Učenje iz neuravnoteženih podataka unaprijeđenim postupcima za odabir značajki, preuzorkovanje i izgradnju radijalnih neuronskih mreža

Dudjak, Mario

Doctoral thesis / Disertacija

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:487624

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Mario Dudjak

Učenje iz neuravnoteženih podataka unaprijeđenim postupcima za odabir značajki, preuzorkovanje i izgradnju radijalnih neuronskih mreža

Doktorska disertacija

Osijek, 2022.

Doktorska disertacija izrađena je na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek, Sveučilišta Josipa Jurja Strossmayera u Osijeku.

Mentor: dr. sc. Goran Martinović, redoviti profesor u trajnom zvanju, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, Sveučilište Josipa Jurja Strossmayera u Osijeku

Sumentor: dr. sc. Dražen Bajer, docent, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, Sveučilište Josipa Jurja Strossmayera u Osijeku

Disertacija ima 153 stranice.

Disertacija broj: 89

Povjerenstvo za ocjenu doktorske disertacije:

1. Dr. sc. Emmanuel Karlo Nyarko, izvanredni profesor, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, Sveučilište Josipa Jurja Strossmayera u Osijeku
2. Dr. sc. Domagoj Jakobović, redoviti profesor u trajnom zvanju, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu
3. Dr. sc. Bruno Zorić, docent, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, Sveučilište Josipa Jurja Strossmayera u Osijeku

Povjerenstvo za obranu doktorske disertacije:

1. Dr. sc. Emmanuel Karlo Nyarko, izvanredni profesor, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, Sveučilište Josipa Jurja Strossmayera u Osijeku
2. Dr. sc. Bruno Zorić, docent, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, Sveučilište Josipa Jurja Strossmayera u Osijeku
3. Dr. sc. Domagoj Jakobović, redoviti profesor u trajnom zvanju, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu

Datum obrane doktorske disertacije: 16. rujna 2022. godine.

Zahvale

Zahvaljujem svom mentoru prof. dr. sc. Goranu Martinoviću na podršci i vođenju tijekom poslijediplomskog studija.

Zahvaljujem svojim roditeljima, Zlatku i Blaženki, na bezuvjetnoj podršci i razumijevanju tijekom cijelog mog školovanja. Također zahvaljujem svojim sestrama, Antoniji i Kristini, na vjeri u mene i prijeko potrebnoj motivaciji koju su mi pružale tijekom studija.

Zahvaljujem svojoj zaručnici Margareti na ljubavi i vjeri u mene, kao i na vremenu koje je provela slušajući moje brige tijekom studija te pružajući jedinstvene savjete za njihovo prevladavanje. Zahvaljujem joj što je bila moj oslonac u najzahtjevnijim trenucima studija.

Zahvaljujem svom sumentoru doc. dr. sc. Draženu Bajeru što me uveo u odabrano područje istraživanja i pomogao mi da zakoračim u svijet znanosti. Svojim znanjem, iskustvom i savjetima uvijek je bio spreman pomoći te je značajno pridonio izradi ove disertacije. Zahvaljujem mu na preko 1500 pisanih komentara te na mnoštvu svakodnevnih smjernica tijekom izrade ove disertacije.

Sadržaj

Popis slika

Popis tablica

Popis algoritama

Popis kratica i oznaka

k-NN	Algoritam k-najbližih susjeda
ABC	Algoritam umjetne kolonije pčela
A_2	Algoritam za traženje čvora koji se dodaje prethodno treniranoj RBFN
A_1	Algoritam za treniranje radijalnih neuronskih mreža
A	Arhiva rješenja u predloženom proširenju bio-inspiriranih omotača
N_M	Broj manjinskih primjeraka u skupu podataka
m	Broj oznaka klasa u skupu podataka
N	Broj primjeraka u skupu podataka
q	Broj stvorenih primjeraka za svaki manjinski primjerak u algoritmu SMOTE
NFEs	Broj vrednovanja funkcije cilja
d	Broj značajki u skupu podataka
c	Broj čvorova u skrivenom sloju radijalne neuronske mreže
$\mathbf{z}$	Centar radijalne funkcije
DE	Diferencijalna evolucija
d_{perf}	Euklidska udaljenost od savršenog klasifikatora
f	Funkcija cilja
GA	Genetski algoritam
G_{mean}	Geometrijska sredina osjetljivosti i specifičnosti
ROC	Krivulja operativnih karakteristika
SE	Kvadrat greške

