ELEKTROTEHNIŠKI VESTNIK 87(4): 156–164, 2020
IZVIRNI ZNANSTVENI PRISPEVEK

Uporaba vektorskih vložitev pri modeliranju poteka košarkarske tekme

Petar Vračar

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana, Slovenija
E-pošta: petar.vracar@fri.uni-lj.si

Povzetek. Pri modeliranju razvoja košarkarske tekme potrebujemo značilke za opis nasprotnikov. Običajno v ta namen uporabimo kazalce zmogljivosti ekip, ki jih definirajo domenski eksperti. V članku predstavimo uporabo vektorskih vložitev za avtomatsko generiranje značilk v latentnem prostoru. V eksperimentalnem delu primerjamo modele, naučene na ekspertnih in avtomatsko zgrajenih značilkah. Rezultati kažejo, da modeli, naučeni v latentnem atributnem prostoru generirajo verodostojne simulacije. Po drugi strani ti modeli niso dobri napovedovalci zmagovalca tekme, kar sugerira, da so latentni atributi samo delno zajeli zmogljivosti ekip.

Ključne besede: modeliranje športa, globoke nevronske mreže, vektorska vložitev, markovski proces, simuliranje tekem

Applying vector embeddings for basketball modeling

Modeling progression of a basketball game requires features to describe abilities of opposing teams. The team skills are usually specified using expert-defined statistics. In the paper we present a method for an automatic construction of features embedded in a latent space. In our experimental evaluation we compare models trained with expert-defined and with automatically constructed features. The results show that the models trained with latent features generate plausible game simulations. These models, on the other hand, are not good at predicting the winner, which suggests that the latent features only partially capture the team capabilities.

Keywords: sports modelling, deep neural network, vector embedding, Markov process, match simulation

1 UVOD

Športna analitika je področje statistične obdelave podatkov, ki z analizo doseženih rezultatov in s projekcijo bodoče uspešnosti poskuša nosilcem odločanja ponuditi kompetitivno prednost v svetu športa. Metode podatkovnega rudarjenja in strojnega učenja se uporabljajo na različnih področjih športne analitike - od napovedovanja izida tekem ([1], [2], [3]), preko kvantitativnega ocenjevanja zmogljivosti ekip [4], modeliranja taktičnih odločitev [5], do preučevanja zakonitosti in prvin športne igre ([6], [7], [8], [9]). Pregled metod podatkovnega rudarjenja v športu najdemo v [10].

V članku se ukvarjamo s praktičnim problemom, kako iz kronoloških zapisov o poteku košarkarskih tekem (t. i. podatki play-by-play) izdelati statistični model za generiranje verodostojnih simulacij tekem med izbranima ekipama. Verodostojen in natančen simulator športnih tekem je lahko sestavni del ekspertnega sistema, namenjenega trenerjem in strokovnemu osebju v klubih, ki bi ga uporabljali kot orodje za analizo, usmerjeno k izboljševanju kakovosti igre in doseženih rezultatov.

Generirane simulacije morajo biti predvsem skladne s pravili košarke. Zajemati morajo tudi značilne vzorce poteka dogodkov, ki sledijo iz ustaljenih taktičnih prijemov in specifike trenutne situacije na tekmi (na primer, ekipa v rezultatskem zaostanku dvigne tempo igre, postane bolj agresivna in prevzema več tveganja). Ne nazadnje, generirane simulacije morajo odražati tudi značilnosti ekip, ki igrajo simulirane tekme. Slednja lastnost implicira potrebo po značilkah za opis zmogljivosti košarkarskih ekip. Preproste značilke so podane v obliki osnovnih opisnih statistik (ang. box score), ki temeljijo na štetju ključnih dogodkov na odigranih tekmah. Z bogatenjem zapisov o poteku tekem so se razvile tudi bolj sofisticirane značilke učinkovitosti igre, ki so praviloma ročno določene na podlagi ekspertnega znanja o športu ([11], [12], [4])). Z raziskovalnega stališča je zanimiva možnost avtomatskega grajenja atributnega prostora za opis značilnosti ekip, ki ne potrebuje ekspertnega predznanja. V nasprotju z metodo [13], ki sestavi atributni prostor v eksplicitni obliki, v članku predstavimo metodo za tvorbo latentnega opisa zmogljivosti ekip na podlagi vektorskih vložitev (angl. embedding).

Vektorska vložitev je preslikava, ki diskretnim entitetam priredi numerično predstavitev v zveznem (nizkodimenzionalnem) prostoru. V strojnem učenju se vektorske vložitve uporabljajo za iskanje ustrezne predstavitve podatkov za dani učni problem. Nedavni veliki preboji na področju obdelave naravnega jezika temeljijo na uporabi vektorske vložitve besed [14], ki zajame semantične relacije med besedami in jih implicitno zakodira v

Prejet 9. marec, 2020
Odobren 30. junij, 2020

njihovo geometrijsko upodobitev v vektorskem prostoru. V zadnjih letih se podobni pristopi uporabljajo na področjih, ki zahtevajo obdelavo strukturiranih podatkov, kot so grafi [15] ali proteini [16].

2 PREGLED PODROČJA

Stern [17] je uporabil 493 tekem ameriške profesionalne košarkarske lige (NBA) ter na podlagi trenutnih izidov ob koncu posameznih četrtin zgradil model Brownovega gibanja razvoja rezultata. Model se je izkazal kot dober napovedovalec verjetnosti zmage na osnovi podane trenutne razlike v rezultatu. Model ni upošteval moči ekip niti ni modeliral posesti žoge.

Goldman in Rao [6] sta uporabila podoben model za proučevanje učinka psihološkega pritiska na kakovost izvedbe. Modelirali so tudi stabilnost napovedane verjetnosti zmage v odvisnosti od majhnih sprememb trenutnega rezultata in jo uporabili za ocenjevanje pomembnosti situacije za končni razplet tekme.

Gabel in Redner [18] sta uporabila model naključnega sprehoda za opisovanje različnih statističnih lastnosti gibanja rezultata med potekom košarkarske tekme. Pokazala sta, da je igralni čas med zadetki eksponentno porazdeljen in je proces doseganja točk brez spomina. Ugotovila sta tudi, da imajo lastne vrline ekip majhen vpliv na splošno sliko gibanja rezultata.

Merrit in Clauset [7] sta modelirala dinamiko gibanja rezultata na povprečni tekmi v različnih timskih športih (košarka, ameriški nogomet in hokej). Uporabila sta dva stohastična procesa. Prvi proces proizvaja zadetke, medtem ko drugi proces določa, katera ekipa je dosegla točke. Ugotovila sta tudi, da je model sposoben zelo natančno predvideti končni izid tekme samo na podlagi opažene dinamike nekaj prvih zadetkov.

