



Creative Commons Attribution –  
NonCommercial 4.0 International License

Izvorni znanstveni rad

<https://doi.org/10.31784/zvr.7.1.4>

Datum primitka rada: 28. 12. 2018.

Datum prihvatanja rada: 7. 3. 2019.

# PRIMJENA ŠUME SLUČAJNIH STABALA ZA PREDVIĐANJE ISHODA ŠAHOVSKE PARTIJE REPREZENTIRANE KOMPLEKSNO MREŽOM

**Jovan Jokić**

Mag. inf., student, Odjel za informatiku, Sveučilište u Rijeci, Radmile Matejčić 2, 51000 Rijeka,  
Hrvatska; e-mail: jovan.jokic@xnet.hr

**Sanda Martinčić-Ipšić**

Dr. sc., izv. prof., Odjel za informatiku, Sveučilište u Rijeci, Radmile Matejčić 2, 51000 Rijeka,  
Hrvatska; e-mail: smarti@uniri.hr

## SAŽETAK

Cilj je ovog rada izgraditi model vrednovanja šahovskih pozicija koji se temelji na informacijama dobivenima iz mrežnih značajki pozicija šahovskih figura na ploči. Međusobni odnosi figura, kao i odnosi figura i polja koja zauzimaju, a koji opisuju određene taktičke i strateške elemente šahovske partije, mogu se modelirati kompleksnom mrežom. Ovim je radom pokazano kako upotrebom samo određenih mjera koje opisuju strukturu kompleksne mreže možemo naučiti klasifikator pozicija koji može predvidjeti krajnji ishod partije bolje od klasične Shannonove evaluacijske funkcije. Shannonova evaluacijska funkcija kvantificira materijalno stanje na ploči obiju strana u šahovskoj igri, mobilnost figura, sigurnost kralja te kvalitetu pješačke strukture. Računalni model koji klasificira na ulazu ima vektore značajki određene iz četiri vrste mreža (mreža podrške, mreža mobilnosti, pozicijska mreža, mreža praćenja) konstruiranih iz baze majstorskih partija, od kojih svaka modelira određeni aspekt šahovske igre. Vektor značajki sadrži značajke dobivene izračunom različitih mjera strukture mreže. Za određivanje značajki prema važnosti, kao i klasifikacijski postupak ishoda igre, upotrebljava se šuma slučajnih stabala. Nadalje, eksperimentalno se određuju osnovne evaluacije statičkih pozicija u šahovskim partijama pomoću Stockfish šahovskog programa. Nakon toga, značajke te pripadne evaluacijske ciljne klase (pobjeda bijelog, pobjeda crnog igrača ili remi) udružuju se u ulazne vektore za učenje modela klasifikacije pozicija, čiji se rezultati uspoređuju sa Shannonovom evaluacijskom funkcijom. Pokazano je kako predložena metoda vrednovanja temeljena na informacijama o strukturi mreže daje bolje rezultate (75 % točnosti) od klasične Shannonove evaluacijske funkcije (52 % točnosti) za testnu bazu partija.

**Ključne riječi:** evaluacijska funkcija u šahu, kompleksne mreže, mrežne značajke, šuma slučajnih stabala, predviđanje ishoda šahovske igre

## **1. UVOD**

Međusobni odnosi figura u šahovskoj partiji mogu se prikazati evoluirajućim kompleksnim mrežama (evolucija grafa u ovom kontekstu podrazumijeva promjenu strukture grafa, odnosno, rast mreže reflektiran u svakom sljedećem potezu) koje opisuju taktičke i strateške elemente šahovske igre (Farren et al., 2013). Mreže sadrže vrijedne informacije o partiji, o njezinim prijelomnim trenucima te ishodu, čak i kada šahovsku partiju prikazemo kao mrežu međusobnih odnosa figura, izostavljajući ostale pojedinosti o partiji, poput vrijednosti figura ili važnih strateških koncepata, poput šahovskog centra. Prikazavši partiju kao niz mreža, od kojih svaka modelira određeni aspekt veza između figura i ploče, moguće je prikupiti još takvih informacija. Nadalje, razmotri li se i evolucija grafova, odnosno, mreža u vremenskom tijeku koji odgovara tijeku partije, konačan model može se proširiti i vremenskom informacijom. Pomoću niza evoluirajućih mreža, odnosno, informacija izlučenih iz mreža rastućih kroz vrijeme, samo s minimalnom reprezentacijom pravila i logike same šahovske igre moguće je generirati vjerodostojnu evaluaciju svake pozicije u igri (Jokić, 2014; Jokić, 2018).

U ovom se radu razmatraju informacije koje mogu biti dobivene iz konstruiranih mreža, tj. kumulativni mrežni podaci dobiveni iz evolucije mreža igre te se pokazuje kako one mogu uspješnije predvidjeti ishod partije od Shannonove klasične evaluacijske funkcije (Shannon, 1988). Evaluacijska je funkcija složena formula, svojevrsni agregirani kvantifikator procjene igre, koji kvantificira sve značajke u proizvoljnoj poziciji i dodjeljuje im težine koje reflektiraju kako taktičke, tako i strateške kvalitete pozicije (Chessprogramming, 2018). Stratešku kvalitetu pozicije zahtjevno je matematički modelirati pa, iako su moderni šahovski programi davno nadmašili najbolje svjetske igrače u toj sposobnosti procjene, to još uvijek predstavlja zahtjevan problem (Atkin i Witten, 1975; Fernández i Salmerón, 2007; Farren et al., 2013; Lei, 2015; Silver et al., 2017; Jokić, 2018). Razmatrajući pokazana svojstva takvih šahovskih programa i sustava, poput današnjih najboljih šahovskih paketa (engl. engines) Stockfish (Stockfish 10, 2018) i Komodo (Komodo 12, 2018), neizbježno se nameće zaključak kako postoji malo mogućnosti i prostora za poboljšanjem rada takvih programa, s posebnim naglaskom na modernoj evaluacijskoj funkciji (Fernández, 2008).

U ovom se radu upotrebljavaju principi strojnog učenja (James et al., 2013; Stuart et al., 2015, Silver et al., 2017) koji se često primjenjuju i pri predviđanju ishoda šahovske igre (Fernández, 2007) ili igre Go (Stern et al., 2006). Nadalje, značajke, odnosno, vektori značajki koji se koriste u postupcima učenja šume slučajnih stabala (Breiman, 2001) temelje se na mjerama kompleksnih mreža (Newman, 2018). Kompleksne mreže primjenjuju se na raznim područjima, poput predviđanja novih veza u društvenim mrežama (Martinčić-Ipšić et al., 2017), određivanja ključnih riječi u tekstu (Beliga et al., 2015; Beliga et al., 2016), modeliranja jezika (Martinčić-Ipšić et al., 2016), analize koautorstva (Meštrović i Grubiša, 2015) i sličnog, dok je šahovska igra samo mjestimično modelirana kroz formalizme kompleksnih mreža, i to u radu (Farren et al., 2013), stoga je to upravo i predmet istraživanja.

U radu se istražuju mogućnosti modeliranja šahovske igre kroz formalizam kompleksnih mreža, s ciljem dobivanja boljeg uvida u razvoj šahovske partije, te uvida u način na koji informacije

dobivene iz strukture kompleksnih mreža utječu na predviđanje konačnog ishoda partije. U radu su navedeni različiti načini prikaza šahovske partije u formalizmu kompleksne mreže te su opisane značajke koje je moguće dobiti analizom strukture mreža. Nakon toga, svojstva mreže, kvantificirana u mjerama mreže, upotrebljavaju se za predviđanje ishoda partije izgradnjom klasifikatora. Klasifikacijski model gradi se pomoću algoritma strojnog učenja – šume slučajnih stabala - koji je po prvi puta primijenjen na problem predviđanja ishoda šahovske igre na temelju značajki dobivenih iz strukture kompleksne mreže. U radu se također pokazuje kako je moguće odrediti skup značajki koji najbolje pridonosi točnosti modela klasifikacije pozicija, odnosno, predviđanja ishoda igre.