$\mathcal{N}_k$	k-susjedstvo promatranog manjinskog primjerka u algoritmu SMOTE
$\text{NFEs}_{\max}$	Maksimalni broj vrednovanja funkcije cilja
GD	Metoda gradijentnog spusta
SVM	Metoda potpornih vektora
ASM	Mjera ASM
F1	Mjera F1
GNB	Naivan Bayesov klasifikator
$\mathbf{t}$	Najbliži susjed iz većinske klase u predloženom algoritmu preuzorkovanja
$\mathbf{r}_c$	Najbolje rješenje pronađeno treniranjem RBFN s c čvorova u skrivenom sloju
c_{min}	Najmanji dozvoljeni broj čvorova u skrivenom sloju RBFN
c_{max}	Najveći dozvoljeni broj čvorova u skrivenom sloju RBFN
$\mathbf{x}^r$	Nasumično odabrani primjerak iz $\mathcal{M}$
$\mathbf{x}^{r(i)}$	Nasumično odabrani primjerak iz $\mathcal{N}_k$
FS	Odabir značajki
IR	Omjer neuravnoteženosti
PSO	Optimizacija rojem čestica
SFFS	Plutajuća slijedna pretraga unaprijed
MCR	Pogreška klasifikacije
P	Populacija bio-inspiriranog algoritma optimizacije
AUC	Površina ispod ROC krivulje
OS	Preuzorkovanje manjinske klase
$\mathbf{x}^e$	Primjerak za koji prethodno trenirana RBFN daje najveći SE
$\mathcal{X}$	Prostor primjeraka
$\hat{\mathcal{N}}_\kappa$	Prošireno susjedstvo promatranog manjinskog primjerka u predloženom algoritmu preuzorkovanja

ϕ	Radijalna funkcija
RBFN	Radijalna neuronska mreža
FR	Rang u smislu kvalitete izveden pomoću Friedmanova testa ranga
R	Rang u smislu kvalitete izveden pomoću Wilcoxonova testa ranga
R_{red}	Rang u smislu veličine
OVO	Shema dekompozicije jedan-naspram-jedan
OVR	Shema dekompozicije jedan-naspram-ostali
$\mathcal{M}$	Skup manjinskih primjeraka
$\mathcal{Q}$	Skup označenih primjeraka
$\mathcal{L}$	Skup svih oznaka klasa
$\mathcal{T}$	Skup ulaznih podataka za treniranje
$\mathcal{V}$	Skup većinskih primjeraka
SFS	Slijedna pretraga unaprijed
SBS	Slijedna pretraga unazad
MSE	Srednja kvadratna greška
DT	Stablo odluke
TNR	Stopa stvarno negativnih predviđanja
TPR	Stopa stvarno pozitivnih predviđanja
$\mathbf{s}$	Sintetički primjerak stvoren preuzorkovanjem
$\mathcal{N}$	Susjedstvo promatranog manjinskog primjerka u predloženom algoritmu preuzorkovanja
σ	Širina radijalne funkcije
SMOTE	Tehnika sintetičkog preuzorkovanja manjine
w_i^j	Težine veza čvorova skrivenog i izlaznog sloja RBFN
CAC	Točnost klasifikacije
$\mathbf{x}$	Ulazni primjerak

ANNs	Umjetne neuronske mreže
$U(0,1)$	Uniformna slučajna varijabla iz $[0,1]$
N_p	Veličina populacije
k	Veličina susjedstva u algoritmu SMOTE
MLP	Višeslojni perceptron
y_j	Vrijednost na j-tom izlaznom čvoru RBFN
SSE	Zbroj kvadrata greški

1 Uvod

Strojno učenje bavi se proučavanjem i oblikovanjem računalnih algoritama koji grade modele problema iz različitih domena na temelju empirijskih podataka. Ti se modeli koriste za obavljanje raznih zadataka, kao što su predviđanje, donošenje odluka te raspoznavanje uzoraka. Obavljanje ovih zadataka često se svodi na izvođenje klasifikacije dostupnih podataka, što je posebno izraženo u zadacima raspoznavanja uzoraka iz slika [1], zvuka [2], teksta [3], biometrijskih obilježja [4] i drugih izvora. Prema tome, problemi klasifikacije predstavljaju istaknut razred problema strojnog učenja te nije iznenađujuće što je predloženo mnoštvo algoritama koji pomažu kategorizirati dostupne podatke u odgovarajuće klase [5]. Međutim, na učinkovitost svakog od tih algoritama uvelike utječu složenost i sadržaj skupova podataka koji opisuju promatrani problem klasifikacije. Skupovi podataka u pravilu obuhvaćaju razne unutarnje karakteristike koje povećavaju njihovu složenost te općenito narušavaju izvedbu ovih algoritama. Neuravnoteženost klasa, koja se izražava neravnomjernom raspodjelom oznaka različitih klasa u skupu podataka, jedna je od najistaknutijih takvih karakteristika. Zbog štetnosti i rasprostranjenosti ove karakteristike, postoji potreba za postupcima koji ublažavaju njezine negativne učinke i time poboljšavaju učinkovitost algoritama za klasifikaciju pri učenju iz neuravnoteženih podataka. Značajna pažnja u literaturi stoga je posvećena razvoju, primjeni te poboljšanju ovih postupaka.