Shirley [19] je modeliral razvoj košarkarske tekme z diskretno markovsko verigo. Stanje je zakodiral v obliki trirazsežnega vektorja. Prva komponenta določa ekipo, ki ima žogo. Druga komponenta določa način, kako je ekipa prišla do žoge. Tretja komponenta predstavlja število točk, doseženih ob prihodu v to stanje. Shirley je za vsako ekipo zgradil svoj model na podlagi njenih opisnih statistik in pokazal, da je tak model dober napovedovalec verjetnosti zmage opazovane ekipe proti povprečnemu nasprotniku. Shirley je opisal tudi možnost učenja modela iz opisov poteka tekem (in ne na podlagi opisnih statistik) ob upoštevanju moči posameznih ekip, toda zaradi pomanjkanja podatkov tega ni izvedel.

Štrumbelj in Vračar [20] sta izhajala iz Shirleyevega modela, le da sta implicitno razširila prostor stanj z atributi za opis moči ekip. Na ta način sta matriko prehoda med stanji Shirleyjevega modela izrazila v odvisnosti od karateristik ekip in tako pridobila splošen model za simuliranje tekem med poljubnima nasprotnikoma. Za opisovanje moči ekip sta uporabila zbirne statistike na podlagi štirih faktorjev [11]. Verjetnosti prehodov med stanji sta modelirala z multinomialno logistično regresijo, pri čemer sta vsako vrstico tranzicijske matrike obravnavala kot ločen klasifikacijski problem (stanja so neodvisna zaradi markovske lastnosti procesa). Ker stanja modela ne vsebujejo informacije o poteku igralnega časa, je simulacije konec po vnaprej določenem številu prehodov.

Vračar in sod. [21] so prevzeli idejo o parametrizaciji verjetnosti prehodov med stanji z značilkami za opis zmogljivosti ekip in jo nadgradili z dodatno razširitvijo opisa prostora stanj s parametri, ki podajajo kontekst trenutka v razvoju tekme. Modelirali so tudi potek igralnega časa med napovedanimi dogodki. S temi nadgraditvami so bolje zajeli dinamiko razvoja igre, kar so tudi empirično pokazali.

Isti avtorji so razvili postopek za avtomatsko generiranje atributnega prostora [13]. Osnovna ideja temelji na identifikaciji množic podobnih dogodkov, pri čemer se podobnost nanaša na neposredno soseščino (prejšnji in naslednji dogodek) v zaporedju dogodkov na košarkarskih tekmah. Atribute so definirali kot verjetnosti prehoda med identificiranimi množicami podobnih dogodkov. Eksperimentalno so pokazali, da so modeli, dobljeni z učenjem na podlagi avtomatsko generiranih atributov, po kakovosti napovedovanja primerljivi z modeli, naučenimi z ekspertnimi atributi.

Alcorn [22] je uporabil tehniko vektorskih vložitev za latentno predstavitev igralcev poklicne bejzbolske lige MLB. Dobljeno prezentacijo je uporabil za razvrščanje igralcev in modeliranje izida naslednjega meta.

3 METODOLOGIJA

3.1 Modeliranje poteka košarkarske tekme

Na razvoj športne tekme lahko gledamo kot na realizacijo stohastičnega procesa, ki se v diskretnih korakih sprehaja v (zveznem) prostoru stanj $\mathcal{S}$. Modeliranje je računsko manj zahtevno, če predpostavimo, da je proces brez spomina, tj. da je prihodnje stanje procesa odvisno samo od sedanjega stanja. Po drugi strani je jasno, da je realnost veliko bolj kompleksna in da dosedanjega poteka tekme ne moremo popolnoma zanemariti pri napovedovanju nadaljnjih dogodkov. Predpostavko o markovski lastnosti procesa zato nekoliko omilimo s tem, da v opis stanja zakodiramo tudi povzetek zgodovine, ki je relevanten za nadaljevanje procesa.

V splošnem je možno prostor stanj $\mathcal{S}$ zakodirati v obliki vektorja, sestavljenega iz spremenljivk treh tipov, to so (1) slučajne spremenljivke, ki neposredno vplivajo na razvoj dogodkov na tekmi, (2) spremenljivke, ki vplivajo na razvoj dogodkov na tekmi, vendar niso slučajne (so posledica prejšnjih dogodkov na tekmi), in (3) parametri, ki vplivajo na potek tekme, vendar se njihova vrednost med tekmo ne spreminja. Pri modeliranju poteka košarkarske tekme smo prostor stanj $\mathcal{S}$ zakodirali v obliki vektorja:

$$\langle Evt, Dur, Qtr, Time, PtsDiff, \mathbf{a}, \mathbf{h} \rangle.$$

Slučajna spremenljivka Evt predstavlja naslednji dogodek, ki se bo zgodil na tekmi. Množica mogočih vrednosti je prikazana v tabeli 1. Slučajna spremenljivka Dur predstavlja igralni čas (v sekundah), ki bo pretekel do naslednjega dogodka. Komponente Qtr, $Time$ in $PtsDiff$ niso slučajne spremenljivke, temveč povzemajo že odigrani del tekme. Časovni vidik povzemata komponenti Qtr (trenutna četrtina) in $Time$ (preostali čas v sekundah do izteka trenutne četrtine). Najbolj relevanten povzetek dosedanjega poteka tekme – trenutni rezultat (podan kot razlika v koših z vidika domačega moštva) – je predstavljen s komponento $PtsDiff$. Motivacija za vključitev komponent Qtr, $Time$ in $PtsDiff$ v opis stanja so empirične študije ([20], [18], [7]), ki potrjujejo ekspertno znanje in intuicijo, da se dinamika igre spreminja v odvisnosti od trenutnega konteksta, ki ga določata predvsem čas do konca tekme in trenutni izid (na primer, ekipa v rezultatskem zaostanku igra bistveno drugače v končnici kot na polovici tekme). V prejšnji raziskavi smo pokazali tudi, da neupoštevanje trenutnega konteksta povzroča generiranje nerealističnih simulacij [21]. Končno, vektorja $\mathbf{a}$ in $\mathbf{h}$ predstavljata opis zmogljivosti gostujoče (away) oziroma domače (home) ekipe. Vključitev vektorjev $\mathbf{a}$ in $\mathbf{h}$ v opis stanja je samoumevna, saj želimo, da se v generiranih simulacijah tekem odražajo karakteristike izbranih nasprotnikov. Zmogljivosti ekip lahko izrazimo v prostoru ekspertnih atributov (če je ta na voljo) oziroma konstruiramo ustrezni atributni prostor v ekplicitni (na primer z uporabo metode iz [13]) ali implicitni obliki (glej razdelek 3.3). Zaradi enostavnosti predpostavimo, da se zmogljivosti ekip $\mathbf{a}$ in $\mathbf{h}$ ne spreminjajo med trajanjem posamezne tekme (čeprav se lahko spreminjajo med sezono).

