

UNIVERZITET U BEOGRADU  
MATEMATIČKI FAKULTET

MASTER RAD

---

Primena grafovskih neuronskih mreža na  
predviđanje vremena izvršavanja SAT  
rešavača

---

*Autor:*  
Nikola AJZENHAMER

*Mentor:*  
doc. dr Mladen NIKOLIĆ

ČLANOVI KOMISIJE:

doc. dr Mladen Nikolić  
prof. dr Predrag Janičić  
prof. dr Filip Marić



Beograd, 2020

UNIVERZITET U BEOGRADU

*Sažetak*

Matematički fakultet

Katedra za Računarstvo i informatiku

**Primena grafovskih neuronskih mreža na predviđanje vremena izvršavanja SAT rešavača**

Nikola AJZENHAMER

Problem zadovoljivosti formula iskazne logike, poznat kao *SAT problem*, predstavlja jedan od najznačajnijih problema teorijskog računarstva (kao jedan od NP-kompletnih problema), a ima značajno mesto i u realnim primenama. Zbog toga, istraživači decenijama rade na novim tehnikama konstrukcije programa koji ispituju zadovoljivost formula iskazne logike, tzv. *SAT rešavača*. Pored same korektnosti rešenja koju SAT rešavači pružaju, važno je voditi računa i o brzini rešavanja. Različitih SAT rešavača ima puno i najčešće neki od njih su veoma efikasni nad nekim familijama formula, dok nad nekim drugim pokazuju lošije performanse. Tako se javila ideja za konstruisanje *portfolija SAT rešavača* koji na osnovu određenih atributa konkretne formule iskazne logike procenjuju koji rešavač iz skupa unapred odabranih rešavača najbrže dolazi do rešenja. Mnogi takvi sistemi koriste metode mašinskog učenja za određivanje najpogodnijeg SAT rešavača. Pokazalo se da ovako konstruisani programi pružaju bolje performanse od najboljih pojedinačnih SAT rešavača.

Grafovske neuronske mreže mogu da rade direktno nad grafovskim reprezentacijama formula iskazne logike, umesto nad specifično definisanim skalarnim atributima. Kako je već poznato da neuronske mreže koje rade nad izvornim reprezentacijama podataka (što za iskazne formule mogu biti grafovi) često po performansama prestižu sisteme zasnovane na atributima koje su definisali ljudi, postoji osnov da se očekuju bolji rezultati od do sada postignutih. S druge strane, ova vrsta neuronskih mreža može biti računski zahtevnija za primenu. U ovom radu razvijen je sistem za predviđanje vremena izvršavanja SAT rešavača korišćenjem grafovskih neuronskih mreža. Uspeh odabranih metoda je upoređen sa regresionim modelima koji su do sada demonstrirali veliki uspeh: *k-najbližih suseda* i *šume nasumičnih stabala*.

## *Zahvalnice*

Najveću zahvalnost dugujem svom mentoru, Mladenu Nikoliću, za sve savete i veliko strpljenje tokom izrade ovog rada, kao i za pružanje inspiracije za istraživanje u interesantnoj oblasti mašinskog učenja.

Zahvaljujem se i članovima komisije, Predragu Janičiću i Filipu Mariću, koji su svojim komentarima poboljšali kvalitet ovog rada.

Konačno, na neizmernoj podršci se zahvaljujem svojoj porodici i prijateljima koji su me bodrili tokom svojih studija kao i tokom izrade rada. Posebno se zahvaljujem svojoj kolegici i dugogodišnjoj prijateljici Anji Bukurov na pažljivom čitanju rada i korisnim sugestijama.

# Sadržaj

<b>Sažetak</b>	<b>ii</b>
<b>Zahvalnice</b>	<b>iii</b>
<b>Spisak slika</b>	<b>vi</b>
<b>Spisak tabela</b>	<b>viii</b>
<b>1 Uvod</b>	<b>1</b>
<b>2 SAT problem</b>	<b>3</b>
2.1 Osnovni pojmovi iskazne logike . . . . .	3
2.2 Definicija SAT problema i NP-kompletnost . . . . .	6
2.3 SAT rešavači . . . . .	6
2.3.1 SAT takmičenja . . . . .	8
2.3.2 Portfolio SAT rešavači . . . . .	8
<b>3 Grafovi</b>	<b>10</b>
3.1 Osnovni pojmovi teorije grafova . . . . .	10
3.2 Računarska reprezentacija grafova . . . . .	12
3.3 Grafovska reprezentacija formula iskazne logike . . . . .	13
<b>4 Odabrani metodi mašinskog učenja</b>	<b>14</b>
4.1 Opšte o regresiji . . . . .	14
4.1.1 Ukratko o klasifikaciji . . . . .	16
4.2 $K$ -najbližih suseda . . . . .	16
4.3 Šuma nasumičnih stabala . . . . .	18
4.3.1 Stablo odlučivanja . . . . .	19
4.4 Neuronske mreže . . . . .	20
4.4.1 Neuronske mreže sa propagacijom unapred . . . . .	21
4.4.2 Konvolutivne neuronske mreže . . . . .	22
Konvolucija . . . . .	23
Agregacija . . . . .	23
4.5 Metodi zasnovani na grafovskim neuronskim mrežama . . . . .	24
4.5.1 Grafovske konvolutivne mreže . . . . .	25
Sloj konvolucije nad grafom . . . . .	25
4.5.2 Grafovske mreže sa mehanizmom pažnje . . . . .	26

Mehanizam pažnje nad grafom . . . . .	26
4.5.3 Klasifikacija i regresija nad grafom . . . . .	28
Kombinacija sa neuronskom mrežom sa propagacijom unapred . . . . .	28
Duboka grafovska konvolutivna neuronska mreža . . . . .	30
<b>5 Sistem za obučavanje portfolio SAT rešavača . . . . .</b>	<b>32</b>
5.1 Funkcionalnosti aplikacije . . . . .	32
5.2 Opisi podataka . . . . .	34
5.2.1 DIMACS format . . . . .	34
5.2.2 Edgelist format . . . . .	34
5.2.3 DGCNN format . . . . .	35
5.2.4 SATzilla atributi . . . . .	36
5.2.5 Node2Vec . . . . .	37
5.3 Opis sistema . . . . .	38
5.3.1 Paket code . . . . .	38
5.3.2 Paket preprocessing . . . . .	39
5.3.3 Paket common . . . . .	40
5.3.4 Paket libs . . . . .	40
5.3.5 Paket main.py i pomoćni skriptovi . . . . .	40
5.4 Implementacione napomene . . . . .	40
<b>6 Eksperimentalna evaluacija . . . . .</b>	<b>42</b>
6.1 Skup podataka i mere evaluacije . . . . .	42
6.2 Evaluacija modela za poredenje . . . . .	44
6.3 Evaluacija metoda zasnovanih na grafovskim neuronskim mrežama . . . . .	46
<b>7 Zaključak . . . . .</b>	<b>52</b>
<b>Literatura . . . . .</b>	<b>53</b>
<b>A Opisi SATzilla atributa . . . . .</b>	<b>56</b>
<b>B Detaljni rezultati eksperimenata . . . . .</b>	<b>59</b>

# Spisak slika

3.1	Grafički prikaz primera tri grafa: $G_1$ (levo), $G_2$ (sredina) i $G_3$ (desno).	11
3.2	Grafički prikaz grafa promenljivih i klauza iskazne formule $(p_1 \vee \neg p_3 \vee p_4) \wedge (p_2 \vee p_3 \vee \neg p_4) \wedge \neg p_4$ .	13
4.1	Primer određivanja 5-najbližih suseda za skup podataka iz tabele 4.1.	17
4.2	Primer konstruisanog stabla odlučivanja za skup podataka iz tabele 4.2.	19
4.3	Grafički prikaz jedne potpuno povezane neuronske mreže.	22
4.4	Grafički prikaz toka podataka kroz jedan konvolutivni sloj u konvolutivnoj neuronskoj mreži.	22
4.5	Ilustracija izvršavanja operacije konvolucije u dvodimenzionalnom slučaju [26].	24
4.6	Mehanizam pažnje u grafovskoj mreži sa mehanizmom pažnje [8].	27
4.7	Ilustracija mehanizma višestruke pažnje ( $K = 3$ ) za čvor 1 u njegovom susedstvu $N_1$ [8].	28
4.8	Ilustracija modela grafovske konvolutivne mreže za klasifikaciju/regresiju grafa.	29
4.9	Grafički prikaz arhitekture duboke grafovske konvolutivne neuronske mreže [9].	31
5.1	Dijagram aktivnosti aplikacije koja obučava modele za predviđanje vremena izvršavanja SAT rešavača i evaluira te modele.	33
5.2	Dijagram paketa <code>code</code> .	38
5.3	Dijagram paketa <code>preprocessing</code> .	39
5.4	Dijagram paketa <code>common</code> .	40
6.1	Ocene najboljeg modela $k$ -najbližih suseda po SAT rešavačima iz portfolija.	46
6.2	Ocene najboljeg modela šume nasumičnih stabala po SAT rešavačima iz portfolija.	46
6.3	Ocene najboljeg modela grafovske konvolutivne mreže po SAT rešavačima iz portfolija.	48
6.4	Ocene najboljeg modela grafovske mreže sa mehanizmom pažnje po SAT rešavačima iz portfolija.	48
6.5	Ocene najboljeg modela duboke grafovske konvolutivne neuronske mreže po SAT rešavačima iz portfolija.	48

6.6	Uporedni prikaz prosečnih ocena $R^2$ (levo) i $RMSE$ (desno) za sve modele koji su obučavani u eksperimentima. . . . .	49
6.7	Grafik funkcija cilja na skupu za obučavanje (plavo) i skupu za validaciju (crveno) prilikom obučavanja modela grafovske konvolutivne mreže na 200 epoha. . . . .	50
6.8	Ocene preprilagođenog modela grafovske konvolutivne mreže na skupu za obučavanje po SAT rešavačima iz portfolija. . . . .	50
6.9	Grafik funkcija cilja na skupu za obučavanje (plavo) i skupu za validaciju (crveno) prilikom obučavanja modela grafovske neuronske mreže sa manjom arhitekturom na 200 epoha. . . . .	51

# Spisak tabela

2.1	Iskazni veznici, njihove oznake i njihova značenja. . . . .	3
4.1	Skup podataka na osnovu kojeg se predviđa 5-najbližih suseda za instancu $x_{new} = (0.5, 0.5)$ sa slike 4.1. Prve dve kolone čine attribute instance, dok je poslednja kolona ciljna promenljiva koja se predviđa. . . . .	17
4.2	Skup podataka na osnovu kojeg je izgrađeno stablo odlučivanja sa slike 4.2. Prve tri kolone čine attribute instance, dok je poslednja kolona ciljna promenljiva koja se predviđa. . . . .	19
6.1	Raspodela instanci na skupove za obučavanje, validaciju i evaluaciju. . . . .	43
6.2	Mreža parametarskih konfiguracija za model $k$ -najbližih suseda. . . . .	44
6.3	Mreža parametarskih konfiguracija za model šuma nasumičnih stabala. . . . .	44
6.4	Prosečne ocene $R^2$ (gore) i $RMSE$ (dole) modela $k$ -najbližih suseda i šume nasumičnih stabala. Najbolji rezultati su prikazani podebljanim slovima. . . . .	45
6.5	Parametri obučavanja modela grafovske konvolutivne mreže. . . . .	47
6.6	Parametri obučavanja modela grafovske mreže sa mehanizmom pažnje. . . . .	47
6.7	Parametri obučavanja modela duboke grafovske konvolutivne neuronske mreže. . . . .	47
6.8	Prosečne ocene $R^2$ (gore) i $RMSE$ (dole) modela zasnovanih na grafovskim neuronskim mrežama. Najbolji rezultati su prikazani podebljanim slovima. . . . .	49
A.1	SATzilla atributi koji opisuju veličinu iskazne formule pre i posle preprocesiranja. . . . .	56
A.2	SATzilla atributi koji opisuju odnos pozitivnih i negativnih literala u klauzama (atributi sa prefiksom POSNEG-RATIO-CLAUSE), odnos pozitivnih i negativnih pojavljivanja promenljivih (atributi sa prefiksom POSNEG-RATIO-VAR) i procenat unarnih, binarnih i ternarnih klauza. . . . .	56
A.3	SATzilla atributi koji opisuju statistike čvorova grafa promenljivih i klauza koje se odnose na klauze (atributi sa prefiksom VCG-CLAUSE) i promenljive (atributi sa prefiksom VCG-VAR). . . . .	57
A.4	SATzilla atributi koji opisuju statistike čvorova grafa promenljivih. . . . .	57



A.5	SATzilla atributi koji opisuju statistike čvorova grafa klauza (atributi sa prefiksom CG-) i koeficijente klasterovanja (atributi sa prefiksom cluster-coeff). . . . .	57
A.6	SATzilla atributi koji opisuju statistike broja pojavljivanja u Hornovim klauzama za svaku promenljivu. . . . .	57
A.7	SATzilla atributi koji opisuju vremena potrebna za izračunavanje nekih od atributa. . . . .	58
B.1	Prosečne ocene $R^2$ najboljih modela $k$ -najbližih suseda i šume nasumičnih stabala, po SAT rešavačima. Najbolji rezultati su prikazani podebljanim slovima. . . . .	59
B.2	Prosečne ocene $RMSE$ najboljih modela $k$ -najbližih suseda i šume nasumičnih stabala, po SAT rešavačima. Najbolji rezultati su prikazani podebljanim slovima. . . . .	60
B.3	Prosečne ocene $R^2$ najboljih modela metoda zasnovanih na grafovskim neuronskim mrežama, po SAT rešavačima. Najbolji rezultati su prikazani podebljanim slovima. . . . .	61
B.4	Prosečne ocene $RMSE$ najboljih modela metoda zasnovanih na grafovskim neuronskim mrežama, po SAT rešavačima. Najbolji rezultati su prikazani podebljanim slovima. . . . .	62



# Glava 1

## Uvod

Veliki broj teško rešivih računarskih problema pripadaju klasi tzv. *NP-kompletnih problema*. Direktno pronalaženje rešenja ovih problema je izazovno jer je za svaki NP-kompletni problem potrebno konstruisati nove algoritme koji ih rešavaju, pa zatim dalje usavršavati te algoritme kako bi se dobilo na brzini dolaženja do rešenja, vrlo često po cenu kvaliteta rešenja (na primer, heuristički pristupi).

Ipak, svi NP-kompletni problemi imaju jednu zajedničku osobinu, a to je da se mogu svesti na druge NP-kompletne probleme, za koje već postoje razrađeni algoritmi koji nude potpuna ili nepotpuna (ali često dovoljno dobra) rešenja u znatno bržem vremenu nego što bi novi algoritmi to uradili. Nakon rešavanja problema na koji je polazni problem sveden, rekonstrukcija rešenja polaznog problema je trivijalan posao. Zbog toga, određeni NP-kompletni problemi su intenzivno izučavani decenijama i za njih su razvijeni različiti algoritmi koji koriste različite tehnike za njihovo rešavanje, sa različitim osobinama. Jedan takav problem, možda najznačajniji, jeste *problem određivanja zadovoljivosti iskazne formule*, odnosno, *SAT problem*.

SAT problem je definisan u okvirima matematičke teorije koja se naziva *iskazna logika*. U svojoj osnovi, iskazna logika se bavi proučavanjem iskaza koji mogu biti tačni ili netačni, njihovim zakonitostima i odnosima koje oni međusobno ostvaruju. U poglavlju 2 uvodimo elementarne pojmove iskazne logike i nad njima definišemo SAT problem. Takođe, dajemo kratak uvid u istorijat SAT problema i tehnika za njegovo rešavanje, razvijenih kroz opisani period [1].

Pristupa rešavanju SAT problema ima raznih i, kao što smo rekli, svaki od njih ima neke specifičnosti. Jedna od važnih osobina ovih algoritama jeste vreme koje je utrošeno u rešavanju konkretne instance SAT problema. Postavlja se pitanje da li je moguće predvideti koliko će vreme neki algoritam utrošiti za rešavanje neke instance SAT problema pre nego što započne rešavanje. Odgovor na ovo pitanje bi nam omogućilo da, za svaki konkretan problem, odaberemo onaj algoritam koji nam najbrže daje rešenje. Glavni cilj ovog rada jeste ispitivanje jednog konkretnog metoda za predviđanje vremena izvršavanja algoritama nad grafovskim reprezentacijama SAT problema i poređenje njegovih rezultata sa tradicionalnijim tehnikama koje su pokazale uspeh. U poglavlju 3 uvodimo elementarne pojmove *teorije grafova* koji su nam važni za razumevanje načina na koji se iskazne formule mogu reprezentovati u računaru [2]. U

poglavljju 4 uvodimo pojmove *regresije* i *klasifikacije* kao probleme koji se bave predviđanjima u mašinskom učenju [3] i predstavljamo metode koji će biti korišćeni u radu za predviđanje vremena izvršavanja SAT rešavanja. Metodi  $k$ -najbližih suseda [4] i šume nasumičnih stabala [5, 6] biće korišćeni kao metodi za upoređivanje rezultata metoda koji se analiziraju u ovom radu i koji su zasnovani na grafovskim neuronskim mrežama: grafovske konvolutivne mreže [7], grafovske mreže sa mehanizmom pažnje [8] i duboke grafovske konvolutivne neuronske mreže [9]. U poglavljju 5 opisujemo implementaciju softvera koji koristi metode opisane u prethodnom poglavljju u cilju predviđanja vremena izvršavanja SAT rešavača nad konkretnim podacima. Dodatno, opisujemo i same podatke koji predstavljaju ulaz u ovaj softver. U poglavljju 6 dajemo detaljnu analizu rezultata eksperimenata koji su izvedeni implementiranim softverom iz prethodnog poglavlja. Evaluacijom opisanih modela smo utvrdili da oni ipak nisu u stanju da dostignu zadovoljavajuće prediktivne sposobnosti. U poslednjem poglavljju 7 sumiramo rezultate iz prethodnog poglavlja i dajemo konačne zaključke i predloge za dalji rad u vidu poboljšanja performansi ovih modela.

## Glava 2

# SAT problem

U ovom poglavlju definišemo SAT problem u okvirima iskazne logike i diskutujemo o postojećim rešenjima ovog problema, kao i o nekim njihovim osobinama koji su nam značajni za definisanje centralnog problema našeg rada.

### 2.1 Osnovni pojmovi iskazne logike

Osnovni pojam u iskaznoj logici je *iskaz* koji može biti tačan ili netačan. Primeri iskaza su:

- U Beogradu je uvek sunčano.
- Ja imam 26 godina.

Iskazi se predstavljaju *iskaznim promenljivim* koje se označavaju malim slovima  $p, q, r$ , itd. Postoje iskazi koje možemo smatrati *uvek tačnim* ili *uvek netačnim*. Njih ćemo označavati simbolom  $\top$ , odnosno, simbolom  $\perp$ , redom. U iskaznoj logici, pored iskaznih promenljivih, postoje i *logički veznici* koji od jednostavnijih iskaza formiraju složenije iskaze. Iskazne promenljive i složene iskaze nazivamo *iskaznim formulama*. Osnovni iskazni veznici i njihove oznake su date u tabeli 2.1, zajedno sa neformalnim opisima njihovih značenja.

Veznik (oznaka)	Značenje
ne ( $\neg$ )	Iskaz $\neg p$ je tačan samo ako je $p$ netačan.
i ( $\wedge$ )	Iskaz $p \wedge q$ je tačan samo ako su iskazi $p$ i $q$ tačni.
ili ( $\vee$ )	Iskaz $p \vee q$ je tačan ako je makar jedan od iskaza $p$ i $q$ tačan.
povlači ( $\implies$ )	Iskaz $p \implies q$ je netačan ako je $p$ tačan, a $q$ netačan.
ekvivalentno ( $\iff$ )	Iskaz $p \iff q$ je tačan ako su oba iskaza $p$ i $q$ bilo tačna ili netačna.

TABELA 2.1: Iskazni veznici, njihove oznake i njihova značenja.

Da bismo formalizovali semantiku iskaznih veznika, uvešćemo neke dodatne oznake i definicije. Tačnost iskazne formule  $F$  zavisi od funkcije koju nazivamo valuacija, te zbog toga govorimo o tačnosti iskazne formule u valuaciji.

**Definicija 2.1** Valuacija je funkcija koja skupu iskaznih promenljivih  $\mathcal{P}$  dodeljuje vrednost 0 ili 1, odnosno,  $v : \mathcal{P} \longrightarrow \{0, 1\}$ .

**Definicija 2.2** *Ako je iskazna formula  $F$  tačna u valuaciji  $v$ , onda kažemo da valuacija  $v$  zadovoljava iskaznu formulu  $F$  ili da je njen model. Ovu činjenicu označavamo  $v \models F$ . U suprotnom, označavamo  $v \not\models F$ .*

Tačnost iskazne formule  $F$  u valuaciji  $v$  se jednostavno određuje izračunavanjem vrednosti te iskazne formule u datoj valuaciji. Ovo je moguće uraditi uvođenjem koncepta interpretacije iskaznih formula.

**Definicija 2.3** *Valuacija  $v$  određuje jedinstvenu funkciju  $I_v$  koja izračunava tačnost neke iskazne formule u valuaciji  $v$  i koja se naziva interpretacija ( $p$  je iskazna promenljiva, dok su  $G$  i  $H$  proizvoljne iskazne formule):*

- *Ako je  $F = \top$ , onda je  $I_v(F) = 1$ . Ako je  $F = \perp$ , onda je  $I_v(F) = 0$ .*
  - *Ako je  $F = p$ , onda je  $I_v(F) = v(p)$ .*
  - *Ako je  $F = \neg G$ , onda je  $I_v(F) = 1$  ako je  $I_v(G) = 0$ . U suprotnom,  $I_v(F) = 0$ .*
  - *Ako je  $F = G \wedge H$ , onda je  $I_v(F) = 1$  ako je istovremeno  $I_v(G) = 1$  i  $I_v(H) = 1$ . U suprotnom,  $I_v(F) = 0$ .*
  - *Ako je  $F = G \vee H$ , onda je  $I_v(F) = 0$  ako je istovremeno  $I_v(G) = 0$  i  $I_v(H) = 0$ . U suprotnom,  $I_v(F) = 1$ .*
  - *Ako je  $F = G \implies H$ , onda je  $I_v(F) = 0$  ako je istovremeno  $I_v(G) = 1$  i  $I_v(H) = 0$ . U suprotnom,  $I_v(F) = 1$ .*
  - *Ako je  $F = G \iff H$ , onda je  $I_v(F) = 1$  ako je ispunjen jedan od naredna dva uslova:*
    - *Istovremeno je  $I_v(G) = 1$  i  $I_v(H) = 1$ .*
    - *Istovremeno je  $I_v(G) = 0$  i  $I_v(H) = 0$ .*
- U suprotnom,  $I_v(F) = 0$ .*

*Važi da  $v \models F$  ako i samo ako  $I_v(F) = 1$ .*

Da bi vrednost formule  $F$  bila dobro definisana u nekoj interpretaciji  $I_v$ , domen  $\mathcal{P}$  valuacije  $v$  mora biti nadskup skupa iskaznih promenljivih koje učestvuju u formuli  $F$ , kojeg ćemo označiti  $\mathcal{P}_F$ . Tako, na primer, ako je data formula  $F = (p \wedge q) \vee \neg q$ , onda je njena vrednost u interpretacijama određenim valuacijama  $v_1 = \{(p, 1), (q, 1)\}$  i  $v_2 = \{(p, 0), (q, 1), (r, 1)\}$  dobro definisana, dok u interpretaciji određenoj valuacijom  $v_3 = \{(p, 0), (r, 0)\}$  nije dobro definisana zbog toga što  $v_3$  ne definiše vrednost iskazne promenljive  $q \in \mathcal{P}_F$ . U nastavku ćemo pretpostaviti da valuacije definišu vrednosti za sve iskazne promenljive koje učestvuju u odgovarajućim iskaznim formulama.