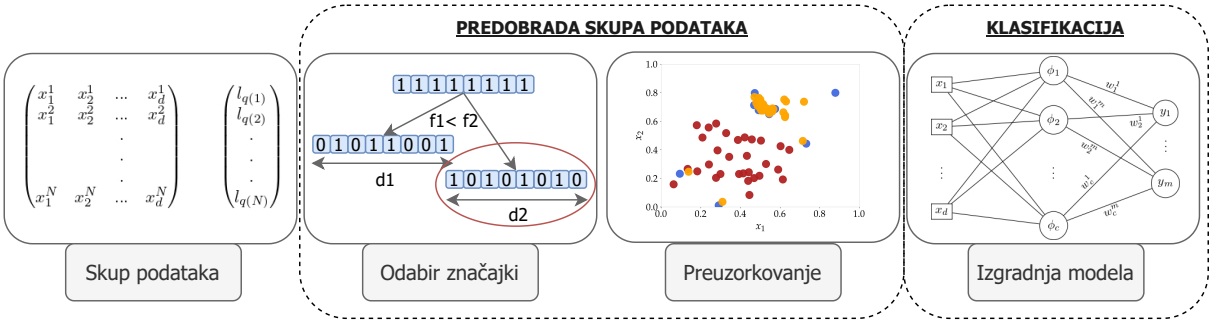
1.1 Motivacija za istraživanje

Opći zadatak algoritama za klasifikaciju jest izgraditi klasifikacijske modele koji primjercima iz skupa podataka dodjeljuju odgovarajuće oznake klasa, što je otežano raznim unutarnjim

karakteristikama skupa podataka koje povećavaju njegovu složenost, jedna od kojih jest neuravnoteženost klasa. U neuravnoteženim skupovima podataka, oznaka jedne klase (tzv. manjinske klase) značajno je slabije zastupljena u odnosu na oznake drugih klasa. Brojni problemi klasifikacije po prirodi su neuravnoteženi, a neki od istaknutijih proizlaze iz područja primjene poput biomedicine, financija, informacijske tehnologije, industrijske proizvodnje te upravljanja sigurnosti [6]. Negativni učinci drugih unutarnjih karakteristika u skupu podataka, poput šuma ili preklapanja klasa, obično su pogoršani neuravnoteženom raspodjelom primjeraka različitih klasa. Ovi se neželjeni učinci uglavnom ogledaju u narušavanju sposobnosti algoritma za klasifikaciju da prepozna manjinsku klasu. Većina standardnih algoritama za klasifikaciju stoga iskazuje pristranost većinskoj klasi pri učenju iz takvih podataka. Ovu poteškoću dodatno otežava činjenica da je u brojnim neuravnoteženim problemima upravo prepoznavanje rijetkih događaja od primarne važnosti, primjerice, u problemima medicinske dijagnostike [7], prepoznavanju izraza lica [8] te otkrivanju upada, grešaka ili prijevara [9].

S obzirom na složenost i zastupljenost problema neuravnoteženosti klasa, nije iznenađujuće da je predloženo mnoštvo pristupa za njegovo ublažavanje. Glavni cilj ovih pristupa jest poboljšati uspješnost prepoznavanja manjinske klase, bez narušavanja opće izvedbe klasifikacije. Postupci predobrade skupova podataka jedini su pristupi koji nastoje izravno izmijeniti strukturu i/ili sadržaj skupova podataka kako bi se ublažio stupanj neuravnoteženosti, dok se drugi pristupi usredotočuju na prilagodbu algoritama za klasifikaciju radi boljeg suočavanja s ovim problemom. Zbog svoje jednostavnosti i učinkovitosti, kao i činjenice da pogoduju izvedbi različitih klasifikatora, postupci predobrade najčešće su korišteni pristupi za ublažavanje problema neuravnoteženosti klasa u literaturi [10–12]. No, valja naglasiti da postoji mnoštvo takvih postupaka, od kojih svaki smanjuje složenost problema na drugačiji način. Stoga nije iznenađujuće da se često nekoliko takvih postupaka zajedno koristi za učinkovitije ublažavanje problema. Međutim, iako ovi postupci značajno doprinose ublažavanju problema, ne mogu ga u potpunosti otkloniti. Prema tome, izbor algoritma za klasifikaciju i dalje je od iznimne važnosti. S obzirom na to da većina algoritama iskazuje pristranost većinskoj klasi pri učenju iz neuravnoteženih podataka, odabir prikladnog algoritma za te probleme nije lak zadatak. Ipak, pri obavljanju tog zadatka moguće je voditi se rezultatima brojnih eksperimentalnih analiza raznih algoritama za klasifikaciju u literaturi [13–15].

