANN modelom s 1 zonom i stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2 22

Slika 3-11 Korelacija normaliziranih potrošnji dobivenih ANN modelom s 2 zone i pravih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijenti determinacije R^2 za svaku zonu 23

Slika 3-12 Korelacija potrošnji dobivenih ANN modelom s 2 zone i stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2 24

Slika 3-13 Korelacija normaliziranih potrošnji dobivenih ANN modelom s 3 zone i normaliziranih stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijenti determinacije R^2 za svaku zonu 25

Slika 3-14 Korelacija potrošnji dobivenih ANN modelom s 3 zone i stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2 26

Slika 3-15 Korelacija potrošnji dobivenih modelom temperaturne korelacije i stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2 27

POPIS TABLICA

Tablica 3-1 Prikaz dijela tablice *combine_ALL_cijela_godina.xlsx* datoteke s potrošnjama plina konstruirane u MS Excelu 10

Tablica 3-2 Prikaz dijela tablice *combine_ALL_cijela_godina.xlsx* datoteke sa satnim temperaturama konstruirane u MS Excelu 11

Tablica 3-3 Prikaz dijela tablice *combine_ALL_cijela_godina.xlsx* datoteke s potrošnjama električne energije konstruirane u MS Excelu 11

Tablica 4-1 Tablični prikaz reprezentativnosti konstruiranih modela 28

Tablica 4-2 Chadockova ljestvica 28

POPIS OZNAKA

ARX - Autoregresivni model s egzogenim ulaznim podacima (engl. *Auto-regressive model with exogenous inputs*

RARX – Promjenjivi (rekurzivni) autoregresivni model s egzogenim ulaznim podacima (engl. *Adaptive auto-regressive model with exogenous inputs*)

NN - Neuronska mreža (engl. *Neuron network*)

RNN - Promjenjiva (rekurzivna) neuronska mreža (engl. *adaptive (recursive) neuron network*)

SVR - Regresija potpornog vektora (engl. *support vector regression*)

RH - Republika Hrvatska

ENTSO-E - Europska mreža operatera prijenosnih sustava električne energije (engl. *the European Network of Transmission System Operators for Electricity*)

HOPS - Hrvatski operator prijenosnog sustava

BiH – Bosna i Hercegovina

HEP ODS – Hrvatska elektroprivreda-Operator distribucijskog sustava d.o.o.

OTS - Operator transportnog sutava

VBS - Voditelji bilanačnih skupina

HROTE – Hrvatski operator tržišta energije

1. UVOD

Planiranje i procjenjivanje potrošnje prirodnog plina ima ključnu ulogu u postizanju stabilnosti distribucijskih sustava. Distribucijski sustav je sustav plinovoda i pripadajućih objekata koji su u vlasništvu i/ili kojima upravlja operator distribucijskog sustava, a koristi se za distribuciju plina, upravljanje, nadzor, mjerenje i prijenos podataka (Zakon o tržištu plina NN 18/18.). U Republici Hrvatskoj operator distribucijskog sustava je tvrka HEP-Operator distribucijskog sustava d.o.o. (HEP ODS). To znači da je HEP ODS odgovoran za rad, razvoj i održavanje distribucijskog sustava. Procjena potrošnje razmatra se na nekoliko različitih područja. To su procjene na svjetskoj razini, državnoj razini, regionalnoj, odnosno u okviru nekog distribucijskog sustava, odvojenih djelova distribucijskog sustava, razini određene industrijske grane te razini individualnih potrošača. Procjena potrošnje plina važna je i zbog ekonomskog utjecaja na potrošače, operatora plinskog sustava i za samu zemlju. Velika pogreška u procjeni potrošnje na pojedini hladni dan imala bi nemjerljive ekonomske posljedice za sve sudionike na plinskom tržištu. Najvažniji čimbenici u širem kontekstu na koje utječu procjene potrošnje plina su:

- Sigurnost opskrbe plinom : procjenjena potrošnja plina je ključan čimbenik za analizu opskrbe plinom jer ukazuje na to hoće li trenutna opskrba plinom moći udovoljiti potrebama za plinom u bližoj ili daljoj budućnosti.
- Skladišne kapacitete : procjene su korisne pri osiguravanju zaliha plina, odnosno smještanjem viška plina iz plinske mreže u skladišta plina. To omogućava neometan i siguran rad plinskog sustava te energetsku sigurnost zemlje.
- Planiranje izgradnje cjevovoda – procjene budućih potrošnji plina su indikator ima li postojeći plinovod kapacitete za transport budućih potrebnih količina plina temeljenih na predviđanjima.
- Izračun tarifnih stavki – procjene potrošnje plina imaju direktan i indirektan utjecaj na metodologiju utvrđivanja tarifnih stavki.

Uredba Komisije (EU) br. 312/2014 o uspostavljanju mrežnih pravila o uravnoteženju plina transportnih mreža donesena 3. ožujka 2014. godine propisuje pravila uravnoteženja plinskog sustava u Europskoj uniji (EU) te ju je RH ulaskom u EU dužna implementirati. U RH za uravnoteženje transportnog sustava se brine operator transportnog sustava (OTS),

tvrtka Plinacro d.o.o. Zona uravnoteženja su granice transportnog sustava. Voditelji bilanačnih skupina (VBS) predaju nominacije (količine plina koje korisnik transportnog sustava "rezervira" izražene u kWh/h). Bilančne skupine su udruženja na tržištu plina organizirane u svrhu optimiranja troškova uravnoteženja sustava za koje je odgovoran voditelj bilančne skupine te je svaki sudionik na tržištu plina (krajnji kupci, trgovci, distributeri, opskrbljivači, operateri) dužan biti član bilančne skupine osim operatora tržišta plina. Jedna od uloga VBS-a je uravnoteženje bilančne skupine koju organizira i vodi, odnosno usklađivanje dnevnih količina plina koje se predaju u transportni sustav i preuzimaju iz transportnog sustava kako bi operator transportnog sustava trebao poduzimati što manje radnji uravnoteženja. Također je odgovoran je za sve troškove vezane za uravnoteženje i odstupanja bilančne skupine koju organizira i vodi. Dnevno odstupanje bilančne skupine utvrđuje operator transportnog sustava i dostavlja to u konačnom izvještaju operatoru tržišta plina (u RH to je HROTE d.o.o.) (Pravila o organizaciji tržišta plina, 2018). Upravo zato da bi se izbjeglo plaćanje raznih naknada i da bi se što više smanjili troškovi uravnoteženja javlja se potreba za što točnijim i preciznijim procjenama potrošnje plina. Općenito na potrošnju plina utječe veliki broj faktora: vanjska temperatura, brzina i smjer vjetra, relativna vlažnost zraka, oborine, naoblaka, godišnje doba, mjesec u godini, vrsta dana (vikend, praznik), potrošnja električne energije te podaci o povijesnoj potrošnji. Posljedica toga su kompliciran i složeni modeli za procjenu potrošnje plina. Cilj ovog rada je izraditi vlastite modele za procjenu potrošnje plina na temelju već postojećih, ali njihovu pojednostavljenu verziju (uzimaju se 3 faktora koja utječu na potrošnju plina) te ih međusobno usporediti i odrediti koji je model najprecizniji.

2. ENERGETSKA BILANCA PLINA U RH

U ovom poglavlju se nalazi sažet opis energetske bilance plina u Republici Hrvatskoj za 2016. godinu. Prirodni plin u RH se proizvodi iz 17 eksploatacijskih polja Panona i tri eksploatacijska područja na Sjevernom Jadranu što podmiruje 63,1 posto domaćih potreba. Najveći dio plina koji se proizvodi na kopnu dolazi iz ležišta Molve, Kalinovac i Gola. Proizvodnja je iznosila 1647,2 milijuna m^3. Uvoz je iznosio 1264,7 milijuna m^3. Ukupno, proizvodnja primarne energije plina je bila 2911,9 milijuna m^3. Izvoz je bio 389,4 milijuna m^3 Tijekom 2016. godine u skladište je utisnuto ukupno 256 milijuna m^3 plina, a tijekom ciklusa povlačenja povučeno je 345 milijuna m^3 plina, odnosno saldo skladišta je 89 milijuna m^3 (Energija u Hrvatskoj, 2016.). To ostavlja ukupnu potrošnju na 2611.4 milijuna m^3.

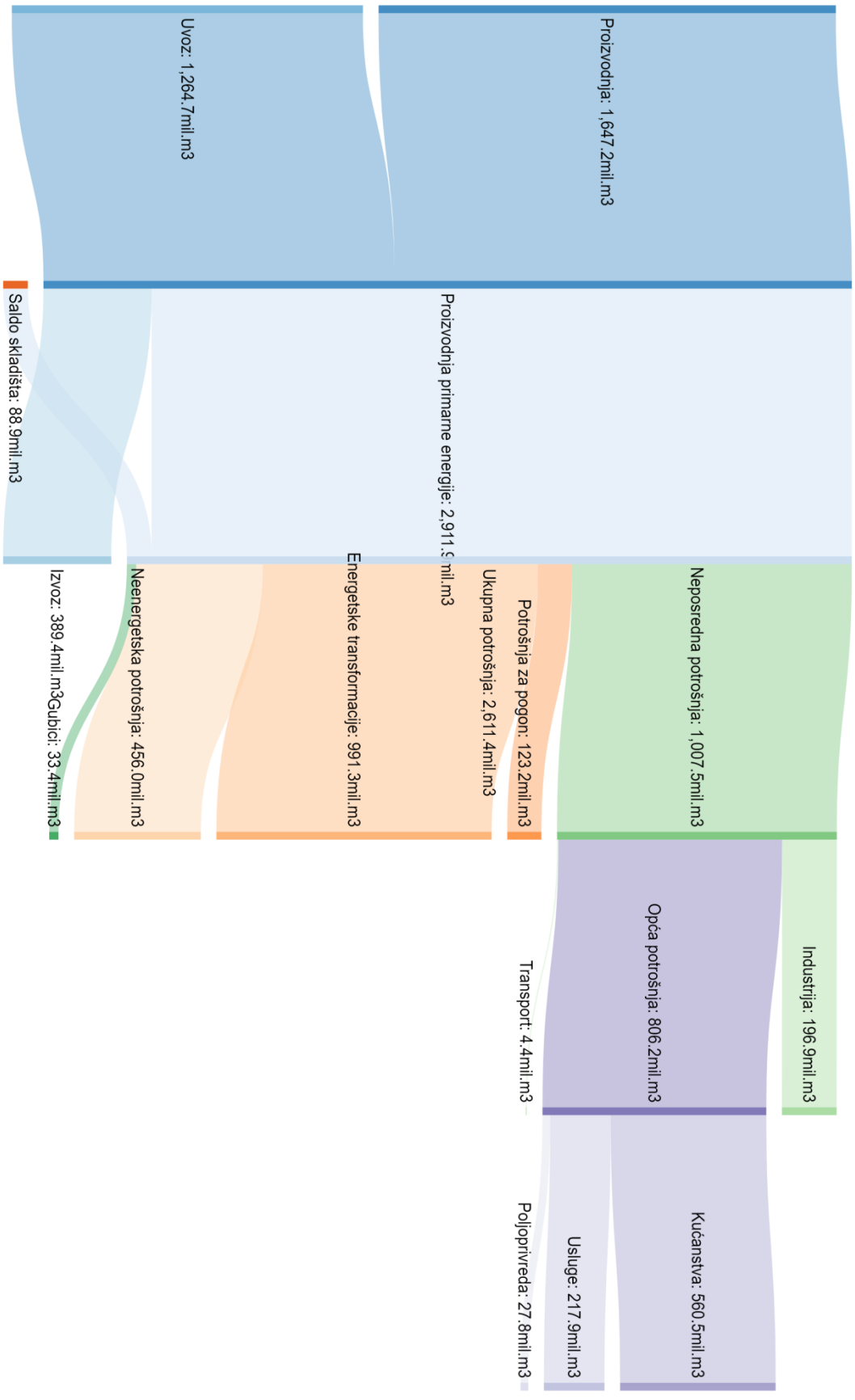
Potrošnja za pogon, odnosno za proizvodnju nafte i plina, rafinerije i degazolinažu iznosila je 123.2 milijuna m^3.

U domenu energetskih transformacija spadaju: termoelektrane, javne toplane, javne kotlovnice, industrijske toplane, industrijske kotlovnice, rafinerije, gradske plinare te degazolinaža i potrošnja je iznosila 991.3 milijuna m^3.

Neenergetska potrošnja koja se odnosi na potrošnju tvrtke Petrokemija Kutina d.o.o. koja se bavi proizvodnjom gnojiva, je iznosila 456 milijuna m^3, a gubici su iznosili 33.4 milijuna $m^{3.}$.

Neposredna potrošnja je iznosila 1007.5 milijuna m^3 od kojih je industrija (kogeneracijska postrojenja – proizvodnja električne i toplinske energije) trošila 196.6 milijuna m^3. Na transport (pretežito autobusni) se potrošilo 4.4 milijuna m^3.

Opća potrošnja je iznosila 806.2 milijuna m^3, od toga je na kućanstva (grijanje i topla voda) otpadalo 560.5 milijuna m^3, usluge (ponajviše hoteli) 217.9 milijuna m^3 te na poljoprivredu 27.6 milijuna m^3. Slika 2-1 prikazuje energetsku bilancu RH koja odgovara opisu u ovom odlomku.



Slika 2-1 Energetska bilanca plina RH za 2016. godinu

3. PREGLED MODELA ZA PREDVIĐANJE POTROŠNJE PRIRODNOG PLINA

U ovom poglavlju prikazani su matematički modeli za predviđanje potrošnje koji se općenito koriste u praksi s pripadajućim matematičkim formulacijama te konstruirani modeli s pripadajućim ulaznim podacima i rezultatima.

3.1. Opći prikaz matematičkih modela

U ovom potpoglavlju se prikazuju matematički modeli koji se općenito koriste za procjenu potrošnje plina koje navode autori (Soldo et al.) znanstvenog članka *Comparison of static and adaptive models for short-term residential natural gas forecasting in Croatia*. Prema autorima postoje 3 vrste matematičkih modela: usporedni modeli (engl. *Benchmark models*), linearni modeli (engl. *Linear models*) i nelinearni modeli (engl. *Nonlinear models*).

3.1.1. Usporedni modeli

a) Model temperaturne korelacije (engl. *Temperature correlation model*)

U modelu temperaturne korelacije u vezi su potrošnja prirodnog plina *y(t+1)* i prosječna dnevna temperatura *T(t+1)*. Matematička formulacija je:

$$y(t+1) = b_0 + b_1 T(t+1) \quad (1)$$

Obrazloženje ovog modela je snažna "negativna" međusobna povezanost potrošnje plina i vanjske temperature.

3.1.2. Linearni modeli

a) Model postupne regresije (engl. *Stepwise regression*)

Model postupne regresije je linearni model koji nastaje iterativnim dodavanjem i uklanjanjem uvjeta iz višelinearnog modela, na temelju njihovog statističkog značaja u regresiji. Metoda započinje s početnim modelom koji se zatim uspoređuje s dodavajućim, većim i manjim modelima. U svakom koraku, izračunava se *p* veličina od *F* statističkog podatka redoslijedom da se ispitaju oba modela, sa ili bez potencijalnog ulaznog podatka. Ispitani ulazni podaci se uzastopno dodavaju ili oduzimaju od modela dok se postupak ne približi najpovoljnijem modelu prognoziranja sa statističkim značajnim ulaznim podacima. Metoda također rješava problem kolinearnosti reducirajući dostupne skupine ulaznih podataka na samo one bitne.

b) Autoregresivni model s egzogenim ulaznim podacima (engl. *Auto-regressive model with exogenous inputs*

Matematička formulacija atoregresivnog model s egzogenim ulaznim podacima (ARX) je:

$$A(q)y(t) = B(q)x(t) + e(t) \quad (2)$$

A(q) i *B(q)* označavaju polinome koji su u odnosu s operatorom promjene vremena *q: A(q) = 1 +* $a_1q^{-1}+...+a_Mq^{-M}$, i *B(q) =* $b_1+b_2q^{-1}+...+b_Mq^{-M+1}$. Za ovaj model ispitivani su razni M (redovi polinoma) da se ostvari nabolji odnos između točnosti i složenosti ovog modela.

c) Promjenjivi autoregresivni model s egzogenim ulaznim podacima (engl. *Adaptive auto-regressive model with exogenous inputs*)

RARX model je promjenjiva (rekurzivna) verzija autoregresivnog modela s egzogenim ulaznim podacima. Ovaj model ima istu formu kao i ARX model, ali uz to koristi zavisni mehanizam prilagodbe s eksponencijalnim zaboravljanjem koje je definirano faktorom zaboravljanja λ. Algoritam faktora zaboravljanja opisan je u sljedećim jednadžbama. Promjena procjene parametara modela $\hat{\theta}(t)$ u vremenu *t* se postiže množenjem pogreške predviđanja potrošnje plina $y(t) - \hat{y}(t))$ s porastom (engl. *gain*) *K(t):*

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)(y(t) - \hat{y}(t)) \quad (3)$$

Porast *K(t)* izražava kao:

$$K(t) = Q(t)\psi(t) \quad (4)$$

gdje *ψ(t)* označava gradijent izlaznog rezultata modela $\hat{y}(t|\hat{\theta})$ uzimajući u obzir parametre $\hat{\theta}$ i $Q(t)$, koji je definiran kao:

$$Q(t) = \frac{Q(t-1)}{\lambda + \psi(t)^T Q(t-1)\psi(t)} \quad (5)$$

Stvarna vrijednost $Q(t)$ postiže se minimizacijom sljedeće funkcije u vremenu *t*:

$$\sum_{k=1}^{t} \lambda^{t-k} e^2(k) \quad (6)$$

Ovaj pristup odbacuje stara mjerenja eksponencijalno, tako da opservacija u kojoj je prikazan prijašnji τ ima vrijednost jednaku umnošku λ^τ s vrijednošću posljednje opservacije. $\tau = 1/(1-\lambda)$ predstavlja horizont pamćenja (engl. *memory horizont*) ovog algoritma. Mjerenja koja su starija od $\tau = 1/(1-\lambda)$ obično imaju vrijednost manju od 0.3. λ se naziva faktor zaboravljanja i vrijednost mu je u pravilu pozitivna, između 0.97 i 0.995.