## 2. METODOLOGIJA ISTRAŽIVANJA

### 2.1 Shannonova evaluacijska funkcija

Šahovska se igra proučava s puno različitih polaznih gledišta. Tako je, u domeni računalne znanosti, Claude Shannon, 50-ih godina 20. stoljeća, u svom teorijskom postulatu o mogućem radu šahovskih programa (Shannon, 1988), naveo donju granicu kompleksnosti stabla igre, odnosno, prostora stanja šaha, te evaluacijsku funkciju oblika:

$$f(K,Q,R,B,N,P,DP,BP,IP,M) = 200(K - K') + 9(Q - Q') + 5(R - R') + 3(B - B' + N - N') + (P - P') - 0.5(DP - DP' + BP - BP' + IP - IP') + 0.1(M - M'), \quad (2.1)$$

pri čemu su:

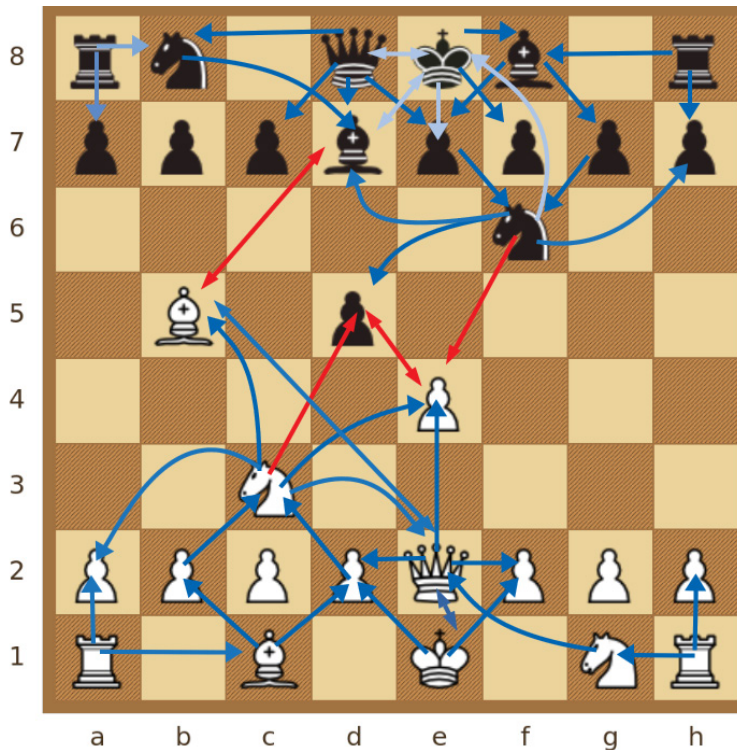
- **K, Q, R, B, N, P** oznake za broj bijelih figura: **K** - kralja, **Q** - dame, **R** - topova, **B** - lovaca, **N** - skakača i **P** – pješaka,
- **DP, BP, IP** oznake za broj **DP** - udvostručenih, **BP** - zaostalih i **IP** - izoliranih bijelih pješaka
- **M** oznaka predstavlja faktor mobilnosti bijelog igrača (mjenjen npr. brojem mogućih legalnih poteza koje ima na raspolaganju u danoj poziciji),
- **negirane oznake** (') predstavljaju analogne vrijednosti navedenih mjera za crnog igrača.

Iako su moderne evaluacijske funkcije značajno unaprijeđene od izvorne Shannonove evaluacijske funkcije, u ovome je radu upravo taj model primijenjen na osnovu s kojom se uspoređuje uspješnost predloženog rješenja. Razlog tomu je, ponajprije, jednostavnost strojne implementacije Shannonove evaluacijske funkcije te činjenica da moderni šahovski programi upotrebljavaju iznimno složene formule za evaluaciju statičkih pozicija (npr. dubinske neuronske mreže), koje je teško interpretirati. Nadalje, evaluacijske funkcije koje je moguće interpretirati najčešće predstavljaju linearnu kombinaciju nezavisnih značajki i njihovih pripadnih vrijednosti, poput Shannonove funkcije upotrijebljene u ovome radu, stoga je za usporedbu rezultata odabrana osnovna, ali jednostavno primjenjiva evaluacijska funkcija.

## 2.2 Kompleksne mreže

### 2.2.1 Potporna mreža (engl. support network)

Slika 1. Prikaz formiranja veza između figura na ploči – izgradnja potporne mreže

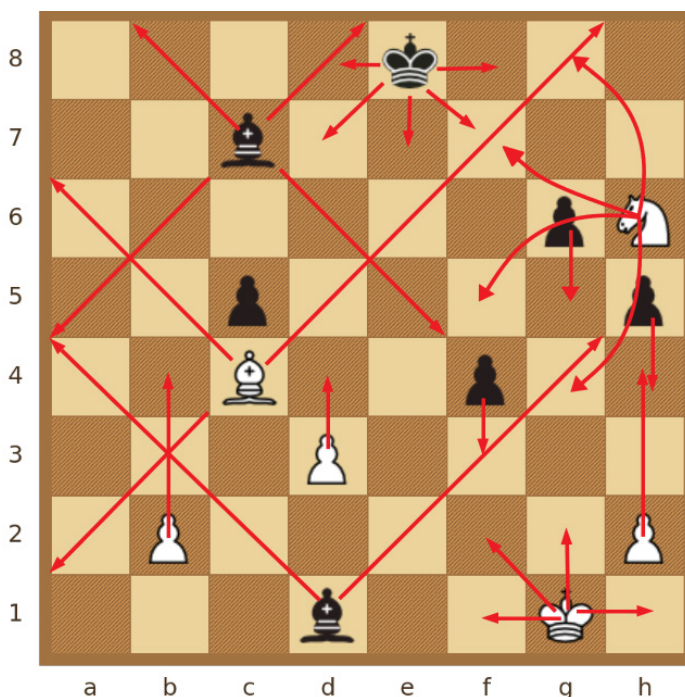


Prva vrsta kompleksne mreže koja se konstruira iz informacija o partiji takozvana je potporna mreža, koja modelira isključivo odnose između figura na ploči. U potpornoj mreži svaki čvor predstavlja figuru i iste je boje kao figuru koju predstavlja. Kad figura ima potencijal uzimanja protivničke figure, taj je potencijal u potpornoj mreži prikazan vezom. U slučaju da figura napada figuru iste boje (što zapravo nije predviđeno pravilima igre), taj isključivo teorijski potencijal uzimanja vlastite figure također je moguće prikazati vezom. Sve su veze usmjerene i obojene s obzirom na boju figure koju predstavlja početni čvor veze. Nadalje, svakom je čvoru dodijeljen atribut koji predstavlja tip potencijala napada s vrijednostima u skupu ('napad', 'obrana'). Napadačke su veze između čvorova različite boje, dok su obrambene veze, analogno tome, veze između čvorova iste boje. Primjer takve mreže, s opisanim svojstvima, ilustriran je na Slici 1, koja sadrži proizvoljno zadanu poziciju na ploči, neposredno iz otvaranja igre. Potencijali uzimanja protivničke figure prikazani su na Slici 1, i to crvenom linijom od ishodišne figure, koja završava s odredišnom strelicom na polju napadnute figure (npr. C3→D5). Analogno tome, obrambeni potencijal svake figure naznačen je strelicama plave boje s ishodištem u polju obrambene figure (npr. G7→F6) te odredištem strelice na polju vlastite branjene figure.

### 2. 2. 2 Mreža mobilnosti (engl. *mobility network*)

Druga je vrsta konstruirane kompleksne mreže tzv. mreža mobilnosti. Ona modelira mobilnost figura tako da je svaki čvor u mreži mobilnosti jedno polje na ploči. Polje može biti zauzeto nekom figurom ili izravno dostupno potezom neke figure. Svaki čvor koji predstavlja polje na kojem se nalazi figura preuzima boju figure koja ga okupira, tj. izvorišne figure. Mrežni čvorovi koji predstavljaju prazna polja nisu obojeni. Veze u mreži mobilnosti predstavljaju potencijal figure da uzme protivničku figuru ili se pomakne na dostupno prazno polje. Svaka je veza usmjerena i izvire iz čvora izvorišne figure. Odredište je svake veze čvor koji predstavlja bilo polje na kojem se nalazi figura napadnuta od strane izvorišne figure (isto kao i u potpornoj mreži), bilo prazno polje koje je dostupno izvorišnoj figuri veze. Ne postoje veze između čvorova koji predstavljaju polja okupirana figurama iste boje. Čvorovima koji predstavljaju prazna polja izlazni je stupanj vrijednosti nula, dakle nisu povezani. Svaka je veza obojena bojom izvorišnjog čvora. Radi dosljednosti, svakoj je vezi dodijeljen atribut s vrijednošću 'napad'.