U središtu istraživanja ove disertacije dva su različita postupka predobrade neuravnoteženih skupova podataka te jedan algoritam za klasifikaciju kao različiti načini suočavanja s problemom neuravnoteženosti klasa. Postupci predobrade koji se razmatraju jesu odabir značajki (engl. *feature selection*, FS) te preuzorkovanje (engl. *oversampling*, OS). Oni smanjuju složenost koncepta manjinske klase te u pravilu pospješuju njezino prepoznavanje, pri čemu svaki na svoj način doprinosi tom smanjenju složenosti. Odabirom značajki smanjuje se dimenzionalnost problema, a time i udaljenost postojećih manjinskih primjeraka, dok se preuzorkovanjem uravnotežuje broj primjeraka različitih klasa u skupu podataka. Njihova jednostavnost i učinkovitost čine ih prikladnim izborom za ublažavanje problema neuravnote-



Slika 1.1: Uobičajeni tijek učenja iz neuravnoteženih podataka na visokoj razini

ženosti klasa, čemu svjedoče i njihove brojne primjene u tu svrhu [16–19]. Nakon predobrade neuravnoteženih skupova podataka obično se pristupa izgradnji klasifikacijskog modela, što zahtijeva odabir prikladnog algoritma za klasifikaciju, posebno jer postupci predobrade ne mogu u potpunosti ukloniti problem. Radijalna neuronska mreža (engl. *radial basis function network*, RBFN) primjer je algoritma za klasifikaciju koji je po kvaliteti izvedbe nadmašio mnoge druge algoritme na brojnim neuravnoteženim problemima u literaturi [13–15, 20]. Zbog svoje sposobnosti lokaliziranog djelovanja, ovaj je algoritam uspješniji od brojnih drugih algoritama u prepoznavanju primjeraka manjinske klase koji su nerijetko rasprostranjeni u skupu podataka u nekoliko grupa.

Prema svemu navedenom, postupci preuzorkovanja te odabira značajki koji ublažavaju stupanj problema neuravnoteženosti klasa, kao i algoritam radijalne neuronske mreže koji se ističe po sposobnosti prepoznavanja manjinske klase, nameću se kao prikladni pristupi za učinkovito učenje iz neuravnoteženih podataka. U literaturi su predložene različite izvedbe ovih postupaka predobrade, kao i postupka izgradnje klasifikacijskih modela RBFN, no svaka od njih ima određene nedostatke. Sukladno tome, razvoj unaprijeđenih postupaka koji nastoje prevladati neke od tih nedostataka iskazuje se kao valjan smjer istraživanja s ciljem učinkovitijeg učenja iz neuravnoteženih podataka.

1.2 Ciljevi disertacije

Prethodno opisani tijek izvođenja postupaka za učenje iz neuravnoteženih skupova podataka može se grafički ilustrirati kao što je prikazano slikom 1.1. Iako se u literaturi mogu naći brojne inačice ovih postupaka, u okviru svake postoje određeni nedostaci koji prvenstveno ograničavaju njihov doprinos uspješnosti klasifikacije. Uz to, neka unaprjeđenja nastoje prevladati te nedostatke uvođenjem dodatnih procedura i parametara koji kontroliraju način rada tih procedura, što otežava njihovo korištenje. Treba napomenuti da ovo povećanje složenosti nije uvijek popraćeno povećanjem njihove učinkovitosti. Slijedom toga, ne nedostaje prostora za poboljšanje spomenutih postupaka, posebice u smislu povećanja njihove učinkovitosti pri učenju iz neuravnoteženih podataka te pojednostavljenja njihova korištenja.

U ovoj disertaciji predlažu se unaprjeđenja uobičajenih izvedbi spomenutih postupaka za učenje iz neuravnoteženih skupova podataka. U sklopu postupka odabira značajki, dan je prijedlog proširenja za omotače zasnovane na bio-inspiriranim algoritmima optimizacije. Unatoč njihovoj sposobnosti otkrivanja složenih interakcija između značajki te postizanja povoljnih performansi u skladu s tim, ovi omotači često pronalaze podskupove značajki koji klasifikatoru dozvoljavaju slabu sposobnost generalizacije te općenito pronalaze različita rješenja kroz višestruka izvođenja što ih čini nestabilnim pristupom odabiru značajki. Predloženim proširenjem nastoje se ublažiti ovi nedostaci omotača te se zauzvrat očekuje povećanje njihova doprinosa generalizaciji algoritma za klasifikaciju i poboljšanje njihove stabilnosti kako u smislu višestrukog izvođenja tako i u smislu preslagivanja skupa podataka. Nadalje, predložena je unaprijeđena inačica algoritma SMOTE (engl. *synthetic minority oversampling technique*), kao jednog od najistaknutijih algoritama preuzorkovanja. Iako je ovaj algoritam iznimno popularan u literaturi zbog svoje korisnosti i jednostavnosti, ima određene nedostatke koji mogu uzrokovati povećanje složenosti skupa podataka te narušavanje izvedbe klasifikacije. Nedostaci ovog algoritma izraženiji su pri neprikladnim postavkama njegovih parametara, a potreba za njihovim podešavanjem otežava korištenje ovog algoritma. Predloženo unaprjeđenje algoritma SMOTE nastoji prevladati ove nedostatke te ukloniti potrebu za parametrima koji upravljaju njegovim načinom preuzorkovanja, kako bi se pojednostavila uporaba algoritma te zadržao ili poboljšao njegov učinak. Konačno, dan je prijedlog novog postupka izgradnje klasifikacijskih modela RBFN koji nastoji pronaći slijed mreža povećane složenosti te povoljne izvedbe klasifikacije. Iako se u literaturi mogu pronaći razni postupci koji grade slijed mreža, većina tih postupaka ne koristi znanje iz izgrađenih mreža manje složenosti pri traženju mreža veće složenosti, a oni koji koriste ne pripisuju pažnju početnom koraku nadogradnje. Stoga se predloženi postupak temelji na korištenju ovog znanja, uz značajne napore da se ono iskoristi na primjeren način kako bi se olakšala i ubrzala izgradnja slijeda mreža povoljne izvedbe.