Kao što smo videli, vrednost iskazne formule  $F$  zavisi od valuacije. Da bismo ovo ilustrovali, posmatrajmo formulu  $F = (p \wedge q) \vee \neg q$ . Izračunajmo vrednost ove

formule u interpretacijama određenim prethodno datim valuacijama  $v_1$  i  $v_2$ . Da bismo to uradili, potrebno je da izračunamo vrednosti potformula koje se nalaze u formuli  $F$  u tim interpretacijama i to počevši od najjednostavnijih potformula (tj. iskaznih promenljivih,  $\top$  i  $\perp$ ) koje predstavlja bazu rekurzivne definicije 2.3, zatim gradeći složenije potformule primenom iskaznih veznika nad jednostavnijim potformulama dok se ne dođe do formule  $F$ . Dakle:

- Kako je  $I_{v_1}(p) = v_1(p) = 1$  i  $I_{v_1}(q) = v_1(q) = 1$ , to je  $I_{v_1}(\neg q) = 0$  i  $I_{v_1}(p \wedge q) = 1$ , pa zaključujemo  $I_{v_1}(F) = I_{v_1}((p \wedge q) \vee \neg q) = 1$ . Otuda,  $v_1 \models F$ .
- Kako je  $I_{v_2}(p) = v_2(p) = 0$  i  $I_{v_2}(q) = v_2(q) = 1$ , to je  $I_{v_2}(\neg q) = 0$  i  $I_{v_2}(p \wedge q) = 0$ , pa zaključujemo  $I_{v_2}(F) = I_{v_2}((p \wedge q) \vee \neg q) = 0$ . Otuda,  $v_2 \not\models F$ .

Osobina *zadovoljivosti* iskaznih formula će nam biti veoma važna za definisanje SAT problema u narednoj sekciji. Pored svojstva zadovoljivosti, postoje još neka interesantna svojstva iskaznih formula:

- Ako postoji makar jedna valuacija  $v$  u kojoj je iskazna formula  $F$  zadovoljena, onda kažemo da je iskazna formula  $F$  *zadovoljiva* (eng. *satisfiable*).
- Ako je iskazna formula  $F$  zadovoljena u svim valuacijama  $v$ , onda kažemo da je iskazna formula  $F$  *tautologija* (eng. *tautology*).
- Ako postoji makar jedna valuacija  $v$  u kojoj iskazna formula  $F$  nije zadovoljena, onda kažemo da je iskazna formula  $F$  *poreciva* (eng. *falsifiable*).
- Ako ne postoji nijedna valuacija  $v$  u kojoj je iskazna formula  $F$  zadovoljena, onda kažemo da je iskazna formula  $F$  *kontradikcija* (eng. *contradictory*).

Kao što vidimo, sva nabrojana svojstva neke iskazne formule se mogu ispitati tako što se ispita zadovoljivost te iskazne formule. Pređimo sada na diskusiju o normalnim formama i logičkoj ekvivalentnosti iskaznih formula.

**Definicija 2.4** *Kažemo da je iskazna formula  $F$  u konjuktivnoj normalnoj formi (skraćeno, KNF) ako je oblika:*

$$c_0 \wedge c_1 \wedge \dots \wedge c_{n-1}$$

pri čemu svaka od potformula  $c_i$ , za svako  $i \in \{0, 1, \dots, n-1\}$ , naziva se klauza  $i$  oblika je:

$$l_0 \vee l_1 \vee \dots \vee l_{m-1}$$

pri čemu svaki od  $l_j$ , za svako  $j \in \{0, 1, \dots, m-1\}$ , naziva se literal  $i$  predstavlja iskaznu promenljivu  $p$  ili njenu negaciju  $\neg p$  iz nekog skupa iskaznih promenljivih  $\mathcal{P}$  (broj  $m$  se može razlikovati između svake klauze).

**Definicija 2.5** Kažemo da su iskazne formule  $F$  i  $G$  logički ekvivalentne ako važi  $I_v(F) = I_v(G)$  u svim valuacijama  $v : \mathcal{P} \rightarrow \{0, 1\}$ .

Logička ekvivalentnost je važan koncept, zato što nam omogućuje da izvršavamo transformacije nad formulama bez menjanja njihove semantike.

**Teorema 2.6** Svaka iskazna formula  $F$ , koja nije u KNF, može se transformisati u drugu formulu  $G$ , pri čemu važi:

1.  $G$  je u KNF.
2.  $F$  i  $G$  su logički ekvivalentne.

## 2.2 Definicija SAT problema i NP-kompletnost

Od svih nabrojanih osobina iskaznih formula, ispostavlja se da zadovoljivost predstavlja verovatno najvažniju osobinu zbog svojih teorijskih i praktičnih primena. Naime, postoji veliki broj značajnih problema u računarstvu koji se smatraju teško rešivim problemima. Ovi problemi pripadaju klasi tzv. *NP-kompletnih problema*. Ova klasa je značajna po tome što postoji na stotine važnih problema koji joj pripadaju, kao što su [10]: *problem trgovačkog putnika*, *problem klika*, *problem pokrivača grana*, *problem dominirajućeg skupa*, itd. Dva važna zapažanja svih problema koje pripadaju ovoj klasi su:

- Do sada, nije poznat algoritam polinomske vremenske složenosti (ovakve algoritme smatramo *efikasnim*) koji rešava NP-kompletne probleme.
- Ako postoji algoritam polinomske vremenske složenosti koji rešava makar jedan NP-kompletan problem, onda se i svi ostali NP-kompletni problemi mogu rešiti u polinomskoj vremenskoj složenosti.

Verovatno najznačajniji NP-kompletan problem je *problem ispitivanja zadovoljivosti iskaznih formula*, koji je poznat pod nazivom *SAT problem*<sup>1</sup>. Ovaj problem je važan zato što predstavlja prvi problem za koji je dokazano da pripada klasi NP-kompletnih problema od strane Stivena Kuka, 1971. godine [10].

*SAT problem* je definisan narednim pitanjem: „Da li za datu formulu iskazne logike  $F$  u KNF postoji valuacija  $v$  takva da  $v \models F$ ?”

Uvedimo i jedan tehnički termin, *instanca SAT problema*, koji označava jednu konkretnu iskaznu formulu koja predstavlja ulaz za SAT problem.

## 2.3 SAT rešavači

Postoje razne tehnike kojima se može doći do odgovora na pitanje koje postavlja SAT problem. Ove tehnike implementiraju računarski programi koji se nazivaju *SAT*

<sup>1</sup>SAT je skraćenica od engleske reči *satisfiability* (zadovoljivost).



rešavači (eng. *SAT solver*). Ulazni podatak za SAT rešavač jeste formula  $F$  iskazne logike transformisana u KNF. Očigledno, postoje dva odgovora na pitanje koje postavlja SAT problem, te samim tim, SAT rešavači mogu proizvesti dva odgovora:

- „Da” (ovaj odgovor se često označava oznakom **SAT**), u kom slučaju se često zahteva od SAT rešavača da proizvede i valuaciju u kojoj je formula  $F$  zadovoljena.
- „Ne” (ovaj odgovor se često označava oznakom **UNSAT**), u kom slučaju se često zahteva od SAT rešavača da proizvede dokaz nezadovoljivosti.

U istoriji automatskog rešavanja SAT problema je bilo nekoliko značajnih pomaka koje vredi spomenuti. Najelementarnija procedura za ispitivanje zadovoljivosti iskazne formule jeste metod *istinitosnih tablica* (eng. *truth tables*). Ovaj metod se zasniva na pronalaženju svih mogućih valuacija skupa iskaznih promenljivih  $\mathcal{P}_F$ , a zatim pronalaženje vrednosti formule  $F$  u svim interpretacijama tih valuacija nekim redosledom dok se ne pronađe valuacija  $v$  za koju je  $v \models F$ , ako takva postoji. Ovih valuacija ima ukupno  $2^{|\mathcal{P}_F|}$ , stoga je ovaj algoritam eksponencijalne vremenske složenosti i smatra se vrlo neefikasnim za iole složenije iskazne formule.

Kako bi se smanjio prostor pretrage valuacija, Martin Dejvis<sup>2</sup> i Hilari Putnam<sup>3</sup> su razvili tzv. *DP proceduru*, koja se zasniva na prepoznavanju određenih pravila u iskaznoj formuli zapisanoj u KNF i eliminaciji ili transformaciji klauza iskazne formule. Oni su 1958. godine predložili KNF kao pogodnu formu za ispitivanje zadovoljivosti iskaznih formula [1].

Džordž Logman<sup>4</sup> i Donald Lavlend<sup>5</sup> su 1962. godine uveli dodatna pravila za transformaciju iskazne formule, čime su još više smanjili prostor pretrage [1]. U kombinaciji sa pravilima iz DP procedure, formirana je *DPLL procedura*, koja je jedna od najpoznatijih tehnika za ispitivanje zadovoljivosti iskaznih formula.

Procedure razvijene krajem 20. i početkom 21. veka su podrazumevale unapređenje DPLL procedure u vidu iterativnih implementacija i uvođenjem novih pravila radi smanjenja prostora pretrage, korišćene memorije i, generalno, ubrzavanja rada SAT rešavača. Primeri takvih rešavača su oni koji koriste *CDCL pristup*. Rešavači zasnovani na pomenutim pristupima imaju važnu osobinu *potpunosti*, što označava da će pronaći tačno rešenje za proizvoljnu instancu SAT problema. Sa druge strane, postoje i *nepotpuni* SAT rešavači koji imaju osobinu da mogu relativno brzo da pronađu valuaciju u kojoj je instanca SAT problema zadovoljiva, ali ne mogu da utvrde da iskazna formula nije zadovoljiva [1]. Savremeni SAT rešavači mogu rešavati iskazne formule sa milionima promenljivih i milijardama klauza u minutima [11].

<sup>2</sup>Martin Davis (1928–), američki matematičar.

<sup>3</sup>Hilary Putnam (1926–2016), američki filozof, matematičar i informatičar.

<sup>4</sup>George Logemann (1938–2012), američki matematičar i informatičar.

<sup>5</sup>Donald W. Loveland (1934–), američki profesor informatike.

### 2.3.1 SAT takmičenja

Razvojem savremenih SAT rešavača se javilo pitanje konkurencije: koji SAT rešavač je najbolji? Da bi ovo pitanje bilo valjano definisano, neophodno je definisati ciljnu funkciju  $f$  koja preslikava SAT rešavač u realan broj. Ovakva funkcija se koristi za upoređivanje osobina SAT rešavača – SAT rešavač za koju funkcija  $f$  proizvede najmanju vrednost smatramo *najboljim u odnosu na ciljnu funkciju  $f$* . Neki primeri funkcija  $f$  su:

- Funkcija  $f_{vreme}$  koja izračunava vreme koje je potrebno SAT rešavaču da da odgovor za neki skup iskaznih formula. Ovu funkciju možemo koristiti ukoliko nam je značajno da što pre dobijemo odgovor od nekog SAT rešavača.
- Funkcija  $f_{odgovor}$  koja izračunava vrednost netačnih odgovora za neki skup iskaznih formula (za koje su odgovori unapred poznati). Ova funkcija je značajna ako želimo da evaluiramo koliko je neki SAT rešavač tačan u svojim odgovorima.
- Moguće je kombinovati prethodne funkcije i njima slične u cilju konstruisanja kompleksnijih funkcija za upoređivanje kvaliteta SAT rešavača.

Kako bi svi oni koji koriste SAT rešavače dobili odgovor na pitanje najboljeg SAT rešavača u odnosu na ciljnu funkciju  $f$ , godinama unazad se organizuju razna takmičenja na kojima je moguće prijaviti svoj SAT rešavač za neku od kategorija. Najveće takvo takmičenje je „SAT takmičenje” [12], koje je prvi put organizovano 2002. godine i koje se od tada organizuje jednom godišnje.

### 2.3.2 Portfolio SAT rešavači

Posmatrajući neke od najboljih SAT rešavača u odnosu na ciljnu funkciju  $f^6$ , jedno od važnih zapažanja koje je rano ustanovljeno je sledeće: ne postoji jedinstven SAT rešavač koji rešava pitanje SAT problema za sve instance najbolje u odnosu na ciljnu funkciju  $f$ .

Na osnovu ovog zapažanja se rodila ideja za kombinovanje više SAT rešavača, pri čemu se za svaki od njih pretpostavlja da je dovoljno dobar za neke familije instanci, premda je loš za neke druge. Prema ovoj ideji, mogao bi se konstruisati sistem koji bi za određenu instancu mogao da *odluči* koji SAT rešavač bi bilo najbolje koristiti (u odnosu na ciljnu funkciju  $f$ ) za rešavanje SAT problema za tu iskaznu formulu. Ovim je opisana osnovna ideja na kojoj su zasnovani *portfolio SAT rešavači*. Reč „portfolio” se odnosi na skup SAT rešavača koji se koriste za rešavanje SAT problema.

Jedan od najpopularnijih portfolio SAT rešavača jeste SATzilla, koji je osvajao medalje na različitim SAT takmičenjima [13], čime je empirijski demonstrirao prethodno pomenuto zapažanje o prednosti portfolio SAT rešavača u odnosu na pojedinačne SAT rešavače.

<sup>6</sup>Najčešće se kao polazna tačka koristi funkcija  $f_{vreme}$ , s obzirom da se često očekuje da je vrednost ciljne funkcije  $f_{odgovor}$  za svaki SAT rešavač jednaka nuli, odnosno, da SAT rešavači ne prave greške u svojim odgovorima. Naravno, moguće je koristiti i složenije ciljne funkcije.

Ključno pitanje koje se nameće je na koji način sistem portfolio SAT rešavača vrši odlučivanje koji SAT rešavač iz portfolija treba pozvati za konkretnu iskaznu formulu? Portfolio SAT rešavači se razlikuju upravo po strategiji za odabir konkretnog SAT rešavača. Već pomenut SATzilla rešavač je u raznim iteracijama imao različite tehnike za odlučivanje:

- Prva verzija je koristila metod grebene regresije za predviđanje vrednosti ciljne funkcije  $f_{vreme}$  [14].
- Verzija SATzilla2012 je koristila metod šume nasumičnih stabala za predviđanje vrednosti ciljne funkcija koja direktno proporcionalno „kažnjava” loše odgovore SAT rešavača u odnosu na njihove performanse, umesto predviđanja vremena izvršavanja [15].

Sa druge strane, postoje SAT rešavači koji su značajno jednostavniji po konstrukciji od SATzilla sistema, a koji nude jednako dobre ili bolje performanse. Jedan takav rešavač je ArgoSmArT-kNN, koji koristi metod  $k$ -najbližih suseda za određivanje SAT rešavača iz portfolija za datu instancu [16].

## Glava 3

# Grafovi

U ovom poglavlju ćemo uvesti neke osnovne elemente teorije grafova i pokazati jedan način za predstavljanje iskaznih formula pomoću grafova. Ovakva reprezentacija će nam biti važna za potrebe korišćenja metoda za predviđanje vremena izvršavanja SAT rešavača koje ćemo izložiti u narednom poglavlju.

### 3.1 Osnovni pojmovi teorije grafova

**Definicija 3.1** Graf  $G$  predstavlja uređen par  $(V, E)$ , gde je  $V$  skup čije elemente nazivamo čvorovi, a  $E \subseteq V \times V$  je skup čije elemente nazivamo grane.

Kažemo da su čvorovi  $u, v \in V$  povezani ako važi  $\{u, v\} \in E$ . Čvor  $v \in V$  je *incidentan* sa granom  $e \in E$  (takođe, grana  $e$  je *incidentna* sa čvorom  $v$ ) ako postoji  $u \in V$  takav da je  $e = \{u, v\} \in E$ .

Grafovi se mogu podeliti prema usmerenosti na:

- *Neusmerene grafove*, za koje važi da grane nemaju svojstvo usmerenja između čvorova sa kojima su incidenti, zbog čega grane smatramo skupovima čvorova i pišemo, na primer,  $e = \{u, v\}$ .
- *Usmerene grafove*, za koje važi da grane imaju svojstvo usmerenja od jednog čvora, koji se naziva *izlazni čvor*, ka drugom čvoru koji se naziva *ulazni čvor*. U tom slučaju, grane grafa ne smatramo skupovima već uređenim parovima i pišemo, na primer,  $e = (u, v)$  ili  $e = (v, u)$ , u zavisnosti od smera usmerenja grane  $e$  incidentne sa čvorovima  $u$  i  $v$ .

U tekstu nadalje ćemo koristiti oznaku  $\{u, v\}$  za granu nekog grafa, pri čemu ćemo podrazumevati da se u slučaju usmerenog grafa sve grane podrazumevaju kao uređeni parovi. Specijalno, ukoliko naznačimo da je neki graf usmeren, korišćićemo oznake  $(u, v)$  ili  $(v, u)$  za grane tog grafa.

Dodatno, ukoliko za neki graf  $G = (V, E)$  važi da je  $E$  multiskup, onda se takav graf naziva *multigraf*. Multigrafovi takođe mogu biti ili usmereni ili neusmereni.

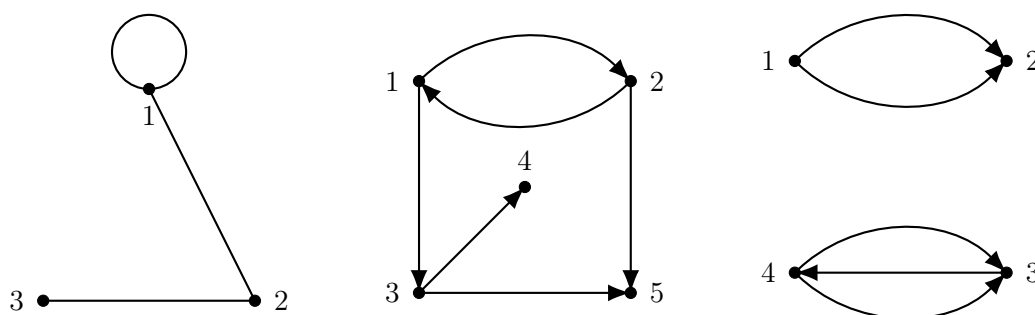
Grafovi se često grafički predstavljaju na sledeći način:

- Svaki čvor  $v \in V$  se predstavlja jednim krugom sa oznakom čvora u njegovoj blizini.

- Svaka grana  $\{u, v\} \in E$  se predstavlja linijom koja povezuje čvorove sa kojima je incidentna. Na liniji se navodi oznaka grane (ovo se može i izbeći pošto se sa slike može jednostavno utvrditi o kojoj grani je reč). Ako je graf usmeren, onda se često umesto linije koristi strelica u smeru od izlaznog čvora ka ulaznom čvoru incidentnim sa tom granom i svaka grana ima svoju strelicu koja grafički ilustruje njeno usmerenje. Takođe, ako je graf multigraf, onda za svaku granu postoji po jedna linija (strelica) koja povezuje odgovarajuće čvorove.

Na slici 3.1 je dat primer prikaza tri grafa:

- Graf  $G_1 = (V_1, E_1)$ , gde je  $V_1 = \{1, 2, 3\}$ ,  $E_1 = \{\{1, 1\}, \{1, 2\}, \{2, 3\}\}$ , primer je neusmerenog grafa.
- Graf  $G_2 = (V_2, E_2)$ , gde je  $V_2 = \{1, 2, 3, 4, 5\}$ ,  $E_2 = \{(1, 3), (1, 2), (3, 4), (2, 5), (2, 1), (3, 5)\}$ , primer je usmerenog grafa.
- Graf  $G_3 = (V_3, E_3)$ , gde je  $V_3 = \{1, 2, 3, 4\}$  i  $E_3 = \{(1, 2), (1, 2), (3, 4), (4, 3), (4, 3)\}$ , primer je usmerenog multigrafa.



SLIKA 3.1: Grafički prikaz primera tri grafa:  $G_1$  (levo),  $G_2$  (sredina) i  $G_3$  (desno).

Grafovi su značajni po tome što se veliki broj koncepata može modelovati njima. Na primer, mapa jedne države se može modelovati (ne)usmerenim multigrafom na sledeći način: svakom gradu se dodeljuje čvor, a postoji grana između čvorova ukoliko postoji direktan put između gradova koji odgovaraju tim čvorovima u grafu. Ako ima više direktnih puteva između dva grada, onda svakom putu odgovara po jedna grana u grafu, te je otuda ovaj graf multigraf. Slično tome, društvene mreže se prirodno mogu opisati grafom u kojem se svakom korisniku društvene mreže dodeljuje čvor, a grana između dva čvora postoji ukoliko su korisnici „prijatelji” (Facebook; neusmeren graf) ili ako jedan korisnik „prati” drugog korisnika (Twitter, Instagram; usmeren graf).

U skladu sa ovim modelovanjem, različiti problemi koji se ispituju nad ovakvim strukturama se mogu matematički modelovati konceptima iz teorije grafova. U skladu sa prethodnim primerima, problem pronalaska najkraćeg puta između dva grada u državi se može modelovati problemom pronaska najkraćeg puta između dva čvora u grafu. Drugi primer jeste korišćenje informacija iz grafa koji modeluje društvenu

mrežu u cilju optimizacije sredstava koje neka kompanija koristi za reklamiranje svog proizvoda. Ukoliko postoji informacija da su neki korisnici kupili određeni proizvod, kao i da postoje njihovi „prijatelji” ili „pratioci” koji dele njihova interesovanja, verovatno ima smisla reklamirati proizvod njima, umesto nasumičnim korisnicima. Ovo predstavlja problem pronalaženja zajednica u grafu.

Čvorovima i granama u grafovima možemo dodeliti odgovarajuća svojstva koje nazivamo *atributi*. Na primer, u slučaju grafa mape države, čvorovima koji odgovaraju gradovima možemo dodeliti attribute poput broja stanovnika, gustine naseljenosti, prosečno vreme za koje je moguće proputovati od jedne strane ka drugoj strani grada, itd. Granama možemo dodeliti attribute poput: prosečnog vremena za koje se put pređe, da li je put u funkciji ili ne, da li postoje radovi na putu ili ne, itd. U grafu društvene mreže, čvorovima možemo dodeliti attribute koji odgovaraju ličnim informacijama korisnika i njihovim interesovanjima, a granama možemo dodeliti informacije o interakcijama između korisnika, na primer, broj „lajkova” na objavama, broj komentara na objavama, niz razmenjenih privatnih poruka, itd. Atributi pridružuju čvorovima i granama grafova značenje iz domena problema koja je moguće koristiti u eksplorativnoj analizi grafova.

## 3.2 Računarska reprezentacija grafova

S obzirom da postoji veliki broj podataka u problemima koji se modeluju grafovima, postavlja se pitanje izbora njihove reprezentacije u računarima. Najčešće, za predstavljanje grafova se koristi matricna reprezentacija.

