

ANSAMBL METODE STROJNOG UČENJA

ENSEMBLE METHODS OF MACHINE LEARNING

Aleksandar Stojanović, Željko Kovačević, Danko Ivošević

Tehničko veleučilište u Zagrebu, Vrbik 8, 10000 Zagreb, Hrvatska

SAŽETAK

Ansambl metode strojnog učenja privukle su značajnu pažnju posljednjih godina zbog svoje sposobnosti poboljšanja točnosti i robusnosti prediktivnih modela. Ove metode kombiniraju rezultate više pojedinačnih modela kako bi proizvele konačno predviđanje. Ansambl metode otpornije su na odstupanja u podacima. Mogu se primijeniti na širok raspon problema u području strojnog učenja, uključujući klasifikaciju, regresiju i klasteriranje. Općenito mogu pomoći u poboljšanju izvedbe modela strojnog učenja i naširoko se koriste u praksi. Zbog njihove velike važnosti i značaja ovaj članak pruža pregled nekih od najčešće korištenih ansambl metoda u strojnom učenju, uključujući pakiranje, pojačavanje i slaganje, te opisuje prednosti i ograničenja svakog od ovih pristupa.

Ključne riječi: *ansambl, strojno učenje, pakiranje, pojačavanje, slaganje.*

ABSTRACT

Ensemble machine learning methods have attracted considerable attention in recent years due to their ability to improve the accuracy and robustness of predictive models. These methods combine the results of multiple individual models to produce a final prediction. Ensemble methods are more resistant to data deviations. They can be applied to a wide range of problems in the field of machine learning, including classification, regression, and clustering. They can generally help improve the performance of machine learning models and are widely used in practice. Due to their great importance and significance, this article provides an overview of some of the most used ensemble methods in machine learning, including bagging, boosting, and stacking, and

describes the advantages and limitations of each of these approaches.

Keywords: *ensemble, machine learning, bagging, boosting, stacking.*

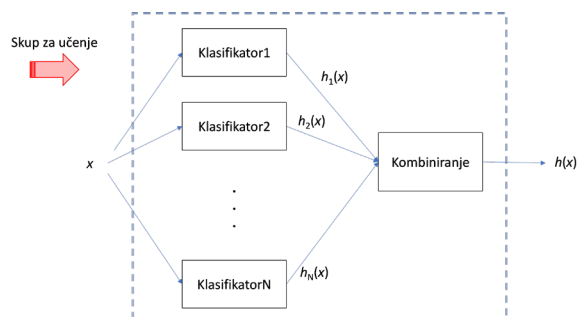
1. UVOD

1. INTRODUCTION

Strojno učenje je grana umjetne inteligencije koja se bavi učenjem, rasuđivanjem i djelovanjem na osnovu podataka [1]. To se postiže izradom računalnih programa koji obrađuju podatke, izlučuju iz njih korisne informacije, naprave predviđanje o nepoznatim svojstvima i predlažu aktivnosti ili odluke. Takvo predviđanje može se postići upotrebom jednog ili više *prediktora* ili *prediktorskih varijabli*. Primjerice, u jednom televizijskom kvizu natjecatelj bi došao u situaciju da ne zna odgovor na postavljeno pitanje. U tom bi slučaju imao dvije opcije: da nazove prijatelja ili da pita publiku. Prijatelja, u ovom slučaju, možemo smatrati jednim dobrim prediktorom, a publiku kolekcijom prediktora. Općenito, ansambl metode ističu činjenicu da korištenje više prediktora može biti bolje od korištenja samo jednog, čak i ako je taj jedan dobar prediktor (u ovom primjeru prijatelj) stručnjak za dotično područje.

Ansambl metoda kombinira više prediktora koji mogu biti istog ili različitog tipa [2]. Ako su prediktori istog tipa onda se koristi više instanci tog prediktora, dok se prediktori različitog tipa koriste kada nije sigurno koji je tip prediktora najbolji za problem koji se rješava.

Ansambl se sastoji od skupa pojedinačnih klasifikatora istog ili različitog tipa treniranih na nekom skupu podataka za učenje čija se predviđanja kombiniraju za klasifikaciju novih instanci [2]. Ovaj postupak prikazuje Slika 1.