Ova unaprjeđenja općenito su usmjerena na poboljšanje učinkovitosti spomenutih postupaka u smislu njihova doprinosa izvedbi klasifikacije. Pritom je od posebne važnosti poboljšanje uspješnosti prepoznavanja manjinske klase koje postižu. Navedeni ciljevi disertacije mogu se prikazati i kroz ispunjavanje sljedećih očekivanih izvornih znanstvenih doprinosa:

1. Proširenje bio-inspiriranih omotača za odabir značajki zasnovano na prikupljanju rješenja tijekom pretrage i objedinjavanju rješenja prema njihovom doprinosu kvaliteti
2. Unaprjeđenje tehnike sintetičkog preuzorkovanja manjine (SMOTE) uklanjanjem parametara algoritma i novim pristupom stvaranja sintetičkih primjeraka prema unutarnjim karakteristikama podataka
3. Postupak izgradnje klasifikacijskih modela radijalne neuronske mreže postupnim povećavanjem složenosti prethodnih modela

1.3 Pregled sadržaja disertacije

U poglavlju 2 objašnjen je problem učenja iz neuravnoteženih podataka te su razmotreni prikladni pristupi za njegovo ublažavanje. S obzirom na to da neuravnoteženi skupovi podataka u pravilu opisuju različite probleme klasifikacije, dan je uvod u opći problem klasifikacije i predstavljene su standardne mjere za vrednovanje uspješnosti algoritama za klasifikaciju te su istaknute one prikladne za neuravnotežene probleme. Nadalje, detaljno je opisan problem neuravnoteženosti klasa, s naglaskom na njegovu manifestaciju kroz pogoršanje štetnih učinaka raznih unutarnjih karakteristika skupova podataka. Izložen je pregled često korištenih pristupa za ublažavanje ovog problema, uz osvrt na njihove prednosti i nedostatke.

U poglavlju 3 opisan je postupak odabira značajki te je razmotrena njegova uloga u ublažavanju problema neuravnoteženosti klasa. Predstavljeni su uobičajeni pristupi iz literature za njegovo provođenje, s posebnim osvrtom na omotače zasnovane na bio-inspiriranim algoritmima optimizacije. Opisan je prijedlog proširenja za ove omotače koje se zasniva na prikupljanju kvalitetnih rješenja tijekom pretrage omotača te njihovu naknadnom objedinjavanju. Eksperimentalnom analizom ispitana je korisnost predloženog proširenja za standardne i unaprijeđene bio-inspirirane omotače iz literature.

Poglavlje 4 bavi se postupkom preuzorkovanja te njegovom važnosti za poboljšanje uspješnosti prepoznavanja manjinske klase. Ukratko je opisan način rada algoritma SMOTE te su istaknuti osnovni nedostaci ovog algoritma. Također, dan je pregled literature kroz koji su sažeto predstavljena postojeća unaprjeđenja ovog algoritma, opet s osvrtom na njihove nedostatke. Izložen je prijedlog unaprijeđene inačice algoritma SMOTE koja određuje susjedstva manjinskih primjeraka te stvara sintetičke primjerke uzimajući u obzir unutarnje karakteristike skupa podataka. Učinak predloženog algoritma eksperimentalno je uspoređen s učinkom algoritma SMOTE i nekoliko njegovih unaprjeđenja iz literature.

U poglavlju 5 predstavljen je problem izgradnje klasifikacijskih modela radijalne neuronske mreže. Pojašnjena je struktura RBFN te značaj njezinih parametara. Također, dan je osvrt na postojeće postupke za određivanje strukture RBFN, s posebnim naglaskom na postupke koji su primarno namijenjeni za izgradnju i treniranje klasifikacijskih modela. Opisan je prijedlog novog postupka izgradnje klasifikacijskih modela RBFN koji nastoji pronaći slijed mreža povećane složenosti koristeći znanje iz prethodno treniranih mreža manje složenosti. Učinkovitost predloženog postupka eksperimentalno je uspoređena s učinkovitosti nekoliko drugih postupaka za izgradnju ovih modela.