Simulacijo košarkarske tekme generiramo s slučajnim sprehodom $(\mathbf{x_0}, \mathbf{x_1}, \ldots, \mathbf{x_f})$, kjer stanji $\mathbf{x_0}$ in $\mathbf{x_f}$ ustrezata začetku in koncu tekme. Naj bo $f(\mathbf{y}|\mathbf{x})$ verjetnostna porazdelitev naslednjega stanja $\mathbf{y}$ v odvisnosti od trenutnega stanja $\mathbf{x}$. V opisu vsakega stanja sta samo dve slučajni spremenljivki, Evt in Dur, zato lahko z uporabo verižnega pravila pogojno porazdelitev $f(\mathbf{y}|\mathbf{x})$ poenostavimo in izrazimo v obliki:

$$f(\mathbf{y}|\mathbf{x}) = f_{\mathbf{Evt}}(y^{(Evt)}|\mathbf{x}) f_{\mathbf{Dur}}(y^{(Dur)}|\mathbf{x}, y^{(Evt)}) \quad (1)$$

Pogojno porazdelitev iz enačbe (1) ocenimo s pomočjo dveh modelov: model M_{Evt} ocenjuje marginalno pogojno porazdelitev naslednjega dogodka ob podanem opisu trenutnega stanja, model M_{Dur} pa ocenjuje čas med dvema dogodkoma glede na opis trenutnega stanja in naslednji dogodek. Vzorčenje iz porazdelitve $f(\mathbf{y}|\mathbf{x})$ izvedemo s postopnim nastavljanjem vrednosti posameznim komponentam iz opisa naslednjega stanja $\mathbf{y}$. Najprej izberemo vrednost komponente Evt z vzorčenjem napovedi modela M_{Evt}, ki na vhodu prejme opis trenutnega stanja $\mathbf{x}$. Nato izberemo vrednost komponente Dur z vzorčenjem napovedi modela M_{Dur}, ki na vhodu poleg opisa trenutnega stanja $\mathbf{x}$ prejme tudi prej izbrano vrednost Evt. Zdaj nastavimo vrednosti komponentam Qtr, $Time$ in $PtsDiff$, saj lahko te rekonstruiramo neposredno na podlagi komponent Evt in Dur. Ker se, po predpostavki, vrednosti komponent $\mathbf{a}$ in $\mathbf{h}$ med tekmo ne spreminjajo, je s tem določen celoten opis naslednjega stanja $\mathbf{y}$.

Tabela 1: Množica dogodkov za košarko.

Oznaka dogodka	Opis
AJB, HJB	Gostujoča / Domača ekipa je dobila sodniški met.
AINB, HINB	Gostujoča / Domača ekipa je vrnila žogo v igro.
AORB, HORB	Gostujoča / Domača ekipa je dobila skok v napadu.
AOREB, HOREB	Gostujoča / Domača ekipa je dobila ekipni skok v napadu.
ADRB, HDRB	Gostujoča / Domača ekipa je dobila skok v obrambi.
ADREB, HDREB	Gostujoča / Domača ekipa je dobila ekipni skok v obrambi.
A2PA, H2PA	Gostujoča / Domača ekipa je zgrešila met za 2 točki.
A2PM, H2PM	Gostujoča / Domača ekipa je zadela met za 2 točki.
A3PA, H3PA	Gostujoča / Domača ekipa je zgrešila met za 3 točke.
A3PM, H3PM	Gostujoča / Domača ekipa je zadela met za 3 točke.
AFTiA, HFTiA	Gostujoča / Domača ekipa je zgrešila prvega izmed i preostalih prostih metov ($i \in [1 \ldots 3]$).
AFTiM, HFTiM	Gostujoča / Domača ekipa je zadela prvega izmed i preostalih prostih metov ($i \in [1 \ldots 3]$).
ATO, HTO	Gostujoča / Domača ekipa je zapravila žogo.
AV, HV	Gostujoča / Domača ekipa je storila prekršek.
AF, HF	Gostujoča / Domača ekipa je storila osebno napako.

3.2 Modeliranje z eksplicitnim opisom zmogljivosti ekip

Modela M_{Evt} in M_{Dur} sta medseboj neodvisna, zato ju lahko učimo vsakega posebej. Pred učenjem je treba izbrati atributni prostor za predstavitev zmogljivosti ekip. V tem razdelku je opisan postopek učenja modelov v primeru, ko so zmogljivosti ekip podane v eksplicitni obliki (na primer z ekspertnimi statistikami na podlagi štirih faktorjev).

Napovedovanje naslednjega dogodka (model M_{Evt}) obravnavamo kot klasifikacijski problem. Učno množico za gradnjo modela M_{Evt} sestavimo tako, da vhodne podatke play-by-play (kronološko zaporedje ključnih dogodkov na dejanskih tekmah) prevedemo v zaporedje prehodov v izbranem prostoru stanj. En učni primer predstavlja tranzicijo med zaporednima dogodkoma. Vsaki komponenti vektorja za kodiranje stanj ustreza en atribut (stolpec) v učni množici, razen komponentam za opis zmogljivosti ekip, ki sta sama vektorja in se raztezata čez več stolpcev (vsak element teh vektorjev predstavlja svoj atribut oziroma konkretno statistiko). Izsek vhodnih podatkov play-by-play je prikazan v tabeli

2. Učna množica, sestavljena na podlagi tega izseka, je prikazana v tabeli 3.

Opis zmogljivosti ekip določimo za vsako tekmo posebej. Vrednosti atributov za tekočo tekmo izračunamo na podlagi statistik, ki jih je opazovana ekipa dosegla na že odigranih tekmah (na primer uporabimo povprečne vrednosti statistik). Pri izračunu atributov za opisovanje domačega moštva upoštevamo samo tekme, ki jih je ta ekipa odigrala doma (ne glede na nasprotnika). Prav tako pri izračunu atributov za gostujočo ekipo upoštevamo samo tekme, ki so jih odigrali v gosteh (ne glede na nasprotnika). Z ločeno obravnavo domačih in gostujočih tekem implicitno upoštevamo prednost domačega igrišča.

Odločitveno drevo z multinomialno logistično regresijo v listih se je izkazalo kot ustrezna izbira za model M_{Evt} [21]. Hierarhična struktura odločitvenega drevesa uspešno povzame diskretna pravila igre in razdeli prostor stanj na disjunktne podmnožice, ki predstavljajo (v logičnem smislu) dovoljena nadaljevanja poteka dogodkov. Tako se izognemo neveljavnim prehodom v prostoru stanj (tistim, ki so v nasprotju s pravili igre). Dodatna prednost drevesnih modelov je njihova razumljivost za domenskega eksperta.