**Definicija 3.2** Matrica susedstva  $A$  grafa  $G = (V, E)$ , dimenzija  $|V| \times |V|$ , definiše se na sledeći način:

$$A_{ij} = \begin{cases} 1, & \{i, j\} \in E \\ 0, & \{i, j\} \notin E \end{cases}$$

Ako je graf neusmeren, onda je matrica  $A$  simetrična, pa nije potrebno čuvati sve elemente matrice, već samo njenu gornju-trougonaonu reprezentaciju. Čak i ako grafovi nisu neusmereni, oni su često *retki*, odnosno, nemaju veliki broj grana<sup>1</sup>, pa se matrice susedstva čuvaju u specifičnim strukturama podataka koje se nazivaju *retke matrice*. Retke matrice se mogu implementirati na različite načine. Jedna od reprezentacija retkih matrica jeste tzv. *rečnička matrica*, koja čuva parove oblika  $(e, x)$ , gde je  $e \in E$ , a  $x$  ne-nula vrednost (u slučaju matrice susedstva, to je vrednost 1). Retke matrice imaju dve značajne pogodnosti za korišćenje u odnosu na ne-retke matrice:

- Ne-retke matrice zahtevaju memorijski prostor asimptotske složenosti  $O(|V|^2)$ , dok retke matrice zahtevaju memorijski prostor asimptotske složenosti  $O(|E|)$ .

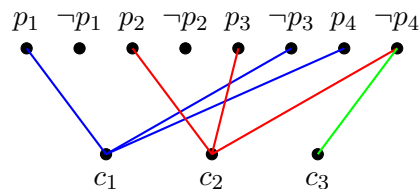
<sup>1</sup>Ne postoji stroga definicija mere *retkosti* nekog grafa, već samo možemo govoriti o tome da li je neki graf ređi od drugog.

Često je  $|E| \in O(|V|)$ , te je otuda neophodan prostor za retke matrice asimptotske složenosti  $O(|V|)$ , što je značajno poboljšanje. U zavisnosti od problema, ova ušteda može biti veoma bliska uštedi od 100%, na primer, 99.99%.

- Operacije koje se izvršavaju nad retkim matricama su efikasnije u odnosu na operacije nad uobičajenim reprezentacijama matricama. U algoritmima mašinskog učenja, značajna operacija je množenje matrice i vektora. Ova operacija u slučaju korišćenja ne-retke matrice zahteva u naivnim implementacijama vremensku asimptotsku složenost  $O(|V|^2)$ . U zavisnosti od vrste retke matrice, ova složenost može biti značajno manja, na primer,  $O(|E|)$  [17]. Kao što smo videli u prethodnoj tački, ovo može predstaviti značajno ubrzanje.

### 3.3 Grafovska reprezentacija formula iskazne logike

Kao što smo prethodno diskutovali, ulazne instance SAT problema predstavljaju iskazne formule prezapisane u KNF. Iskazne formule je moguće predstaviti grafovima na različite načine. Jedan od načina koji je pogodan za iskazne formule u KNF jeste tzv. *graf promenljivih i klauza* (eng. *variable clause graph*, skr. *VCG*). U ovoj reprezentaciji, svakom literalu i svakoj klauzi iz formule se dodeljuje po jedan čvor, a grana  $\{l, c\}$ , gde je  $l$  čvor koji odgovara nekom literalu i  $c$  čvor koji odgovara nekoj klauzi, sadrži se u  $E$  akko se literal  $l$  pojavljuje u klauzi  $c$ . Na slici 3.2 dat je prikaz grafa promenljivih i klauza za iskaznu formulu  $(p_1 \vee \neg p_3 \vee p_4) \wedge (p_2 \vee p_3 \vee \neg p_4) \wedge \neg p_4$ . Plavom bojom su obeležene grane koje spajaju literale sa prvom klauzom, crvenom bojom sa drugom klauzom i zelenom bojom sa trećom klauzom iz formule.



SLIKA 3.2: Grafički prikaz grafa promenljivih i klauza iskazne formule  $(p_1 \vee \neg p_3 \vee p_4) \wedge (p_2 \vee p_3 \vee \neg p_4) \wedge \neg p_4$ .

## Glava 4

# Odabrani metodi mašinskog učenja

U ovom poglavlju predstavljamo neke metode za predviđanje vrednosti ciljnih funkcija  $f$ , koje su nam važne u kontekstu izbora SAT rešavača iz portfolija, na primer, vreme izvršavanja SAT rešavača. Problemi predviđanja vrednosti ovakvih funkcija pripadaju klasi opštijih problema koji se u mašinskom učenju nazivaju regresioni problemi.

### 4.1 Opšte o regresiji

**Definicija 4.1** Regresija (*eng. regression*) predstavlja problem predviđanja vrednosti ciljne promenljive  $y \in \mathbf{R}^m$  u zavisnosti od datih atributa ulaznih podataka  $x \in D$ ,  $D = D_1 \times D_2 \times \dots \times D_n$ .

Neki primeri problema koji se mogu opisati kao regresioni problemi su [18, 19]:

- Analiza kredibiliteta korisnika bankarskih usluga;
- Analiza odnosa između nedeljnih prodaja proizvoda i promocija;
- Analiza zavisnosti između mesečne cene zakupnine i tipa, lokacije i uslova apartmana koji se izdaje;
- Predviđanje količine telesnih masti u odnosu na promenljive koje se jednostavno mere, poput visine, godina, težine, i dr.

Ovaj problem se rešava pronalaženjem matematičkog modela  $F$  koji, primenjen na attribute ulaznih podataka  $x$ , izračunava vrednost  $F(x)$  koja najpribližnije odgovara pravoj vrednosti  $y$ , koja nam je u trenutku predviđanja nepoznata. Drugim rečima, cilj je konstruisati model zavisnosti raspodele verovatnoće ciljne promenljive od atributa.

Vrednost  $y$  može biti skalarna vrednost, ukoliko se predviđa samo jedna vrednost (slučaj kada je  $m = 1$ ) ili može biti vektor vrednosti, ukoliko se predviđaju dve ili više vrednosti zajedno (slučaj kada je  $m > 1$ ). U slučaju  $m > 1$ , neki metodi imaju mogućnost predviđanja vrednosti vektora  $y \in \mathbf{R}^m$ . Sa druge strane, ukoliko je metod efikasan za konstruisanje ili ga je moguće jednostavno paralelizovati, moguće je pristupiti izradi po jednog regresionog modela  $F_1, F_2, \dots, F_m$  za svaki atribut ciljnog vektora  $y = (y_1, y_2, \dots, y_m)$ . Zatim, kada se vrši predviđanje vektora vrednosti ciljne



promenljive  $y$  instance  $x$ , konsultuju se svi regresioni modeli i predviđene vrednosti se nadovezuju u vektor predviđanja. Za neke modele je jednostavnije implementirati ovaj pristup umesto predviđanja svih vrednosti vektora  $y \in \mathbf{R}^m$  odjednom.

Problem određivanja najboljih SAT rešavača u odnosu na ciljnu funkciju  $f$  se u terminima mašinskog učenja može predstaviti kao problem regresije, gde je vrednost koja se predviđa upravo vrednost ciljne funkcije  $f$ <sup>1</sup>. Mi ćemo za ciljnu funkciju  $f$  koristiti prethodno uvedenu funkciju  $f_{vreme}$ , tj.  $f = f_{vreme}$ , te ćemo koristiti regresione metode za predviđanje vremena izvršavanja SAT rešavača.

U zavisnosti od odabira metoda, razlikuju se i matematički modeli koji se koriste za rešavanje regresionog problema. U narednim sekcijama ćemo uvesti neke od metoda koji se mogu koristiti za predviđanje vremena izvršavanja SAT rešavača. U poglavlju 5 ćemo detaljnije diskutovati o načinu upotrebe ovih metoda nad instancama SAT problema. Od metoda koje ćemo uvesti u narednim sekcijama, prva dva imaju već demonstriranu upotrebu u rešavanju ovog problema sa značajnim uspehom. Preostala tri metoda, koja su zasnovana na grafovskim neuronskim mrežama, biće eksperimentalno ispitana i poređena sa prethodna dva metoda pod istim uslovima.

Metodi zasnovani na grafovskim neuronskim mrežama su pokazali uspeh u različitim oblastima. U domenu SAT problema, izdvajamo neke uspehe ovih metoda u nastavku:

- NeuroSAT je binarni klasifikator zasnovan na grafovskoj neuronskoj mreži koji predviđa da li je instanca SAT problema zadovoljiva ili ne. Ovaj klasifikator, obučavan nad nasumično konstruisanim iskaznim formulama u KNF, pokazuje solidne prediktivne performanse nad instancama SAT problema iz domena koje nisu bile dostupne prilikom obučavanja (procenat tačnosti prijavljen u referentnom radu je 85%) [20].
- GQSAT je heuristika koja, kombinacijom pristupa učenja potkrepljivanjem i grafovske neuronske mreže, poboljšava kvalitet odabira pravila grananja na početku analize problema od strane SAT rešavača [21].
- Grafovska neuronska mreža je iskorišćena za učenje reprezentacije logičkih formula. Ova reprezentacija je testirana za potrebe ispitivanja logičkih posledica parova iskaznih formula<sup>2</sup>, kao i za predviđanje broja koraka za dokazivanje neke teoreme logike prvog reda [22].

<sup>1</sup>Ukoliko diskutujemo o portfolio SAT rešavačima, onda se vrednost ciljne funkcije  $f$  predviđa za svaki rešavač iz portfolija. Kako SAT rešavača iz portfolija ima  $m > 1$ , to je potrebno da regresioni model predviđi  $m$  vrednosti (za svaki SAT rešavač iz portfolija po jednu) koje se nadovezuju u vektor ciljnih vrednosti.

<sup>2</sup>Za par iskaznih formula  $(F, G)$  kažemo da je iskazna formula  $G$  *logička posledica* iskazne formule  $F$  ako i samo ako je svaki model za iskaznu formulu  $F$  istovremeno i model za iskaznu formulu  $G$ . U tom slučaju, par iskaznih formula  $(F, G)$  pripada klasi 1. U suprotnom, taj par pripada klasi 0.

### 4.1.1 Ukratko o klasifikaciji

Za potrebe razmatranja konstrukcije portfolio sistema za rešavanje SAT problema, biće nam značajno da uvedemo pojam klasifikacije iz mašinskog učenja.

**Definicija 4.2** Klasifikacija (*eng. classification*) predstavlja problem predviđanja vrednosti ciljne promenljive  $c \in C$ , gde je, bez gubitka na opštosti,  $C = \{1, 2, \dots, J\}$ , u zavisnosti od datih atributa ulaznih podataka  $x \in D$ ,  $D = D_1 \times D_2 \times \dots \times D_n$ .

Vrednost  $c$  koja se predviđa, koja se naziva *klasa* ili *oznaka*, predstavlja vrednost iz nekog unapred poznatog, diskretnog i konačnog skupa  $C$ .

Neki primeri problema koji se mogu opisati kao klasifikacioni problemi su [3]:

- Dijagnostifikacija bolesti u oblasti medicine;
- Detekcija da li je neka transakcija pomoću kreditne kartice prevara ili ne;
- Odlučivanje da li treba promovisati proizvod određenom kupcu ili ne.

Klasifikacioni problemi se rešavaju na isti način kao i regresioni, sa razlikom u odabiru matematičkog modela ili neke njegove varijante. Naime, često se isti matematički model može koristiti i za klasifikaciju i za regresiju, sa manjim implementacionim razlikama u izlazu algoritama tih metoda.

Pojam klasifikacije nam je značajan iz razloga što određivanje najboljeg SAT rešavača u odnosu na ciljnu funkciju  $f$  se može definisati i kao klasifikacioni problem. Naime, moguće je svakom SAT rešavaču iz portfolija dodeliti po jednu klasu  $c_i$ , pa na osnovu njega konstruisati klasifikacioni matematički model  $F$  koji će predviđati onu klasu rešavača koja je najbolja u odnosu na ciljnu funkciju  $f$  za datu instancu. Ipak, često se dešava da više SAT rešavača imaju dovoljno približne vrednosti ciljne funkcije  $f$ , pa bi u tom slučaju bilo pogodnije prijaviti skup SAT rešavača odabranih kao najbolje. Zbog toga se i ograničavamo na definisanje ovog problema u terminima regresionih problema.

## 4.2 $K$ -najbližih suseda

Jedan od najelementarnijih regresionih metoda jeste metod *k-najbližih suseda* (*eng. k-nearest neighbors*, skr. *KNN*). Glavna ideja algoritma koji implementira ovaj metod se sastoji u pronalaženju  $k$  instanci  $x_1, x_2, \dots, x_k$  iz skupa za obučavanje koji su „najbliži” instanci  $x$  za koju se vrši predviđanje nepoznate vrednosti  $y$ . Pojam „bliskosti” se definiše izračunavanjem nekog rastojanja  $d(x, x_i)$  u prostoru atributa instanci  $D$ . Rastojanja koja ćemo mi koristiti su [23]:

- Familija  $L_p$  rastojanja Minkovskog. Primer rastojanja iz ove familije je Euklidsko rastojanje, koje se dobija za  $p = 2$  u narednoj formuli:

$$d(x, y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{1/p} \quad (4.1)$$

- Familija  $L_1$  mera rastojanja. Primer rastojanja iz ove familije je rastojanje Lorenca:

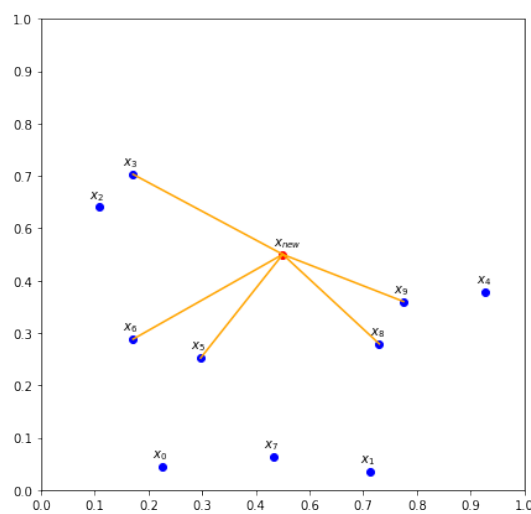
$$d(x, y) = \sum_{i=1}^n \ln(1 + |x_i - y_i|) \quad (4.2)$$

- Familija mera sličnosti zasnovanih na skalarnom proizvodu. Primer je ugaona sličnost:

$$s(x, y) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (4.3)$$

Atribut 1	Atribut 2	Vrednost ciljne promenljive
0.25	0.05	0.61
0.68	0.04	0.73
0.12	0.60	0.42
0.19	0.67	0.39
0.92	0.42	0.23
0.33	0.28	0.44
0.19	0.32	0.37
0.48	0.07	0.58
0.70	0.31	0.10
0.75	0.40	0.74

TABELA 4.1: Skup podataka na osnovu kojeg se predviđa 5-najbližih suseda za instancu  $x_{new} = (0.5, 0.5)$  sa slike 4.1. Prve dve kolone čine attribute instance, dok je poslednja kolona ciljna promenljiva koja se predviđa.



SLIKA 4.1: Primer određivanja 5-najbližih suseda za skup podataka iz tabele 4.1.

Na slici 4.1 prikazan je primer određivanja 5-najbližih suseda za instancu označenu  $x_{new} = (0.5, 0.5)$ , u smislu Euklidskog rastojanja, za dati skup podataka iz tabele 4.1.

U slučaju regresije, nakon što se pronađu  $k$  najbližih instanci  $x_1, x_2, \dots, x_k$ , primenjuje se neka agregatna funkcija na vrednosti ciljne promenljive  $y_1, y_2, \dots, y_k$ , najčešće prosek, koja određuje vrednost koja se prijavljuje kao predviđena, odnosno:

$$F(x) = \frac{1}{n} \sum_{i=1}^k y_i$$

### 4.3 Šuma nasumičnih stabala

Šuma nasumičnih stabala (eng. *random forests*, skr. *RF*) predstavlja jedan od tzv. metoda *ansambla* (eng. *ensemble*). Ideja ovakvih metoda jeste izgradnja velikog broja relativno jednostavnih metoda, pri čemu se svaki konstruisani metod koristi u cilju poboljšanja do sada konstruisanih metoda. Ovo „poboljšanje” se može izvršiti na nekoliko načina, kao što su:

- Odabir podskupa skupa atributa  $\{D_1, D_2, \dots, D_n\}$  na osnovu kojih se metod konstruiše.
- Odabir različitih parametara metoda.
- Odabir podskupa skupa instanci  $\{x_1, x_2, \dots, x_N\}$  na osnovu kojih se metod konstruiše.

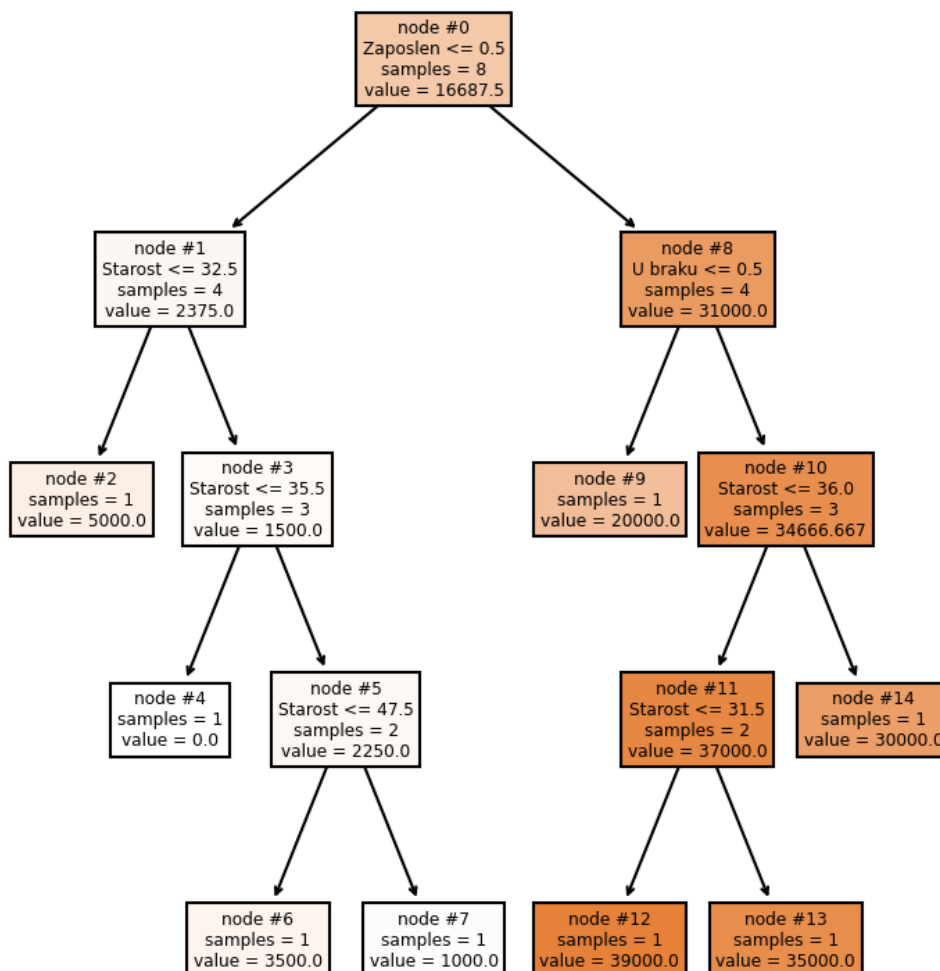
Metod šume nasumičnih stabala konstruiše skup metoda koji se nazivaju *stabla odlučivanja*. Nakon konstrukcije datog skupa stabala odlučivanja, koristi se pristup zasnovan na *agregaciji* (eng. *bagging*) radi popravljivanja predikcije i kontrole preprilagodavanja. Osnovna ideja agregacije je sledeća [4]:

- Kreirati podskupove  $X_1, X_2, \dots, X_t$  jednake kardinalnosti skupa instanci  $X$  metodom slučajnog uzorkovanja sa ponavljanjem.
- Za svaki podskup  $X_i$  konstruisati po jedan regresioni model  $F_i$ .
- Prilikom predviđanja vrednosti ciljne promenljive za novu instancu  $x$ , predvideti vrednost ciljne promenljive pomoću svakog regresionog modela  $F_i$ , čime se dobija skup predviđenih vrednosti  $\{y_1, y_2, \dots, y_t\}$ .
- Prijaviti vrednost  $\frac{1}{n} \sum_{i=1}^t y_i$  kao predviđenu vrednost za instancu  $x$ .

Predimo sada na opis metoda za konstrukciju regresionog metoda koji se koristi u agregacionom pristupu šuma nasumičnih stabala.

### 4.3.1 Stablo odlučivanja

*Stablo odlučivanja* (eng. *decision tree*) predstavlja drvoliku strukturu u čijim unutrašnjim čvorovima se nalaze *uslovi atributa*, dok se u listovima nalaze predviđene vrednosti.



SLIKA 4.2: Primer konstruisanog stabla odlučivanja za skup podataka iz tabele 4.2.

Starost	Zaposlen	U braku	Odobrena suma kredita
30	0	1	5000
33	1	1	35000
35	1	0	20000
30	1	1	39000
59	0	1	1000
35	0	0	0
36	0	1	35000
39	1	1	30000

TABELA 4.2: Skup podataka na osnovu kojeg je izgrađeno stablo odlučivanja sa slike 4.2. Prve tri kolone čine attribute instance, dok je poslednja kolona ciljna promenljiva koja se predviđa.

Na slici 4.2 je dat primer jednog stabla odlučivanja za skup podataka iz tabele 4.2. Prikažimo kako će prikazano stablo predvideti vrednost nepoznate instance  $x_{new} = (40, 1, 1)$ :

- Pretraga započinje u korenu stabla. tj. u čvoru 0. U njemu se nalazi uslov „Zaposlen  $\leq 0.5$ ”. S obzirom da instanca  $x_{new}$  ne ispunjava uslov, pretraga prelazi u čvor 8.
- U čvoru 8 se nalazi uslov „U braku  $\leq 0.5$ ”. S obzirom da instanca  $x_{new}$  ne ispunjava uslov, pretraga prelazi u čvor 10.
- U čvoru 10 se nalazi uslov „Starost  $\leq 36$ ”. S obzirom da instanca  $x_{new}$  ne ispunjava uslov, pretraga prelazi u čvor 14.
- Kako je čvor 14 list stabla, onda se kao predviđena vrednost prijavljuje vrednost u tom čvoru, a to je 30000, tj.  $F(x_{new}) = 30000$ .

Postoji više algoritama pomoću kojih je moguće konstruisati stablo odlučivanja. Jedan takav algoritam je poznat pod akronimom CART (Classification and Regression Trees). Ovaj algoritam konstruiše isključivo binarna stabla odlučivanja počevši od jednog čvora (korena). Nakon formiranja uslova grananja, algoritam rekurzivno pronalazi uslove grananja u korenima podstabala tekućeg čvora, sve dok se ne dođe do listova. Uslov grananja u slučaju klasifikacije je određen merom *Ginijevog indeksa*, u oznaci  $Gini_{split}$ , čija je formula data sa [24]:

$$Gini(S) = 1 - \sum_{i=1}^{|C|} p_i^2$$

$$Gini_{split} = \sum_{i=1}^k \frac{n_i}{n} Gini(S_i)$$

Objasnimo i oznake iz prethodnih formula:  $S$  je skup podataka,  $|C|$  je broj klasa,  $s_i$  je broj instanci koji pripadaju klasi  $c_i$ ,  $p_i = s_i/S$  je frekvencija pojavljivanja klase  $c_i$  u skupu  $S$ ,  $k$  je broj podela skupa  $S$  na podskupove  $S_i$ ,  $n_i$  je broj instanci u podskupu  $S_i$  nakon podele i  $n$  je broj instanci u tekućem čvoru.