Zaključci i moguće smjernice za budući rad predstavljeni su u poglavlju 6. Ukratko su razmotreni ključni rezultati ostvareni primjenom predloženih unaprijeđenih postupaka za odabir značajki, preuzorkovanje te izgradnju klasifikacijskih modela RBFN. Istaknuti su doprinosi predloženih unaprjeđenja te su ponuđene smjernice za moguće dorade i daljnje istraživanje.

2

Učenje iz neuravnoteženih podataka

UČENJE iz neuravnoteženih podataka odnosi se na problem manjkave izvedbe standardnih algoritama za klasifikaciju pri kategorizaciji primjeraka manjinske klase u skupu podataka. Takvi primjerci obično predstavljaju rijetke događaje čije je prepoznavanje od kritične važnosti u području primjene iz kojeg proizlaze. Ovaj je problem zadobio značajan istraživački interes zbog svoje rasprostranjenosti i složenosti. Slijedom toga, razvijen je niz pristupa za njegovo ublažavanje. Primarni zadatak ovih pristupa jest pospješiti prepoznavanje manjinske klase, a njihovo vrednovanje obično je vođeno sveobuhvatnim mjerama uspješnosti klasifikacije. U ovom poglavlju, približena je priroda problema učenja iz neuravnoteženih podataka te su predstavljeni prikladni pristupi za njegovo ublažavanje.

2.1 Uvod u problem klasifikacije

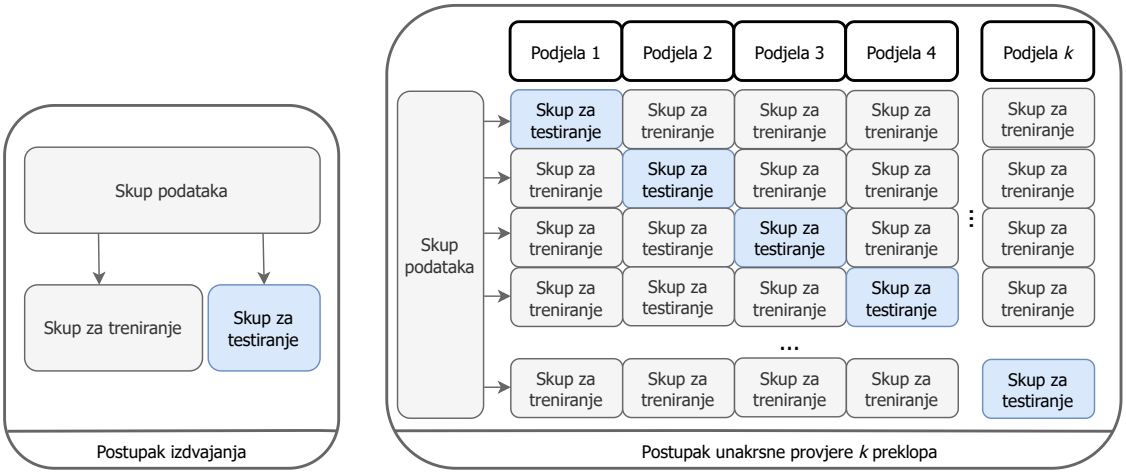
Grupiranje i klasifikacija osnovni su oblici općenitijeg zadatka raspoznavanja uzoraka (engl. *pattern recognition*) koji ima za cilj rasporediti ulazne podatke u konačan broj kategorija, odnosno klasa [5]. Zadatak klasifikacije jest definiranje funkcije koja primjercima iz skupa podataka dodjeljuje oznake klase iz unaprijed definiranog skupa klasa. Prema terminologiji strojnog učenja, takva funkcija poznata je kao hipoteza, dok skup parametriziranih hipoteza čini klasifikacijski model [21]. Svaki primjerak u skupu podataka predstavlja jednu podatkovnu točku za koju hipoteza određuje oznaku klase. Kako bi se moglo baratati primjercima, nužno je prikladno odrediti njihovu reprezentaciju. Stoga je uobičajeno opisati pojedini primjerak s nizom kategoričkih, numeričkih, logičkih i drugih vrijednosti, koje se nazivaju značajkama. Dakle, svaki primjerak u skupu podataka određen je pomoću vektora

$$\mathbf{X} = \begin{pmatrix} x_1^1 & x_2^1 & ... & x_d^1 \\ x_1^2 & x_2^2 & ... & x_d^2 \\ & & \cdot & \\ & & \cdot & \\ & & \cdot & \\ x_1^N & x_2^N & ... & x_d^N \end{pmatrix} \qquad \mathbf{l} = \begin{pmatrix} l_{q(1)} \\ l_{q(2)} \\ \cdot \\ \cdot \\ \cdot \\ l_{q(N)} \end{pmatrix}$$

