Otkrivanje anomalija u vremenskim nizovima

Udovičić, Josipa

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1649

OTKRIVANJE ANOMALIJA U VREMENSKIM NIZOVIMA

Josipa Udovičić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1649

OTKRIVANJE ANOMALIJA U VREMENSKIM NIZOVIMA

Josipa Udovičić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1649

Pristupnica:	**Josipa Udovičić (0036541648)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	doc. dr. sc. Adrian Satja Kurdija

Zadatak: **Otkrivanje anomalija u vremenskim nizovima**

Opis zadatka:

Motivirati i opisati metode za otkrivanje anomalija u skupovima podataka s naglaskom na vremenske nizove. Oblikovati i programski ostvariti nekoliko modela za otkrivanje anomalija. Primijeniti i eksperimentalno usporediti ostvarene modele na nekoliko javno dostupnih skupova podataka o povijesnim vremenskim prilikama. Prikazati i objasniti dobivene rezultate te izvesti zaključke. Uz rad je potrebno predati i dokumentirati izvorni kod ostvarenih modela, korišteni skup podataka te navesti korištenu literaturu i primljenu pomoć.

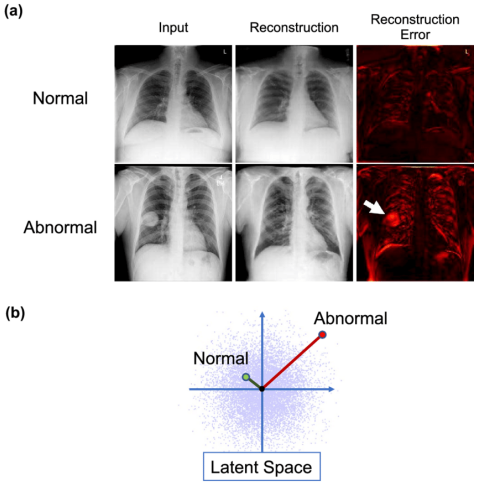
Rok za predaju rada: 14. lipnja 2024.

U radu mi je pomagao sa savjetima i uputama mentor završnog rada, doc. dr. sc. Adrian Satja Kurdija te mu iskreno zahvaljujem. Također, zahvaljujem svojim kolegama, te svojoj obitelji na podršci tijekom studiranja.

Sadržaj

1. Uvod

U današnjem digitalnom dobu, velike količine podataka generiraju se u kontinuiranim vremenskim serijama. Unutar ovih serija, čest je susret s pojavom anomalija. Anomalije su uzorci u podacima koji se ne usklađuju s očekivanim ponašanjem podataka. Dakle, otkrivanje anomalija se odnosi na pronalaženje uzoraka koji se razlikuju od ostatka podataka [1]. Primjena otkrivanja anomalija proteže se kroz različite domene, od medicinske dijagnostike i financijskih transakcija do industrijskih postrojenja. Na primjer, u medicinskoj dijagnostici, otkrivanje nepravilnosti na medicinskim slikama može omogućiti rano otkrivanje bolesti i poboljšati ishod liječenja, kako ilustrira primjer na slici 1.1.



Slika 1.1. Anomalije u medicini [2]

Kao što je naznačeno, jedno od područja primjene otkrivanja anomalija je analiza vremenskih nizova. Vremenski nizovi su vrste podataka koji se uzorkuju na temelju neke vrste dimenzije povezane s vremenom kao što su godine, mjeseci ili sekunde [3]. U ovom radu, fokus će biti na otkrivanju anomalija u vremenskim nizovima temperaturnih podataka te padalina. Prvo će biti opisana tri različita algoritma za otkrivanje anomalija, nakon toga će biti prikazani podaci korišteni u praktičnom dijelu ovog rada te kratki opis

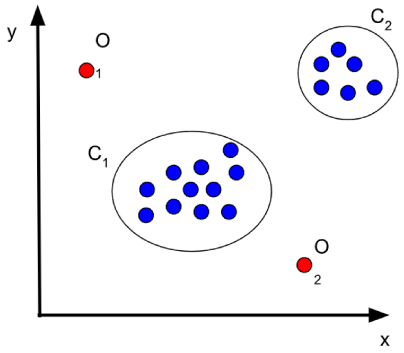
njihove strukture. Zatim, bit će prikazani grafovi koji prikazuju anomalije u temperaturnim podacima te padalinama. Grafovi su rađeni s pomoću *SeasonalAD* biblioteke koja će isto biti detaljnije objašnjena. Time se dolazi do kraja ovog rada i zaključka koji će sažeti sve navedene informacije.

Otkrivanje anomalija je važno jer anomalije u podacima pružaju značajne (i često kritične) informacije koje se mogu primijeniti u različitim područjima primjene [1]. Razumijevanje ovih anomalija omogućuje identifikaciju potencijalnih prijetnji, unapređenje sustava za upravljanje rizicima i optimizaciju procesa u različitim sektorima.

Rad je organiziran na sljedeći način. U poglavlju 2. bit će opisano što su anomalije i koje vrste postoje. Zatim će biti opisana motivacija za ovaj rad u potpoglavlju 2.1. te nakon toga slijedi potpoglavlje 2.2. u kojem će biti opisani algoritmi za otkrivanje anomalija korišteni u ovom radu. U poglavlju 3. bit će prikazan eksperimentalni dio ovog rada. To poglavlje se dijeli na potpoglavlja u kojima će biti opisani podaci korišteni u radu (potpoglavlje 3.1.) te nakon toga implementacija (potpoglavlje 3.2.). Na kraju tog poglavlja slijedi potpoglavlje 3.3. u kojem će biti prikazani i objašnjeni grafovi koji prikazuju anomalije u temperaturnim podacima te padalinama.

2. Općenito o anomalijama

Anomalije su uzorci u podacima koji se ne usklađuju s očekivanim ponašanjem podataka.



Slika 2.1. Anomalije u podacima [4]

Na slici 2.1. su crveno označene anomalije u podacima, a plavom normalni podaci. Anomalije odskaču od ostalih podataka i ne prate očekivano ponašanje podataka.

Postoje 3 vrste anomalija. Prve su točkaste anomalije koje se smatraju najjednostavnijim anomalijama. Primjer na slici 2.1. je primjer točkastih anomalija. Postoji skup podataka i jedna točka koja se ne uklapa u taj skup podataka [1]. Na primjer, ako pacijent koji obično ima stabilnu krvnu razinu šećera iznenada doživi nagli i neobjašnjivi skok ili pad razine šećera u krvi, to bi moglo ukazivati na točkastu anomaliju. Takav neočekivan događaj može signalizirati potencijalne zdravstvene probleme.

Druga vrsta su kontekstualne anomalije. Kontekstualne anomalije su anomalije koje se pojavljuju samo u određenom kontekstu. One su definirane pomoću dvije vrste atributa: kontekstualnih i ponašajnih. Kontekstualni atributi se koriste za određivanje konteksta ili susjedstva za određenu instancu podataka, dok ponašajni atributi definiraju

nekontekstualne karakteristike instance. Anomalno ponašanje se određuje korištenjem vrijednosti ponašajnih atributa unutar specifičnog konteksta [1]. Na primjer, temperatura od 3°C može biti normalna tijekom zime, ali ista vrijednost tijekom ljeta bi bila anomalija. Odabir primjene tehnike za otkrivanje kontekstualnih anomalija ovisi o značenju tih anomalija u ciljanoj domeni primjene i dostupnosti kontekstualnih atributa.

Zadnja vrsta su kolektivne anomalije. Kolektivne anomalije su anomalije koje se pojavljuju kada je skup povezanih instanci podataka anomalno u odnosu na cijeli skup podataka. Individualne instance podataka u kolektivnoj anomaliji same po sebi možda nisu anomalije, ali njihovo zajedničko pojavljivanje kao skup je anomalno [1]. Na primjer, u ljudskom elektrokardiogramu, određeno razdoblje s niskom vrijednošću može biti označeno kao anomalija ako traje neobično dugo, iako sama ta niska vrijednost nije anomalija.

Ovaj rad će se baviti kontekstualnim anomalijama u vremenskim nizovima.

2.1. Motivacija

Anomalije predstavljaju neočekivane i često značajne promjene ili događaje koji zahtijevaju posebnu pažnju i analizu. Bez obzira radi li se o financijskim transakcijama, medicinskim podacima, ili meteorološkim mjerenjima, otkrivanje anomalija je ključni korak u razumijevanju podataka i donošenju informiranih odluka.