Slika 2. Prikaz veza u mreži mobilnosti



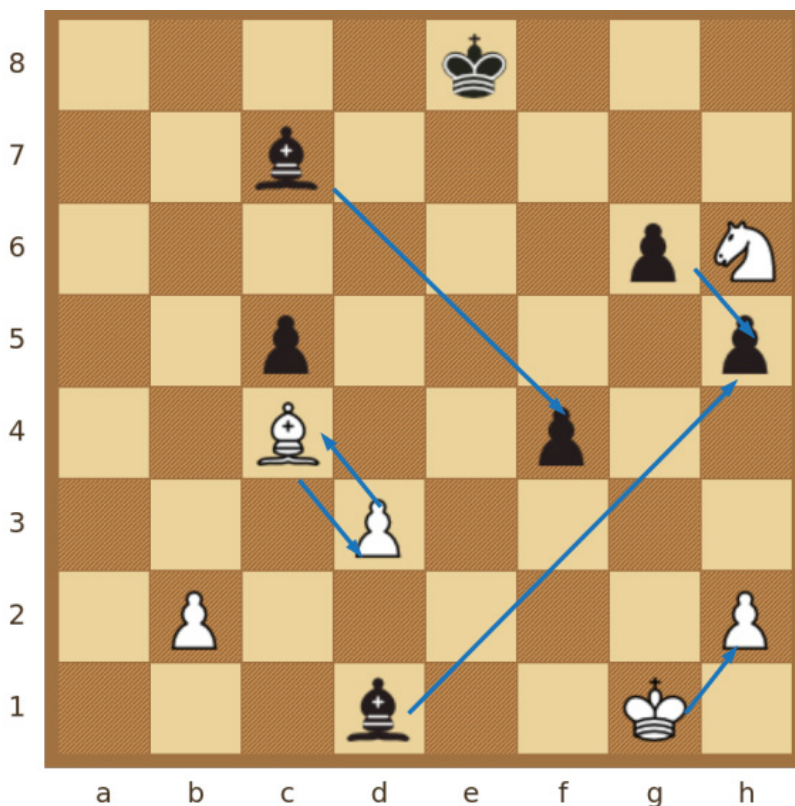
Izvor: autori

### 2. 2. 3 Pozicijska mreža (engl. *position network*)

Treća je vrsta konstruirane kompleksne mreže tzv. pozicijska mreža - u osnovi spoj prethodnih dviju mreža - mreže mobilnosti i potporne mreže, a modelira odnose između figura i polja koja

mogu doseći. Primjer pozicije na dijagramu na kojem su naznačene dodatne veze u pozicijskoj mreži, u odnosu na mrežu mobilnosti, prikazan je na Slici 3.

Slika 3. Prikaz veza u pozicijskoj mreži



Izvor: autori

#### 2.2.4 Mreža praćenja (engl. *tracking network*)

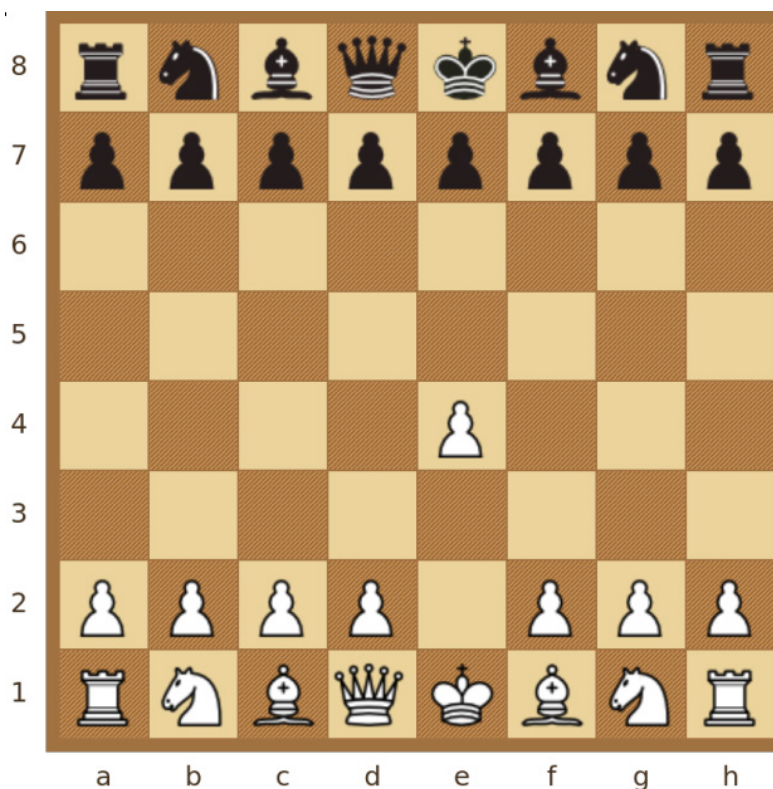
Četvrta je vrsta konstruirane kompleksne mreže tzv. mreža praćenja, koja modelira povijest kretanja figura kroz polja na šahovskoj ploči. To je tip mreže koji se upotrebljava za daljnje postupke izračuna, koji nije moguće dobiti iz statičke pozicije na ploči. U mreži praćenja svako je polje ploče predstavljeno čvorom. U tijeku igre, kada igrač na potezu pomakne figuru s izvorišnog na odredišno polje, taj se potez u mreži praćenja označava vezom od izvorišnog do odredišnog polja, odnosno, čvora mreže. Jednom kada se uspostavi veza, ona se ne briše do kraja partije. Svaki potez inkrementalno dodaje veze u mrežu, što znači da konačna mreža sadrži informacije, tj. povijest svih poteza odigranih u partiji, no ne i njihov vremenski slijed. Također, za figure dame, topa, lovca i pješaka, koji se mogu pomaknuti dalje od susjednog polja, stvaraju se dodatne veze za svako od prijedjenih prethodnih polja, počevši s izvorišnim poljem i prijedjenim poljem te završavajući s odredišnim poljem. Svaka veza u mreži obojena je kao i

figura čije je kretanje rezultiralo stvaranjem te veze. Radi dosljednosti, svim je vezama dodijeljen atribut tipa 'napad'.

Na Slici 4, bijeli igrač odigrao je svoj prvi potez u partiji, potez kraljevim pješakom s polja E2 na polje E4. Tim prvim potezom, kao i svakim sljedećim potezom, nastaju nove veze iz izvorne mreže slijeda, koja je analogna početnoj poziciji u partiji prije prvog poteza. Tim prvim potezom, nastaju prve tri veze u mreži slijeda, a to su:

- $E2 \rightarrow E4$ ,  $E2 \rightarrow E3$ ,  $E3 \rightarrow E4$ .

Slika 4. Pozicija nakon prvog poteza bijelog igrača na E4



Izvor: autori

### 3. EMPIRIJSKI PODACI I ANALIZA

#### 3.1 Upotrijebljene mrežne mjere u modelu

U radu se eksperimentalno analizira niz mrežnih svojstava, tj. mjera koje se upotrebljavaju za predviđanje krajnjeg ishoda partije. Upotrebljavaju se mrežne mjere u vektorima značajki, točnije, razlika njihovih vrijednosti za bijelu i crnu komponentu svake od mreža, i to za svaku

od četiri konstruirane mreže pojedinačno. Upotrijebljene mjere kompleksne mreže definirane su standardno prema formulama u Jokić (2018), Martinčić-Ipšić (2019), Newman (2018), stoga su ovdje samo navedene, bez definicije: najmanji stupanj čvora u mreži, najveći stupanj čvora u mreži, prosječni stupanj čvora, medijan distribucije stupnjeva čvorova, gustoća mreže/grafa, broj čvorova, broj veza i broj jako povezanih komponenti.

