

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1533

**PRIMJENA SKRIVENIH MARKOVLJEVIH
MODELA ZA MODELIRANJE I
PREDVIĐANJE METEOROLOŠKIH
POJAVA**

Antonela Blažević

Zagreb, lipanj 2017.

DIPLOMSKI ZADATAK br. 1533

Pristupnik: **Antonela Blažević (0036466620)**
Studij: Elektrotehnika i informacijska tehnologija
Profil: Elektroničko i računalno inženjerstvo

Zadatak: **Primjena skrivenih Markovljevih modela za modeliranje i predviđanje meteoroloških pojava**

Opis zadatka:

Skriveni Markovljevi modeli su statistički modeli koji omogućavaju modeliranje sustava s pretpostavkom da su stanja sustava skrivena. Stanje sustava nije vidljivo, ali svako stanje s određenim vjerojatnostima emitira opservacijske simbole na temelju kojih se modeliranjem može procijeniti najvjerojatniji niz stanja koji je uzrokovao niz opservacijskih simbola. U ovom diplomskom radu istražuju se skriveni Markovljevi modeli i njihova mogućnost prilagodbe za meteorološke pojave, ponašanje vremenskih prilika i stanja atmosfere u različitim dobima kalendarske godine. U okviru diplomskog rada potrebno je projektirati i izgraditi jednostavnu meteorološku stanicu za prikupljanje osnovnih podataka iz prirode i njihov prikaz korištenjem ugradbenog računalnog sustava. Na temelju istraženog, provesti gradnju modela koji opisuje meteorološke prilike, te ovaj model iskoristiti u meteorološkoj stanici za predviđanje vremenskih pojava.

Zadatak uručen pristupniku: 10. ožujka 2017.

Rok za predaju rada: 29. lipnja 2017.

Mentor:



Prof. dr. sc. Davor Petrinović

Djelovođa:



Prof. dr. sc. Dražen Jurišić

Predsjednik odbora za
diplomski rad profila:



Prof. dr. sc. Mladen Vučić

Sadržaj

Uvod	1
1. Povijest i metode meteoroloških mjerenja.....	2
1.1. Metode mjerenja vremena	3
1.1.1. Metoda dosljednosti.....	3
1.1.2. Metoda trendova	4
1.1.3. Metoda klimatologije.....	4
1.1.4. Analogna metoda.....	4
1.1.5. Numerička metoda.....	5
1.2. Važnost godišnjih doba	5
2. Analiza vremenskih parametara	7
3. Skriveni Markovljevi modeli.....	13
3.1. Problem evaluacije	15
3.2. Problem dekodiranja.....	17
3.3. Problem učenja	17
4. Realizacija modela.....	19
5. Meteorološka stanica	25
5.1. Komponente	25
5.1.1. Raspberry Pi	25
5.1.2. Senzori	26
5.2. Implementacija	28
5.3. Testiranje	31
6. Primjena modela na RPI2	34
7. Mjerenja i rezultati.....	41
Zaključak	44
Literatura	45

Sažetak.....	46
Summary.....	47
Skraćenice.....	48
Privitak A. Tablice prikupljenih podataka.....	49

Uvod

Vrijeme je jedna od važnijih pojava koja se kroz povijest proučavala i pratila. Vremenske promjene i uvjeti igraju važnu ulogu u planiranju svakodnevnih aktivnosti.

Zbog toga, meteorološke pojave i njihovi parametri bilježe se svakodnevno na velikom broju lokacija na tlu i u zraku kako bi se što točnije analizirali i predvidjeli vremenski uvjeti.

Markovljev model naziv je za model u teoriji vjerojatnosti kojim se modeliraju sustavi koji imaju slučajne promjene stanja. U Markovljevom modelu pretpostavlja se da naredno stanje ovisi o određenom broju prethodnih stanja.

Skrivenim Markovljevim modelom modeliraju se sustavi s pretpostavkom da su stanja skrivena, ali istodobno opservacije uzrokovane pojedinim stanjima vidljive. Svako stanje ima distribuciju vjerojatnosti pojave pojedinog mogućeg opservacijskog simbola. Ovi modeli imaju primjenu u prepoznavanju govora, pisanja, bioinformatički te mnogim drugim.

Upravo zbog karakteristika međusobne ovisnosti skrivenih stanja i opservacijskih simbola, skriveni Markovljevi modeli odabrani su za pokušaj modeliranja meteoroloških pojava.

U ovom radu istražuju se skriveni Markovljevi modeli te moguće primjene. Istražiti će se i analizirati ponašanje meteoroloških prilika i stanja atmosfere u različitim dobima kalendarske godine. Rezultati analize će se prikazati te će se istražiti mogućnosti prilagodbe skrivenih Markovljevih modela sa statičkim vjerojatnostima za modeliranje meteoroloških pojava.

Upotrebom ugradbenog računalnog sustava sastavit će se meteorološka stanica za prikupljanje podataka iz atmosfere koji će zatim biti upotrijebljeni za testiranje i prikaz rezultata modela dobivenim iz istraživanja meteoroloških uvjeta.

1. Povijest i metode meteoroloških mjerenja

Vrijeme je vrlo kompleksna pojava te zahtijeva korištenje mnogih parametara za određivanje vremena. Korištenjem najnaprednijih algoritama i metoda, i dalje ostaje vrlo nepredvidivo i zahtjevno ispravno predvidjeti vrijeme.

Kroz povijest su se predviđanja temeljila uglavnom na opažanju i iskustvu ljudi. Tek nakon razvoja tehnologije i upoznavanja fizikalnih zakona, predviđanja su se počela temeljiti na matematičkim izračunima u kombinaciji sa opažanjima.

Za izradu vremenske prognoze koriste se podaci dobiveni mjerenjem sa raznih meteoroloških postaja smještenih na tlu (Slika 1) i u zraku. Ta se mjerenja vrše svakodnevno, periodički te se pohranjuju u arhiv Državnog hidrometeorološkog zavoda.



Slika 1.1 Primjer meteorološke stanice na tlu¹

¹ Izvor slike: https://upload.wikimedia.org/wikipedia/commons/3/35/AWS%28Automatic_Weather_station%29.JPG

Neki od osnovnih parametara koji se koriste prilikom mjerenja su temperatura, tlak, vlaga, smjer i brzina vjetra, doba godine, vremenski uvjeti prethodnih dana i mnogi drugi.

1.1. Metode mjerenja vremena

Danas postoji veliki broj različitih metoda mjerenja i predviđanja vremenskih pojava [1]. Način mjerenja ovisi i o potrebama za koje se dobiveni podaci namjeravaju koristiti, pa se tako vrijeme može mjeriti [2]:

- Kratkoročno (do 3 dana unaprijed)
- Srednjoročno (3 do 10 dana unaprijed)
- Dugoročno (od 10 dana do mjesec dana unaprijed)

Neke od suvremenih metoda za predikciju vremena su sljedeće:

- „*Persistence*“ metoda (metoda ustrajnosti)
- „*Trends*“ metoda (metoda trendova)
- Metoda klimatologije
- Analogna metoda
- Numerička metoda

I dan danas postoji veliki broj istraživanja vremenskih pojava i načina kako ih najpreciznije izmjeriti te predvidjeti. Metoda koja se koristi bira se na temelju iskustva u predviđanju vremena, složenosti situacije te potrebne točnosti predviđanja vremena.

1.1.1. Metoda dosljednosti

Metoda dosljednosti je najjednostavnija metoda predviđanja vremena. Ova metoda se temelji na tome da se vrijeme ne mijenja, odnosno vrijeme kakvo je bilo jučer je najvjerojatnije i vrijeme danas [3].

Ova metoda je dobra kada se vremenski uvjeti rijetko mijenjaju iz dana u dan kroz duži period vremena. Upravo zbog toga, u uvjetima gdje se vrijeme učestalo mijenja, ova metoda ne daje zadovoljavajuće rezultate.

Međutim, metoda dosljednosti ipak daje dobre rezultate kada želimo predvidjeti dugoročno vrijeme. Na primjer, u slučaju predikcije hoće li neko godišnje doba biti kišovito ili

sunčano, odnosno koja će vremenska pojava prevladavati u nekom dužem vremenskom periodu.

1.1.2. Metoda trendova

Metoda trendova uključuje mjerenje brzine i smjera fronti, visokog i niskog tlaka, odnosno ciklone i anticiklone te područja oblaka i padalina [4].

Upravo zbog navedenih parametara koji se ovom metodom uzimaju u obzir, ona odlično služi za predviđanje vjerojatnosti padalina u kraćem vremenskom razdoblju (unutar 24 sata). Uvjet za dobru predikciju je kontinuiran i nepromjenjiv smjer kretanja fronte koja se prati.

Problem kod predviđanja nastaje kada se fronte kreću brzo te često mijenjaju smjer kretanja, ili prilikom dužeg vremenskog perioda budući da se fronte rijetko ponašaju isto u vremenskom razdoblju većem od 24 sata. U takvim slučajevima ova metoda gubi na točnosti.

1.1.3. Metoda klimatologije

Metoda klimatologije je metoda koja se temelji na predviđanju vremena koristeći podatke prikupljene i spremljene kroz duže vremensko razdoblje. Ova metoda koristi izmjerene podatke u prošlosti na datum mjerenja te uzima prosjek prijašnjih vremenskih uvjeta koji zatim služe kao predikcija [4].

Glavna mana ove metode je, ukoliko trenutno vrijeme znatno odudara od prethodnih godina zbog promjene klime, ili je naprosto iznimka, metoda tada podbacuje.

1.1.4. Analogna metoda

Analogna metoda uzima u obzir trenutno vrijeme te ga uspoređuje sa najslićnijim vremenom u prošlosti. Zatim pretpostavlja da će se vrijeme ponašati prema tom modelu[4].

Ova metoda je kompleksne prirode jer je vrlo teško pronaći idealnu analogiju u vremenu, a kod vremena čak i najmanje različitosti mogu dovesti do potpuno drugačijih rezultata od očekivanih.

Iako, sa povećanjem baza izmjerenih podataka, s vremenom postaje sve lakše pronaći odgovarajuću analogiju.

1.1.5. Numerička metoda

Numerička metoda koristi računala pri predikciji vremena. Koriste se superračunala sa velikom bazom podataka i stalno nadolazećih izmjerenih parametara sa brojnih meteoroloških stanica [4].

Ni ova metoda nije potpuno precizna s obzirom da nema pristup svim vremenskim uvjetima na svim lokacijama. Ako je početno stanje na nekoj lokaciji nepoznato, tada računalo nije u mogućnosti točno predvidjeti ponašanje vremena.

Unatoč tome što i ova metoda ima mane, i dalje je jedna od trenutno najboljih postojećih metoda za kratkoročnu predikciju vremena.

1.2. Važnost godišnjih doba

Kako bi odredili vrijeme, potrebno je uzeti u obzir veći broj parametara. Kao što je već navedeno, osnovni parametri za predikciju su tlak, vlaga i temperatura zraka te će se ova tri parametra u ovom radu detaljnije razmatrati.

Svaki od ova tri parametra ima neka karakteristična, odnosno slična ponašanja koja se mogu pripisati određenim vremenskim pojavama (sunčanom, kišovitom, oblačnom, itd.).

Osim toga, prognoza vremena ovisi i u kojem se godišnjem dobu vrše mjerenja. Poznato je kako je ljeti vjerojatnost za sunčano vrijeme mnogo veća nego za kišovito vrijeme, ili zimi veća vjerojatnost za oblačno vrijeme nego li za sunčano.

Dakako, i dalje se radi o vjerojatnostima, pa se ne može za sigurnošću tvrditi kako su ove tvrdnje uvijek ispravne zbog nepredvidivosti vremenskih pojava. U velikom broju slučajeva, predikcije se pokažu točnima, ali postoje iznimke, koje s obzirom da se radi o mjerenju vremenskih pojava mogu biti i vrlo česte ako se radi o neobičnim i specifičnim okolnostima za trenutno mjereno vrijeme koje nije moguće predvidjeti.

Tlak, vlaga i temperatura zraka imaju različite vrijednosti te različito ponašanje ovisno u godišnjem dobu. Zbog toga će biti vrlo važno model prilagoditi određenom godišnjem dobu za koje želimo modelirati vrijeme.

Godišnja doba dijele se prema zajedničkim obilježjima unutar jedne godine. Postoje dvije osnovne podjele godišnjih doba:

- Astronomska
- Meteorološka

Pod astronomskom podjelom podrazumijeva se općepoznata podjela godišnjih doba koja se mijenjaju prema odnosu položaja Zemlje i Sunca. Ovom podjelom postoje 4 godišnja doba, redom proljeće, ljeto, jesen i zima, svako sa datumom početka i kraja:

- Proljeće sa početkom 20.3.
- Ljeto sa početkom 21.6.
- Jesen sa početkom 23.9.
- Zima sa početkom 21.12.

S druge strane, meteorološka podjela godišnjih doba ovisi o meteorološkim, odnosno vremenskim uvjetima karakteristični za neko doba godine te se ovi datumi nešto razlikuju od onih astronomskih. Stoga, kada govorimo o meteorološkim godišnjim dobima, oni imaju nešto drugačije datume:

- Proljeće sa početkom 1.3.
- Ljeto sa početkom 1.6.
- Jesen sa početkom 1.9.
- Zima sa početkom 1.12.

U daljnjem tekstu rada govoriti će se o meteorološkim godišnjim dobima jer oni povezuju karakteristične vremenske parametre za neko doba što je od izuzetne važnosti kod izgradnje modela za prepoznavanje vremena.

2. Analiza vremenskih parametara

Kako bi proučili ponašanje vremenskih parametara u različitim godišnjim dobima, preuzeti su podaci iz arhive za proteklu 2016. godinu te parametri vremena.