Slika 2.1: Matrični zapis skupa označenih primjeraka $\mathcal{Q}$

značajki $\mathbf{x} = (x_1, x_2, ..., x_d)$, gdje su x_j za $j = 1, \ldots, d$ pojedine značajke, a d predstavlja broj značajki, odnosno dimenzionalnost problema. Skup svih primjeraka čini prostor primjeraka ili ulazni prostor $\mathcal{X}$. Svakom je primjerku u skupu podataka pridružena kategorička oznaka klase $l_k \in \mathcal{L}$, gdje $\mathcal{L} = \{l_k : k = 1, \ldots, m\}$ predstavlja skup svih oznaka klasa, a m broj mogućih klasa. Prema tome, može se reći da je svaki označeni primjerak uređeni par vektora značajki i oznake klase, a skup svih označenih primjeraka može se izraziti kao $\mathcal{Q} = \{(\mathbf{x}^i, l_{q(i)}) : \mathbf{x}^i \in \mathcal{X}, l_{q(i)} \in \mathcal{L}, i = 1, ..., N\}$, pri čemu N označava broj primjeraka. Skup $\mathcal{Q}$ jednostavnije se prikazuje matričnim zapisom, pomoću matrice (neoznačenih) primjeraka $\mathbf{X}$ te stupčastog vektora njihovih oznaka $\mathbf{l}$, kao na slici 2.1. Formalni iskaz cilja klasifikacije može se predstaviti hipotezom $h : \mathcal{X} \to \mathcal{L}$, koja primjercima iz $\mathcal{X}$ dodjeljuje oznake iz $\mathcal{L}$. Definiranje takve hipoteze zadaća je algoritma za klasifikaciju, odnosno klasifikatora.

Klasifikator u suštini trenira model nad raspoloživim podacima za treniranje (engl. *training set*), s ciljem pronalaženja optimalne hipoteze, odnosno one hipoteze koja kategorizira primjerke u pripadne klase uz minimalnu pogrešku. Može se reći da klasifikacijski model tako uči koncepte svake klase. Prije samog treniranja, u pravilu se odabire složenost modela podešavanjem njegovih hiperparametara, odnosno parametara klasifikatora. Obično se preferiraju jednostavniji modeli s obzirom na to da složeniji mogu uzrokovati prenaučenost (engl. *overfitting*) klasifikatora, smanjujući time njegovu sposobnost generalizacije. S druge strane, odabir previše jednostavnih modela dovodi do podnaučenosti (engl. *underfitting*) klasifikatora koji stoga ne uspijeva ostvariti zadovoljavajuću izvedbu ni na skupu za treniranje. Ako trenirani model ne može ispravno klasificirati primjerke iz skupa za treniranje, izgledno je da će neispravno klasificirati i neviđene primjerke. Kako bi se procijenila sposobnost generalizacije klasifikatora, izvedba treniranog modela obično se testira na skupu za testiranje (engl. *test set*) koji čine primjerci iz skupa podataka izdvojeni prethodno treniranju modela. Najčešći postupci raspodjele skupa podataka za potrebe testiranja su postupak izdvajanja (engl. *holdout*) te postupak unakrsne provjere k preklopa (engl. *k-fold cross-validation*), a načela rada svakog od njih ilustrirani su slikom 2.2. Postupak izdvajanja dijeli skup podataka na dva disjunktna podskupa, odnosno na skup za treniranje te na skup za testiranje. Postupak unakrsne provjere k preklopa koristi se kada je veličina skupa za treniranje, dobivenog postupkom izdvajanja, nedostatna za treniranje učinkovitih klasifikacijskih modela. Stoga se skup podataka često dijeli na k disjunktnih podskupova (preklopa), pri čemu se svaki od



Slika 2.2: Postupci raspodjele skupa podataka za potrebe testiranja klasifikacijskog modela

njih samo jednom koristi za testiranje, a $k-1$ puta za treniranje modela. Konačna kvaliteta klasifikacijskog modela izražava se kao aritmetička sredina njegovih vrednovanih izvedbi na svakom od k preklopa korištenih za testiranje. Navedeni postupci mogu se podesiti tako da ujedno provode i stratificiranu raspodjelu skupa podataka, odnosno zadržavaju originalni omjer primjeraka po klasama u izvedenim podskupovima. Na taj način izvedeni podskupovi za treniranje i testiranje preciznije odražavaju prirodu promatranog problema klasifikacije.

U literaturi je predloženo pregršt klasifikatora koji se generalno razlikuju prema klasifikacijskom modelu koji treniraju, optimizacijskom postupku korištenom za treniranje te načinu vrednovanja modela tijekom treniranja [22]. Poznato je da niti jedan od njih nije načelno uspješniji prilikom usporedbe na velikom broju raznolikih problema [23]. U suštini, odabir samog klasifikatora uvelike je zasnovan na složenosti i prirodi problema koji se nastoji naučiti. Skupovi podataka koji opisuju različite probleme klasifikacije mogu se značajno razlikovati prema strukturi i unutarnjim karakteristikama, što ponekad može pogodovati specifičnom tipu klasifikatora. Međutim, stjecanje uvida u stvarnu strukturu skupova podataka u pravilu nije ostvarivo, pa se u praksi odabir klasifikatora temelji na rezultatima eksperimentalnih usporedbi. One uključuju provedbu postupka podešavanja parametara klasifikatora i usporedbu njihovih performansi pomoću standardnih mjera za vrednovanje uspješnosti klasifikacije.