Iako mrežna svojstva, tj. mjere, modeliraju neke pozicijske i strateške aspekte igre, promatrana agregirano, količina je informacija o poziciji koje možemo dobiti iz jedne mreže mala te stoga nedostatna za zadovoljavajuću točnost predviđanja ishoda partije. Zbog toga će se navedene agregirane mrežne značajke (za svaku od četiri vrste opisanih mreža) kombinirati u jedan konačan vektor značajki s ukupno 33 vrijednosti, a koji će biti opisan u sljedećem poglavlju.

### 3.2 Učenje šume slučajnih stabala i određivanje vektora značajki

Klasifikacija u strojnom učenju ima zadatak razvrstati objekte u unaprijed definirane klase ili razrede. Kako bi se objekt mogao klasificirati, potrebno ga je vjerodostojno opisati podacima - značajkama (engl. *features*). Značajke se dobivaju pomoću različitih numeričkih izračuna nad izvornim podacima (James et al., 2013). Primjer je jedne takve značajke srednja vrijednost intenziteta svjetline određene regije slike. Budući da jedna značajka ne može vjerodostojno predstavljati sve osobine modeliranog objekta, u većini se slučajeva izračunava veći broj značajki.

Nakon izračuna značajki i izgradnje vektora značajki za svaki potez u partiji, za svaku od partija iz baze majstorskih partija iz PGN (engl. *Portable Game Notation*) datoteka upotrebljava se algoritam strojnog učenja - šuma slučajnih stabala (engl. *Random forest*) (Breiman, 2001). Algoritam iz raspoloživih podataka za učenje stvara veći broj (ansambl) (engl. *ensemble*) stabala odlučivanja - šumu slučajnih stabla. Pri konstrukciji svakog od stabala u šumi algoritam slučajnim odabirom uzima podskup podatkovnih instanci i podskup ulaznih varijabli. Prilikom izgradnje stabla, odluka o binarnoj podjeli podatka u čvoru donosi se prema funkciji koja određuje najbolju podjelu. Vremenska je kompleksnost tog algoritma procijenjena s  $O(MKN \log^2 N)$ , gdje je  $M$  broj stabala,  $N$  broj podatkovnih instanci,  $K$  broj varijabli. Uobičajeno se kao funkcija za određivanje grananja upotrebljava Gini indeks. Varijabla koja postigne najveću vrijednost Gini indeksa upotrebljava se za podjelu podataka u tekućem čvoru.

Gini indeks definiran je kao:

$$I_G(t) = 1 - \sum_{i=1}^m p_i^2$$

gdje je  $t$  trenutni čvor,  $p_i$  vjerojatnost klase  $i$  u čvoru  $t$ , a  $m$  je broj klasa u modelu (u ovom je radu  $m=3$ ).

Konačno predviđanje šume slučajnih stabala izračunava se kao prosjek predviđanja pojedinačnih stabala u šumi. Pokazano je da je taj model robustan i otporan na prekomjerno prilagođavanje podacima za učenje (engl. *overfitting*) te da postiže bolju točnost predviđanja od pojedinačnih



stabala. Zbog svih dobrih svojstava, ovaj se algoritam često primjenjuje u različitim područjima istraživanja (Martinčić-Ipšić, 2019). Nadalje, dobro je svojstvo tog algoritma i njegova sposobnost da odredi značaj varijabli (atributa). Ukoliko usporedimo pogrešku predviđanja kod stabla naučenog na cijelom skupu podataka za učenje s pogreškom predviđanja u stablu naučenom na podskupu podataka, koje smo slučajno permutirali tako da vrijednosti varijable (značajke) koju promatramo slučajno rasporedimo preko svih primjera u tom skupu, utvrdit ćemo utjecaj tj. važnost varijable. Veliko odstupanje pogreške predviđanja u tim dvama modelima ukazuje na važnost te varijable. To znači da slučajna permutacija vrijednosti značajno utječe na točnost predviđanja, što posljedično znači da je utjecaj te varijable na točnost predviđanja velik, dakle varijabla ima veliki značaj (engl. *variable importance*). Vrijedi i obrnuto: ukoliko je razlika u pogreškama mala, značaj varijable nije velik. U radu se upotrebljava MeanDecreaseGini - mjera značaja varijabli koja je zasnovana na Gini indeksu i koja se upotrebljava pri postupku određivanja važnosti varijabli (engl. *variable importance*). Testni skup partija različit je od skupa partija upotrebljivanih za učenje modela predviđanja - skupa za učenje (Stuart et al., 2005; Lai, 2015). Dobiveni rezultati predviđanja uspoređuju se sa Shannonovom evaluacijskom funkcijom na testnom skupu partija.

Za izgradnju modela predviđanja upotrebljavaju se vektori značajki koji sadrže vrijednosti značajki za svaku od četiri opisane moguće vrste mreža upotrijebljenih za analizu, dakle, ukupno 32 vrijednosti (4x8), te za klasu (razred) koja je zapisana u 33. varijabli. Detaljan popis značajki nalazi se u privitku A ovoga rada, dok je ovdje naveden skraćeni oblik. Značajke V1 - V8 redom odgovaraju najmanjem, najvećem i prosječnom stupnju čvora u potpornoj mreži, medijanu distribucije stupnjeva čvorova, gustoći, broju čvorova, broju veza i broju jako povezanih komponenti u potpornoj mreži. Značajke V9 - V17 kvantificiraju osobine mreže mobilnosti, V18 - V25 pozicijske mreže, dok značajke V26 - V32 opisuju mrežu praćenja. Zadnja vrijednost, V33, predstavlja kategoričku varijablu odabira, tj. jednu od tri moguće vrijednosti {pobjeda crnog, remi, pobjeda bijelog}, od kojih svaka predstavlja predviđeni najvjerojatniji ishod partije iz dane pozicije, uzevši u obzir optimalnu igru oba igrača. To je klasa predviđena modelom, a može dobiti sljedeće vrijednosti:

-1 → pobjeda igrača s crnim figurama,

0.5 → remi-ishod,

1 → pobjeda igrača s bijelim figurama.

Dobivene evaluacijske vrijednosti preslikaju se u krajnju evaluacijsku mjeru na temelju raspona vrijednosti kojemu evaluacijska vrijednost pripada. Preslikavanje izračunanih evaluacijskih vrijednosti za svaku poziciju u skup od tri moguće vrijednosti dobiveno je distribuiranom analizom baze majstorskih partija pomoću šahovskog programa Stockfish, i to na sljedeći način: vrijednosti iz intervala  $[-MAX\_VRIJEDNOST, -0.51]$  pretvore se u vrijednost klase -1 - pobjeda crnog; vrijednosti iz  $[-0.5, +0.5]$  u 0 - remi, te vrijednosti iz  $[0.51, +MAX\_VRIJEDNOST]$  u 1 - pobjeda bijelog igrača.

Jedinice upotrebjavane u evaluacijskim vrijednostima šahovskih programa, poput Stockfisha takozvani su centi-pješaci, odnosno, 100-ti dio vrijednosti materijalne prednosti u vidu jednog

pješaka više (100 centi-pješaka). S obzirom na trenutnu poziciju, nema konkretno definiranih granica, tj. raspona tako definiranih evaluacijskih vrijednosti koje 100% koreliraju s objektivnim krajnjim vjerojatnim ishodima partije, odnosno, evaluacijskom ocjenom dane statičke pozicije za oba igrača, dakako, pretpostavivši optimalnu igru oba igrača. Njemački je logičar i matematičar Ernst Zermelo 1913. godine objavio djelo u kojemu je dokazao da je optimalna strategija pri igranju šaha strogo određena (Zermelo, 1913). Zbog kompleksnosti šahovske igre, nije uvijek vidljivo koje su to strategije, no dokazano je da postoje. U ovom kontekstu, pojam optimalno odnosi se na onakvu strategiju koja će igraču maksimalno pospješiti vjerojatnost pobjede, odnosno, umanjiti vjerojatnost poraza (min-max strategija).