3.1.3. Nelinearni modeli

a) Model umjetne neuronske mreže (engl. *Artificial neuron network model*)

Umjetne neuronske mreže (NN) su automatizirana obrada podataka pogodna za izvršavanje na računalu te su primjer statističkog modela za procjenu potrošnje plina definirane kao predviđajuće (engl. *feed-forward*) mreže koje sadrže jedan skriveni sloj (engl. *hidden layer*) sa sigmoidnim neuronima i izlazni sloj s linearnom funkcijom. Princip rada "kopira" rad ljudskog mozga. Broj skrivenih neurona L je parametar s otvorenom mogučnošću oblikovanja. Jedan od glavnih problema pri izradi neuronske mreže je odabrati optimalan broj neurona, slojeva i ulaza u neuronsku mrežu. Mreži je potreban dovoljan broj parametara da na temelju ulaznih podataka nauči (engl. *train*) ponašanje samog sustava potrošnje prirodnog plina. Međutim, ako se broj parametara može usporediti s količinom podataka o prijašnjoj potrošnji plina, proces učenja mreže neće biti potpuno precizan. U tom slučaju, mreža gubi svoje svojstvo poopćavanja te će naučiti šum (koji je prisutan u svim podacima), što u konačnici rezultira smanjenom preciznošću procjenjivanja potrošnje plina. Stoga je važno imati što veću količinu podataka o prijašnjoj potrošnji prirodnog plina.

Izlaz (engl. *output*) neuronske mreže je predstavljen opći izrazon ulaznih podataka *u*, na primjer, za mrežu s *K* veličinom ulaznog podatka $u = \{u_1, u_{2,\dots,}u_K\}$, *L* neuroni u pojedinom skrivenom sloju, i pojedini izlaz su opisani matematičkom formulacijom:

$$y(t+1) = F_0\left(\sum_{j=0}^{L} w_j F_h\left(\sum_{i=0}^{K} w_{ji} u_i\right)\right) \tag{7}$$

F0 i Fh predstavljaju linearni izlaz (engl. linear output) i aktivacijsku, prijenosnu funkciju skrivenog. NN mreže uče pomoću algoritma korekcije pogreške i postoje 3 vrste: Levenberg-Marquardt algoritam, algoritam širenja unatrag (engl. *back-propagation*) i Gauss Newton algoritam.

b) Model promjenjive neuronske mreže (engl. *adaptive neuron network model*)

Promjenjiva neuronska mreža (RNN) ima istu strukturu kao neuronska mreža, ali s dodatnom mogučnošću mrežne prilagodbe (engl. *online adaptation*). Mrežna prilagodba se postiže algoritmon širenja pogreške unazad (engl. *back-propagation algorythm*) definiran

brzinom učenja (engl. *learn rate*) η = 0.01 i konstantom momenta i α = 0.1. Matematička formulacija mrežne prilagodbe je:

$$\Delta w_{ij}(t+1) = (1-\alpha)\eta\delta_j o_i + \alpha\Delta w_{ij}(t) \quad (8)$$

$\Delta w_{ij}(t+1)$ označava promjenu težine (pondera) $w_{ij}(t+1)$ u vezi neurona *i* s neuronom *j* u vremenu $(t+1)$, veličina η je brzina učenja, δ_j je šum (engl. *error signal*) neurona *j*, o_i je izlazna vrijednost neurona *i* (ujedno i ulaz trenutnog neurona *j*) te konstanta momenta α koja predstavlja promjenu težine (pondera) u prethodnom trenutku. Svaka jedinica (neuron) skrivenog sloja prima sumu svih ponderiranih inputa, a zatim primjenjuje aktivacijsku funkciju da bi se dobio izlaz neurona skrivenog sloja. Izračunati output mreže uspoređuje se sa stvarnim outputom (stvarne potrošnje plina) , i razlika između njih je lokalna greška ε . Greška se zatim koristi za podešavanje pondera (težina) u ulaznom sloju prema algoritmu širenja unazad. Ovaj tip neuronske mreže koristi se u ovom radu.

c) Model regresije potpornog vektora (engl. *support vector regression model*)

Model regresije potpornog vektora (SVR) definiran je linearnon kombinacijon N kernelovih funkcija opisanih s φ_i , težinama (engl. *weights*) w_i i pomacima (engl. *bias*) *b*:

$$y(t+1) = \sum_{i=0}^{n} \varphi_i w_i(u_i) + b \quad (9)$$

Proračun SVR rješenja ovisi o parametru poopćavanja (engl. *generalization parameter*) *C* i o kernelovom parametru γ. Optimalne vrijednosti obaju parametara određuju se metodom krosvalidacije (engl. *crossvalidation method*) kroz širok raspon mogućih vrijednosti.

3.2. Konstruirani modeli za procjenu potrošnje plina

U odlomku su opisani modeli, odnosno metode kojima se izračunava procjenjena potrošnja plina u ovom radu. Dva modela spadaju u linearne modele, tip autoregresivnog modela s egzogenim ulaznim podacima: linearni model s 2 zone i linearni model s 3 zone. Tri modela su nelinearna, tip umjetne promjenjive neuronske mreže: ANN s 1 zonom, ANN s 2 zone i ANN s 3 zone. Posljednji model je usporedni, model temperaturne korelacije.

3.2.1. Spyder platforma i programski jezik Python

Svi modeli konstruirani su u otvorenoj softverskoj (engl. *open-source software*) platformi Spyder za znanstveno programiranje s programskim jezikom Pythonom. Spyder je u principu skup alata za uređivanje, analizu, traženje pogrešaka i funkcionalnost za određene skupine i baze podataka. Izdaje se pod MIT licencom. Spyder nudi integraciju i rad s poznatim znanstvenim paketima i bibliotekama. Pri izradi modela za procjenu potrošnje plina koristili su se sljedeći znanstveni paketi: NumPy, SciPy, Pandas, Matplotlib. Ulazni podaci učitani su iz MS Excel datoteke *combine_ALL_cijela_godina.xlsx*.

Python je objektno orijentiran, programski jezik s dinamičkom programskom strukturom koji nastaje u kasnim osamdesetim godinama od strane nizozemskog programera Guido van Rossuma koji je imao viziju stvaranja novog jezika koji će biti jednostavan i ugodan za korištenje. Pythonova jednostavna i lako učljiva sintaksa ističe se čitljivošću te smanjuje troškove programskog održavanja (engl. *program maintenance*). Posebno je popularan u znanosti i inženjerstvu jer je pogodan za obradu velikog broja podataka.

3.2.2. Tipovi podataka koji se koriste pri konstruiranju modela za procjenu potrošnje plina

Za konstrukciju modela za procjenu potrošnje plina u okviru ovog rada koriste se 3 vrste ulaznih podataka. Modeli su konstruirani, odnosno "istrenirani" prikupljenim podacima iz vremenskog razdoblja od 1.4.2017. godine do 31.3.2018. godine za područje Republike Hrvatske i to s njihovim satnim vrijednostima. Svi podaci su spremljeni u MS Excel datoteku *combine_ALL_cijela_godina.xlsx*. Obzirom da Plinacro d.o.o. u svojim dnevnim izvještajima o potrošnji plina započinje plinski dan u 7 sati i 15 minuta (u principu, plinski dan započinje u 6:00 h i traje do sutradan u 6:00 h, ali prvo mjerenje

potrošnje plina je u 7:15 h), onda se za potrebe ovog rada, kao početak dana uzima isto vrijeme (satne potrošnje za 1 dan su u rasponu od 7:15 h do 6:15 h idućeg dana).

Prva skupina podataka odnosi se na prijašnje potrošnje plina. Podaci o satnoj potrošnji prirodnog plina u RH preuzeti su od tvrtke Plinacro d.o.o., koja obavlja funkciju operatora plinskog transportnog sustava te objavljuje satne potrošnje svakodnevno na svojoj internet stranici. Satne potrošnje, od 1.4.2017. godine do 31.3.2018. godine, iskazane su u jedinici energije, kWh. Od 1. siječnja 2012. godine na tržištu prirodnog plina Republike Hrvatske primjenjuje se mjerna jedinica kWh (kWh/h), što znači da operator transportnog sustava objavljuje iznose donjih ogrjevnih vrijednosti prirodnog plina koje zatim opskrbljivači i voditelji bilančnih skupina koriste pri nominacijama korištenja transportnog sustava. Za objašnjenje ove pretvorbe, za donju ogrjevnu vrijednost uzima se H_d= 35 083 kJ/Sm^3 (podatak preuzet od opskrbljivača Gradska plinara Zagreb). Sm^3 je standardni metar kubni prirodnog plina pri standardnim uvjetima tlaka od 101.325 Pa i temperaturom od 288,15 K (odnosno 15,15 °C). Izgaranjem 1 Sm^3 oslobađa se 9.2607 kWh energije. (NN128/11, 2011). Stoga se uvodi izraz energetska vrijednost pojedinih količina plina, E_{plin}. Tablica 3-1 prikazuje dio baze podataka *combine_ALL_cijela_godina.xlsx* gdje su sadržane potrošnje plina i koja je konstuirana u MS Excelu.

Tablica 3-1 Prikaz dijela tablice *combine_ALL_cijela_godina.xlsx* datoteke s potrošnjama plina konstruirane u MS Excelu (Plinacro, 2018)

g	m	d	7:15:00	8:15:00	9:15:00	10:15:00	11:15:00
2017	4	1	866535	1036404	990584	843603	748884
2017	4	2	661003	789778	802733	760227	714767
2017	4	3	1151952	1213147	1049079	914510	801562
2017	4	4	1074675	1144281	1055680	951100	869276
2017	4	5	1116466	1230288	1186338	1108033	1064619
2017	4	6	1232971	1321935	1314261	1263368	1146677
2017	4	7	1529631	1692508	1624573	1390707	1230448
2017	4	8	1065764	1266833	1248101	1140541	1030711
2017	4	9	876911	1013135	1042587	958207	854090
2017	4	10	1343713	1376082	1216757	1002028	866428

Druga skupina podataka se odnosi na satne vrijednosti vanjske temperature iskazane u stupnjevima Celzijusa (°C), a prikupljene su s internet servisa za vremensku prognozu Timeanddate za razdoblje od 1.4.2017. godine do 31.3.2018. godine.

Temperature zraka za potrebe ovog rada uzete su za grad Zagreb obzirom da je na tom području najveći broj potrošača plina. Tablica 3-2 prikazuje dio baze podataka *combine_ALL_cijela_godina.xlsx* gdje su sadržane satne temperature i koja je konstuirana u MS Excelu.

Tablica 3-2 Prikaz dijela tablice *combine_ALL_cijela_godina.xlsx* datoteke sa satnim temperaturama konstruirane u MS Excelu (Timeanddate, 2018)

g	m	d	7:00	8:00	9:00	10:00	11:00
2017	4	1	11	14	18	20	23
2017	4	2	13	17	19	21	23
2017	4	3	13	16	20	21	23
2017	4	4	16	17	17	17	19
2017	4	5	12	13	14	15	15
2017	4	6	11	11	13	13	14
2017	4	7	7	10	13	14	15
2017	4	8	11	14	15	18	19
2017	4	9	12	15	17	18	19
2017	4	10	12	15	18	20	22

Temperature Timeanddate servis mjeri na pune sate, a plinski sat prema kojemu se referira počinje na četvrtini punog sata, međutim odstupanje se zanemaruje jer promjena temperature u 15 min je beznačajna u okviru ovog rada.

Treći set podataka je potražnja električne energije (engl. *electricity demand*), odnosno potrošnja električne energije Republike Hrvatske iskazana u MWh za razdoblje od 1.4.2017. godine do 31.3.2018. godine. Potrošnje električne energije preuzete su s ENTSO-E platforme za transparentnost. Tablica 3-3 prikazuje dio baze podataka *combine_ALL_cijela_godina.xlsx* gdje su sadržane potrošnje električne energije i koja je konstuirana u MS Excelu.

Tablica 3-3 Prikaz dijela tablice *combine_ALL_cijela_godina.xlsx* datoteke s potrošnjama električne energije konstruirane u MS Excelu (ENTSO-E, 2018)

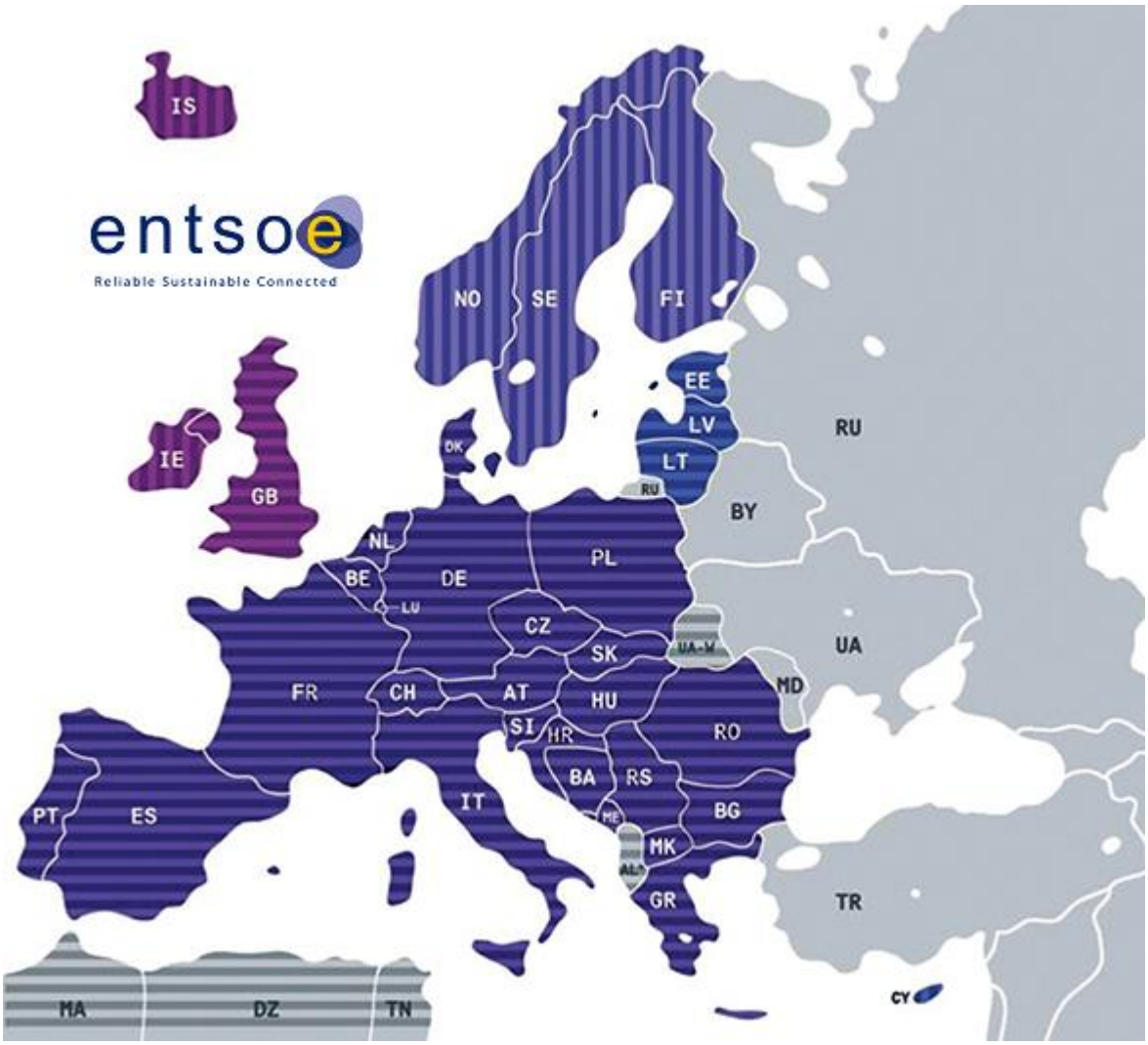
g	m	d	7:30:	08:30	09:30	10:30	11:30
2017	4	1	1564	1726	1860	1927	1922
2017	4	2	1385	1479	1623	1755	1844
2017	4	3	1718	1952	2036	2047	2012
2017	4	4	1751	1982	2057	2073	2050
2017	4	5	1797	2049	2131	2157	2126
2017	4	6	1810	2059	2158	2196	2176
2017	4	7	1863	2115	2194	2205	2159
2017	4	8	1605	1799	1937	1992	1983

2017	4	9	1521	1545	1685	1796	1856
2017	4	10	1759	2039	2107	2116	2073

Treba napomenti da postoji odstupanje kod vremena mjerenja od 15 min jer ENTSO –E mjeri na polovici punog sata, a potrošnje iz Plinacro-a su za prvu četvrtinu sata. Međutim to malo odstupanje se za okvir ovog diplomskog rada i zbog dostupnosti podataka zanemaruje.