Napovedovanje časa med dvema dogodkoma (model M_{Dur}) je v osnovi regresijski problem, saj ciljna spremenljivka lahko zavzame poljubno zvezno vrednost med 0 in 24 (omejitev dolžine napada, izražena v sekundah). Po drugi strani pa lahko čas med dvema dogodkoma obravnavamo tudi kot ordinalno diskretno spremenljivko zaradi vsesplošne sekundne granulacije časa v podatkih play-by-play.

Učna množica za model M_{Dur} je po strukturi skoraj identična tisti za M_{Evt} (glej tabelo 3) – razlika je dodatni stolpec Dur, ki označuje čas prehoda iz trenutnega dogodka (atribut $PrevEvt$) v napovedan dogodek (atribut Evt) in predstavlja ciljno spremenljivko.

Regresijsko drevo se je izkazalo kot ustrezen model za M_{Dur} [21]. Hierarhična struktura drevesa razdeli problemski prostor glede na par dogodkov: tistega, ki se je nazadnje zgodil, in tistega, ki je napovedan, da se bo naslednji zgodil. Posamezne napovedi o pretoku igralnega časa med omenjenima dogodkoma pa dobimo z vzorčenjem iz empirične porazdelitve primerov v listih drevesa.

Prednost uporabe eksplicitnega atributnega prostora za opis zmogljivosti ekip je v tem, da modela M_{Evt} in M_{Dur}, potem ko sta naučena, ni treba naknadno posodabljati (na primer ob večjih spremembah v zmogljivosti nekaterih ekip, kot so poškodbe ključnih igralcev ali prihod okrepitev). Vse morebitne spremembe v zmogljivosti ekip bodo zajete v vrednostih atributov. Modela M_{Evt} in M_{Dur} bosta v svojih napovedih upoštevala novo situacijo brez potrebe po dodatnem učenju.

3.3 Modeliranje brez opisa zmogljivosti ekip

V tem razdelku je predstavljen način učenja modelov M_{Evt} in M_{Dur} brez eksplicitnega opisa zmogljivosti ekip. Osnovna ideja temelji na uporabi tehnike vektorske vložitve, ki med učenjem ciljne funkcije hkrati išče bolj ustrezno reprezentacijo vhodnih podatkov.

Na sliki 1 je prikazana arhitektura nevronske mreže za učenje modela M_{Evt}. Na vhodu sta domača (sloj I_3) in gostujoča ekipa (sloj I_4) predstavljeni z uporabo eničnega kodiranja (angl. one-hot encoding) – vsaka ekipa je predstavljena z binarnim vektorjem, katerega edini neničelni element je na poziciji, ki enolično določa opazovano ekipo. Na podoben način je na vhodu predstavljen tudi nominalen atribut $PrevEvt$ (sloj I_1). Preostale atribute (Qtr, $Time$, $PtsDiff$ in $PrevDur$), ki so na vhodu sloja I_2, obravnavamo kot zvezne. Vsi skriti sloji (H_1 do H_5) so polno povezani in uporabljajo aktivacijsko funkcijo ReLU (Rectified Linear Unit). Izhod (sloj O_1) je prav tako polno povezan, uporablja aktivacijsko funkcijo softmax in predstavlja verjetnostno porazdelitev napovedi kategorične ciljne spremenljivke Evt.

Po končanem učenju mreže uteži na povezavah med sloji I_3 in H_3 ter I_4 in H_4 predstavljajo vektorsko vložitev domače oziroma gostujoče ekipe. Zaradi eničnega kodiranja ekip se v nevrone na skritih nivojih H_3 in H_4 vpišejo samo vrednosti vhodnih nevronov, ki ustrezajo izbranima ekipama. Iz tega razloga lahko sloja H_3 in H_4 obravnavamo kot skrito oziroma implicitno atributno predstavitev ekip.

Da bi dobili čim bolj robusten model M_{Evt}, učenje mreže izvedemo z naslednjim iterativnim postopkom, pri katerem gradimo čedalje kompleksnejše modele. V prvem koraku izvedemo učenje nevronske mreže, ki je sestavljena iz slojev I_1 in H_1 ter polno povezanega izhodnega sloja O_1. Ta mreža se nauči veljavnih sosledij dogodkov. V drugem koraku izvedemo učenje nevronske mreže, sestavljene iz slojev I_1, I_2, H_1 in H_2 ter polno povezanega izhodnega sloja O_1, pri čemer začetne vrednosti vhodnih uteži v sloj H_1 nastavimo na vrednosti iz mreže, naučene v prvem koraku. Tako dobimo model, ki upošteva lokalni kontekst pri napovedovanju naslednjega dogodka. V tretjem koraku izvedemo učenje celotne nevronske mreže, pri čemer začetne vrednosti vstopnih uteži v sloja H_1 in H_2 nastavimo na vrednosti iz mreže, naučene v drugem koraku. Tako dobimo celoten model M_{Evt}, ki pri napovedovanju naslednjega dogodka upošteva tudi latentno predstavitev ekip.

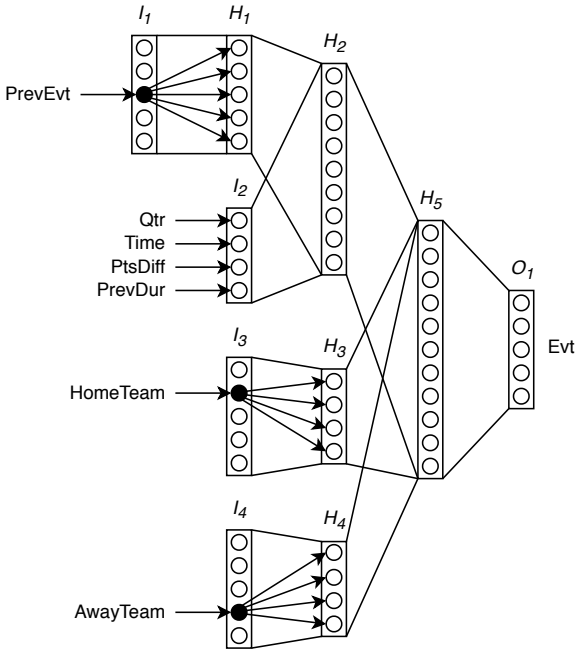
Napovedovanje časa med dogodki (model M_{Dur}) tokrat obravnavamo kot 25-razredni klasifikacijski problem (zaloga vrednosti ciljne spremenljivke Dur, ki predstavlja časovni interval v sekundah, so cela števila na intervalu od 0 do 24). Osnovna motivacija za takšen pristop je možnost vzorčenja iz napovedane verjetnostne porazdelitve za ciljno spremenljivko, kar omogoča generiranje bolj pestrih in s tem tudi bolj realističnih

Tabela 2: Izsek podatkov play-by-play iz prve četrtine tekme NBA, odigrane 28. marca 2010 med gostujočim Denverjem in domačim Orlandom. En zapis vsebuje podatek o četrtini (stolpec Qtr), času do konca četrtine (stolpec Time), ekipi (stolpec Team) in igralcu (stolpec Player), ki je izvedel akcijo (stolpec Action). Stolpec Score vsebuje oznako vodilne ekipe in trenutni rezultat.