Pošto se radi o velikom broju podataka, uzeti će se podaci vremena od svakog godišnjeg doba u trajanju od 60 dana u sredini ciklusa svakog dana u podne. Proučavanih 60 dana po godišnjim dobima su sljedeći:

- Proljeće – 16.3.-14.5.
- Ljeto – 16.6.-14.8.
- Jesen – 16.9.-14.11
- Zima – 15.12-12.2

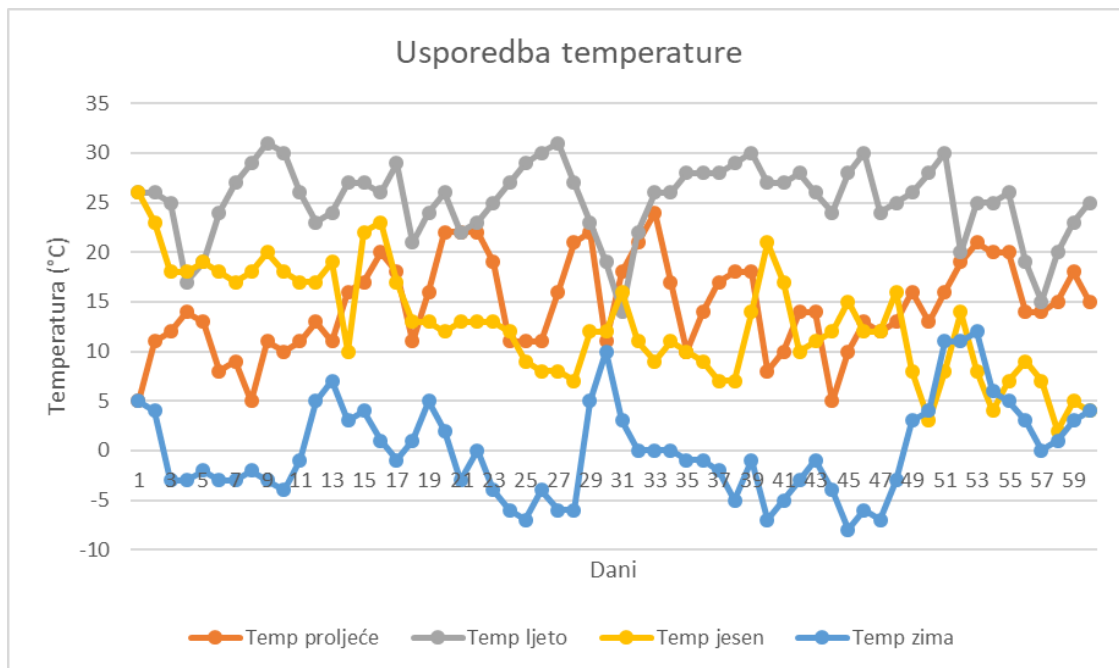
Ovi datumi su izabrani na način da se uzme srednjih 60 dana iz svakog godišnjeg doba. Iz toga je potom moguće prepoznati uzorke ponašanja vremena u pojedinom godišnjem dobu te u konačnici izgraditi specifični model za odabrano godišnje doba.

Podaci o vremenskim prilikama u navedenim razdobljima godine preuzeti su sa internetske stranice koja sadrži podatke, odnosno parametre vremena i vremenske uvjete za željeno područje mjerenja i doba dana od zadnjih 20 godina [4].

Proučavanjem parametara i vremena koje uzrokuju prvo se zamjećuje očita razlika pojave vremenskih uvjeta u različitim godišnjim dobima.

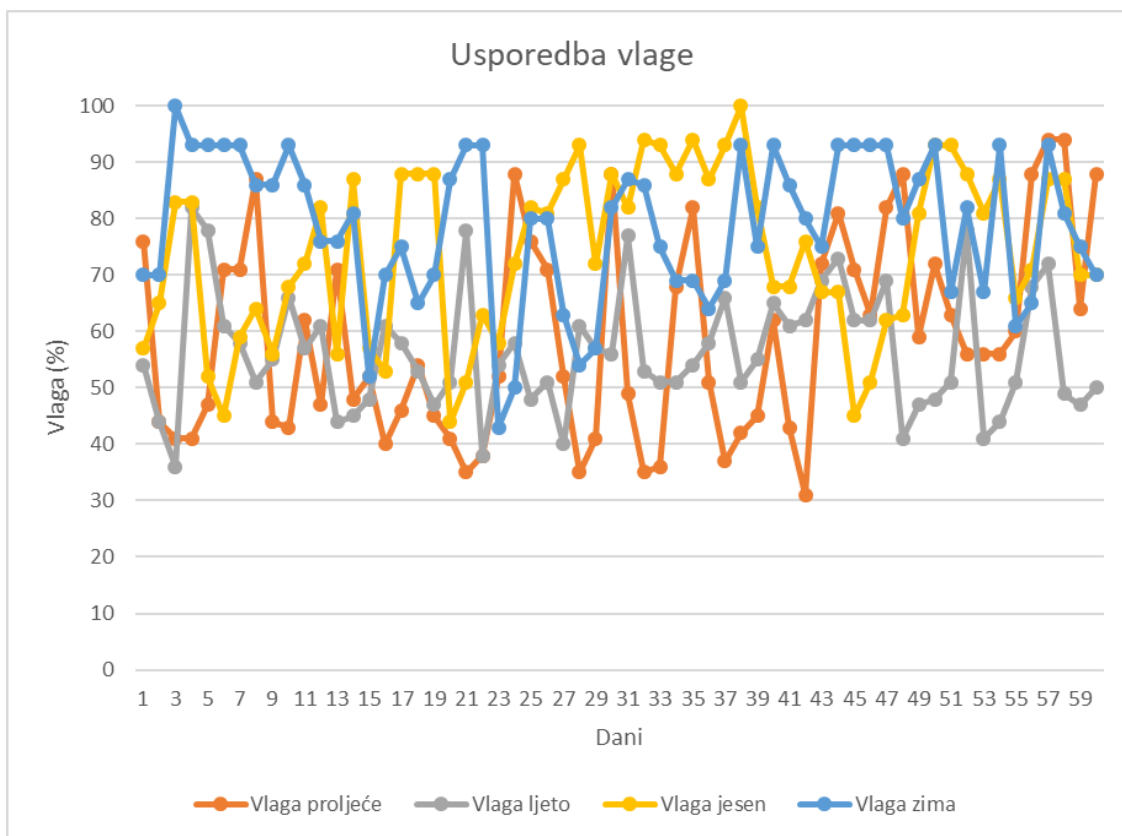
Ljeti je najučestalije sunčano vrijeme i visoke temperature, jesen i proljeće obiluju kišama, dok zima ima karakteristične niske temperature i tmurno, oblačno vrijeme s čestim padalinama.

Ako se uspoređi temperatura kroz sva četiri godišnja doba (Slika 2.1) vidljive su karakteristične linije za svako godišnje doba: visoke temperature ljeti, niske temperature zimi, te variranje temperature u jeseni i proljeću.



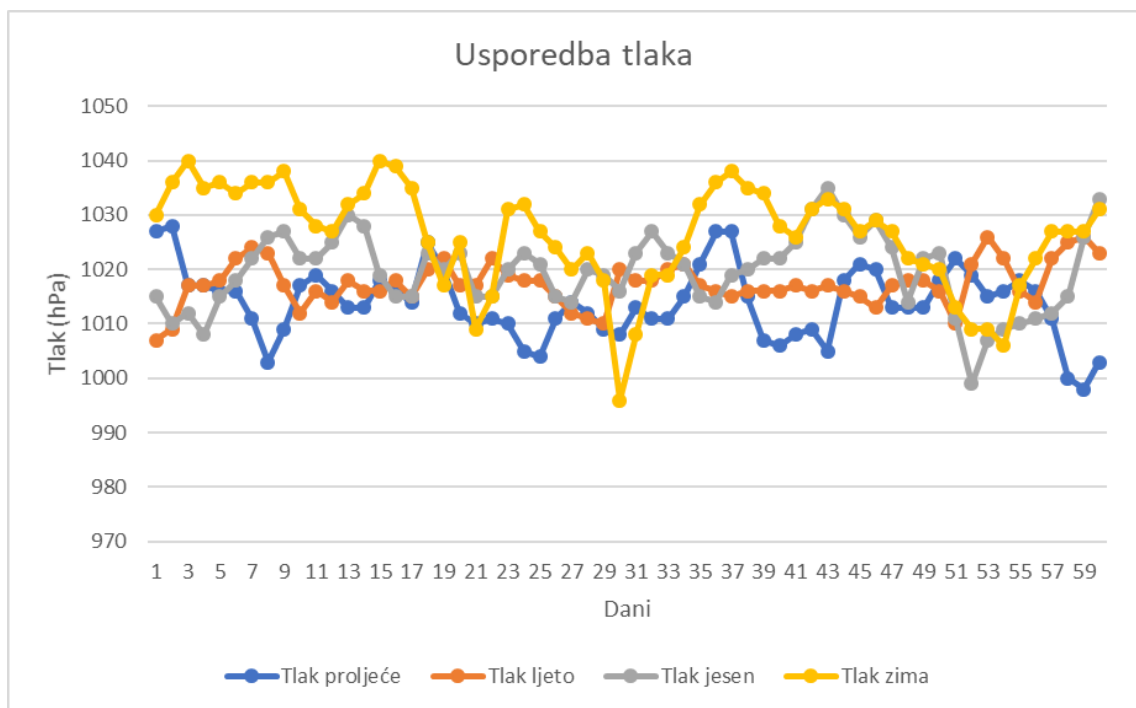
Slika 2. 1 Usporedba temperature kroz godišnja doba

Usporedbom vlage kroz godišnja doba (Slika 2.2), očita razlika je nešto manje vidljiva, odnosno usporedbom vlage kroz godišnja doba se ne dobije korisna informacija, budući da vlaga u svim dobima često varira. Međutim, kasnije, usporedbom vlage sa pojedinom vremenskom pojavom pojavit će se jasniji rezultati.



Slika 2. 2 Usporedba vlage kroz godišnja doba

Na kraju, usporede li se vrijednosti tlaka kroz godišnja doba (Slika 2.3), također se dobije variranje u vrijednostima. Naime, tlak ima slične vrijednosti u svakom godišnjem dobu, te on sam, ako nije u kombinaciji sa drugim stanjima atmosfere, nije osobito koristan prilikom predikcije vremena.

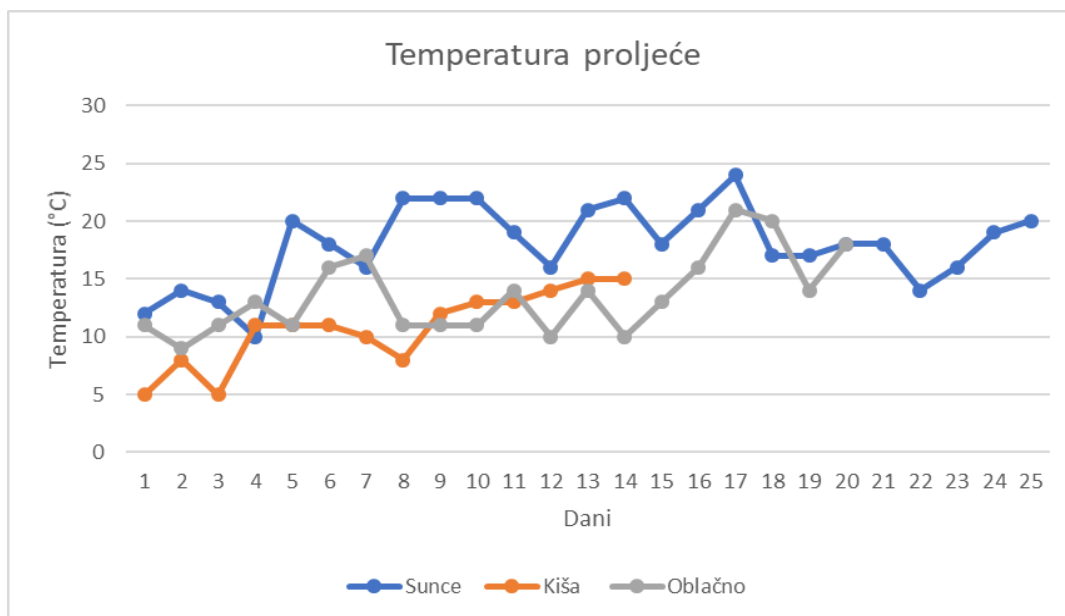


Slika 2.3 Usporedba tlaka kroz godišnja doba

Budući da se trenutno nalazimo u proljeću te će sva praktična mjerenja biti izvođena u tim uvjetima, za modeliranje je izabrano godišnje doba proljeće.

Kako bi se modelirale meteorološke pojave u proljeću potrebno je iz podataka proljeća istražiti karakteristična ponašanja ili variranja tlaka, temperature i vlage za pojedinu vremensku pojavu.

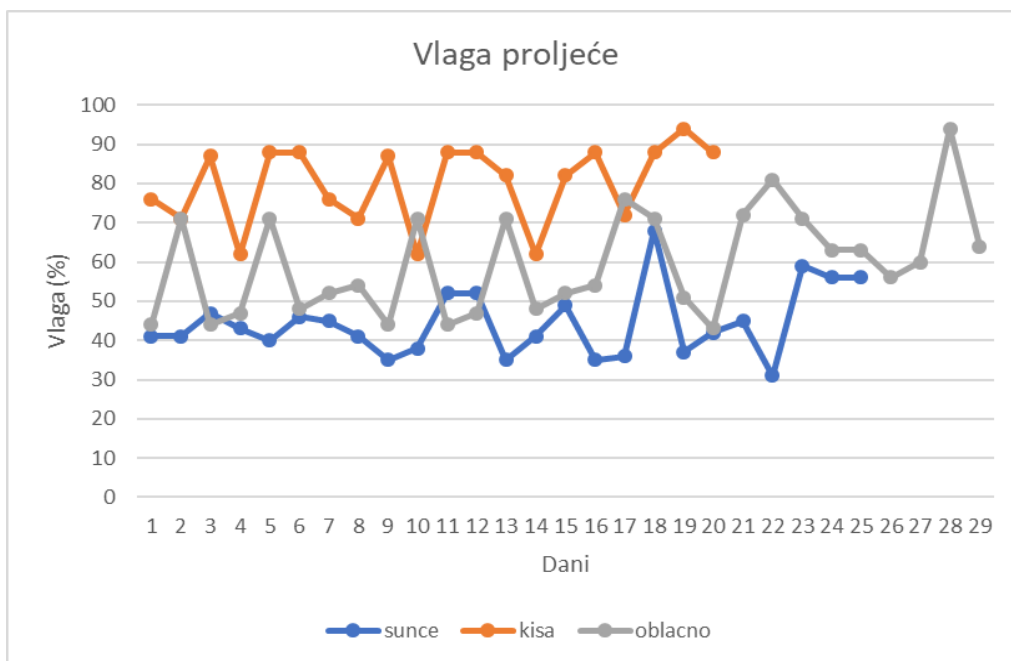
Izdvajanjem parametara temperatura proljeća kroz odabranih 60 dana, vidljivo je (Slika 2.4) kako visoke temperature najčešće znače sunčano vrijeme, dok najniže najčešće znače kišu.