Ovaj završni rad istražuje anomalije u vremenskim nizovima. Proučavajući različite metode otkrivanja, od klasičnih statističkih pristupa do suvremenih tehnika strojnog učenja, istražuje se kako različiti algoritmi mogu razotkriti skrivene uzorke i nepravilnosti u podacima.

Kroz primjere iz stvarnog svijeta, ovaj rad osvjetljava važnost otkrivanja anomalija u različitim domenama, ističući kako ove neočekivane pojave mogu imati velik utjecaj na poslovne procese, zdravstvene dijagnoze ili donošenje ključnih meteoroloških prognoza.

Ova istraživanja bi mogla potaknuti razmišljanje i daljnju analizu u području otkrivanja anomalija.

2.2. Algoritmi za otkrivanje anomalija

Anomalije je moguće otkriti raznim tehnikama kao što su tehnike otkrivanja anomalija na najbližem susjedu, tehnike otkrivanja anomalija temeljene na grupiranju, informacijsko-teorijske tehnike otkrivanja anomalija, tehnike otkrivanja spektralnih anomalija te statističke tehnike otkrivanja anomalija [1]. Kratak opis svake tehnike je naveden u nastavku.

Tehnike otkrivanja anomalija na najbližem susjedu otkrivaju anomalije na temelju udaljenosti između podataka. Normalni podaci imaju bliske susjede, dok su anomalije daleko od svojih najbližih susjeda. Anomalijski rezultat temelji se na prosječnoj udaljenosti do najbližih susjeda ili udaljenosti do k-tog najbližeg susjeda. Tehnike otkrivanja anomalija temeljene na grupiranju temelje se na pretpostavci da se normalni podaci skupljaju u gusto povezane grupe (klastere), dok se anomalije ne uklapaju ni u jedan klaster ili čine vrlo male, rijetke klastere. Informacijsko-teorijske tehnike otkrivanja anomalija koriste mjere informacijske teorije, poput entropije, za identifikaciju anomalija. Podaci s visokom entropijom ili neočekivanim informacijskim obrascima smatraju se anomalijama. Tehnike otkrivanja spektralnih anomalija koriste analizu frekvencijskih svojstava podataka. Podaci se transformiraju u frekvencijsku domenu kako bi se otkrili neuobičajeni obrasci. Anomalije se identificiraju kao podaci s neuobičajenim spektralnim svojstvima. Statističke tehnike otkrivanja anomalija koriste distribucijske modele podataka za identifikaciju anomalija. Podaci koji značajno odstupaju od očekivane statističke distribucije smatraju se anomalijama [1].

U ovom radu će se opisati i koristiti algoritmi koji spadaju u statističke tehnike otkrivanja anomalija. Algoritmi statističkih tehnika koriste Gaussov model za otkrivanje anomalija. Model promatra udaljenost podataka od procijenjene srednje vrijednosti i procjenjuje je kao anomaliju ako je udaljenost veća od određenog praga [1]. Algoritmi koji koriste opisani model računaju udaljenost od srednje vrijednosti i praga na različite načine. U ovom radu će biti pokazana jednostavna tehnika otkrivanja izvanrednih vrijednosti ($\mu \pm 3\sigma$), zatim *box plot* pravilo te na kraju Grubbsov test (z-score).

2.2.1. Algoritam $\mu \pm 3\sigma$

Algoritam $\mu \pm 3\sigma$ je jednostavan algoritam za otkrivanje anomalija kojem je u cilju deklarirati sve instance podataka koje su udaljene više od 3σ udaljenosti od μ. μ je srednja vrijednost distribucije, a σ je standardna devijacija za distribuciju.

1 algoritam: $\mu \pm 3\sigma$

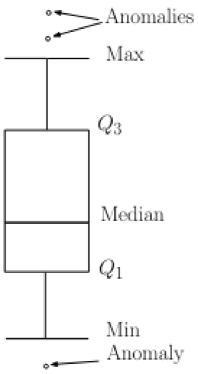
```
[1]: data = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1000]

[3]: import numpy as np

def calMiandSigma(data):
    mi = np.mean(data)
    sigma = np.std(data)
    return mi, sigma

def anomalies_f(data):
    mi, sigma = calMiandSigma(data)
    print(mi, sigma)
    lower = mi - 3 * sigma
    upper = mi + 3 * sigma
    print("lower limit: ",lower)
    print("upper limit: ",upper)

    anomalies = []
    for i in data:
        if i < lower or i > upper:
            anomalies.append(i)

    return anomalies

anomalies = anomalies_f(data)
print(anomalies)

95.9090909090909 285.9117646933799
lower limit:  -761.8262031710489
upper limit:  953.6443849892306
[1000]
```

Slika 2.2. Algoritam $\mu \pm 3\sigma$

Na slici 2.2. prikazan je jednostavan primjer algoritma $\mu \pm 3\sigma$. Izračuna se srednja vrijednost (μ) i standardna devijacija (σ) za distribuciju podataka. Zatim se pomoću izračunatih μ i σ odredi donja granica ($\mu - 3\sigma$) i gornja granica ($\mu + 3\sigma$). Sve instance podataka koje se nalaze izvan tih granica smatraju se anomalijama te se pomoću for petlje koja je vidljiva na slici 2.2. spremaju u listu i ispisuju. Na ovom jednostavnom primjeru se vidi kako algoritam $\mu \pm 3\sigma$ radi te će kao takav biti korišten kasnije u vremenskim nizovima. Također, $\mu \pm 3\sigma$ regija sadržava 997% opažanja [1].

2.2.2. *Box plot* pravilo

Sljedeći algoritam koji će biti prikazan je *box plot* pravilo. *Box plot* pravilo grafički prikazuje podatke koristeći sažete atribute kao što su: najmanja opažanja bez anomalije (min), donji kvartil ($Q1$), medijan, gornji kvartil ($Q3$) i najveće opažanje bez anomalije (max) kao što se vidi na slici 2.3.



Slika 2.3. *Box plot* pravilo [1]

Na slici 2.3. je vidljivo da je opet riječ o nekom rasponu i granicama te sve izvan tih granica se smatra anomalijom. Sada slijedi opis kako se *box plot* pravilo koristi za otkrivanje anomalija.

2 algoritam: The box plot rule - kvartili

```
[3]: dataset = [2, 4, 6, 8, 10, 100, 1000]

[4]: import numpy as np

def boxPlotRule(data):
    q1 = np.percentile(data, 25)
    median = np.percentile(data, 50)
    q3 = np.percentile(data, 75)

    iqr = q3 - q1

    lower = q1 - 1.5 * iqr
    upper = q3 + 1.5 * iqr
    print("lower limit: ",lower)
    print("upper limit: ",upper)

    anomalies = []
    for i in data:
        if i < lower or i > upper:
            anomalies.append(i)

    return anomalies

anomaliesIQR = boxPlotRule(dataset)
print(anomaliesIQR)

lower limit:  -70.0
upper limit:  130.0
[1000]
```

Slika 2.4. Algoritam *box plot* pravila

Na slici 2.4. prikazan je jednostavan primjer algoritma *box plot* pravila. Prvo se računaju $Q1$ i $Q3$ kvartili i medijan pomoću funkcije *percentile* iz *NumPy* biblioteke. Zatim se računa IQR koji je razlika između $Q3$ i $Q1$. Ta veličina se naziva međukvartilni raspon. Nakon toga se računa donja granica koja je $Q1 - 1,5 * IQR$ i gornja granica koja je $Q3 + 1,5 * IQR$. Isto kao i kod $\mu \pm 3\sigma$ algoritma, sve što je izvan navedenih granica će se tretirati kao anomalija. Također, opet pomoću for petlje u listi se spremaju sve anomalije te se ispisuju.

Algoritam *box plot* pravila će se kasnije koristiti u vremenskim nizovima za otkrivanje

anomalija. Područje između $Q1 - 1,5 * IQR$ i $Q3 + 1,5 * IQR$ sadrži 993% opažanja, zato je ekvivalentna 3σ tehnici za Gaussove podatke [1].

2.2.3. Grubbsov test (z-score)

Zadnji algoritam koji će biti prikazan je Grubbsov test (z-score). Grubbsov test, koji je poznat i kao maksimalni normirani rezidualni test, koristi se za otkrivanje anomalije u jednovarijantnom skupu podataka pod pretpostavkom da su podaci generirani Gaussovim distribucijama. Za svaku testnu instancu x, njezin z rezultat izračunava se na sljedeći način:

$$z = \frac{|x - \bar{x}|}{s} \tag{2.1}$$

gdje su x i s srednja vrijednost i standardna devijacija uzorka podataka. Testna instanca se proglašava anomalijom ako:

$$z > \frac{N-1}{\sqrt{N}} \cdot \sqrt{\frac{t^2_{\alpha/2N,N-2}}{N-2+t^2_{\alpha/2N,N-2}}} \tag{2.2}$$

gdje je N veličina podataka i $t_{\frac{\alpha}{2N},N-2}$ je prag koji se koristi za deklariranje instance (anomalno ili normalno) [1].