Qtr	Time	Team	Player	Action	Score
1	6:51	DEN	N Hilario	2 Point Miss	DEN 10-8
1	6:48	DEN	A Afflalo	Offensive Rebound	DEN 10-8
1	6:40	ORL	D Howard	Foul	DEN 10-8
1	6:40	DEN	N Hilario	1 Point Free Throw	DEN 11-8
1	6:40	DEN	N Hilario	Missed Free Throw	DEN 11-8
1	6:38	ORL	M Barnes	Defensive Rebound	DEN 11-8

Tabela 3: Učni primeri, ki ustrezajo dogodkom play-by-play iz tabele 2. Posamezna vrstica predstavlja tranzicijo in dogodka PrevEvt v Evt (ciljna spremenljivka). Stolpci od Aatt_1 do Aatt_w predstavljajo vrednosti atributov, ki opisujejo zmogljivosti gostujoče ekipe. Podobno stolpci od Hatt_1 do Hatt_w opisujejo zmogljivosti domače ekipe. Stolpci Qtr, Time in PtsDiff predstavljajo četrtino, čas do konca četrtine (v sekundah) in trenutno razliko v rezultatu (z vidika domače ekipe) ob začetku tranzicije. Stolpec PrevDur je čas trajanja prejšnje tranzicije (tiste, ki je pripeljala do dogodka PrevEvt).

Aatt_1	...	Aatt_w	Hatt_1	...	Hatt_w	Qtr	Time	PtsDiff	PrevDur	PrevEvt	Evt
a_1	...	a_w	h_1	...	h_w	1	411	-2	-	A2PA	AORB
a_1	...	a_w	h_1	...	h_w	1	408	-2	3	AORB	HF
a_1	...	a_w	h_1	...	h_w	1	400	-2	8	HF	AFT2M
a_1	...	a_w	h_1	...	h_w	1	400	-3	0	AFT2M	AFT1A
a_1	...	a_w	h_1	...	h_w	1	400	-3	0	AFT1A	HDRB



Slika 1: Arhitektura nevronske mreže za klasifikacijski problem napovedovanja naslednjega dogodka na tekmi (model M_{Evt}).

simulacij. Arhitektura nevronske mreže za model M_{Dur} je zelo podobna tisti, prikazani na sliki 1. Prva razlika je v tem, da ima vhodni sloj I_1 dvakrat več nevronov in prejema enično kodirane vrednosti nominalnih atributov $PrevEvt$ in Evt (prejšnji in naslednji dogodek na tekmi). Druga razlika je izhodni sloj O_1, ki ima zdaj 25 nevronov. Učenje mreže za model M_{Dur} prav tako izvedemo z iterativnim postopkom, ki je analogen tistemu za učenje modela M_{Evt}.

Pomanjkljivost opisanega pristopa za učenje modelov M_{Evt} in M_{Dur} z vektorsko vložitvijo je v statičnosti latentne predstavitve ekip. Če želimo z modeloma zajeti spremembe v zmogljivosti ekip, ju je treba dodatno učiti in tako prilagoditi latentno predstavitev novi situaciji. Robustnost dodatnega učenja modelov povečamo tako, da fiksiramo sloja H_1 in H_2 ter s tem dovolimo mreži posodobitev le latentnih predstavitev ekip.

4 EKSPERIMENTALNA EVALVACIJA

Za eksperimentalno evalvacijo predstavljenih metod smo uporabili podatke play-by-play tekem, odigranih v rednem delu osmih sezon lige NBA (od 2008/09 do 2015/16). Podatke smo pridobili na spletnem naslovu stats.nba.com.

4.1 Primerjava različnih modelov za M_{Evt}

Primerjali smo različne modele po točnosti napovedovanja naslednjega dogodka na košarkarski tekmi (za podrobnejši opis glej [13]).

HOM_{Evt} je najpreprostejši med obravnavanimi modeli za M_{Evt}. Verjetnostna porazdelitev naslednjega dogodka je modelirana kot relativna frekvenca mogočih nadaljevanj, pogojena s trenutnim dogodkom (preostale atribute iz učne množice ignoriramo).

Naslednji trije modeli so v obliki odločitvenega drevesa z multinomialno logistično regresijo v listih. Korensko vozlišče drevesa je pri vseh modelih ročno izbrano in predstavlja nebinarno razbitje problemskega prostora glede na trenutni dogodek (predstavlja ga vrednost atributa $PrevEvt$), ustrezna poddrevesa pa so avtomatsko

zgrajena z rekurzivno binarno delitvijo, pri čemer se kot mera za izbiro delitvenega kriterija uporablja ocena MDL [23]. Gradnja drevesa se ustavi, ko število primerov v vozlišču pade pod 500 ali ko nobeno razbitje ni kompresivno glede na oceno MDL. Logistični modeli v listih drevesa uporabljajo celoten nabor atributov iz učne množice, na podlagi katere je drevo zgrajeno.

QTD_{Evt} je model, naučen na podmnožici atributov, ki se nanašajo samo na lokalni kontekst tekme (četrtina, čas do konca četrtine, razlika v rezultatu, trajanje prejšnje akcije in trenutni dogodek), medtem ko se opisi zmogljivosti ekip ignorirajo.

FF_{Evt} je model, naučen na podlagi vseh atributov v učni množici, vključno z zmogljivostmi ekip. Slednje so opisane z ekspertnimi atributi, znanimi kot štirje faktorji [11].

ACA_{Evt} je model, podoben zgornjemu, le da so atributi za opisovanje zmogljivosti ekip dobljeni avtomatsko z uporabo metode iz [13]. Definicijo atributov smo sestavili na podlagi 3.690 tekem iz treh sezon NBA (od 2008/09 do 2010/11).

Zadnji model v primerjavi je naš novi model NN_{Evt}, ki ustreza nevronski mreži, prikazani na sliki 1. Število nevronov na slojih H_3 in H_4 je nastavljeno na 4 (vektorska vložitev v štiridimenzionalni prostor). Število nevronov na preostalih skritih slojih smo izbrali empirično, in sicer: H_1(38), H_2(42) in H_5(50). Celotna mreža ima 8.024 prostih parametrov. Za učenje mreže smo uporabili 40 iteracij (angl. epoch) pri velikosti serije (angl. batch size) 128.