Slika 2. 4 Prikaz temperature u ovisnosti o vremenskih pojavama (proljeće)

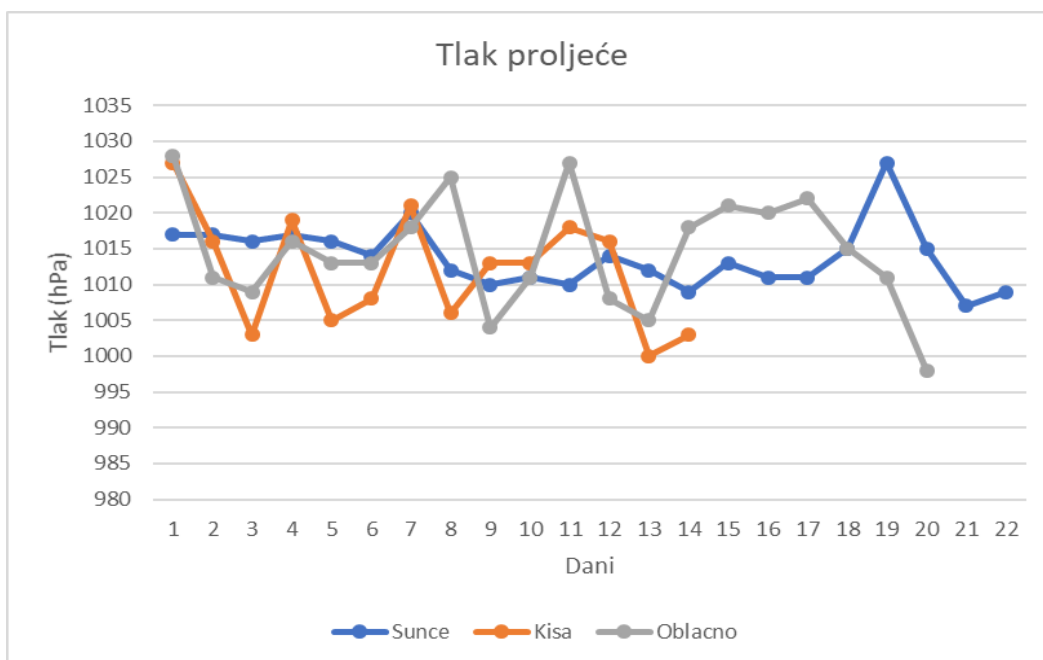
Prikaz vlage (Slika 2.5) daje do znanja kako vlaga gotovo uvijek jednoznačno određuje vrijeme. Vidljivo je kako vlaga iznad vrijednosti 80% gotovo uvijek znači kišu. Isto tako, vlaga ispod vrijednosti 40% gotovo uvijek znači vedro i sunčano vrijeme. Vrijednost vlage između te dvije vrijednosti najčešće znači oblačno vrijeme, ali postoji mogućnost i za kišom ili sunčanim vremenom. Naravno, ovdje se promatraju podaci karakteristični za proljeće, za neko drugo godišnje doba parametri bi imali drukčije karakteristike.

Budući da se traže vjerojatnosti za pojedinu pojavu, ovi podaci će biti izuzetno bitni i korisni prilikom izgradnje modela.



Slika 2. 5 Prikaz vlage u ovisnosti o vremenskim pojavama (proljeće)

Ponašanje tlaka ovisno u vremenskoj pojavi (Slika 2.6) su zapetljane linije, odnosno vrijednost tlaka nema čvrste granice koje bi pomogle pri određivanju vremenskog uvjeta proučavanjem samo tlaka i vremena.

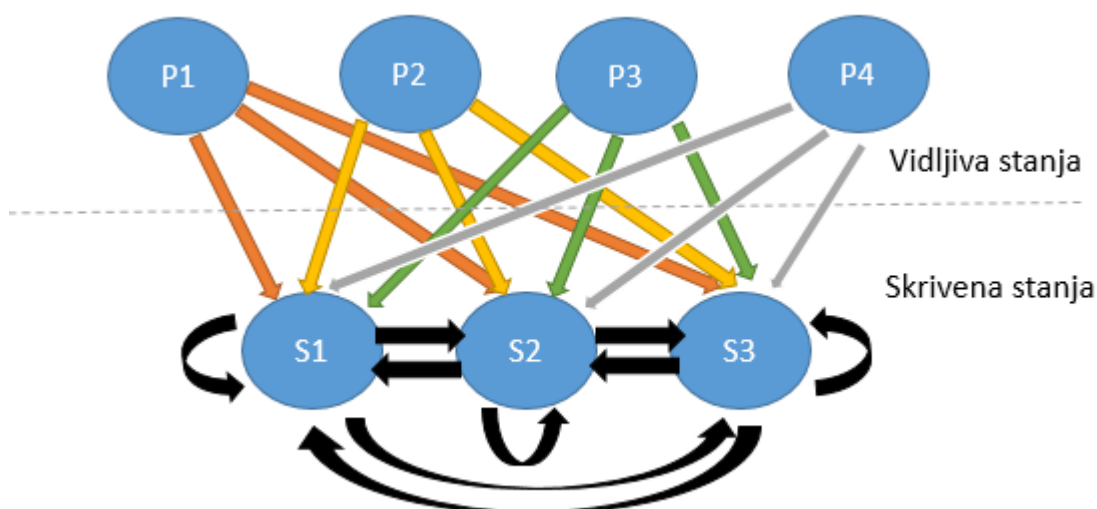


Slika 2. 6 Prikaz tlaka u ovisnosti o vremenskim pojavama (proljeće)

3. Skriveni Markovljevi modeli

Skriveni Markovljev model kombinira automat konačnog broja stanja sa teorijom stohastičkih procesa. Može se okarakterizirati kao sustav sa konačnim brojem stanja koja nisu direktno vidljiva, ali mogu biti predviđena korištenjem raznih vanjskih parametara koji su povezani sa skrivenim stanjima.

U nekim situacijama klasični Markovljev model nije dovoljan za opisati određene pojave. Moguće je proučavati određene situacije u kojima je onemogućen pristup parametrima sustava, odnosno neki parametri sustava su nevidljivi (skriveni). Na temelju mjerenja određene veličine koja nam je poznata, potrebno je predvidjeti veličinu koja nam je potrebna, odnosno koja nam je nepoznata. Ovisno o kojem se procesu radi, postoji više mogućnosti broja skrivenih stanja te onih vidljivih. Svako promatrano stanje je određenom vjerojatnošću povezano sa skrivenim procesom.



Slika 3. 1 Dijagram stanja i opservacijskih simbola skrivenog Markovljevog modela

Kao što je prikazano na *Slici 3.1* svako skriveno stanje je povezano sa vidljivim stanjem, odnosno parametrom dobivenim opservacijom preko vjerojatnosti koja određuje pojavu određenog skrivenog stanja.

Drugim riječima, svako skriveno stanje je sa određenom vjerojatnošću povezano sa vidljivim stanjem, odnosno opservacijom, što znači da se radi o uvjetnim vjerojatnostima.

Uvjetna vjerojatnost se može zapisati prema Bayesovom pravilu:

$$P(q_1, \dots, q_N | o_1, \dots, o_N) = \frac{P(o_1, \dots, o_N | q_1, \dots, q_N) P(q_1, \dots, q_N)}{P(o_1, \dots, o_N)}$$

, gdje je $Q = \{q_1, \dots, q_N\}$ skup skrivenih stanja, a $O = \{o_1, \dots, o_N\}$ slijed vidljivih (opservacijskih) simbola. Iz promatranih opservacijskih simbola potrebno je izvući zaključke o vrijednostima skrivenih stanja. Općenito, proces koji je u nekom stanju S_i , može sa određenom vjerojatnošću emitirati bilo koji element iz skupa:

$$T = \{b_1, b_2, \dots, b_M\}$$

Značajku koju proces emitira u nekom trenutku t , stanju q_t označi se sa v_t . Što znači, osim niza Q stanja kroz koje proces prolazi, može se promatrati i niz V (ili O) opservacija u tim stanjima. Općenito, kada su stanja s_i skrivena stanja, a O_t jedina vidljiva stanja, naziva se skrivenim Markovljevim modelima. U modelima se prilikom prijelaza emitira opservacijski simbol (vidljivo stanje).

Osnovne komponente skrivenih Markovljevih modela su sljedeće [6]:

- Konačni skup skrivenih stanja
- Konačni skup opservacijskih simbola
- Vektor inicijalnih prijelaznih vjerojatnosti
- Matrica vjerojatnosti prijelaza
- Matrica vjerojatnosti promatranja

Skriveni Markovljev model je definiran sa petorkom $\theta = \{S, V, \pi, A, B\}$, gdje je:

- $S = \{S_1, \dots, S_N\}$, skup skrivenih stanja, gdje je N broj stanja u kojima se proces može nalaziti
- $V = \{v_1, \dots, v_M\}$, skup mogućih opservacijskih simbola, gdje je M broj različitih opservacijskih simbola
- π je N -dimenzionalni vektor inicijalnih prijelaznih vjerojatnosti iz stanja u stanje:

$$\pi = |\pi_i| \pi_i = P(q_0 S_i), 1 \leq i \leq N$$
- A je matrica prijelaznih vjerojatnosti iz stanja u stanje, dimenzija $N \times N$ odnosno vjerojatnosti prijelaza iz stanja i u stanje j :

$$A = [a_{i,j}], a_{i,j} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i, j \leq N,$$

- B je matrica vjerojatnosti opservacija, dimenzija $N \times M$:

$$B = [b_j(k)], b_j(k) = P(O_t = b_k | q_t = S_j), 1 \leq j \leq N, 1 \leq k \leq M$$

Budući da su prva dva elementa petorke, S i V , implicitno definirana u ostalima, uređena trojka $\theta = \{\pi, A, B\}$ može se koristiti kao potpuni opis svakog skrivenog Markovljevog modela.

Skriveni Markovljevi modeli omogućuju različito modeliranje mnogih stvarnih procesa, ali prilikom implementacije postoje tri ključna problema:

- Problem evaluacije
- Problem dekodiranja
- Problem učenja

3.1. Problem evaluacije

Ukoliko je dan slijed opservacija $O^T = O_0 O_1 \dots O_M$ te ako je dan model $\theta = \{\pi, A, B\}$, potrebno je odrediti $P(O|\theta)$, odnosno vjerojatnost da je dani model generirao slijed opservacija.

Vjerojatnost da je generiran slijed je $P(V^T) = P(O^T | s_r^T) P(s_r^T)$, gdje je r pojedina sekvenca od $S_r^T = \{s(1), s(2), \dots, s(T)\}$ od T skrivenih stanja. Općenito, ako postoji c skrivenih stanja analitički dobivamo postojanje $r_{max} = c^T$ mogućih članova u sumi. Potrebno je uzeti u obzir sve moguće kombinacije skrivenih stanja, izračunati vjerojatnost da je ta kombinacija emitirala dati slijed i zbrojiti te vjerojatnosti.

Vjerojatnost da se pojavio određeni opservacijski slijed je umnožak tranzicijske vjerojatnosti a_{ij} i vjerojatnosti emitiranih simbola b_{jk} u svakom koraku.[6]

Izraz $P(s_r^T)$ možemo zapisati kao umnožak svih elemenata a_{ij} koji odgovaraju skrivenom slijedu stanja:

$$P(s_r^T) = \prod_{t=1}^T P(s(t)|s(t-1))$$

Izraz $P(O^T | s_r^T)$ možemo zapisati kao umnožak svih b_{jk} definiranih u odnosu na skriveno i odgovarajuće vidljivo stanje:

$$P(O^T | s_r^T) = \prod_{t=1}^T P(O(t)|s(t))$$

Iz gore navedenih izraza sada možemo vjerojatnost slijeda T vidljivih stanja O^T zapisati kao sumu vjerojatnosti generiranja pojedinog prijelaza pomnoženog sa vjerojatnošću emitiranja odgovarajućeg promatranog vidljivog stanja:

$$P(O^T) = \sum_{r=1}^{r_{\max}} \prod_{t=1}^T P(O(t)|s(t)) * P(s(t)|s(t-1))$$

Ova vjerojatnost se može izračunati direktno iz izraza, ali postoje efikasniji načini izračuna poput unaprijednog pristupa koji koristi rekurziju.

Postupak unaprijednog pristupa („Forward algorithm“):

- Definicija varijable $\alpha_t(i) = P(o_1, o_2, \dots, o_t, o_t, q_t = i | \Theta)$, gdje je $\alpha_t(i)$ vjerojatnost ostvarivanja parcijalnog slijeda (o_1, o_2, \dots, o_t) uz uvjet da se skriveni Markovljev model nalazi u vremenu t u stanju $q_t = i$
- Algoritam:
 - Inicijalizacija $\alpha_1(i) = \pi_i b_i(o_1)$
 - Indukcija $\alpha_{t+1}(j) = [\sum_{i=1}^N \alpha_t(i) \cdot b_j(o_{t+1})]$
 - Kraj $P(O|\theta) = \sum_{i=1}^N \alpha_r(i)$

Analogno unaprijednom pristupu, postoji i unazadni pristup („Backward algorithm“):

- Definicija varijable $\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T | q_t = i, \theta)$ gdje je $\beta_t(i)$ vjerojatnost ostvarivanja parcijalnog slijeda $(o_{t+1}, o_{t+2}, \dots, o_T)$ uz uvjet da se skriveni Markovljev model nalazi u trenutku t u stanju $q_t=i$

- Algoritam:
 - Inicijalizacija $\beta_T(i) = 1$
 - Indukcija $\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$ uz
 $1 \leq i \leq N, t = T - 1, T - 2, \dots, 1$

3.2. Problem dekodiranja

Sljedeći problem prilikom primjene skrivenih Markovljevih modela je problem dekodiranja. Nastoji se odrediti skriveni slijed stanja koji je najvjerojatnije generirao dobiveni slijed opservacijskih simbola. Rješava se *Viterbi* algoritmom [6]:

- Inicijalizacija $\delta_1(i) = \pi_i b_i(o_1), \varphi_1(i) = 0$
- Rekurzija $\delta_t(j) = \max[\delta_{t-1}(i) a_{ij}] \cdot b_j(o_t)$
 $\varphi_t(j) = \arg \max[\delta_{t-1}(i) a_{ij}], \quad 2 \leq t \leq T, 1 \leq N$
- Prekid $p^{max}(q|O, \theta) = \max[\delta_T(i)], q_T^{max} = \arg \max[\delta_T(i)]$
- Izlaz -slijed $q_t^{max} = \varphi_{t+1}(q_{t+1}^{max}), t = T - 1, T - 2, \dots, 1$

3.3. Problem učenja

Prilikom slijeda opservacija $\{O = o_1, o_2, \dots, o_T\}$ osnovni zadatak Markovljevih modela je određivanje parametara a_{ij} i b_{jk} koristeći skup uzoraka. Za rješavanje ovog problema koristi se *Baum-Welch* algoritam. Osnova je iterativno ažuriranje parametara modela s ciljem aproksimacije parametara koji su najvjerojatnije generirali slijed opservacija [6].

