Predviđanje brzine vjetra

Ivan Avirović
Prirodoslovno-matematički fakultet
Sveučilište u Zagrebu
Zagreb, Hrvatska
ivan.avirovic@student.math.hr

Antonio Đurić
Prirodoslovno-matematički fakultet
Sveučilište u Zagrebu
Zagreb, Hrvatska
antonio.duric@student.math.hr

Mario Marjanović
Prirodoslovno-matematički fakultet
Sveučilište u Zagrebu
Zagreb, Hrvatska
mario.marjanovic@student.math.hr

***Abstract*—U ovom dokumentu se opisuje naš pristup rješavanju problema predviđanja brzine vjetra. Prije same izradnje modela za predviđanje brzine vjetra isprobali smo razne načine data imputationa. Također, prilikom izgradnje modela isprobali smo koristiti atribut datuma odnosno ne koristiti ga u drugim modelima. Očekivano, modeli koji korsite atribut datum su se ispostavili boljima. Konačno, najgori model koji smo izgradili je imao apsolutnu grešku u iznosu od 3.64 km/h.**

***Index Terms*—strojno učenje, data imputation, brzina vjetra**

I. Uvod

Cilj ovog projekta je predvidjeti brzinu vjetra pomoću dobivenih atributa iz metereoloških senzora. Dani atributi su dnevne prosječne oborine, maksimalna i minimalna dnevna temperatura, minimalna temperatura trave te nekoliko indikatora. Precizno predviđanje brzine vjetra može biti od iznimne ekonomske važnosti (npr. agrikultura, vjetroelektrane) te može pomoći u sprječavanju prirodnih katastrofa.

Prije same izrade modela isprobali smo razne metode data imputationa te smo odlučili pristupiti problemu na dva načina. Na početku zanemarujemo da se radi o vremenskom nizu te ignoriramo atribut datuma. Naknadno, uzimamo u obzir datum te uspoređujemo dobivene razultate. Naravno, očekujemo da će model koji koristi datum biti precizniji.

II. Opis problema

A. Skup podataka

Skup podataka koje koristimo smo preuzeli sa stranice Kagglea. Skup podataka se sastoji od 6574 dnevnih mjerenja, od 1.1.1961. do 31.12.1978., koristeći 5 meteoroloških senzora ugrađenih u meteorološkoj stanici. Uređaj koji je prikupljao podatke se nalazio na značajno praznom području. Prilikom analize podataka primijetili smo kako neki podatci nedostaju, što možemo vidjeti iz Fig. 1.

Također, na prikazanom heatmapu Fig. 2. i dendrogramu Fig. 3. možemo vidjeti da kada nedostaje maksimalna dnevna temperatura nedostaje i minimalna dnevna temperatura. Također je isti slučaju i s indikatorom 1 i indikatorom 2. Zbog značajnog nedostataka određenih navedenih atributa koristimo data imputation.