Vse modele razen NN_{Evt} smo učili na tekmah iz dveh zaporednih sezon, testirali pa na naslednji, tretji sezoni. Pri tem so vrednosti atributov v učnih primerih izračunane za vsako sezono posebej (tj. zmogljivosti ekip na začetku ene sezone se ne navezujejo na njihovo uspešnost v predhodni sezoni).

Model NN_{Evt} smo najprej naučili na vseh tekmah ene sezone, nato smo ga testirali na naslednji sezoni. Testiranje smo izvedli po igralnih dneh. Po končanem testiranju modela na tekmah tekočega igralnega dneva smo model posodobili z dodatnim učenjem na tistih tekmah.

Za vsak testni primer so vsi obravnavani modeli vrnili verjetnostno porazdelitev dogodka, ki se bo naslednji zgodil na tekmi. Različnost med vrnjenimi predikcijami in dejanskimi dogodki smo ocenili z Brierjevo mero [24]. Rezultati so prikazani v tabeli 4.

Iz rezultatov lahko sklepamo, da je model HOM_{Evt} bistveno slabši od preostalih modelov, ki so po uspešnosti napovedovanja naslednjega dogodka precej primerljivi. Model QTD_{Evt} je presenetljivo dober napovedovalec naslednjega dogodka, čeprav ne upošteva značilnosti ekip. Iz tega lahko sklepamo, da lokalni kontekst trenutka bolj vpliva na naslednji dogodek kot značilnosti ekip – to zlasti drži za ligo NBA, kjer so razlike med ekipami načrtno precej majhne, da bi se spodbujala negotovost tekmovanja. V obeh sezonah se je kot najboljši izkazal model NN_{Evt}.

Tabela 4: Povprečne vrednosti ocene Brier score, izmerjene pri napovedovanju naslednjega dogodka v rednem delu tekme (četrtine 1–4) rednega dela sezon NBA 2014/15 in 2015/16. Standardna napaka pri vseh rezultatih je 5.5×10^{-4}.

Model	2014/15 Brier score	2015/16 Brier score
HOM_{Evt}	0.5806	0.5795
QTD_{Evt}	0.5666	0.5658
FF_{Evt}	0.5670	0.5661
ACA_{Evt}	0.5674	0.5665
NN_{Evt}	0.5651	0.5646
	N = 591397	N = 596345

4.2 Analiza verodostojnosti generiranih simulacij

Za generiranje simulacij poteka košarkarske tekme potrebujemo par modelov: M_{Evt} za napovedovanje naslednjega dogodka in M_{Dur} za napovedovanje, koliko igralnega časa preteče do nastopa tega dogodka. V nadaljevanju predstavljamo uporabljene modele za M_{Dur}.

HOM_{Dur} je najpreprostejši med obravnavanimi modeli. Čas med zaporednima dogodkoma modelira z vzorčenjem iz populacije vseh prehodov (opaženih v učni množici) med tema dogodkoma.

$HOMA_{Dur}$ je podoben modelu HOM_{Dur}, le da ločeno modeliramo končnico (zadnjih 60 sekund) od preostanka četrtine. Motivacija za ta pristop je ugotovitev, da se dinamika igre bistveno spremeni v končnicah četrtin [21].

Naslednji trije modeli so v obliki regresijskega drevesa. Prva dva nivoja drevesne strukture sta ročno določena in razdelita problemski prostor glede na predhodni dogodek (korensko vozlišče) in napovedani naslednji dogodek (vozlišče pod korenom). Preostanek drevesa pa je avtomatsko zgrajen z rekurzivno delitvijo po kriteriju najmanjših kvadratov na podlagi celotnega nabora atributov iz učne množice. Gradnja drevesa se ustavi, ko v listu ostane premalo učnih primerov, pri čemer se prag eksperimentalno določi na podlagi validacijske množice. Predikcijo modela v obliki diskretne verjetnostne porazdelitve igralnega časa med zaporednima dogodkoma dobimo iz empirične porazdelitve primerov v listih drevesa.

QTD_{Dur} je model, naučen na podmnožici atributov, ki se nanašajo samo na lokalni kontekst tekme (četrtina, čas do konca četrtine, razlika v rezultatu, trajanje prejšnje akcije, trenutni in naslednji dogodek), medtem ko se opisi zmogljivosti ekip ignorirajo.

FF_{Dur} je model, naučen na celotnem naboru atributov vključno z atributi za opis zmogljivosti ekip. Uporabljeni so ekspertni atributi na podlagi štirih faktorjev [11].

ACA_{Dur} je prav tako model, naučen na celotnem naboru atributov, le da so tokrat atributi za opisovanje

zmogljivosti ekip dobljeni z avtomatskim postopkom iz [13]. Uporabljeni atributi so zgrajeni z namenom napovedovanja časa med zaporednima dogodkoma na tekmi. Definicija atributov je zgrajena na podlagi 3690 tekem iz treh NBA sezon (od 2008/2009 do 2010/2011).

NN_{Dur} je nevronska mreža, ki modelira čas med dogodkoma na podlagi vektorske vložitve v štiridimenzionalni prostor. Število nevronov na skritih slojih je bilo izbrano empirično: H_1(76), H_2(80), H_3(4), H_4(4) in H_5(88). Celotna mreža ima 22.637 prostih parametrov. Za učenje mreže smo uporabili 40 iteracij pri velikosti serije 128.

Vse modele smo učili na podoben način kot pri napovedovanju naslednjega dogodka. Drevesne modele smo po učenju porezali na globino, ki optimizira kakovost napovedi na validacijski množici (redni del sezone 2010/11).

Z različnimi pari modelov M_{Evt}-M_{Dur} smo generirali 10 simulacij za vsako tekmo iz sezon 2014/15 in 2015/16. Tako smo z vsakim parom modelov dobili 23.800 simulacij*. Generirane simulacije smo nato primerjali z dejanskimi tekmami.

Verodostojnost generiranih simulacij smo najprej ocenili s pomočjo tradicionalnih zbirnih statistik (angl. box score), ki predstavljajo števce osnovnih dogodkov na posamezni tekmi. Primerjali smo porazdelitev vrednosti zbirnih statistik iz simuliranih tekem s porazdelitvami na dejanskih tekmah. Kot mero različnosti med diskretnima porazdelitvama S (empirično dobljena iz generiranih simulacij) in E (empirično dobljena iz dejanskih tekem) smo uporabili Kullback-Leiblerjevo (KL) divergenco, ki jo lahko interpretiramo kot količino izgubljene informacije, ko za aproksimacijo porazdelitve E uporabimo porazdelitev S. V tabeli 5 je prikazana parna primerjava različnih modelov glede na vrednost KL-divergence med porazdelitvami osnovnih košarkarskih statistik.