Slika 1 Konceptualni dijagram ansambl učenja. Predviđanja pojedinačnih klasifikatora kombiniraju se u konačni rezultat. [2]

Figure 1 Conceptual diagram of ensemble learning. Prediction of each classifier is combined with the others into the final result. [2]

Za jedan ulaz x svaki klasifikator proizvede svoju hipotezu $h_i(x)$, a cilj je da oni svi zajedno proizvedu bolju hipotezu. Nadalje, možemo pretpostaviti da klasifikatori imaju međusobno jednaku ili različitu važnost. U osnovnom obliku ansambla izvodi se ravnomjerno ili težinsko glasanje da bi se donijela konačna odluka. Odluka se donosi zbrajanjem umnožaka težinskih udjela klasifikatora s njihovim pojedinačnim izlazima $h_i(x)$ te se na temelju toga donosi konačna odluka $h(x)$.

Neophodan i dovoljan uvjet da bi ansambl klasifikatori bili precizniji od svojih pojedinačnih članova je taj da klasifikatori moraju biti različiti i komplementarni [3]. Točan klasifikator je onaj koji ima stopu pogrešaka manju od one kod slučajnog pogađanja na novim vrijednostima x . Dva klasifikatora su raznovrsna ako daju različite greške nad istim ulaznim podacima [3]. Uopće, načini kojima se postiže različitost ili komplementarnost klasifikatora uključuju upotrebu različitih algoritama, primjenu različitih skupova za učenje ili odabir različitih skupova značajki koje se predviđaju. Osnovni klasifikatori su homogeni ako koriste isti algoritam klasifikacije odnosno heterogeni ako koriste različite algoritme klasifikacije [4].

2. OSNOVNI ALGORITMI KLASIFIKACIJE

2. BASIC CLASSIFICATION ALGORITHMS

U nastavku su opisani osnovni algoritmi klasifikacije s pojmovima koji su bitni za

razumijevanje područja strojnog učenja, a zatim se predstavljaju ansambl metode koje se temelje na uporabi tih algoritama kao sastavnog dijela konačne klasifikacije koju obavljaju.

2.1. STABLA ODLUKE

2.1. DECISION TREES

Jedan od najčešće korištenih klasifikatora u ansambl učenju su stabla odluke (engl. *decision trees*) [2] [5]. Stablo odluke je struktura podataka koja predstavlja skup odluka i njihovih posljedica. U najjednostavnijem obliku sastoje se od čvorova, grana i listova. Čvorovi predstavljaju značajke rješavanog problema, grane moguće vrijednosti čvorova te listovi izlaze, odnosno razrede klasifikacije ulaznih podataka. U osnovi se ulazni prostor rekurzivno dijeli na lokalne potprostore klasifikacije. Jedan od klasičnih primjera iz literature je odluka o igranju tenisa u ovisnosti o meteorološkim uvjetima (Slika 2).



Slika 2 Primjer stabla odluke. [6]

Figure 2 Decision tree example. [6]

Stabla odluke relativno su jednostavna za razumjeti, lako im je dodati nove scenarije te mogu donijeti odluku i s malom količinom podataka. Primjeri korištenja su u analizi isplativosti poslovnih odluka kao primjerice pri procjeni rizika pri ulaganjima te u operacijskim istraživanjima [7]. S druge strane, i manja promjena u stablu može utjecati na njegov ishod, tj. dovesti do velikih promjena u rezultatu i zahtijevati ponovno izračunavanje vjerojatnosti različitih mogućih grana što u složenim sustavima s mnogo varijabli i ishoda može predstavljati problem.