3.2.2.1. ENTSO-E

ENTOSE-E (engl. *the European Network of Transmission System Operators for Electricity*) je europska mreža operatera prijenosnih sustava električne energije. Operateri prijenosnih sustava su subjekti nezavisni od ostalih sudionika na tržištu električne energije i odgovorni su za prijenos (transmisiju) električne energije na glavnim, visokonaponskim električnim mrežama. ENTSO-E omogućuje uvid u električnu mrežu svim sudionicima na tržištu (proizvođačima, trgovcima, opskrbljivačima, distributerima te direktnim potrošačima) prema pravilima o transparentnosti i protiv diskrimancije na tržištu. Zbog sigurnosti opskrbe, ENTSO-E garantira siguran rad te održavanje sustava. U mnogim državama Europe, operateri prijenosnih sustava su zaduženi za razvoj infrastrukture prijenosnog sustava. 43 operatera prijenosnih sustava iz 36 zemalja su članovi ENTOSE-E grupacije. Slika 3-1 prikazuje političku kartu Europe s osjenčanim zemljama članicama ENTSO-E grupacije.



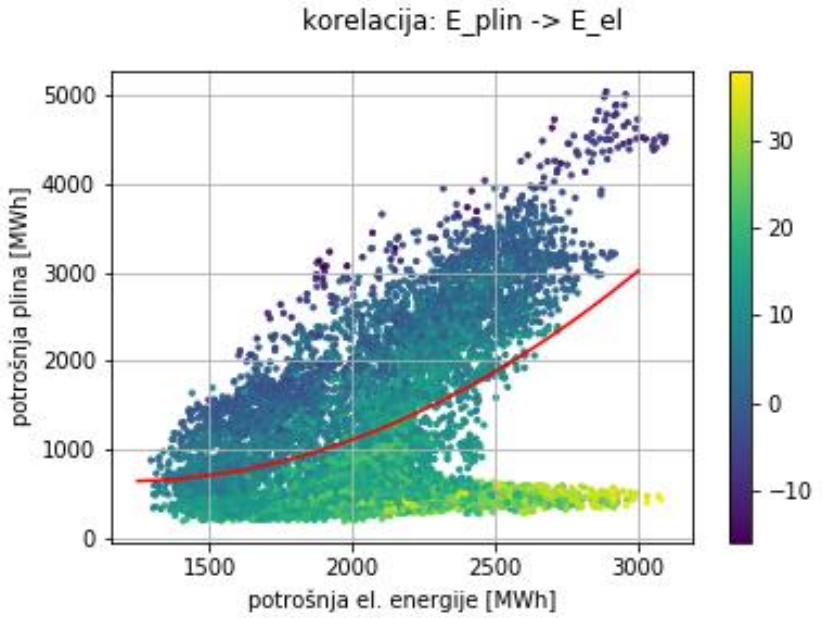
Slika 3-1 Prikaz zemalja članica ENTSO-E grupacije (Energoinfo, 2018)

3.2.2.2. Hrvatski prijenosni sustav

Hrvatski elektroenergetski sustav (EES) se sastoji od proizvodnih postrojenja, prijenosne i distribucijske mreže te potrošača električne energije na području Republike Hrvatske. Zbog sigurne opskrbe kupaca te razmjene električnom energijom, hrvatski EES povezan je s EES-ovima susjednih država kao i ostalim sustavima zemalja članica ENTSO-E platforme te zajedno čine sinkronu mrežu kontinentalne Europe. Također, zajedno s EES sustavima Slovenije i BiH čini upravljački blok SLO - HR – BIH unutar ENTSO-E udruženja. Svojom veličinom hrvatski EES spada u manje sustave u Europi i regulacijsko je područje koje vodi HOPS (Hrvatski operator prijenosnog sustava) (HOPS, 2017).

3.2.3. Grafički prikaz skupina podataka s odgovarajućim korelacijama

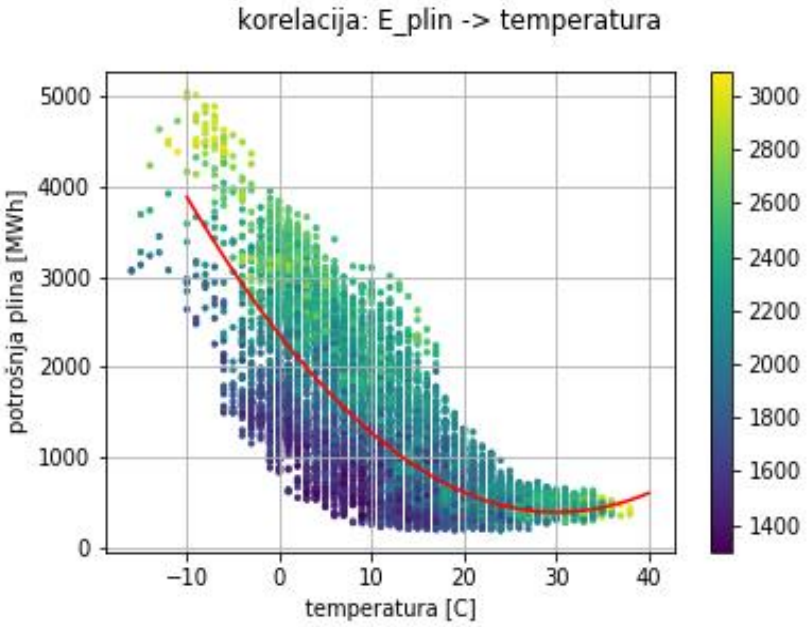
Učitavanje ulaznih podataka te prikaz njihovog međuodnosa obavlja se u kodu main. Taj kod je osnova za konstrukciju svih modela u ovom radu. Slika 3-2 prikazuje međuodnos (korelaciju) potrošnji plina (iskazane kao energetska vrijednost u MWh) i potrošnji električne energije (također u MWh) u razdoblju 1.4.2017. do 31.3.2018. godine. Svakoj točki na grafu (1 točka predstavlja 1 satnu vrijednost) pridružene su odgovarajuće vrijednosti temperature čije su vrijednosti prikazane brojčanom skalom u °C i bojama prema legendi s desne strane dijagrama.



Slika 3-2 Međuodnos (korelacija) potrošnje plina i potrošnje električne energije od 1.4.2017. do 31.3.2018. godine te s pripadajućom satnom temperaturom zraka za svaku točku

Iz dijagrama je vidljivo da su najveće potrošnje plina kada je vanjska temperatura najniža (tamno plava boja), jer je primarna svrha potrošnje prirodnog plina kao energent za potrebe grijanja, te kada je potrošnja električne energije najviša jer brojne elektrane u RH kao glavno gorivo koriste prirodni plin. Upravo zato postoji bazna potrošnja plina koja ne ovisi o vanjskoj temperaturi. Međutim, uočljivo je da je potreba za električnom energijom vrlo visoka (od 2000 do 3000 MWh) kada je vanjska temperatura 30 °C i više (točke obojana nijansama žute boje). To su potrebe rashladnih uređaja te povećanog broja potrošača za vrijeme ljetnih mjeseci pa je tako plin posredno i energent za potrebe hlađenja. Interpolacijom točaka dobivena je korelacija koja odgovara polinomu 2. stupnja (crvena krivulja na dijagramu).

Slika 3-3 prikazuje međuodnos (korelaciju) satnih potrošnji plina (u MWh) i satnih temperatura °C u razdoblju 1.4.2017. do 31.3.2018. godine. Svakoj točki na grafu (1 točka predstavlja 1 satnu vrijednost) pridružene su odgovarajuće vrijednosti potrošnji električne energiji čije su vrijednosti prikazane brojčanom skalom u MWh i bojama prema legendi s desne strane dijagrama.



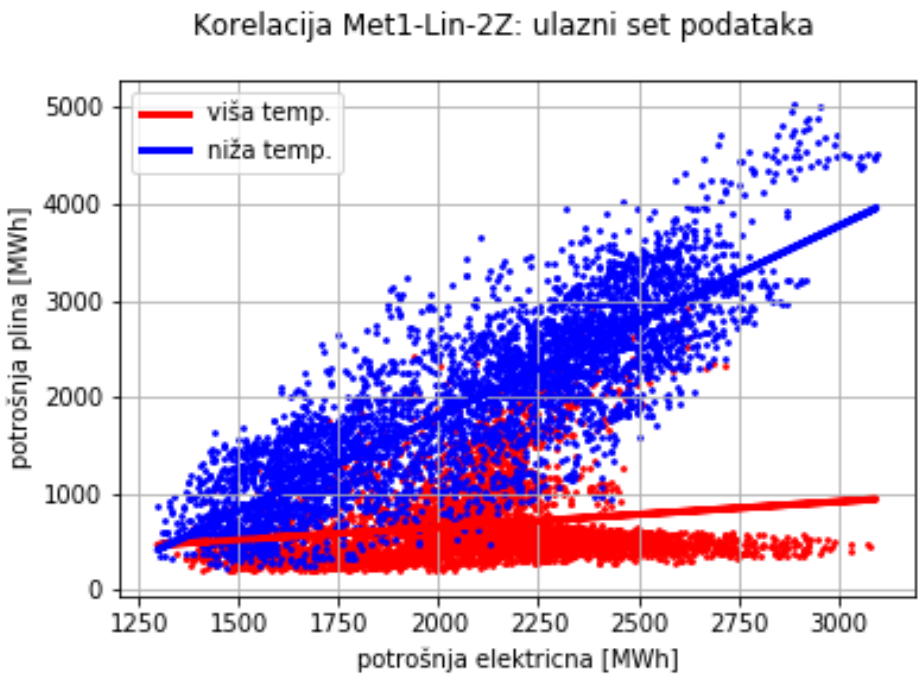
Slika 3-3 Međuodnos (korelacija) potrošnje plina i temperatura zraka od 1.4.2017. do 31.3.2018. godine te s pripadajućom satnom potrošnjom električne energije za svaku točku

Iz dijagrama je jasno vidljivo da potrošnja plina opada s porastom temperature i obratno. Pri ektremnim temperaturama zraka (oko 35 do 40 °C) potrošnja električne energije je visoka (od 2600 do preko 3000 MWh) zbog potreba rashladnih uređaja te većeg broja potrošača (te točke su osjenčane nijansama žute boje) što je već viđeno na prethodnom dijagramu koji prikazuje Slika 3-2. Najveća potrošnja plina (preko 4000 MWh) je u slučaju izrazito hladnih dana (temperature niže od -5 °C) u kombinaciji s velikom potrošnjom električne energije (opet žuto osjenačane točke). Korelacijska krivulja koja najbolje opisuje odnos potrošnje plina i temperature je polinom 2. stupnja koji je označen crvenom bojom na dijagramu..

3.2.4. Linearni model s 2 temperaturne zone

Linearni model s 2 temperaturne zone je tip autoregresivnog modela s egzogenim ulaznim podacima (ARX). Slika 3-4 prikazuje podjelu ulaznih podataka (potrošnji plina,

potrošnji električne energije i temperature) na zonu više (crvena boja) i niže temperature (plava boja).



Slika 3-4 Ulazni set podataka za linearni 2Z model s podjelom na zonu niže i više temperature

Kao aktivacijska temperatura (engl. *mid-point temperature*) T_0, odnosno temperatura prema kojoj se odvija podjela na 2 zone uzima se 11 °C. Kada je temperatura u nekoj točki (točka odgovara satnoj vrijednosti u skupu podataka) viša od T_0, model izračunava potrošnju za zonu više temperature po funkciji:

$$E_{plin_V} = m_1 E_{el} + n_1 \tag{10}$$

U funkciji, E_{el} je potrošnja električne energije, m_1 nagib aproskimiranog pravca i n_1 odječak aproksimiranog pravca. U principu, m_1 i n_1 su koeficijenti dobiveni naredbom np.polyfit , koja izračunava koeficijente polinoma (u ovom slučaju, pravca) koji "najbolje" povezuju ulazne podatke (prijašnju potrošnju plina i potrošnju el. energije) po metodi najmanjih kvadrata. Kada je vrijednost temperature manja ili jednaka od T_0 tada model računa potrošnju plina za zonu niže temperature po funkciji:

$$E_{plin_N} = m_2 E_{el} + n_2 \tag{11}$$

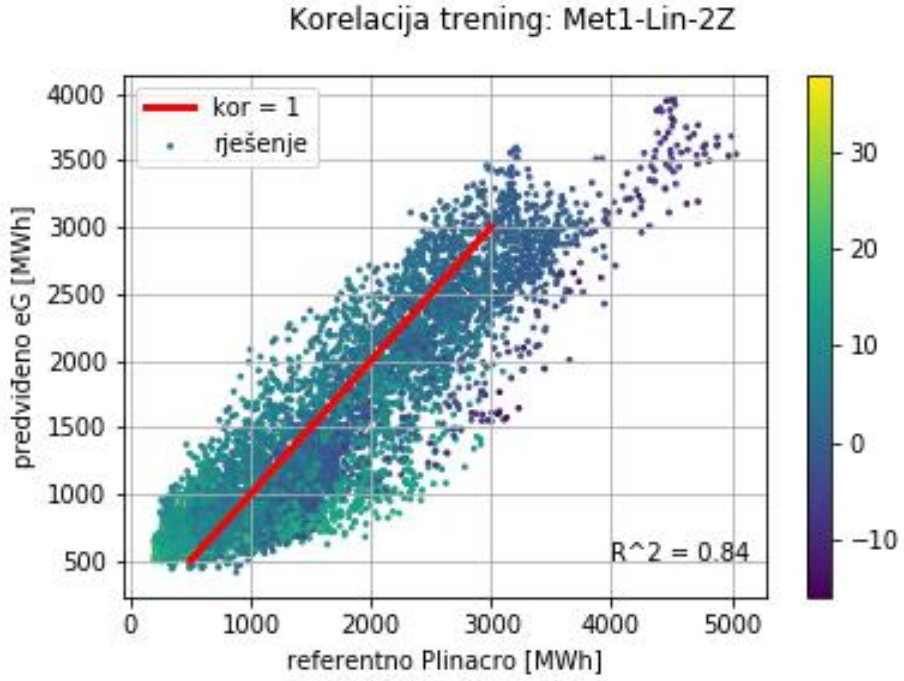
U funkciji, E_{el} je potrošnja električne energije, m_2 nagib aproskimiranog pravca i n_2 odsječak aproksimiranog pravca. Kao i kod slučaja s višom temperaturom m_2 i n_2 su u biti koeficijenti dobiveni naredbom np.polyfit. Sigmoidna funkcija u kojoj je sadržana ulazna

temperatura (T) se koristi da se u modelu razdvoji proračun konačne potrošnje po zoni više (ako njezina vrijednost ide prema 1) i zoni niže temperature (ako ide prema 0). Točka infleksije je kada je temperatura jednaka *mid-point* temperaturi i vrijednost funkcije tada iznosi 0.5 :

$$\sigma = \frac{1}{1 + e^{-k(T-T_0)}} \tag{12}$$

T_0 je aktivacijska temperatura, a k konstanta koja utječe na strminu funkcije. Potrošnja plina koju linearni 2Z model računa u konačnici je:

$$E_{plin} = \sigma E_{plin_V} + (1 - \sigma) E_{plin_N} \tag{13}$$



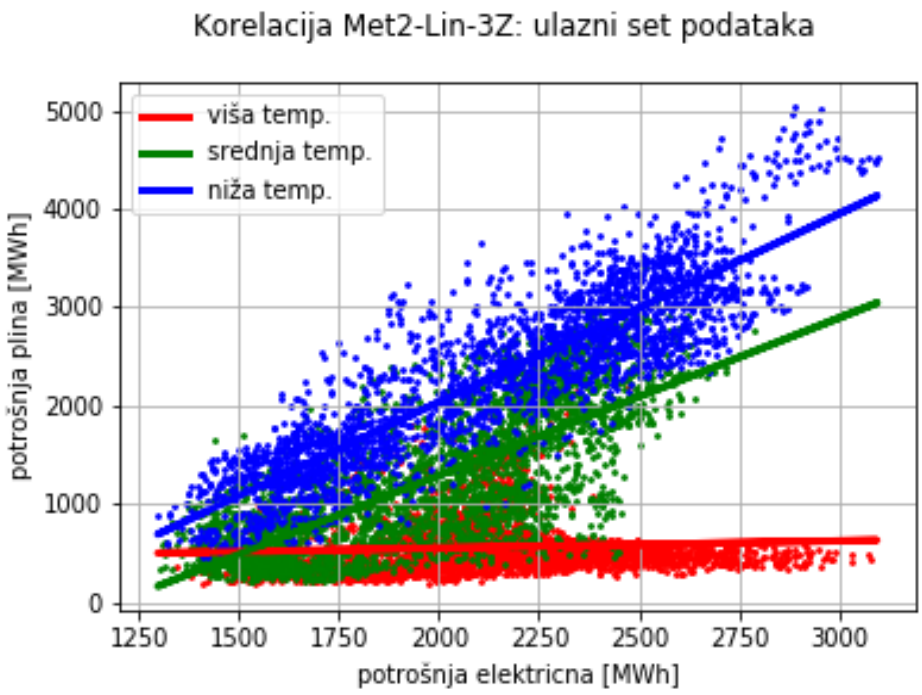
Slika 3-5 Korelacija potrošnji dobivenih linearnim 2Z modelom i stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2

Slika 3-5 prikazuje korelacijski pravac (crvena boja) za izračunate (predviđene) potrošnje plina linearnim 2Z modelom (y-os) i pravih potrošnji plina (referentne iz Plinacro-a) za razdoblje od 1.4.2017. do 31.3.2018. Koeficijent determinacije R^2 pokazuje koliko "točno" model prognozira potrošnju plina u usporedbi sa stvarnim rezultatuma. R^2 za linearni 2Z model iznosi 0.84.