Prvo je potrebno definirati novu varijablu $\gamma_{ij}(t)$ koja predstavlja vjerojatnost prijelaza između $q_i(t-1)$ i $q_j(t)$ uz uvjet da je model generirao cijeli slijed O^T iz čega slijedi izraz:

$$\gamma_{ij}(t) = \frac{\alpha_i(t-1) a_{ij} b_{jk} \beta_j(t)}{P(O^T | \lambda)}$$

gdje je $P(O^T | \lambda)$ vjerojatnost da je model generirao slijed O^T . Očekivani broj prijelaza iz stanja s_i u stanje s_j promatrano na intervalu vremena T iznosi $\sum_t^T \gamma_{ij}(t)$, a očekivani broj

prijelaza iz bilo kojeg drugog stanja u s_j iznosi $\sum_t^T \sum_i \gamma_{ij}(t)$. Procjena a_{ij} , označena kao \hat{a}_{ij} se računa kao omjer očekivanog prijelaza stanja iz s_i u stanje s_j i ukupnog broja prijelaza u to stanje:

$$\hat{a}_{ij} = \frac{\sum_t^T \gamma_{ij}(t)}{\sum_t^T \sum_i \gamma_{ij}(t)}$$

Analogno gore navedenom, dobiva se poboljšana vrijednost parametra b_{jk} , računanjem omjera između očekivanog broja posjećivanja stanja s_j iz kojeg se emitira opservacija O_k i očekivanog ukupnog broja posjećivanja stanja s_j sa emitiranjem bilo koje opservacije:

$$\hat{b}_{jk} = \frac{\sum_{t:s(t)=s_k}^T \sum_l \gamma_{jl}(t) | O_k}{\sum_t^T \sum_l \gamma_{jl}(t)}$$

- *Baum- Welch* algoritam:
 - Inicijalizacija trenutnog HMM modela $\theta = \{S, V, \pi, \mathbf{A}, \mathbf{B}\}$
 - Procjena novog HMM modela iz trenutnog: $\theta^{\wedge} = \{S, V, \pi^{\wedge}, \mathbf{A}^{\wedge}, \mathbf{B}^{\wedge}\}$
 - Ako je $P(O|\theta^{\wedge}) - P(O|\theta) < \delta$ gdje je δ prethodno definirana granica, ažuriraj trenutni model θ^{\wedge} i vrati se na drugi korak ako ne vrijedi $P(O|\theta^{\wedge}) = P(O|\theta)$

4. Realizacija modela

Kako bismo realizirali model ponašanja vremena pomoću prethodno opisanih skrivenih Markovljevih modela, potrebno je odrediti sve potrebne matrice vjerojatnosti.

Budući da se gradi model za godišnje doba proljeće, biti će uzeti podaci tlaka zraka, temperature, vlage zraka te vremena od prethodne tri godine.

Od prethodne dvije godine (2016., 2015.), podaci od proljeća će biti korišteni za izgradnju modela, dok će podaci od treće godine (2014.) poslužiti za testiranje dobivenog modela.

Ovaj model će imati tri skrivena stanja:

- Sunčano vrijeme
- Oblačno vrijeme
- Kišovito vrijeme

Svaki promatrani parametar (temperatura, vlaga, tlak) će se podijeliti na tri intervala. Svaki interval će najviše indicirati na jednu od tri skrivena stanja.

Time vrijednosti temperature dijelimo na sljedeći način:

- TL: $\{[T_{\min}] - 9 \text{ }^\circ\text{C}\}$, gdje je najviše kišovito vrijeme
- TM: $\{10 - 17 \text{ }^\circ\text{C}\}$, gdje je najviše oblačno vrijeme
- TH: $\{18 - [T_{\max}] \text{ }^\circ\text{C}\}$, gdje je najviše sunčano vrijeme

Vrijednosti vlage dijelimo na sljedeći način:

- ML: $\{[M_{\min}] - 52\%\}$, gdje je najviše sunčano vrijeme
- MM: $\{53 - 81 \%\}$, gdje je najviše oblačno vrijeme
- MH: $\{82 - [M_{\max}] \%\}$, gdje je najviše kišovito vrijeme

Vrijednosti tlaka zraka dijelimo na sljedeći način:

- PL: $\{[P_{\min}] - 1010 \text{ hPa}\}$, gdje je najviše kišovito vrijeme
- PM: $\{1011 - 1017 \text{ hPa}\}$, gdje je najviše sunčano vrijeme
- PH: $\{1017 - [P_{\max}] \text{ hPa}\}$, gdje je najviše oblačno vrijeme

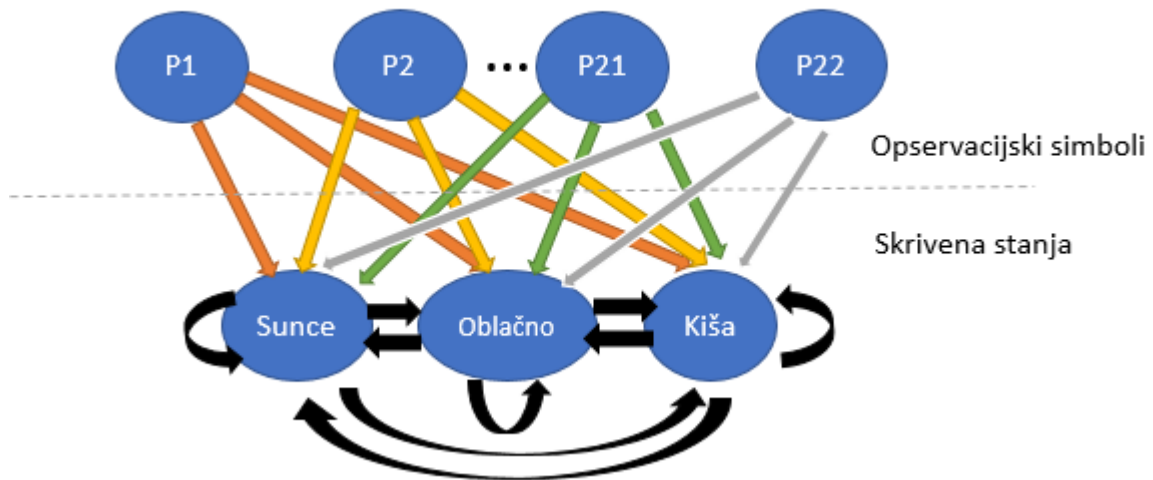
Ako se prema ovoj podjeli označe svi parametri te se uzmu u obzir sve kombinacije, tada bi ukupno trebalo biti 27 mogućih opservacija.

Međutim, proučavanjem svih kombinacija kroz prikupljene podatke, postaje jasno da se neke kombinacije nikada ne pojavljuju, što znači da se određene opservacije mogu izbaciti iz modela.

Time se dobivaju 22 moguća opservacijska simbola:

- 1) PL/TL/ML
- 2) PL/TL/MH
- 3) PL/TM/ML
- 4) PL/TM/MM
- 5) PL/TM/MH
- 6) PL/TH/ML
- 7) PL/TH/MM
- 8) PM/TL/ML
- 9) PM/TL/MM
- 10) PM/TM/ML
- 11) PM/TM/MM
- 12) PM/TM/MH
- 13) PM/TH/ML
- 14) PM/TH/MM
- 15) PH/TL/ML
- 16) PH/TL/MM
- 17) PH/TL/MH
- 18) PH/TM/ML
- 19) PH/TM/MM
- 20) PH/TM/MH
- 21) PH/TH/ML

22) PH/TH/MM



Slika 4. 1 Grafički prikaz modela

Na Slici 4.1 je prikazan grafički prikaz modela sa 22 moguća opservacijska simbola i 3 skrivena stanja.

Nakon određivanja broja skrivenih stanja i broja opservacijskih simbola, potrebno je odrediti matricu prijelaznih vjerojatnosti stanja (matrica **A**) i matricu vjerojatnosti opservacija (matrica **B**).

Matrica **A** će određivati vjerojatnosti prijelaza iz jednog stanja u drugo. Budući da se promatraju tri stanja, matrica **A** će imati dimenzije 3x3.

Kako bi se odredili parametri unutar matrice **A**, potrebno je detaljno proučiti izvatke vremenskih uvjeta protekle 2 godine u proljeću te izračunati vjerojatnosti prijelaza stanja iz jednog vremenskog uvjeta u drugi (dotične vjerojatnosti se mogu izračunati provjerom na koji način se mijenjalo vrijeme iz dana u dan iz preuzetih podataka).

Od ukupno 120 dana proljeća preuzetih od protekle dvije godine, detaljnijom analizom vidljivo je sljedeće:

- 18 dana je bilo kišovito vrijeme
- 62 dana je bilo sunčano vrijeme
- 40 dana je bilo oblačno vrijeme

Kao i očekivano, budući da se radi o proljeću, najviše dana je sunčano vrijeme.

Analizirajući dane kišovito vremena, može se primijetiti kako:

- Kišovito vrijeme prijelazi u sunčano tri puta
- Kišovito vrijeme prijelazi u oblačno vrijeme 14 puta
- Kišovito vrijeme prijelazi u kišovito vrijeme jedanput

Primijeni li se analiza na dane sunčanog vremena slijedi:

- Sunčano vrijeme prijelazi u sunčano vrijeme ukupno 35 puta
- Sunčano vrijeme prijelazi u oblačno vrijeme 17 puta
- Sunčano vrijeme prijelazi u kišovito vrijeme 10 puta

Konačno, analizom dana oblačnog vremena:

- Oblačno vrijeme prijelazi u sunčano vrijeme 21 puta
- Oblačno vrijeme prijelazi u oblačno vrijeme 12 puta
- Oblačno vrijeme prijelazi u kišovito vrijeme 7 puta

Iz dobivenih podataka, matricu **A** je jednostavno prikazati na sljedeći način:

	Sunce	Oblačno	Kiša
Sunce	0.58	0.29	0.13
Oblačno	0.525	0.3	0.175
Kiša	0.167	0.78	0.053

Slika 4. 2 Matrica A

Nakon matrice **A**, potrebno je odrediti matricu vjerojatnosti opservacija **B**.

Budući da ovaj model ima ukupno 22 moguće opservacije, odnosno opservacijska simbola te ukupno 3 stanja, matrica **B** bit će dimenzija 3x22.

Nadalje, daljnjom analizom svih 120 dana proljeća, kroz protekle dvije godine, potrebno je odrediti vjerojatnosti pojave svakog stanja u odnosu na pojavu pojedinog opservacijskog simbola.

Naravno, moguće je da neki opservacijski simbol uvijek daje isti vremenski uvjet, ili da je vjerojatnost nekog opservacijskog simbola nula.

Što se više podataka koristi prilikom gradnje modela, to će model davati bolje rezultate na kraju. Upravo iz tog razloga su uzete dvije godine, umjesto samo jedne.

Ukoliko se označe opservacijski simboli brojevima od 1 do 22, matrica **B** ima sljedeći oblik:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Sunce	0	0	0.5	0.17	0	1	0	1	0	0.67	0.33	0	0.94	0.4	0.5	0	0	0.69	0	0	0.79	1
Oblačno	0	0	0.5	0.67	0	0	1	0	0.33	0.33	0.67	0.25	0.06	0.6	0.5	0.5	0	0.31	0.75	0	0.21	0
Kiša	1	1	0	0.17	1	0	0	0	0.75	0	0	0.75	0	0	1	0.5	1	0	0.25	1	0	0

Slika 4. 3 Matrica B

Nakon određivanja stanja, opservacijskih simbola, matrice **A** i matrice **B**, još je preostalo odrediti vektor inicijalnih stanja **II**, koji je iz trenutno poznatih podataka jednostavno odrediti kao ukupnu vjerojatnost da je inicijalno vrijeme sunčano, oblačno ili kišovito vrijeme.

Budući da je od 120 dana podataka, ukupno 62 dana sunčano vrijeme, početna vjerojatnost da je sunčano vrijeme je 0.52.

Od 120 promatranih dana, ukupno 40 dana je oblačno vrijeme, pa slijedi da je početna vjerojatnost da je oblačno vrijeme 0.33.

Konačno, od 120 dana, 18 dana je kišovito vrijeme, pa slijedi da je početna vjerojatnost da je kišovito vrijeme 0.15.

Iz navedenoga, vektor **II** ima sljedeći oblik:

$$\mathbf{II} = [0.52 \ 0.33 \ 0.15]$$

Za implementaciju modela koristiti će se programski paket Matlab za jednostavnije primjere te Python skripte za implementaciju modela vremenskih pojava za proljeće i *Viterbi* algoritma.

Budući da Matlab ima ugrađene HMM (*Hidden Markov model*) funkcije koje omogućuju stvaranje modela jednostavnim unosom potrebnih parametara i zatim poduzimanje daljnjih akcija poput učenja modela ukoliko nam tablica vjerojatnosti prijelaza opservacija nije poznata ili je nepotpuna.

Iz tog razloga Matlab je poslužio provjeri matrice prijelaza opservacija. Matricu prijelaza opservacije potrebno je i ručno prilagoditi, budući da algoritam učenja ne daje uvijek najbolje rezultate.

S obzirom da je model potrebno implementirati da radi u stvarnom vremenu te da je potrebno konačni rezultat ispisivati na web stranicu, model je potrebno implementirati u programskom jeziku s kojim je to moguće.

Izabran je Python radi jednostavnosti i kompatibilnosti sa Linux operacijskim sustavom te raznim web serverima.