2.2. STROJEVI POTPORNIH VEKTORA

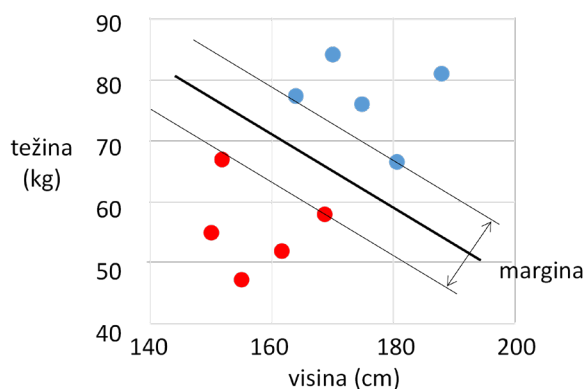
2.2. SUPPORT VECTOR MACHINES

Strojevi potpornih vektora (engl. *Support Vector*

Machines ili SVM) vrsta su algoritma nadziranog učenja koja se koristi za klasifikaciju i regresijsku analizu [2] [8]. SVM algoritmi rade pronalazeći hiperravninu koja razdvaja podatke u različite klase. Hiperravnina je odabrana tako da maksimizira udaljenost između najbližih podatkovnih točaka iz svake klase, poznatu kao *marginu*. Ovaj pristup pomaže da se postigne visoka točnost i robusnost u zadacima klasifikacije čak i u slučajevima kada podaci nisu linearno odvojivi. Zbog svoje učinkovitosti i svestranosti SVM pristup naširoko je korišten u raznim područjima kao što su bioinformatika, financije i obrada slika [9].

Jedna od ključnih prednosti strojeva potpornih vektora je sposobnost učinkovitog rukovanja visokodimenzionalnim podacima što ih čini prikladnima za zadatke kao što je klasifikacija teksta [10]. Koriste se i za otkrivanje anomalija gdje se obučavaju na normalnim podacima i koriste za prepoznavanje odstupanja od normalnog uzorka [11]. Također se upotrebljavaju za prepoznavanje slika i detekciju objekata odnosno za klasifikaciju slika na temelju njihovog sadržaja [8].

Pretpostavimo, primjerice, da se želi klasificirati spol osobe na temelju podataka o visini i težini osobe. Na temelju uzorka takvih podataka može se primijeniti algoritam strojeva potpornih vektora (Slika 3).



Slika 3 Distribucija visina i težina muškaraca (plavo) i žena (crveno).

Figure 3 Male (blue) and female (red) height and weight distribution.

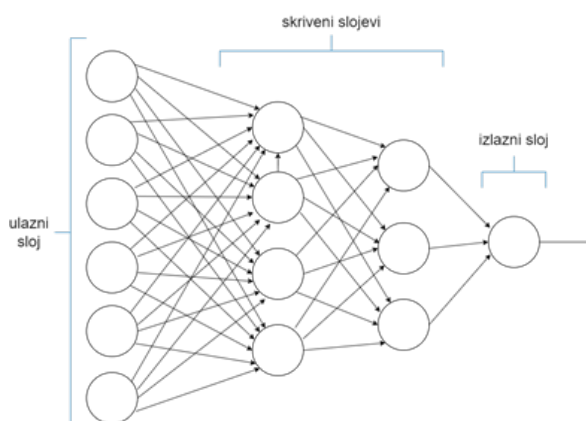
Podebljana crna linija predstavlja najbolju razdjelnu hiperravninu s obzirom da daje najbolju marginu između potpornih vektora (točaka). Općenito, što je veća margina bolja je predikcija.

2.3. NEURONSKE MREŽE

2.3. NEURAL NETWORKS

Neuronske mreže su klasa algoritama strojnog učenja inspiriranih strukturom i funkcijom ljudskog mozga [2] [12]. Često se upotrebljavaju za utvrđivanje obrazaca u podacima i za izradu predviđanja na temelju tih obrazaca. Sastoje se od slojeva međusobno povezanih čvorova ili neurona, od kojih svaki izvodi jednostavnu matematičku operaciju na svom ulazu i prosljeđuje rezultat sljedećem sloju.