3.2.5. Linearni model s 3 temperaturne zone

Linearni model s 3 temperaturne zone je tip autoregresivnog modela s egzogenim ulaznim podacima (ARX) kao i dvozonski model. Slika 3-6 prikazuje podjelu ulaznih

podataka (potrošnji plina, potrošnji električne energije i temperature) na zonu više (crvena boja), srednje (zelena boja) i niže temperature (plava boja).



Slika 3-6 Ulazni set podataka za linearni 3Z model s podjelom na zonu niske,srednje i visoke temperature

Razlika 3Z modela od 2Z modela sastoji se u tome što su sada 2 aktivacijske temperature (prema kojima se formiraju zone). To su T_1 koja iznosi 5 °C i T_2 koja je 15 °C (zona više temperature za $T > T_2$, zona srednje temperature za $T_2 > T > T_1$ i zona niže temperature za $T < T_1$). Potrošnje za svaku zonu model računa s odgovorajućim funkcijama prema temperaturnoj podjeli:

$$E_{plin_N} = m_1 E_{el} + n_1 \quad (14)$$
$$E_{plin_S} = m_2 E_{el} + n_2 \quad (15)$$
$$E_{plin_V} = m_3 E_{el} + n_3 \quad (16)$$

Parametri m iz sve tri funkcije predstavljaju aproksimirane nagibe pravaca, n parametri su odsječci aproksimiranih pravaca kao što je bilo i kod 2Z modela. Znači to su koeficijenti dobiveni naredbom np.polyfit, koja izračunava koeficijente polinoma (u ovom slučaju, pravca) koji "najbolje" povezuju ulazne podatke (prijašnju potrošnju plina i potrošnju el. energije) po metodi najmanjih kvadrata. 3Z model zahtjeva 2 sigmoidne funkcije, jednu za T_1 , drugu za T_2.:

$$\sigma_{1,2} = \frac{1}{1 + e^{-k(T - T_{1,2})}} \quad (17)$$

Sigmoida σ_1 razdvaja proračun konačne potrošnje po višoj (kad ide prema 1) i srednjoj zoni (kad ide prema 0). Sigmoida σ_2 razdvaja proračun konačne potrošnje po srednjoj (kad ide prema 1) po zoni niže temperature (kad ide prema 0). Zatim se uvodi dodatna temperatura T_{AR}, aritmetička sredina T_1 i T_2 prema kojoj će model računati potrošnju:

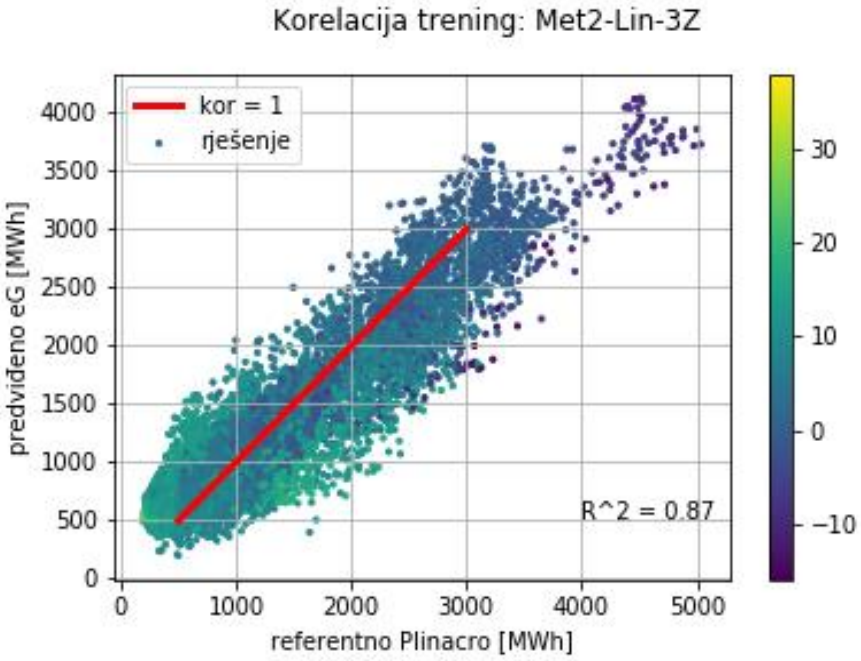
$$T_{AR} = \frac{(T_1 + T_2)}{2} \tag{18}$$

Linearni 3Z model predviđa potrošnju prema dvije formule. Ako je $T > T_{AR}$ onda računa prema formuli:

$$E_{plin} = \sigma_1 E_{plin_V} + (1 - \sigma_1) E_{plin_S} \tag{19}$$

Ukoliko je $T < T_{AR}$ onda model računa prema formuli:

$$E_{plin} = \sigma_2 E_{plin_S} + (1 - \sigma_2) E_{plin_N} \tag{20}$$

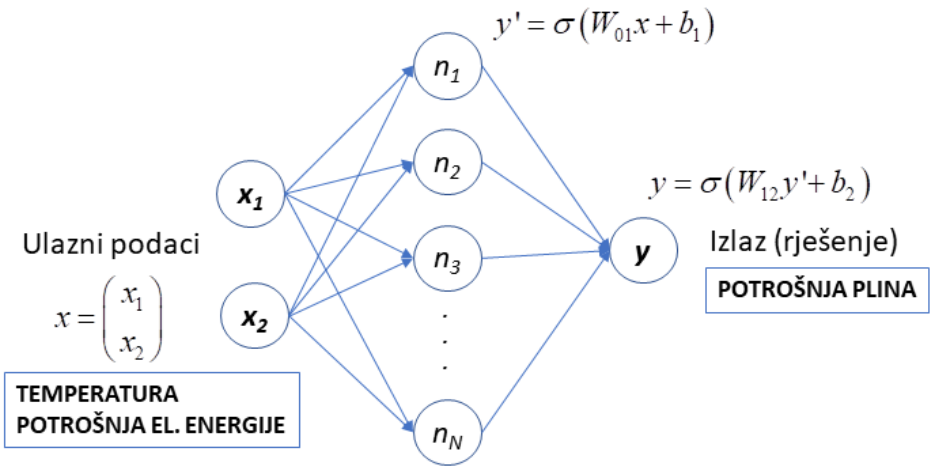


Slika 3-7 Korelacija potrošnji dobivenih linearnim 3Z modelom i stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2

Slika 3-7 prikazuje korelacijski pravac (crvena boja) za izračunate (predviđene) potrošnje plina linearnim 3Z modelom (y-os) i pravih potrošnji plina (referentne iz Plinacro-a) za razdoblje od 1.4.2017. do 31.3.2018. Koeficijent determinacije R^2 iznosi 0.87.

3.2.6. Model umjetne neuronske mreže s 1 temperaturnom zonom (ANN s 1 zonom)

Model umjetne neuronske mreže s 1 temperaturnom zonom (ANN s 1 zonom) pripada skupini nelinearnih modela. Slika 3-8 prikazuje shemu jednog seta podataka s pripadajućim slojevima.



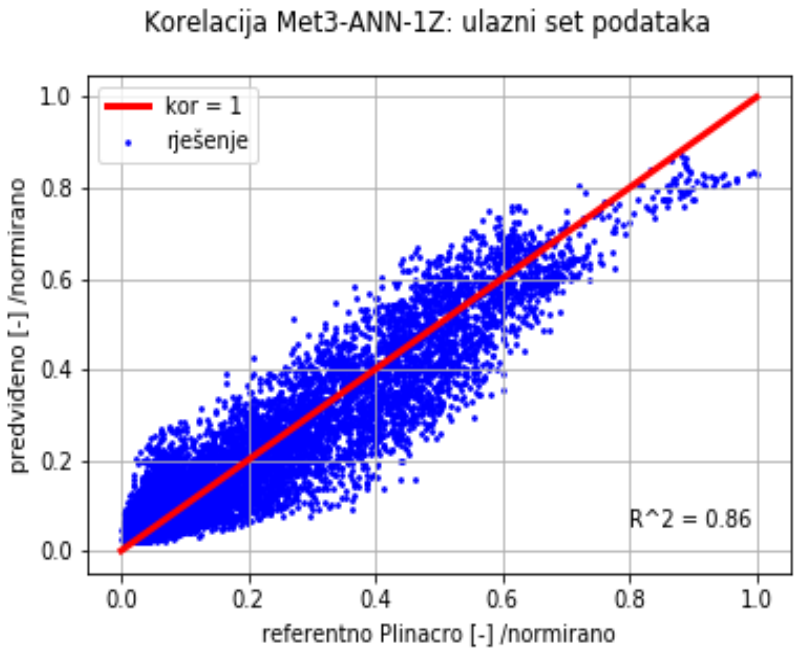
Slika 3-8 Shema izgleda jednog seta podataka sa slojevima i pripadajućim formulama

Ovakvih setova podataka postoji 8760 u cijeloj neuronskoj mreži što odgovara broju sati u ulaznim podacima (1 set podataka čine 1 satna temperatura, 1 satna potrošnja el.energije, 1 satna potrošnja plina). Iako je uobičajno da se prilikom izrade mreže setovi podataka podjele na 3 dijela (dio za učenje ili treniranje, validaciju i testiranje) u ovom diplomskom radu se za učenje mreže koriste se svi ulazni podaci (svih 8760 temperatura i potrošnji el. energije), a za testiranje se koriste sve stvarne potrošnje plina (8760 satnih potrošnji). Učenje mreže i učitavanja *inputa* opisano je u prvom dijelu koda. Prvi sloj čine ulazni podaci x_1 i x_2 (temperature i potrošnje el.energije). $n_1, n_2 \dots n_N$ su neuroni u skrivenom sloju (srednji sloj sa slike) s pripadajućom formulama. Učenje mreže odvija se propagiranjem ulaznih podataka kroz slojeve te promjenom vrijednosti "težine" među varijablama i te težine su označene su s w. Pomoću izračunate pogreške na izlazu podešavaju se težine od skrivenog do izlaznog sloja i tako redom do ulaznog sloja s ciljem minimiziranja izlazne pogreške. U izlaznom sloju *output* mreže se uspoređuje s stvarnom potrošnjom plina te se računa pogreška. Pogreška je razlika između željenog i dobivenog *outputa*. Pogreška se tako

propagira unatrag i istovremeno se podešavaju težine (ponderi). b (engl. *bias*) je pomak odnosno dodatni težinski faktor koji omogućava modelu što veću preciznost. Početna vrijednost težina i pomaka je nasumično određena naredbom np.random. Varijabla y' predstavlja povijesnu potrošnju plina (iz kreirane baze podataka) prema kojoj mreža "uči" kakav izlaz (engl.*output*) treba dati. Prijenosna, aktivacijska funkcija prema kojoj se računa izlaz neurona je sigmoidna, σ i definirana je na samom početku koda. Ona omogućuje da se težine postupno mjenjaju (da mreža gradijentalno uči) te se tako dobivaju bolji izlazni rezultati:

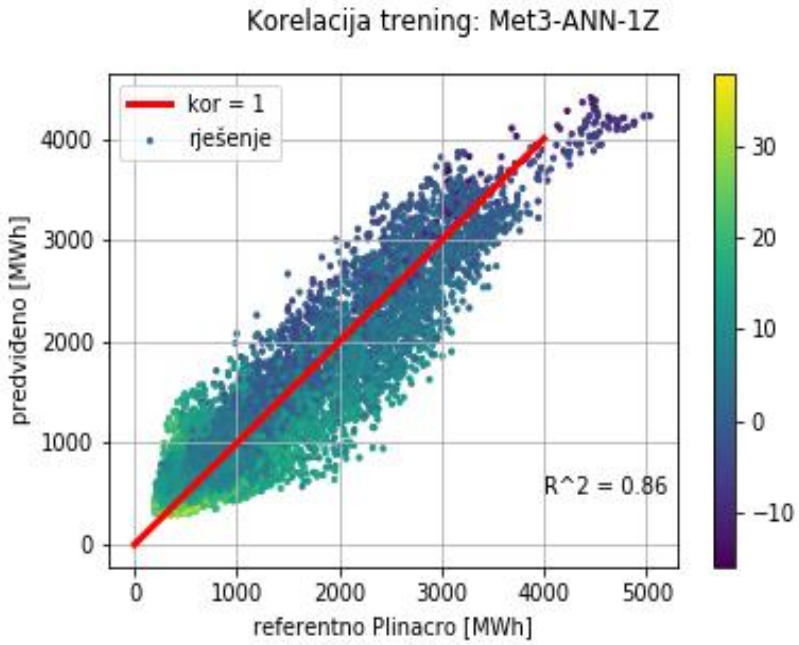
$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{21}$$

x predstavlja ulazne u neuronsku mrežu. U drugom dijelu koda ulazne veličine i stvarne potrošnje se normaliziraju na jediničnu vrijednost zato što temperatura i potrošnja el. energije nemaju jednaku mjerne jedinice te nisu istog reda veličine. Također se normalizira i stvarna potrošnja plina da bi se output iz mreže mogao uspoređivati odnosno testirati. Slika 3-9 prikazuje korelaciju normaliziranih potrošnji plina dobivenih ANN modelom s 1 zonom s normaliziranim stvarnim potrošnjama i pripadajućim koeficijentom determinacije R^2.



Slika 3-9 Korelacija normaliziranih potrošnji dobivenih ANN modelom s 1 zonom i pravih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2

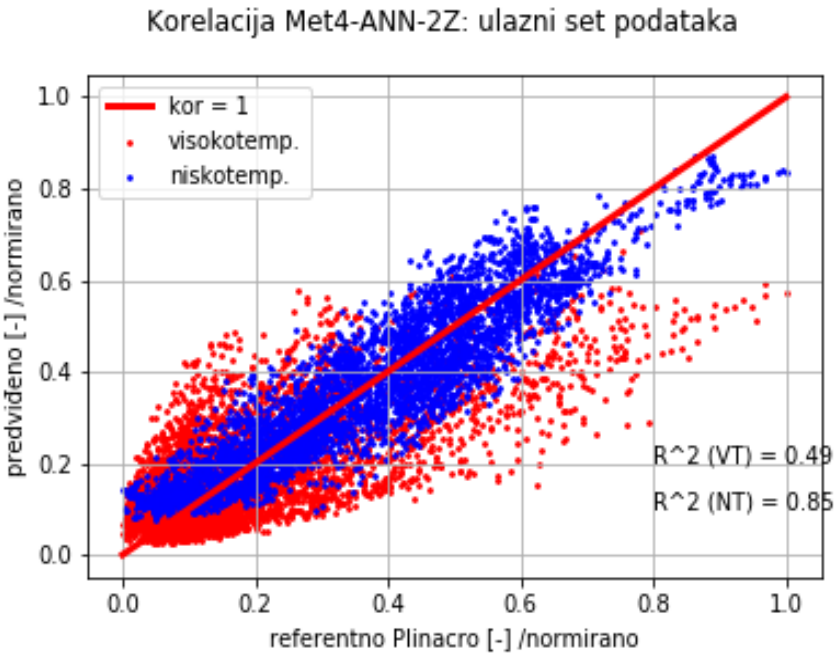
Na samom kraju koda, radi se odnormalizacija, odnosno svođenje na stvarne vrijednosti. Slika 3-10 prikazuje korelacijski pravac (crvena boja) za izračunate (predviđene) potrošnje plina modelom umjetne neuronske mreže s 1 temperaturnom zonom (y-os) i pravih potrošnji plina (referentne iz Plinacro-a) za razdoblje od 1.4.2017. do 31.3.2018. Koeficijent determinacije R^2 iznosi 0.86.



Slika 3-10 Korelacija potrošnji dobivenih s ANN modelom s 1 zonom i stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2

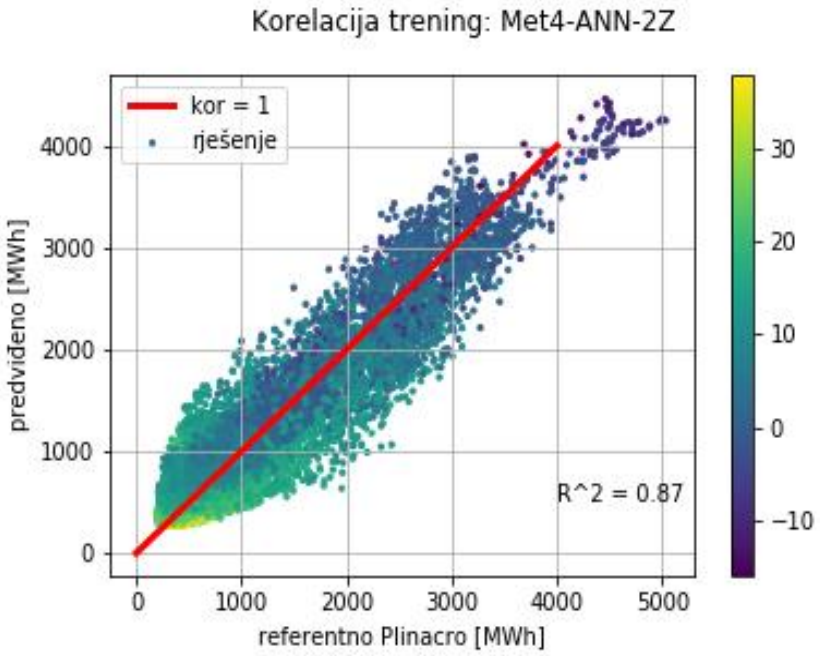
3.2.7. Model umjetne neuronske mreže s 2 temperaturne zonom (ANN s 2 zone)

Model neuronske mreže s 2 temperaturne zone (ANN s 2 zone) pripada skupini nelinearnih modela. Slika 3-8 također vrijedi i za ovaj model jer je struktura mreže i neurona ista, samo kod ANN modela s 2 zone ulazni podaci su podjeljeni na 2 temperaturne zone: zonu više temperature (VT) i zonu niže temperature (NT). Temperatura koja dijeli set podataka T_m je 11°C. Slika 3-11 prikazuje korelaciju normaliziranih potrošnji plina za zonu visoke i zonu niske temperature dobivenih ANN modelom s 2 zone i normaliziranih stvarnih potrošnji (isto tako su u modelu normalizirani i ulazni podaci) s pripadajućim koeficijentima determinacije R^2. Zona više temperature (VT) obojena je crvenom bojom, a niže temperature (NT) plavom bojom.