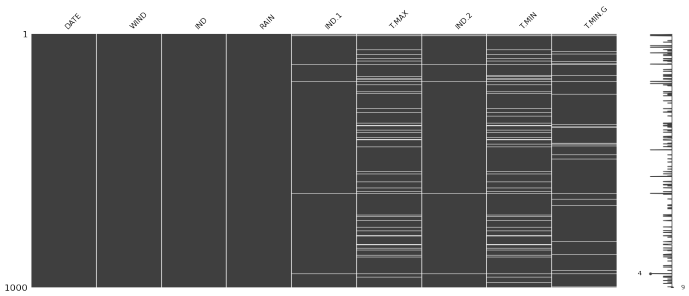

Fig. 1. Nedostatak određenih atributa

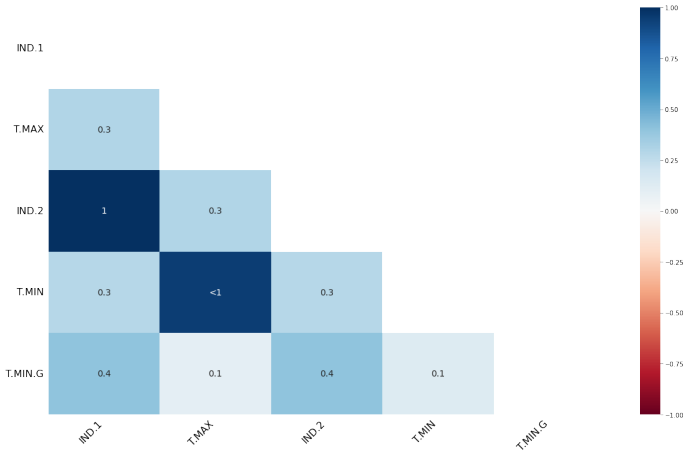

Fig. 2. Heatmap

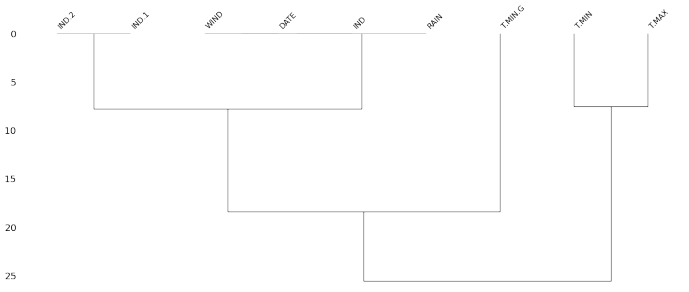

Fig. 3. Dendrogram

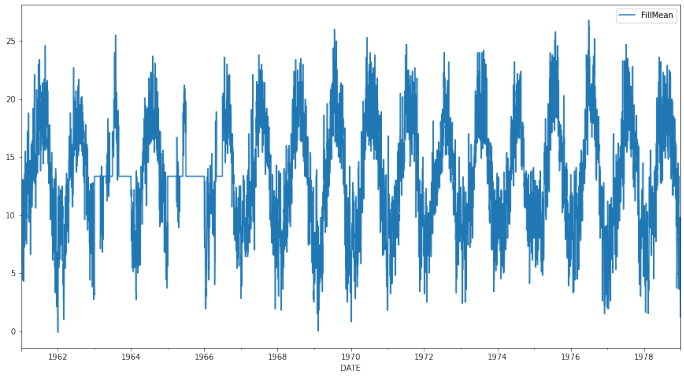

Fig. 4. Mean

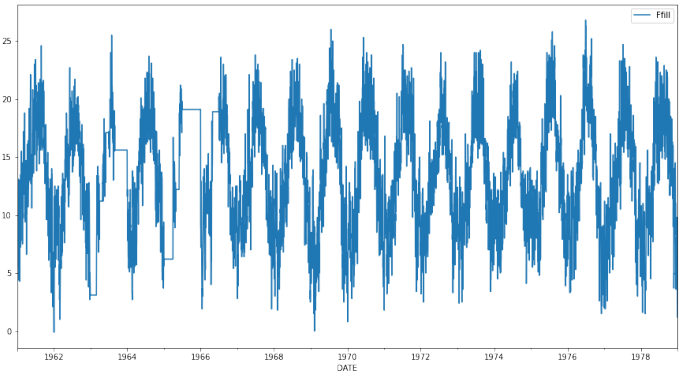

Fig. 5. FFil

B. Data imputation

Zbog navedenog nedostatka određenih podataka odlučili smo isprobati razne metode data imputationa. Data imputation je proces nadopunjivanja nedostajajućih podataka s nekim podacima. Metode data imputation-a se razlikuju upravo u načinu odabira vrijednosti podataka koji nedostaju.

Neke od metoda koje smo isprobali su:

- Zamjena nedostajajućih podataka aritmetičkom sredinom podataka
- Zamjena nedostajajućih podataka medijanom podataka
- Zamjena nedostajajućih podataka prethodnom ispravnom vrijednosti (FFill)
- Zamjena nedostajajućih podataka budućom ispravnom vrijednosti (BFill)
- Zamjena nedostajajućih podataka interpolacijom, a neke od tih interpolacija su
  - linearna interpolacija
  - kubna interpolacija
  - akima interpolacija
  - interpolacija polinomom višeg stupnja
  - interpolacija splajnom

Na slikama gore možemo vidjeti popunjavanje podataka maksimalne tempreature zraka metodom zamjene aritmetičkom sredinom 4, Ffill metodom 5, linearnom interpolacijom 6 te interpolacijom splajnom trećeg reda 7.