3 algoritam: Grubbsov test - z score

```
]: import numpy as np
from scipy.stats import t

def grubbsTest(data, alpha=0.05):
    N = len(data)
    x = np.mean(data)
    s = np.std(data, ddof=1)

    t_critical = t.ppf(1 - alpha / (2 * N), N - 2)
    critical = ((N - 1) / np.sqrt(N)) * np.sqrt(t_critical**2 / (N - 2 + t_critical**2))

    z = np.abs((data - x) / s)
    maxZ = np.max(z)

    anomalies = [data[i] for i in range(N) if z[i] > critical]

    return anomalies, critical

data = [0, 2, 4, 6, 8, 10, 100, 1000]
anomaliesZ, critical = grubbsTest(data)
print(anomaliesZ)
print("Critical value:", critical)
```

```
[1000]
Critical value: 2.1266450871956257
```

Slika 2.5. Algoritam Grubbsovog testa

Na slici 2.5. prikazan je jednostavan primjer algoritma Grubbsovog testa. Prvo se računa duljina podataka N, srednja vrijednost x i standardna devijaciju s. Zatim se računa kritična vrijednost t pomoću funkcije *t.ppf* iz *SciPy* biblioteke. Formula koja se koristi za

izračun t vrijednosti je:

$$t = t.ppf(1 - \frac{\alpha}{2N}, N - 2) \tag{2.3}$$

gdje je α razina značajnosti testa. To je vjerojatnost da će se odbaciti nulta hipoteza kada je ona zapravo istinita. Nakon toga se računa z vrijednost prema formuli 2.3 Zatim se računa z vrijednost prema formuli 2.1 Podaci prolaze kroz for petlju te ako je z veći od kritične vrijednosti t, onda je ta instanca podataka anomalija.

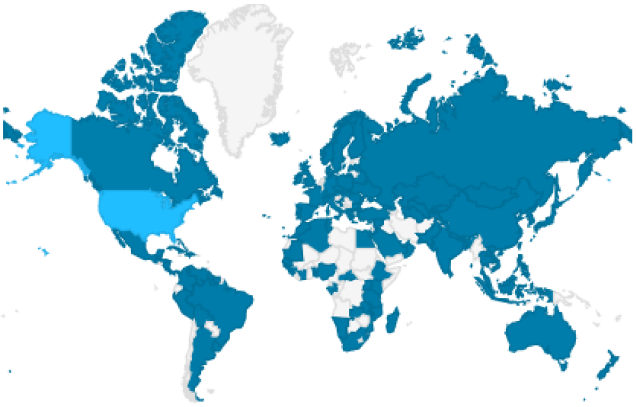
3. Eksperimentalni rezultati

U ovom poglavlju se opisuju eksperimenti kojima se ispitivala učinkovitost algoritama za otkrivanje anomalija u vremenskim nizovima. Prvo će biti opisani skupovi podataka korišteni u ovom radu (potpoglavlje 3.1.). Zatim slijedi potpoglavlje 3.2. u kojem će biti opisano kako se koristi *ADTK* biblioteka, kako se učitavaju podaci te kako se izrađuju grafovi pomoću *ADTK* biblioteke. Poglavlje se završava potpoglavljem 3.3. u kojem će biti prikazani i objašnjeni razni grafovi koji prikazuju anomalije u temperaturnim podacima te padalinama.

3.1. Skupovi podataka

U radu su korištena dva različita skupa podataka za temperaturu i jedan skup podataka za padaline.

Prvi skup podataka koji će biti opisan je skup podataka za temperaturu. Preuzet je s *Kaggle* platforme [5]. Skup podataka koristi temperature u većim svjetskim gradovima. Gradovi koji se nalaze u ovom skupu podataka su veći gradovi sljedećih država: Rusija, Alžir, Australija, Brazil, Kanada, Kina, Francuska, Njemačka, Indija, Italija, Španjolska, Ujedinjeno Kraljevstvo, Sjedinjene Američke Države, i tako dalje. Sve države koje su u ovom skupu podataka su plavom bojom označene na slici 3.1.



Slika 3.1. Države u prvom skupu podataka [5]

Pri učitavanju podataka, redci s istim godinama su grupirani i izračunata je njihova srednja vrijednost.

	index	AvgTemperature
1	1995-01-01	44.11510067114094
2	1995-01-02	41.78338762214984
3	1995-01-03	39.199342105263156
4	1995-01-04	37.294444444444444
5	1995-01-05	35.762622950819676
6	1995-01-06	40.599673202614376
7	1995-01-07	43.02894736842105
8	1995-01-08	42.07920792079208
9	1995-01-09	43.41442307692307
10	1995-01-10	44.830351437699676
11	1995-01-11	46.954983922829584
12	1995-01-12	49.77129032258065
13	1995-01-13	50.271844660194176
14	1995-01-14	49.93562091503268
15	1995-01-15	48.98950819672131

Slika 3.2. Prvi skup podataka za temperaturu

Na slici 3.2. je prikazno prvih 15 redaka skupa podataka za temperaturu. Skup podataka sadrži samo dva stupca, a to su index i srednja vrijednost temperature. Index su datumi oblika godina-mjesec-dan te kreće od 1995-01-01 do 2020-05-13. Znači, skup podataka sadrži srednju temperaturu u Fahrenheitima za svaki dan od 1995-01-01 do 2020-05-13.

Drugi skup podataka je također za temperaturu. Preuzet je s *DataHub* platforme [6]. Skup podataka koristi analizu površinske temperature pomoću NASA Goddard Instituta

za svemirske studije. Opis ovog skupa podataka je naveden u nastavku.

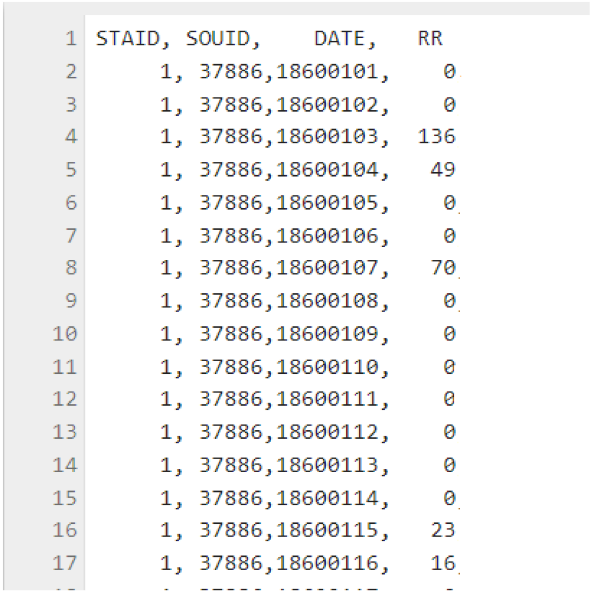
	Year	Month	Mean
1	1880	1	-0.14955
2	1880	2	-0.16645
3	1880	3	-0.15785
4	1880	4	-0.15995
5	1880	5	-0.10690000000000001
6	1880	6	-0.22959999999999997
7	1880	7	-0.19455
8	1880	8	-0.07625
9	1880	9	-0.12585000000000002
10	1880	10	-0.1693
11	1880	11	-0.2286
12	1880	12	-0.14875
13	1881	1	-0.060250000000000005
14	1881	2	-0.08480000000000001
15	1881	3	0.0224

Slika 3.3. Drugi skup podataka za temperaturu

Na slici 3.3. je prikazno prvih 15 redaka drugog skupa podataka za temperaturu. Ovaj skup podataka sadrži tri stupca, koji redom idu: godina, mjesec i temperaturni indeks kopna i mora. Temperaturni indeks je odstupanje od odgovarajućih srednjih vrijednosti.

Pomoću ovog skupa podataka će se vidjeti kako se anomalije prilagođavaju sa stalnim mijenjanjem vrijednosti. To jest, postepeno povećanje temperature će postati normalno te visoke temperature koje su došle s vremenom neće biti anomalije.