V naslednjem eksperimentu smo generirane simulacije primerjali z empiričnimi rezultati na podlagi nekaterih karakteristik dinamike igre in gibanja rezultata (dolžina posesti, čas med zaporednima košema, dolžina niza zaporednih košev, število izmenjav vodilnega na tekmi in podobno), ki sta jih predstavila Gabel in Redner [18]. Parna primerjava med nekaterimi modeli je prikazana v tabeli 6.

Rezultati sugerirajo, da najbolj verodostojne simulacije glede na predstavljene kriterije generirata kombinaciji NN_{Evt}-HOM_{Dur} in NN_{Evt}-$HOMA_{Dur}$. Manj prepričljive simulacije generirajo modeli, ki ne upoštevajo konteksta (HOM_{Evt}-HOM_{Dur}) oziroma značilnosti ekip (QTD_{Evt}-QTD_{Dur}).

4.3 Napovedovanje zmagovalca

Kakovost napovedovanja zmagovalca lahko uporabimo kot še eno možno oceno verodostojnosti generiranih simulacij. Od dobrega simulatorja pričakujemo, da bodo izidi generiranih tekem v povprečju skladni z rezultati dejanskih tekem.

Posamezen par modelov M_{Evt}-M_{Dur} smo ovrednotili glede na kakovost napovedovanja zmagovalca tekme s pomočjo naslednjega postopka. Za vsako tekmo iz sezon 2014/15 in 2015/16 smo generirali po 1.000 simulacij. Delež simuliranih tekem, ki jih je zmagalo domače moštvo, smo uporabili kot verjetnostno napoved za zmago domačina. Nato smo uporabili Brierjevo mero za oceno kakovosti napovedi kot povprečno kvadratno razliko med napovedanimi verjetnostmi zmag domačinov in dejanskimi izidi tekem.

Iz testa smo izločili dejanske tekme, ki so po rednem delu končane brez zmagovalca. Tako smo kakovost napovedovanja zmagovalca za sezono 2014/15 določili na podlagi 1.117 parov, medtem ko smo za sezono 2015/16 uporabili 1.113 parov (od teoretičnih 1.230 tekem, odigranih v posameznih sezonah). Dobljeni rezultati so zbrani v tabeli 7. Najboljši napovedovalec zmagovalca je par FF_{Evt}-FF_{Dur}, ki je naučen z uporabo ekspertnih atributov. Nekoliko slabši je par ACA_{Evt}-ACA_{Dur}, naučen z avtomatsko generiranimi atributi. Modela HOM_{Evt}-HOM_{Dur} in QTD_{Evt}-QTD_{Dur} sta pričakovano najslabša, saj ne upoštevata dejanskih karakteristik ekip in vedno napovedujeta povprečen izid. Pri tem eksperimentu sta para NN_{Evt}-NN_{Dur} in NN_{Evt}-$HOMA_{Dur}$ dosegla relativno slab rezultat, kar sugerira, da je latentni opis ekip samo delno zajel njihove zmogljivosti.

Za primerjavo, model, ki vedno napove zmago domačina v skladu z deležem domačih zmag v učnih podatkih, za sezono 2014/15 doseže klasifikacijsko točnost 0.58 in povprečno Brierjevo mero 0.244 ($\pm$ 0.0001). Za sezono 2015/16 pa so rezultati tega modela: klasifikacijska točnost 0.59 in povprečna Brierjeva mera 0.242 ($\pm$ 0.0001).

Najboljši javni vir napovedi športnih izidov so kvote stavnic. Stavna borza Betfair je za sezono 2014/15 dosegla klasifikacijsko točnost 0.71 in povprečno Brierjevo mero 0.196 ($\pm$ 0.0051).

5 SKLEP

V članku je predstavljena metodologija za modeliranje razvoja košarkarske tekme. Izhajali smo iz markovskega modela, pri čemer smo prostor stanj razširili z opisom trenutnega konteksta tekme. Tako smo v opis stanja vključili tudi del zgodovine razvoja tekme in s tem omilili markovsko lastnost modela. Verjetnost prehoda aproksimiramo z dvema modeloma, ki zaporedno (in pogojeno eden na drugega) napovedujeta posamezne dele opisa naslednjega stanja. Metodologija je uporabna pri modeliranju vseh športov, ki jih lahko dobro opišemo s stanji in prehodi med njimi.

Predstavili smo postopek za avtomatsko generiranje latentnega atributnega prostora za opis zmogljivosti ekip,

*Teoretično bi morali dobiti 24.600 simulacij, toda začetnih tekem v sezoni ni mogoče simulirati z modeli, ki uporabljajo eksplicitne opise zmogljivosti ekip, saj za nje ne moremo določiti vrednosti atributov.

Tabela 5: Parna primerjava glede na vrednost KL-divergence med porazdelitvami osnovnih košarkarskih statistik, ki so izmerjene na dejanskih in simuliranih tekmah (združeni sezoni NBA 2014/15 in 2015/16). Zapis oblike X : Y v i-ti vrstici in j-tem stolpcu pomeni, da je v neposredni primerjavi i-ti model boljši X-krat, j-ti model pa Y-krat. Zaradi enostavnosti zapisa so uporabljene kratice parov modelov. Oznake HOM, QTD, FF, ACA in NN predstavljajo pare istoimenskih modelov M_{Evt}-M_{Dur}. Oznaka NH predstavlja par NN_{Evt}-HOM_{Dur}, oznaka NHA pa par NN_{Evt}-$HOMA_{Dur}$.

	HOM	QTD	FF	ACA	NN	NH	NHA
HOM	-	16 : 8	13 : 11	8 : 16	9 : 15	6 : 18	5 : 19
QTD	8 : 16	-	4 : 20	3 : 21	4 : 20	3 : 21	3 : 21
FF	11 : 13	20 : 4	-	9 : 15	10 : 14	6 : 18	4 : 20
ACA	16 : 8	21 : 3	15 : 9	-	12 : 12	7 : 17	5 : 19
NN	15 : 9	20 : 4	14 : 10	12 : 12	-	10 : 14	9 : 15
NH	18 : 6	21 : 3	18 : 6	17 : 7	14 : 10	-	7 : 17
NHA	19 : 5	21 : 3	20 : 4	19 : 5	15 : 9	17 : 7	-

Tabela 6: Parna primerjava glede na vrednost KL-divergence med porazdelitvami košarkarskih statistik za opis dinamike doseganja točk, ki so izmerjene na dejanskih in simuliranih tekmah (združeni sezoni NBA 2014/15 in 2015/16). Oblika zapisov in kratice so enake kot v tabeli 5.