5. Meteorološka stanica

Prethodno su opisane metode mjerenja i prognoziranja vremena. Nekoć su metode mjerenja vremena bile skup i dugotrajan proces koji se ponajviše oslanjao na opažanja.

Danas su mjerenja vremenskih parametara znatno olakšana razvojem tehnologije. Postoje mnogobrojna gotova rješenja, odnosno male meteorološke stanice koje mogu poslužiti za osobna istraživanja vremenskih uvjeta.

Ako se ipak želi samostalno odabrati senzore te sastaviti vlastitu meteorološku stanicu, kao koristan uređaj se pokazao Raspberry Pi, malo računalo koje je pronašlo primjenu u mnogim projektima.

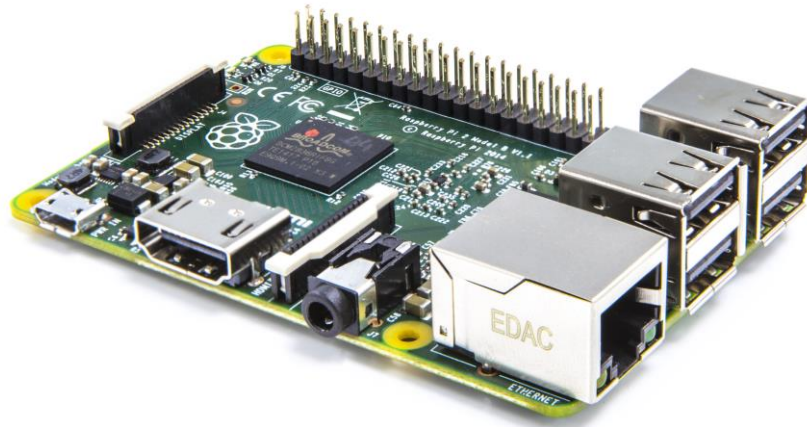
Korištenjem prethodno opisanih skrivenih Markovljevih modela stvoren je model koji na temelju izmjerenih parametara temperature, vlage i tlaka zraka predviđa je li vrijeme trenutno sunčano, oblačno ili kišovito. Meteorološka stanica će poslužiti testiranju modela u stvarnom vremenu kako bi saznali u kojoj mjeri model uistinu prepozna trenutno vrijeme.

5.1. Komponente

Kako bi se sastavila meteorološka stanica koja će prikupljati podatke iz okoline, potreban je Raspberry Pi, senzori za mjerenje temperature, vlage i tlaka te pomoćne komponente poput tipkovnice, miša, eksperimentalne pločice, otpornika, kratkospojnika, HDMI i Ethernet kabela.

5.1.1. Raspberry Pi

Kao osnovna komponenta meteorološke stanice koristit će se Raspberry Pi 2 (RPI2). RPI2 je druga generacija Raspberry Pi računala koji je izlaskom na tržište postao jedan od najpopularnijih ugradbenih računala.



Slika 5. 1 Raspberry Pi 2

RPI2 (Slika 5.1) sastoji se od:

- SoC: Broadcom BCM2836
- CPU: 900 MHz quad-core ARM Cortex-A7
- Memorija: 1 GB RAM
- 4 USB 2.0 priključka
- 26 GPIO pinova
- HDMI priključak
- Ethernet priključak
- 3.5 mm audio priključak
- MicroSD utor

RPI2 ima mogućnost rada na velikom broju Linux distribucija, ali i na Windowsu 10.

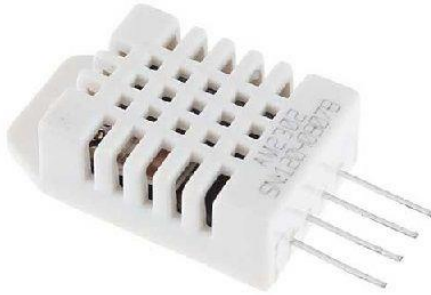
Iako ima brojne mogućnosti operacijskih sustava, odabrana je *Linux* distribucija *Raspbian*, budući da je ova distribucija namijenjena upravo za RPI i dovoljna je za sve potrebe ovog rada.

5.1.2. Senzori

Za mjerenje temperature, vlage i tlaka odabrani su sljedeći senzori:

- *DHT22* za mjerenje vlage i temperature zraka
- *BMP180* za mjerenje tlaka zraka

DHT22 je jednostavan senzor koji se sastoji od kapacitivnog senzora vlage i termistora.



Slika 5. 2 DHT22

Glavne karakteristike ovog senzora su sljedeće [8]:

- Niska cijena
- 3 -5 V napajanje
- Raspon temperature -40 do 125 °C sa točnošću ± 0.5 °C
- Raspon vlage 0 – 100 % sa točnošću 2 -5 %
- Frekvencija uzorkovanja: 0.5 Hz

BMP180 je senzor koji mjeri tlak, nadmorsku visinu i temperaturu te se spaja preko *I2C* sabirnice sa 7-bitnom adresom 0x77.



Slika 5. 3 BMP180

Glavne karakteristike ovog senzora su sljedeće [9]:

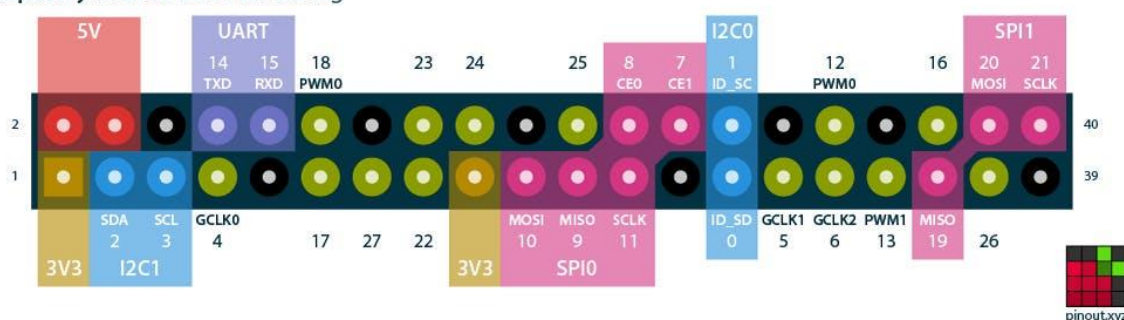
- Niska cijena
- Mala potrošnja energije
- Raspon tlaka: 300-1100 hPa sa točnošću ± 0.03 hPa
- Raspon temperature: -40 do 80 °C sa točnošću ± 2 °C
- Raspon nadmorske visine: -500 do 9000 m sa točnošću ± 0.25 m
- 3-5 V napajanje

5.2. Implementacija

Nakon prikupljanja svih potrebnih komponenti za sastavljanje meteorološke stanice, potrebno je ispravno povezati komponente na RPI2. To ćemo ostvariti pomoću kratkospojnika i eksperimentalne pločice. Senzor *DHT22* ima 4 pina, od kojih je prvi potrebno spojiti na napajanje 3.3 V, drugi na GPIO pin na RPI2, treći se ne koristi, a četvrti je potrebno spojiti na GND, odnosno uzemljenje [10].

Kako bi se znalo na koji način se pinovi na RPI2 spajaju te što pojedini pin predstavlja, potrebno je proučiti raspored pinova (Slika 5.4).

Raspberry Pi GPIO BCM numbering



Slika 5. 4 RPI2 raspored pinova

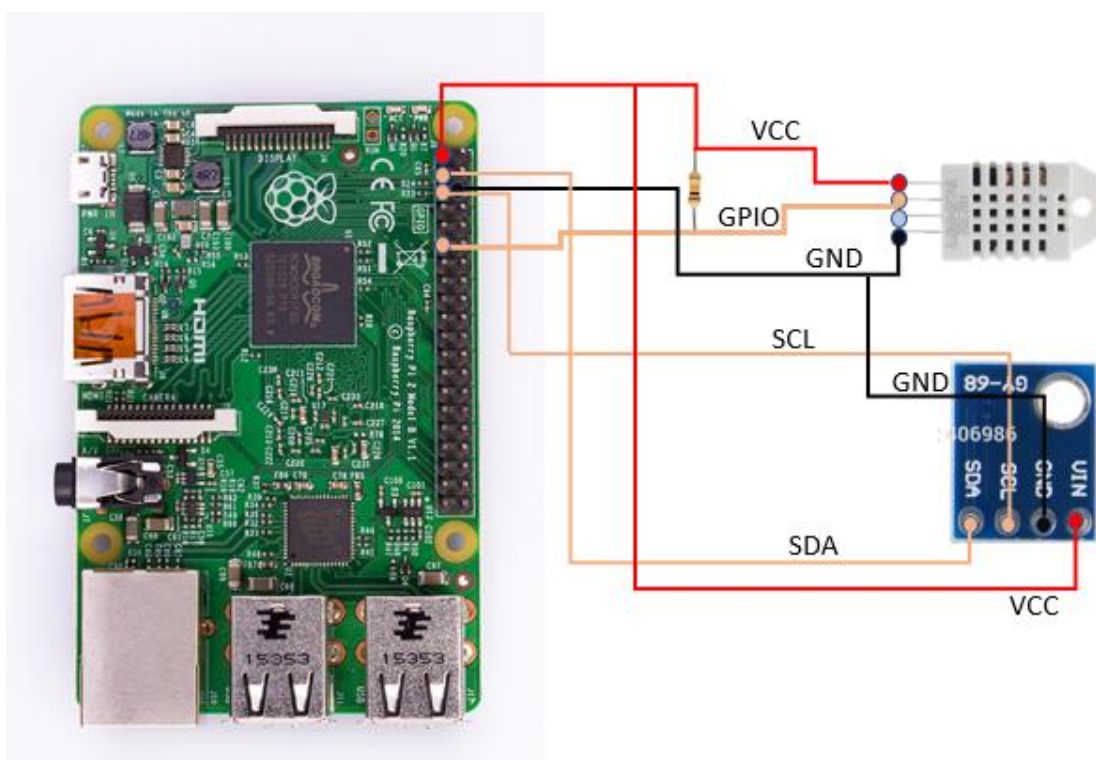
Prema Slici 5.4, vidljivo je da sva zelena polja predstavljaju GPIO pinove, tako da je potrebno spojiti drugi pin sa senzora na bilo koji GPIO pin na RPI2, u ovom slučaju spojiti će se na pin 17.

Broj pina na koji spajamo senzor je važan jer je potreban kako bi se pokrenula skripta koja će prikazati rezultate mjerenja senzora.

S druge strane, senzor *BMP180*, spaja se preko *I2C* sabirnice te također kao *DHT22* ima 4 pina. RPI2 ima dva pina koji odgovaraju *I2C* protokolu, jedan predstavlja prijenos podataka (*SDA*), dok drugi predstavlja takt (*SCL*). Prema Slici 5.4, vidljivo je da su ta dva pina na RPI2 pin 2 (*SDA*) i pin 3 (*SCL*). Ostala dva pina ovog senzora su pin za napajanje 3.3 V i pin za uzemljenje (*GND*) [11].

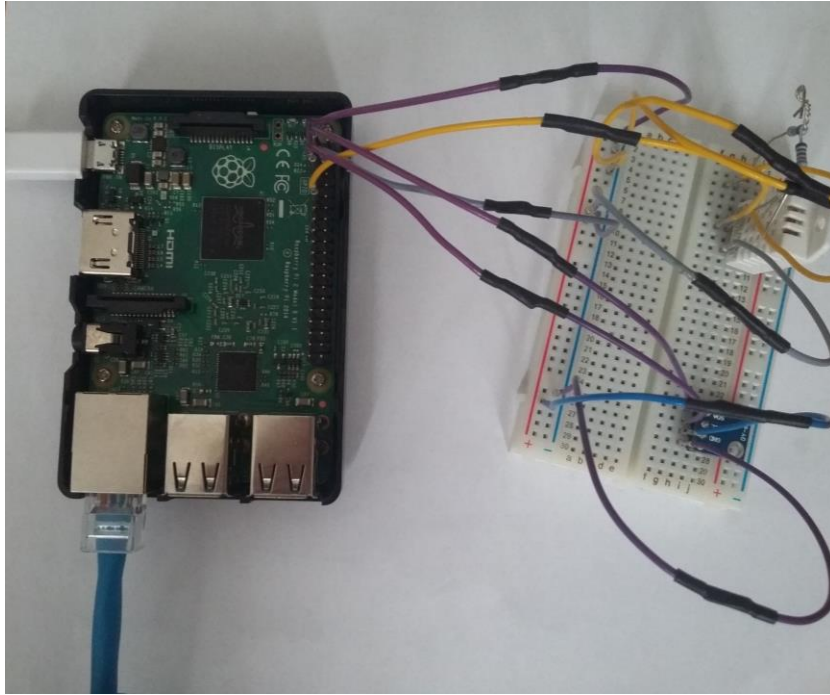
Nakon što je poznato koji se pinovi koriste i kako se spajaju potrebno je sve povezati pomoću eksperimentalne pločice.

Način spajanja RPI2 sa sensorima te koji su pinovi pritom korišteni prikazano je na Slici 5.5.

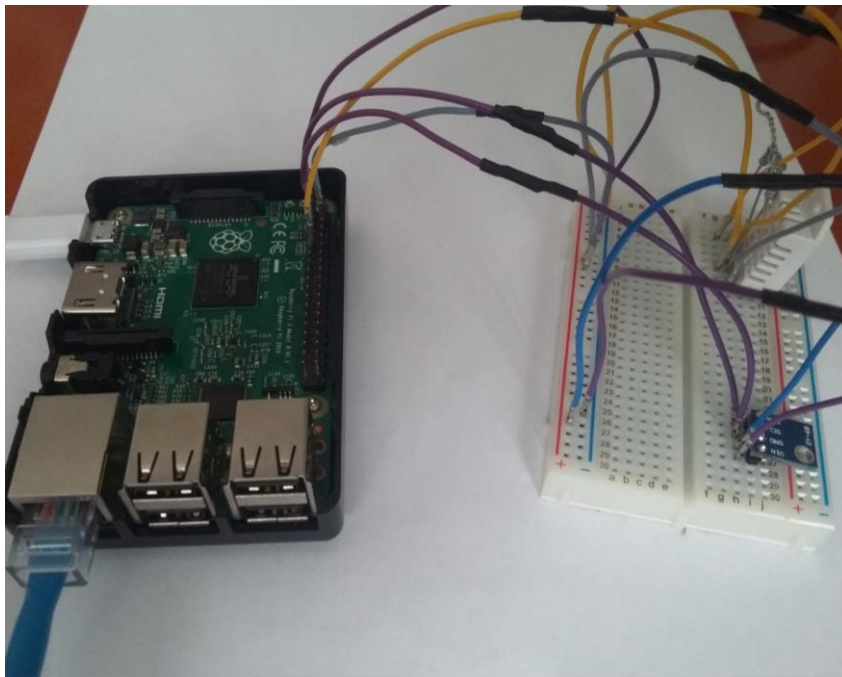


Slika 5. 5 Shema spajanja senzora i RPI2

Konačni izgled meteorološke stanice prikazan je na slikama 5.6 i 5.7.