### **3.3 Distribuirana obrada baze majstorskih partija**

U radu se koristi ICOfy IB1421 PGN (engl. Portable Game Notation) baza koja sadrži sve partije za potrebe ove analize. Baza je filtrirana tako da uključuje odlučujuće partije s pobjedom jednog od igrača, bez remi-ishoda. Time se osigurava dovoljan broj pozicija za analizu u kojima jedan od igrača ima odlučujuću prednost nad protivnikom, s obzirom na to da je većina pozicija tijekom majstorskih partija u okviru remi-ishoda. Naime, odlučujuća se prednost u takvim partijama stječe tek u poodmakloj fazi partije. Filtriranjem remi-ishoda podatkovni se skup iz šahovske baze partija smanji na približno 62 200 partija. Važno je napomenuti da su na ovaj način preuzeti svi potezi iz filtriranih partija, koji su nazvani statičkim pozicijama. Za svaku statičku poziciju moguće je odrediti utječe li na pobjedu crnog, remi ili pobjedu bijelog igrača, što je opisano u nastavku.

Šahovska se baza dodatno predprocesira tako da se podijeli u PGN datoteke od kojih svaka sadrži 5000 partija. Naposljetku, u radu se upotrebljava samo prvih 5000 partija iz te baze, što se pokazalo dostatnim za eksperimente izvedene i opisane u nastavku rada.

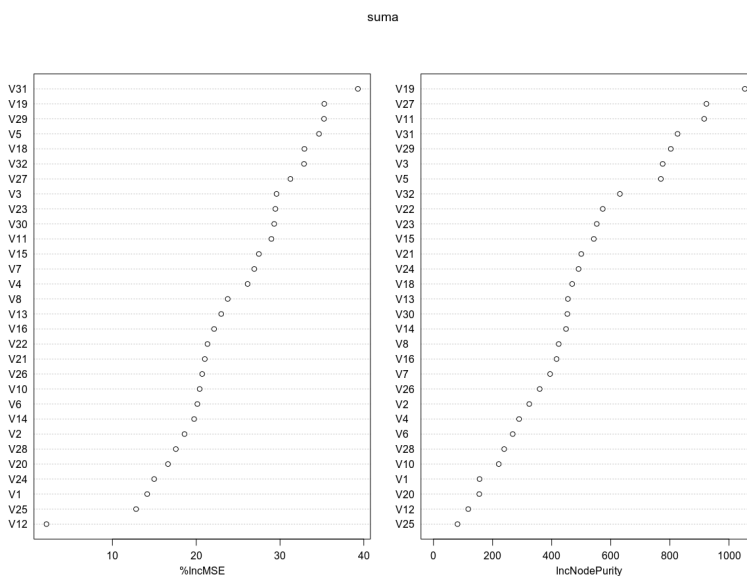
Za obradu je napisana Python skripta koja distribuirano učitava svaku od partija u jednom od ukupno 2-4 konkurentna procesa, od kojih svaki izvršava svoju instancu Stockfish šahovskog programa (V8 u trenutku pisanja rada (Stockfish, 2018)) nad jednom partijom te analizira sve poteze partije. Skripta, odnosno, svaka pokrenuta instanca Stockfish programa izračunava svaki potez, tj. statičku poziciju nastalu na ploči kroz tijek partije te zapisuje u globalni rječnik ključ-vrijednost par koji sadrži numeričku evaluaciju pozicije. Globalni je rječnik Python rječnik koji dijeli svaki od zasebno pokrenutih procesa, a višeprocesni upravljač koordinira radom procesa i brine se o kontroli pristupa za dijeljeni resurs. Svaki par sastoji se od ključa koji predstavlja statičku poziciju pomoću Zobrist\* hash-vrijednosti (Zobrist, 1969), te njezinu Shannonovu evaluaciju. Time je izbjegnuto udvostručavanje izračuna evaluacije za već poznate pozicije, nauštrb veće varijacije u samom izračunu. No s obzirom na to da nije potrebna veća preciznost, tj. dubina pretrage za svaku poziciju, ovaj je pristup dostatan, posebice stoga što se navedene vrijednosti ne analiziraju numerički, već se pozicije klasificiraju u tri moguća skupa, odnosno, kategoričke vrijednosti koje predstavljaju tri moguća ishoda partije – pobjedu jednog od igrača te remi-ishod. Python skripte za distribuiranu obradu partija imaju linearnu vremensku složenost, tj.  $O(n)$ .

## 4. REZULTATI KLASIFIKACIJE

### 4.1 Izgradnja šume slučajnih stabala i određivanje značaja varijabli

Nakon tvorbe konačnog skupa podataka za analizu, koji sadrži instance koje opisuju poteze, uči se model određivanja ishoda partije pomoću Random Forest (RF) biblioteke u R-programskom jeziku (Liaw, Wiener, 2018), odnosno, alatu za statističku analizu podataka koji omogućuje provođenje postupaka linearne regresije i klasifikacije (James, 2013). Prvi je korak pronalazak varijabli/mrežnih značajki koje više utječu na točnost predviđanja – klasifikacije poteza. U radu je pokazano i kako je moguća izgradnja manjeg klasifikacijskog modela na temelju smanjenog skupa značajki (visokorangiranih s obzirom na značaj), a koji nema smanjenu točnost predviđanja. Taj postupak opisuju rezultati na Slici 5, koja prikazuje dva grafa koja sadrže odabrane mrežne značajke u RF modelu (y-os), te dvije mjere na x-osi koje opisuju njihov utjecaj na porast, odnosno, pad točnosti u generiranom RF modelu (engl. variable importance). Dvije su mjere: %IncMSE (u lijevom dijelu) te IncNodePurity (u desnom dijelu slike). %IncMSE odražava porast MSE po varijabli (engl. mean-squared error), dakle odražava važnost varijable s obzirom na točnost predviđanja. IncNodePurity odražava gubitak koji proizlazi iz postupka odabira grananja na svakom koraku algoritma. Veći značaj varijabli (značajki) odgovara većoj vrijednosti %IncMSE i IncNodePurity vrijednosti s x-osi, i obrnuto, dakle, značajka V31 najviše je rangirana prema %IncMSE kriteriju, dok je značajka V19 najviše rangirana prema IncNodePurity kriteriju. Nadalje, iz slike je razvidno da je pet najviše rangiranih značajki V31, V19, V29, V5 i V18 prema %IncMSE kriteriju, odnosno, V19, V27, V11, V31 i V29 prema IncNodePurity kriteriju.

Slika 5. Graf utjecaja važnosti značajki dobiven u R-u



Izvor: autori

#### 4.2 Rezultati predviđanja ishoda šahovskih partija

U nastavku su prikazani rezultati predviđanja ishoda šahovskih partija na temelju svih 32 značajki uključenih u učenje modela. Za evaluaciju se upotrebljava konfuzijska matrica i računa mjera točnosti (engl. accuracy - ACC). U konfuzijskoj je matrici za svaku od klasa napisan broj točnih (TP) i broj pogrešnih (TN) klasifikacijskih primjera. Mjera točnosti tada predstavlja omjer pravilno klasificiranih primjera (TP + TN) u odnosu na ukupan broj primjera:

$$ACC = \frac{TP+TN}{TP+FP+TN+FN} \quad (4.2)$$

gdje FP predstavlja broj lažno pozitivnih klasifikacijskih primjera, a FN broj lažno negativnih klasifikacijskih primjera. Budući da u ovome radu nema klasične binarne klasifikacije, već se radi s tri klase, za evaluaciju se upotrebljava prilagođena mjera točnosti. Računa se kao omjer broja točno klasificiranih primjera za danu klasu prema ukupnom broju stvarnih primjera za tu klasu. Uz to, radi se i s neuravnoteženim klasama (broj pojedinih primjera iz svake klase nije ujednačen: klasa -1 sadrži 9 % primjera, klasa 0.5 73 %, dok klasa 1 sadrži 18 % primjera), za koje je primjerena mjera točnosti za klasu (engl. per-class accuracy). Rezultati evaluacije dobiveni su za sve značajke (ukupno 32) koje su upotrijebljene za izgradnju klasifikacijskog modela RF. Tablica 1 prikazuje matricu konfuzije koja sadrži rezultate klasifikacije na testnom skupu podataka.