Slika 3-11 Korelacija normaliziranih potrošnji dobivenih ANN modelom s 2 zone i pravih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijenti determinacije R^2 za svaku zonu

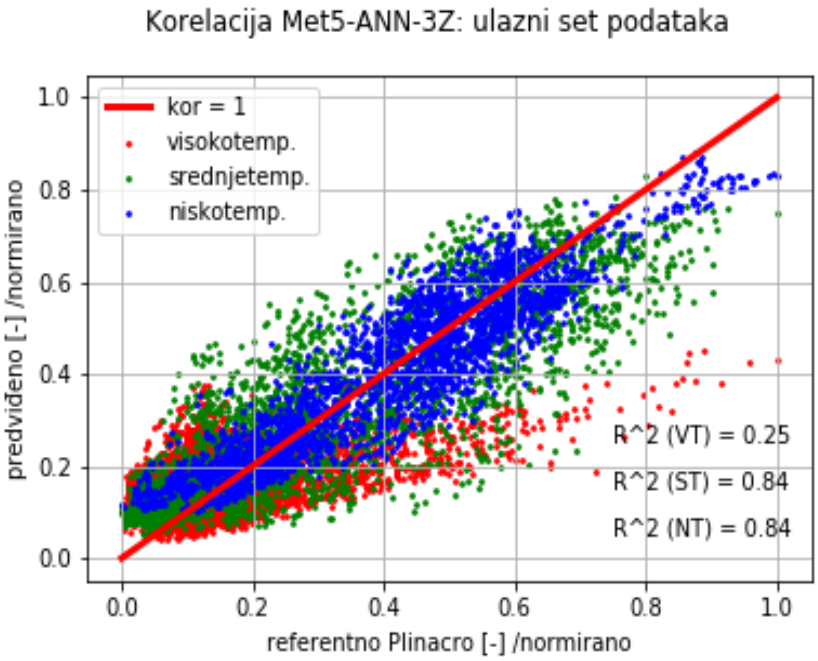
Iz slike je vidljivo da će pri višim temperaturama (ljetni mjeseci), odnosno manjom potrošnjom plina mreža davati lošije rezultate što pokazuje pripadajući $R^2 = 0.49$, dok je za zonu niže temperature (sezona grijanja) $R^2 = 0.85$,. Slika 3-12 prikazuje korelaciju potrošnji dobivenih ANN modelom s 2 zone i stvarnih potrošnji plina (nakon odnormalizacije) od 1.4.2017. do 31.3.2018. godine sa stvarnim vrijednostima (brojčane skale u MWh) te prikazuje u principu stvarni *output* modela. Koeficijent determinacije ovog modela, R^2 iznosi 0.87.



Slika 3-12 Korelacija potrošnji dobivenih ANN modelom s 2 zone i stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2

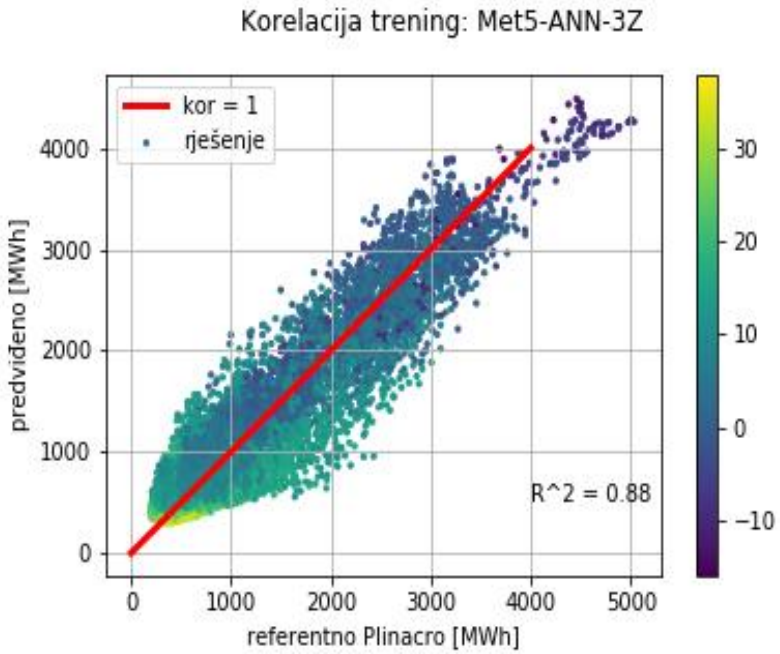
3.2.8. Model umjetne neuronske mreže s 3 temperaturne zonom (ANN s 3 zone)

Model neuronske mreže s 3 temperaturne zone (ANN s 3 zone) pripada skupini nelinearnih modela. Slika 3-8 koja prikazuje shemu jednog seta podataka s pripadajućim slojevima vrijedi i za ovaj model. Jedini detalj koji razlikuje ovaj model i dvozonski je uvođenje još jedne temperaturne zone. U ovom modelu su 2 aktivacijske temperature koje dijele set podataka (prema kojima se formiraju zone). To su T_1 koja iznosi 5 °C i T_2 koja je 15 °C (zona više temperature za $T > T_2$, zona srednje temperature za $T_2 > T > T_1$ i zona niže temperature za $T < T_1$). Slika 3-13 prikazuje korelaciju normaliziranih potrošnji plina za zonu visoke, srednje i niske temperature dobivenih ANN modelom s 3 zone i normaliziranih stvarnih potrošnji s pripadajućim koeficijentima determinacije R^2. Zona više temperature (VT) obojena je crvenom bojom, srednje temperature (ST) zelenom, a niže temperature (NT) plavom bojom.



Slika 3-13 Korelacija normaliziranih potrošnji dobivenih ANN modelom s 3 zone i normaliziranih stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijenti determinacije R^2 za svaku zonu

Iz slike je vidljivo da će pri višim dnevnim temperaturama (VT zona), $T > 15°C$, odnosno kod perioda manje potrošnje plina za svrhu grijanja mreža davati najlošije rezultate što pokazuje pripadajući R^2 = 0.25. Za zonu srednje i niže temperature mreža prognozira potrošnju s jednakom točnošću, R^2 = 84. Slika 3-14 prikazuje korelaciju potrošnji dobivenih ANN modelom s 3 zone i stvarnih potrošnji plina (nakon odnormalizacije) od 1.4.2017. do 31.3.2018. godine sa stvarnim vrijednostima (brojčane skale u MWh). Koeficijent determinacije ovog modela, R^2 iznosi 0.88.



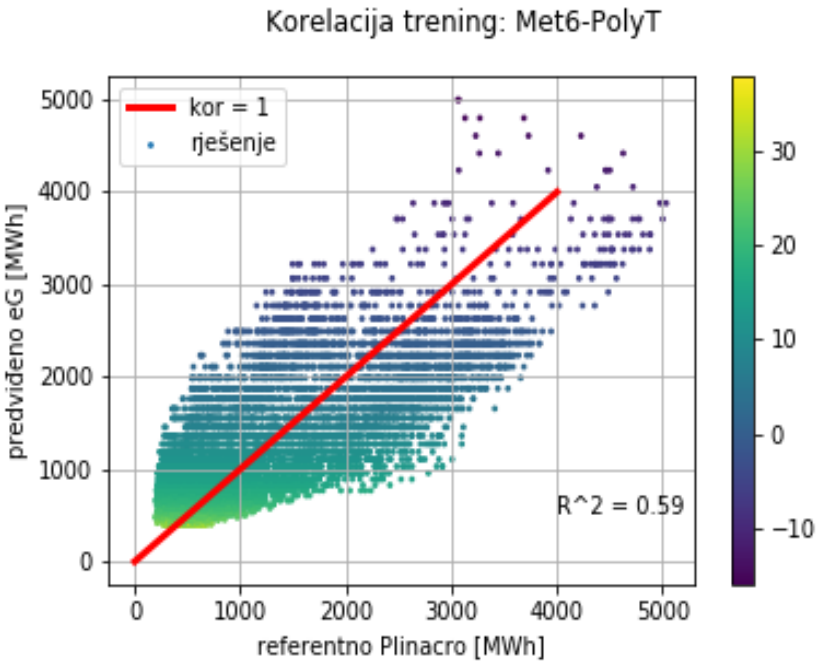
Slika 3-14 Korelacija potrošnji dobivenih ANN modelom s 3 zone i stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2

3.2.9. Model temperaturne korelacije

Model temperaturne korelacije je usporedni model i najjednostavniji je od konstruiranih modela. U ovom modelu, temperatura i potrošnja plina su direktno povezani prema sljedećoj formuli:

$$E_{plin} = aT^2 + bT + c \tag{22}$$

Slika 3-3 prikazuje korelaciju potrošnji plina i temperature iz baze ulaznih podataka. Korelacijska krivulja je polinom 2. stupnja te na temelju toga što polinom 2. stupnja najbolje opisuje korelaciju potrošnji i temperatura je konstruiran ovaj model. Koeficijenti *a, b* i *c* dobiveni su dobiveni naredbom np.polyfit , koja izračunava koeficijente aproksimirajućeg pravca koji "najbolje" povezuju ulazne podatke regresijskom analizom ulaznih podataka (potrošnji i temperature) od 1.4.2017. do 31.3.2018. Slika 3-15 prikazuje korelaciju potrošnji dobivenih modelom temperaturne korelacije i stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine sa stvarnim vrijednostima (brojčane skale u MWh). R^2 iznosi 0.59.



Slika 3-15 Korelacija potrošnji dobivenih modelom temperaturne korelacije i stvarnih potrošnji plina od 1.4.2017. do 31.3.2018. godine te pripadajući koeficijent determinacije R^2

4. ANALIZA REZULTATA I DISKUSIJA

Da bi se analizirala kvaliteta modela odnosno točnost u procjenjivanju potrošnje pline potrebno je sagledati i usporediti koeficijente determinacije R^2 , obzirom da je R^2 specifičan pokazatelj reprezentativnosti regresije, odnosno povezanosti potrošnji koje modeli računaju s stvarnim potrošnjama. Model je reprezentativniji što je koeficijent bliže jedinici. Tablica **4-1** prikazuje popis konstruiranih modela s pripadajućim koeficijentima determinacije R^2.

Tablica 4-1 Tablični prikaz reprezentativnosti konstruiranih modela

Matematički model:	**Koeficijent determinacije R^2**
Linearni model s 2 zone	0.84
Linearni model s 3 zone	0.87
ANN s 1 zonom	0.86
ANN s 2 zone	0.87
ANN s 3 zone	0.88
Model temperaturne korelacije	0.59

Tablica **4-2** je druga tablica koja služi za referiranje, u kojoj su opisane jačine korelacijske veze za odgovarajuću vrijednost koeficijenta determinacije.

Tablica 4-2 Chadockova ljestvica (Hinkle i Wiersma, 2003)

R^2	**Tumačenje**
0.90 - 1	Vrlo jaka korelacijska veza
0.70 – 0.90	Jaka korelacija veza
0.50 – 0.70	Umjerena korelacijska veza
0.30 . 0.50	Slaba korelacijska veza
0 – 0.30	Odsutnost korelacijske veze

Iako potrošnja plina uvelike ovisi o vanjskoj temperaturi obzirom da je jedna od glavnih upotreba prirodnog plina za svrhu grijanja, model temperaturne korelacije se pokazao kao najmanje prezican pri prognoziranju potrošnje. Njegova korelacijska veza se tumači kao umjereno jaka i to pokazuje kako prognoze koje se oslanjaju samo na temperaturu nisu prihvatljive. Linearni modeli s 2 i 3 temperaturne zone, umjetne neuronske mreže s 1, 2 i 3 zone imaju jake korelacijske veze. Pokazalo se da se razdvajanjem setova podataka na temperaturne zone povećava točnost, iako ne mnogo (kod linearnih modela R^2 se povećava za 0.03, kod umjetnih neuronskih mreža za 0.1). Kao najpreciznija metoda pokazala se umjetna neuronska mreža s 3 temperaturne zone. U praksi, modeli za predviđanje potrošnje plina moraju imati iznimnu preciznost (R^2 je 0.994 pa i više) te koriste veći broj ulaznih podataka (povijesne potrošnje, temperaturu, tip dana, sunčevo zračenje, oborine i sl.) i daleko su složeniji modeli. (Soldo et al.,2014). Važno je napomenuti da su se vanjske

temperature uzimale za Grad Zagreb (najveći broj potrošača), a ostali podaci su za RH. To je u konačnici rezultiralo nešto lošijim korelacijama, međutim zbog dostupnosti podataka i informacija te okvira ovog rada nije bilo moguće sve uskladiti. Obzirom da su se koristila samo 3 tipa podataka, te da je kod 3 modela (linearni 3Z, ANN 2Z i 3Z) R^2 blizu 0.90 što se smatra vrlo jakom korelacijskom vezom, rezultati se mogu smatrati zadovoljavajućim.

5. ZAKLJUČAK

U ovom diplomskom radu samostalno su konstruirani modeli za procjenu potrošnje plina na temelju već postojećih pomoću platforme Spyder i programskog jezika Python. Uredba Komisije (EU) br. 312/2014 o uspostavljanju mrežnih pravila o uravnoteženju plina transportnih mreža donesena 3. ožujka 2014. godine propisuje pravila uravnoteženja plinskog sustava u Europskoj uniji te ju je Hrvatska ulaskom u EU dužna implementirati. Upravo zato se javlja potreba za što boljim i preciznijim modelima jer što je prognoza točnija to će biti manja potreba za uravnoteženjem sustava i plaćat će se manje naknade za odstupanje.

U prvom dijelu rada opisana je energetska bilanca potrošnje prirodnog plina u RH da bi se dobio dojam o plinskom sustavu i količinama plina u RH te za što se troše. Zatim je napravljen kratki pregled naprednih, matematičkih modela koji se općenito koriste te su oni poslužili kao ideja i temelj za konstrukciju vlastitih.

Jedna od ključnih faktora za izradu modela bilo je prikupiti ulazne podatke za relativno duži vremenski period. Prikupljene su satne vrijednosti povjesnih potrošnji prirodnog plina, potrošnje električne energije i vanjske temperature. Podaci su prikupljeni za vremenski period 1.4.2017. do 31.3.2018 što je ukupno 8760 sati. To znači da se pri kreiranju modela koristilo 8760 setova podataka što čini respektabilnu bazu podataka.

Konačno, dobijeni su korelacijski odnosi potrošnji plina dobivenih konstruiranim modelima i stvarnih potrošnji plina te su se izračunali koeficijenti determinacije R^2 prema kojima se odredila preciznost prognoziranja pojedinog modela. Pokazalo se kako prognoziranje potrošnje plina oslanjajući se samo na vanjsku temperaturu nije pouzdano, što je pokazao model temperaturne korelacije već je potreban veći broj vanjskih faktora. Kao najtočniji model pokazao se model umjetne neuronske mreže s 3 temperaturne zone s pripadajućim R^2 koji je iznosio 0.88. Iako se u praksi od modela zahtjeva preciznost gdje je R^2 0.994 pa i više, ANN 3Z model se približio vrijednosti koeficijenta od 0.90 koji se tumači kao vrlo jaka korelacijska veza. Obzirom da su komercijalne metode vrlo komplicirani modeli s osjetno većim brojem ulaznih podataka, samostalna konstrukcija modela može se smatrati uspješnom.

6. LITERATURA

1. ENERGETSKI INSTITUT HRVOJE POŽAR, 2016. Energija u Hrvatskoj 2016:, Ministarstvo zaštite okoliša i energetike Republike Hrvatske, Zagreb.
2. HINKLE, DENNIS; WIERSMA, WILLIAM; JURS, STEPHEN, 2003. Applied statistics for the behavioral sciences. 5. izdanje. Boston: London : Hi Marketing.
3. LI, Y., FU, Y., LI, H., ZHANG S., 2009. The Improved Training Algorithm of Back Propagation Neural Network with Self-adaptive Learning Rate. *2009,* International Conference on Computational Intelligence and Natural Computing, str. 73-76.
4. NARODNE NOVINE 18/2018, Pravila o organizaciji tržišta plina, Članak 4., Članak 13.
5. NARODNE NOVINE 18/2018, Zakon o tržištu plina , Članak 3.
6. NARODNE NOVINE 128/2011 , Pravilnik o izmjenama i dopunama pravnilnika za organizaciju tržišta plina, Članak 2a
7. POTOČNIK, P.; SOLDO, B.; ŠIMUNOVIĆ, G.; ŠARIĆ, T.; JEROMEN, A.; GOVEKAR, E., 2014. Comparison of static and adaptive models for short-term residental natural gas forecasting in Croatia, Applied Energy 129, str. 94-103.
8. SOLDO, B., 2012. Forecasting natural gas consumption, Applied energy 92, str. 26-37.
9. SZOPLIK, J., 2015. Forecasting of natural gas consumption with artifical neural networks. Energy 85, str. 208-220.

MREŽNI IZVORI

1. BAŠIĆ, BOJANA; ČUPIĆ, MARKO; ŠNAJDER, JAN, 2008. URL https://www.fer.hr/_download/repository/UmjetneNeuronskeMreze.pdf (10.11.2018.)
2. ENERGOINFO 2017 URLhttp://www.energoinfo.com/wp-content/uploads/2017/06/entso-big.png (5.11.2018.)