Zadnji skup podataka sadrži podatke o padalinama. Preuzet je s *European Climate Assessment & Dataset* platforme [7]. Na stranici se može odabrati država ili meteorološke postaje za koju se žele podaci o padalinama. U ovom radu se koristi skup podataka za Švedsku te je korištena meteorološka postaja *Vaexjoe*. Na slici 3.4. je prikazno prvih 17 redaka skupa podataka za padaline. Skup sadrži četiri stupca. Prvi stupac je STAID koji je jedinstveni identifikator meteorološke postaje. Zatim slijedi stupac SOUID koji je identifikator izvora podataka. Stupac iza njega je DATE koji predstavlja datum u formatu YYYYMMDD. Zadnji stupac je RR koji predstavlja količinu padalina u 0.1 mm.



```
STAID, SOUID,    DATE,   RR
     1, 37886,18600101,    0
     1, 37886,18600102,    0
     1, 37886,18600103,  136
     1, 37886,18600104,   49
     1, 37886,18600105,    0
     1, 37886,18600106,    0
     1, 37886,18600107,   70
     1, 37886,18600108,    0
     1, 37886,18600109,    0
     1, 37886,18600110,    0
     1, 37886,18600111,    0
     1, 37886,18600112,    0
     1, 37886,18600113,    0
     1, 37886,18600114,    0
     1, 37886,18600115,   23
     1, 37886,18600116,   16
```

Slika 3.4. Skup podataka za padaline

U cijelom skupu podataka je STAID 1 i SOUID 37886, što znači da su svi podaci iz iste meteorološke postaje *Vaexjoe*, a to je negdje na području Švedske. Dakle, skup podataka sadrži podatke o svakodnevnim padalinama od 1860-01-01 do 2024-03-31 na području Švedske.

Postoje rupe u podacima, ali to neće utjecati na otkrivanje anomalija. Na područjima di je RR označen s -9999 znači da nije zabilježeno mjerenje te su u uređenom skupu podataka te vrijednosti izbačene.

3.2. Implementacija

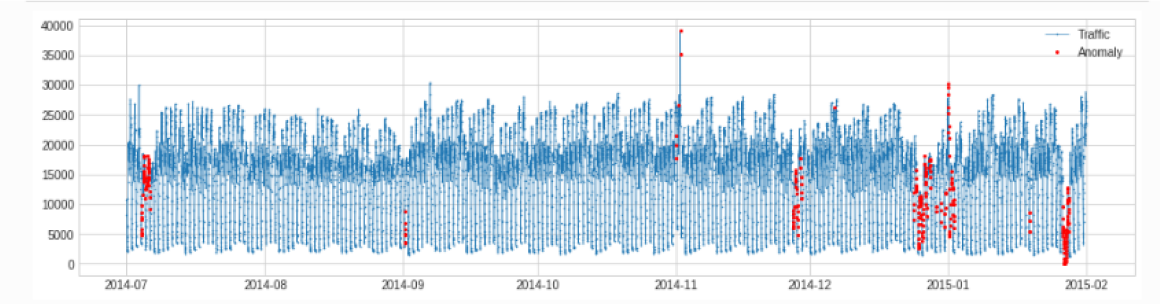
U ovom potpoglavlju će prvo biti opisano što je uopće *ADTK* te kako se koristi (podsekcija 3.2.1.). Za korištenje paketa iz *ADTK* biblioteke potrebno je imati određeni oblik podataka. U podsekciji 3.2.2. je prikazan način kako doći do tog oblika podataka. Na kraju, u podsekciji 3.2.3., će biti prikazano kako se izrađuje graf pomoću *ADTK* biblioteke.

3.2.1. ADTK

ADTK, odnosno *Anomaly Detection Toolkit*, je biblioteka dizajnirana za nenadzirano ili na pravilima temeljeno otkrivanje anomalija u vremenskim nizovima. Ključ za izgradnju učinkovitog modela detekcije anomalija leži u pravilnom odabiru i kombinaciji algo-

ritama za detekciju (detektori), metoda za kreiranje značajki (transformatori) i metoda za agregaciju (agregatori). Ovaj paket pruža niz detektora, transformatora i agregatora s unificiranim API-jima, kao i klase koje ih povezuju u cjeloviti model [8].

ADTK biblioteka ima mnoge pakete za otkrivanje anomalija, no u ovom radu je korišten paket *SeasonalAD*. *SeasonalAD* detektira nepravilnosti u sezonskim obrascima. Interno je implementiran kao cjevovod koji koristi transformator *ClassicSeasonalDecomposition* [9].



Slika 3.5. SeasonalAD

Na slici 3.5. se vidi kako izgleda graf napravljen pomoću *SeasonalAD* paketa. Takvim grafovima se bavi ovaj rad.

3.2.2. Učitavanje podataka

Budući da su kasnije u grafovima korišteni posebni alati za otkrivanje anomalija u vremenskim nizovima, podaci su morali biti učitani na poseban način. U ovom potpoglavlju će biti objašnjeno i pokazano kako podaci moraju izgledati kako bi se mogli dalje koristiti.

```
import pandas as pd
from adtk.data import validate_series
import datetime

data = pd.read_csv('result.csv')
data['Time'] = pd.to_datetime(data[['Year', 'Month']].assign(day=15))
data['Time'] = data['Time'].dt.to_period('M').dt.to_timestamp()
ms_counter = -0.000000001

def increment_ms(x):
    global ms_counter
    ms_counter += 0.000000001
    return ms_counter

data['Time'] += pd.to_timedelta(data.groupby('Time').cumcount().apply(increment_ms), unit='s')
data.set_index('Time', inplace=True)
data.sort_index(inplace=True)
#data.reset_index(inplace=True)
data.rename(columns={'Index': 'Value'}, inplace=True)

s = data['Value']
s.index = pd.to_datetime(data.index)
s.index.freq = 'MS'
s = validate_series(s)
print(s)
```

Slika 3.6. Učitavanje podataka

Na primjeru će biti objašnjeno i pokazano kakvi podaci moraju biti. Na slici 3.6. je prikazan način kako se učitavaju podaci. Prvo se uvoze potrebni paketi *pandas* za rad s podacima i *validate_series* iz *ADTK* za validaciju vremenskih nizova. O *ADTK* će više biti rečeno kasnije. Učitavaju se podaci iz .csv datoteke. Budući da su podaci na mjesečnoj razini, a za daljnji rad nam trebaju svakodnevni podaci, podaci se kombiniraju i kreira se *DateTime* objekt koji je namješten na sredinu mjeseca (15. dan u mjesecu). U vremenskim oznakama se mora osigurati da nema duplikata, pa se svakoj vremenskoj oznaci dodaje mala razlika u milisekundama. Nakon toga se vremenske oznake postavljaju kao indeks *DataFramea* te se sortiraju po tim oznakama. Na kraju se stvara niz iz stupca value i indeks se postavlja na frekvenciju ms. Pomoću funkcije *validate_series* se provjerava i osigurava da vremenski niz zadovoljava određene kriterije potrebne za analizu anomalija. U zadnjem retku se ispisuje niz i dobit će se rezultat kao na slici 3.7.

```
Time
1762-09-01 00:00:00.000000000   -0.227
1762-10-01 00:00:00.000000001    0.491
1762-11-01 00:00:00.000000002    1.915
1762-12-01 00:00:00.000000003    0.456
1763-01-01 00:00:00.000000004    1.883
                                 ...
2010-08-01 00:00:00.000002975    1.067
2010-09-01 00:00:00.000002976    1.020
2010-10-01 00:00:00.000002977    1.207
2010-11-01 00:00:00.000002978    1.509
2010-12-01 00:00:00.000002979    0.625
Freq: MS, Name: Value, Length: 2980, dtype: float64
```

Slika 3.7. Rezultat učitavanja podataka

Na slici 3.7. se vidi kakvi podaci su potrebni za daljnju analizu. Prije izrade svakog grafa je potrebno učitati podatke na ovaj način kako bi se mogli koristiti u alatima za otkrivanje anomalija u vremenskim nizovima.