U slučaju regresije, CART algoritam minimizuje meru *srednjekvadratne greške* 6.1 kako bi pronašao najpogodnije grananje za tekući čvor.

## 4.4 Neuronske mreže

S obzirom da metodi zasnovani na grafovskim neuronskim mrežama koje ćemo opisati koriste neuronske mreže za potrebe svoje implementacije, u ovom odeljku ćemo dati uvodne pojmove vezane za dve vrste neuronskih mreža: *neuronske mreže sa propagacijom unapred* [25] i *konvolutivne neuronske mreže* [26].

### 4.4.1 Neuronske mreže sa propagacijom unapred

*Neuronska mreža sa propagacijom unapred* (eng. *feedforward neural network*), nadalje samo *neuronska mreža*, predstavlja najelementarniji vid neuronskih mreža. Jedna neuronska mreža se opisuje direktnim usmerenim grafom  $G = (V, E)$  i funkcijom težina  $w : E \rightarrow \mathbf{R}$ . Čvorove ovakvog grafa nazivamo *neuroni* (eng. *neuron*). Svaki neuron je modeliran funkcijom  $\sigma : \mathbf{R} \rightarrow \mathbf{R}$ , koju nazivamo *aktivaciona funkcija* (eng. *activation function*). Zbog toga, jedna neuronska mreža predstavlja uređenu četvorku  $(V, E, w, \sigma)$ .

Za graf  $G$  važi da se skup čvorova  $V$  može razbiti u pozitivan broj disjunktih i nepraznih podskupova čvorova koje nazivamo *slojevi* (eng. *layer*), tj.  $V = \bigsqcup_{t=0}^T V_t$ , i svaka grana iz  $E$  je oblika  $(\bar{u}_t, \bar{u}_{t+1})$ , gde je  $\bar{u}_t$  oznaka za neki čvor iz  $V_t$ , za  $t = \{0, 1, \dots, T-1\}$ . Sloj  $V_0$  se naziva *ulazni sloj*; slojevi  $V_1, V_2, \dots, V_{T-1}$  su *skriveni slojevi*; sloj  $V_T$  je *izlazni sloj*. Često se govori o *potpuno povezanim* neuronskim mrežama u kojima postoji grana iz svakog čvora u sloju  $t$  ka svakom čvoru u sloju  $t+1$ .

Broj neurona u svakom sloju neuronske mreže, zajedno sa aktivacionom funkcijom, određuje arhitekturu te neuronske mreže. U slučaju ulaznog sloja, očekuje se da broj neurona odgovara dimenziji ulaznih podataka  $n$ . Za svaki neuron  $i$  iz  $V_0$ , izlaz tog neurona je upravo  $x_i$ . Dodatno, često se dodaje jedan konstantan neuron čiji je izlaz 1, te je veličina sloja  $V_0$  jednaka  $n+1$ . Dimenzije skrivenih slojeva mogu biti različite, dok dimenzija izlaznog sloja odgovara broju vrednosti koje neuronska mreža predviđa.

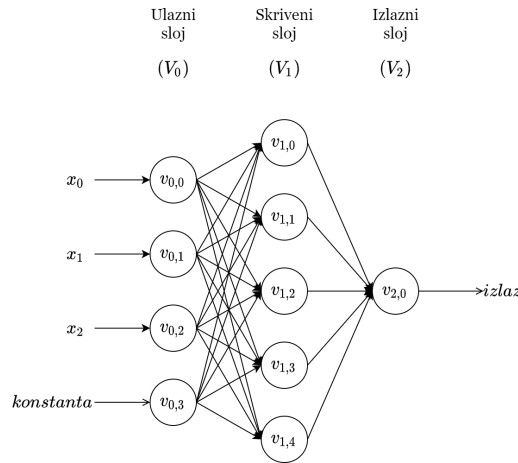
Neka  $v_{t,i}$  predstavlja  $i$ -ti neuron u sloju  $t$  i neka  $o_{t,i}(x)$  predstavlja izlaz neurona  $v_{t,i}$  za neki vektor  $x$  koji se mreži daje na ulazu. Da bi neuronska mreža izračunala vrednost izlaza neurona na sloju  $t+1$ , potrebno je da izračuna vrednost izlaza neurona na sloju  $t$ . Time se izračunavanje izlaza nekog neurona može definisati narednim formulama [25]:

$$a_{t+1,j}(x) = \sum_{r:(v_{t,r}, v_{t+1,j}) \in E} w((v_{t,r}, v_{t+1,j})) o_{t,r}(x)$$

$$o_{t+1,j}(x) = \sigma(a_{t+1,j}(x)),$$

za sve  $t \in \{0, 1, \dots, T-1\}$ . Specijalno, važi  $o_{0,i}(x) = x_i$  za  $i \in \{0, 1, \dots, n-1\}$  i  $o_{0,n}(x) = 1$ . Oznaka  $a_{t+1,j}(x)$  označava ulaz neurona  $v_{t+1,j}(x)$  za neki ulazni vektor  $x$ .

Na slici 4.3 dat je grafički prikaz jedne potpuno povezane neuronske mreže za koju važi da je  $T = 3$  i koja prihvata vektor dimenzije 3 i predviđa jednu vrednost.

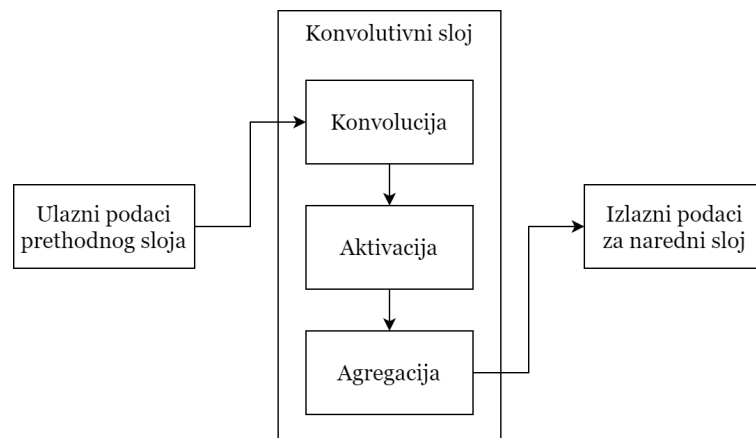


SLIKA 4.3: Grafički prikaz jedne potpuno povezane neuronske mreže.

#### 4.4.2 Konvolutivne neuronske mreže

*Konvolutivne neuronske mreže* (eng. *convolutional neural network*) predstavljaju varijantu neuronskih mreža koje su veliku popularnost stekle u praktičnim primenama na podacima koji su prirodno organizovani kao *mreže* (eng. *grid*), kao što su, na primer, zapisi zvuka (1-dimenzionalna mreža uzoraka zvuka) ili slike (2-dimenzionalna mreža piksela).

Konvolutivne neuronske mreže se sastoje od konvolutivnih slojeva u sličnoj meri kao što se neuronske mreže sastoje od skrivenih slojeva. Razlika između ove dve vrste neuronskih mreža jeste u operacijama koje se izvršavaju u njihovim slojevima. Neuronsku mrežu odlikuju množenja matrica, dok konvolutivne mreže odlikuju operacije *konvolucije* i *agregacije*, koje ćemo predstaviti u nastavku. Napomenimo da se jedan konvolutivni sloj sastoji od primene jedne operacije konvolucije, koju prati aktivaciona funkcija, a na koju se zatim primenjuje operacija agregacije, kao što je prikazano na slici 4.4 [26].



SLIKA 4.4: Grafički prikaz toka podataka kroz jedan konvolutivni sloj u konvolutivnoj neuronskoj mreži.



## Konvolucija

U opštem slučaju, *konvolucija* (eng. *convolution*) predstavlja operaciju nad dvema funkcijama  $f$  i  $g$  sa realnim parametrima:

$$s(x) = (f * g)(x),$$

U terminologiji konvolutivnih neuronskih mreža, funkcija  $f$  se naziva *signal* (eng. *input*), dok se funkcija  $g$  naziva *filter* (eng. *kernel*). U praktičnim primenama, signal predstavlja kao višedimenzioni niz podataka, dok filter predstavlja višedimenzioni niz parametara koji se uče prilikom obučavanja mreže. Ovi višedimenzioni nizovi se nazivaju *tenzorima* (eng. *tensor*). Na primer, u slučaju obrade zvuka, konvolucija se definiše formulom:

$$S(i) = (f * g)(i) = \sum_n f(n)g(i - n). \quad (4.4)$$

Sa druge strane, u slučaju obrade slika, konvolucija se definiše formulom:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (4.5)$$

gde je  $I$  slika, a  $K$  dvodimenzionalni filter. Primitimo da je u formulama 4.4 i 4.5 filter obrnut u odnosu na ulaz. Ovo se radi da bi konvolucija imala svojstvo komutativnosti. Na primer, u slučaju obrade slika:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n). \quad (4.6)$$

Formula 4.6 se jednostavnije implementira, pa se češće i koristi. Međutim, svojstvo komutativnosti konvolucije je značajnije u teorijskim izvođenjima dokaza više nego u praksi. Zbog toga, razne biblioteke koje implementiraju konvoluciju, zapravo implementiraju operaciju *unakrsne korelacije* (eng. *cross-correlation*):

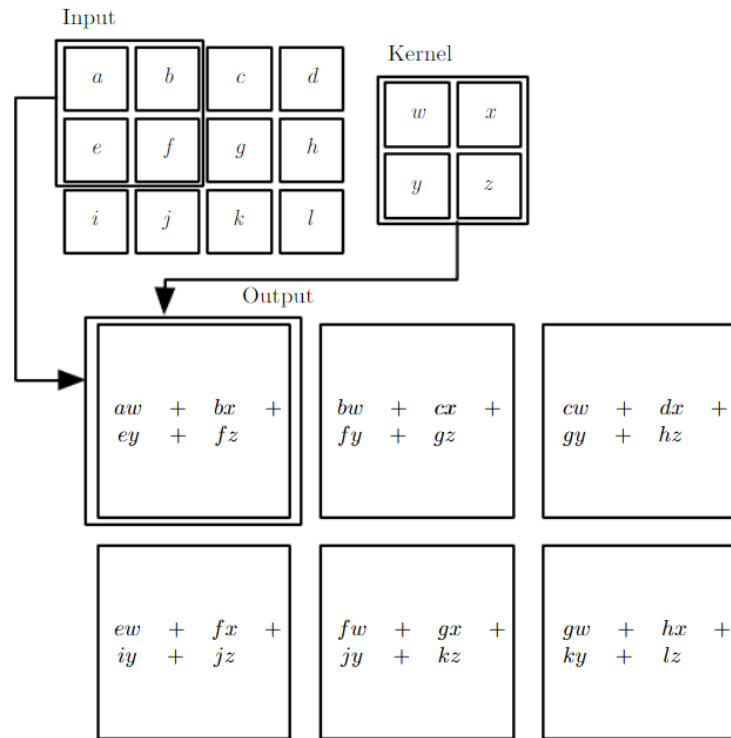
$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n). \quad (4.7)$$

Na slici 4.5 data je ilustracija izvršavanja konvolucije (u smislu formule 4.7) nad slikom dimenzija  $3 \times 4$  i filterom  $2 \times 2$ . Primitimo da je rezultat konvolucije ponovo slika, ali ovoga puta dimenzija  $2 \times 3$  [26].

## Agregacija

Nakon što se rezultati primene konvolucije propuste kroz aktivacionu funkciju, konvolutivni sloj koristi neku *agregaciju* (eng. *pooling*) u cilju agregiranja do tada stečenih informacija u tenzoru koji predstavlja izlaz nakon aktivacione funkcije. Agregacija je funkcija koja zamenjuje sve vrednosti u nekom podtenzoru jednom vrednošću (dakle,

agregira informacije). Jedan čest primer jeste agregacija maksimumom, koja zamenjuje odgovarajuće vrednosti najvećom vrednošću među njima. Neke druge tehnike agregacije su: uprosečavanje, zamena  $L_2$ -normom podtenzora ili težinsko uprosečavanje u smislu rastojanja od centra podtenzora. Agregacija je korisna operacija koja omogućava aproksimativnu invarijantnost na male promene u ulaznim podacima, kao i zbog toga što smanjuje broj parametara u mreži [26].



SLIKA 4.5: Ilustracija izvršavanja operacije konvolucije u dvodimenzionalnom slučaju [26].

## 4.5 Metodi zasnovani na grafovskim neuronskim mrežama

Metodi  $k$ -najbližih suseda i šuma nasumičnih stabala operišu nad podacima u vektorskom zapisu, bez uzimanja u obzir dodatnih informacija o tim podacima koje dolaze iz njihove strukture. Za očekivati je da podaci, koji se prirodno predstavljaju grafovima, sadrže skrivene obrasce u samoj strukturi tih grafova koje nose značajnu informaciju. Grafovske mreže operišu nad grafovskim reprezentacijama podataka u cilju pronalaženja upravo onih informacija koje se kriju u njihovoj strukturi. U ovom odeljku ćemo predstaviti tri varijante grafovskih neuronskih mreža: *grafovske konvolutivne mreže* [7], *grafovske mreže sa mehanizmom pažnje* [8] i *duboke grafovske konvolutivne neuronske mreže* [9].

### 4.5.1 Grafovske konvolutivne mreže

*Grafovske konvolutivne mreže* (eng. *graph convolutional network*, skr. *GCN*) direktno kodiraju strukturu grafa korišćenjem diferencijabilne neuronske mreže  $f(X, A)$ , koja uči optimizacijom funkcije cilja za sve čvorove u grafu.

#### Sloj konvolucije nad grafom

Srž upotrebe grafovskih konvolutivnih neuronskih mreža se sastoji u pravilu propagacije koje definiše način za izračunavanje atributa narednog sloja mreže  $l + 1$  u zavisnosti od prethodnog sloja  $l$  mreže narednom formulom [7]:

$$H^{(l+1)} = \sigma \left( \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \quad (4.8)$$

gde je  $\tilde{A} = A + I_N$  matrica susedstva grafa  $G = (V, E)$  sa dodatim petljama za svaki čvor,  $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ ,  $W^{(l)}$  je matrica parametara modela za sloj  $l$ ,  $\sigma$  je aktivaciona funkcija i  $H^{(l)} \in \mathbf{R}^{N \times d}$  matrica aktivacija u sloju  $l$ ; važi  $H^{(0)} = X$ . Svaka primena pravila propagacije koristi skrivene attribute čvorova grafa iz prethodnog sloja  $H^{(l)}$  kako bi propagirala te vrednosti njihovim susedima (množenje simetrično-normalizovanom matricom susedstva  $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ ). Dodatno, množenje matricom naučenih parametara modela  $W^{(l)}$  omogućava neuronskoj mreži da odabere samo one vrednosti atributa iz prethodnog sloja za koje je naučila da sadrže korisne informacije.

Ovo pravilo propagacije opisuje jedan sloj konvolucije. Da bi se konstruisala duboka neuronska mreža, potrebno je nadovezati nekoliko slojeva konvolucije. Upravo zbog ove činjenice se matrica susedstva normalizuje, kako bi proces optimizacije vrednosti kroz duboku neuronsku mrežu bio stabilniji<sup>3</sup>. Jedan primer modela grafovske konvolutivne mreže sa 2 sloja je:

$$Z = f(X, A) = \text{softmax} \left( \hat{A} \text{ReLU} \left( \hat{A} X W^{(0)} \right) W^{(1)} \right) \quad (4.9)$$

gde je  $\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ . Matrica  $W^{(0)} \in \mathbf{R}^{d \times h^{(0)}}$  je matrica parametara za sloj koji spaja ulazne podatke dimenzije  $d$  i podatke prvog skrivenog sloja dimenzije  $h^{(0)}$ . Slično,  $W^{(1)} \in \mathbf{R}^{h^{(0)} \times h^{(1)}}$  je matrica parametara za sloj koji spaja skrivene podatke i izlazne podatke. Na izlazu iz sloja 0, primenjuje se aktivaciona funkcija ReLU, dok se na izlazu iz sloja 1 primenjuje aktivaciona funkcija softmax, definisane formulama:

$$\text{ReLU}(x) = \max(0, x) \quad (4.10)$$

$$\text{softmax}(x) = \frac{\exp(x)}{\sum_i \exp(x_i)} \quad (4.11)$$

Za potrebe višeklasne klasifikacije čvorova, funkcija cilja je *unakrsna entropija* (eng. *cross-entropy*), definisana formulom:

<sup>3</sup>Najveća sopstvena vrednost simetrično-normalizovane matrice susedstva je 1.

$$\mathcal{L} = - \sum_{l \in \mathcal{Y}_L} \sum_{f=1}^F Y_{lf} \ln Z_{lf}$$

gde je  $\mathcal{Y}_L$  skup indeksa čvorova koji imaju oznake.

### 4.5.2 Grafovske mreže sa mehanizmom pažnje

*Grafovske mreže sa mehanizmom pažnje* (eng. *graph attention network*, skr. *GAT*) predstavljaju koncept sličan grafovskim konvolutivnim mrežama, u kojem je glavna ideja izračunavanje skrivenih reprezentacija svakog čvora u grafu pomoću obilaska njegovih suseda. Međutim, za razliku od grafovskih konvolutivnih mreža, grafovske mreže sa mehanizmom pažnje, kao što i ime govori, uvode mehanizam pažnje. Ovaj mehanizam se primenjuje nad tekućim reprezentacijama suseda čvorova u cilju izračunavanja nove reprezentacije tog čvora. Slično kao i u prethodnoj sekciji, opisaćemo način na koji funkcioniše jedan sloj sa mehanizmom pažnje na način na koji je izložen u originalnom radu [8].

#### Mehanizam pažnje nad grafom

Ulazni podaci za jedan sloj sa mehanizmom pažnje čine skup  $h = \{\vec{h}_1, \vec{h}_2, \dots, \vec{h}_N\}$ ,  $\vec{h}_i \in \mathbf{R}^F$ , gde je  $N$  broj čvorova grafa, a  $F$  je dimenzija atributa za svaki čvor. Izlazni podaci ovog sloja čine novi skup atributa, potencijalno različite dimenzije  $F'$ ,  $h' = \{\vec{h}'_1, \vec{h}'_2, \dots, \vec{h}'_N\}$ ,  $\vec{h}'_i \in \mathbf{R}^{F'}$ .

Postupak za izračunavanje *koeficijenata pažnje* (eng. *attention coefficients*) dat je narednom formulom:

$$e_{ij} = a(W\vec{h}_i, W\vec{h}_j) \quad (4.12)$$

gde je  $W \in \mathbf{R}^{F' \times F}$  parametarska linearna transformacija ulaznih atributa i funkcija  $a : \mathbf{R}^{F'} \times \mathbf{R}^{F'} \rightarrow \mathbf{R}$  prethodno pomenuti mehanizam pažnje. Koeficijent pažnje određuje „važnost” atributa  $j$ -tog čvora za  $i$ -ti čvor.

Važno je napomenuti da se koeficijenti pažnje za neki čvor  $i$  izračunavaju samo za one čvorove  $j \in N_i$ , gde je  $N_i$  skup suseda čvora  $i$  (uključujući i njega). Dodatno, izvršava se postprocesiranje koeficijenata 4.12 funkcijom softmax kako bi se dobili *normalizovani koeficijenti pažnje* (eng. *normalised attention coefficients*):

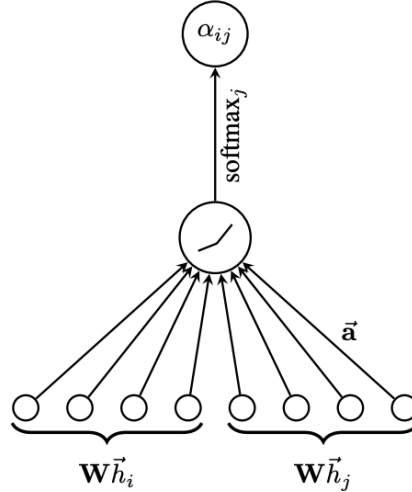
$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})} \quad (4.13)$$

Na slici 4.6 dat je grafički prikaz opisane operacije izračunavanja normalizovanih koeficijenata pažnje.

Za odabir mehanizma pažnje  $a$  koristi se jednoslojna neuronska mreža, parametrizovana vektorom težina  $\vec{a} \in \mathbf{R}^{2F'}$ , na koju se primenjuje aktivaciona funkcija LeakyReLU definisana formulom:

$$\text{LeakyReLU}(x) = \max(0, x) + \alpha \cdot \min(0, x) \quad (4.14)$$

pri čemu se za parametar negativnog nagiba ( $\alpha$ ) bira 0.2.



SLIKA 4.6: Mehanizam pažnje u grafovskoj mreži sa mehanizmom pažnje [8].

Nakon izračunavanja normalizovanih koeficijenata pažnje 4.13, izlazni atributi se računaju formulom:

$$\vec{h}'_i = \sigma \left( \sum_{j \in N_i} \alpha_{ij} W \vec{h}_j \right) \quad (4.15)$$

gde je  $\sigma$  neka nelinearna aktivaciona funkcija. Dodatno, radi stabilizacije procesa učenja pažnje, korisno je uključiti mehanizam *višestruke pažnje* (eng. *multi-head attention*) u kojem  $K$  nezavisnih mehanizama pažnje izvršavaju transformaciju 4.15, nakon čega se izlazni atributi nadovezuju, dobijajući narednu reprezentaciju izlaznih atributa:

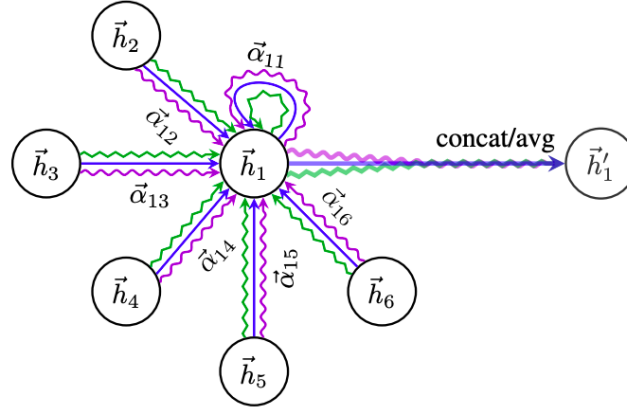
$$\vec{h}'_i = \parallel_{k=1}^K \sigma \left( \sum_{j \in N_i} \alpha_{ij}^k W^k \vec{h}_j \right) \quad (4.16)$$

gde je  $\parallel$  operacija nadovezivanja,  $\alpha_{ij}^k$  su normalizovani koeficijenti pažnje izračunati pomoću  $k$ -tog mehanizma pažnje  $a^k$  i  $W^k$  je odgovarajuća matrica težina ulazne linearne transformacije. Primetimo da će, u ovom slučaju, dimenzija izlaznih atributa  $h'$  biti  $K F'$  umesto  $F'$ , kako je to bilo u slučaju operacije 4.15.