3. ENTOSE, 2015.
URL https://www.entsoe.eu/data/power-stats/hourly_load/
(24.9.2018.)

4. FISTONIĆ, M.; BILANDŽIJA, M.; VUKADINOVIĆ, B.,2013. Predviđanje potrošnje prirodnog plina - Zahtjevi i iskustva u primjeni neuronskih mreža,
URL http://www.ptmg.hr/media/Predvianjepotronjeprirodnogplina.pdf
(16.9.2018.)
5. GASNETWORKS, 2014.
URL https://www.gasnetworks.ie/corporate/company/our-network/Methodology-for-forecasting-gas-demand.pdf
(4.11.2018.)
6. HOPS, 2017.
URLhttps://www.hops.hr/wps/portal/hr/web/hees
(5.10.2018.)
7. PLINACRO, 2018.
URL http://www.plinacro.hr/default.aspx?id=172
(11.9.2018.)
8. PREGLED ZAHTJEVA UREDBE KOMISIJE (EU) BR. 312/2014 O USPOSTAVLJANJU MREŽNIH PRAVILA O URAVNOTEŽENJU PLINA TRANSPORTNIH MREŽA
URLhttp://www.plinacro.hr/UserDocsImages/dokumenti/Uskla%C4%91enje%20s%20Uredbom%20EU%20312-2014%20-%20pregled%20zahtjeva_10.3.2017_122505.pdf
(15.10.2018.)
9. PYTHON SOFTWARE FOUNDATION, 2018. (US)
URL https://www.python.org/doc/essays/blurb/
(5.11.2018.)
10. SANKEYMATIC, 2018.
URL http://sankeymatic.com/build/
(19.11.2018.)
11. THE SPYDER PROJECT CONTRIBUTORS, 2018.
URL https://pypi.org/project/spyder/
(3.11.2018.)

12. TIMEANDDATE, 2015.
URLhttps://www.timeanddate.com/weather/croatia/zagreb/historic
(23.9.2018.)

PRILOZI

Kod main za učitavanje ulaznih podataka i prikaz njihovog međuodnosa

```
1.
2. #
3. from IPython import get_ipython
4. get_ipython().magic('reset -sf') #//resetiraj
   (izbrisi) varijable
5. #
6. import pandas as pd
7. import numpy as np
8. import matplotlib.pyplot as plt
9. from   pandas import DataFrame
10.   from   scipy.stats.stats import pearsonr
11.   #
12.   data_file = 'combine_ALL_cijela_godina.xlsx'
13.   sheets = ['gas_kWh', 'elec_MWh', 'temp_C']
14.   #
15.   df0in = pd.read_excel(data_file,sheet_name =
   sheets[0])
16.   df1in = pd.read_excel(data_file,sheet_name =
   sheets[1])
17.   df2in = pd.read_excel(data_file,sheet_name =
   sheets[2])
18.   #
19.   df0 = df0in.values
20.   df0 = np.multiply(df0,0.001)
21.   df1 = df1in.values
22.   df2 = df2in.values
23.   #
24.   N1    = df0.shape[0]
25.   N2    = df0.shape[1]
26.   subL1 = N1
27.   subL2 = N2 - 3
28.   #
29.   input_to_ANN = np.empty([subL1*subL2, 3])
30.   #
31.   iRow = 0
32.   for i in range(0,N1):
33.       j1 = 1
34.       for j in range(3,N2):
35.           #print(j)
36.           input_to_ANN[iRow,0] = df0[i,j]
   # gas consumption in MWh
37.           input_to_ANN[iRow,1] = df1[i,j]
   # electricity consumption in MWh
38.           input_to_ANN[iRow,2] = df2[i,3 + (j1-1)*2 +
   1] # temperature in C
39.           j1 = j1 + 1
40.           iRow = iRow + 1
41.   # find Paerson correlation coefficient
42.   # gas consmpt. - electricity consmpt.
43.   a = input_to_ANN[:,0] # gas conspt
44.   b = input_to_ANN[:,1] # electricity conspt
45.   corr_1 = pearsonr(a,b)
46.   res1 = np.polyfit(b,a,2)
47.   # gas consmpt. - temperature
48.   a = input_to_ANN[:,0] # gas conspt
49.   b = input_to_ANN[:,2] # temp
50.   corr_2 = pearsonr(a,b)
```

```
51.  res2 = np.polyfit(b,a,2)
52.  #
53.  x1 = np.linspace(1250,3000,1000)
54.  x2 = np.linspace(-10,40,1000)
55.  y1  = res1[0]*x1**2 + res1[1]*x1 + res1[2]
56.  y2  = res2[0]*x2**2 + res2[1]*x2 + res2[2]
57.  # PLOTS
58.  fig = plt.figure(1)
59.  plt.scatter(input_to_ANN[:,1],input_to_ANN[:,0],5,i
     nput_to_ANN[:,2])
60.  plt.plot(x1,y1,'r')
61.  fig.suptitle('korelacija: E_plin -> E_el')
62.  plt.xlabel('potrošnja el. energije [MWh]')
63.  plt.ylabel('potrošnja plina [MWh]')
64.  plt.grid()
65.  plt.colorbar()
66.  plt.savefig('ulazni_podaci_korelacija_Epl_Eel.tiff'
     )
67.
68.  #
69.  fig = plt.figure(2)
70.  plt.scatter(input_to_ANN[:,2],input_to_ANN[:,0],5,i
     nput_to_ANN[:,1])
71.  plt.plot(x2,y2,'r')
72.  fig.suptitle('korelacija: E_plin -> temperatura')
73.  plt.xlabel('temperatura [C]')
74.  plt.ylabel('potrošnja plina [MWh]')
75.  plt.grid()
76.  plt.colorbar()
77.  plt.savefig('ulazni_podaci_korelacija_Epl_T.tiff')
78. #
```

Kod za linearni model s dvije temperaturne zone

```
1.
2. #
3. import numpy as np
4. import matplotlib.pyplot as plt
5. import pickle
6. #
7. #
8. #T0 = np.arange(-5,36,1) # mid-point temperatura
   (aktivacijska temperatura)
9. T0 = 11
10.     k  = 0.5 # steepness
11.     #
12.     a = input_to_ANN[:,0] # gas conspt
13.     b = input_to_ANN[:,1] # electricity conspt
14.     c = input_to_ANN[:,2] # temp
15.     #
16.     #N_T0 = len(T0)
17.     #for i in range (0,N_T0)
18.     N   = len(a)
19.     a1 = []
20.     b1 = []
21.     c1 = []
```

```
22.      a2 = []
23.      b2 = []
24.      c2 = []
25.      #
26.      for i in range (0,N):
27.          #
28.          #sig = 1/(1+np.exp(-k*(c[i]-T0))) # sigmoid
   funkcija
29.          #
30.          if c[i] > T0: #  T > T0 -> visokotemperaturni dio
31.              a1   = np.append(a1,a[i])
32.              b1   = np.append(b1,b[i])
33.              c1   = np.append(c1,c[i])
34.          else: #  T < T0 -> niskotemperaturni dio
35.              a2   = np.append(a2,a[i])
36.              b2   = np.append(b2,b[i])
37.              c2   = np.append(c2,c[i])
38.      #
39.      res1 = np.polyfit(b1,a1,1) # visa temperatura
40.      res2 = np.polyfit(b2,a2,1) # niza temperatura
41.      #
42.      e_G = np.zeros((N,1)) # <- inicijalizacija (=0)
   vektora koji sadrzi rjesenje za input podatke
43.      #
44.      for i in range(0,N):
45.          sig = 1/(1+np.exp(-k*(c[i]-T0))) # sigmoid
   function
46.          if np.isnan(sig):
47.              print(['yep'])
48.          e_G_highT = res1[0]*b[i] + res1[1]
49.          e_G_lowT  = res2[0]*b[i] + res2[1]
50.          e_G[i] = sig*e_G_highT + (1 - sig)*e_G_lowT # <-
   potrošnja plina koja je izračunata
51.      #
52.      # coefficent of correlation
53.      x = np.stack((a,e_G[:,0]))
54.      corrcoef = np.corrcoef(x)
55.      Rsquared = corrcoef**2
56.      # snimi u datoteku
57.      with open('m01_parametri.pkl', 'wb') as f:  # Python
   3: open(..., 'wb')
58.          pickle.dump([T0,k,res1,res2], f)
59.      #====================================================
60.      #PLOT
61.      fig = plt.figure(1)
62.      plt.plot(b,res1[0]*b + res1[1],'r',linewidth=3.0)
63.      plt.plot(b,res2[0]*b + res2[1],'b',linewidth=3.0)
64.      plt.scatter(b1,a1,3,'r')
65.      plt.scatter(b2,a2,3,'b')
66.      fig.suptitle('Korelacija Met1-Lin-2Z: ulazni set
   podataka')
67.      plt.xlabel('potrošnja elektricna [MWh]')
68.      plt.ylabel('potrošnja plina [MWh]')
69.      plt.grid()
70.      plt.legend(['viša temp.','niža temp.'])
71.      plt.savefig('m01_korelacija_ulazni_set.tiff')
72.      #
73.      fig = plt.figure(2)
74.      plt.plot([500,3000],[500,3000],'r',linewidth=3.0)
75.      plt.scatter(a,e_G,3,c)
76.      fig.suptitle('Korelacija trening: Met1-Lin-2Z')
```

```
77.      plt.xlabel('referentno Plinacro [MWh]')
78.      plt.ylabel('predviđeno eG [MWh]')
79.      plt.grid()
80.      plt.colorbar()
81.      plt.legend(['kor = 1','rješenje'])
82.      strtmp = 'R^2 = ' + str(round(Rsquared[0,1],2))
83.      plt.annotate(strtmp,xy=(2000, 500),xytext=(4000, 500))
84.      plt.savefig('m01_korelacija.tiff')
85.
```

Kod za linearni model s 3 temperaturne zone

```
1.
2. #
3. import matplotlib.pyplot as plt
4. import numpy as np
5. import pickle
6. #
7. T1 = 5
8. T2 = 15
9. k  = 0.5 # steepness
10.      #
11.      a = input_to_ANN[:,0] # gas conspt
12.      b = input_to_ANN[:,1] # electricity conspt
13.      c = input_to_ANN[:,2] # temp
14.      #
15.      #N_T0 = len(T0)
16.      #
17.      #for i in range (0, N, T0)
18.      N   = len(a)
19.      a1 = []
20.      b1 = []
21.      c1 = []
22.      a2 = []
23.      b2 = []
24.      c2 = []
25.      a3 = []
26.      b3 = []
27.      c3 = []
28.
29.      #
30.      for i in range (0,N):
31.          #
32.          #sig = 1/(1+np.exp(-k*(c[i]-T0))) # sigmoid
   funkcija
33.          #
34.          if c[i] > T2: #  T > T2 -> visokotemperaturni dio
35.              a1   = np.append(a1,a[i]) # gas
36.              b1   = np.append(b1,b[i]) # electricity
37.              c1   = np.append(c1,c[i]) # temp
38.          elif c[i] > T1 and c[i] <= T2 : # T2 > T > T1 ->
   srednjetemperaturni dio
39.              a2   = np.append(a2,a[i])
40.              b2   = np.append(b2,b[i])
41.              c2   = np.append(c2,c[i])
42.          elif c[i] <= T1: # T1 > T -> niskotemperaturni dio
43.              a3   = np.append(a3,a[i])
```

```
            b3   = np.append(b3,b[i])
            c3   = np.append(c3,c[i])
    #
    res1 = np.polyfit(b1,a1,1) # visa temperatura
    res2 = np.polyfit(b2,a2,1) # srednja temperatura
    res3 = np.polyfit(b3,a3,1) # niza temperatura
    #
    e_G = np.zeros((N,1)) # <- inicijalizacija (=0) vektora koji sadrzi rjesenje za input podatke
    #
    for i in range(0,N):
        sig1 = 1/(1+np.exp(-k*(c[i]-T2))) # sigmoid function 1
        sig2 = 1/(1+np.exp(-k*(c[i]-T1))) # sigmoid function 2
        e_G_highT = res1[0]*b[i] + res1[1] # visoka
        e_G_midT  = res2[0]*b[i] + res2[1] # srednja
        e_G_lowT  = res3[0]*b[i] + res3[1] # niska
        Tmid = (T1 + T2)*0.5
        if c[i] >= Tmid:
            e_G[i] = sig1*e_G_highT + (1 - sig1)*e_G_midT # <- potrošnja plina koja je izračunata
        elif c[i] < Tmid:
            e_G[i] = sig2*e_G_midT  + (1 - sig2)*e_G_lowT # <- potrošnja plina koja je izračunata
    #
    # coefficent of correlation
    x = np.stack((a,e_G[:,0]))
    corrcoef = np.corrcoef(x)
    Rsquared = corrcoef**2
    # snimi u datoteku
    with open('m02_parametri.pkl', 'wb') as f:  # Python 3: open(..., 'wb')
        pickle.dump([T1,T2,k,res1,res2,res3], f)
    #=====================================================
    #PLOT
    fig = plt.figure(1)
    plt.plot(b,res1[0]*b + res1[1],'r',linewidth=3.0)
    plt.plot(b,res2[0]*b + res2[1],'g',linewidth=3.0)
    plt.plot(b,res3[0]*b + res3[1],'b',linewidth=3.0)
    plt.scatter(b1,a1,3,'r')
    plt.scatter(b2,a2,3,'g')
    plt.scatter(b3,a3,3,'b')
    fig.suptitle('Korelacija Met2-Lin-3Z: ulazni set podataka')
    plt.xlabel('potrošnja elektricna [MWh]')
    plt.ylabel('potrošnja plina [MWh]')
    plt.grid()
    plt.legend(['viša temp.', 'srednja temp.', 'niža temp.'])
    plt.savefig('m02_korelacija_ulazni_set.tiff')
    #
    fig = plt.figure(2)
    plt.plot([500,3000],[500,3000],'r',linewidth=3.0)
    plt.scatter(a,e_G,5,c)
    plt.colorbar()
    fig.suptitle('Korelacija trening: Met2-Lin-2Z')
    plt.xlabel('referentno Plinacro [MWh]')
    plt.ylabel('predviđeno eG [MWh]')
    plt.grid()
    plt.legend(['kor = 1','rješenje'])
```

```
    strtmp = 'R^2 = ' + str(round(Rsquared[0,1],2))
    plt.annotate(strtmp,xy=(2000, 500),xytext=(4000, 500))
    plt.savefig('m02_korelacija.tiff')
    plt.show()
    #
```