Od navedenih metoda odlučili smo koristiti metodu zamjene nedostajajućih podataka linearnom interpolacijom za neprekidne vrijednosti (maksimalna i minimalna temperatur zraka i minimalna temperatura trave) te metodu zamjene nedostajajućih podataka najčešćim elementom za diskretne vrijednosti (indikatori 1 i 2).

Ipak, kako ne bi koristili samo već unaprijed definirane i jednostavnije metode u pythonu za dopunu neodstajajućih podataka, odlučili smo napraviti drugi dataset u kojem korisitmo naprednije metode.

Kao i u prvom datasetu razlikujemo metode zamjene podataka za neprekidne varijable te za diskretne varijable. Za zamjenu nedostajajućih neprekidnih podataka odlučili

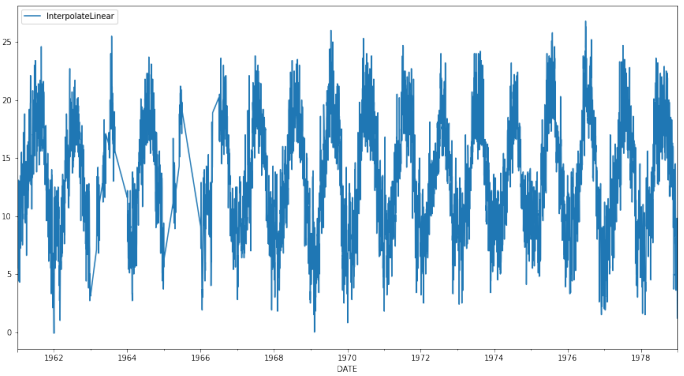

Fig. 6. Linearna interpolacija

smo koristiti linearnu regresiju, a za zamjenu nedostajajućih diskretnih podataka odlučili smo koristiti algoritam k-najbližih susjeda.

Na sljedećoj stranici možemo vidjeti način dopunjavanja na primjeru minimalne temperature zraka Fig 8 i 9 te indikatora 2 Fig 10 i 11. Možemo vidjeti da koristeći linearnu regresiju za razliku od prijašnjih metoda zamjene dobivamo na pogled ispravniju dopunu podataka te stoga očekujemo i bolje rezultate koristeći drugi dataset.