Posebno, u slučaju izvršavanja mehanizma višestruke pažnje u poslednjem (predikcionom) sloju neuronske mreže umesto nadovezivanja vrši se uprosečavanje, kako bi se dobio izlaz odgovarajuće dimenzije, nakon čega se vrši konačna nelinearna transformacija (na primer, funkcijom softmax u slučaju klasifikacije):

$$\vec{h}'_i = \sigma \left( \frac{1}{K} \sum_{k=1}^K \sum_{j \in N_i} \alpha_{ij}^k W^k \vec{h}_j \right)$$

Na slici 4.7 dat je grafički prikaz procesa višestruke pažnje za čvor sa oznakom 1. Različite linije po granama prikazuju različite mehanizme pažnje. Kao što vidimo, nakon izračunavanja normalizovanih koeficijenata pažnje, dobijene vrednosti se nadovezuju (concat) ili uprosečavaju (avg) kako bi se dobili izlazni atributi  $\vec{h}'_1$ .



SLIKA 4.7: Ilustracija mehanizma višestruke pažnje ( $K = 3$ ) za čvor 1 u njegovom susedstvu  $N_1$  [8].

### 4.5.3 Klasifikacija i regresija nad grafom

Već smo napomenuli da opisani pristupi proizvode izlazne attribute za svaki čvor u grafu nakon izračunavanja poslednjeg sloja, bilo u slučaju konvolutivnog sloja ili sloja sa mehanizmom pažnje. Postavlja se pitanje kako je moguće iskoristiti ovu informaciju za predviđanje jedne osobine (ili više njih) za ceo graf. U ovom radu ćemo predstaviti dva pristupa. Prvi pristup jednostavno prilagođava opisane metode za potrebe predviđanja nad grafovima čime se dobijaju prva dva metoda za eksperimentalnu evaluaciju, dok drugi pristup uvodi treći i finalni metod koji će biti eksperimentalno evaluiran.

#### Kombinacija sa neuronskom mrežom sa propagacijom unapred

U ovoj podsekciji ćemo prikazati postupak za slučaj grafovske konvolutivne mreže, ali ista procedura se jednako koristi i u slučaju grafovske mreže sa mehanizmom pažnje.

Pokažimo kako je moguće izmeniti definiciju modela 4.9 tako da se pomoću njega može vršiti klasifikacija, odnosno, regresija čitavog grafa, umesto pojedinačnih čvorova. Formula modela kojim je ovo moguće postići je:

$$Z = f(X, A) = \mu \left( p \left( \hat{A} \operatorname{ReLU} \left( \hat{A} X W^{(0)} \right) W^{(1)} \right) \right) \quad (4.17)$$

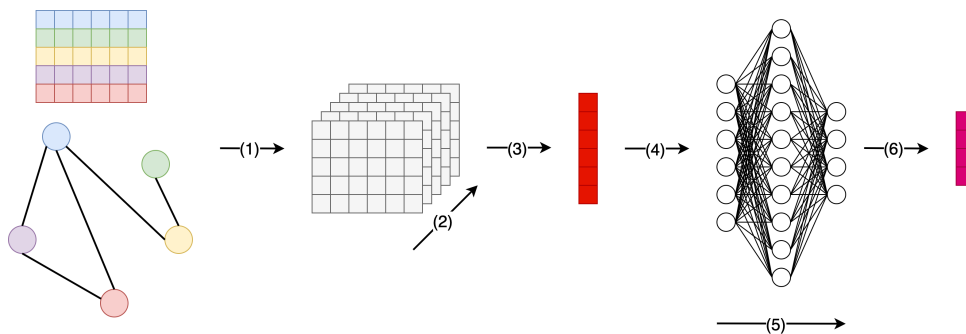
U formuli 4.17 vidimo naredna dva nova elementa:

- Funkcija  $p$  predstavlja funkciju *globalne agregacije* (eng. *global pooling* ili *graph pooling*), koja sažima naučene vektorske reprezentacije poslednjeg skrivenog sloja u jedan vektor. Jedna takva funkcija jeste *globalna prosečna agregacija* (eng. *global average pooling*), definisana formulom:

$$r^{(i)} = \frac{1}{N} \sum_{k=1}^N x_k^{(i)}, \text{ za svako } i \in \{0, 1, \dots, h^{(l-1)}\} \quad (4.18)$$

Kao što vidimo, rezultat globalne agregacije je vektor čija dimenzija odgovara dimenziji poslednjeg skrivenog sloja  $h^{(l-1)}$  iz grafovske konvolutivne mreže (u modelu 4.17 iznad:  $h^{(l-1)} = h^{(1)}$ ).

- Funkcija  $\mu$  predstavlja diferencijabilnu potpuno povezanu neuronsku mrežu (sa najčešće jednim slojem). Dimenzija ulaznih parametara ove neuronske mreže odgovara dimenziji vektora iz globalne agregacije, dok dimenzija njenih izlaznih parametara odgovara dimenziji koja se predviđa. U slučaju višeklasne klasifikacije, to je broj klasa, a u slučaju višeciljne regresije, to je broj vrednosti koji se predviđa.



SLIKA 4.8: Ilustracija modela grafovske konvolutivne mreže za klasifikaciju/regresiju grafa.

Primitimo da smo na izlazu iz sloja 0 u formuli 4.17 koristili aktivacionu funkciju ReLU, kako bismo napravili paralelu sa formulom 4.9. U praksi, naravno, moguće je odlučiti se za proizvoljnu aktivacionu funkciju, kako i pravilo propagacije 4.8 u svom opštem obliku i nalaže.

Na slici 4.8 dat je grafički prikaz modela grafovske konvolutivne mreže za klasifikaciju/regresiju grafa. Operacije koje su na ovoj slici označene brojevima su:

1. Ulaz u grafovsku konvolutivnu mrežu.
2. Primena pravila propagacije atributa kroz slojeve konvolucije, tj. konstruisanje skrivenih atributa.
3. Globalna agregacija.
4. Ulaz u neuronsku mrežu sa propagacijom unapred.

5. Primena pravila propagacije kroz potpuno povezanu neuronsku mrežu.
6. Izlaz, tj. prijavljivanje predviđenih vrednosti.

### Duboka grafovska konvolutivna neuronska mreža

Odabir jednostavne globalne agregacije poput uprosečavanja vodi ka gubitku informacija koje kodiraju čvorovi grafa pojedinačno. Zbog toga, mogli bismo se zapitati da li je moguće iskoristiti naučene skrivene reprezentacije čvorova grafa na neki drugi način. Ukoliko bismo eliminisali globalnu agregaciju, već iskoristili izlazne attribute poslednjeg skrivenog sloja (konvolucije ili mehanizma pažnje) kao ulaze za potpuno povezanu neuronsku mrežu, onda bismo uveli dva nova problema:

- Prvi problem je u tome što bismo doveli do eksplozije parametara koje je potrebno obučavati za iole veće dimenzije atributa skrivenih slojeva. Naime, u slučaju korišćenja jednostavnih globalnih agregacija poput uprosečavanja, svi čvorovi u grafu bivaju reprezentovani zajedno jednim vektorom fiksne dimenzije, koja je jednaka dimenziji atributa poslednjeg skrivenog sloja. Ukoliko bismo za svaki čvor koristili vektor atributa na poslednjem skrivenom sloju, ili nadovezali vektore atributa svih skrivenih slojeva, dimenzija ulaznog sloja neuronske mreže u narednom koraku bi bila toliko puta veća. Ovo zahteva značajno veće računarske resurse kao i vreme koje je potrebno za obučavanje modela.
- Drugi problem je fundamentalniji i sastoji se od činjenice da različiti grafovi imaju različit broj čvorova, te se broj ulaznih parametara neuronske mreže menja dinamički, što nije dopustivo.

Kako bi se donekle prevazišli ovi problemi, autori rada [9] predlažu novu arhitekturu zasnovanu na grafovskim neuronskim mrežama koja prihvata grafove proizvoljne strukture i uči klasifikacionu/regresionu funkciju nad tim podacima. Predložena arhitektura, čiji je naziv *duboka grafovska konvolutivna neuronska mreža* (eng. *deep graph convolutional neural network*, skr. *DGCNN*), sastoji se od 3 koraka:

1. U prvom koraku se koristi model grafovske neuronske mreže za dobijanje skrivenih reprezentacija za svaki čvor grafa. Za ovu potrebu autori predlažu konvolutivni sloj koji definiše pravilo propagacije formulom:

$$H^{(l+1)} = \sigma \left( \tilde{D}^{-1} \tilde{A} H^{(l)} W^{(l)} \right) \quad (4.19)$$

sa istim oznakama kao u formuli 4.8.

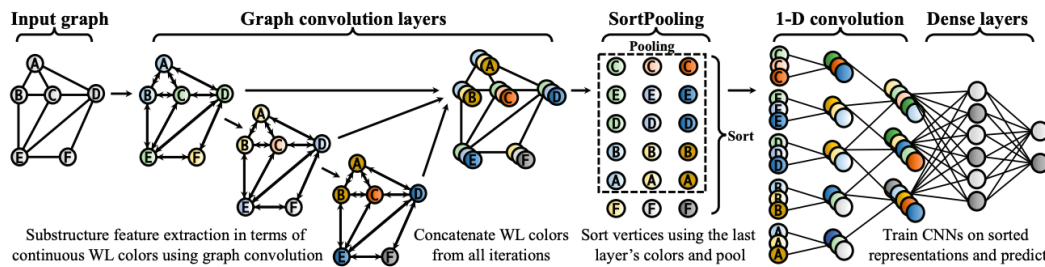
2. Drugi korak podrazumeva korišćenje novog sloja, tzv. *sloj sortiranja* (eng. *sort pooling*), koji uređuje naučene attribute (vektore) svih skrivenih slojeva u konzistentnom redosledu. Za potrebe određivanja redosleda, autori se oslanjaju na Weisfeiler-Lehman (WL) grafovski kernel, koji koristi istoimeni algoritam za izračunavanje višestrukih atributa podstabala za potrebe klasifikacije grafa [9].



Sloj sortiranja ima i jedan metaparametar  $k$ , koji određuje broj čvorova i njihovih atributa (nakon sortiranja), čime se ograničava broj ulaznih parametara za neuronsku mrežu koja vrši predikciju.

3. Treći korak podrazumeva korišćenje tradicionalne konvolutivne neuronske mreže koja izračunava informacije iz vektora atributa svakog ulaznog čvora i na osnovu njih vrši predviđanje.

Opisana arhitektura je prikazana na slici 4.9. Ulazni graf proizvoljne strukture se prosleđuje kroz nekoliko slojeva grafovske konvolucije, čime se informacija o čvorovima propagira između suseda. Zatim se atributi čvorova sortiraju i agregiraju pomoću sloja sortiranja. Nakon toga, dobija se tenzor veličine  $k \times \sum_{i=0}^{l-1} h^{(i)}$ , gde je  $h^{(i)}$  dimenzija atributa  $i$ -tog skrivenog sloja. Svaki red ovog tenzora odgovara jednom čvoru, a svaka kolona predstavlja jedan konvolutivni kanal atributa. Kako bi se nad ovakvim podacima obučavale konvolutivne neuronske mreže, potrebno je prvo promeniti dimenziju tenzora na  $k \sum_{i=0}^{l-1} h^{(i)} \times 1$ . Zatim se na ovo nadoveže jednodimenzionalni konvolutivni sloj sa veličinom filtera i korakom dužine  $\sum_{i=0}^{l-1} h^{(i)}$  (dakle, takvih slojeva ima  $k$ ). Ovaj sloj prate tradicionalni slojevi *agregacije maksimumom* (eng. *maximum pooling*) i jednodimenzionalnih konvolucija kako bi se naučili lokalni obrasci za svaki vektor atributa čvora. Ovu konvolutivnu neuronsku mrežu prati sloj potpuno povezane neuronske mreže koja predviđa željene osobine grafa. U slučaju klasifikacije, izlaze je potrebno procesirati nekom nelinearnom transformacijom poput funkcije softmax.



SLIKA 4.9: Grafički prikaz arhitekture duboke grafovske konvolutivne neuronske mreže [9].

## Glava 5

# Sistem za obučavanje portfolio SAT rešavača

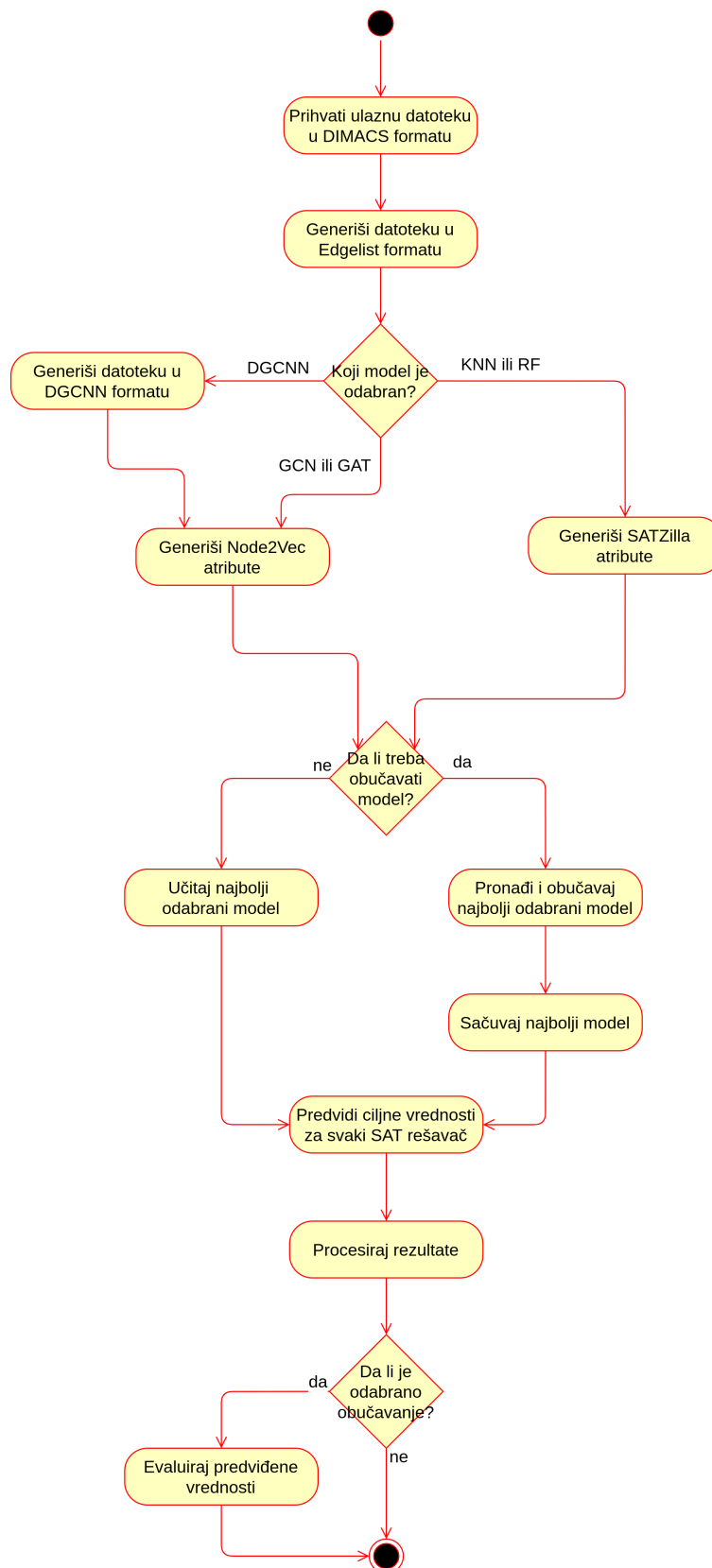
U ovoj glavi opisaćemo arhitekturu aplikacije koja je razvijena u sklopu rada. Aplikacija služi za obučavanje opisanih metoda iz prethodnog poglavlja, njihovu evaluaciju i predviđanje vremena izvršavanja SAT rešavača koristeći obučene modele.

### 5.1 Funkcionalnosti aplikacije

Aplikacija koja je razvijena uz ovaj rad implementira naredne funkcionalnosti:

- Priprema ulaznih podataka – instanci SAT problema – koji su zapisani u DIMACS formatu.
- Obučavanje modela koji su zasnovani na metodima opisanim u prethodnoj glavi u cilju predviđanja vremena izvršavanja SAT rešavača iz unapred fiksiranog portfolija. Portfolio se sastoji od 31 SAT rešavača: `ebglucose`, `ebminisat`, `glucose2`, `glueminisat`, `lingeling`, `lrglshr`, `minisatpsm`, `mphaseSAT64`, `precosat`, `qutersat`, `rcl`, `restartsat`, `cryptominisat2011`, `spear-sw`, `spear-hw`, `eagleup`, `sparrow`, `marchrw`, `mphaseSATm`, `satime11`, `tnm`, `mxc09`, `gnoveltyp2`, `sattime`, `sattimep`, `clasp2`, `clasp1`, `picosat`, `mphaseSAT`, `sapperlot` i `sol`. Aplikacija takođe vrši serijalizaciju obučanih modela u cilju bržeg predviđanja instanci u budućnosti.
- Predviđanje vremena izvršavanja SAT rešavača obučanih modela. U slučaju da se vrši obučavanje modela, aplikacija vrši njihovu evaluaciju i prikazivanje rezultata.

Na slici 5.1 dat je dijagram aktivnosti aplikacije koja je korišćena u ovom radu za potrebe eksperimentalne evaluacije opisanih modela u zadatku predviđanja vremena izvršavanja SAT rešavača. U zavisnosti od odabira modela, na početku je potrebno izvršiti odgovarajuća pretprocesiranja ulaznih podataka. Nakon toga sledi obučavanje modela i njegova evaluacija. Konačno, aplikacija generiše rezultate eksperimenata.



SLIKA 5.1: Dijagram aktivnosti aplikacije koja obučava modele za predviđanje vremena izvršavanja SAT rešavača i evaluira te modele.

## 5.2 Opisi podataka

U nastavku dajemo opise nekoliko formata u kojima su zapisani ulazni podaci, odnosno, međupodaci koje naš sistem generiše, kao i opise podataka koji se koriste za obučavanje regresionih modela.

### 5.2.1 DIMACS format

Kao što smo rekli, ulazni podaci za SAT rešavače su iskazne formule zapisane u KNF. Postoji široko prihvaćen format u kojem se ovakve iskazne formule zapisuju i on se naziva DIMACS format. Opis ovog formata je sledeći:

- Datoteka zapisana u ovom formatu započinje opcionim linijama koje predstavljaju komentare i one započinju karakterom `c`.
- Nakon opcionih komentara, neophodno je da datoteka sadrži tačno jednu liniju koja počinje karakterima `p cnf`, nakon kojih se nalazi broj iskaznih promenljivih i broj klauza.
- Svaka naredna linija predstavlja jednu klauzu. Pozitivni literal neke iskazne promenljive je označen pozitivnim celim brojem, a njegov odgovarajući negativni literal je označen negativnim celim brojem. Poslednji broj u liniji mora biti nula, čime se označava kraj klauze.

Jedan primer datoteke zapisane u DIMACS formatu je:

```
c A sample .cnf file.
p cnf 4 3
1 -3 4 0
2 3 -4 0
-4 0
```

Kao što vidimo, ova datoteka kodira iskaznu formulu koja ima četiri iskazne promenljive  $p_1, p_2, p_3, p_4$  i tri klauze. Prva klauza je  $p_1 \vee \neg p_3 \vee p_4$ , druga klauza je  $p_2 \vee p_3 \vee \neg p_4$ , a treća klauza je  $\neg p_4$ . Odatle zaključujemo da je iskazna formula kodirana ovom datotekom  $(p_1 \vee \neg p_3 \vee p_4) \wedge (p_2 \vee p_3 \vee \neg p_4) \wedge \neg p_4$ .

### 5.2.2 Edgelist format

Iskaznu formulu koja se zapisana u DIMACS formatu je potrebno zapisati u formatu koji kodira njenu grafovsku reprezentaciju. Jedan popularan format jeste *list of edges* (eng. *edgelist*). U ovom formatu, svaka linija u datoteci se sastoji od dve vrednosti koje odgovaraju po jednom čvoru u grafu, pri čemu postoji grana u tom grafu u smeru od prvog čvora ka drugom čvoru.

Prisetimo se grafičkog prikaza na slici 3.2 grafa promenljivih i klauza iskazne formule  $(p_1 \vee \neg p_3 \vee p_4) \wedge (p_2 \vee p_3 \vee \neg p_4) \wedge \neg p_4$ . Ako klauzama  $c_1, c_2$  i  $c_3$  redom dodelimo oznake 0, 1 i 2, zatim literalima  $p_1, p_2, p_3$  i  $p_4$  redom dodelimo oznake 3, 4, 5 i 6,

i konačno, literalima  $\neg p_1, \neg p_2, \neg p_3$  i  $\neg p_4$  redom dodelimo oznake 7, 8, 9 i 10, onda možemo ovu iskaznu formulu zapisati u formatu lista grana na sledeći način:

```
0 3
0 6
0 9
1 4
1 5
1 10
2 10
3 0
4 1
5 1
6 0
9 0
10 1
10 2
```

### 5.2.3 DGCNN format

Implementacija metoda duboke grafovske konvolutivne neuronske mreže, koja je preuzeta od autora ovog metoda, zahteva da se svaki graf predstavi u formatu kojeg ćemo nadalje nazivati *DGCNN format*. Svaka datoteka u ovom formatu sadrži zaglavlje kojeg čini broj čvorova u grafu  $n$ , a kojeg prati jedna ili više vrednosti koje predstavljaju oznake grafa (u slučaju klasifikacije) ili neprekidne vrednosti (u slučaju regresije) koje se predviđaju. Zatim sledi  $n$  linija od kojih  $i$ -ta linija opisuje  $i$ -ti čvor u grafu (indeksiranje počinje od 0). Ove linije počinju vrednostima  $t$  i  $m$  koje određuju oznaku čvora i broj suseda tekućeg čvora, redom. Nakon njih, a u istoj liniji, nalazi se tačno  $m$  indeksa suseda tekućeg čvora (indeksiranje takođe počinje od 0).

Iskaznu formulu  $(p_1 \vee \neg p_3 \vee p_4) \wedge (p_2 \vee p_3 \vee \neg p_4) \wedge \neg p_4$  možemo zapisati u DGCNN formatu na sledeći način:

```
11 Y1 Y2 ... Ym
0 3 3 6 9
0 3 4 5 10
0 1 10
1 1 0
1 1 1
1 1 1
1 1 0
-1 0
-1 0
-1 1 0
-1 1 2
```

Napomenimo da u primeru formata iznad koristimo oznaku  $Y1 Y2 \dots Ym$  umesto pravih vrednosti obeležja koja je potrebno predvideti za dati graf. Takođe, za čvor

koji odgovara klauzi koristimo oznaku 0, a za čvorove koji odgovaraju pozitivnom, odnosno, negativnom literalu koristimo oznake 1 i -1, redom.