Tablica 1. Matrica konfuzije za sve varijable u modelu za testni skup podataka o šahovskim partijama

stvarne klase - ishodi partija	predviđene klase – ishodi partija				pogreška klasifikacije
	-1	0.5	1		
-1	391	76	447		0.5722
0.5	497	6281	582		0.1466
1	710	229	893		0.5126

Od svih uzoraka za testiranje (ukupno 914), u stablu koje pripada klasi -1 (pobjeda crnog), točno je klasificiran (predviđeno ishoda partije) 391 uzorak. Dakle, točnost klasifikacije za klasu -1  $ACC_{-1}$  iznosi:

$$ACC_{-1} = \frac{391}{391+76+447} = 0.427789, \quad (4.3)$$

odnosno, 42.78 %, čime pogreška klasifikacije iznosi 57.22 %.

Na isti način računa se točnost, odnosno, pogreška klasifikacije za preostale dvije klase. Za klasu 0.5 (remi) točnost je  $PPV_{0.5}$ :

$$ACC_{0.5} = \frac{6281}{6281+497+582} = 0.853396, \quad (4.4)$$

odnosno, 85.34 %, čime pogreška klasifikacije iznosi 14.66 % te točnost klasifikacije  $PPV_1$  za klasu 1 (pobjeda bijelog) iznosi:

$$ACC_1 = \frac{893}{893+710+229} = 0.487445, \quad (4.5)$$

odnosno, 48.74 %, čime pogreška klasifikacije iznosi 51.26 %.

Analogno izračunu točnosti klasifikacije, za pojedinu se klasu izračunava i ukupna točnost klasifikatora izračunavanjem omjera zbroja svih točno klasificiranih uzoraka za sve klase naspram zbroja svih pogrešno klasificiranih uzoraka. Ukupna točnost klasifikacije ACC tada je:

$$ACC = \frac{391+6281+893}{391+76+447+497+6281+582+710+229+893} = 0.748565, \quad (4.6)$$

odnosno, izračunana točnost iznosi 74.86 % s pogreškom klasifikacije od 25.14 %.

### 4.3 Značaj varijabli po Gini kriteriju

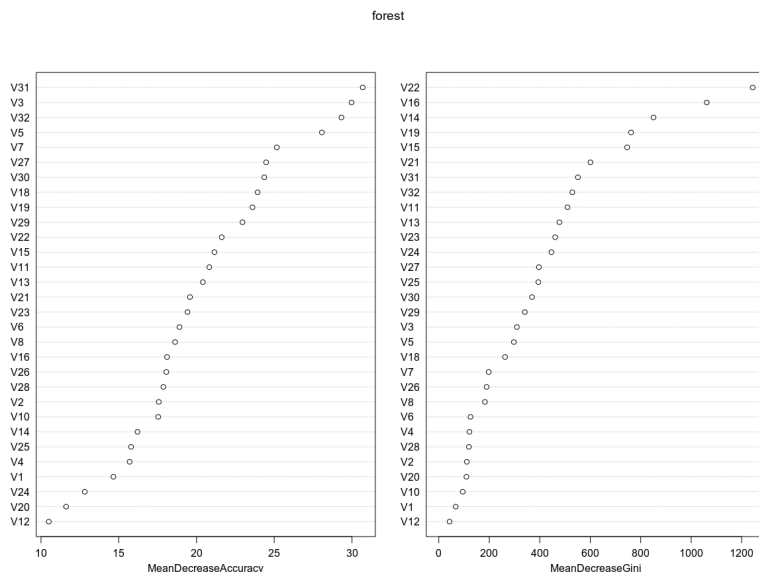
*MeanDecreaseGini* mjera je procjene važnosti atributa/varijable po Gini kriteriju, koji se uobičajeno upotrebljava za izračun grananja pri izgradnji klasifikacijskih stabala (James, 2013). *MeanDecreaseAccuracy* mjera je procjene važnosti atributa/varijable koja se određuje prilikom faze izračuna OOB (engl. *out-of-bag*) klasifikacijske pogreške. Pri uzorkovanju nisu svi podaci uključeni u konačni skup za učenje – oni koji nisu, zovu se *out-of-bag* podaci i upotrebljavaju se pri procjeni pogreške generalizacije generiranog stabla, pa na kraju i čitave šume. Važnost varijable pozitivno je korelirana i s padom točnosti RF klasifikatora zbog izdvajanja (ili permutacije) jedne varijable, stoga su varijable s velikom MDA (engl. *mean decrease accuracy*) vrijednošću važnije za klasifikaciju.

U radu je odabrano prvih 10+ najvažnijih varijabli prema svakom od kriterija s obzirom na opisani kriterij prekida, i tako je dobivena formula za RF model, što je u R-u zapisano kao:

```
mdaFormula <- factor(V33) ~ V31+V3+V32+V5+V7+V27+V30+V18+V19+V29
```

```
giniFormula <- factor(V33) ~ V22+V16+V14+V19+V15+V21+V31+V32+V11+V13+V23  
+V24
```

Slika 6. Klasifikacijski "variable importance" graf po promjeni u točnosti (lijevo) i promjeni po Gini kriteriju (desno)



Izvor: autori

#### 4. 4 Konačni evaluacijski model za smanjen broj varijabli po Gini kriteriju

Nakon određivanja varijabli koje imaju najveći utjecaj (značaj) na točnost klasifikacije, odnosno, najveću točnost predviđanja ishoda, izgradi se i sam model predviđanja – RF model – koji prilikom učenja gradi model klasifikacije samo za odabrane varijable, na sljedeći način:

```
formula <- factor(V33) ~ V3+V5+V29+V31+V11+V27+V19
```

```
forest_chop <- randomForest(formula, data=training_set)
```

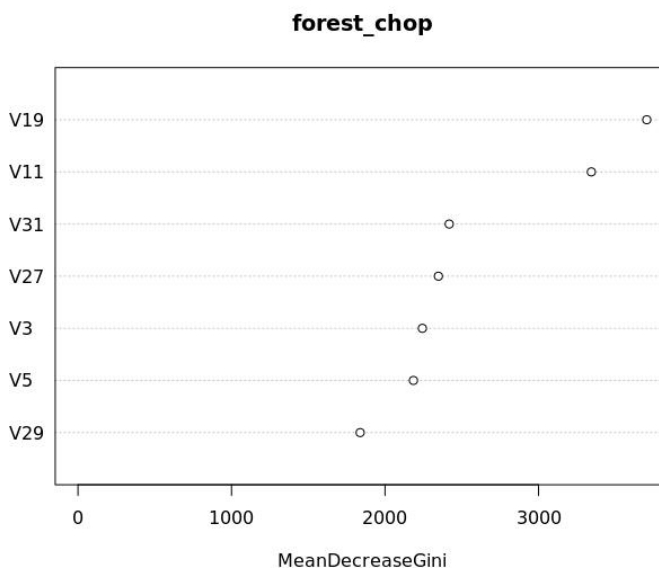
```
forest_chop
```

Varijable za konačni RF model odabrane su na temelju grafa sa Slike 6 jer se usporedivi rezultati klasifikacije nad testnim skupom podataka dobivaju i s odabirom preostalih dviju navedenih formula (mdaFormula i giniFormula) ovdje nisu prikazane. Tablica 2 prikazuje matricu konfuzije sa smanjenim brojem varijabli za učenje.

Tablica 2. Matrica konfuzije sa smanjenim brojem varijabli za učenje

stvarne klase – ishodi partija	predviđene klase – ishodi partija			pogreška klasifikacije
	-1	0.5	1	
-1	1788	1157	1847	0.626878
0.5	436	18602	720	0.058507
1	1514	1525	2727	0.527055

Slika 7. MeanDecreaseGini graf za konačni RF model- smanjen broj ulaznih varijabli na 7



Na temelju rezultata na Slici 7, u daljnji se postupak uvode najviše rangirane značajke prema Mean decrease gini indexu - V19, V11, V31, V27, V3, V5 i V29.

```
pred <- predict(forest_chop, newdata = test_set)
```

```
table(pred, test_set$V33)
```

Naposljetku, rezultati prikazani u Tablici 3 – konfuzijskoj matrici sa smanjenim brojem varijabli (7 – odabranih prema vrijednostima na slici 7) – za testni skup podataka pokazuju da je točnost klasifikacije ishoda novih šahovskih partija moguća s točnošću od 75.27 %.

