Klasifikacija mina i kamenja iz sonar dataseta

Alen Andrašek, Monika Majstorović, Luka Valenta

***Sadržaj*—U ovom radu opisujemo odabir i treniranje klasifikatora za problem klasifikacije mina i kamenja u podacima dobivenim od sonara. Koristimo poznati sonar dataset preuzet s UCI repozitorija za strojno učenje. Treniramo SVM, k-NN i Random Forest modele. Rezultate uspoređujemo međusobno te s rezultatima u literaturi. Za mjeru performansa koristimo točnost. Dobivamo da SVM ima najbolji prosječni performans u odnosu na k-NN i Random forest koje smo mi trenirali, ali nije bolji od performansa k-NN-a iz literature.**

I. UVOD

SONAR (**So**und **Na**vigation and **R**anging) je tehnika koja se koristi za navigaciju, detekciju drugih objekata ili komunikaciju s drugim objektima na ili ispod površine vode pomoću zvuka. Najčešće ju koriste podmornice za detekciju podvodnih mina, drugih podmornica i neprijateljskih brodova te ribari za detekciju jata riba i dubinu mora. Termin sonar odnosi se također na opremu koja generira i prima zvuk pri korištenju tehnike SONAR. Razlikujemo pasivni sonar, koji prima zvukove drugih objekata, te aktivni sonar koji emitira zvučne impulse te osluškuje njihovu jeku kada se odbiju od objekt.

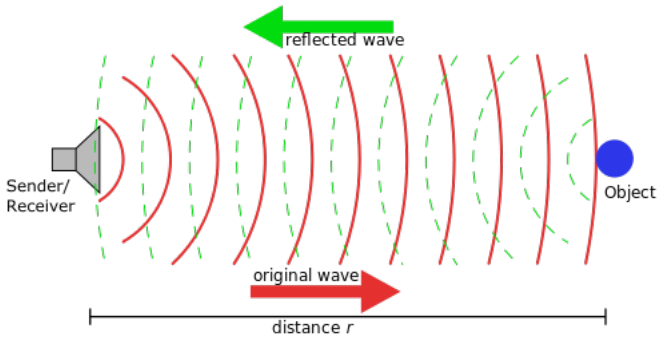

Fig. 1: Ilustracija rada aktivnog sonara

Na Fig. 2 i Fig. 3 primjećujemo da su jeke zvuka koji se odbije o minu i o kamen prilično međusobno slične iz čega se razvila potreba za načinom raspoznavanja ta dva objekta. U daljnjem tekstu opisat ćemo treniranje klasifikatora upravo za taj problem.

II. PRIKUPLJANJE PODATAKA

Preuzeli smo skup podataka naziva „*Connectionist Bench (Sonar, Mines vs. Rocks) Data Set*" (skraćeno, sonar dataset) iz UCI repozitorija za strojno učenje [11]. Podaci su skupljeni tako što su na pjeskovito morsko dno stavljeni metalni cilindar i kamenje otprilike cilindričnog oblika, oboje duljine oko 1.52 m (5 ft) te je pomoću aktivnog sonara emitiran širokopojasni linearni chirp (frekvencija pulsa mijenja se linearno s vremenom). Jeka je prikupljena s udaljenosti 10 m te iz kuteva raspona do 90° za cilindar i raspona do 180° za kamen.

Od 1200 primljenih zvukova, odabrano ih je 208 koji su imali omjer šuma i signala između 4 dB i 15 dB. Od tih 208 primljenih signala, njih 111 pripada metalnom cilindru, a 97 kamenju.

Daljnjom spektralnom analizom signala te normalizacijom dobivenih podataka svaki od 208 signala predstavljen je konačno 60-dimenzionalnim vektorom u kojem svaka komponenta poprima vrijednosti između 0.0 i 1.0. Svaki od tih 60 brojeva predstavlja energiju unutar određenog frekvencijskog pojasa, integriranu tijekom određenog vremenskog razdoblja. U ovom skupu podataka nema vrijednosti koje nedostaju.