Postoji nekoliko tipova neuronskih mreža uključujući neuronske mreže s unaprijednim prijenosom podataka (engl. *Feed Forward Neural Networks*), rekurentne neuronske mreže (engl. *Recurrent Neural Networks*) i konvolucijske neuronske mreže (engl. *Convolutional Neural Networks*) [2] [13]. Neuronske mreže s unaprijednim prijenosom podataka najjednostavnija su vrsta neuronskih mreža koja se sastoji od ulaznog, skrivenog i izlaznog sloja, a koristi se za zadatke kao što su klasifikacija slika i obrada prirodnog jezika. Rekurentne neuronske mreže, s druge strane, imaju petlje u svojoj arhitekturi koje im omogućuju obradu nizova podataka, a koriste se za zadatke kao što su prepoznavanje govora i prevođenje jezika. Konvolucijske neuronske mreže dizajnirane su specifično za zadatke obrade slike i koriste niz konvolucijskih i skupnih slojeva za izdvajanje značajki iz ulaznih podataka [13].



Slika 4 Struktura neuronske mreže. Ulazni podaci prolaze kroz više skrivenih slojeva te na kraju dolaze na izlazni sloj koji daje konačan rezultat [12]

Figure 4 Neural network structure. The the input data passes through several hidden layers and finally comes to the output layer that outputs the final result. [12]

Neuronska mreža sastoji se od tri vrste slojeva: ulaznog, skrivenog i izlaznog sloja koji su povezani neuronima (Slika 4). Podaci iz ulaznog sloja šalju se kao ulazni podaci prvom skrivenom sloju. Neuroni u skrivenom sloju aktiviraju se na temelju ulaza i aktivacijske funkcije te se rezultat prvog skrivenog sloja obrađuje i šalje na sljedeći skriveni sloj i na kraju na izlazni sloj koji daje konačan rezultat.

Neuronske mreže korištene su za postizanje zapaženih rezultata u širokom spektru aplikacija, uključujući računalni vid, obradu prirodnog jezika, prepoznavanje govora i igranje igara na ploči kao što su šah i Go. Neki značajni primjeri uključuju AlphaGo, prvi računalni program koji je pobijedio ljudskog svjetskog prvaka u igri Go [14] i GPT-3, model obrade prirodnog jezika koji može generirati tekst sličan ljudskom [15].

3. ANSAMBL METODE

3. ENSEMBLE METHODS

Metoda ansambla jedna je tehnika strojnog učenja koja uključuje kombiniranje predviđanja više modela radi poboljšanja točnosti i robusnosti. Kombiniranjem izlaza više modela ova metoda može smanjiti rizik od prenaučivosti i poboljšati generalizaciju [3]. Metode ansambla pokazale su se učinkovitima u širokom rasponu zadataka strojnog učenja uključujući klasifikaciju, regresiju i klasteriranje.

Jedna uobičajena vrsta ove metode je *pakiranje* koja uključuje učenje više modela na različitim podskupovima podataka i kombiniranje njihovih predviđanja. Druga popularna skupna metoda je *pojačavanje* koja uključuje iterativno učenje slabih

modela na različitim podskupovima podataka i kombiniranje njihovih izlaza na način koji daje veću težinu primjerima koji su pogrešno klasificirani u prethodnim iteracijama. Još jedna ansambl metoda je *generalizacija slaganjem* kod koje je izlaz jednog ili više modela ulaz u drugi model.

3.1. GENERALIZACIJA SLAGANJEM

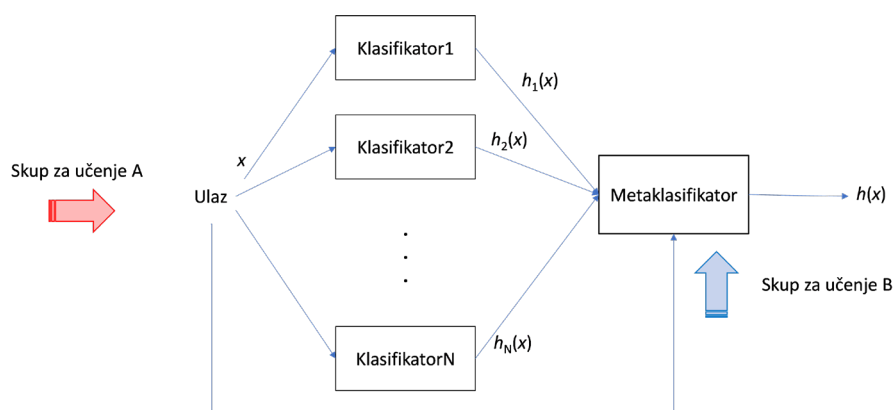
3.1. STACKED GENERALIZATION (engl. *stacking*)