Slika 5. 6 Izgled meteorološke stanice



Slika 5. 7 Izgled meteorološke stanice

Nakon uspješnog spajanja svih komponenata potrebno je provjeriti ispravnost senzora.

5.3. Testiranje

Kako bi se nastavilo sa implementacijom meteorološke stanice i testiranjem modela detekcije vremena, potrebno je provjeriti ispravnost senzora.

Komunikacija između računala i RPI2 odvija se putem *SSH* protokola. Programi korišteni za komunikaciju su, *WinSCP* za prijenos skripti sa računala na RPI2 te *Putty* za udaljeni pristup RPI2 i zadavanje naredbi.

Za povezivanje RPI2 i računala uz ove programe, dovoljno je spojiti RPI2 i računalo ETHERNET kabelom.

Nakon toga potrebno je samo poznavati IP adresu RPI2 ili dodijeljeno ime RPI2 (u ovom slučaju dodijeljeno ime je nelapi, a IP adresa 169.254.226.191).

Nakon što je komunikacija uspješno uspostavljena moguće je provesti instalaciju i postavljanje potrebnih postavki i alata potrebnih za ispravno funkcioniranje senzora.

Za prikupljanje podataka sa senzora korištene su *Adafruit* biblioteke. *Adafruit* biblioteke omogućuju testiranje senzora i ispis izmjerenih parametara na jednostavan način u terminalu. Također, *Adafruit* biblioteke imaju raširenu mrežu podrške za rješavanje raznih problema prilikom instalacije i upotrebe, što uvelike olakšava i ubrzava proces pokretanja i korištenja senzora.

Za potrebe testiranja senzora *DHT22*, potrebno je instalirati biblioteku *ADAFRUIT_PYTHON_DHT* koja sadrži primjere te omogućuje pokretanje očitavanja senzora jednostavno naredbom iz terminala te ispis rezultata u terminal.

Nakon uspješne instalacije i pozicioniranja u stvoreni direktorij biblioteke, pokretanje senzora se omogućuje sljedećom naredbom:

```
sudo python AdafruitDHT.py 22 17
```

Broj 22 u naredbi predstavlja model korištenog senzora, dok je broj 17 broj pina na RPI2 na koji je senzor spojen.

Ukoliko je sve ispravno instalirano i spojeni, rezultat izvođenja ove naredbe je ispis rezultata mjerenja temperature i vlage (Slika 5.8).

```
pi@nelapi:~/Metstanica/library/Adafruit_Python_DHT/examples $ python AdafruitDHT.py 22 17
Temp=24.8* Humidity=41.1%
```

Slika 5. 8 Rezultat mjerenja senzorom DHT22

Za potrebe testiranja senzora *BMP180* potrebno je instalirati python modul *smbus*, koji služi za komunikaciju *I2C* protokolom. RPI2 ima dva pina koja inicijalno služe kao *SCL* (takt) i *SDA*(podaci) koji se koriste prilikom komunikacije pomoću *I2C* protokola.

I2C protokol je potrebno omogućiti uz pomoć *raspi-config* opcije.

Kao i kod *DHT22*, i ovdje postoji *Adafruit* biblioteka koja sadrži jednostavan kod za testiranje *BMP180* senzora: *ADAFRUIT_PYTHON_BMP*.

Pozicioniranjem u ispravni direktorij, prvo je potrebno provjeriti ispravnost vidi li RPI2 senzor na ispravnoj adresi:

sudo i2cdetect -y 1

Prema Slici 5.9 vidljivo je kako se senzor nalazi na adresi 0x77, kao i očekivano.

```
pi@nelapi:~/Metstanica/library/Adafruit_Python_BMP $ sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
60:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
70:  -- -- -- -- -- -- 77
```

Slika 5. 9 Uspješna provjera detekcije senzora BMP180

Ako je provjera prošla uspješno, moguće je testirati senzor pozicioniranjem u direktorij unutar *ADAFRUIT_PYTHON_BMP*, koji sadrži skriptu *simpletest.py* te ju pokrenuti sljedećom naredbom:

sudo python simpletest.py

Ako je sve prošlo u redu, u terminalu bi se trebao ispisati rezultat mjerenja senzora, a to su vrijednosti temperature, tlaka, visine i tlaka na razini mora.

```
pi@nelapi:~/Metstanica/library/Adafruit_Python_BMP/examples $ python simpletest.py
Temp = 25.20 *C
Pressure = 99925.00 Pa
Altitude = 117.47 m
Sealevel Pressure = 99937.00 Pa
```

Slika 5. 10 Rezultat mjerenja senzorom BMP180

6. Primjena modela na RPI2

Nakon uspješne provjere spojenih senzora, potrebno je napisati skriptu kojom će se prikupljati podaci temperature, tlaka i vlage zraka. Ti podaci se zatim trebaju upotrijebiti u modelu koji će na temelju izmjerenih parametara odrediti o kojem se opservacijskom simbolu radi te nakon toga *Viterbi* algoritmom odrediti najvjerojatnije stanje vremena, odnosno je li trenutno vrijeme sunčano, oblačno ili kišovito.

Za početak, potrebno je napisati funkciju koja će očitavati podatke sa senzora:

```
def read_sensors():
    # čitaj stanje senzora DHT22
    sensor = Adafruit_DHT.DHT22
    pin = 17
    humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
    humidity = round(humidity, 2)
    temperature = round(temperature, 2)

    # čitaj stanje senzora BMP180
    bmp = BMP085.BMP085()
    pressure = bmp.read_pressure() / 100.0
    print("Novi podaci: " + str(humidity) + ", " + str(temperature) + ", " +
          str(pressure))

    return humidity, temperature, pressure
```

Nakon što je očitavanje potrebnih parametara uspješno implementirano, slijedi implementacija modela u kojem je potrebno unijeti prethodno navedene matrice **A**, **B**, **Π**, moguća stanja te niz opservacijskih simbola.

Budući da se opservacijski simboli prate iz dana u dan, svakim mjerenjem taj niz postaje za jedan opservacijski simbol duži.

Kao početni opservacijski simbol, kako bi predviđanje vremenske pojave bilo što vjerojatnije, zadan je niz opservacijskih simbola koji odgovaraju uvjetima od nekoliko prethodnih dana.

Svaki idući opservacijski simbol se dodaje na kraj niza.

Kako bi se proveo *Viterbi* algoritam i saznalo o kojem vremenskom stanju se trenutno radi, potrebno je prvo odrediti koji opservacijski simbol odgovara trenutno izmjerenim parametrima. To će se provjeriti unutar funkcije modela nizom *if/else* operacija kako je prikazano u nastavku:

```
import viterbi as vt

def hmm_alg(observations, pressure, temperature, humidity):
    print(observations)
    obs =list(observations)
    print(observations)
    #obs = tuple(observations)
    print(observations)

    # određivanje kojem opservacijskom simbolu pripadaju trenutno izmjereni
    # parametri
    if pressure <= 1010.0 and temperature <= 9 and 53 <= humidity <= 81:
        obs.append(1)
    elif pressure <= 1010.0 and temperature <= 9 and humidity >= 82:
        obs.append(2)
    elif pressure <= 1010.0 and 10 <= temperature <= 17 and humidity <= 52:
        obs.append(3)
    elif pressure <= 1010.0 and 10 <= temperature <= 17 and 53 <= humidity <=
81:
        obs.append(4)
    elif pressure <= 1010.0 and 10 <= temperature <= 17 and humidity >= 82:
        obs.append(5)

    ...
```


Tek nakon što je određen i spremljen opservacijski simbol, moguće je dekodirati simbol i odrediti najvjerojatnije stanje koje je uzrokovalo dobiveni opservacijski simbol. To se postiže *Viterbi* algoritmom koji se pokreće sljedećom naredbom:

```
probability, row = vt.viterbi_alg(observations, states,  
start_probability, transition_probability, emission_probability)
```

U *Viterbi* algoritam kao parametre je potrebno predati niz opservacija, stanja, vektor početnih stanja Π , matricu vjerojatnosti prijelaza stanja \mathbf{A} te matricu vjerojatnosti prijelaza opservacija \mathbf{B} .

Kao izlaz *Viterbi* funkcija daje vjerojatnost dobivenog najvjerojatnijeg niza stanja te taj niz stanja. *Viterbi* funkcija prikazana je u nastavku:

```

def viterbi_alg(observations, states, start_probability, transition_probability,
emission_probability):

    trellis = [{}]

    path = {}

    #dodaj vjerojatnosti početka niza sa svakim od stanja
    for state in states:
        trellis[0][state] = start_probability[state] * emission_probability[state][observations[0]]
        path[state] = [state]

    # dodaj vjerojatnosti prijelaza sljedećeg stanja u svako moguće stanje
    for observations_index in range(1,len(observations)):
        #dodaj novi put
        trellis.append({})
        new_path = {}
        #za svako moguće stanje
        for state in states:

            (probability, possible_state) = max(
                [(trellis[observations_index-1][y0] * transition_probability[y0][state]
                * emission_probability[state][observations[observations_index]], y0) for y0 in
states])

            trellis[observations_index][state] = probability

            new_path[state] = path[possible_state] + [state]

        path = new_path

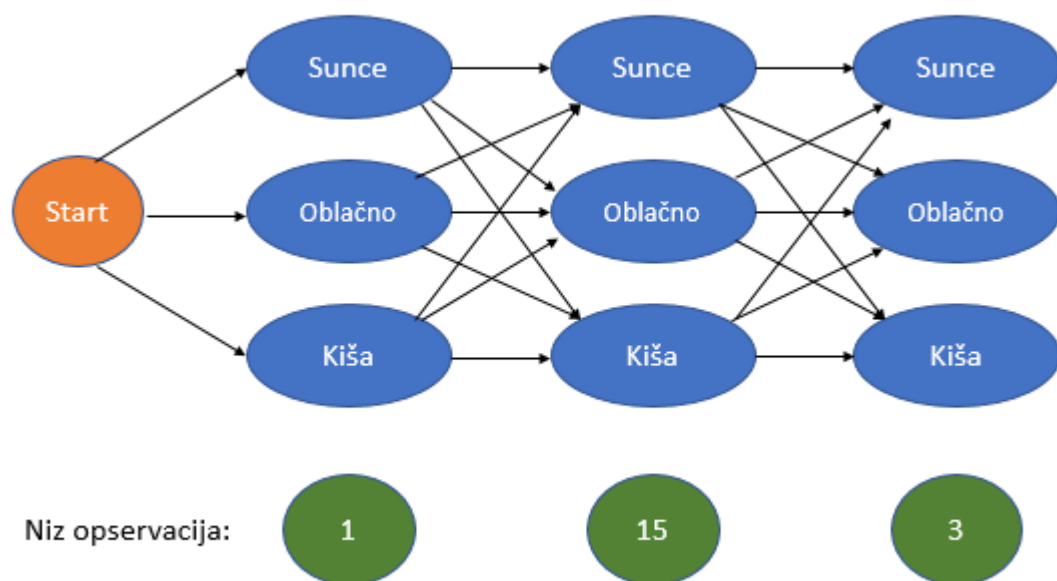
    #kreiraj listu sa svim dobivem putevima i njihovim vjerojatnostima i odaberi onu sa
najvećom vjerojatnosti te tu vrati
    (probability, state) = max([(trellis[len(observations) - 1][state], state) for state in states])

    return (probability, path[state])

```

Operacije koje se provode Viterbi algoritmom se mogu prikazati *Trellis* dijagramom, odnosno stablom. Prolaskom kroz dijagram algoritam traži slijed stanja s najvećom vjerojatnošću.

Jednostavniji prikaz *Trellis* dijagrama za model predviđanja vremena prikazan je na Slici 6.1.

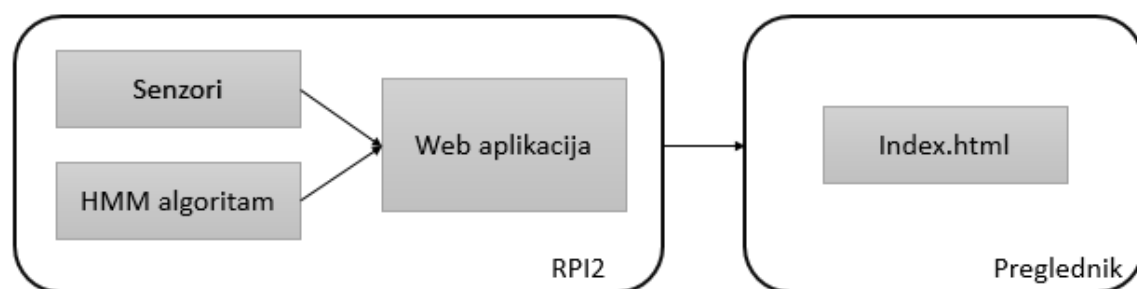


Slika 6. 1 Prikaz Trellis dijagrama

S obzirom da je algoritam kao opservacijski niz dobio niz opservacijskih simbola, kao povratni parametar se vraća niz najvjerojatnijih stanja.

Za detekciju trenutnog vremena potrebno je izdvojiti samo zadnje dobiveno stanje jer je upravo to stanje ono koje je trenutno promatrano i ono koje želimo prikazati na web stranici.

Za prikaz vremenske prognoze na webu, koristiti će se Jinja template alat za Python koji ima široku primjenu i jednostavnu upotrebu. Dijagram strukture sustava prikupljanja podataka sa senzora, izvođenje algoritma te prikaz na web stranici prikazan je na Slici 6.2.