III. DOSADAŠNJA ISTRAŽIVANJA

Sonar dataset često je korišten za razne edukacijsko-demonstrativne svrhe u području strojnog učenja te za benchmark novih modela i algoritama. U ovom poglavlju dat ćemo kratak osvrt na tri istraživanja provedena u zadnjih nekoliko godina u kojima se koristio sonar dataset.

A. Gray wolf classifier (GWC)

Trojac s Iranskog sveučilišta znanosti i tehnologije u Teheranu, M.R. Mosavi, M. Khishe i A. Ghamgosar, (2016.) rješavao je problem optimiziranja treniranja neuralnih mreža pomoću algoritma sivog vuka (eng. *Gray Wolf Optimization, GWO*) [1].

GWO je meta-heuristika inspirirana hijerarhijskom organizacijom čopora i načinom lova sivog vuka. Matematički model za ovo biološko ponašanje predstavljen je 2014. godine (iste godine kada je i rad [1] primljen na recenziju). Od tada se GWO zbog dobrog balansiranja eksploracije (globalna pretraga) i eksploatacije (lokalna pretraga) te lakoće implementacije koristi u raznim poljima.

Klasifikator baziran na algoritmu sivog vuka (GWC) testirali su na tri skupa podataka: Iris, Lenses i Sonar.

Dimenzionalnost Sonar dataseta smanjili su sa 60 na 9 koristeći analizu glavnih komponenti (PCA) zbog jednostavnosti i male računalne kompleksnosti te metode. Odlučili su se na PCA bez nadzora jer je ta varijanta pokazala bolje rezultate na stvarnim skupovima podataka.

Rezultate GWC-a uspoređivali su s rezultatima klasifikatora baziranih na algoritmu roja čestica (eng. *Particle Swarm Optimization, PSO*), algoritmu gravitacijske pretrage (eng. *Gravitational Search Algorithm, GSA*) te hibridu prethodna dva: PSOGSA. Promatrali su classification rate, convergence rate i sposobnost izbjegavanja zaglavljivanja u lokalnom minimumu.

Zaključili su da GWC daje najbolje rezultate u sva tri promatrana aspekta na sva tri skupa podataka.

B. Online multiple kernel klasifikacija (OMKC)

Online učenje metoda je strojnog učenja u kojoj podaci postaju dostupni redoslijedom i koriste se za ažuriranje našeg najboljeg prediktora za buduće podatke u svakom koraku.

U [3] autori R. K. Jade, L. K. Verma i K. Verma klasificiraju sonar dataset koristeći online multiple kernel klasifikaciju.

OMKC koristi online učenje da bi naučila predikciju koja se temelji na kernelima tako što odabire podskup unaprijed definiranih kernela. Algoritam koji su koristili za online učenje je kombinacija perceptron algoritma koji uči klasifikator za dani kernel i hedge algoritma koji kombinira klasifikatore s linearnim težinama.

Razvili su stohastičke strategije odabira koje slučajno odabiru podskup kernela za ažuriranje kombinacije i modela, čime su poboljšali učinkovitost učenja.

Testiranja su provedena na sonar datasetu primjenjujući sljedeće metode:

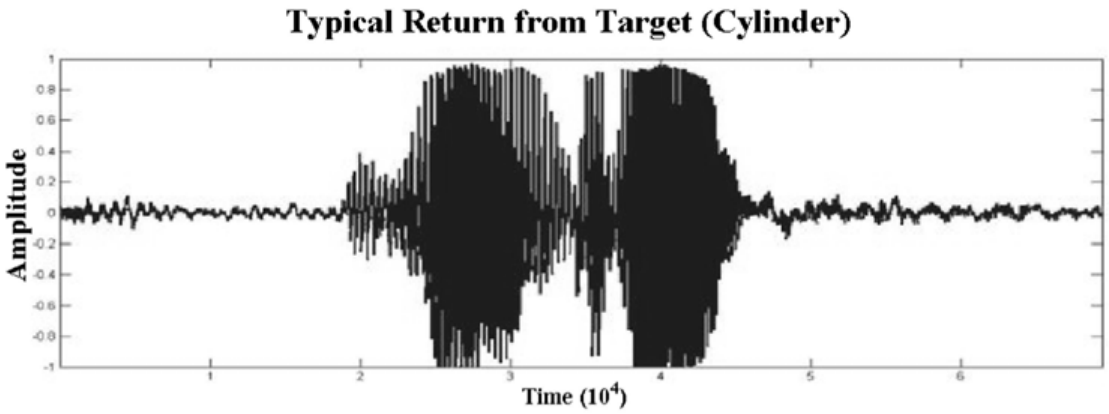

Fig. 2: Tipičan povratni signal sonara na mini

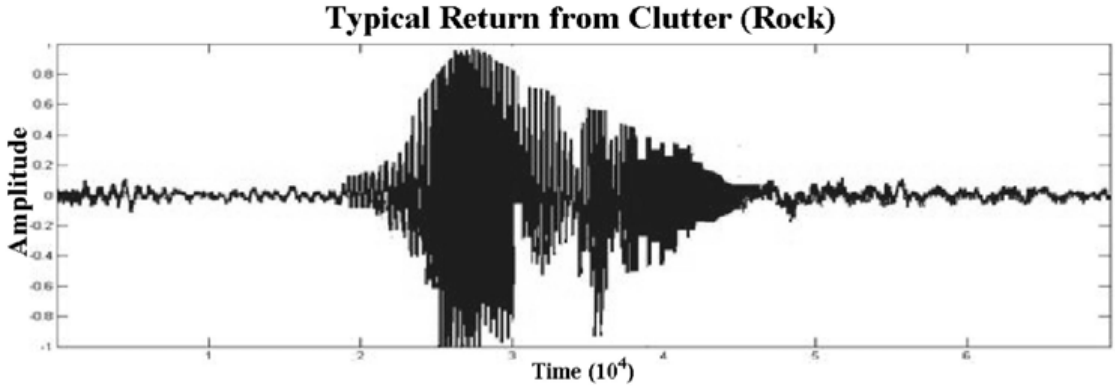

Fig. 3: Tipičan povratni signal sonara na kamenu

- Neuralna mreža
- Perceptron
- Perceptron (uniforman)
- Online učenje
- OMKC s deterministčkim ažuriranjem i deterministčkom kombinacijom algoritmama
- OMKC s stohastičkim ažuriranjem i deterministčkom kombinacijom algoritmama
- OMKC s deterministčkim ažuriranjem i stohastičkim kombinacijom algoritmama
- OMKC s stohastičkim ažuriranjem i stohastičkim kombinacijom algoritmama

Neuralnim mrežama dobivena je preciznost od 89% na trening setu i 83% na setu za testiranje. Najmanje greške dobivene su koristeći OMKC s deterministčkim ažuriranjem i deterministčkom kombinacijom algoritmama (24.74%) i OMKC sa stohastičkim ažuriranjem i deterministčkom kombinacijom algoritmama (24.83%). Sve kombinacije koristeći OMKC imale su grešku ispod 26%.

Ova eksperimentalna studija tako se pokazala kao obećavajuća izvedba predloženih algoritama za OMKC u učinkovitosti učenja i točnosti predviđanja.

C. Ansambli neuronskih mreža

U članku [4] proučava se korištenje ansambl tehnike bagging (skraćeno od bootstrap aggregating) na klasifikatore dobivene neuronskim mrežama. Ansambl je skup nezavisno treniranih klasifikatora čija predviđanja se kombiniraju nekim statističkim metodama. Proučava se efekt na sonar i ionosphere datasetu.

Jedna neuronska mreža bi mogla dati nezadovoljavajuće rezultate zbog mnogih razloga. U medicinskim, financijskim i sl. primjenama se zahtjeva najveća točnost, ali treniranjem više mreža i biranjem najbolje, malo je vjerojatno dobivanje znatno veće efikasnosti.

Umjesto toga, moguće je koristiti bagging - istrenirati veći broj klasfikatora te ih spojiti u jedan konačni, najbolji klasifikator. Tako je konačni klasifikator (težinsko) usrednjenje više njih. Bagging je samo jedna od metoda učenja ansamblima.