Pretpostavka da kombiniranje predviđanja više različitih modela često daje bolja predviđanja od bilo koje metode zasebno osnova je za generalizaciju slaganjem gdje se kombiniraju rezultati više različitih vrsta klasifikatora kako bi se stvorio bolji metaklasifikator (engl. *meta learner* ili *blender*) [2] [3] (Slika 5).

Tako se, primjerice, predviđene klasifikacije iz klasifikatora stabla na prvoj razini mogu koristiti kao ulazne varijable u novi metaklasifikator na idućoj razini. Taj će metaklasifikator pokušat naučiti kako kombinirati predviđanja iz prethodnih različitih klasifikatora kako bi se postigla maksimalna točnost klasifikacije.

Postupak slaganja je sljedeći:

1. Podijeli se skup podataka za učenje u dva nepovezana skupa.
2. Nekoliko osnovnih klasifikatora se uči na prvom skupu.
3. Testira se osnovne klasifikatore na drugom skupu.
4. Koristeći predviđanja iz 3) kao ulaze i ispravne odgovore kao ishode, uče se klasifikatori na višoj razini.



Slika 5 Konceptualni dijagram slaganja. Metaklasifikator kombinira predviđanja više različitih klasifikatora. [2]

Figure 5 Conceptual diagram for stacking. A meta-classifier combines the predictions of several different classifiers. [2]

Treba imati na umu da su koraci 1) do 3) isti kao i unakrsna provjera valjanosti, ali umjesto da se uzme samo klasifikator s najboljim rezultatima kombiniraju se osnovni klasifikatori, po mogućnosti i nelinearno [2].

Iako dobra ideja, slaganje se manje koristi od pakiranja i pojačavanja, metoda koje su opisane u nastavku.

3.2. PAKIRANJE

3.2. BAGGING (BOOTSTRAP AGGREGATION)

Jedan način da se napravi ansambl je da se koriste klasifikatori temeljeni na istom algoritmu učenja, ali da se svaki uči s različitim skupom podataka. To se zove pakiranje (engl. *bootstrap aggregation*) i jedan je oblik uprosječivanja modela s ciljem poboljšanja stabilnosti i točnosti [2] [3]. Ovaj način također umanjuje varijancu i pomaže u izbjegavanju prenaučivosti (engl. *overfitting*).

Polazeći od početnog skupa podataka napravi se više modela klasifikatora na način da je svaki od njih naučen slučajnim podskupom polaznih podataka (Slika 6). S obzirom da su podaci u takvom podskupu odabrani slučajno može se desiti da se istim dijelovima podataka uči više klasifikacijskih modela. To, međutim, nije problem ako se ne radi o potpuno istim podskupovima podataka za više klasifikatora. Rezultat svakog od ovih klasifikatora jedna je predikcija te se nakon glasanja donosi konačna predikcija.

Pakiranje općenito daje bolje prediktore ako je podskup podataka za učenje veći. Obično se prediktore uči sa do 60% slučajno odabranih podataka iz polaznog skupa za učenje, a poznato je i pravilo „0.632“ koje kaže da bi vjerojatnost uključivanja primjera iz skupa trebala biti 63.2% [17].