3.2.3. Pronalaženje anomalija

Na prvom grafu će biti prikazan način izrade grafa za otkrivanje anomalija u vremenskim nizovima koristeći *SeasonalAD* paket. Prvi graf koristi prvi skup podataka koji je opisan u potpoglavlju 3.1. Prvo se podaci uređuju kao što je bilo opisano u potpoglavlju 3.2.2. Nakon sređivanja podaci izgledaju kao na slici 3.8.

```
Time
1995-01-01 00:00:00     6.730611
1995-01-01 00:30:00     6.730611
1995-01-01 01:00:00     6.730611
1995-01-01 01:30:00     6.730611
1995-01-01 02:00:00     6.730611
                          ...
2020-05-12 22:00:00    16.043981
2020-05-12 22:30:00    16.043981
2020-05-12 23:00:00    16.043981
2020-05-12 23:30:00    16.043981
2020-05-13 00:00:00    14.535973
Freq: 30min, Name: Temp, Length: 444673, dtype: float64
```

Slika 3.8. Podaci za prvi graf

Nakon što su podaci sređeni, može se krenuti dalje. Sljedeći korak je odrediti što je u podacima anomalija. U prvom grafu koristi se ugrađena funkcija *fit_detect* koja pronalazi anomalije. Dio koda za pronalaženje anomalija je prikazan na slici 3.9.

```
from adtk.detector import SeasonalAD

seasonal_ad = SeasonalAD(c=2.0, side="both")
anomalies = seasonal_ad.fit_detect(s)
print(anomalies)
anomalies_sum = anomalies.sum()
print(anomalies_sum)
```

Slika 3.9. Pronalaženje anomalija

Kad pronađe anomaliju vraća vrijednost *True*, a kad ne pronađe vraća *False*. Dobiva se boolean niz koji se zatim može koristiti za prikazivanje anomalija na grafu. Taj niz izgleda kao na slici 3.10.

```
Time
1995-01-01 00:00:00    False
1995-01-01 00:30:00    False
1995-01-01 01:00:00    False
1995-01-01 01:30:00    False
1995-01-01 02:00:00    False
                         ...
2020-05-12 22:00:00    False
2020-05-12 22:30:00    False
2020-05-12 23:00:00    False
2020-05-12 23:30:00    False
2020-05-13 00:00:00     True
Freq: 30min, Name: Temp, Length: 444673, dtype: bool
```

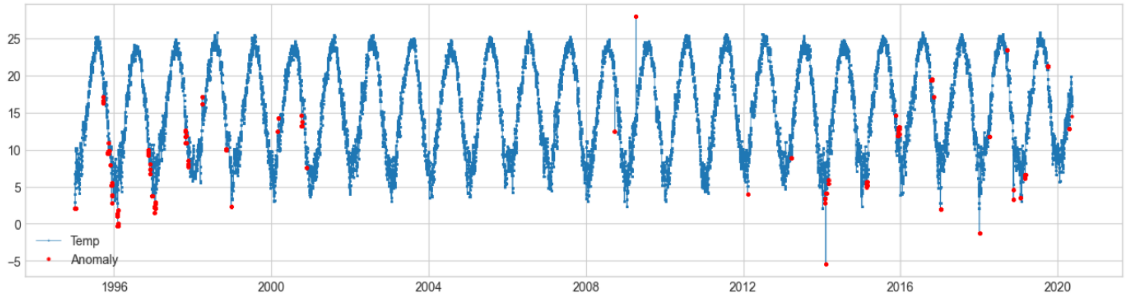
Slika 3.10. Anomalije u podacima

Kad je uspješno dobiven ovaj niz, uvoze se *matplotlib* i *adtk.visualization* paketi za crtanje grafa. Zatim, pomoću funkcije sa slike 3.11. se crta graf.

```
from adtk.visualization import plot
import matplotlib.pyplot as plt

plot(s, anomaly=anomalies, ts_markersize=1, anomaly_color='red', anomaly_tag="marker", anomaly_markersize=2)
plt.show()
```

Slika 3.11. Funkcija za crtanje grafa

Kad se pokrene kod sa slike 3.11. dobije se graf kao na slici 3.12.



Slika 3.12. Graf anomalija

Crvenim bojama su označene anomalije. Na grafu se vidi da su anomalije pronađene na mjestima gdje je temperatura bila iznad ili ispod prosjeka. Može se primijetiti da na početku grafa ima više anomalija nego na ostatku grafa. To je zbog manjka podataka na početku, pa se svaka veća promjena tretira kao anomalija. Vidi se na grafu da kasnije uzima samo vrijednosti koje stvarno odstupaju kao anomalije.

Drugi graf koji će biti prikazan koristi drugi skup podataka koji je opisan u potpoglavlju 3.1. Opet se prvo podaci uređuju kao što je bilo opisano u potpoglavlju 3.2.2. Nakon sređivanja podaci izgledaju kao na slici 3.13.

```
Date
1880-01-01 00:00:00   -0.14955
1880-01-01 12:00:00   -0.14955
1880-01-02 00:00:00   -0.14955
1880-01-02 12:00:00   -0.14955
1880-01-03 00:00:00   -0.14955
                          ...
2016-11-29 00:00:00    0.84020
2016-11-29 12:00:00    0.84020
2016-11-30 00:00:00    0.84020
2016-11-30 12:00:00    0.84020
2016-12-01 00:00:00    0.79975
Freq: 12h, Name: Mean, Length: 100017, dtype: float64
```

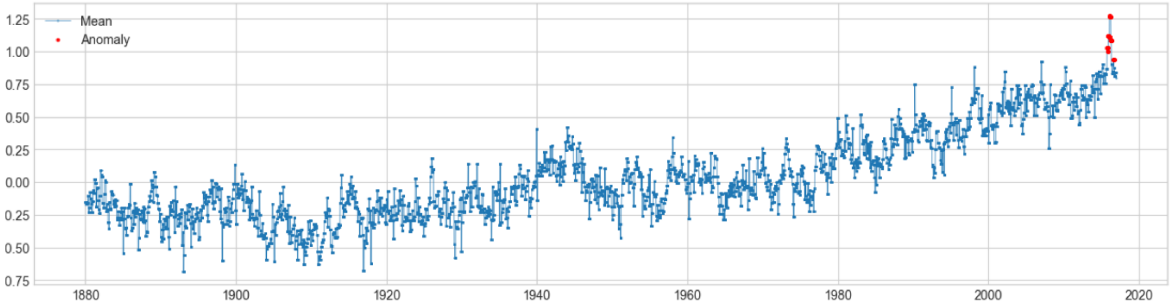
Slika 3.13. Podaci za drugi graf

Ponavlja se postupak kao i kod prvog grafa. Koristi se ugrađena funkcija *fit_detect* koja pronalazi anomalije. Kad ih pronađe dobiju se podaci kao na slici 3.14.

```
Date
1880-01-01 00:00:00    False
1880-01-01 12:00:00    False
1880-01-02 00:00:00    False
1880-01-02 12:00:00    False
1880-01-03 00:00:00    False
                          ...
2016-11-29 00:00:00    False
2016-11-29 12:00:00    False
2016-11-30 00:00:00    False
2016-11-30 12:00:00    False
2016-12-01 00:00:00    False
Freq: 12h, Name: Mean, Length: 100017, dtype: bool
```

Slika 3.14. Anomalije u podacima

Ponavlja se i postupak crtanja grafa i dobiva se graf na slici 3.15.



Slika 3.15. Graf anomalija

Ostaje još prikazati grafove za treći skup podataka iz potpoglavlja 3.1. Prate se koraci koji su na početku potpoglavlja bili opisani. Prvo, se uređuju podaci kao što je bilo opisano u potpoglavlju 3.2.2. Dobiva se skup podataka kao na slici 3.16.

```
Date
1860-01-01 00:00:00      0
1860-01-01 12:00:00      0
1860-01-02 00:00:00      0
1860-01-02 12:00:00      0
1860-01-03 00:00:00    136
                       ...
2019-09-14 00:00:00      1
2019-09-14 12:00:00      1
2019-09-15 00:00:00     13
2019-09-15 12:00:00     13
2019-09-16 00:00:00      2
Freq: 12h, Name: RR, Length: 116665, dtype: int64
```

Slika 3.16. Podaci za treći graf

Drugi korak je pronaći anomalije. Koristi se ugrađena funkcija *fit_detect* koja pronalazi anomalije. Kad ih pronađe dobiju se podaci kao na slici 3.17.

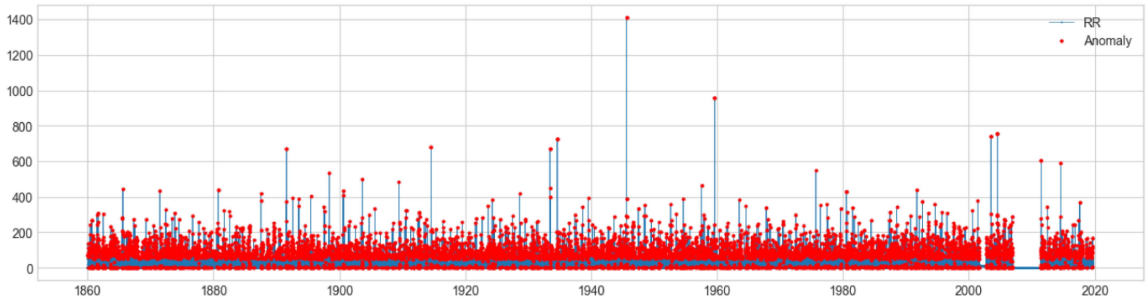
```
Date
1860-01-01 00:00:00    False
1860-01-01 12:00:00    False
1860-01-02 00:00:00    False
1860-01-02 12:00:00    False
1860-01-03 00:00:00    False
                       ...
2019-09-14 00:00:00    False
2019-09-14 12:00:00    False
2019-09-15 00:00:00    False
2019-09-15 12:00:00    False
2019-09-16 00:00:00    False
Freq: 12h, Name: RR, Length: 116665, dtype: bool
```

Slika 3.17. Anomalije u podacima

Funkcijom

```
plot(s, anomaly=anomalies, ts_markersize=1, anomaly_color='red',
 anomaly_tag="marker", anomaly_markersize=2)
```

kao dosad crta se graf i dobiveni rezultat je na slici 3.18.