Također, napomenimo kako smo napravili skaliranje

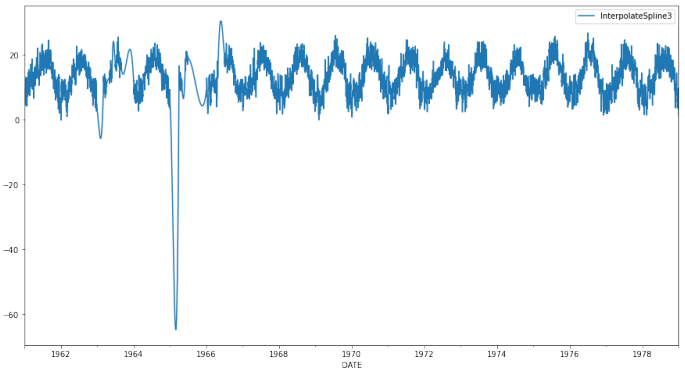

Fig. 7. Interpolacija splajnom

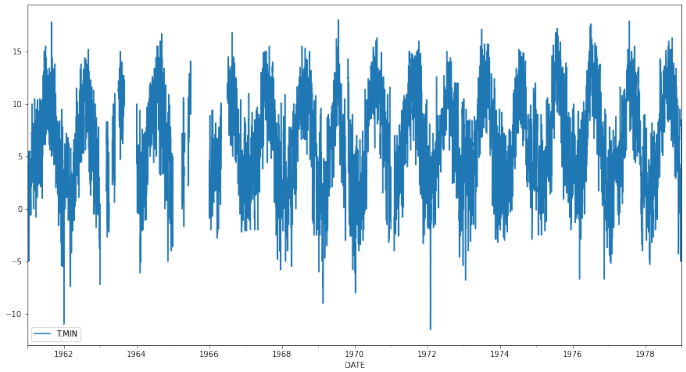

Fig. 8. Minimalna tmperatura zraka prije data imputation-a

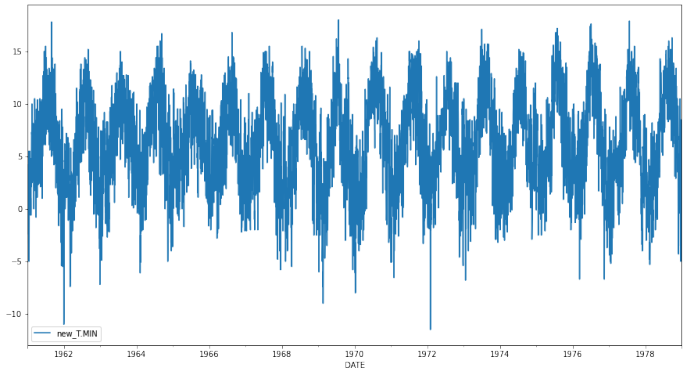

Fig. 9. Minimalna tmperatura zraka poslije data imputation-a

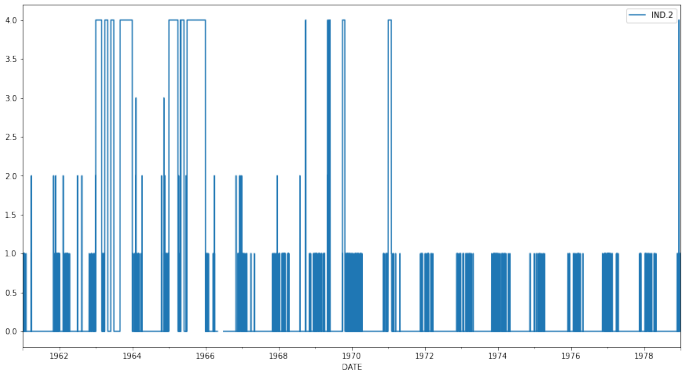

Fig. 10. Indikator 2 prije data imputation-a

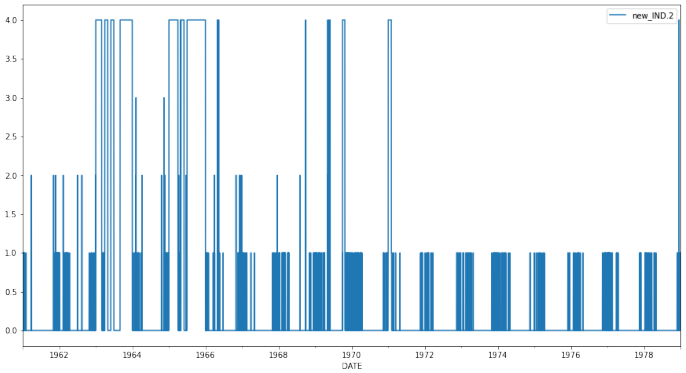

Fig. 11. Indikator 2 poslije data imputation-a

podataka zbog velike razlike u iznosu pojedinih atributa te one hot encoding za atribute indikator 1 i indikator 2.

III. Opis metoda i pristupa za rješavanje problema

Nakon data imputation-a, opisanog pravljenja dva dataseta te data scaling-a i one hot encoding-a, kao što smo opisali u projektnog predlošku, promatrali smo dane podatke bez datuma i s datumom, očekujući naravno bolje rezultate koristeći datum kao jedan od atributa.

Skup podataka smo podijelili na train (70%) i test (30%) podatke, ali tako da smo uzeli prvih 70% i zadnjih 30% kako bismo mogli uspoređivat rezultate naša dva pristupa. Modeli koje smo napravili ignorirajući atribut datuma su

- linearna regresija
- SVM-regresija
- XGBoost

Modeli koje smo napravili koristeći atribut datum su

- XGBoost
- VAR model
- neuronske mreže

Ponovno naglašavamo kako smo sve modele pravili na dva različita dataseta čiju smo izgradnju opisali u prethodnoj sekciji. Svi modeli su napravljeni koristeći Python u obliku Jupyter bilježnica najviše koristeći biblioteke sklearn [5] i matplotlib [6].

A. *Stationarity, causality i cointegration testovi*

Također, prije same izrade time series modela napravili smo test stacionarnosti, kako bi bili sigurni da će naši modeli dobro raditi. Test stacionarnosti koji smo radili je Augmented Dickey–Fuller test te smo njime potvrdili stacionarnost naših podataka za oba dataseta.