Klasifikator korišten u članku je BNPP (Back Propagation Neural Network). Istražena je točnost: jedne mreže zasebno, usrednjene tri mreže (3-bagged BNPP) i 25 (25-bagged) usrednjenih mreža te 25 usrednjenih mreža s PCA (Principle Component Analysis). (Implementirano u Matlab 7). Training set je bio 70% podataka (66 kamena i 80 mina).

Točnosti pojedinih algoritama za Sonar dataset:

- BNPP: 88.709%,
- 3-bagged BNPP: 80.645%,
- 25-bagged BNPP: 93.54%,
- 25-bagged BNPP, PCA: 91.935%.

Može se uočiti kako usrednjavanje malog broja je oštetilo efektivnost klasifikatora, ali usrednjavanjem većeg broja uočavamo bitno poboljšanje. Vidimo da je korištenje PCA nešto smanjilo točnost za ovu metodu i autori zaključuju da se vjerojatno ne isplati koristiti.

IV. PRISTUP PROBLEMU

Sonar dataset je malen pa je teže izbjeći overfitting, a outlieri i buka (eng. *noise*) imaju veći utjecaj nego kod većih datasetova. U slučaju malih datasetova preporučuje se korištenje jednostavnijih

modela i modela s manje parametara [6]. Uzimajući u obzir prethodne tvrdnje, odlučili smo koristiti *SVM (Suppport Vector Machine)* i *k-NN (k-Nearest Neighbour)* i *Random Forest*. Dobivene rezultate usporedit ćemo međusobno te i s rezultatima dobivenima u [5]. Dodatno, prokrenut ćemo i `autosklearn`.

A. Smanjenje dimenzionalnosti

Interpretabilnost featurea nam nije bitna pa smo radili smanjenje dimenzionalnosti koristeći *PCA (Principle Component Analysis)* na cijelom datasetu. Broj dimenzija smanjili smo s početnih 60 na 17 uz zadržavanje 95% varijance.

B. Podjela primjera

Kod malih skupova podataka podaci testiranja mogu jako ovisiti o odabranom testnom skupu. Budući da imamo malo podataka, veći testni skup nije opcija jer ne bi ostalo dovoljno podataka za treniranje modela. Kako bismo ipak smanjili ovisnost rezultata o testnom skupu podjelili smo dataset na 10 disjunktnih podskupova. Ta podjela nije potpuno nasumična jer smo koristili `StratifiedKFold`. Koristili smo stratifikaciju jer klase u datasetu nisu potpuno balansirane (mina ima više nego kamenja). Stratifikacijom smo osigurali da svaki podskup dobro reprezentira cijeli skup s obzirom na postotak mina i kamenja u njemu. Iako disbalans u sonar datasetu nije velik, u slučaju malog dataseta kao što je sonar dataset stratifikacija je preporučena kao mjera osiguranja. [8]

Od tih 10 podskupova konstruirali smo 10 parova skupova za treniranje i testiranje tako da je 9 podskupova činilo skup za treniranje, a 1 skup za testiranje. Na taj način svaki primjer je u nekom skupu za testiranje i na prosječnim rezultatima možemo bolje usporediti točnost različih algoritama.

Na svakom od skupova za testiranje 3-fold cross-validacijom odredili smo hiperparametre i istrenirali model dan najboljim hiperparametrima. Zatim smo testirali istreniran model na pripadnom skupu za testiranje.

Promatrali smo rezultate dobivene na svakom paru skupova za treniranje i testiranje kao i prosječne rezultate.

C. Metoda potpornih vektora

SVM smo koristili s *RBF (Gaussian Radial Basis Function)* kernelom. Ovaj kernel često je korišten u primjenama, a odabrali smo ga umjesto linearnog kernela zbog smjernica Andrewa Nga na Coursera Machine Learning kursu (Week 7).