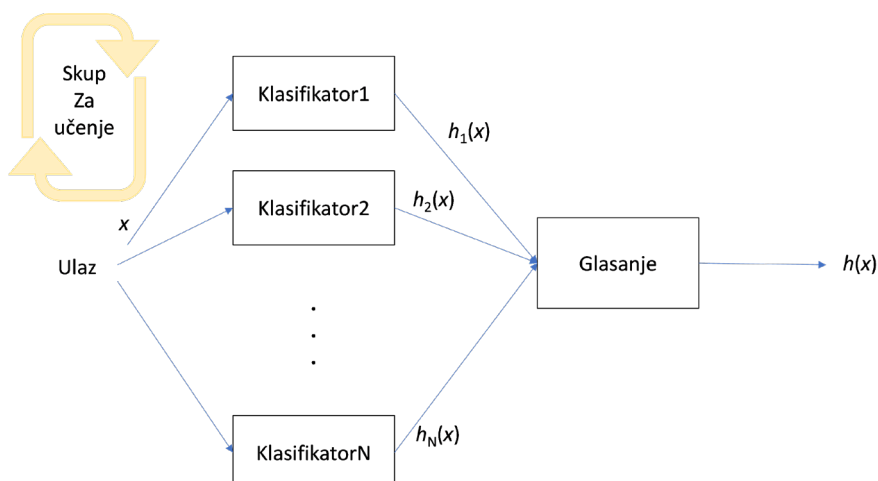
Iako pakiranje stvara slučajne „pakete“ podataka, ponekad ti podaci neće biti dovoljno slučajni i osnovni klasifikatori će biti visoko korelirani. Pristup metode pakiranja gdje se kao osnovni klasifikatori koriste stabla odluke i gdje im se veća raznolikost postiže slučajnim odabirom podskupa značajki koje se promatraju naziva se algoritam „slučajne šume“ (engl. *random forest*) [2].

3.3. POJAČAVANJE

3.3. BOOSTING

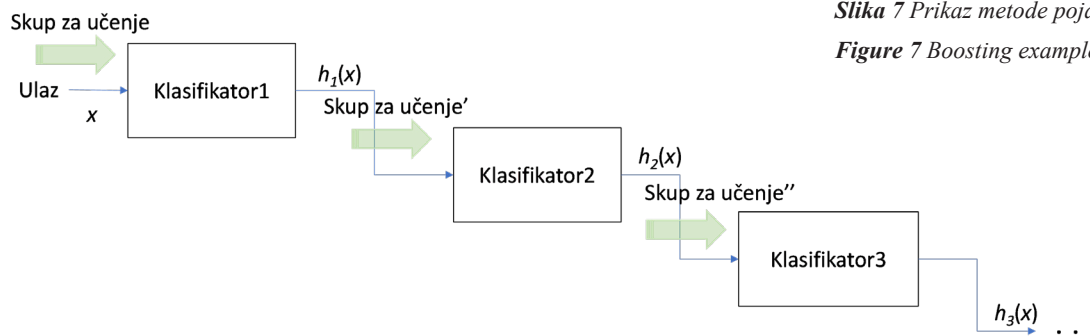
Cilj pojačavanja kao vrste ansambl metode je poboljšanje osnovnih klasifikatora tamo gdje ne daju dobre rezultate [2] [3]. Za razliku od prethodnih metoda, pojačavanje je slijedni postupak u kojem je svaki klasifikator kreiran na temelju grešaka prethodnog klasifikatora (Slika 7). Počinje se s učenjem jednostavnog prediktora i procjenom njegove greške s ciljem da sljedeći prediktor ispravno radi s podacima na kojima je prethodni pogriješio.

Jedan od načina da novi prediktor ispravi onog prethodnog je da uzme u obzir podatke za učenje na kojima je prethodnik dao slabije rezultate. To rezultira novim prediktorima koji su sve bolje



Slika 6 Prikaz postupka pakiranja. Svaki klasifikator treniran je slučajnim podskupom podataka te se konačni rezultat donosi glasanjem. [16]

Figure 6 Bootstrap aggregation process. Each classifier is trained with a random subset of data and the final result is determined by voting. [16]



Slika 7 Prikaz metode pojačavanja. [2]

Figure 7 Boosting example. [2]

prilagođeni slučajevima na kojima je prethodni prediktor griješio. Ovo je tehnika koja se zove *adaptivno pojačanje* (engl. *AdaBoost*) [2]. Na primjer, za izgradnju AdaBoost klasifikatora prvi osnovni klasifikator kao što je stablo odluke uči se na početnom skupu podataka, a relativne težine pogrešno klasificiranih instanci podataka za učenje se u daljnjim stupnjevima povećavaju. Drugi osnovni klasifikator se uči pomoću ažuriranih težinskih vrijednosti podataka skupa za učenje i daje svoja predviđanja nakon čega se težinske vrijednosti opet ažuriraju itd. Konačna predikcija se radi težinskim glasanjem po svim osnovnim klasifikatorima.