Za test kauzalnosti koristili smo Grangerov test kauzalnosti kojim provjeravamo možemo li jednim vremenskim slijedom (time series) predvidjeti drugi.

Konačno, testom kointegracije potvrdili smo da dugoročno postoji korelacija između nekoliko vremenskih slijedova.

IV. Rezultati

Kao što smo već napomenuli, sve modele smo trenirali na početnih 70% podataka, a testirali na završnih 30% podataka. Kod metoda koje ne koriste datum smo još napravili i K-struku unakrsnu validaciju kako bismo potvrdili rezultate, no modele smo međusobno uspoređivali na temelju rezultata na završnih 30% podataka. Koristili smo prosječnu apsolutnu grešku za metriku. Računali smo je na skaliranim podacima jer nam je bitno jedino da možemo uspoređivati modele međusobno, a i radi jednostavnosti.

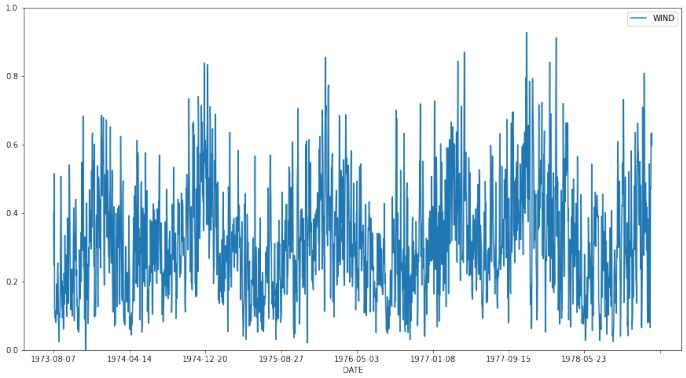

Fig. 12. Test podaci za brzinu vjetra

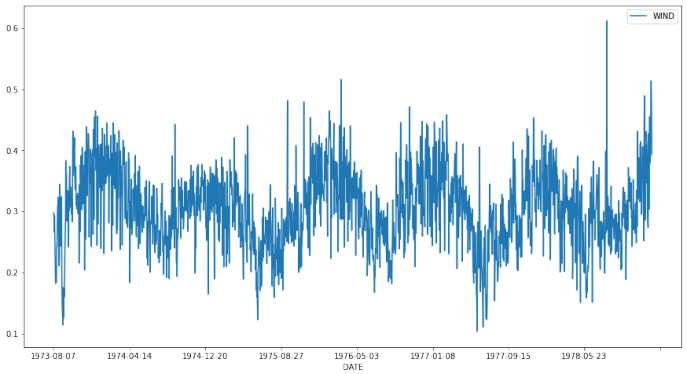

Fig. 13. Predviđanje linearne regresije za prvi dataset

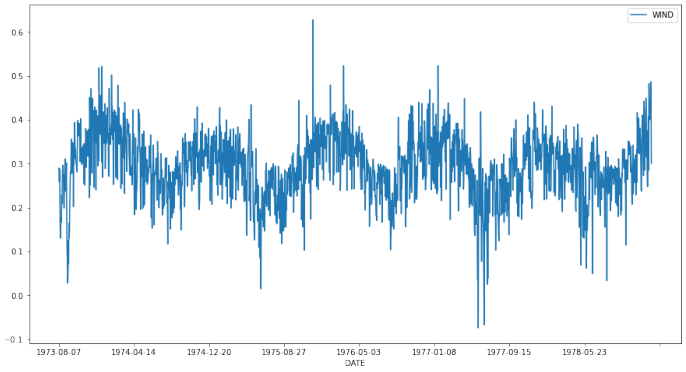

Fig. 14. Predviđanje SVM-regresije za drugi dataset

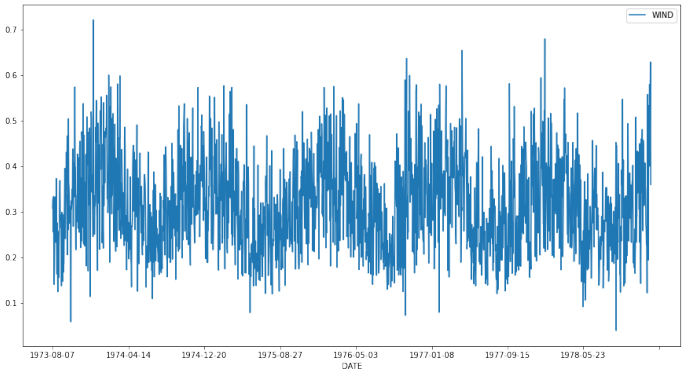

Fig. 15. XGBoost regresija predviđanja za prvi dataset

A. *Linearna regresija*

Model linearne regresije koristimo kao polazni model preko kojeg možemo uspoređivati uspješnost ostalih, naprednijih modela. Za linearan model najbolje rezultate dobili smo korištenjem defaultnih parametara. Za oba dataseta model linearne regresije nam daje prosječnu apsolutnu grešku (MAE) 0.118.

B. *SVM-regresija*