Kod SVM-a koristili smo 3-fold cross-validation s *Grid Search* i *Randomized Search*. Očekujemo da će rezultati s *Randomized searchom* biti bolji zbog zaključaka u [10]. Za Grid Search su korišteni parametri C i γ dobiveni pomoću `numpy.logspace(-1, 3, 100)` odnosno `numpy.linspace(0.0001, 10, 100)`, a za Randomized Search $C \in [1.0, 1000.0]$ i $\gamma \in [0.001, 1.0]$ uz 100 iteracija.

D. k-NN

Kod k-NN, pokazano je u [11] da odabir metrike uvelike može utjecati na poboljšanje rezultata na različitim klasifikacijskim problemima. Također, u istoj studiji pokazalo se da k-NN s pravilno odabranom metrikom može dati performans bolji od SVM-a za isti klasifikacijski problem. Mi smo promatrali Euklidsku, Manhattan, Čebiševljevu metriku. Koristili smo težinski k-NN koji pridaje veću vanost bližim susjedima nego daljim. Kao i kod SVM, tražili smo parametre s *Grid Search* cross-validation i to među spomenute 3 metrike i k izmeu 1 i 10. Dobrim k se pokazao 1 ili 3, dok je najbolja metrika pretežito bila Čebiševljeva.

E. Random Forest

Kod *Random Forest* klasifikatora određivali smo parametre *n_estimators* (broj stabala u šumi) i *max_features* (broj značajki koje pojedina stabla procesiraju). Nije se mogla uočiti pravilnost među najboljim parametrima u pojedinim modelima. Vidimo da je konačna točnost znatno slabija u usporedbi s drugim algoritmima.

F. `autosklearn`

Također, koristili smo i `autosklearn` kako bismo vidjeli koliko dobri rezultati se mogu postići na taj način i usporedili njegovu točnost s točnošću SVM-a, k-NN i Random Foresta.

`autosklearn` je baziran na *scikit-learn*. Koristi 15 klasifikatora, 14 metoda za predprocesiranje značajki i 4 metode predprocesiranja podataka, što dovodi do strukturiranog prostora hipoteza sa 110 hiperparametara. Ovaj sistem poboljšava postojeće AutoML metode tako što automatski uzima u obzir prošlu uspješnost na sličnim skupovima podataka i konstruiranjem ansambala od modela evaluiranih tijekom optimizacije. [14]

`AutoSklearnClassifier` smo izvršili na istih 10 parova skupova za treniranje i testiranje uz razliku da na skupovima na kojima je on pokrenut nije prethodno napravljen PCA. Na svakom od 10 parova izvršavanje `AutoSklearnClassifier` je trajalo po sat vremena.

V. KORIŠTENI ALATI

Programirali smo u Python-u koristeći Jupyter Notebook okruženje i standardne Python biblioteke za strojno učenje kao što su `sklearn` i `autosklearn` dok smo za prikaz rezultata koristili `pandas` i `matplotlib`.

VI. REZULTATI

Najboljim se u prosjeku pokazao SVM s *Randomized Searchom* s 85.974%, dok je *Random Forest* s 79.1883% pokazao najlošiji prosječni preformans. Performans svih treniranih modela prikazan je u Tablici II. te u Fig. 4 i Fig. 5. Također prikazane su i matrice konfuzije za SVM (*Randomized search*), *k-NN*, *Random forest* i `autosklearn`.

Rezultate uspoređujemo i s rezultatima u [5] što je prikazano u Tablici 2. Autorima članka [5] bio je cilj usporediti performans SVM-a s dotad najboljim performansama raznih drugih metoda strojnog učenja, među kojima su i *k-NN* i *Random Forest* na raznim datasetovima uključujući i sonar dataset. Naš SVM pokazuje bolji prosječni performans nego SVM u [5], ali nije bolji od *k-NN*-a, koji je ujedno pokazao i bolje performanse (87.31%) od svih drugih modela u [5] na sonar datasetu.