#### 5.2.4 SATzilla atributi

Metodi  $k$ -najbližih suseda i šume nasumičnih stabala nemaju sposobnost da uče nad grafovskim podacima. Stoga je neophodno konstruisati attribute koji opisuju iskazne formule u KNF. Jedan skup atributa koji se često koristi u ove svrhe je skup *SATzilla atributa*, koji je svoje ime dobio po portfolio SAT rešavaču koji ih koristi. Ceo skup atributa se može podeliti na naredne podskupove:

1. Atributi koji opisuju veličinu ulazne instance.
2. Atributi grafa promenljivih i klauza.
3. Atributi grafa promenljivih.
4. Atributi grafa klauza.
5. Atributi balansa promenljivih i klauza ulazne instance.
6. Atributi bliskosti sa Hornovom formulom.
7. DPLL-atributi, koji se dobijaju pozivom DPLL procedure (bez koraka unazad) do određenih dubina pretrage i izdavanjem određenih statistika na tim dubinama, kao što su: broj jediničnih propagacija, procena veličina prostora pretrage, i druge.
8. LP-atributi, koji se dobijaju rešavanjem problema linearnog programiranja koji kodira datu instancu SAT problema.
9. LSP-atributi, koji se dobijaju višestrukim pozivanjima dva stohastička algoritma lokalne pretrage, GSAT i SAPS. Statistike koje se dobijaju se uprosečavaju kroz date pozive.
10. CL-atributi čine statistike o naučenim klauzama dobijene pozivom SAT rešavača `Zchaff_rand` u trajanju od 2 sekunde.
11. SP-atributi, koji nastaju pozivom SAT rešavača `VARSAT` koji, za svaku promenljivu, procenjuje verovatnoću da promenljiva mora biti tačna ili netačna, ili da nema ograničenja.
12. Meta-atributi o vremenu izračunavanja atributa svakog od prethodnih skupova.

U ovom radu, biće korišćeni atributi iz skupova 1–6, kao i neki atributi iz skupa 12. U dodatku [A](#) navedeni su opisi za svaki atribut koji je korišćen u implementaciji.

### 5.2.5 Node2Vec

Zbog načina na koji grafovske neuronske mreže funkcionišu, potrebno je odabrati attribute za čvorove grafova nad kojima će biti izvršeno učenje. Mi smo odlučili da za tu potrebu odaberemo metod Node2Vec, popularan sistem za pronalaženje vektorskih reprezentacija čvorova u grafu. Pri tome, reprezentacije koje Node2Vec pronalazi su naučene iz strukture samog grafa i stoga nose značajniju semantiku u odnosu na neke elementarne statistike čvorova grafa, poput stepena čvora.

Osnova sistema Node2Vec jeste u simulaciji slučajnih puteva kroz graf. Za neki čvor  $u$  u grafu  $G = (V, E)$ , *slučajni put* dužine  $l$  se generiše na sledeći način. Neka  $c_i$  predstavlja  $i$ -ti čvor u tom putu, pri čemu je  $c_0 = u$ . Tada je uslovna verovatnoća da naredni čvor u putu bude  $x$ , ako je tekući čvor  $v$ , jednaka:

$$P(c_i = x \mid c_{i-1} = v) = \begin{cases} \pi_{vx} & \text{ako je } \{v, x\} \in E \\ 0 & \text{inače} \end{cases}$$

pri čemu je  $\pi_{vx}$  verovatnoća prelaska između čvorova  $v$  i  $x$  u grafu  $G$ .

Node2Vec definiše slučajni put koristeći mehanizam dvokoračnog odabira sledećeg čvora, definisanog parametrima  $p$  i  $q$  na sledeći način. Neka je poslednji prelaz u konstrukciji slučajnog puta izvršen preko grane  $\{t, v\}$ , te se sada konstrukcija slučajnog puta nalazi u čvoru  $v$ . Algoritam sada treba da odluči kojom granom će nastaviti dalje. U ovom trenutku je potrebno izračunati verovatnoću prelaska između čvorova  $v$  i  $x$ ,  $\pi_{vx}$ , preko grana  $\{v, x\}$  koji vode od tekućeg čvora  $v$  do svih njegovih suseda  $x \in N_v$ . Ova verovatnoća se izračunava formulom:

$$\pi_{vx} = \alpha_{pq}(t, x) \cdot A_{vx},$$

gde je

$$\alpha_{pq}(t, x) = \begin{cases} 1/p & \text{ako je } d_{tx} = 0 \\ 1 & \text{ako je } d_{tx} = 1 \\ 1/q, & \text{ako je } d_{tx} = 2 \end{cases}$$

pri čemu je  $A$  matrica susedstva, a  $d_{tx}$  dužina najkraćeg puta od čvora  $t$  do čvora  $x$ .

Parametar  $p$  kontroliše verovatnoću neposrednog ponovnog posećivanja čvora u putu. Velike vrednosti ovog parametra ( $> \max(1, q)$ ) vode ka niskoj verovatnoći ponovnog posećivanja već posećenog čvora u naredna dva koraka. Parametar  $q$  omogućava pretrazi da razlikuje dolazeće i odlazeće čvorove. Za vrednosti  $q > 1$ , slučajni put će birati čvorove bliske čvoru  $t$ . Ovaj pristup vodi ka izdvajanju lokalnih informacija u grafu. Nasuprot tome, vrednosti parametra  $q < 1$  vode ka većoj tendenciji za istraživanje, udaljavanjem od čvora  $t$  i izdvajanjem globalnih informacija iz strukture grafa [27].

### 5.3 Opis sistema

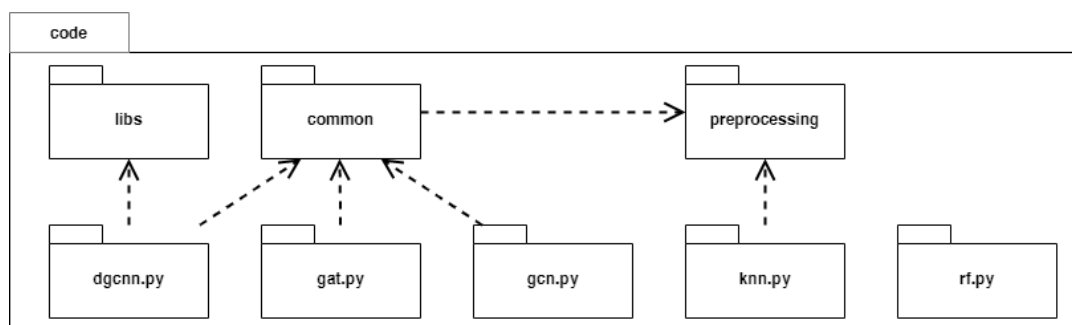
U nastavku sledi opis implementacije projekta koji je javno dostupan na adresi <https://github.com/theikeofficial/Master-Thesis>. Sistem je razvijen prevažno u programskom jeziku Python, sa određenim elementima u programskom jeziku C++. U nastavku podrazumevano da je kod implementiran u programskom jeziku Python, osim ukoliko to nije naznačeno drugačije.

U korenu projekta se nalaze naredni direktorijumi:

- Direktorijum `instances` sadrži podatke neophodne za obučavanje modela, poput podele instanci na skupove za obučavanje, validaciju i evaluaciju, kao i skripte za preuzimanje instanci.
- Direktorijum `code` sadrži izvorni kod aplikacije.
- Direktorijum `third-party` sadrži izvorni kod i izvršne aplikacije razvijene od strane *trećih lica* koji se koriste za potrebe izvršavanja aplikacije.

#### 5.3.1 Paket code

Kao što smo rekli, direktorijum `code` sadrži izvorni kod cele aplikacije. Na slici 5.2 je prikazan dijagram paketa za ovaj paket. Kao što vidimo, on je podeljen u tri pomoćna direktorijuma: `common`, `preprocessing` i `libs` koji predstavljaju pakete sa implementacijama raznih pomoćnih procedura koje opisujemo u nastavku. Pored njih, ovaj direktorijum sadrži i Python pakete koji implementiraju glavne procedure za obučavanje i evaluaciju modela. Paketi `knn.py`, `rf.py`, `gcn.py`, `gat.py` i `dgcnn.py` implementiraju glavne procedure za obučavanje i evaluaciju modela zasnovanih na metodima  $k$ -najbližih suseda, šuma nasumičnih stabala, grafovskih konvolutivnih mreža, grafovskih mreža sa mehanizmom pažnje i dubokih grafovskih konvolutivnih neuronskih mreža, redom. U nastavku dajemo opise pomoćnih paketa.



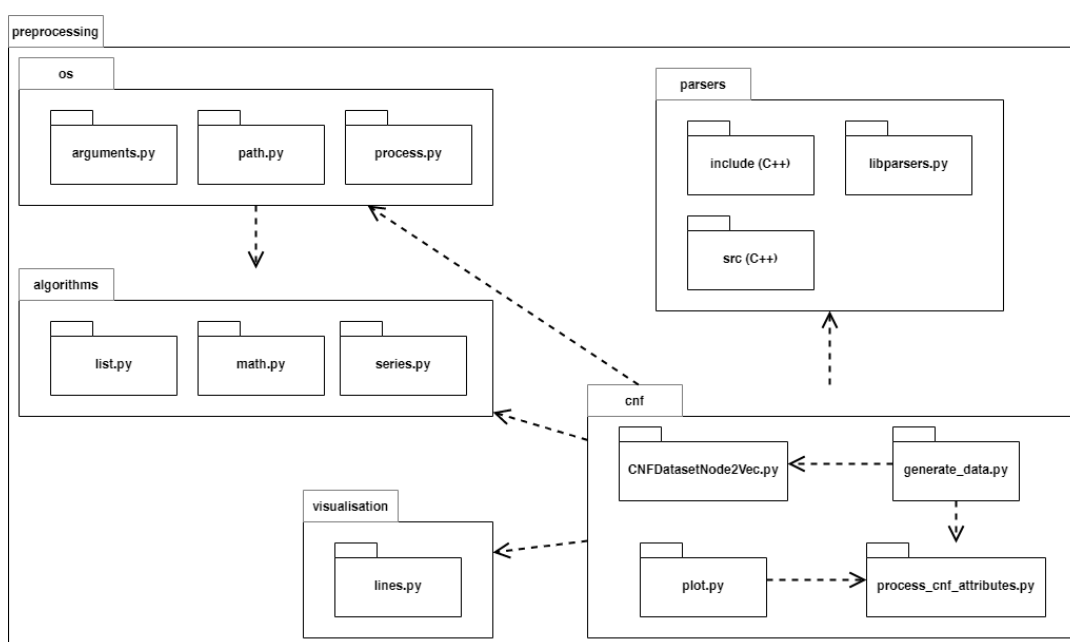
SLIKA 5.2: Dijagram paketa `code`.



### 5.3.2 Paket preprocessing

Direktorijum `preprocessing` sadrži izvorni kod koji implementira procedure za preprocesiranje podataka. Paket `preprocessing` je podeljen u narednih pet paketa, a njihove zavisnosti su prikazane na slici 5.3:

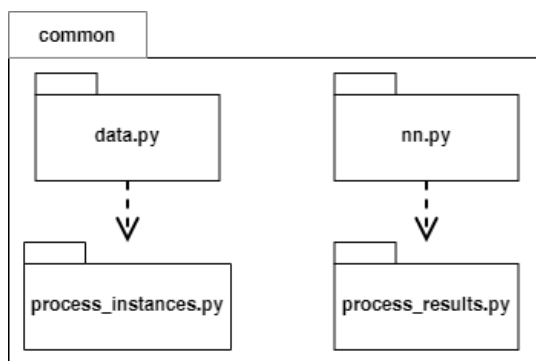
- Paket `preprocessing.algorithms` implementira procedure nad strukturama podataka koje se koriste u aplikaciji, poput Python listi i `Series` iz paketa `pandas`, kao i nekih matematičkih funkcija za preprocesiranje podataka.
- Paket `preprocessing.cnf` implementira procedure za generisanje međupodataka od ulaznih podataka. Ovde se mogu pronaći funkcije za generisanje datoteka u raznim formatima, kao i kod za generisanje atributa. Dodatno, ovaj paket implementira procedure za pronalaženje raznih atributa iskaznih formula u KNF zapisanih u DIMACS formatu kao i procedure za generisanje grafikona na osnovu ovih atributa.
- Paket `preprocessing.os` implementira funkcije za rad sa operativnim sistemom, kao što su: procesiranje argumenata komandne linije, izračunavanje veličina datoteka i pozivanje spoljašnjih aplikacija.
- Paket `preprocessing.parsers` implementira biblioteku koja služi za konverziju datoteka zapisanih u DIMACS formatu u datoteke zapisane u drugim potrebnim formatima. Radi ubrzanja ovih procedura, ove funkcije su implementirane u programskom jeziku C++. Dodatno, implementiran je Python API za lakše pozivanje ovih procedura iz programskog jezika Python.
- Paket `preprocessing.visualisation` implementira procedure za generisanje različitih tipova grafikona na osnovu prosleđenih podataka.



SLIKA 5.3: Dijagram paketa `preprocessing`.

### 5.3.3 Paket common

Paket `common` implementira procedure koje su zajedničke za sve modele (ili za podskup tih modela). U okviru ovog paketa se mogu pronaći procedure za učitavanje podataka i njihovu pripremu (`data.py` i `process_instances.py`), pomoćne procedure za obučavanje i evaluaciju modela zasnovanih na grafovskim neuronskim mrežama (`nn.py`) i procesiranje rezultata (`process_results.py`). Dijagram paketa `common` je dat na slici 5.4.



SLIKA 5.4: Dijagram paketa `common`.

### 5.3.4 Paket libs

U slučaju kada je implementacija nekog modela preuzeta od projekta otvorenog koda, kao što je to slučaj sa implementacijom duboke grafovske konvolutivne neuronske mreže, pomoćne datoteke koje su deo takvih projekata je potrebno smestiti u direktorijum `libs`.

### 5.3.5 Paket `main.py` i pomoćni skriptovi

Kako bi ilustrovali korišćenje ovih paketa, konstruisali smo paket `main.py` koji, na osnovu argumenata komandne linije, implementira sistem opisan dijagramom 5.1. Radi lakšeg pokretanja sistema, na raspolaganju su naredni skriptovi:

- Skript `prepare_data.sh` preuzima skup podataka nad kojim je vršeno obučavanje i evaluacija sistema.
- Skript `train.sh` služi za pronalaženje i čuvanje najboljih modela, za svaki od ponuđenih metoda.
- Skript `test.sh` služi za evaluaciju pronađenih najboljih modela.

## 5.4 Implementacione napomene

Paket `main.py` zavisi od velikog broja argumenata komandne linije kojima se zadaju: tip izvršavanja aplikacije (obučavanje, evaluacija, predviđanje i sl.), parametri obučavanja modela, parametri evaluacije, lokacije instanci i dr.

Za implementaciju metoda  $k$ -najbližih suseda i šume nasumičnih stabala korišćene su klase iz Python paketa `scikit-learn` [28]: za prvi metod je korišćena klasa `sklearn.neighbors.KNeighborsRegressor`, dok je za drugi metod korišćena klasa `sklearn.ensemble.RandomForestRegressor`. U ovim metodima je za svaki SAT rešavač iz portfolija konstruisan po jedan regresioni metod koji predviđa vreme izvršavanja upravo tog SAT rešavača. Ovakav mehanizam višeciljne regresije je ostvaren klasom `sklearn.multioutput.MultiOutputRegressor` iz istog Python paketa.

Implementacija grafovske konvolutivne mreže i grafovske mreže sa mehanizmom pažnje je izvršena pomoću biblioteke *Deep Graph Library* [29]. Implementacija duboke grafovske konvolutivne neuronske mreže je preuzeta od autora ovog modela [9] i prilagođena tako da se uklopi u opisanu arhitekturu sistema.

Pronalaženje SATzilla atributa iskazne formule je izvršeno pomoću aplikacije preuzete od autora SATzilla portfolio SAT rešavača [13]. Pronalaženje Node2Vec atributa čvorova u grafu je izvršeno pomoću biblioteke *GraphVite* [30].

## Glava 6

# Eksperimentalna evaluacija

Ova glava opisuje neke karakteristike skupa podataka kojeg ćemo koristiti u eksperimentima. Zatim opisujemo proceduru za obučavanje, validaciju i evaluaciju svakog od opisanih regresionih modela i predstavljamo rezultate eksperimenata. Cilj eksperimenta jeste utvrđivanje kvaliteta predviđanja vremena izvršavanja SAT rešavača pomoću metoda zasnovanih na grafovskim neuronskim mrežama u poređenju sa metodima koji su demonstrirali značajan uspeh do sada, kao što su  $k$ -najbližih suseda i šume nasumičnih stabala. Kako bi komparativna analiza ovih metoda bila što merodavnija, eksperimente ćemo izvršiti nad istim skupovima podataka među svim metodima, a korišćemo i iste tehnike izbora i ocene modela. S obzirom na veliki uspeh grafovskih neuronskih mreža u drugim oblastima, za očekivati je da one ostvaruju bolje performanse od tradicionalnih metoda. Međutim, kao što ćemo videti, za dati skup podataka koji smo koristili, one su ostvarile ne samo lošije rezultate, već su ti rezultati čak i negativni. Na kraju glave analiziramo dobijene rezultate i dajemo objašnjenja za negativne rezultate, kao i pravce daljeg istraživanja.

### 6.1 Skup podataka i mere evaluacije

Skup podataka koji je korišćen za obučavanje i evaluaciju metoda se sastoji od instanci korišćenih u takmičenju *SAT Competition* iz 2012. godine [12]. Metapodaci o ovom skupu podataka su preuzeti iz javno dostupne *biblioteke selekcije algoritama*, *ASlib* [31]. Broj SAT rešavača koji je korišćen u ovom skupu je 31. Ovaj skup podataka je odabran kao jedan od najvećih korpusa formula za koje postoje podaci o vremenima izvršavanja velikog broja SAT rešavača. Ceo skup podataka sadrži 3296 instance, koje se mogu podeliti u tri kategorije:

- Kategorija *SAT12-HAND* sadrži ručno konstruisane instance i broji 767 instanci.
- Kategorija *SAT12-INDU* sadrži instance iz industrije i broji 1167 instanci.
- Kategorija *SAT12-RAND* sadrži nasumično generisane instance i broji 1362 instance.

Napomenimo da se u ovim kategorijama nalaze i potkategorije u okviru kojih su instance raspoređene u odnosu na oblasti problema koje te instance kodiraju, kao što su: bioinformatika, raspoređivanje resursa, verifikacija, sudoku, kriptografija i dr.

Eksperimenti su izvršeni na računaru sa šestojezgarnim AMD Ryzen 5 1600 3.20GHz procesorom, radnom memorijom od 8GB i grafičkom kartom NVIDIA GeForce GTX 1050 Ti 3GB. Zbog veličine memorije grafičke karte na kojoj su eksperimenti vezani za grafovske neuronske mreže sprovedeni, bilo je potrebno smanjiti skup podataka. Izvršili smo naredne operacije nad celim skupom podataka, redom:

- Eliminirali smo sve instance za koje nijedan rešavač iz portfolija nije uspeo da ih reši. Smatramo da bi ovakve instance smanjile mogućnost učenja modela.
- Eliminirali smo sve instance iz skupa *SAT12-RAND*. Fokusirali smo se samo na instance koje su ručno konstruisane, kao i na probleme iz industrije, pošto su ovakve instance značajnije u praksi od nasumičnih.
- Eliminirali smo sve instance koje su imale više od 50000 promenljivih ili 600000 klauza. Razlog za ovo je ograničenost memorije računarskih resursa.
- Eliminirali smo sve potkategorije koje nisu imale dovoljan broj instanci za raspoređivanje po skupovima za obučavanje, validaciju i evaluaciju.

Nakon ovih operacija, skup podataka se sastoji od 720 instanci. Ove instance smo, dalje, podelili na skupove za obučavanje, validaciju i evaluaciju modela prema tabeli 6.1. Prvobitna podela podataka je podrazumevala naredni odnos: 65% na skup za obučavanje, 15% na skup za validaciju i 20% na skup za evaluaciju. Nakon toga, primetili smo da instance iz nekih potkategorija nisu bile raspoređene u svim skupovima, pa ih je bilo potrebno ručno rasporediti tako da svi skupovi sadrže barem po jednu instancu iz svih potkategorija.

<i>Skup podataka</i>	<i>Broj instanci</i>
Obučavanje	428
Validacija	133
Evaluacija	159

TABELA 6.1: Raspodela instanci na skupove za obučavanje, validaciju i evaluaciju.

Pre obučavanja modela, izvršeno je skaliranje ciljnih promenljivih logaritamskom funkcijom  $\log_{10}$ , radi stabilnosti izvršavanja algoritama. Pre primene logaritamske funkcije, sve vrednosti ciljne funkcije koje su bile manje od  $10^{-3}$  su bile postavljene na ovu vrednost.

Za izbor i ocenu modela su korišćene sledeće tri mere:

- Srednjekvadratna greška:

$$MSE(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad (6.1)$$

- Koeficijent determinacije:

$$R^2(x, y) = 1 - \frac{MSE(x, y)}{\sum_{i=1}^n (\bar{y} - y_i)^2}, \text{ gde je } \bar{y} = \frac{1}{n} \sum_{j=1}^n y_j \quad (6.2)$$

- Koren srednjekvadratne greške:

$$RMSE(x, y) = \sqrt{MSE(x, y)} \quad (6.3)$$

## 6.2 Evaluacija modela za poređenje

U slučaju obučavanja modela  $k$ -najbližih suseda i šuma nasumičnih stabala, modeli različitih parametarskih konfiguracija su obučavani nad SATzilla atributima instanci iz skupa za obučavanje. SATzilla atributi su pre obučavanja standardizovani na interval  $[0, 1]$ . Zatim, svaki od obučanih modela je testiran nad skupom za validaciju. Na osnovu tih rezultata, odabran je model čija je prosečna  $R^2$  mera za sve SAT rešavače najmanja i taj model je ponovo obučavan nad instancama iz skupa za obučavanje i validaciju. Ovako dobijen model je testiran nad skupom za evaluaciju kako bi se dobila konačna ocena modela.

Prilikom obučavanja modela  $k$ -najbližih suseda, parametri koji su korišćeni su dati u tabeli 6.2. Parametar `n_neighbors` se odnosi na broj suseda  $k$ . Parametar `weights` se odnosi na funkciju težine koja se koristi prilikom predviđanja. Vrednost `uniform` označava da je svakom susedu dodeljena ista težina, dok vrednost `distance` postavlja vrednost težine na inverz rastojanja do tog suseda. Drugim rečima, susedi koji su bliži instanci koja se predviđa će imati veću težinu nego susedi koji su udaljeniji od te instance. Konačno, parametar `metric` određuje funkciju izračunavanja rastojanja do suseda, što može biti `euclidean` (4.1 za  $p = 2$ ), `manhattan` (4.1 za  $p = 1$ ), `lorentzian` (4.2) ili `angular` (4.3).

Parametar metoda	Dopuštene vrednosti
<code>n_neighbors</code>	$\{1, 2, \dots, 10\}$
<code>weights</code>	<code>uniform, distance</code>
<code>metric</code>	<code>euclidean, manhattan, lorentzian, angular</code>

TABELA 6.2: Mreža parametarskih konfiguracija za model  $k$ -najbližih suseda.

Parametar metoda	Dopuštene vrednosti
<code>n_estimators</code>	$\{10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$
<code>min_samples_split</code>	$\{0.05, 0.1, 0.2, 0.3, 2, 10, 20, 50\}$

TABELA 6.3: Mreža parametarskih konfiguracija za model šuma nasumičnih stabala.