Pri radu s modelom SVM-regresija potrebno je bilo postaviti kernel te parametre epsilon, C i gamma. Gdje epsilon određuje epsiloninsensitive tube SVM modela, C je parametar regularizacije, a gamma je koeficijent kernela. Najbolje rezultate dobili smo korištenjem radial basic function (RBF) kernela i postavljanjem parametara epsilon na vrijednost 0.07, parametra C na vrijednost 3 i parametar gamma na defaultnu vrijednost scale. Za te parametre dobijemo model koji za prvi i drugi dataset ima prosječnu apsolutnu grešku 0.116, malo bolju nego za model linearne regresije.

C. *XGBoost*

Za problem regresije koristili smo i gradient boosting framework XGBoost, mijenjanjem defaultnih vrijednosti parametara nije došlo do značajne promjene u kvaliteti modela, jedino smo eta (learning rate) postavili na vrijednost 0.25. Ovaj model nam je za oba dataseta imao prosječnu apsolutnu grešku 0.12, što je lošije od prethodna dva modela.

XGBoost smo također primjenili i na time series problemu. Kako mi imamo više značajki koje promatramo kroz vrijeme ovdje se radi o multivarijantnom time series problemu. Kod rješavanja time-series problema dogovorili smo se da ćemo uzeti zadnjih 5 dana i pomoću njihovih podataka pokušati predvidjeti brzinu vjetra za sljedeći dan. Sve modele smo testirali tako da smo zadnjih 30% dana uzeli i od njih napravili pomoću metode pomicajućeg prozora napravili parove ulaza (svi podaci za 5 dana) i očekivanog izlaza (brzina vjetra 1 dan nakon). Za neke modele smo i podatke za treniranje preoblikovali na isti način.

XGBoost smo testirali sa raznim parametrima, ali su svi davali vrlo slične rezultate pa smo na kraju odlučili koristiti defaultne parametre. Na prvom datasetu smo dobili prosječnu apsolutnu grešku 0.103, a na drugom 0.102. Kao što smo i očekivali XGBoost na time series problemu daje bolje rezultate od regresijskog problema.

D. *Neuronske mreže*

Pošto radimo na time series problemu logično je bilo koristiti rekurentne neuronske mreže. Klasične rekurentne