%	SVM	k-NN	Random Forest
Naši rezultati	85.974	83.0931	79.1883
Rezultati u [5]	84.56	87.31	83.8

TABLE I: Usporedba rezultata s rezultatima u članku [5]

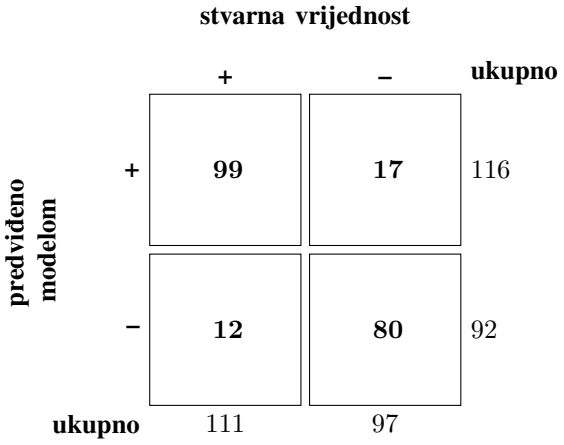

Fig. 6: Matrica konfuzije za SVM s Randomized Search (suma vrijednosti po podjelama)

%	SVM (GS)	SVM (RS)	k-NN (Čebišev)	Random Forest	autosklearn
1	90.9091	95.4545	95.4545	81.8182	77.2727
2	76.1905	85.743	71.4286	76.1905	76.1905
3	85.7143	90.4762	76.1905	80.9524	95.2381
4	85.7143	76.1905	71.4286	76.1905	90.4762
5	85.7143	95.2381	90.4762	76.1905	90.4762
6	90.4762	90.4762	76.1905	80.9524	90.4762
7	76.1905	76.1905	85.7143	85.7143	76.1905
8	90	85	80	75	70
9	85	85	85	85	85
10	80	80	80	65	65
AVG	84.5909	85.974	83.0931	79.1883	81.632