	HOM	QTD	FF	ACA	NN	NH	NHA
HOM	-	4 : 4	2 : 6	2 : 6	5 : 3	1 : 7	0 : 8
QTD	4 : 4	-	2 : 6	3 : 5	5 : 3	2 : 6	2 : 6
FF	6 : 2	6 : 2	-	4 : 4	5 : 3	2 : 6	1 : 7
ACA	6 : 2	5 : 3	4 : 4	-	5 : 3	2 : 6	1 : 7
NN	3 : 5	3 : 5	3 : 5	3 : 5	-	1 : 6	1 : 6
NH	7 : 1	6 : 2	6 : 2	6 : 2	6 : 1	-	5 : 2
NHA	8 : 0	6 : 2	7 : 1	7 : 1	6 : 1	2 : 5	-

Tabela 7: Evalvacija parov M_{Evt}-M_{Dur} na podlagi napovedovanja zmagovalca, izmerjeno na tekmah rednega dela lige NBA v sezonah 2014/15 in 2015/16. Klasifikacijska točnost in povprečna Brierjeva mera se nanašata na napovedano verjetnost zmage domačega moštva. Vrednosti v oklepajih predstavljajo standardno napako.

	2014/15		2015/16	
Model	Točnost	Brier score	Točnost	Brier score
HOM_{Evt}-HOM_{Dur}	0.58	0.244 (± 0.0022)	0.59	0.243 (± 0.0019)
QTD_{Evt}-QTD_{Dur}	0.58	0.244 (± 0.0026)	0.59	0.241 (± 0.0022)
FF_{Evt}-FF_{Dur}	0.65	0.215 (± 0.0042)	0.65	0.214 (± 0.0043)
ACA_{Evt}-ACA_{Dur}	0.63	0.223 (± 0.0043)	0.65	0.217 (± 0.0044)
NN_{Evt}-NN_{Dur}	0.61	0.231 (± 0.0038)	0.61	0.232 (± 0.0034)
NN_{Evt}-$HOMA_{Dur}$	0.62	0.230 (± 0.0038)	0.61	0.231 (± 0.0033)
	N = 1117		N = 1113	

ki temelji na tehniki vektorskih vložitev. Eksperimentalna evalvacija, izvedena na sezonah lige NBA 2014/15 in 2015/16, je pokazala, da modeli, naučeni v latentnem atributnem prostoru, generirajo verodostojne simulacije. Po drugi strani so ti modeli izkazali nizko točnost napovedovanja zmagovalca, kar sugerira, da je latentni opis samo delno zajel zmogljivosti ekip.

V nadaljnjem delu bomo preizkušali različne nastavitve dimenzionalnosti latentnega prostora pri vektorski vložitvi ekip. Preizkušali bomo še drugačne arhitekture nevronske mreže kakor tudi sam postopek in parametre učenja.

LITERATURA

[1] R. T. Stefani, "Improved least squares football, basketball, and soccer predictions," *IEEE transactions on systems, man, and cybernetics*, vol. 10, no. 2, pp. 116–123, 1980.

[2] R. D. Baker and I. G. McHale, "Forecasting exact scores in National Football League games," *International Journal of Forecasting*, vol. 29, no. 1, pp. 122–130, 2013.

[3] H. Manner, "Modeling and forecasting the outcomes of nba basketball games," *Journal of Quantitative Analysis in Sports*, vol. 12, no. 1, pp. 31–41, 2016.

[4] J. Kubatko, D. Oliver, K. Pelton, and D. T. Rosenbaum, "A starting point for analyzing basketball statistics," *Journal of Quantitative Analysis in Sports*, vol. 3, no. 3, pp. 1–22, 2007.

[5] N. Hirotsu and M. Wright, "Using a markov process model of an association football match to determine the optimal timing of substitution and tactical decisions," *Journal of the Operational Research Society*, vol. 53, no. 1, pp. 88–96, 2002.

[6] M. Goldman and J. M. Rao, "Effort vs. concentration: the asymmetric impact of pressure on NBA performance," in *Proceedings MIT Sloan sports analytics conference*, pp. 1–10, 2012.

[7] S. Merritt and A. Clauset, "Scoring dynamics across professional team sports: tempo, balance and predictability," *EPJ Data Science*, vol. 3, no. 1, p. 1, 2014.

[8] A. Clauset, M. Kogan, and S. Redner, "Safe leads and lead changes in competitive team sports," *Physical Review E*, vol. 91, no. 6, p. 062815, 2015.

[9] A. Bocskocsky, J. Ezekowitz, and C. Stein, "The hot hand: A new

approach to an old 'fallacy'," in *8th Annual MIT Sloan Sports Analytics Conference*, pp. 1–10, 2014.
[10] R. P. Schumaker, O. K. Solieman, and H. Chen, "Predictive modeling for sports and gaming," in *Sports Data Mining*, pp. 55–63, Springer, 2010.
[11] D. Oliver, *Basketball on paper: rules and tools for performance analysis*. Potomac Books, Inc., 2004.
[12] J. Hollinger, *Pro Basketball Prospectus 2003-2004*. Brassey's, 2003.
[13] P. Vračar, E. Štrumbelj, and I. Kononenko, "Automatic attribute construction for basketball modelling," *Knowledge and Information Systems*, vol. 62, no. 2, pp. 541–570, 2020.
[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
[15] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.
[16] E. Asgari and M. R. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PloS one*, vol. 10, no. 11, 2015.
[17] H. S. Stern, "A Brownian motion model for the progress of sports scores," *Journal of the American Statistical Association*, vol. 89, no. 427, pp. 1128–1134, 1994.
[18] Alan Gabel, Sidney Redner, et al. Random walk picture of basketball scoring. *Journal of Quantitative Analysis in Sports*, 8(1):1416, 2012.
[19] K. Shirley, "A Markov model for basketball," in *New England Symposium for Statistics in Sports*, 2007.
[20] E. Štrumbelj and P. Vračar, "Simulating a basketball match with a homogeneous Markov model and forecasting the outcome," *International Journal of Forecasting*, vol. 28, no. 2, pp. 532–542, 2012.
[21] P. Vračar, E. Štrumbelj, and I. Kononenko, "Modeling basketball play-by-play data," *Expert Systems with Applications*, vol. 44, pp. 58–66, 2016.
[22] M. A. Alcorn, "2vec: statistic-free talent modeling with neural player embeddings," MIT Sloan Sports Analytics Conference, 2016.
[23] I. Kononenko, "On biases in estimating multi-valued attributes," in *Ijcai*, vol. 95, pp. 1034–1040, 1995.
[24] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly Weather Review*, vol. 75, pp. 1–3, 1950.

Petar Vračar je leta 2017 doktoriral s področja računalništva in informatike na Univerzi v Ljubljani. Je asistent na Fakulteti za računalništvo in informatiko. Raziskovalno se ukvarja z uporabo metod strojnega učenja pri modeliranju športnih dogodkov. Poleg izdelave verodostojnih simulatorjev je zainteresiran za razvoj metod za avtomatsko ocenjevanje zmogljivosti ekip brez uporabe domenskega predznanja.