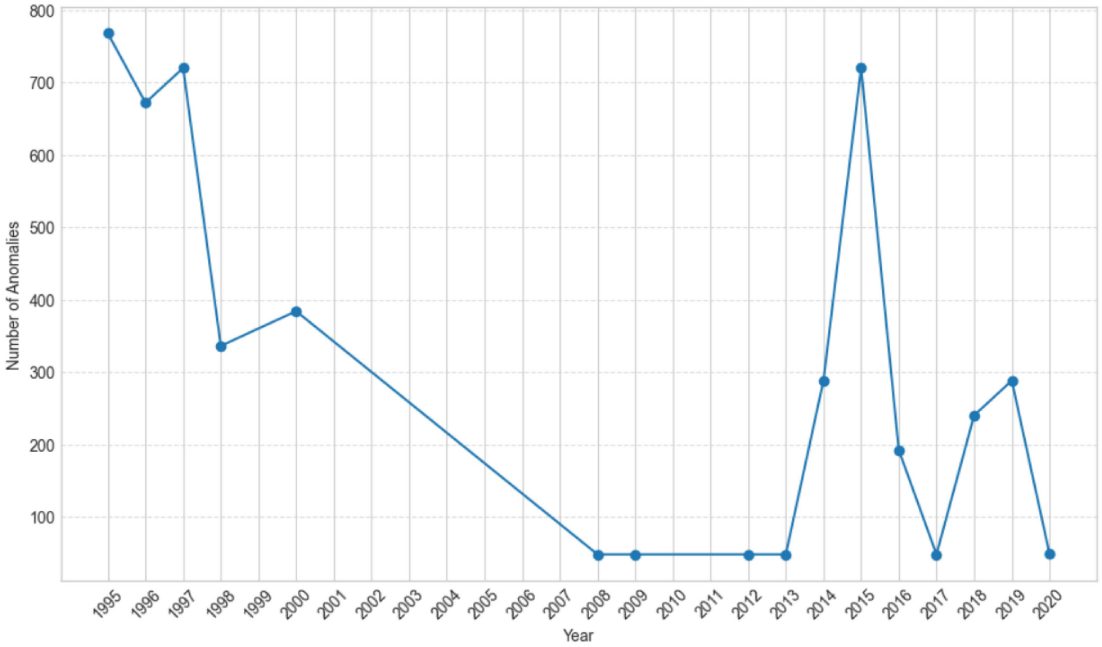


Slika 3.18. Graf anomalija

3.3. Rezultati i rasprava

U ovom potpoglavlju će biti prikazani i opisani svi grafovi koji su napravljeni u ovom radu. Prvo će biti prikazan graf koji će objasniti što se događa s grafom napravljenim u potpoglavlju 3.2.3.

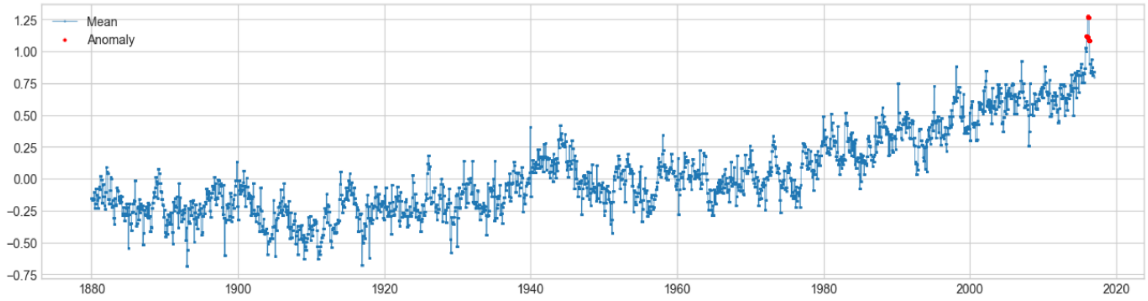
Za bolji pregled što se događa s anomalijama i koliko ih pronalazi, napravljen je drugi graf s istim podacima. Za svaku godinu se zbraja koliko je anomalija bilo te godine i stvara se jednostavan graf koji na x osi ima godine, a na y osi broj anomalija. Takav graf se može vidjeti na slici 3.19.



Slika 3.19. Graf anomalija po godinama

Podaci iz ranih godina mogu biti manje precizni zbog manjih tehničkih mogućnosti i metoda prikupljanja podataka. To može dovesti do većeg broja detektiranih anomalija. Kako se vrijeme približava današnjem vremenu, broj anomalija se smanjuje jer su podaci precizniji i metode prikupljanja podataka su bolje. No, 2015. se opet događa veliki skok u broju anomalija. U ovom slučaju, klimatske promjene dovode do većih varijacija u temperaturama, što rezultira većim brojem anomalija. 2017. i 2020. godine se događa pad. 2017. godine je došlo do pada zbog premalo zabilježenih mjerenja, a 2020. jer su podaci obuhvaćali samo razdoblje do 13. svibnja, a i ta ograničena količina podataka bila je djelomično nezabilježena.

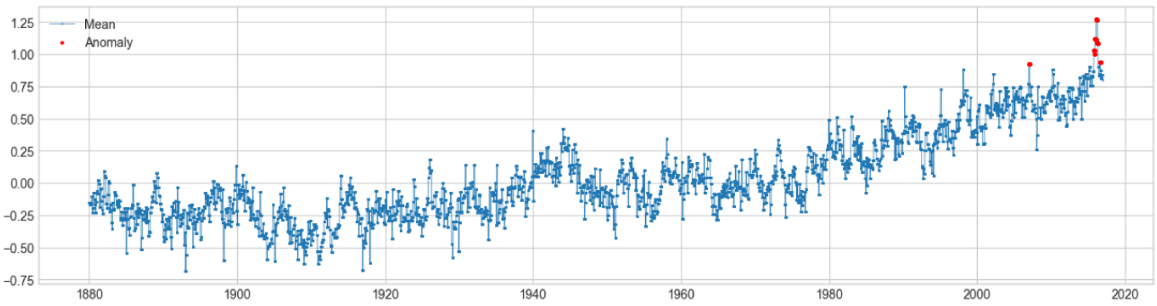
Završetkom prvog grafa, prelazi se na drugi graf, a time i na novi skup podataka. Gleda se graf napravljen drugim po redu opisanim skupom podataka, a taj graf se može vidjeti na slici 3.15. S tim podacima su isprobani i drugi algoritmi za otkrivanje anomalija. Sljedeća slika prikazuje kako izgleda graf s anomalijama koje su pronađene pomoću $\mu \pm 3\sigma$ algoritma.



Slika 3.20. Graf anomalija pomoću $\mu \pm 3\sigma$ algoritma

Na slici 3.20. se vidi da graf s anomalijama koje su pronađene pomoću $\mu \pm 3\sigma$ algoritma nije puno različit od grafa koji je napravljen pomoću ugrađene funkcije. To je dobar znak jer znači da su anomalije pronađene na istim mjestima.

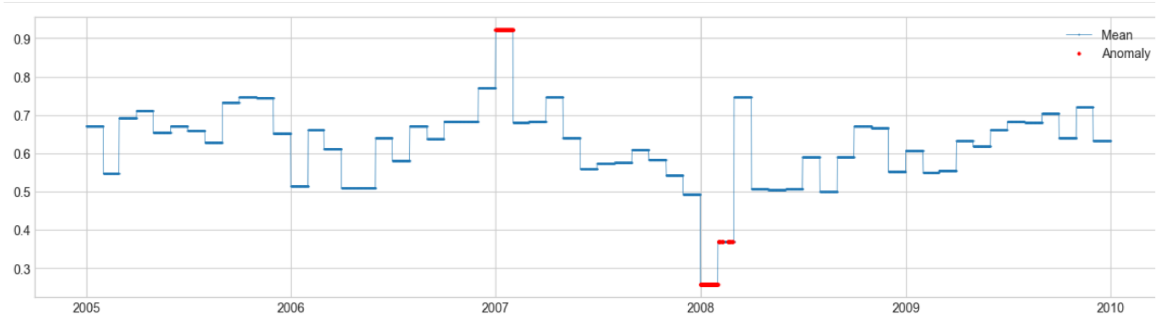
Isproban je i algoritam *box plot* pravila. Na slici 3.21. je prikazan graf s anomalijama koje su pronađene pomoću *box plot* pravila.