Kod za model umjetne neuronske mreže s 1 temperaturnom zonom

```
#
import numpy as np
import matplotlib.pyplot as plt
import pickle
import pdb
#
# sigmoid funkcija
def nonlin(x,deriv=False):
        if (deriv==True):
            return x*(1-x)
        else:
            return 1/(1 + np.exp(-x))
    #
    # ANN function
    def metoda_ANN(x):
        X, Y, Niters, LR1, LR2, Nneurons = x
        #
        #pdb.set_trace()
        NinputsX = X.shape[1] # number of inputs
        NsetsX   = X.shape[0] # number of independent
variables
        # sigmoid function is defined outside
        #
        # seed random numbers
        np.random.seed(1)
        err = np.zeros(shape = (int(Niters),1)) # <- ovdje
se pohranjuje greška
        #
        # initialize weights
        W01 = 2*np.random.random((NinputsX,Nneurons)) - 1
# oko nule!
        W12 = 2*np.random.random((Nneurons,1)) - 1 # oko
nule!
        b1  = 2*np.random.random((1,Nneurons)) - 1 # oko
nule!
        b2  = 2*np.random.random((1,1)) - 1 # oko nule!
        #
        #
        for iter1 in range(int(Niters)):
            # print([iter])
            LR = LR1 + (LR2 - LR1)/int(Niters)*iter1
            if iter1 > 1e4:
                LR = LR2
            # forward propagation
            z1 = np.dot(X,W01) + b1
            g1 = nonlin(z1)
            z2 = np.dot(g1,W12) + b2
            g2 = nonlin(z2)      # main solution
            #print([g2])
            #
            # error W2
```

```
48.             err_z2 = Y.T - g2.T # main error
49.             #
50.             # backpropagation
51.             # output layer -> hidden layer
52.             der_z2 = nonlin(g2,True) # <- derivacija
   aktivacijske funkcije po z2
53.             d_z2   = np.multiply(err_z2.T,der_z2)
54.             d_b2   = d_z2
55.             #pdb.set_trace()
56.             #
57.             # hidden layer -> input layer
58.             err_z1 = np.dot(d_z2,W12.T)
59.             d_z1   = err_z1*nonlin(g1,True)
60.             d_b1   = d_z1
61.             #pdb.set trace()
62.             # update weights
63.             W12 = W12 + LR*np.dot(g1.T,d_z2)
64.             b2  = b2  + LR*np.dot(d_b2.T,np.ones(shape =
   (NsetsX,1)))
65.             W01 = W01 + LR*np.dot(X.T,d_z1)
66.             b1  = b1  + LR*np.dot(d_b1.T,np.ones(shape =
   (NsetsX,1))).T
67.             #
68.             err[iter1] = np.linalg.norm(err_z2) # <- L-2
   vektorska norma
69.             #print([b1])
70.             #print([iter1,err[iter1]])
71.             #
72.             #pdb.set_trace()
73.         #print([g2])
74.         return W01,W12,b1,b2,err,g2
75.     ##
76.     # INPUTS & CALL FUNCTION
77.     #
78.     i_test = 1 # <- test flag // staviti ==0 ako se želi
   testiati
79.     if i_test == 1:
80.         # data
81.         y  = input_to_ANN[:,0] # gas conspt
82.         x1 = input_to_ANN[:,1] # electricity conspt
83.         x2 = input_to_ANN[:,2] # temp
84.         # normalization of inputs between 0 and 1
85.         y_norm  = (y-min(y))/(max(y)-min(y))
86.         x1_norm = (x1-min(x1))/(max(x1)-min(x1))
87.         x2_norm = (x2-min(x2))/(max(x2)-min(x2))
88.         X = np.transpose(np.stack((x1_norm,x2_norm)))
89.         Y = y_norm
90.         #
91.     else:
92.         X = np.array([[0,0,1],[0,1,1],[1,0,1],[1,1,1]])
93.         Y = np.array([[0],[1],[1],[0]])
94.     #
95.     # ANN parameters
96.     N_iters   = 1e4
97.     N_neurons = 10
98.     LearningRateStart  = 0.0002
99.     LearningRateFinish = 0.001
100.    #
101.    # define inputs to function
102.    x    = (X, Y, int(N_iters), LearningRateStart,
   LearningRateFinish, N_neurons)
```

```
    #
    # call function
    [W01,W12,b1,b2,err,output] = metoda_ANN(x)
    #
    # coefficent of correlation
    xcorr = np.stack((Y,output[:,0]))
    corrcoef = np.corrcoef(xcorr)
    Rsquared = corrcoef**2
    #
    # testianje
    N_hours = len(y) # broj sati u analizi
    y_calculated = np.zeros(shape = (N_hours,1))
    x1_min = min(x1)
    x1_max = max(x1)
    x2_min = min(x2)
    x2_max = max(x2)
    y_min = min(y)
    y_max = max(y)
    #
    for t in range(0,N_hours):
        x_1  = (x1[t] - x1_min)/(x1_max - x1_min) # normirana potrošnja el. energije
        x_2  = (x2[t] - x2_min)/(x2_max - x2_min) # normirana temp.
        X_in = np.transpose(np.stack((x_1,x_2)))  # <- ektor ulaznih velicina
        ytmp1 = np.dot(X_in,W01) + b1
        ytmp2 = nonlin(ytmp1)
        ytmp3 = np.dot(ytmp2,W12) + b2
        ytmp4 = nonlin(ytmp3)
        y_calculated[t] = y_min + ytmp4*(y_max - y_min)
    #
    # coefficent of correlation
    xcorr = np.stack((y,y_calculated[:,0]))
    corrcoef = np.corrcoef(xcorr)
    Rsquared1 = corrcoef**2
    # snimi u datoteku
    with open('m03_parametri_ANN_1Z.pkl', 'wb') as f:  # Python 3: open(..., 'wb')
        pickle.dump([x1_min, x1_max, x2_min, x2_max, W01, b1, W12, b2, y_min, y_max], f)
    #=========================================================================
    # PLOT-ovi
    #=========================================================================
    fig = plt.figure(1)
    plt.loglog(err)
    fig.suptitle('Met3-ANN-1Z: Greška vs. iteracije')
    plt.xlabel('iteracije')
    plt.ylabel('greška (norma)')
    plt.grid(True,which="both",ls="-")
    plt.annotate(('error=' ,err[-1]),xy=(N_iters/2, (max(err)+min(err))/2), xytext=(N_iters/2, (max(err)+min(err))/2))
    plt.savefig('m03_ANN_greska.tiff')
    #
    fig = plt.figure(2)
    plt.plot([0,1],[0,1],'r',linewidth=3.0)
    plt.scatter(Y,output,3,'b')
    fig.suptitle('Korelacija Met3-ANN-1Z: ulazni set
```

```
    podataka')
155.    plt.xlabel('referentno Plinacro [-] /normirano')
156.    plt.ylabel('predviđeno [-] /normirano')
157.    plt.legend(['kor = 1','rješenje'])
158.    plt.grid()
159.    strtmp = 'R^2 = ' + str(round(Rsquared[0,1],2))
160.    plt.annotate(strtmp,xy=(0.2, 0.05),xytext=(0.8, 0.05))
161.    plt.savefig('m03_korelacija_ulazni_set.tiff')
162.    #
163.    fig = plt.figure(3)
164.    plt.plot([0,4000],[0,4000],'r',linewidth=3.0)
165.    plt.scatter(y,y_calculated,5,x2)
166.    fig.suptitle('Korelacija trening: Met3-ANN-1Z')
167.    plt.xlabel('referentno Plinacro [MWh]')
168.    plt.ylabel('predviđeno [MWh]')
169.    plt.legend(['kor = 1','rješenje'])
170.    plt.grid()
171.    plt.colorbar()
172.    strtmp = 'R^2 = ' + str(round(Rsquared1[0,1],2))
173.    plt.annotate(strtmp,xy=(2000, 500),xytext=(4000, 500))
174.    plt.savefig('m03_korelacija.tiff')
175.    #
176.    #fig = plt.figure(4)
177.    #plt.plot(y)
178.    #plt.plot(y_calculated)
179.    #fig.suptitle('vremenska potrosnja plina')
180.    #plt.xlabel('vrijeme [h]')
181.    #plt.ylabel('potrosnja plina [MWh]')
182.    #plt.legend(['referentna','predvidena'])
183.    #plt.grid()
```

Kod za model umjetne neuronske mreže s 2 temperaturne zone

```
1.
2. #
3. import numpy as np
4. import matplotlib.pyplot as plt
5. import pickle
6. #
7. # sigmoid funkcija
8. def nonlin(x,deriv=False):
9.     if (deriv==True):
10.            return x*(1-x)
11.        else:
12.            return 1/(1 + np.exp(-x))
13.    #
14.    Tm = 11 # temp koja dijeli set podataka
15.    ksig = 0.5
16.    # ANN function
17.    def metoda_ANN(x):
18.        X, Y, Niters, LR1, LR2, Nneurons = x
19.        #
20.        #pdb.set_trace()
21.        NinputsX = X.shape[1] # number of inputs
22.        NsetsX   = X.shape[0] # number of independent
   variables
23.        # sigmoid function is defined outside
24.        #
```

```
25.          # seed random numbers
26.          np.random.seed(1)
27.          err = np.zeros(shape = (int(Niters),1)) # <- ovdje
   se pohranjuje greška
28.          #
29.          # initialize weights
30.          W01 = 2*np.random.random((NinputsX,Nneurons)) - 1
   # oko nule!
31.          W12 = 2*np.random.random((Nneurons,1)) - 1 # oko
   nule!
32.          b1  = 2*np.random.random((1,Nneurons)) - 1 # oko
   nule!
33.          b2  = 2*np.random.random((1,1)) - 1 # oko nule!
34.          #
35.          #
36.          for iter1 in range(int(Niters)):
37.              #print([iter/Niters])
38.              LR = LR1 + (LR2 - LR1)/5e4*iter1
39.              if iter1 > 1e4:
40.                  LR = LR2
41.              # forward propagation
42.              z1 = np.dot(X,W01) + b1
43.              g1 = nonlin(z1)
44.              z2 = np.dot(g1,W12) + b2
45.              g2 = nonlin(z2)      # main solution
46.              #
47.              # error W2
48.              err_z2 = Y.T - g2.T # main error
49.              #
50.              # backpropagation
51.              # output layer -> hidden layer
52.              der_z2 = nonlin(g2,True) # <- derivacija
   aktivacijske funkcije po z2
53.              d_z2   = np.multiply(err_z2.T,der_z2)
54.              d_b2   = d_z2
55.              #pdb.set_trace()
56.              #
57.              # hidden layer -> input layer
58.              err_z1 = np.dot(d_z2,W12.T)
59.              d_z1   = err_z1*nonlin(g1,True)
60.              d_b1   = d_z1
61.              #pdb.set_trace()
62.              # update weights
63.              W12 = W12 + LR*np.dot(g1.T,d_z2)
64.              b2  = b2  + LR*np.dot(d_b2.T,np.ones(shape =
   (NsetsX,1)))
65.              W01 = W01 + LR*np.dot(X.T,d_z1)
66.              b1  = b1  + LR*np.dot(d_b1.T,np.ones(shape =
   (NsetsX,1))).T
67.              #pdb.set_trace()
68.              err[iter1] = np.linalg.norm(err_z2) # <- L-2
   vektorska norma
69.              #print([iter1,err[iter1]])
70.              #
71.          #print([g2])
72.          return W01,W12,b1,b2,err,g2
73.      ##
74.      # INPUTS & CALL FUNCTION
75.      #
76.      y_1 = []
77.      x1_1 = []
```

```
78.      x2_1 = []
79.      y_2 = []
80.      x1_2 = []
81.      x2_2 = []
82.      #
83.      i_test = 1 # <- test flag // staviti ==0 ako se želi
   testiati
84.      if i_test == 1:
85.          # data
86.          y  = input_to_ANN[:,0] # gas conspt
87.          x1 = input_to_ANN[:,1] # electricity conspt
88.          x2 = input_to_ANN[:,2] # temp
89.          #
90.          NN = y.shape[0]
91.          for i in range(0,NN):
92.              # normalization of inputs between 0 and 1
93.              if (x2[i] >= Tm):
94.                  y_1  = np.append(y_1,y[i])
95.                  x1_1 = np.append(x1_1,x1[i])
96.                  x2_1 = np.append(x2_1,x2[i])
97.              elif (x2[i] < Tm):
98.                  y_2  = np.append(y_2,y[i])
99.                  x1_2 = np.append(x1_2,x1[i])
100.                 x2_2 = np.append(x2_2,x2[i])
101.         #
102.         min_y1 = min(y_1)
103.         max_y1 = max(y_1)
104.         min_y2 = min(y_2)
105.         max_y2 = max(y_2)
106.         #
107.         y_norm_1  = (y_1-min_y1)/(max_y1-min_y1)
108.         x1_norm_1 = (x1_1-min(x1_1))/(max(x1_1)-min(x1_1))
109.         x2_norm_1 = (x2_1-min(x2_1))/(max(x2_1)-min(x2_1))
110.         X_1 =
   np.transpose(np.stack((x1_norm_1,x2_norm_1)))
111.         Y_1 = y_norm_1
112.         #
113.         y_norm_2  = (y_2-min_y2)/(max_y2-min_y2)
114.         x1_norm_2 = (x1_2-min(x1_2))/(max(x1_2)-min(x1_2))
115.         x2_norm_2 = (x2_2-min(x2_2))/(max(x2_2)-min(x2_2))
116.         X_2 =
   np.transpose(np.stack((x1_norm_2,x2_norm_2)))
117.         Y_2 = y_norm_2
118.         #
119.     else:
120.         X = np.array([[0,0,1],[0,1,1],[1,0,1],[1,1,1]])
121.         Y = np.array([[0],[1],[1],[0]])
122.     #
123.     # ANN parameters
124.     N_iters    = 1e4
125.     N_neurons = 10
126.     LearningRateStart  = 0.0002
127.     LearningRateFinish = 0.001
128.     #
129.     # define inputs to function
130.     xInput_1    = (X_1, Y_1, int(N_iters),
   LearningRateStart, LearningRateFinish, N_neurons)
131.     xInput_2    = (X_2, Y_2, int(N_iters),
   LearningRateStart, LearningRateFinish, N_neurons)
132.     #
133.     # call function
```

```
134.     [W01_1,W12_1,b1_1,b2_1,err_1,output_1] =
   metoda_ANN(xInput_1)
135.     [W01_2,W12_2,b1_2,b2_2,err_2,output_2] =
   metoda_ANN(xInput_2)
136.     #
137.     # coefficent of correlation
138.     xcorr_1 = np.stack((Y_1,output_1[:,0]))
139.     corrcoef_1 = np.corrcoef(xcorr_1)
140.     Rsquared_1 = corrcoef_1**2
141.     xcorr_2 = np.stack((Y_2,output_2[:,0]))
142.     corrcoef_2 = np.corrcoef(xcorr_2)
143.     Rsquared_2 = corrcoef_2**2
144.     #
145.     # testianje
146.     N_hours = len(y) # broj sati u analizi
147.     y_calculated = np.zeros(shape = (N_hours,1))
148.     x1_min = min(x1)
149.     x1_max = max(x1)
150.     x2_min = min(x2)
151.     x2_max = max(x2)
152.     y_min = min(y)
153.     y_max = max(y)
154.     #
155.     for t in range(0,N_hours):
156.         #
157.         sig = 1/(1+np.exp(-ksig*(x2[t]-Tm)))
158.         #
159.
160.         x1_norm_1p = (x1[t]-min(x1_1))/(max(x1_1)-
   min(x1_1))
161.         x2_norm_1p = (x2[t]-min(x2_1))/(max(x2_1)-
   min(x2_1))
162.         x1_norm_2p = (x1[t]-min(x1_2))/(max(x1_2)-
   min(x1_2))
163.         x2_norm_2p = (x2[t]-min(x2_2))/(max(x2_2)-
   min(x2_2))
164.         #
165.         X_in_1 =
   np.transpose(np.stack((x1_norm_1p,x2_norm_1p)))  # <- ektor
   ulaznih velicina
166.         X_in_2 =
   np.transpose(np.stack((x1_norm_2p,x2_norm_2p)))
167.         #visoko temp. T >= Tm
168.         ytmp1_1 = np.dot(X_in_1,W01_1) + b1_1
169.         ytmp2_1 = nonlin(ytmp1_1)
170.         ytmp3_1 = np.dot(ytmp2_1,W12_1) + b2_1
171.         ytmp4_1 = nonlin(ytmp3_1)
172.         # nisko temp. T < Tm
173.         ytmp1_2 = np.dot(X_in_2,W01_2) + b1_2
174.         ytmp2_2 = nonlin(ytmp1_2)
175.         ytmp3_2 = np.dot(ytmp2_2,W12_2) + b2_2
176.         ytmp4_2 = nonlin(ytmp3_2)
177.         #
178.         ytmp4_11 = min_y1 + ytmp4_1*(max_y1-min_y1)
179.         ytmp4_22 = min_y2 + ytmp4_2*(max_y2-min_y2)
180.         #
181.         y_calculated[t] = sig*ytmp4_11 + (1-sig)*ytmp4_22
182.         #
183.     #
184.     # coefficent of correlation
185.     xcorr = np.stack((y,y_calculated[:,0]))
```

```
186.     corrcoef = np.corrcoef(xcorr)
187.     Rsquared1 = corrcoef**2
188.     # snimi u datoteku
189.     with open('m04_parametri_ANN_2Z.pkl', 'wb') as f:  #
   Python 3: open(..., 'wb')
190.         pickle.dump([x1_1, x2_1, x1_2, x2_2, min_y1,
   max_y1, min_y2, max_y2, Tm, ksig, x1_min, x1_max, x2_min,
   x2_max, W01_1, b1_1, W12_1, b2_1, W01_2, b1_2, W12_2, b2_2,
   y_min, y_max], f)
191.     #=================================================
   ========================
192.     # PLOT-ovi
193.     #=================================================
   ========================
194.     fig = plt.figure(1)
195.     plt.loglog(err_1)
196.     plt.loglog(err_2)
197.     fig.suptitle('Met4-ANN-2Z: Greška vs. iteracije')
198.     plt.xlabel('iteracije')
199.     plt.ylabel('greška (norma)')
200.     plt.grid(True,which="both",ls="-")
201.     plt.legend(['visokotemp.', 'niskotemp.'])
202.     plt.savefig('m04_ANN_greska.tiff')
203.     #plt.annotate(('error=' ,err_1[-1]),xy=(N_iters/2,
   (max(err)+min(err))/2), xytext=(N_iters/2,
   (max(err)+min(err))/2))
204.     #
205.     fig = plt.figure(2)
206.     plt.plot([0,1],[0,1],'r',linewidth=3.0)
207.     plt.scatter(Y_1,output_1,3,'r')
208.     plt.scatter(Y_2,output_2,3,'b')
209.     fig.suptitle('Korelacija Met4-ANN-2Z: ulazni set
   podataka')
210.     plt.xlabel('referentno Plinacro [-] /normirano')
211.     plt.ylabel('predviđeno [-] /normirano')
212.     plt.legend(['kor = 1','visokotemp.', 'niskotemp.'])
213.     plt.grid()
214.     strtmp = 'R^2 (VT) = ' + str(round(Rsquared_1[0,1],2))
215.     plt.annotate(strtmp,xy=(0.2, 0.05),xytext=(0.8, 0.2))
216.     strtmp = 'R^2 (NT) = ' + str(round(Rsquared_2[0,1],2))
217.     plt.annotate(strtmp,xy=(0.1, 0.05),xytext=(0.8, 0.1))
218.     plt.savefig('m04_korelacija_ulazni_set.tiff')
219.     #
220.     fig = plt.figure(3)
221.     plt.plot([0,4000],[0,4000],'r',linewidth=3.0)
222.     plt.scatter(y,y_calculated,5,x2)
223.     fig.suptitle('Korelacija trening: Met4-ANN-2Z')
224.     plt.xlabel('referentno Plinacro [MWh]')
225.     plt.ylabel('predviđeno [MWh]')
226.     plt.legend(['kor = 1','rješenje'])
227.     plt.grid()
228.     plt.colorbar()
229.     strtmp = 'R^2 = ' + str(round(Rsquared1[0,1],2))
230.     plt.annotate(strtmp,xy=(2000, 500),xytext=(4000, 500))
231.     plt.savefig('m04_korelacija.tiff')
232.     #
233.     #fig = plt.figure(4)
234.     #plt.plot(y)
235.     #plt.plot(y_calculated)
236.     #fig.suptitle('vremenska potrosnja plina')
237.     #plt.xlabel('vrijeme [h]')
```

```
    #plt.ylabel('potrosnja plina [MWh]')
    #plt.legend(['referentna','predvidena'])
    #plt.grid()
```