TABLE II: Točnost (%) dobivena raznim algoritmima na 10 train–test parova

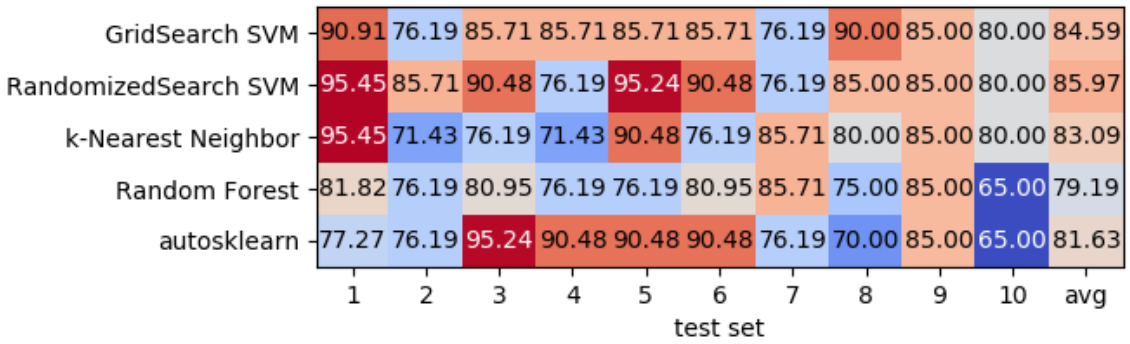

Fig. 4: Točnost (%) testiranih algoritama na testnim skupovima

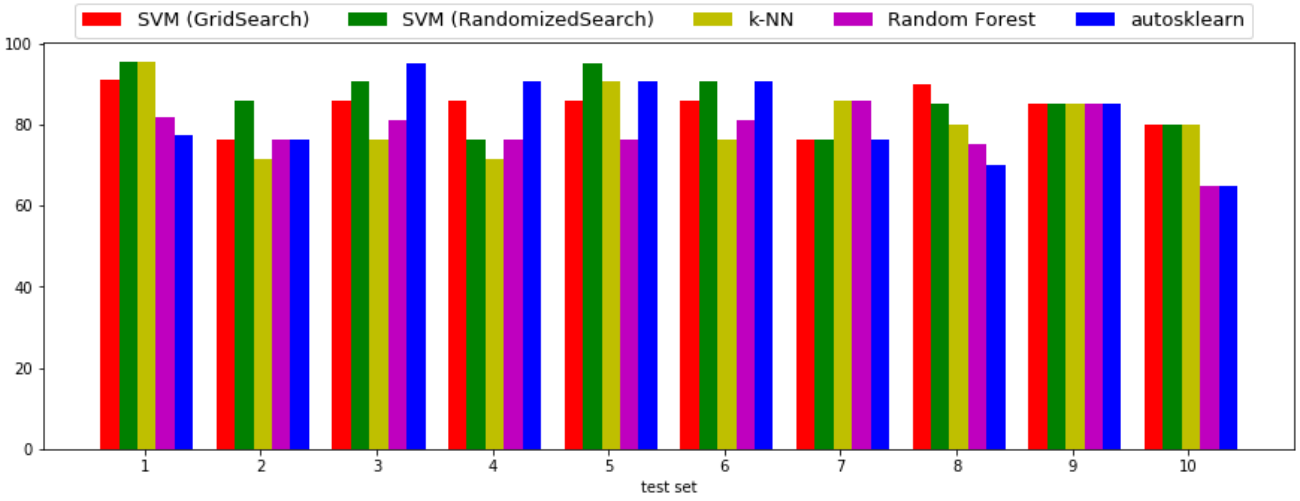

Fig. 5: Točnost (%) testiranih algoritama na testnim skupovima

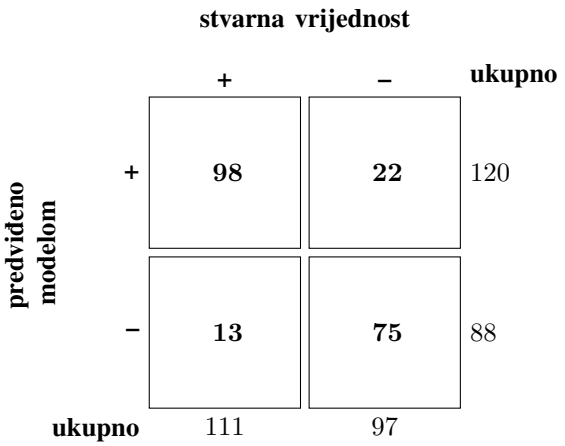

Fig. 7: Matrica konfuzije za k-NN s Čebiševljevom metrikom (suma vrijednosti po podjelama)

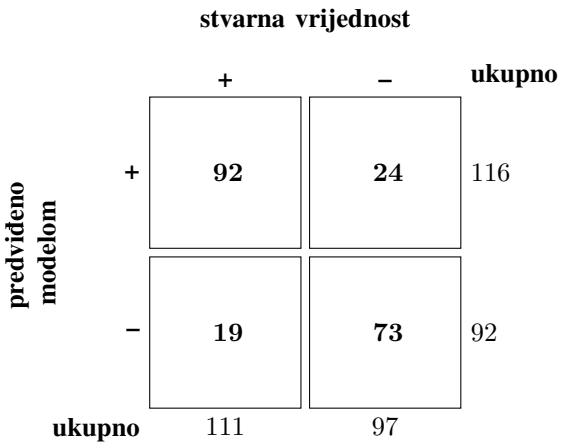

Fig. 8: Matrica konfuzije za Random Forest (suma vrijednosti po podjelama)

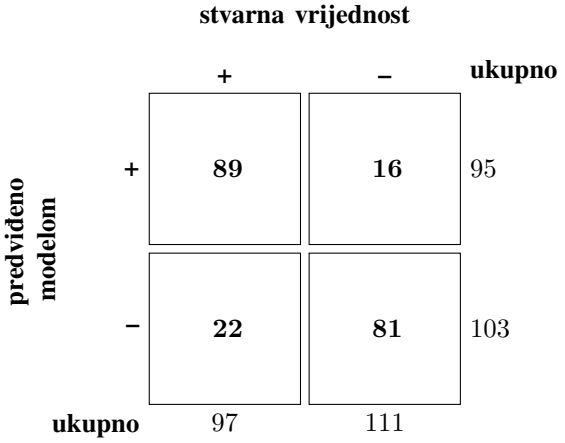

Fig. 9: Matrica konfuzije za `autosklearn` (suma vrijednosti po podjelama)

VII. ZAKLJUČAK

U prosjeku je *SVM* s cross-validacijom koristeći `RandomizedSearch` dao najveću točnost, međutim ta razlika u odnosu na *k-NN* i SVM s cross-validacijom koristeći `GridSearch` je vrlo mala. *Random Forest* i `autosklearn` su u prosjeku bili nešto losiji premda je i ta razlika u točnosti ispod 7 odnosno 5%.