Prilikom obučavanja modela šuma nasumičnih stabala, parametri koji su korišćeni su dati u tabeli 6.3. Parametar `n_estimators` se odnosi na broj stabala odlučivanja koji se obučavaju nad različitim podskupovima skupa podataka. Parametar `min_samples_split` predstavlja najmanji broj (apsolutnu vrednost u slučaju celog broja ili procenat u slučaju broja u pokretnom zarezu) instanci koje čvor mora da sadrži kako bi se izvršilo grananje. Kao mera grananja je korišćena srednjekvadratna greška 6.1.

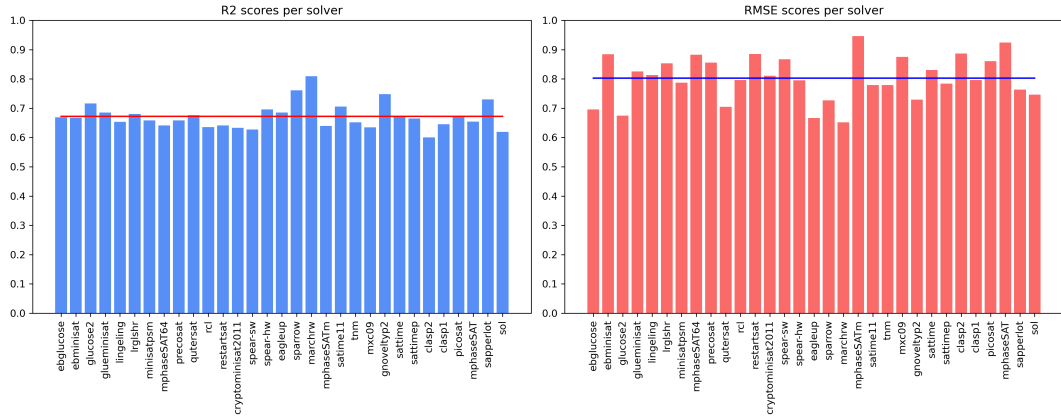
Prilikom odabira modela za oba ova metoda, nakon što se odabere jedna konfiguracija, konstruiše je po jedan regresioni model za svaki SAT rešavač iz portfolija sa datom konfiguracijom koji predviđa vreme izvršavanja tog SAT rešavača. Tako, za broj konfiguracija  $N$  i broj SAT rešavača iz portfolija  $P$ , u fazi obučavanja konstruišemo ukupno  $N \times P$  regresionih modela. U oba slučaja, ovaj broj je  $80 \times 31 = 2480$ .

Na slikama 6.1 i 6.2 su dati grafički prikazi ocena najboljih modela metoda  $k$ -najbližih suseda i šume nasumičnih stabala, redom. Za ocenu su korišćene mere  $R^2$  (levo) i  $RMSE$  (desno). Ocene su prikazane stubičastim dijagramima za svaki SAT rešavač iz portfolija, a horizontalnom linijom je označena prosečna vrednost. U tabeli 6.4 su date prosečne ocene najboljih modela, a u dodatku B se mogu pronaći detaljni rezultati eksperimenata. Ocene koje su najbolje su označene podebljanim fontom. Kao što vidimo, metod  $k$ -najbližih suseda ima nešto slabiju prediktivnu moć u odnosu na šume nasumičnih stabala, međutim, dobijene ocene su očekivane na osnovu prethodnih istraživanja na ovu temu [32]. Parametri najboljeg modela  $k$ -najbližih suseda su: `n_neighbors=8`, `weights=distance` i `metric=manhattan`. Parametri najboljeg modela šume nasumičnih stabala su: `n_estimators=1000` i `min_samples_split=2`.

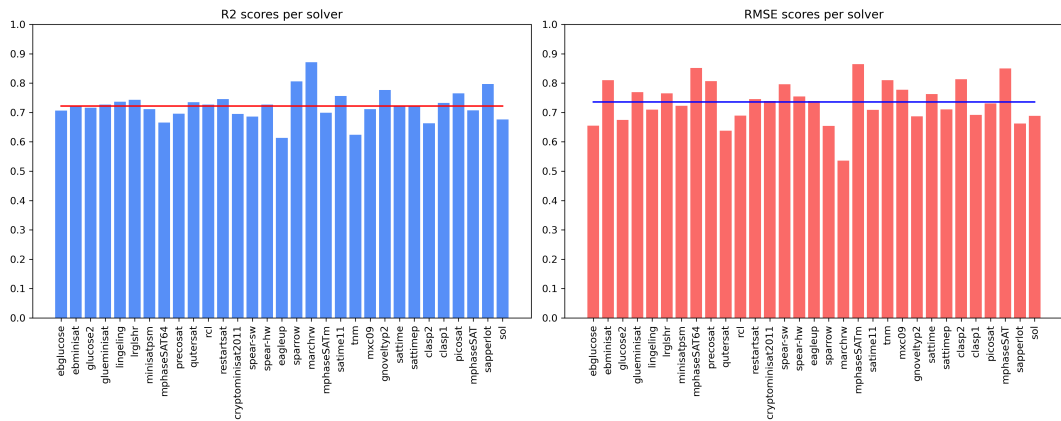
Neki od portfolio sistema koji su izgrađeni nad ovim metodima prijavljuju solidne performanse u smislu odabira SAT rešavača iz portfolija koji najbrže rešava SAT problem za datu instancu. Na primer, u slučaju  $k$ -najbližih suseda, portfolio sistem iz [16] je ocenjen tačnošću od 61.7%, dok je portfolio sistem zasnovan na šumi nasumičnih stabala iz [15], pod drugačijim okolnostima, ocenjen tačnošću od 88.5% [33].

$R^2$ (KNN)	$R^2$ (RF)
0.6721	<b>0.7221</b>
$RMSE$ (KNN)	$RMSE$ (RF)
0.8025	<b>0.7359</b>

TABELA 6.4: Prosečne ocene  $R^2$  (gore) i  $RMSE$  (dole) modela  $k$ -najbližih suseda i šume nasumičnih stabala. Najbolji rezultati su prikazani podebljanim slovima.



SLIKA 6.1: Ocene najboljeg modela  $k$ -najbližih suseda po SAT rešavačima iz portfolija.



SLIKA 6.2: Ocene najboljeg modela šume nasumičnih stabala po SAT rešavačima iz portfolija.

### 6.3 Evaluacija metoda zasnovanih na grafovskim neuronskim mrežama

Što se tiče eksperimenata korišćenjem modela zasnovanih na grafovskim neuronskim mrežama, zbog ograničenih resursa, konstruisali smo po jedan model za svaki od tri metoda koje smo opisali u poglavlju 4. Parametri obučavanja grafovskih neuronskih mreža su predstavljeni u tabelama 6.5, 6.6 i 6.7. Za sva tri modela je za funkciju cilja odabrana srednjekvadratna greška 6.1. Kao optimizator je odabran algoritam ADAM sa parametrom učenja 0.001.

Opisani modeli su obučavani na skupu za obučavanje najviše onoliko epoha koliko je navedeno u odgovarajućim tabelama. Prilikom svake epohe, praćena je vrednost funkcije cilja na skupu za validaciju. U slučaju kada vrednost funkcije cilja u poslednjoj episi ne prevalizazi napredak od 5% u odnosu na prosečnu vrednost ciljne funkcije, treniranje se obustavlja. Konačno, epoha u kojoj je model imao najbolju vrednost funkcije cilja na validacionom skupu je proglašena za najbolju i model je



finalno treniran na skupovima za obučavanje i validaciju zajedno do najbolje epohe. Dobijeni model je testiran na skupu za evaluaciju.

<i>Parametar metoda</i>	<i>Vrednost</i>
Atributi čvorova	Node2Vec dimenzije 64
Skriveni slojevi	64 – 64 – 64
Aktivaciona funkcija na konvolutivnim slojevima	LeakyReLU, $\alpha = 0.2$ (4.14)
Globalna agregatna funkcija	Uprosečavanje (4.18)
Slojevi neuronske mreže na izlazu	31
Broj epoha	50

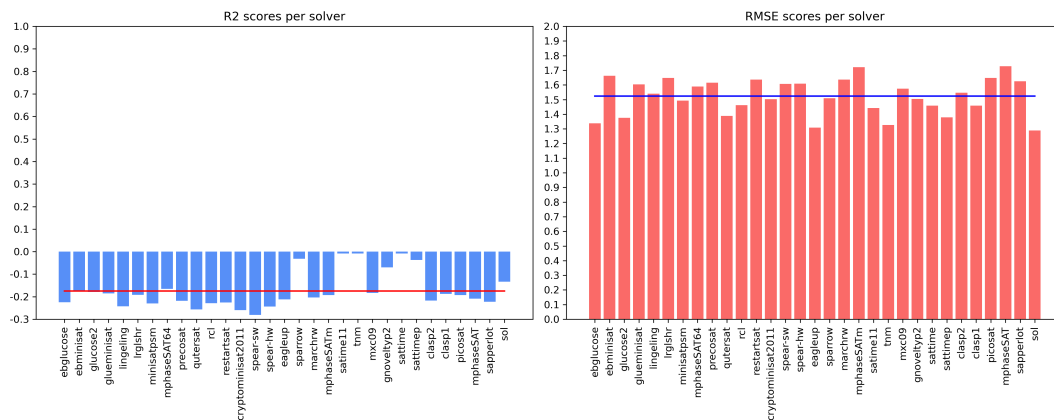
TABELA 6.5: Parametri obučavanja modela grafovske konvolutivne mreže.

<i>Parametar metoda</i>	<i>Vrednost</i>
Atributi čvorova	Node2Vec dimenzije 64
Skriveni slojevi sa meh. pažnje	10 – 10
Višestrukost pažnje	4 – 4
Aktivaciona funkcija na konvolutivnim slojevima	LeakyReLU, $\alpha = 0.2$ (4.14)
Globalna agregatna funkcija	Uprosečavanje (4.18)
Slojevi neuronske mreže na izlazu	31
Broj epoha	50

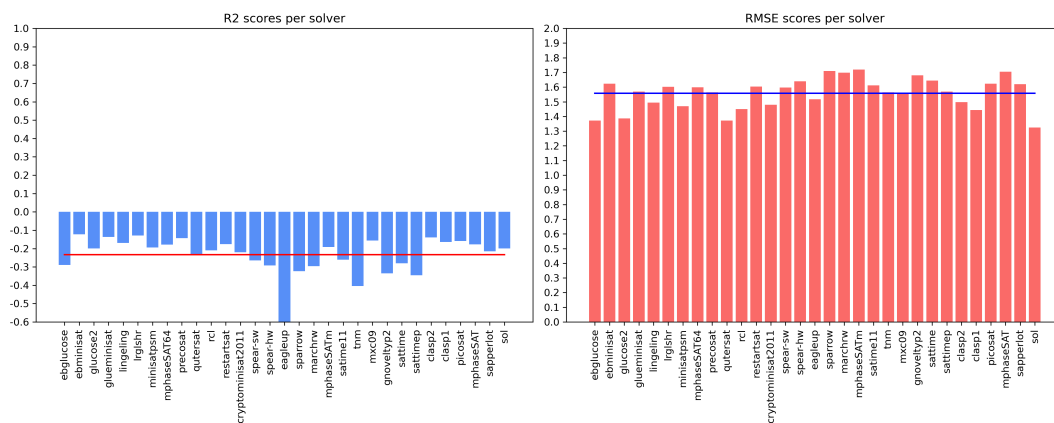
TABELA 6.6: Parametri obučavanja modela grafovske mreže sa mehanizmom pažnje.

<i>Parametar metoda</i>	<i>Vrednost</i>
Atributi čvorova	Node2Vec dimenzije 64
Skriveni konvolutivni slojevi	64 – 64
Aktivaciona funkcija na konvolutivnim slojevima	LeakyReLU, $\alpha = 0.2$ (4.14)
Globalna agregatna funkcija	Sloj sortiranja; $k = 55\%$
Slojevi konvolutivne neuronske mreže	16 – 32
Filteri slojeva konvolutivne neuronske mreže	$\sum_{i=1}^l h^{(i)} - 5$
Slojevi neuronske mreže na izlazu	31
Broj epoha	50

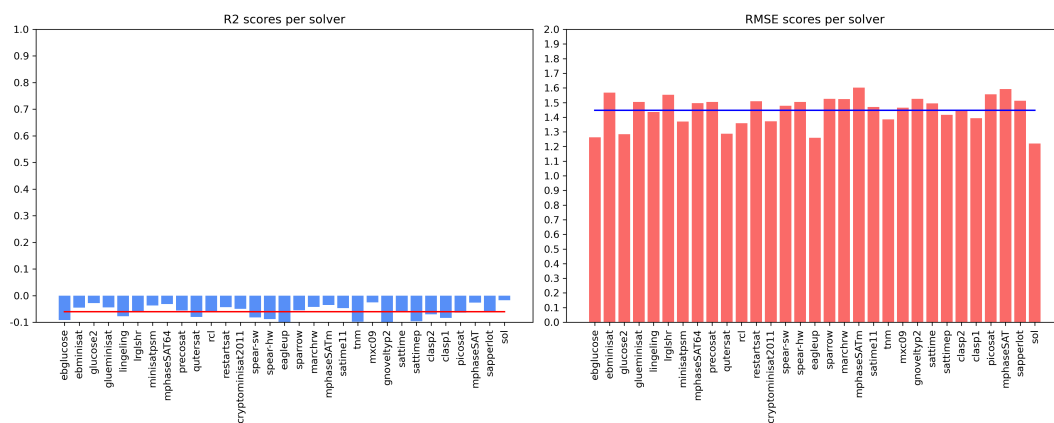
TABELA 6.7: Parametri obučavanja modela duboke grafovske konvolutivne neuronske mreže.



SLIKA 6.3: Ocene najboljeg modela grafovske konvolutivne mreže po SAT rešavačima iz portfolija.



SLIKA 6.4: Ocene najboljeg modela grafovske mreže sa mehanizmom pažnje po SAT rešavačima iz portfolija.



SLIKA 6.5: Ocene najboljeg modela duboke grafovske konvolutivne neuronske mreže po SAT rešavačima iz portfolija.

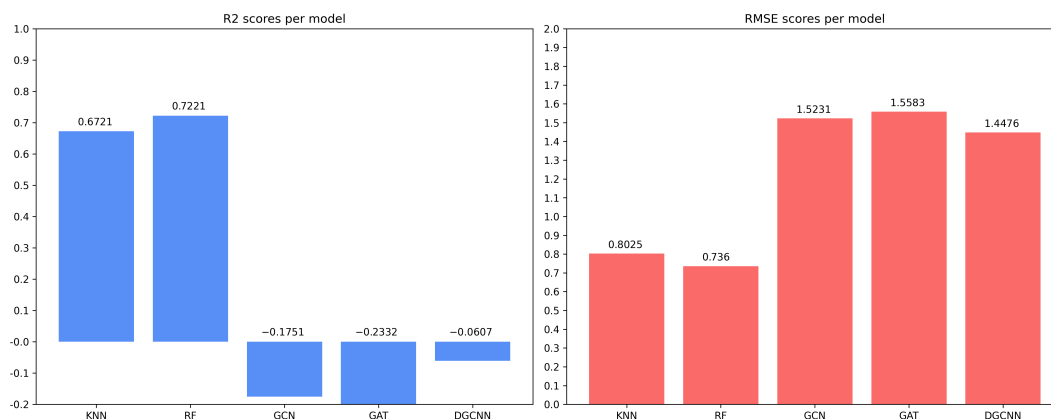
$R^2$ (GCN)	$R^2$ (GAT)	$R^2$ (DGCNN)
-0.1751	-0.2332	<b>-0.0607</b>

RMSE (GCN)	RMSE (GAT)	RMSE (DGCNN)
1.5231	1.5583	<b>1.4476</b>

TABELA 6.8: Prosečne ocene  $R^2$  (gore) i  $RMSE$  (dole) modela zasnovanih na grafovskim neuronskim mrežama. Najbolji rezultati su prikazani podebljanim slovima.

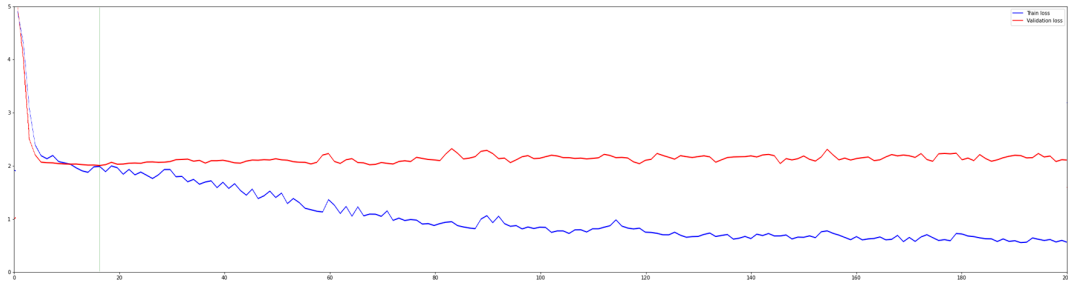
Na slikama 6.3, 6.4 i 6.5 dat je grafički prikaz prosečnih ocena  $R^2$  i  $RMSE$  po SAT rešavačima iz portfolija. Kao što vidimo sa ovih slika, kao i na osnovu informacija iz tabele 6.8, ovi metodi nisu uspeli da nauče korisne informacije dovoljne za adekvatno predviđanje vremena izvršavanja SAT rešavača i njihove performanse su značajno lošije u odnosu na tradicionalnije metode, kao što je vidljivo na slici 6.6. U dodatku B se mogu pronaći detaljni rezultati eksperimenata.



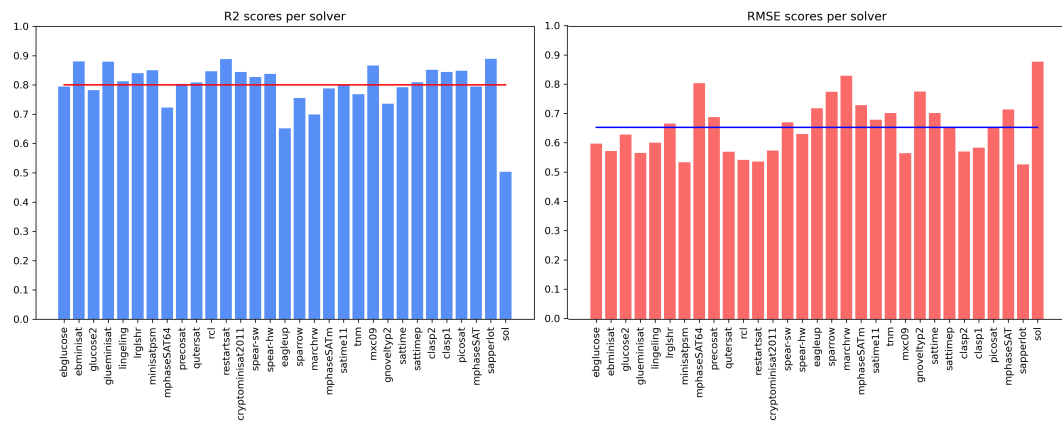
SLIKA 6.6: Uporedni prikaz prosečnih ocena  $R^2$  (levo) i  $RMSE$  (desno) za sve modele koji su obučavani u eksperimentima.

S obzirom na dobijene rezultate, zapitali smo se da li su ovi metodi uopšte u stanju da nauče korisne informacije iz strukture grafovskih reprezentacija SAT instanci ili ne. Zbog toga smo dodatno obučavali još jedan model grafovske konvolutivne mreže na 200 epoha bez prekida.

Na slici 6.7 su dati grafici funkcije cilja prilikom ovog procesa obučavanja. Kao što vidimo, funkcija cilja na skupu za obučavanje (grafik plave boje) postepeno opada i dostiže najmanju vrednost od približno 0.7. Međutim, primećujemo i da se funkcija cilja na skupu za validaciju blago povećava u odnosu na najmanju vrednost dostignutu u 17. eposi. Na osnovu ovoga možemo zaključiti da se modeli prilagođavaju podacima, što je i potpomognuto prikazu ocena kvaliteta na skupu za obučavanje (slika 6.8).



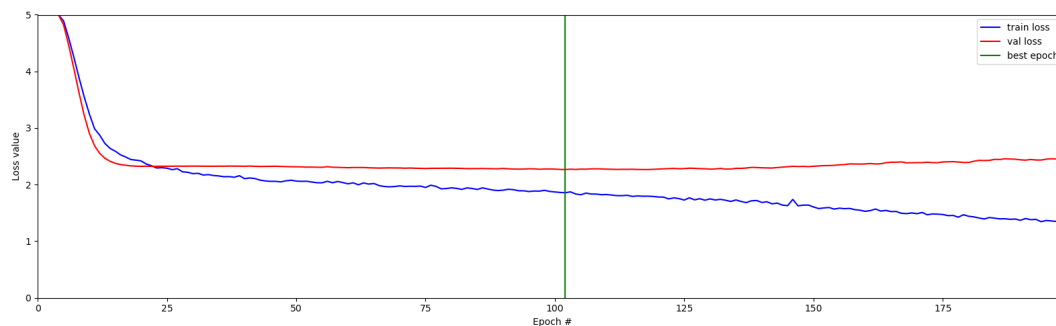
SLIKA 6.7: Grafik funkcija cilja na skupu za obučavanje (plavo) i skupu za validaciju (crveno) prilikom obučavanja modela grafovske konvolutivne mreže na 200 epoha.



SLIKA 6.8: Ocene preprilagođenog modela grafovske konvolutivne mreže na skupu za obučavanje po SAT rešavačima iz portfolija.

S obzirom na primećeno preprilagođavanje podacima, zapitali smo se da li bi manja arhitektura mreže bila uspešnija. Zato sto obučili još jedan model grafovske konvolutivne mreže sa parametrima kao u tabeli 6.5, sa razlikom u konvolutivnim slojevima; ovaj model je koristio dva konvolutivna sloja sa dimenzijama atributa skrivenih slojeva 10 i 20. I ovaj model je obučen na 200 epoha bez prekida. Kao što vidimo na slici 6.9, ovaj model je imao još lošije performanse prilikom obučavanja, što se i ogleda kroz evaluaciju na test skupu: prosečna ocena za  $R^2$  meru je  $-0.3823$ , dok je prosečna ocena za  $RMSE$  meru 1.6510.

Kao što je poznato, neuronske mreže zahtevaju veliku količinu podataka kako bi uspele da nauče korisne informacije iz podataka. Kao što smo na početku poglavlja istakli, zbog računarskih resursa bilo je neophodno da smanjimo broj instanci na skupu za obučavanje, što je svakako uticalo na prediktivne sposobnosti modela. Zbog toga, postoji očekivanje da bi korišćenje većeg i bogatijeg skup podataka smanjilo preprilagođavanje i povećalo uspeh predviđanja.



SLIKA 6.9: Grafik funkcija cilja na skupu za obučavanje (plavo) i skupu za validaciju (crveno) prilikom obučavanja modela grafovske neuronske mreže sa manjom arhitekturom na 200 epoha.