Svaki algoritam učenja prikladniji je za neke vrste problema od drugih i obično ima puno različitih parametara i konfiguracija za prilagodbu prije nego što postigne optimalne performanse na skupu podataka. AdaBoost sa stablima odluke kao slabim klasifikatorima često se smatra najboljim klasičnim klasifikatorom [4].

4. ZAKLJUČAK

4. CONCLUSION

Cilj rješavanja problema strojnim učenjem je pronaći jedan model koji će najbolje predvidjeti željene rezultate. Umjesto fokusiranja na jedan model, ansambl metode uzimaju u obzir više modela i njihovih rezultata koje kombiniraju u konačni rezultat. Takav pristup u praksi obično daje bolje rezultate kada postoji raznolikost među modelima. Ovaj je rad opisao tri osnovna algoritma klasifikacije i tri modela ansambl metode strojnog učenja gdje su metode pakiranja i pojačavanja u pravilu homogenog tipa kada im osnovni klasifikatori koriste isti algoritam klasifikacije, dok je slaganje heterogenog tipa jer koristi različite algoritme klasifikacija u fazi

učenja. S druge strane, kod pakiranja i slaganja klasifikatori u fazi učenja rade neovisno jedan od drugoga, tj. u paraleli, dok kod pojačavanja slabiji klasifikatori rade slijedno, tj. na adaptivan način u međusobnoj ovisnosti [18]. Kod pakiranja se raznolikost postiže korištenjem različitih skupova ili poduzoraka iz istog skupa podataka. Fokus budućeg rada biti će na ansambl metodama u području dubokog učenja koje se koriste u svrhu poboljšanja robusnosti i generalizacije modela. Njihovom primjenom može se smanjiti problem prenaučivosti te pojednostaviti učenje modela.

5. REFERENCE

5. REFERENCES

- [1.] A. Lindholm, N. Wahlström, F. Lindsten i T. B. Schön, *Machine Learning: A First Course for Engineers and Scientists*, Cambridge University Press, 2022, ISBN-10: 1108843603.
- [2.] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly Media, 2019, ISBN-10: 1492032646.
- [3.] T. Dietterich, »Ensemble Methods in Machine Learning,« *Computer Science*, svez. 1857, 2000.
- [4.] R. Caruana i A. Niculescu-Mizil, »An Empirical Comparison of Supervised Learning Algorithms,« u *Proceedings of the 23rd International Conference of Machine Learning*, 2006.
- [5.] R. Maclin i D. Opitz, »Popular Ensemble Methods: An Empirical Study,« *Journal of Artificial Intelligence Research*, svez. 11, pp. 169-198, 1999.
- [6.] V. Kolobara, Rješavanje klasifikacijskih problema evolucijom stabala odluke, svez. 26, Fakultet elektrotehnike i računarstva,