Kod za model umjetne neuronske mreže s 3 temperaturne zone

```
#
import numpy as np
import matplotlib.pyplot as plt
import pickle
#
# sigmoid funkcija
def nonlin(x,deriv=False):
    if (deriv==True):
            return x*(1-x)
        else:
            return 1/(1 + np.exp(-x))
     #
     Tm1 = 5  # C temp koja dijeli set podataka
     Tm2 = 15 # C
     ksig = 0.5
     # ANN function
     def metoda_ANN(x):
         X, Y, Niters, LR1, LR2, Nneurons = x
         #
         #pdb.set_trace()
         NinputsX = X.shape[1] # number of inputs
         NsetsX   = X.shape[0] # number of independent variables
         # sigmoid function is defined outside
         #
         # seed random numbers
         np.random.seed(1)
         err = np.zeros(shape = (int(Niters),1)) # <- ovdje se pohranjuje greška
         #
         # initialize weights
         W01 = 2*np.random.random((NinputsX,Nneurons)) - 1 # oko nule!
         W12 = 2*np.random.random((Nneurons,1)) - 1 # oko nule!
         b1  = 2*np.random.random((1,Nneurons)) - 1 # oko nule!
         b2  = 2*np.random.random((1,1)) - 1 # oko nule!
         #
         #
         for iter1 in range(int(Niters)):
             #print([iter/Niters])
             LR = LR1 + (LR2 - LR1)/5e4*iter1
             if iter1 > 1e4:
                 LR = LR2
             # forward propagation
             z1 = np.dot(X,W01) + b1
             g1 = nonlin(z1)
             z2 = np.dot(g1,W12) + b2
             g2 = nonlin(z2)      # main solution
```

```
47.             #
48.             # error W2
49.             err_z2 = Y.T - g2.T # main error
50.             #
51.             # backpropagation
52.             # output layer -> hidden layer
53.             der_z2 = nonlin(g2,True) # <- derivacija
   aktivacijske funkcije po z2
54.             d_z2   = np.multiply(err_z2.T,der_z2)
55.             d_b2   = d_z2
56.             #pdb.set_trace()
57.             #
58.             # hidden layer -> input layer
59.             err_z1 = np.dot(d_z2,W12.T)
60.             d_z1   = err_z1*nonlin(g1,True)
61.             d_b1   = d_z1
62.             #pdb.set_trace()
63.             # update weights
64.             W12 = W12 + LR*np.dot(g1.T,d_z2)
65.             b2  = b2  + LR*np.dot(d_b2.T,np.ones(shape =
   (NsetsX,1)))
66.             W01 = W01 + LR*np.dot(X.T,d_z1)
67.             b1  = b1  + LR*np.dot(d_b1.T,np.ones(shape =
   (NsetsX,1))).T
68.             #pdb.set_trace()
69.             err[iter1] = np.linalg.norm(err_z2) # <- L-2
   vektorska norma
70.             errplot = err[iter1]
71.             print([iter1,errplot])
72.             #
73.         #print([g2])
74.         return W01,W12,b1,b2,err,g2
75.     ##
76.     # INPUTS & CALL FUNCTION
77.     #
78.     y_1 = []
79.     x1_1 = []
80.     x2_1 = []
81.     y_2 = []
82.     x1_2 = []
83.     x2_2 = []
84.     y_3 = []
85.     x1_3 = []
86.     x2_3 = []
87.     #
88.     i_test = 1 # <- test flag // staviti ==0 ako se želi
   testiati
89.     if i_test == 1:
90.         # data
91.         y  = input_to_ANN[:,0] # gas conspt
92.         x1 = input_to_ANN[:,1] # electricity conspt
93.         x2 = input_to_ANN[:,2] # temp
94.         #
95.         NN = y.shape[0]
96.         for i in range(0,NN):
97.             # normalization of inputs between 0 and 1
98.             if (x2[i] >= Tm2):
99.                 y_1  = np.append(y_1,y[i])
100.                x1_1 = np.append(x1_1,x1[i])
101.                x2_1 = np.append(x2_1,x2[i])
102.            elif x2[i] > Tm1 and x2[i] <= Tm2 : # T2 > T >
```

```
      T1 -> srednjetemperaturni dio
103.                 y_2  = np.append(y_2,y[i])
104.                 x1_2 = np.append(x1_2,x1[i])
105.                 x2_2 = np.append(x2_2,x2[i])
106.             elif (x2[i] < Tm2):
107.                 y_3  = np.append(y_3,y[i])
108.                 x1_3 = np.append(x1_3,x1[i])
109.                 x2_3 = np.append(x2_3,x2[i])
110.         #
111.         min_y1 = min(y_1)
112.         max_y1 = max(y_1)
113.         min_y2 = min(y_2)
114.         max_y2 = max(y_2)
115.         min_y3 = min(y_3)
116.         max_y3 = max(y_3)
117.         #
118.         y_norm_1  = (y_1-min_y1)/(max_y1-min_y1)
119.         x1_norm_1 = (x1_1-min(x1_1))/(max(x1_1)-min(x1_1))
120.         x2_norm_1 = (x2_1-min(x2_1))/(max(x2_1)-min(x2_1))
121.         X_1 =
    np.transpose(np.stack((x1_norm_1,x2_norm_1)))
122.         Y_1 = y_norm_1
123.         #
124.         y_norm_2  = (y_2-min_y2)/(max_y2-min_y2)
125.         x1_norm_2 = (x1_2-min(x1_2))/(max(x1_2)-min(x1_2))
126.         x2_norm_2 = (x2_2-min(x2_2))/(max(x2_2)-min(x2_2))
127.         X_2 =
    np.transpose(np.stack((x1_norm_2,x2_norm_2)))
128.         Y_2 = y_norm_2
129.         #
130.         y_norm_3  = (y_3-min_y3)/(max_y3-min_y3)
131.         x1_norm_3 = (x1_3-min(x1_3))/(max(x1_3)-min(x1_3))
132.         x2_norm_3 = (x2_3-min(x2_3))/(max(x2_3)-min(x2_3))
133.         X_3 =
    np.transpose(np.stack((x1_norm_3,x2_norm_3)))
134.         Y_3 = y_norm_3
135.     else:
136.         X = np.array([[0,0,1],[0,1,1],[1,0,1],[1,1,1]])
137.         Y = np.array([[0],[1],[1],[0]])
138.     #
139.     # ANN parameters
140.     N_iters   = 1e4
141.     N_neurons = 10
142.     LearningRateStart  = 0.0002
143.     LearningRateFinish = 0.001
144.     #
145.     # define inputs to function
146.     xInput_1    = (X_1, Y_1, int(N_iters),
    LearningRateStart, LearningRateFinish, N_neurons)
147.     xInput_2    = (X_2, Y_2, int(N_iters),
    LearningRateStart, LearningRateFinish, N_neurons)
148.     xInput_3    = (X_3, Y_3, int(N_iters),
    LearningRateStart, LearningRateFinish, N_neurons)
149.     #
150.     # call function
151.     [W01_1,W12_1,b1_1,b2_1,err_1,output_1] =
    metoda_ANN(xInput_1)
152.     [W01_2,W12_2,b1_2,b2_2,err_2,output_2] =
    metoda_ANN(xInput_2)
153.     [W01_3,W12_3,b1_3,b2_3,err_3,output_3] =
    metoda_ANN(xInput_3)
```

```
154.     #
155.     # coefficent of correlation
156.     xcorr_1 = np.stack((Y_1,output_1[:,0]))
157.     corrcoef_1 = np.corrcoef(xcorr_1)
158.     Rsquared_1 = corrcoef_1**2
159.     xcorr_2 = np.stack((Y_2,output_2[:,0]))
160.     corrcoef_2 = np.corrcoef(xcorr_2)
161.     Rsquared_2 = corrcoef_2**2
162.     xcorr_3 = np.stack((Y_3,output_3[:,0]))
163.     corrcoef_3 = np.corrcoef(xcorr_3)
164.     Rsquared_3 = corrcoef_3**2
165.     #
166.     # testianje
167.     N_hours = len(y) # broj sati u analizi
168.     y_calculated = np.zeros(shape = (N_hours,1))
169.     x1_min = min(x1)
170.     x1_max = max(x1)
171.     x2_min = min(x2)
172.     x2_max = max(x2)
173.     y_min = min(y)
174.     y_max = max(y)
175.     #
176.     Tmid = (Tm1 + Tm2)*0.5
177.     for t in range(0,N_hours):
178.         #
179.         sig2 = 1/(1+np.exp(-ksig*(x2[t]-Tm2)))
180.         sig1 = 1/(1+np.exp(-ksig*(x2[t]-Tm1)))
181.         #
182.         x1_norm_1p = (x1[t]-min(x1_1))/(max(x1_1)-
    min(x1_1))
183.         x2_norm_1p = (x2[t]-min(x2_1))/(max(x2_1)-
    min(x2_1))
184.         x1_norm_2p = (x1[t]-min(x1_2))/(max(x1_2)-
    min(x1_2))
185.         x2_norm_2p = (x2[t]-min(x2_2))/(max(x2_2)-
    min(x2_2))
186.         x1_norm_3p = (x1[t]-min(x1_3))/(max(x1_3)-
    min(x1_3))
187.         x2_norm_3p = (x2[t]-min(x2_3))/(max(x2_3)-
    min(x2_3))
188.         #
189.         X_in_1 =
    np.transpose(np.stack((x1_norm_1p,x2_norm_1p)))  # <- ektor
    ulaznih velicina
190.         X_in_2 =
    np.transpose(np.stack((x1_norm_2p,x2_norm_2p)))
191.         X_in_3 =
    np.transpose(np.stack((x1_norm_3p,x2_norm_3p)))
192.         #visoko temp. T >= Tm
193.         ytmp1_1 = np.dot(X_in_1,W01_1) + b1_1
194.         ytmp2_1 = nonlin(ytmp1_1)
195.         ytmp3_1 = np.dot(ytmp2_1,W12_1) + b2_1
196.         ytmp4_1 = nonlin(ytmp3_1)
197.         # nisko temp. T < Tm
198.         ytmp1_2 = np.dot(X_in_2,W01_2) + b1_2
199.         ytmp2_2 = nonlin(ytmp1_2)
200.         ytmp3_2 = np.dot(ytmp2_2,W12_2) + b2_2
201.         ytmp4_2 = nonlin(ytmp3_2)
202.         #
203.         ytmp1_3 = np.dot(X_in_3,W01_3) + b1_3
204.         ytmp2_3 = nonlin(ytmp1_3)
```

```
205.          ytmp3_3 = np.dot(ytmp2_3,W12_3) + b2_3
206.          ytmp4_3 = nonlin(ytmp3_3)
207.          #
208.          ytmp4_11 = min_y1 + ytmp4_1*(max_y1-min_y1)
209.          ytmp4_22 = min_y2 + ytmp4_2*(max_y2-min_y2)
210.          ytmp4_33 = min_y3 + ytmp4_3*(max_y3-min_y3)
211.          #
212.          if x2[t] >= Tmid:
213.              y_calculated[t] = sig2*ytmp4_11 + (1-
    sig2)*ytmp4_22# <- potrošnja plina koja je izračunata
214.          elif x2[t] < Tmid:
215.              y_calculated[t] = sig1*ytmp4_22 + (1-
    sig1)*ytmp4_33 # <- potrošnja plina koja je izračunata
216.          #
217.      #
218.      # coefficent of correlation
219.      xcorr = np.stack((y,y_calculated[:,0]))
220.      corrcoef = np.corrcoef(xcorr)
221.      Rsquared1 = corrcoef**2
222.      # snimi u datoteku
223.      with open('m05_parametri_ANN_3Z.pkl', 'wb') as f:  #
    Python 3: open(..., 'wb')
224.          pickle.dump([x1_1, x2_1, x1_2, x2_2, x1_3, x2_3,
    min_y1, max_y1, min_y2, max_y2, min_y3, max_y3, Tm1, Tm2,
    ksig, x1_min, x1_max, x2_min, x2_max, W01_1, b1_1, W12_1,
    b2_1, W01_2, b1_2, W12_2, b2_2, W01_3, b1_3, W12_3, b2_3,
    y_min, y_max], f)
225.      #=====================================================
    ========================
226.      # PLOT-ovi
227.      #=====================================================
    ========================
228.      fig = plt.figure(1)
229.      plt.loglog(err_1)
230.      plt.loglog(err_2)
231.      plt.loglog(err_3)
232.      fig.suptitle('Met5-ANN-3Z: Greška vs. iteracije')
233.      plt.xlabel('iteracije')
234.      plt.ylabel('greška (norma)')
235.      plt.grid(True,which="both",ls="-")
236.      plt.legend(['visokotemp.', 'srednjetemp.',
    'niskotemp.'])
237.      plt.savefig('m05_ANN_greska.tiff')
238.      #plt.annotate(('error=' ,err_1[-1]),xy=(N_iters/2,
    (max(err)+min(err))/2), xytext=(N_iters/2,
    (max(err)+min(err))/2))
239.      #
240.      fig = plt.figure(2)
241.      plt.plot([0,1],[0,1],'r',linewidth=3.0)
242.      plt.scatter(Y_1,output_1,3,'r')
243.      plt.scatter(Y_2,output_2,3,'g')
244.      plt.scatter(Y_3,output_3,3,'b')
245.      fig.suptitle('Korelacija Met5-ANN-3Z: ulazni set
    podataka')
246.      plt.xlabel('referentno Plinacro [-] /normirano')
247.      plt.ylabel('predviđeno [-] /normirano')
248.      plt.legend(['kor = 1','visokotemp.', 'srednjetemp.',
    'niskotemp.'])
249.      plt.grid()
250.      strtmp = 'R^2 (VT) = ' + str(round(Rsquared_1[0,1],2))
251.      plt.annotate(strtmp,xy=(0.2, 0.05),xytext=(0.75,
```

```
      0.25))
252.      strtmp = 'R^2 (ST) = ' + str(round(Rsquared_3[0,1],2))
253.      plt.annotate(strtmp,xy=(0.1, 0.05),xytext=(0.75,
      0.15))
254.      strtmp = 'R^2 (NT) = ' + str(round(Rsquared_3[0,1],2))
255.      plt.annotate(strtmp,xy=(0.0, 0.05),xytext=(0.75,
      0.05))
256.      plt.savefig('m05_korelacija_ulazni_set.tiff')
257.      #
258.      fig = plt.figure(3)
259.      plt.plot([0,4000],[0,4000],'r',linewidth=3.0)
260.      plt.scatter(y,y_calculated,5,x2)
261.      fig.suptitle('Korelacija trening: Met5-ANN-3Z')
262.      plt.xlabel('referentno Plinacro [MWh]')
263.      plt.ylabel('predviđeno [MWh]')
264.      plt.legend(['kor = 1','rješenje'])
265.      plt.grid()
266.      plt.colorbar()
267.      strtmp = 'R^2 = ' + str(round(Rsquared1[0,1],2))
268.      plt.annotate(strtmp,xy=(2000, 500),xytext=(4000, 500))
269.      plt.savefig('m05_korelacija.tiff')
270.      #
271.      #fig = plt.figure(4)
272.      #plt.plot(y)
273.      #plt.plot(y_calculated)
274.      #fig.suptitle('vremenska potrosnja plina')
275.      #plt.xlabel('vrijeme [h]')
276.      #plt.ylabel('potrosnja plina [MWh]')
277.      #plt.legend(['referentna','predvidena'])
278.      #plt.grid()
```

Kod za model temperaturne korelacije

```
import numpy as np
import matplotlib.pyplot as plt
import pickle
#
a = input_to_ANN[:,0] # gas conspt
b = input_to_ANN[:,1] # electricity conspt
c = input_to_ANN[:,2] # temp
#
res1st = np.polyfit(c,a,1) # prvog stupnja
res2st = np.polyfit(c,a,2) # drugog stupnja
#
N   = len(a)
e_G_1st = np.zeros((N,1)) # <- inicijalizacija (=0) vektora koji
sadrzi rjesenje za input podatke
e_G_2st = np.zeros((N,1)) # <- inicijalizacija (=0) vektora koji
sadrzi rjesenje za input podatke
#
for i in range(0,N):
    e_G_1st[i] = res1st[0]*c[i] + res1st[1]
    e_G_2st[i] = res2st[0]*c[i]**2 + res2st[1]*c[i] + res2st[2]
#
# coefficent of correlation
x = np.stack((a,e_G_2st[:,0]))
```

```
corrcoef = np.corrcoef(x)
Rsquared = corrcoef**2
# snimi u datoteku
with open('m06_LinReg_T.pkl', 'wb') as f:  # Python 3: open(..., 
'wb')
    pickle.dump([res1st, res2st], f)
#==========
# PLOT
#=========
fig = plt.figure(1)
plt.plot([0,4000],[0,4000],'r',linewidth=3.0)
plt.scatter(a,e_G_2st,3,c)
fig.suptitle('Korelacija trening: Met6-PolyT')
plt.xlabel('referentno Plinacro [MWh]')
plt.ylabel('predviđeno eG [MWh]')
plt.grid()
plt.colorbar()
plt.legend(['kor = 1','rješenje'])
strtmp = 'R^2 = ' + str(round(Rsquared[0,1],2))
plt.annotate(strtmp,xy=(2000, 500),xytext=(4000, 500))
plt.savefig('m06_korelacija.tiff')
```

IZJAVA

Izjavljujem da sam ovaj rad s naslovom „Modeli za procjenu potrošnje prirodnog plina“ izradio samostalno uz pomoć i savjete mentora te na temelju znanja stečenih na Rudarsko-geološko-naftnom fakultetu služeći se navedenom literaturom.

_____

Mislav Matić