Tablica 3. Matrica konfuzije sa smanjenim brojem varijabli (7) za testni skup

stvarne klase - ishodi partija	predviđene klase – ishodi partija				pogreška klasifikacije
		-1	0.5	1	
-1		557	171	495	0.544562
0.5		418	6175	552	0.135759
1		623	240	875	0.496547

#### 4.5 Usporedba rezultata predviđanja sa Shannonovom evaluacijom

Nakon izgradnje klasifikatora pozicija koji upotrebljava odabrane mrežne značajke, dakle one koje su se, empirijski gledano, pokazale najboljima po pitanju uspješnosti predviđanja, odnosno, evaluacije valjanosti pozicije, uspoređuje se točnost s klasičnom Shannonovom funkcijom. Rezultati točnosti predviđanja valjanosti svih pozicija iz skupa za testiranje moraju se dobiti i za Shannonovu evaluacijsku funkciju. To je postignuto s jednostavnom Python skriptom koja izračunava vrijednost Shannonove funkcije za sve pozicije iz testnog skupa pozicija te ih uspoređuje s već izračunanim evaluacijskim vrijednostima iz rječnika vrijednosti koji je izgrađen na temelju fiksne evaluacije Stockfish šahovskog programa svih pozicija iz testnog skupa partija. Od 145 846 analiziranih pozicija dobiveno je 77 143 točnih, čime točnost Shannonove evaluacijske funkcije iznosi 52.89%.

Rezultat analize točnosti evaluacije svodi se na jednostavnu usporedbu točnosti naučenog modela RF sa Shannonovom evaluacijskom funkcijom za testni skup podataka prikazanima u Tablici 4.

Tablica 4. Usporedba točnosti predviđanja naučenog modela RF sa Shannonovom evaluacijskom funkcijom za testni skup podataka

Točnost Shannonove evaluacijske funkcije	52.89 %
Točnost RF modela	75.27 %
<b>Poboljšanje</b>	<b>22.38 %</b>

Na osnovi analize odabrane statičke pozicije na ploči razvidno je da Shannonova evaluacijska funkcija ima značajno lošiju točnost predviđanja ishoda trenutne partije, s obzirom na rezultate RF klasifikatora koji upotrebljava odabrane strukturne značajke mreže.

## 5. ZAKLJUČAK

U ovom je radu predstavljena mogućnost modeliranja šahovskih pozicija u šahovskim partijama nizom kompleksnih mreža, odnosno, izračunom mrežnih značajki. Kombiniranjem tih značajki i izgradnjom klasifikatora koji upotrebljava model šume slučajnih stabala postignuta je veća točnost predviđanja. Točnost klasifikatora nije samo dostatna za računalni šah, već i značajno nadmašuje točnost klasične Shannonove evaluacijske funkcije za čak 22.38%. Dobiveni rezultat nije iznenađujuć za šahovske igrače jer je poznato da suvremeni šahovski programi, koji su već odavno nadmašili najbolje ljudske igrače današnjice, imaju evaluacijske funkcije koje su nemjerljivo kompleksnije od



jednostavnog predviđanja kvalitete pozicije koju odražava Shannonova evaluacijska funkcija. To, naravno, i nije čudno, s obzirom na to da je upravo Shannon udario čvrste temelje računalnom šahu sredinom prošlog stoljeća te se u praksi pokazalo da je njegov pristup računalnom šahu, utemeljen na Shannon A tipu računalnih šahovskih programa, još uvijek superioran u odnosu na programe koji nastoje oponašati ljudski način pristupa igri. Međutim, primjena takvog modela predviđanja, koji ne upotrebljava znanje o šahu niti pravila same igre, a daje obećavajuće rezultate, odaje dojam da će se u bližoj budućnosti pojaviti šahovski programi koji će nadmašiti postojeće ekstremno optimizirane klasične evaluacijske funkcije u modernim šahovskim programima. Neki od projekata koji danas upošljavaju duboke konvolutivne neuralne mreže, poput Google AlphaZero (Silver, 2017; Alphazero, 2018) ili istraživanja opisanog u (Lei, 2015), te projekta zajednice entuzijasta otvorenih tehnologija, Leela Chess Zero (Silver, 2018), daju naslutiti da će se taj odnos uskoro promijeniti u korist potonjeg pristupa. Daljnja bi istraživanja kompleksnih svojstava i odnosa između evolucija šahovskih mrežnih modela i šahovskih strategija zbog toga mogla svakako biti vrlo plodonosna, s konkretnim praktičnim rezultatima. Nije isključena ni mogućnost primjene takvog pristupa pri daljnjem poboljšanju postojećih modernih šahovskih programa u vidu svojevrsnog hibridnog pristupa evaluaciji šahovskih pozicija i pronalaska optimalne strategije igre. Potrebno je naglasiti da glavna kritika svih dubokih neuralnih modela, unatoč postignutim visokim performansama na različitim područjima primjene, proizlazi iz njihove nerazumljivosti. Naime, neuronske mreže ponašaju se poput modela crne kutije, koji ne omogućava obrazloženje ponašanja, tj. objašnjenja dobivene evaluacijske vrijednosti u ovome slučaju, te stoga nisu primijenjene u ovome radu. Hibridni pristup pokušava razriješiti problem nerazumljivosti modela te će se u budućem radu posvetiti proučavanju hibridnog pristupa.

## LITERATURA

- AlphaZero. (2018) En.wikipedia.org, 2018. [Na internetu]. Dostupno: <https://en.wikipedia.org/wiki/AlphaZero>. [Pristupano: 20. 4. 2018.].
- Atkin R. H., Witten, I. H. (1975) A Multi-dimensional Approach to Positional Chess, *Int. J. Man-Machine Studies*, 7, 727-750. [https://doi.org/10.1016/S0020-7373\(75\)80035-6](https://doi.org/10.1016/S0020-7373(75)80035-6)
- Beliga, S., Meštović, A., Martinčić-Ipšić, S. (2015). An overview of graph-based keyword extraction methods and approaches. *Journal of information and organizational sciences*. 39(1), 1-20.
- Beliga, S., Meštović, A., Martinčić-Ipšić, S. (2016) Selectivity-based keyword extraction method. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 12(3), 1-26. <https://doi.org/10.4018/IJSWIS.2016070101>
- Breiman, L. (2001) Random Forests, *Machine Learning*, 45(1), 5-32. <https://doi.org/10.1023/A:1010933404324>
- Chessprogramming - Evaluation, (2018) Chessprogramming.wikispaces.com, 2018. [Na internetu]. Dostupno: <https://chessprogramming.wikispaces.com/Evaluation>. [Pristupano: 11.03.2018].
- Elo rating system. (2018) En.wikipedia.org, 2018. [Na internetu]. Dostupno: [https://en.wikipedia.org/wiki/Elo\\_rating\\_system](https://en.wikipedia.org/wiki/Elo_rating_system). [Pristupano: 17. 4. 2018.].
- Farren, D. Templeton D. and Wang, M. (2013) Analysis of networks in chess, Stanford University, Tech. Rep. <http://snap.stanford.edu/class/cs224w-2013/projects2013/cs224w-023-final.pdf> [Pristupano: 15. 4. 2018.].
- Fernández A., Salmerón A. (2008) BayesChess: A computer chess program based on Bayesian networks, *Pattern Recognition Letters*, 29(8), 1154-1159. <https://doi.org/10.1016/j.patrec.2007.06.013>
- Home - Stockfish 10 - Open Source Chess Engine, (2018) Stockfishchess.org. [Na internetu]. Dostupno: <https://stockfishchess.org/files/stockfish-10-linux.zip> [Pristupano: 10. 4. 2018].