Postoje mnogi razlozi zbog kojih su dobijeni predstavljeni rezultati. Svaki od datih razloga predstavlja moguće pravac daljeg istraživanja na ovu temu. Mi navodimo neke od njih u nastavku:

- Skup podataka nije dovoljne veličine ili nije dovoljno reprezentativan za potrebe predviđanja vremena izvršavanja SAT rešavača. U tu svrhu, moguće je da bi metodi pokazali bolje performanse obogaćivanjem skupa podataka.
- Zbog ograničenih resursa, obučavani modeli imaju manji broj parametara u odnosu na korišćene arhitekture modela iz radova u kojima su oni dostigli značajne uspehe. Ovakvi modeli uče reprezentacije ulaznih podataka koji vode ka preprilagođavanju i slaboj generalizaciji. Moguće je da bi obučavanje modela sa značajno većim brojem parametara, uz inteligentan odabir tehnika regularizacije i obogaćivanja skupa za obučavanje, doprinelo povećanju prediktivne moći modela.
- Kako su grafovske neuronske mreže poslednjih nekoliko godina veoma goruća tema u zajednici mašinskog učenja, naučnici širom sveta su već predložili veliki broj različitih arhitektura grafovskih neuronskih mreža i njihovih pojedinačnih elemenata (arhitektura skrivenih slojeva, funkcije globalne agregacije, itd.). Odabir odgovarajuće arhitekture takođe može doprineti poboljšanju kvaliteta metoda, pa je potrebno ispitati i druge arhitekture zasnovane na grafovskim neuronskim mrežama.

## Glava 7

# Zaključak

U ovom radu smo opisali tri metoda koja su zasnovana na grafovskim neuronskim mrežama i izvršili smo eksperimentalnu evaluaciju tih modela u kontekstu predviđanja vremena izvršavanja različitih SAT rešavača. Iako smo predstavili teorijske osnove zbog kojih smo očekivali da opisani metodi postizu dobre rezultate, kao što smo pokazali u glavi 6, ovi metodi ipak nisu uspeli da opravdaju naša očekivanja. Kao glavni razlog za dobijene rezultate navodimo mali korpus podataka za obučavanje ovih metoda usled ograničenih računarskih resursa na kojima su modeli obučavani, kao i odabir arhitektura sa manjim brojem parametara.

Za razliku od njih, pokazali smo da tradicionalni metodi  $k$ -najbližih suseda i šuma nasumičnih stabala imaju vrlo dobre prediktivne moći, čak i na manjem uzorku podataka. Zbog toga, ukoliko je za praktične potrebe neophodno konstruisati sistem koji predviđa vreme izvršavanja SAT rešavača na nekom manjem skupu SAT instanci i njihovih poznatih vremena, ovi metodi će biti prirodan prvi izbor.

Pored eksperimentalnih rezultata, još jedan doprinos ovog rada jeste softver otvorenog koda koji se može koristiti za predviđanje vremena izvršavanja SAT rešavača opisanim metodima. Već smo prikazali da arhitektura implementiranog softvera dozvoljava nadograđivanje aplikacije tako da uvrsti i druge metode, tako da se ona može koristiti i za eksperimentalnu analizu drugih metoda jednostavnim proširivanjem.

Napomenimo da, zbog dobijenih negativnih vrednosti mera  $R^2$ , nije bilo potrebe implementirati i evaluirati portfolio SAT rešavač zasnovan na grafovskim neuronskim mrežama. Ipak, ukoliko bi se prevazišla ograničenja o kojima smo diskutovali i tom prilikom dobili rezultati slični modelima  $k$ -najbližih suseda i šume nasumičnih stabala, onda bi u tom slučaju bilo značajno izvršiti i uporednu evaluaciju portfolio SAT rešavača zasnovanih na ovim metodima. Za te potrebe je moguće vrlo jednostavno proširiti softver za predviđanje vremena SAT rešavača razvijen u ovom radu.

# Literatura

- [1] A. Biere, M. Heule, H. Van Maaren i T. Walsh. *Handbook of Satisfiability*. Februar 2009, br. strana 980. ISBN: 978-1-60750-376-7.
- [2] Z. Stanić. *Diskretne strukture 2: osnovi kombinatorike, teorije brojeva i teorije grafova*. Matematički fakultet, 2018, br. strana 207. ISBN: 978-86-7589-126-0.
- [3] N. Ye. *The Handbook of Data Mining*. Human factors and ergonomics. CRC Press, 2003. ISBN: 9781410607515. URL: <https://books.google.rs/books?id=3T9ZDwAAQBAJ>.
- [4] M. Kubat. *An Introduction to Machine Learning*. Springer International Publishing, 2015. ISBN: 9783319200101. URL: <https://books.google.rs/books?id=2EAwCgAAQBAJ>.
- [5] L. Breiman. *Classification and Regression Trees*. New York: Routledge, 1984. ISBN: 9781315139470. DOI: [10.1201/9781315139470](https://doi.org/10.1201/9781315139470).
- [6] L. Breiman. „Random Forests”. Izvor: *Machine Learning* 45.1 (2001), strane 5–32. DOI: [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324).
- [7] T. N Kipf i M. Welling. „Semi-Supervised Classification with Graph Convolutional Networks”. Izvor: *arXiv preprint arXiv:1609.02907* (2016).
- [8] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò i Y. Bengio. „Graph Attention Networks”. Izvor: *International Conference on Learning Representations* (2018). accepted as poster. URL: <https://openreview.net/forum?id=rJXMpikCZ>.
- [9] Muhan Zhang, Zhicheng Cui, Marion Neumann i Yixin Chen. „An End-to-End Deep Learning Architecture for Graph Classification”. Izvor: *AAAI*. 2018, strane 4438–4445.
- [10] M. Živković. *Algoritmi*. Matematički fakultet, 2000, br. strana 405. URL: <http://poincare.matf.bg.ac.rs/~ezivkovm/nastava/algoritmi.pdf>.
- [11] M. Hořeňovský. *Modern SAT solvers: fast, neat and underused (part 3 of N)*. 2020. URL: <https://codingnest.com/modern-sat-solvers-fast-neat-and-underused-part-3-of-n/#fn7>.
- [12] *SAT Competitions*. 2020. URL: <http://www.satcompetition.org/>.
- [13] *SATzilla: Portfolio-based algorithm selection for SAT*. 2020. URL: <https://www.dgl.ai/>.

- [14] E. Nudelman, A. Devkar, Y. Shoham, K. Leyton-Brown i H. Hoos. *SATzilla: An Algorithm Portfolio for SAT*. Maj 2004.
- [15] L. Xu, F. Hutter, J. Shen, Holger Hoos i Kevin Leyton-Brown. „SATzilla2012: Improved algorithm selection based on cost-sensitive classification models”. Izvor: *Proceedings of SAT Challenge 2012: Solver and Benchmark Descriptions* (januar 2012), strane 55–58.
- [16] M. Nikolić, F. Marić i P. Janičić. „Simple Algorithm Portfolio for SAT”. Izvor: *Artificial Intelligence Review* 40 (jul 2011). DOI: [10.1007/s10462-011-9290-2](https://doi.org/10.1007/s10462-011-9290-2).
- [17] A. Buluç, J. R. Gilbert i V. B. Shah. „Implementing Sparse Matrices for Graph Algorithms”. Izvor: *Graph Algorithms in the Language of Linear Algebra*. 2011.
- [18] L. Fahrmeir, T. Kneib, S. Lang i B. Marx. *Regression: Models, Methods and Applications*. Januar 2013. DOI: [10.1007/978-3-642-34333-9](https://doi.org/10.1007/978-3-642-34333-9).
- [19] N. Matloff. *Statistical regression and classification: From linear models to machine learning*. Januar 2017, strane 1–493. DOI: [10.1201/9781315119588](https://doi.org/10.1201/9781315119588).
- [20] D. Selsam, M. Lamm, B. Bünz, P. Liang, L. de Moura i D. L. Dill. „Learning a SAT Solver from Single-Bit Supervision”. Izvor: *ArXiv* abs/1802.03685 (2019).
- [21] V. Kurin, S. Godil, S. Whiteson i B. Catanzaro. *Improving SAT Solver Heuristics with Graph Networks and Reinforcement Learning*. 2019. arXiv: [1909.11830](https://arxiv.org/abs/1909.11830) [cs.LG].
- [22] X. Glorot, A. Anand, E. Aygün, S. Mourad, P. Kohli i D. Precup. *Learning representations of Logical Formulae using Graph Neural Networks*. 2019.
- [23] H. Abu Alfeilat, A. Hassanat, O. Lasassmeh, A. Tarawneh, M. Alhasanat, H. Eyal-Salman i S. Prasath. „Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review”. Izvor: *Big Data* 7 (avgust 2019). DOI: [10.1089/big.2018.0175](https://doi.org/10.1089/big.2018.0175).
- [24] M. Kantardžić. *Data Mining: Concepts, Models, Methods, and Algorithms, 3rd Edition*. Springer International Publishing, 2019. ISBN: 978-1-119-51607-1.
- [25] S. Shalev-Schwartz i S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. ISBN: 978-1-107-05713-5.
- [26] Ian Goodfellow, Yoshua Bengio i Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [27] A. Grover i J. Leskovec. „node2vec: Scalable Feature Learning for Networks”. Izvor: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (2016).
- [28] *scikit-learn: machine learning in Python – scikit-learn 0.23.2 documentation*. 2020. URL: <https://scikit-learn.org/stable/index.html>.
- [29] *Deep Graph Library*. 2020. URL: <http://www.cs.ubc.ca/labs/beta/Projects/SATzilla/>.



- 
- [30] *GraphVite*. 2020. URL: <https://graphvite.io/>.
- [31] B. Bischl, P. Kerschke, L. Kotthoff, M. Lindauer, Y. Malitsky, A. Fréchet, H. H. Hoos, F. Hutter, K. Leyton-Brown, K. Tierney i J. Vanschoren. „ASlib: A Benchmark Library for Algorithm Selection”. Izvor: *Artificial Intelligence Journal (AIJ)* 237 (2016), strane 41–58.
- [32] J. Marković. „Primena Gausovih uslovnih slučajnih polja na predviđanje vremena izvršavanja SAT rešavača”. Master teza. 2018.
- [33] *SAT Challenge 2012: Results*. 2020. URL: <https://baldur.iti.kit.edu/SAT-Challenge-2012/results.html>.

## Dodatak A

# Opisi SATzilla atributa

<i>Oznaka atributa</i>	<i>Opis atributa</i>
nvarsOrig	Broj promenljivih u originalnoj formuli, $v$
nclausesOrig	Broj klauza u originalnoj formuli, $c$
nvars	Broj promenljivih nakon pojednostavljivanja, $v'$
nclauses	Broj klauza nakon pojednostavljivanja, $c'$
reducedVars	$(v - v')/v'$
reducedClauses	$(c - c')/c'$
Pre-featuretime	Vreme potrebno za pojednostavljivanje
vars-clauses-ratio	$v'/c'$

TABELA A.1: SATzilla atributi koji opisuju veličinu iskazne formule pre i posle preprocesiranja.

<i>Oznaka atributa</i>	<i>Opis atributa</i>
POSNEG-RATIO-CLAUSE-mean	Prosek
POSNEG-RATIO-CLAUSE-coeff-variation	Koeficijent varijacije
POSNEG-RATIO-CLAUSE-min	Minimum
POSNEG-RATIO-CLAUSE-max	Maksimum
POSNEG-RATIO-CLAUSE-entropy	Entropija
POSNEG-RATIO-VAR-mean	Prosek
POSNEG-RATIO-VAR-stdev	Koeficijent varijacije
POSNEG-RATIO-VAR-min	Minimum
POSNEG-RATIO-VAR-max	Maksimum
POSNEG-RATIO-VAR-entropy	Entropija
UNARY	Procenat klauza sa jednom promenljivom
BINARY+	Procenat klauza sa dve promenljive
TRINARY+	Procenat klauza sa tri promenljive

TABELA A.2: SATzilla atributi koji opisuju odnos pozitivnih i negativnih literala u klauzama (atributi sa prefiksom POSNEG-RATIO-CLAUSE), odnos pozitivnih i negativnih pojavljivanja promenljivih (atributi sa prefiksom POSNEG-RATIO-VAR) i procenat unarnih, binarnih i ternarnih klauza.

<i>Oznaka atributa</i>	<i>Opis atributa</i>
VCG-CLAUSE-mean	Prosek
VCG-CLAUSE-coeff-variation	Koeficijent varijacije
VCG-CLAUSE-min	Minimum
VCG-CLAUSE-max	Maksimum
VCG-CLAUSE-entropy	Entropija
VCG-VAR-mean	Prosek
VCG-VAR-coeff-variation	Koeficijent varijacije
VCG-VAR-min	Minimum
VCG-VAR-max	Maksimum
VCG-VAR-entropy	Entropija

TABELA A.3: SATzilla atributi koji opisuju statistike čvorova grafa promenljivih i klauza koje se odnose na klauze (atributi sa prefiksom VCG-CLAUSE) i promenljive (atributi sa prefiksom VCG-VAR).

<i>Oznaka atributa</i>	<i>Opis atributa</i>
VG-mean	Prosek
VG-coeff-variation	Koeficijent varijacije
VG-min	Minimum
VG-max	Maksimum

TABELA A.4: SATzilla atributi koji opisuju statistike čvorova grafa promenljivih.

<i>Oznaka atributa</i>	<i>Opis atributa</i>
CG-mean	Prosek
CG-coeff-variation	Koeficijent varijacije
CG-min	Minimum
CG-max	Maksimum
CG-entropy	Entropija
cluster-coeff-mean	Prosek
cluster-coeff-coeff-variation	Koeficijent varijacije
cluster-coeff-min	Minimum
cluster-coeff-max	Maksimum
cluster-coeff-entropy	Entropija

TABELA A.5: SATzilla atributi koji opisuju statistike čvorova grafa klauza (atributi sa prefiksom CG-) i koeficijente klasterovanja (atributi sa prefiksom cluster-coeff).

<i>Oznaka atributa</i>	<i>Opis atributa</i>
HORNY-VAR-mean	Prosek
HORNY-VAR-coeff-variation	Koeficijent varijacije
HORNY-VAR-min	Minimum
HORNY-VAR-max	Maksimum
HORNY-VAR-entropy	Entropija
horn-clauses-fraction	Procenat Hornovih klauza

TABELA A.6: SATzilla atributi koji opisuju statistike broja pojavljivanja u Hornovim klauzama za svaku promenljivu.

<i>Oznaka atributa</i>	<i>Opis atributa</i>
Pre-featuretime	Vreme potrebno za pojednostavljivanje
Basic-featuretime	Vreme potrebno za izračunavanje elementarnih statistika
KLB-featuretime	Vreme potrebno za izračunavanje atributa za graf promenljivih i klauza
CG-featuretime	Vreme potrebno za izračunavanje atributa za graf klauza

TABELA A.7: SATzilla atributi koji opisuju vremena potrebna za izračunavanje nekih od atributa.

## Dodatak B

## Detaljni rezultati eksperimenata

<i>SAT rešavač</i>	$R^2$ (KNN)	$R^2$ (RF)
ebglucose	0.6688	<b>0.7062</b>
ebminisat	0.6673	<b>0.7207</b>
glucose2	<b>0.7161</b>	0.7159
glueminisat	0.6856	<b>0.7271</b>
lingeling	0.6536	<b>0.7362</b>
lrglshr	0.68	<b>0.7429</b>
minisatpsm	0.6579	<b>0.7112</b>
mphaseSAT64	0.6408	<b>0.6653</b>
precosat	0.6584	<b>0.6958</b>
qutersat	0.6759	<b>0.7348</b>
rcl	0.6358	<b>0.7267</b>
restartsat	0.6413	<b>0.7453</b>
cryptominisat2011	0.6332	<b>0.6952</b>
spear-sw	0.6274	<b>0.6859</b>
spear-hw	0.6957	<b>0.7266</b>
eagleup	<b>0.6853</b>	0.6133
sparrow	0.7607	<b>0.8059</b>
marchrw	0.8091	<b>0.8709</b>
mphaseSATm	0.6391	<b>0.6989</b>
satime11	0.7059	<b>0.7565</b>
tnm	<b>0.6521</b>	0.6237
mx09	0.6343	<b>0.7117</b>
gnoveltyp2	0.7483	<b>0.7768</b>
sattime	0.6731	<b>0.7246</b>
sattimep	0.6646	<b>0.7242</b>
clasp2	0.6002	<b>0.6635</b>
clasp1	0.6455	<b>0.7326</b>
picosat	0.6748	<b>0.7652</b>
mphaseSAT	0.654	<b>0.7074</b>
sapperlot	0.73	<b>0.7967</b>
sol	0.6191	<b>0.6764</b>

TABELA B.1: Prosečne ocene  $R^2$  najboljih modela  $k$ -najbližih suseda i šume nasumičnih stabala, po SAT rešavačima. Najbolji rezultati su prikazani podebljanim slovima.

<i>SAT rešavač</i>	<i>RMSE (KNN)</i>	<i>RMSE (RF)</i>
ebglucose	0.6954	<b>0.655</b>
ebminisat	0.8842	<b>0.8102</b>
glucose2	<b>0.6747</b>	0.6749
glueminisat	0.8257	<b>0.7693</b>
lingeling	0.8134	<b>0.7098</b>
lrglshr	0.8533	<b>0.7649</b>
minisatpsm	0.7869	<b>0.7229</b>
mphaseSAT64	0.8824	<b>0.8518</b>
precosat	0.8553	<b>0.8071</b>
qutersat	0.7049	<b>0.6376</b>
rcl	0.7959	<b>0.6894</b>
restartsat	0.885	<b>0.7458</b>
cryptominisat2011	0.8108	<b>0.7392</b>
spear-sw	0.867	<b>0.796</b>
spear-hw	0.7956	<b>0.7541</b>
eagleup	<b>0.6667</b>	0.7391
sparrow	0.7265	<b>0.6544</b>
marchrw	0.6519	<b>0.5361</b>
mphaseSATm	0.9461	<b>0.8643</b>
satime11	0.7789	<b>0.7087</b>
tnm	<b>0.779</b>	0.8102
mxco9	0.875	<b>0.777</b>
gnoveltyp2	0.7293	<b>0.6868</b>
sattime	0.8307	<b>0.7623</b>
sattimep	0.7835	<b>0.7106</b>
clasp2	0.8867	<b>0.8135</b>
clasp1	0.7964	<b>0.6917</b>
picosat	0.8603	<b>0.731</b>
mphaseSAT	0.9243	<b>0.8501</b>
sapperlot	0.7635	<b>0.6625</b>
sol	0.7467	<b>0.6882</b>

TABELA B.2: Prosečne ocene *RMSE* najboljih modela *k*-najbližih suseda i šume nasumičnih stabala, po SAT rešavačima. Najbolji rezultati su prikazani podebljanim slovima.

<i>SAT rešavač</i>	$R^2$ (GCN)	$R^2$ (GAT)	$R^2$ (DGCNN)
ebglucose	-0.2244	-0.2892	<b>-0.0915</b>
ebminisat	-0.177	-0.1223	<b>-0.0452</b>
glucose2	-0.1794	-0.1996	<b>-0.0285</b>
glueminisat	-0.1858	-0.1367	<b>-0.0441</b>
lingeling	-0.2422	-0.1696	<b>-0.0779</b>
lrglshr	-0.1919	-0.1284	<b>-0.0605</b>
minisatpsm	-0.2302	-0.1946	<b>-0.0374</b>
mphaseSAT64	-0.1647	-0.1789	<b>-0.032</b>
precosat	-0.218	-0.1431	<b>-0.0557</b>
qutersat	-0.2569	-0.2289	<b>-0.0805</b>
rcl	-0.2286	-0.2093	<b>-0.0618</b>
restartsat	-0.2254	-0.1764	<b>-0.0434</b>
cryptominisat2011	-0.2592	-0.2205	<b>-0.0498</b>
spear-sw	-0.2807	-0.2643	<b>-0.082</b>
spear-hw	-0.2437	-0.2921	<b>-0.0883</b>
eagleup	-0.2117	-0.6304	<b>-0.1239</b>
sparrow	<b>-0.0316</b>	-0.3238	-0.0552
marchrw	-0.2034	-0.2954	<b>-0.0423</b>
mphaseSATm	-0.1932	-0.1917	<b>-0.0352</b>
satime11	<b>-0.0088</b>	-0.2607	-0.0467
tnm	<b>-0.0079</b>	-0.4039	-0.0985
mxco9	-0.1836	-0.1558	<b>-0.0256</b>
gnoveltyp2	<b>-0.0697</b>	-0.3353	-0.1001
sattime	<b>-0.0084</b>	-0.28	-0.0584
sattimep	<b>-0.0374</b>	-0.3457	-0.0964
clasp2	-0.2169	-0.1398	<b>-0.0703</b>
clasp1	-0.1879	-0.1645	<b>-0.0838</b>
picosat	-0.1933	-0.1582	<b>-0.0641</b>
mphaseSAT	-0.209	-0.1772	<b>-0.0264</b>
sapperlot	-0.2229	-0.2146	<b>-0.0599</b>
sol	-0.1338	-0.1987	<b>-0.0171</b>

TABELA B.3: Prosečne ocene  $R^2$  najboljih modela metoda zasnovanih na grafovskim neuronskim mrežama, po SAT rešavačima. Najbolji rezultati su prikazani podebljanim slovima.

<i>SAT rešavač</i>	<i>RMSE (GCN)</i>	<i>RMSE (GAT)</i>	<i>RMSE (DGCNN)</i>
ebglucose	1.337	1.372	<b>1.2624</b>
ebminisat	1.6631	1.624	<b>1.5672</b>
glucose2	1.3751	1.3869	<b>1.2842</b>
glueminisat	1.6035	1.57	<b>1.5047</b>
lingeling	1.5404	1.4947	<b>1.4349</b>
lrglshr	1.6469	1.6025	<b>1.5535</b>
minisatpsm	1.4922	1.4704	<b>1.3703</b>
mphaseSAT64	1.5891	1.5988	<b>1.4959</b>
precosat	1.615	1.5646	<b>1.5036</b>
qutersat	1.3881	1.3725	<b>1.287</b>
rcl	1.4618	1.4503	<b>1.359</b>
restartsat	1.6359	1.6029	<b>1.5095</b>
cryptominisat2011	1.5023	1.4791	<b>1.3717</b>
spear-sw	1.6074	1.5971	<b>1.4774</b>
spear-hw	1.6085	1.6395	<b>1.5047</b>
eagleup	1.3083	1.5176	<b>1.26</b>
sparrow	<b>1.5086</b>	1.709	1.5258
marchrw	1.6368	1.6983	<b>1.5234</b>
mphaseSATm	1.7205	1.7194	<b>1.6025</b>
satime11	<b>1.4424</b>	1.6124	1.4692
tnm	<b>1.326</b>	1.5649	1.3843
mx09	1.5744	1.5558	<b>1.4655</b>
gnoveltyp2	<b>1.5035</b>	1.6798	1.5247
sattime	<b>1.4589</b>	1.6436	1.4946
sattimep	<b>1.378</b>	1.5694	1.4166
clasp2	1.547	1.4972	<b>1.4508</b>
clasp1	1.4579	1.4435	<b>1.3925</b>
picosat	1.6479	1.6235	<b>1.5562</b>
mphaseSAT	1.7279	1.705	<b>1.5921</b>
sapperlot	1.6248	1.6192	<b>1.5126</b>
sol	1.2883	1.3246	<b>1.2201</b>

TABELA B.4: Prosečne ocene *RMSE* najboljih modela metoda zasnovanih na grafovskim neuronskim mrežama, po SAT rešavačima. Najbolji rezultati su prikazani podebljanim slovima.