Slika 3.21. Graf anomalija pomoću *box plot* pravila

Opet se vidi da su anomalije pronađene na istim mjestima kao i kod prethodnih algoritama. Zasad se čini da su svi algoritmi jednako dobri za otkrivanje anomalija.

No, grafovi koji su prikazani za drugi skup podataka, ne izgledaju kao da pronalaze puno anomalija i kao da su korisni. Time se dokazuje da nekad nije dobro imati preveliki skup podataka za pronalaženje anomalija (situacije kad podaci postepeno rastu ili padaju). Zbog velikog raspona u godinama, algoritmi teško pronalaze anomalije zato što su se anomalije prilagodile normalnim podacima. Nešto što bi prije bilo anomalija, s vremenom postaje normalno. Zato je bolje imati manji skup podataka kako bi se anomalije mogle lakše pronaći. Ako se pogleda isti skup podataka na manjem vremenskom rasponu, može se primijetiti da se anomalije lakše pronalaze.

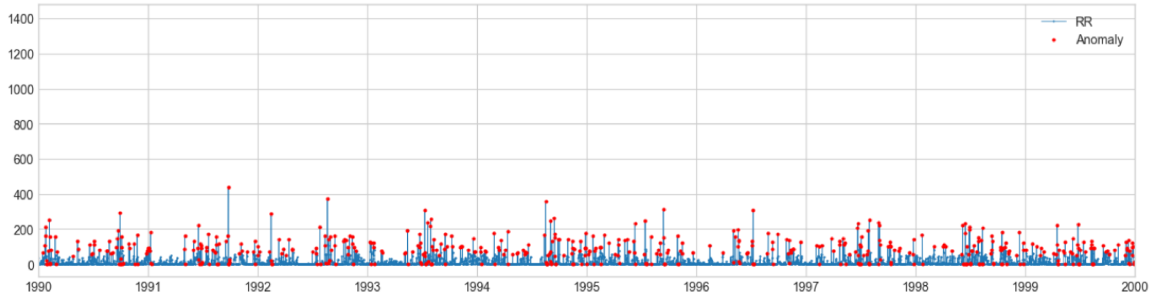


Slika 3.22. Graf anomalija na manjem vremenskom rasponu

Smanjen je vremenski raspon sa 140 godina na samo 5 godina. Uzet je period od 2005. do 2010. godine i na tom periodu su pronađene anomalije. Kad se pogleda cijeli graf od 1880. do 2020. godine, anomalije u periodu od 2005. do 2010. godine uopće ne postoje. To je dokaz prilagođivanja anomalija i loših rezultata na velikim skupovima podataka, dok podaci imaju velike promjene.

Na kraju, treba opisati grafove nastale s trećim skupom podataka. Prvi graf koji će biti opisan je graf koji je opisan i pokazan u potpoglavlju 3.2.3. Na slici 3.18. se vidi kako izgleda taj graf.

Kad se pogleda dobiveni graf, na prvu se može pomisliti da je došlo do neke greške. Skoro svako mjerenje izgleda kao anomalija. Zapravo je problem opet u velikom skupu podataka, ali u ovom slučaju podaci se postepeno ne mijenjaju tako da je svako veće odstupanje zabilježeno kao anomalija. Za bolje razumijevanje ovog grafa, samo će se suziti vremenski raspon na manji period. U ovom slučaju se podaci neće mijenjati kao što je bio slučaj s grafom 3.22.

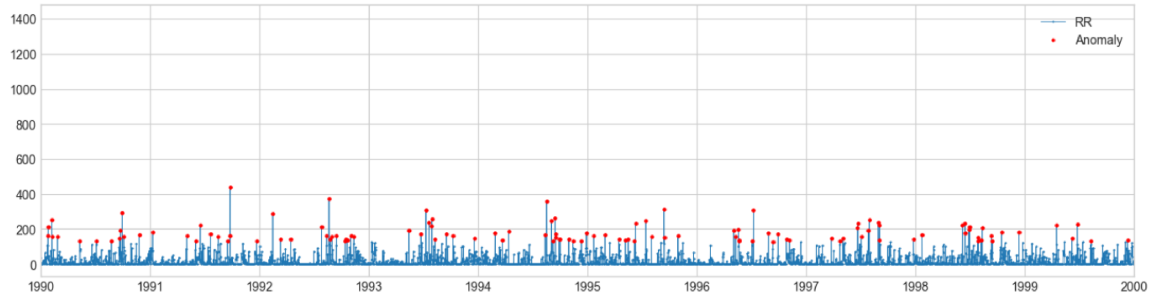


Slika 3.23. Graf anomalija padalina na smanjenom vremenskom rasponu

Na slici 3.23. se vidi da nije svako mjerenje anomalija, već samo ona koja stvarno odstupaju od prosjeka. Ovaj graf je puno bolji i lakše se može vidjeti što su prave anomalije.

Ponovit će se postupak s drugim algoritmima za pronalazak anomalija.

Prvo se radi graf s algoritmom $\mu \pm 3\sigma$. Algoritam tog grafa je vidljiv u potpoglavlju 2.2.1.

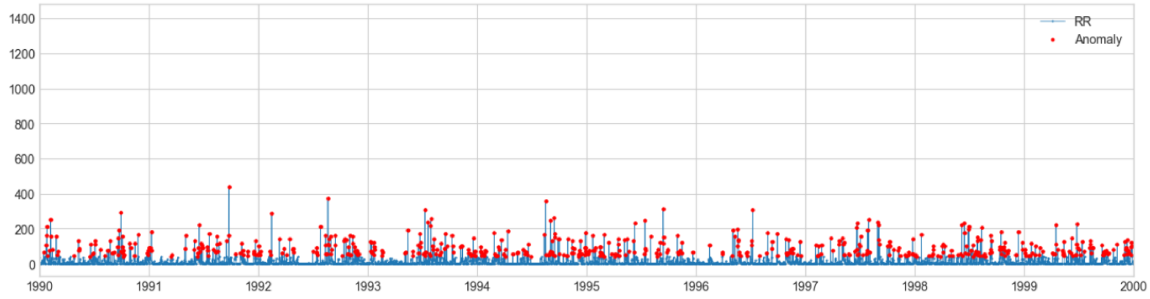


Slika 3.24. Graf anomalija pomoću $\mu \pm 3\sigma$ algoritma

Na slici 3.24. se vidi da je ovaj algoritam našao manje anomalija nego ugrađena funk-

cija, ali pronalazi sve veća odstupanja.

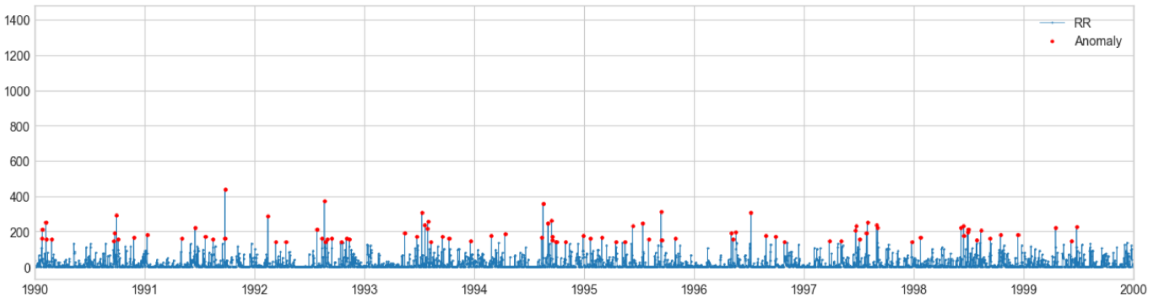
Sljedeći graf će biti napravljen algoritmom *box plot* pravila. Algoritam tog grafa je vidljiv u potpoglavlju 2.2.2.