- James, G., Witten, D., Hastie, T., Tibshirani, R. (2013) An introduction to statistical learning. (Vol. 112). New York: Springer. <https://doi.org/10.1007/978-1-4614-7138-7>
- Jokić, J. (2014) Construction of a Chess Engine, Bachelor Thesis, Supervisor: Martinčić-Ipšić, S., University of Rijeka, Department of Informatics.
- Jokić, J. (2018) Data Mining for Chess, Master Thesis, Supervisor: Martinčić-Ipšić, S., University of Rijeka, Department of Informatics.
- Komodo chess engine by Don Dailey, Larry Kaufman and Mark Leffler, (2018), Komodochess.com, [Na internetu]. Dostupno: <https://komodochess.com/Komodo12.htm>. [Pristupano: 20. 4. 2018.].
- Lai, M., (2015) Giraffe: Using Deep Reinforcement Learning to Play Chess, MSc thesis, Imperial College London.
- Leela Chess Zero, (2018) [Na internetu]. Dostupno: [https://en.wikipedia.org/wiki/Leela\\_Chess\\_Zero](https://en.wikipedia.org/wiki/Leela_Chess_Zero) <https://komodochess.com/>. [Pristupano: 20. 4. 2018.].
- Liaw, D., Wiener M. (2018) randomForest v4.6-14, CRAN. [Dostupno: <https://cran.r-project.org/web/packages/randomForest/index.html>
- Martinčić-Ipšić, S., Margan, D., Meštrović, A. (2016) Multilayer network of language: A unified framework for structural analysis of linguistic subsystems. *Physica A: Statistical Mechanics and its Applications* 457, 117-128. <https://doi.org/10.1016/j.physa.2016.03.082>
- Martinčić-Ipšić, S., Miličić, T., Todorovski, Lj. (2019) The Influence of Feature Representation of Text on the Performance of Document Classification. *Applied Sciences*, 9, 4. 743. <https://doi.org/10.3390/app9040743>
- Martinčić-Ipšić, S., Močibob, E., Perc, M. (2017) Link prediction on Twitter. *PloS one* 12.7 (2017): e0181079. <https://doi.org/10.1371/journal.pone.0181079>
- Meštrović, A., Grubiša, Z. (2015) Preliminarna analiza mreža koautorstva sveučilišta u Rijeci. *Zbornik Veleučilišta u Rijeci* 3.1: 159-178.
- Newman, M.. (2018) *Networks*. Oxford university press. <https://doi.org/10.1093/oso/9780198805090.001.0001>
- Shannon C.E., (1988) *Programming a Computer for Playing Chess*. Dostupno u: Levy D. (eds) *Computer Chess Compendium*. Springer, New York, NY, 1988. [https://doi.org/10.1007/978-1-4757-1968-0\\_1](https://doi.org/10.1007/978-1-4757-1968-0_1)
- Silver, A. (2018). *Leela Chess Zero: AlphaZero for the PC*. Chess News.
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T. et al. (2017) Mastering chess and Shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.
- Stern, D., Herbrich, R., Graepel, T. (2006) Bayesian pattern ranking for move prediction in the game of Go. *Proceedings of the 23rd International conference on Machine learning*. ACM, 2006. <https://doi.org/10.1145/1143844.1143954>
- Stuart J., Peter Norvig, and John Canny, (2015) *AI: A Modern Approach*, Russell, 3d Edition.
- Zermelo, E. (1913) *Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels*, Proc. Fifth Congress Mathematicians, (Cambridge 1912), Cambridge University Press 1913, 501-504.
- Zobrist, A. L. (1969) *A New Hashing Method with Application for Game Playing*, Tech. Rep. 88, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 1969.
- Zobristovo heširanje, [Sh.wikipedia.org](https://sh.wikipedia.org/wiki/Zobristovo_he%C5%A1iranje), (2018). [Na internetu]. Dostupno: [https://sh.wikipedia.org/wiki/Zobristovo\\_he%C5%A1iranje](https://sh.wikipedia.org/wiki/Zobristovo_he%C5%A1iranje). [Pristupano: 10. 2. 2018.].

## **Privitak A: Popis značajki po vrstama mreža**

### **Potporna mreža (V1 – V8):**

- V1 - najmanji stupanj čvora u mreži
- V2 - najveći stupanj čvora u mreži
- V3 - prosječni stupanj čvora
- V4 - medijan distribucije stupnjeva čvorova
- V5 - gustoća mreže/grafa
- V6 - broj čvorova
- V7 - broj veza
- V8 - broj jako povezanih komponenti.

### **Mreža mobilnosti (V9 – V16):**

- V9 - najmanji stupanj čvora u mreži
- V10 - najveći stupanj čvora u mreži
- V11 - prosječni stupanj čvora
- V12 - medijan distribucije stupnjeva čvorova
- V13 - gustoća mreže/grafa
- V14 - broj čvorova
- V15 - broj veza
- V16 - broj jako povezanih komponenti.

### **Pozicijska mreža (V17 - V24):**

- V17 - najmanji stupanj čvora u mreži
- V18 - najveći stupanj čvora u mreži
- V19 - prosječni stupanj čvora
- V20 - medijan distribucije stupnjeva čvorova
- V21 - gustoća mreže/grafa
- V22 - broj čvorova
- V23 - broj veza
- V24 - broj jako povezanih komponenti.

**Mreža praćenja (V25 - V32):**

- V25 - najmanji stupanj čvora u mreži
- V26 - najveći stupanj čvora u mreži
- V27 - prosječni stupanj čvora
- V28 - medijan distribucije stupnjeva čvorova
- V29 - gustoća mreže/grafa
- V30 - broj čvorova
- V31 - broj veza
- V32 - broj jako povezanih komponenti
- V33 - klasa.

# PREDICTING THE OUTCOME OF THE CHESS GAME REPRESENTED AS A COMPLEX NETWORK WITH RANDOM FOREST CLASSIFIER

**Jovan Jokić**

MA in Informatics, Student, Department of Informatics, University of Rijeka, Radmile Matejčić 2,  
51000 Rijeka, Croatia; e-mail: jovan.jokic@xnet.hr

**Sanda Martinčić-Ipšić**

PhD, Associate Professor, Department of Informatics, University of Rijeka, Radmile Matejčić 2,  
51000 Rijeka, Croatia; e-mail: smarti@uniri.hr

## ABSTRACT

*This work addresses the problem of constructing a static chess position evaluation model which utilizes only information extracted from complex networks' features of positions of chess pieces on the board. The mutual relations of chess figures, the complex relations of figures and the positions on the chess board, as well as the information on which fields they are attacked, describing tactical and strategic elements of the chess game, all these are modeled by a complex network formalism. The goal of this work is to demonstrate that it is possible to train a classifier which would have better prediction results of the game outcome, utilizing only selected measurements of the complex network's features rather than the corresponding Shannon's evaluation function. Shannon's evaluation function quantifies the material state of both players on the board, the mobility of the figures, king safety and pawn structure quality. The input to the classification model consists of feature vectors defined by four types of networks (support, mobility, position, tracking) constructed from static game positions. The game information is obtained from the available chess games database in a Portable Game Notation format. Features vector contains different complex network's measures that quantify the structural properties of the network. The machine-learning algorithm of random forest is used for the training of the classification model and for selecting the most important features which have the highest impact on the prediction results. Experimentally, with the custom Python script and employing the chess engine Stockfish for analysis, a baseline fixed-depth evaluation of the static positions in chess games are extracted for determining the target classification classes – labels - win for white, win for black or draw. Next, the selected network measurements and the corresponding class labels form the input features vectors. The features vectors are used to train the Random Forest classifier: the results of*

*the trained classification model (75% correctly predicted outcomes are compared to the baseline obtained by the standard Shannon's evaluation function (52% correctly classified outcomes)). PGN database (Portable Game Notation) reading and parsing of games from the games database, the representation of static positions from the games, and the generation of networks/graphs are done with the suitable Python tools, namely the chess library chess-py and NetworkX Python module.*

**Keywords:** *evaluation function in chess, complex networks, network features, random forest, predicting outcome of chess game.*