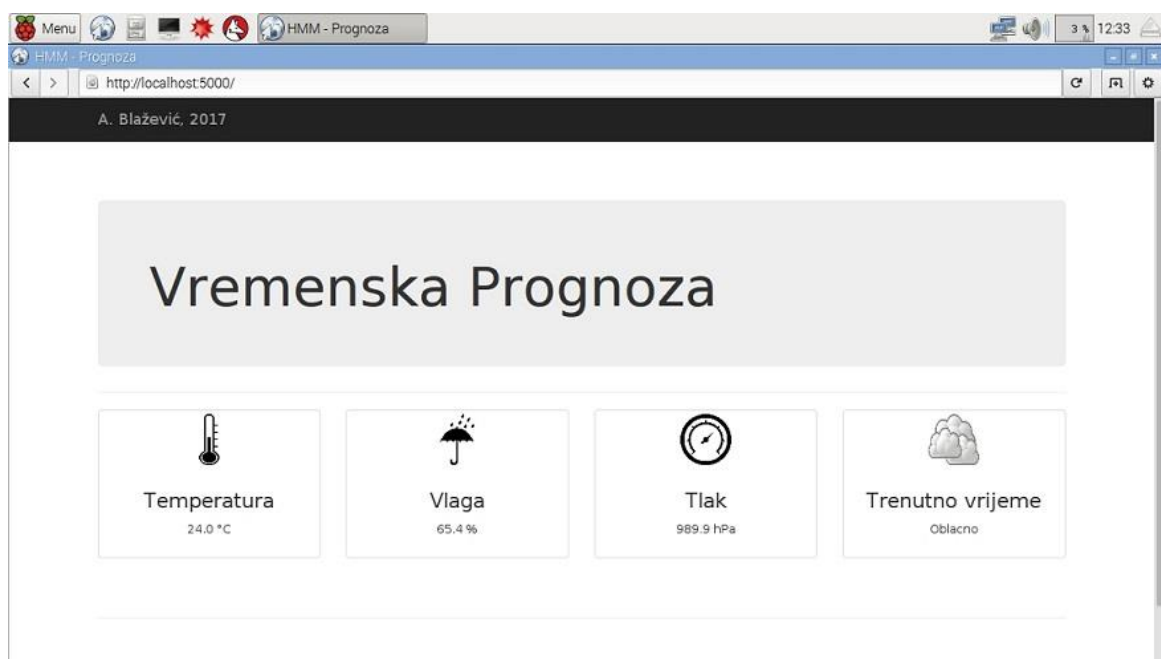
Slika 6. 2 Dijagram strukture sustava

Web stranica sadržavati će podatke o trenutnoj temperaturi, vlazi, tlaku zraka i prikaz rezultata modela.

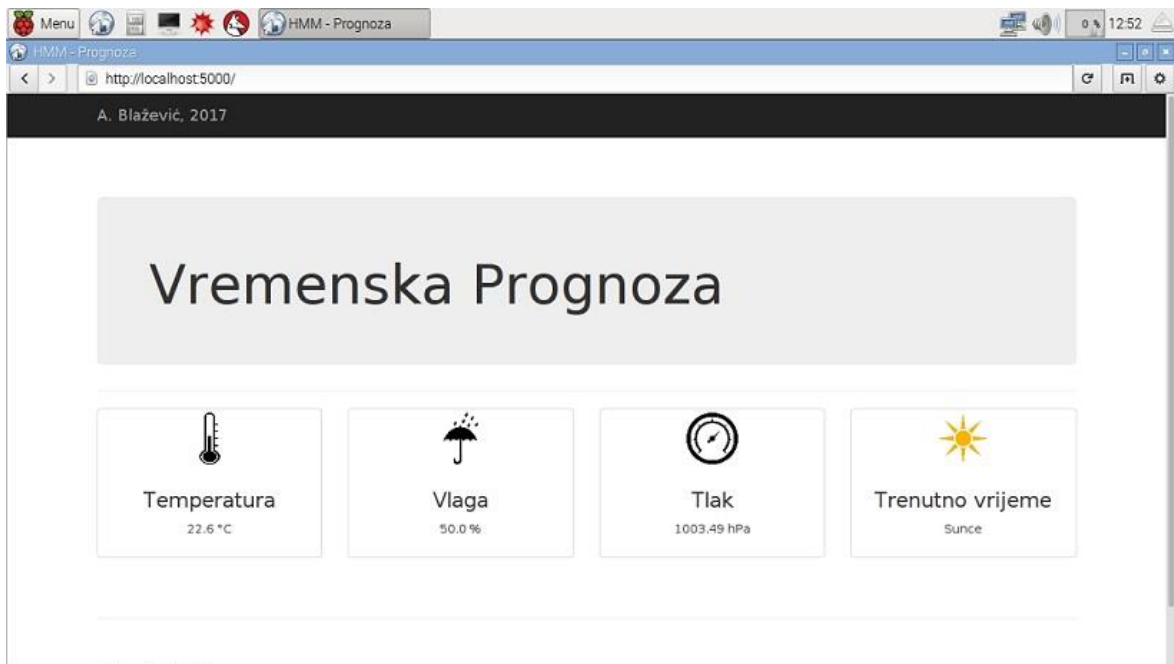
Ukoliko model kao izlaz da kao najvjerojatnije stanje sunčano, tada će se na web stranici prikazati to stanje kao trenutno vrijeme.

Isto tako, ukoliko je kao izlaz dobiveno oblačno ili kišovito vrijeme, to će stanje biti prikazano na stranici.

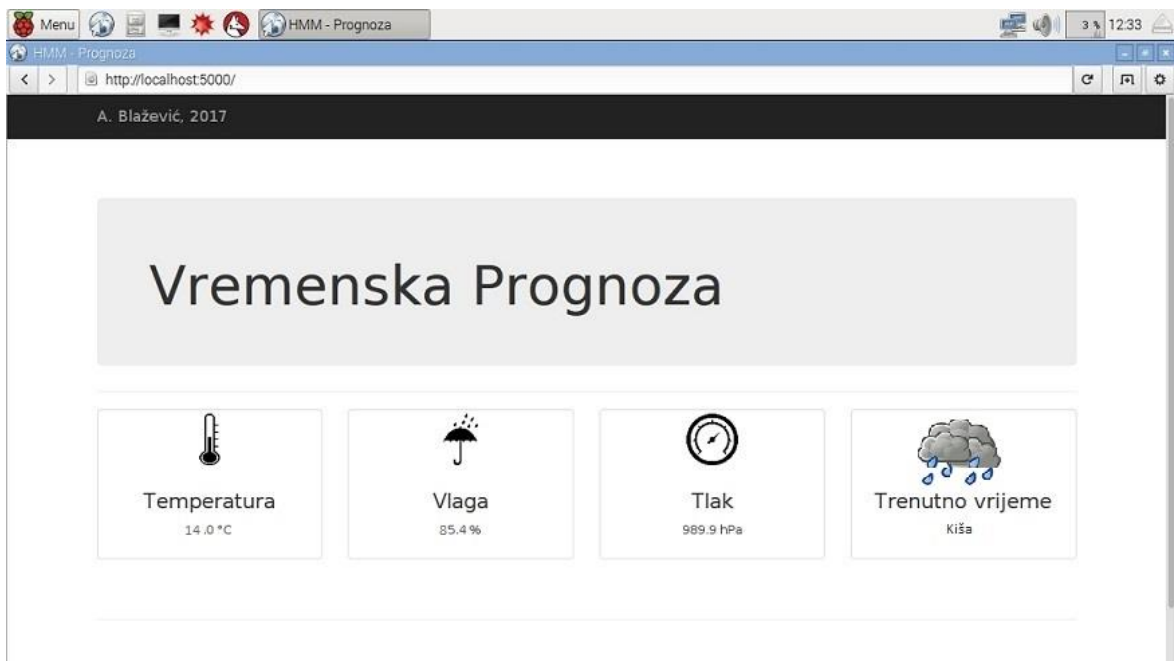
Izgled stranice prikazan je na slikama 6.3, 6.4 i 6.5.



Slika 6.3 Izgled web stranice kada je rezultat modela oblačno vrijeme



Slika 6. 4 Izgled web stranice kada je rezultat modela sunčano vrijeme



Slika 6. 5 Izgled web stranice kada je rezultat modela kišovito vrijeme

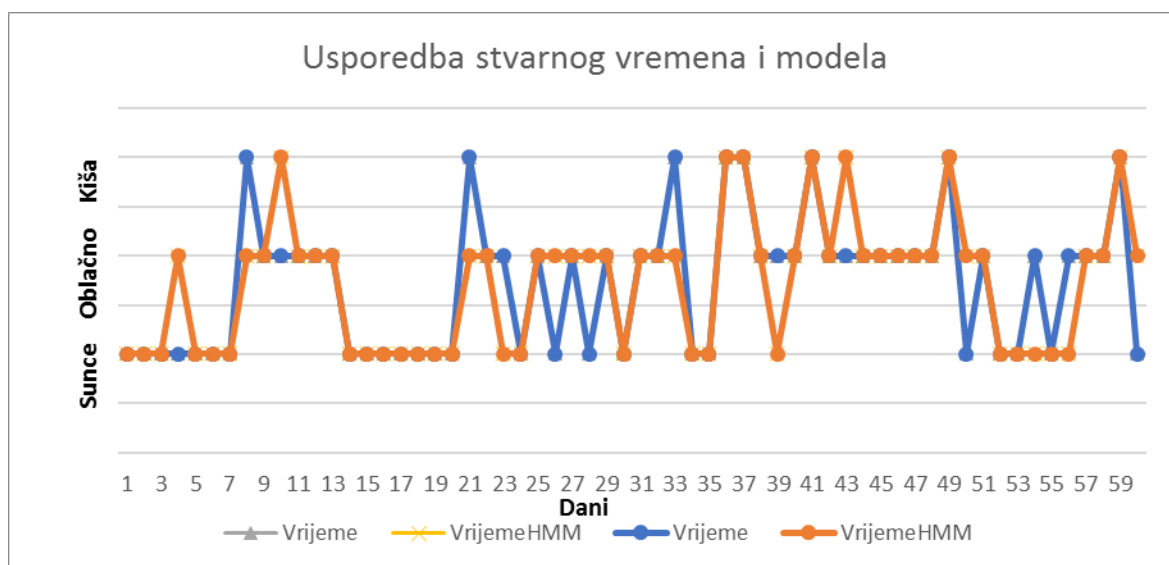
7. Mjerenja i rezultati

Testiranje modela je jednostavan proces s obzirom da se radi o vremenu te je dovoljno usporediti dobivene rezultate kroz nekoliko dana i stvarnu situaciju u prirodi.

Također, budući da je model građen na temelju dvije prethodne godine, moguće ga je testirati unosom niza opservacijskih simbola iz podataka godine koja nije korištena pri gradnji modela te na taj način provjeriti točnost modela.

Dovoljno je usporediti stvarna stanja vremena sa stanjima dobivenim na izlazu *Viterbi* algoritma.

Rezultat je prikazan na Slici 7.1.



Slika 7.1 Usporedba stvarnog vremena i vremena dobivenog modela

Iz Slike 7.1 vidljivo je u koliko točaka je model točno detektirao vremensku pojavu te u koliko točaka nije. Iz tih podataka moguće je odrediti postotak točnosti modela.

Budući da se radi o 60 dana, od kojih je model točno detektirao vrijeme ukupno 45 puta. Iz toga slijedi da je pogrešna detekcija uslijedila 15 puta.

Može se zaključiti kako model na uzorku od 60 ulaznih opservacijskih simbola, ima točnost 75% što je zadovoljavajući rezultat s obzirom da se radi o detekciji meteoroloških uvjeta i o malom broju mjerenih parametara.

Nakon simulacije mjerenja, slijedi testiranje modela u stvarnom vremenu kroz period vremena.

S obzirom da je model građen na temelju podataka očitanih svaki dan u podne, za očekivati je kako će model najbolje rezultate davati u sličnom vremenskom rasponu. Iz tog razloga mjerenja će svakodnevno biti izvođena oko podne radi dobivanja što boljih rezultata.

Tablica 1. Usporedba stvarnog vremena sa vremenom dobivenim modelom

Datum	05.06.	06.06..	07.06..	08.06.	09.06.
Stvarno vrijeme	Sunčano	Oblačno	Oblačno	Sunčano	Sunčano
Vrijeme predviđeno modelom	Sunčano	Oblačno	Sunčano	Sunčano	Oblačno

Datum	10.06.	11.06..	12.06..	13.06.	14.06.
Stvarno vrijeme	Oblačno	Sunčano	Sunčano	Sunčano	Oblačno
Vrijeme predviđeno modelom	Oblačno	Sunčano	Sunčano	Oblačno	Sunčano

U tablici 1 dan je prikaz stvarnog vremena i vremena koje je predvidio model kroz 10 dana.

Iz tablice 1 također je vidljivo kako je model točno predvidio vremenske uvjete u 6 od 10 slučajeva. Važno je napomenuti kako su se vremenske pojave prilikom gradnje modela podijelile u tri kategorije radi jednostavnosti, što znači da su neki vremenski uvjeti izostavljeni poput djelomično oblačno. Budući da je djelomično oblačno česta pojava u svakom godišnjem dobu, osobito proljeću, promatranjem uvjeta je problematično odlučiti kada će takav uvjet pripadati sunčanom, a kada oblačnom vremenu.

Mjerenja izvedena u ovih 10 dana su sadržavala često takve vremenske uvjete, pa je iz tog razloga i točnost modela nešto manja nego li sa testiranjem uvjeta jedne cijele godine, odnosno doba.

Može se zaključiti kako model daje zadovoljavajuće rezultate s obzirom da se radi o statičkim vjerojatnostima koje su unaprijed određene i nemaju mogućnost kasnijeg mijenjanja.

Model bi davao mnogo bolje rezultate kada bi se u kasnijim unaprjeđenjima koristile Gaussove mješavine umjesto statičkih vjerojatnosti.

Gaussova mješavina je model vjerojatnosti koji pretpostavlja da su sve podatkovne točke generirane od mješavina Gaussovih distribucija s nepoznatim parametrima. Zatim se određuje najveća vjerojatnost pripadanja neke točke pojedinoj distribuciji.

S obzirom da se korištenjem Gaussovih mješavina ne koriste unaprijed definirane statičke vjerojatnosti, za očekivati je kako bi se dobila bolja točnost modela.

Zaključak

U okviru ovog diplomskog rada istražili su se skriveni Markovljevi modeli te primjene. Jedna od mogućih primjena je u modeliranju meteoroloških uvjeta.