Došlo je do izražaja i to koliko rezultati pojedine metode ovise o skupu za testiranje. Tako je na pojedinim testnim skupovima `autosklearn` je dao bolje rezultate nego SVM i k-NN, ali zbog znatno loših rezultata na drugima imao je nižu prosječnu točnost. S druge strane, promatrajući točnost Random Foresta na svim testnim skupovima utvrđujemo da se ni na kojem testnom skupu nije pokazao kao bolji od svih ostalih dok je na više testnih skupova podbacio pa na temelju naših rezultata zaključujemo je lošiji izbor za klasifikaciju podatak iz sonar dataseta.

Smatramo da je ovisnost rezultata o testnom skupu potvrdila potrebu za testiranjem na više skupova za testiranje. Zaključujemo da je od testiranjih algoritama SVM u prosjeku najbolji za sonar dataset, ali u na pojedinim primjerima će k-NN biti bolji.

VIII. DALJNI RAD

U daljnjim istraživanjima bilo bi korisno bolje utvrditi utjecaj načina podjele dataseta na train/test/validation skupove na prosječni performans modela te pronaći bolju podjelu za ovaj dataset od podjele koju smo mi koristili.

REFERENCES

[1] M.R. Mosavi, M. Khishe, A. Ghamgosar; Classification of Sonar Dataset Using Neural Network Trained by Grey Wolf Optimization; Iran University of Science and Technology, Teheran; Iran, 2016.

[2] R.P. Gorman, T. J. Stejnowski; Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets; 1987.

[3] R. K. Jade, L. K. Verma, K. Verma; Classification using Neural Network and Support Vector Machine for Sonar dataset; International Journal of Computer Trends and Technology; Vol. 4, Issue 2; pg 116-119; 2013.

[4] H. T. Hassan, M. U. Khalid, K. Imran; Intelligent Object and Pattern Recognition using Ensembles in Back Propagation Neural Network; International Journal of Electrical & Computer Sciences IJECS-IJENS; Vol. 10, No. 6; pg 52-59; 2010.

[5] D. Meyer, F. Leisch, K. Hornik; Benchmarking Support Vector Machines; 2002.

[6] https://medium.com/rants-on-machine-learning/what-to-do-with-small-data d253254d1a89 (Zadnje pristupljeno: 30. travnja 2018.)

[7] https://pdfs.semanticscholar.org/d6dc/df86df3ece94c2c5effe205d105c561ed5eb.pdf (Zadnje pristupljeno: 30. travnja 2018.)

[8] https://stats.stackexchange.com/questions/117643/why-use-stratified-cross-validation-why-does-this-not-damage-variance-related-b (Zadnje pristupljeno: 30. travnja 2018.)

[9] http://cs229.stanford.edu/notes/cs229-notes5.pdf (Zadnje pristupljeno: 30. travnja 2018.)

[10] J. Bergstra, Y. Bengio; Random Search for Hyper-Parameter Optimization; Journal of Machine Learning Research 13, pg. 281-305; 2012.

[11] Kilian Q. et al., „Distance Metric Learning for Large Margin Nearest Neighbor Classification", Journal of Machine Learning Research 10, pg. 207-244., 2009.

[12] https://archive.ics.uci.edu/ml/datasets/connectionist+bench+(sonar,+mines+vs.+roc (Zadnje pristupljeno: 30. travnja 2018.)

[13] https://stats.stackexchange.com/questions/61546/optimal-number-of-folds-in-k-fold-cross-validation-is-leave-one-out-cv-always (Zadnje pristupljeno: 30. travnja 2018.)

[14] M. Feuer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, F. Hutter; Efficient and Robust Automated Machine Learning; NIPS 2015.