- 2016, pp. 135-159.
- [7.] P. Szufel i M. Jakubczyk, »Modeling Decision Trees with SilverDecisions,« SGH Publishing House, 2022. [Mrežno]. Available: https://szufel.pl/Modeling_Decision_Trees_with_SilverDecisions_2022_edited_by_P_Szufel_M_Jakubczyk_B_Kaminski.pdf. [Pokušaj pristupa 3 2023].
- [8.] C. Cortes i V. Vapnik, »Support-vector networks,« Machine Learning, svez. 20, br. 3, pp. 273-297, 1995.
- [9.] B. E. Boser, I. M. Guyon i V. N. Vapnik, »A training algorithm for optimal margin classifiers,« u Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, 1992.
- [10.] T. Joachims, »Text categorization with support vector machines: Learning with many relevant features,« u Proceedings of the 10th European Conference on Machine Learning, 1998.
- [11.] D. Tax i R. Duin, »Support vector data description,« Machine Learning, svez. 54, br. 1, pp. 45-66, 1999.
- [12.] M. A. Nielsen, »Neural Networks and Deep Learning,« Determination Press, 2015.
- [13.] Y. LeCun, Y. Bengio i G. Hinton, »Deep learning,« Nature, svez. 521, pp. 436-444, 2015.
- [14.] D. Silver, A. Huang, C. Maddison i et. al., »Mastering the game of Go with deep neural networks and tree search,« Nature, svez. 529, pp. 484-489, 2016.
- [15.] T. Brown i a. et., »Language Models are Few-Shot Learners,« u Advances in Neural Information Processing Systems, 2020.
- [16.] S. Raschka, »Bagging – building an ensemble of classifiers from bootstrap samples,« Safari Books Online, [Mrežno]. Available: <https://www.safaribooksonline.com/library/view/python-machine-learning/9781783555130/ch07s04.html>. [Pokušaj pristupa 2018 4].
- [17.] »Strojno učenje, TakeLab, Fakultet elektrotehnike i računarstva,« [Mrežno]. Available: <https://strojnoucenje.takelab.fer.hr>. [Pokušaj pristupa 3 2023].
- [18.] J. Rocca, »Ensemble methods: bagging, boosting and stacking,« [Mrežno]. Available: <https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205>. [Pokušaj pristupa 3 2023].

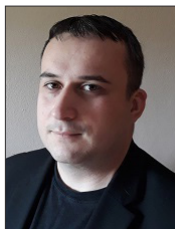
AUTORI · AUTHORS

• **Aleksandar Stojanović** – je viši predavač na Tehničkom veleučilištu u Zagrebu i izvodi nastavu na predmetima iz područja programiranja i algoritama i struktura podataka.

Godine 1996. diplomirao je informacijske i komunikacijske znanosti na Filozofskom fakultetu sveučilišta u Zagrebu, a 1999. godine magistrirao računarstvo u SAD-u na sveučilištu Midwestern State University te je radio kao programski inženjer u području telekomunikacija, financija i energetike. Godine 2019. doktorirao je na Filozofskom fakultetu sveučilišta u Zagrebu s disertacijom pod naslovom "Metoda automatske detekcije naglašenih riječi u zvučnom zapisu". Područja interesa su mu integracija podataka, razvoj uniformnih sučelja za heterogene izvore podataka te primjena evolucijskih algoritama u rješavanju problema. Autor je knjige "Elementi računalnih programa s primjerima u Pythonu i Scali".

Korespondencija · Correspondence

aleksandar.stojanovic@tvz.hr



• **Željko Kovačević** - Nepromjenjena biografija nalazi se u časopisu Polytechnic & Design Vol. 10, No. 4, 2022

Korespondencija · Correspondence

zeljko.kovacevic@tvz.hr

• **Danko Ivošević** -

Diplomirao je 1999. i doktorirao 2013. godine na Fakultetu elektrotehnike i računarstva. Od 2019. godine zaposlen je na Tehničkom veleučilištu u Zagrebu na radnom mjestu

predavača gdje sudjeluje na kolegijima Algoritmi i strukture podataka, Objektno orijentirano programiranje, Napredne tehnike programiranja, Objektno orijentirani razvoj programa i Napredna teorija algoritama i sustava na prijediplomskom i diplomskom studiju te kao mentor studentima na završnim i diplomskim radovima. Prije toga je nakon završetka studija bio zaposlen na više radnih mjesta na poslovima inženjera programske potpore nakon čega je bio zaposlen na mjestu znanstvenog novaka i poslijedoktoranda na Fakultetu elektrotehnike i računarstva gdje je također sudjelovao u provedbi nastave na više kolegija preddiplomskog i diplomskog studija. Objavio je više znanstvenih i stručnih radova. Član je strukovne udruge IEEE od 2007. godine gdje je u razdoblju od 2015. do 2022. obnašao službe predsjednika i dopredsjednika Odjela za računarstvo Hrvatske sekcije.

Korespondencija · Correspondence

danko.ivošević@tvz.hr