Analizirano je ponašanje vremenskih uvjeta i stanja atmosfere u različitim godišnjim dobima. Pokazan je utjecaj godišnjeg doba na vremenske uvjete i time na stvaranje individualiziranih modela ponašanja ovisno o dobu.

Analizirani su podaci vremenskih uvjeta od 3 godine. Podaci uzeti u obzir bili su temperatura, tlak, vlaga zraka te vrijeme. Vrijeme je podijeljeno u tri osnovne kategorije: sunčano, oblačno i kišovito.

Na temelju analiziranih podataka modeliran je model sa statičkim vjerojatnostima pomoću skrivenih Markovljevih modela koji opisuju i predviđaju ponašanje vremena u proljeću.

Izgrađena je meteorološka stanica na ugradbenom računaru Raspberry Pi 2. Meteorološka stanica se sastoji od senzora koji mjere temperaturu, vlagu i tlak te web stranice za prikaz podataka i rezultata modela.

Model je testiran sa podacima od godine koja nije bila korištena prilikom gradnje modela i mjerenjima u stvarnom vremenu. Točnost predviđanja modela se kreće od 60% do 75%. Model daje zadovoljavajuće rezultate s obzirom da se radi o modelu s unaprijed određenim statičkim vjerojatnostima.

Jedno od mogućih unaprjeđenja i poboljšanja modela je korištenje Gaussovih mješavina umjesto statičkih vjerojatnosti, čime bi se znatno poboljšala točnost modela u predviđanju meteoroloških pojava.

Literatura

- [1] <http://www.factsmag.com/weather-forecasting-explained/> , datum pristupa: ožujak, 2017
- [2] <http://www.fms-tivat.me/predavanja3god/PMIO9.pdf> , datum pristupa: travanj, 2017
- [3] [http://ww2010.atmos.uiuc.edu/\(Gh\)/guides/mtr/fcst/mth/prst.rxml](http://ww2010.atmos.uiuc.edu/(Gh)/guides/mtr/fcst/mth/prst.rxml) , datum pristupa: ožujak, 2017
- [4] <http://www.met.tamu.edu/class/atmo202/Forecastdir/forecast-stu.html> , datum pristupa: travanj, 2017
- [5] <https://www.wunderground.com/history/airport/LDZA/2016/4/10/DailyHistory.html>, datum pristupa: ožujak, 2017
- [6] Lajos, Jaroslav. Computer Modeling Using Hidden Markov Model Approach applied to the financial Markets. Oklahoma State University, 2011.
- [7] An Introduction to Hidden Markov Models, Rabiner L.R, Juang, B.H., IEEE članak, siječanj, 1986.
- [8] <https://learn.adafruit.com/dht/overview> , datum pristupa. ožujak, 2017.
- [9] <https://www.adafruit.com/product/1603> , datum pristupa. ožujak, 2017.
- [10] <https://www.hackster.io/adamgarbo/raspberry-pi-2-iot-thingspeak-dht22-sensor-b208f4> , datum pristupa: travanj, 2017.
- [11] <https://thepihut.com/blogs/raspberry-pi-tutorials/18025084-sensors-pressure-temperature-and-altitude-with-the-bmp180> , datum pristupa: travanj, 2017

Sažetak

Primjena skrivenih Markovljevih modela za modeliranje i predviđanje meteoroloških pojava

U ovom diplomskom radu istraženi su skriveni Markovljevi modeli te njihove moguće primjene. Analizirano je ponašanje meteoroloških prilika i stanja atmosfere u različitim godišnjim dobima. Prilikom analize, u obzir su uzeti podaci temperature, tlaka, vlage zraka te vremenskih uvjeta od prethodne 3 godine. Vremenski uvjeti podijeljeni su u 3 osnovne kategorije: sunčano, oblačno i kišovito. Na temelju analiziranih podataka određeni su parametri modela te je izgrađen model ponašanja meteoroloških uvjeta u proljeću. Korištenjem ugradbenog računala Raspberry Pi 2 izgrađena je meteorološka stanica za prikupljanje podataka o temperaturi, vlazi i tlaku kako bi se ti podaci koristili za izvođenje modela i prikaz rezultata u stvarnom vremenu na web stranici. Model je testiran sa podacima od godine koja nije bila korištena u izgradnji modela i mjerenjima u stvarnom vremenu. Točnost modela kreće se od 60% do 75%.

Ključne riječi: skriveni Markovljevi modeli, vrijeme, temperatura, tlak, vlaga, meteorološka stanica, web stranica

Summary

Modeling and prediction of weather states using Hidden Markov models

In this Master thesis Hidden Markov models and their possible applications are researched. The behavior of meteorological phenomena and states through different seasons is analysed. In the analysis, temperature, pressure, moisture and current weather from the past 3 years are taken into consideration. The weather is divided into 3 basic categories: sunny, cloudy and rainy. Based on the analysed data, the parameters of the model are determined and a model for spring is built. With the use of the embedded computer Raspberry Pi 2 a meteorological station is built to collect the data about temperature, pressure and moisture. Collected data is used in the real time modelling and display of the results on a web page. The model is tested with the data of the year which has not been used in the build of the model and real time metering. The accuracy of the model is between 60% and 75%.

Keywords: Hidden Markov models, weather, temperature, pressure, moisture, meteorological station, web page

Skraćenice

RPI2 – Raspberry Pi 2

GPIO – *General Purpose Input Output* (ulazno-izlazno sklopovlje)

TL – *Temperature Low* (niska temperatura)

TM – *Temperature Medium* (srednja temperatura)

TH – *Temperature High* (visoka temperatura)

ML – *Moisture Low* (niska vlaga)

MM – *Moisture Medium* (srednja vlaga)

MH – *Moisture High* (visoka vlaga)

PL – *Pressure Low* (niski tlak)

PM – *Pressure Medium* (srednji tlak)

PH – *Pressure High* (visoki tlak)

SDA – *Serial Data Line* (podatkovna linija)

SCL – *Serial Clock Line* (linija takta)

SSH – *Secure Shell* (mrežni protokol)

Privitak A. Tablice prikupljenih podataka

Tablica A.1 Tablica prikupljenih podataka 2016. godine od 16.03. do 14.05.

Tlak (hPa)	Temperatura (°C)	Vlaga(%)	Vrijeme	Opservacijski simbol
1027	5	76	Kiša	16
1028	11	44	Oblačno	18
1017	12	41	Sunčano	10
1017	14	41	Sunčano	10
1016	13	47	Sunčano	10
1016	8	71	Kiša	9
1011	9	71	Oblačno	9
1003	5	87	Kiša	2
1009	11	44	Oblačno	3
1017	10	43	Sunčano	10
1019	11	62	Kiša	19
1016	13	47	Oblačno	10
1013	11	71	Oblačno	11
1013	16	48	Oblačno	10
1018	17	52	Oblačno	18
1016	20	40	Sunčano	13
1014	18	46	Sunčano	13
1025	11	54	Oblačno	19
1020	16	45	Sunčano	18
1012	22	41	Sunčano	16
1010	22	35	Sunčano	6
1011	22	38	Sunčano	13
1010	19	52	Sunčano	14
1005	11	88	Kiša	5
1004	11	76	Oblačno	4
1011	11	71	oblačno	11
1014	16	52	Sunčano	10
1012	21	35	Sunčano	13
1009	22	41	Sunčano	6
1008	11	88	kisa	5
1013	18	49	Sunčano	13
1011	21	35	Sunčano	13
1011	24	36	Sunčano	13
1015	17	68	Sunčano	17

1021	10	82	Kiša	20
1027	14	51	Oblačno	18
1027	17	37	Sunčano	18
1015	18	42	Sunčano	13
1007	18	45	Sunčano	6
1006	8	62	Kiša	1
1008	10	43	Oblačno	3
1009	14	31	Sunčano	3
1005	14	72	Oblačno	4
1018	5	81	Oblačno	16
1021	10	71	Oblačno	19
1020	13	63	Oblačno	19
1013	12	82	Kiša	12
1013	13	88	Kiša	12
1013	16	59	Sunčano	11
1018	13	72	Kiša	19
1022	16	63	Oblačno	19
1019	19	56	Sunčano	18
1015	21	56	Oblačno	14
1016	20	56	Sunčano	14
1018	20	60	Oblačno	22
1016	14	88	Kiša	12
1011	14	94	Oblačno	12
1000	15	94	Kiša	7
998	18	64	Oblačno	7
1003	15	88	Kiša	5

Tablica A.2 Tablica prikupljenih podataka 2015. godine od 16.03. do 14.05.

Tlak (hPa)	Temperatura(°C)	Vlaga(%)	Vrijeme	Opservacijski simbol
1011	11	71	Oblačno	11
1030	11	58	Oblačno	19
1025	12	41	Oblačno	18
1025	12	35	Sunčano	18
1025	11	44	Sunčano	18
1017	16	29	Sunčano	10
1016	14	48	Oblačno	10
1019	11	50	Sunčano	18
1013	13	54	Sunčano	11
1007	12	72	Kiša	4
1007	17	68	Oblačno	4
1010	9	87	Kiša	2
1017	13	31	Sunčano	10
1013	12	38	Sunčano	10
999	17	39	Sunčano	3
1005	16	55	Sunčano	4
1013	8	71	Kiša	9
1013	13	41	Sunčano	10
1020	10	35	Sunčano	18
1013	10	50	Oblačno	10
1014	9	43	Sunčano	8
1019	9	34	Sunčano	15
1028	9	43	Oblačno	15
1026	11	41	Sunčano	18
1031	11	58	Oblačno	19
1027	16	42	Sunčano	18
1022	20	30	Sunčano	21
1030	20	46	Sunčano	21
1026	22	35	Sunčano	21
1029	14	38	Oblačno	18
1023	21	38	Sunčano	21
1016	24	29	Sunčano	13
1013	18	49	Sunčano	13
1020	7	81	Kiša	17
1023	11	47	Oblačno	18
1020	15	39	Sunčano	18
1023	17	45	Sunčano	18

1022	20	35	Sunčano	21
1016	21	38	Sunčano	13
1017	16	63	Oblačno	11
1016	19	49	Sunčano	13
1012	21	46	Oblačno	13
1008	21	49	Sunčano	6
1004	16	77	Oblačno	4
1021	16	39	Sunčano	18
1017	17	42	Sunčano	10
1012	17	52	Oblačno	10
1009	20	64	Oblačno	7
1013	20	68	Oblačno	14
1013	23	53	Oblačno	14
1012	25	47	Sunčano	13
1011	26	44	Sunčano	13
1019	22	38	Oblačno	21
1017	23	41	Sunčano	13
1017	24	34	Sunčano	13
1024	22	50	Oblačno	21
1027	20	37	Sunčano	21
1026	22	33	Sunčano	21
1018	25	41	Sunčano	21
1009	24	41	Sunčano	6

U tablici A.3 uz prikupljene podatke o temperaturi, tlaku, vlazi zraka i vremenu u 60 dana proljeća 2014. godine, dani su i vremenski uvjeti dobiveni modelom prilikom testiranja, budući da su ti podaci bili korišteni samo u testiranju, dok su podaci iz tablica A.1 i A.2 korišteni prilikom gradnje modela.

Tablica A.3 Tablica prikupljenih podataka 2014. godine od 16.03 do 14.05

Tlak(hPa)	Temperatura	Vlaga(%)	Vrijeme	Opservacijski simbol	Vrijeme HMM
1011	17	42	Sunčano	10	Sunčano
1016	19	46	Sunčano	13	Sunčano
1018	19	43	Sunčano	21	Sunčano
1020	14	67	Sunčano	19	Oblačno
1026	18	49	Sunčano	21	Sunčano
1017	20	33	Sunčano	13	Sunčano
1013	19	46	Sunčano	13	Sunčano
1004	15	67	Kiša	4	Oblačno
1006	11	67	Oblačno	4	Oblačno
1010	9	66	Oblačno	1	Kiša
1012	12	51	Oblačno	10	Oblačno
1011	13	63	Oblačno	11	Oblačno
1014	15	58	Oblačno	11	Oblačno
1022	18	45	Sunčano	21	Sunčano
1018	18	49	Sunčano	21	Sunčano
1013	18	37	Sunčano	13	Sunčano
1014	18	45	Sunčano	13	Sunčano
1012	19	43	Sunčano	13	Sunčano
1010	19	43	Sunčano	6	Sunčano
1008	19	37	Sunčano	6	Sunčano
1009	15	77	Kiša	4	Oblačno
1017	15	82	Oblačno	12	Oblačno
1020	20	56	Oblačno	22	Sunčano
1016	21	49	Sunčano	13	Sunčano
1021	12	77	Oblačno	19	Oblačno
1018	12	58	Sunčano	19	Oblačno
1020	13	54	Oblačno	19	Oblačno
1017	14	63	Sunčano	11	Oblačno
1016	16	63	Oblačno	11	Oblačno
1013	19	43	Sunčano	13	Sunčano
1017	10	66	Oblačno	11	Oblačno
1020	10	58	Oblačno	19	Oblačno
1018	8	71	Kiša	16	Oblačno
1009	17	42	Sunčano	3	Sunčano
1007	16	52	Sunčano	3	Sunčano
1009	13	88	Kiša	5	Kiša

1009	13	88	Kiša	5	Kiša
1012	13	73	Oblačno	11	Oblačno
1017	19	68	Oblačno	14	Sunčano
1014	18	77	Oblačno	14	Oblačno
1011	15	94	Kiša	12	Kiša
1011	16	77	Oblačno	11	Oblačno
1007	16	82	Oblačno	5	Kiša
1005	18	64	Oblačno	7	Oblačno
1007	18	77	Oblačno	7	Oblačno
1008	19	64	Oblačno	7	Oblačno
1009	11	56	Oblačno	4	Oblačno
1009	20	53	Oblačno	7	Oblačno
1009	13	94	Kiša	5	Kiša
1012	12	44	Sunčano	10	Oblačno
1020	14	55	Oblačno	19	Oblačno
1019	20	46	Sunčano	21	Sunčano
1015	21	40	Sunčano	13	Sunčano
1019	19	64	Oblačno	22	Sunčano
1017	22	46	Sunčano	13	Sunčano
1017	22	53	Oblačno	14	Sunčano
1008	23	53	Oblačno	7	Oblačno
1015	15	63	Oblačno	11	Oblačno
1013	11	88	Kiša	12	Kiša
1014	16	45	Sunčano	10	Oblačno