Slika 3.25. Graf anomalija pomoću *box plot* pravila

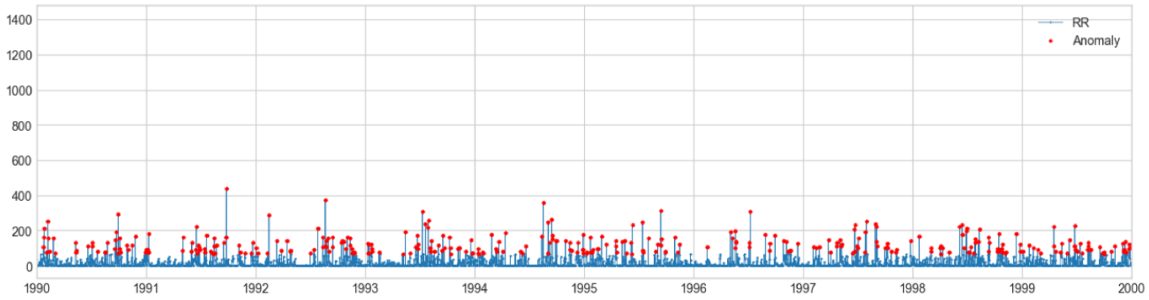
Vidi se na slici 3.25. da je ovaj algoritam bolji od prethodnog. Puno je sličniji ugrađenoj funkcija i nalazi više anomalija. Zasad je *box plot* pravilo najtočnije rješenje koje je isprogramirano u ovom radu.

Još treba provjeriti Grubbsov test (z-score) algoritam. Algoritam tog grafa je vidljiv u potpoglavlju 2.2.3.



Slika 3.26. Graf anomalija pomoću Grubbsovog testa

Na slici 3.26. se vidi da je ovaj algoritam sličan $\mu \pm 3\sigma$ algoritmu. Znači, ne nalazi najbolje anomalije. No, ako se ovom algoritmu poveća vrijednost α, mogu se dobiti bolji rezultati. Nakon povećane vrijednosti α dobije se graf kao na slici 3.27.



Slika 3.27. Graf anomalija pomoću Grubbsovog testa s većom vrijednosti α

Na slici 3.27. se vidi da je ovaj algoritam s većom vrijednosti α bolji od prethodnog. Sada je Grubbsov test sličan *box plot* pravilu za kojeg je rečeno da je dobar algoritam. Znači da je Grubbsov test dobar algoritam za otkrivanje anomalija, ali je potrebno točno podesiti vrijednost α kako bi se dobili dobri rezultati.

4. Zaključak

Ovaj rad se bavi problematikom otkrivanja anomalija u vremenskim nizovima, s naglaskom na vremenske podatke kao što su temperatura i padaline. Kroz detaljnu analizu i primjenu različitih algoritama, dobiveni su vrijedni uvidi u efikasnost i točnost različitih pristupa otkrivanju anomalija.

Korišteni algoritmi uključuju $\mu \pm 3\sigma$, *box plot* pravilo i Grubbsov test (z-score). Svi ovi algoritmi pokazali su sposobnost identifikacije anomalija, ali s različitim stupnjevima preciznosti i primjenjivosti. Na primjer, algoritam $\mu \pm 3\sigma$ je jednostavan za implementaciju i daje solidne rezultate za distribucije koje slijede Gaussovu raspodjelu. *Box plot* pravilo se pokazalo vrlo učinkovitim u identifikaciji anomalija u podacima s manje izraženim sezonskim varijacijama, dok je Grubbsov test bio koristan u jednovarijantnim skupovima podataka. Grubbsov test ovisi o točnom odabiru α parametra.

Eksperimentalni rezultati su pokazali da su svi algoritmi otkrili veći broj anomalija u ranijim godinama zbog manje preciznosti prikupljenih podataka. Međutim, kako se kvaliteta podataka poboljšavala s vremenom, broj detektiranih anomalija se smanjivao, osim u slučajevima značajnih klimatskih promjena. Također, primijećeno je da na velikim skupovima podataka algoritmi mogu imati poteškoća u prepoznavanju anomalija koje su postale uobičajene s vremenom.

Rad je također istaknuo važnost pravilne pripreme podataka i korištenja alata kao što je *ADTK (Anomaly Detection Toolkit)*, koji omogućava fleksibilnost u kombiniranju različitih detektora i transformatora kako bi se postigla optimalna točnost otkrivanja anomalija. Primjenom *ADTK-a* na stvarne skupove podataka, vizualizirane su anomalije te je prikazana njihova distribucija kroz godine, što je dalo dodatni kontekst i razumijevanje rezultata.

Zaključno, ovaj rad pruža sveobuhvatan pregled metodologija i praktičnih aspekata otkrivanja anomalija u vremenskim nizovima. Prikazani rezultati i analize mogu poslužiti kao temelj za daljnja istraživanja i razvoj naprednijih modela za otkrivanje anomalija, koji bi mogli još bolje odgovarati specifičnim zahtjevima različitih domena primjene.

Literatura

[1] V. Chandola, A. Banerjee, i V. Kumar, “Anomaly detection: A survey”, 2009., [PDF dokument, stranica posjećena: lipanj 2024.]. [Mrežno]. Adresa: http://cucis.ece.northwestern.edu/projects/DMS/publications/AnomalyDetection.pdf

[2] ELF11, “Data science: Anomaly detection”, 2018., pristup 13.06.2024. [Mrežno]. Adresa: https://elf11.github.io/2018/09/20/data-science-anomaly-detection.html

[3] Neptune AI, “Anomaly Detection in Time Series”, https://neptune.ai/blog/anomaly-detection-in-time-series, 2023., [online; pristupljeno: lipanj 2024.].

[4] ResearchGate, “Overview of our anomaly detection system”, 2021., pristup 13.06.2024. [Mrežno]. Adresa: https://www.researchgate.net/figure/Overview-of-our-anomaly-detection-system-a-Anomaly-detection-based-on-reconstruction_fig1_349144159

[5] S. Rajkumar, “Daily temperature of major cities”, 2023., accessed: 2024-06-11. [Mrežno]. Adresa: https://www.kaggle.com/datasets/sudalairajkumar/daily-temperature-of-major-cities?resource=download

[6] DataHub.io, “Global temperature time series”, 2023., accessed: 2024-06-11. [Mrežno]. Adresa: https://datahub.io/core/global-temp#readme

[7] E. C. A. . D. (ECA&D), “Eca&d daily data series”, 2023., accessed: 2024-06-11. [Mrežno]. Adresa: https://www.ecad.eu/dailydata/predefinedseries.php

[8] Arundo ADTK Authors, “Arundo ADTK Documentation”, https://arundo-adtk.readthedocs-hosted.com/en/stable/, [mrežno; pristupljeno: lipanj 2024.].

[9] ——, “Arundo ADTK Demo Notebook”, https://arundo-adtk.readthedocs-hosted.com/en/stable/notebooks/demo.html, [mrežno; pristupljeno: lipanj 2024.].

Sažetak

Otkrivanje anomalija u vremenskim nizovima

Josipa Udovičić

Ovaj rad istražuje metode za otkrivanje anomalija u vremenskim nizovima, s posebnim naglaskom na vremenske podatke poput temperature i padalina. Primjenom različitih algoritama, uključujući $\mu \pm 3\sigma$, *box plot* pravilo i Grubbsov test (z-score), analizirana je njihova učinkovitost i točnost u identifikaciji anomalija. Eksperimentalni rezultati pokazuju da su algoritmi uspješno identificirali anomalije, iako se njihova točnost razlikovala ovisno o kvaliteti podataka i duljini vremenskog razdoblja. Korištenjem *ADTK* alata omogućena je fleksibilnost u detekciji anomalija, pružajući vrijedne uvide u sezonske i druge obrasce. Ovaj rad doprinosi razumijevanju metodologija za otkrivanje anomalija te može poslužiti kao osnova za daljnja istraživanja u ovom području.

Ključne riječi: anomalije; vremenski nizovi; otkrivanje anomalija; algoritam $\mu \pm 3\sigma$; *box plot* pravilo; Grubbsov test; *ADTK*; *SeasonalAD*

Abstract

Detecting anomalies in time series

Josipa Udovičić

This study investigates methods for anomaly detection in time series, with a particular focus on weather data such as temperature and precipitation. By applying various algorithms, including $\mu \pm 3\sigma$, box plot rule, and Grubbs' test (z-score), their effectiveness and accuracy in identifying anomalies were analyzed. Experimental results show that the algorithms successfully identified anomalies, although their accuracy varied depending on the data quality and the length of the time period. The use of the ADTK tool provided flexibility in anomaly detection, offering valuable insights into seasonal and other patterns. This study contributes to the understanding of anomaly detection methodologies and can serve as a foundation for further research in this field.

Keywords: anomalies; time series; anomaly detection; $\mu \pm 3\sigma$ algorithm; box plot rule; Grubbs' test; ADTK; SeasonalAD