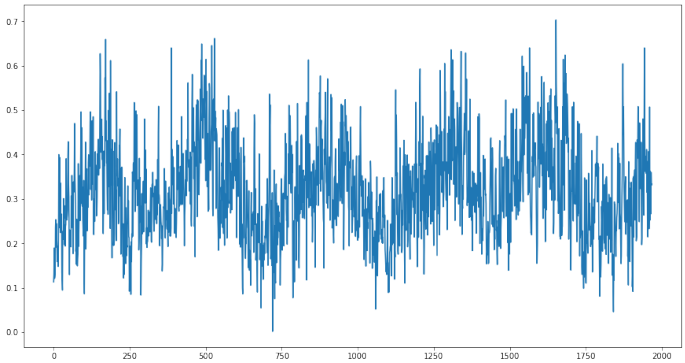

Fig. 16. XGBoost time series predviđenja za prvi dataset

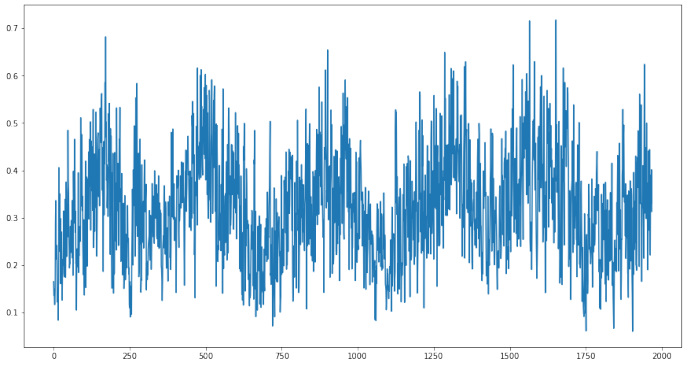

Fig. 17. XGBoost time series predviđenja za drugi dataset

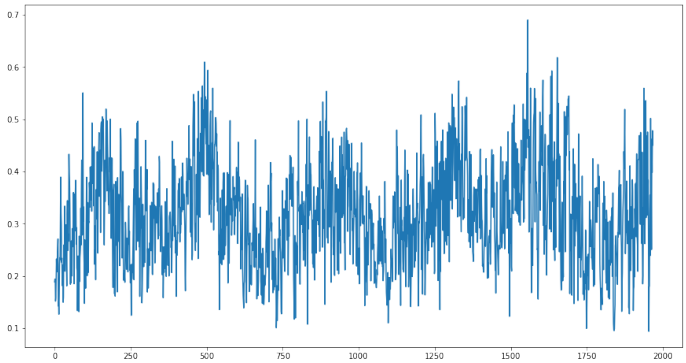

Fig. 18. Predviđanje LSTM mreže s jednim slojem za drugi dataset

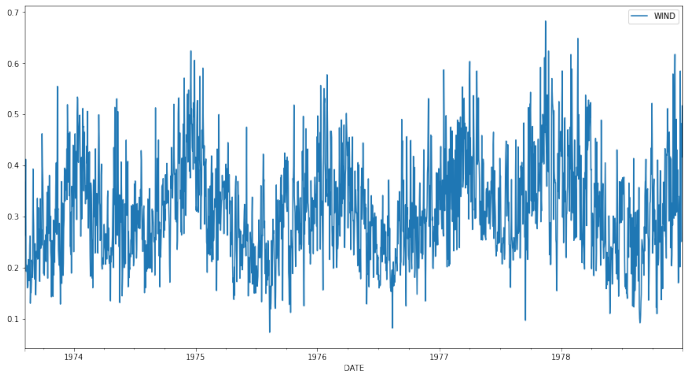

Fig. 19. Predviđanje VAR-a na prvom datasetu

neuronske mreže imaju problema sa dugotrajnim zavisnostima, zato ćemo većinom koristiti LSTM (long short term memory) mrežu koja obično daje bolje rezultate. Podaci su pripremljeni na isti način kao kod XGBoost time series problema. Kako bismo dobili što bolje rezultate isprobali smo razne oblike LSTM-ova. Jednostavna LSTM mreže s jednim slojem davala je rezultat 0.099 na prvom datasetu i 0.097 na drugom, s dva sloja rezultat je bio 0.097 na prvom datasetu i 0.099 na drugom, a s tri sloja 0.098 na prvom datasetu i 0.104 na drugom. Napravili smo obostranu LSTM mrežu gdje se ulazni podaci daju mreži u kronološkom i obratnom obliku, tu smo dobili rezultat 0.098 na prvom datasetu i 0.1 na drugom. Kako su konvolucijske neuronske mreže dobre u izdvajanju značajki, iskoristili smo konvolucijku neuronsku mrežu kako bi od dva podniza ulaznih podataka izvukla najbitnije značajke te ih proslijedila našoj LSTM mreži. Koristili smo jedan konvolucijski sloj kojeg slijedi max pooling sloj, te smo spljoštili te podatke i dali ih LSTM sloju. Ovakav pristup dao je rezultat 0.105 na prvom datasetu i 0.104 na drugom.

E. VAR

VAR(vektorska autoregresija) je već unaprijed prilagođena za korištenje na time series podacima, pa nismo morali raditi nikakve promjene na trening podacima. Kako bi mogli što objektivnije uspoređivat rezultate test podatke smo preoblikovali na isti način kao kod ostalih metoda koje smo koristili za time series problem. VAR je dao rezultat 0.098 na oba dataseta.

F. Zaključak

Najgori model nam ima prosječnu apsolutnu grešku od 0.12, jer su podaci skalirani zapravo imamo prosječnu apsolutnu grešku u iznosu od 3.64 km/h. Zaključujemo da su modeli koje smo koristili za problem predviđanja brzine vjetra relativno dobri.

Na prvom datasetu SVM-regresija se pokazuje kao najbolji među modelima koji ne uključuju datum, dok je

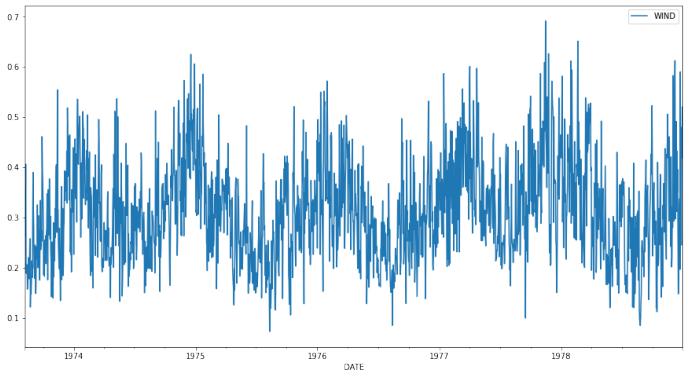

Fig. 20. Predviđanje VAR-a na drugom datasetu

LSTM s dva sloja najbolji među onima koji uključuju datum. LSTM s dva sloja je u prosjeku precizniji za 0.019 (0.58 km/h) od modela SVM-regresije. Na drugom datasetu dolazimo do identične razlike, ovaj put između modela SVM-regresija i LSTM s jednim slojem. Kao što smo i očekivali modeli koji uključuju atribut datum se pokazuju znatno bolji od onih koji ga ne uključuju.

Vidimo i da se nijedna metoda data imputationa nije pokazala značajno bolja od druge.

Model	MAE 1	MAE 2
Linearna regresija	0.118	0.118
SVM-regresija	0.116	0.116
XGBoost regresija	0.12	0.12
XGBoost time series	0.103	0.102
LSTM 1 sloj	0.099	0.097
LSTM 2 sloja	0.097	0.099
LSTM 3 sloja	0.098	0.104
Obostrani LSTM	0.097	0.1
CNN LSTM	0.105	0.104
VAR	0.098	0.098

V. Osvrt na druge pristupe

Postoji popriličan broj istraživanja koji se bavi predviđanjem brzine vjetra. [2] koristeći drugi skup podataka sličnim atributima objašnjava osnovne mogućnosti neuronskih mreža strojnih učenja. [3] također koristeći drugi skup podataka i višedimenzionalne konvolucijske neuronske mreže pokušava se predvidjeti brzina vjetra. [4] pokazuje kako se može iskoristiti napredak u dubokom učenju u primjeni na obnovljivim izvorima energije između ostalog i vjetra. Na Kaggleu na istom skupu podataka trenutno postoji značno sažetija analiza podataka osobe koja je iste te podatke objavila.

VI. Budući nastavak istraživanja

Kako bi daljnje poboljšali model mogli bismo promatrati ansambl metode gdje kombiniramo neke od naših modela zajedno. Isto tako mogli bismo zadati drugačije parametre za time series problem, kod nas smo riješavali problem da dobijemo 5 dana podataka i trebamo predvidjeti za sljedeći dan. To naravno ovisi o uporabi modela, ali mogli bismo promatrati veći vremenski raspon ulaznih i izlaznih podataka.

References

[1] Wind Speed Prediction Dataset, Kaggle, https://www.kaggle.com/datasets/fedesoriano/wind-speed-prediction-dataset

[2] https://www.codespeedy.com/wind-direction-and-speed-prediction-using-machine-learning-in-python/

[3] K. Trebing and S. Mehrkanoon, “Wind speed prediction using multidimensional convolutional neural networks,” arXiv preprint arXiv:2007.12567, 2020.

[4] Gu, C.; Li, H. Review on Deep Learning Research and Applications in Wind and Wave Energy. Energies 2022, 15, 1510. https://doi.org/10.3390/en15041510

[5] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[6] J. D. Hunter, ”Matplotlib: A 2D Graphics Environment”, Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007