S obzirom na to da su mnogi algoritmi za klasifikaciju inherentno binarni klasifikatori, glavnina istraživačke pozornosti za potrebe klasifikacije primjeraka u veći broj klasa posvećena je dekompoziciji problema višeklasne klasifikacije u više problema binarne klasifikacije [24]. Najpoznatije sheme takve dekompozicije su shema jedan-naspram-jedan (engl. *one-vs-one*, OVO) te jedan-naspram-ostali (engl. *one-vs-rest*, OVR). U prvoj se jedan problem višeklasne klasifikacije svodi na $\binom{m}{2}$ nezavisnih problema binarne klasifikacije, za svaki par klasa [25], dok se u potonjoj izvodi m problema binarne klasifikacije, po jedan za svaku klasu [26]. Iako dekompozicija OVR shemom rezultira manjim brojem problema binarne klasifi-

kacije, oni sadržavaju neuravnotežen broj primjeraka dobivenih klasa jer se primjerci jedne klase suprotstavljaju primjercima svih ostalih klasa. Odabir između OVO i OVR shema dekompozicije u praksi se svodi na kompromis između broja problema klasifikacije s jedne i neuravnoteženosti klasa s druge strane.

2.2 Mjere za vrednovanje uspješnosti klasifikacije

Za predstavljanje izvedbe klasifikatora rabe se standardne mjere uspješnosti u literaturi. Mnogi klasifikatori koriste funkciju gubitka nula-jedan (engl. *zero-one loss*) za vrednovanje hipoteza tijekom treniranja, koja računa broj netočno klasificiranih primjeraka iz skupa za treniranje [18]. Stoga je treniranje vođeno s ciljem minimizacije ukupnog troška, odnosno s ciljem minimizacije ukupnog broja pogrešnih predviđanja. Empirijska pogreška koja iskazuje koliko dobro trenirani model klasificira sve primjerke iz skupa podataka, a proizlazi iz tako definirane funkcije gubitka, naziva se još i pogreškom klasifikacije (engl. *misclassification error*)

$$\mathrm{MCR} = \frac{1}{N}\sum_{i=1}^{N} e_i, \quad e_i = \begin{cases} 1, & \text{ako } h(\mathbf{x}^i) \neq l_i \\ 0, & \text{u suprotnom} \end{cases} . \tag{2.1}$$

Iz pogreške klasifikacije, može se izravno izračunati i točnost klasifikacije (engl. *classification accuracy*)

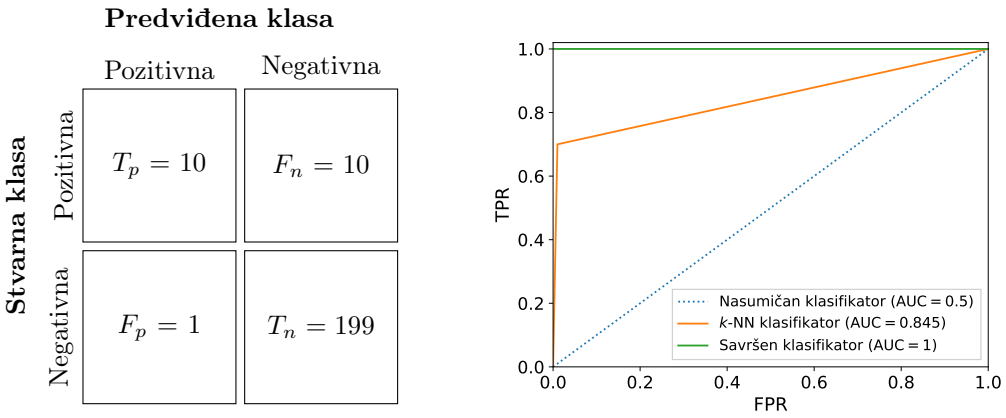
$$\mathrm{CAC} = 1 - \mathrm{MCR} \ . \tag{2.2}$$

Iako je postizanje visoke razine točnosti klasifikacije poželjno, ova mjera može biti nepouzdana prilikom učenja iz neuravnoteženih podataka. Detaljan uvid u izvedbu klasifikatora omogućava matrica zabune (engl. *confusion matrix*), čija je struktura prikazana na slici 2.3a. Njome su predstavljeni apsolutni brojevi točno i netočno klasificiranih primjeraka za svaku klasu. Objedinjavanjem tih pokazatelja moguće je odrediti uspješnost klasifikacije za pojedinačnu klasu. U binarnom problemu klasifikacije, manjinska klasa još se naziva i pozitivnom klasom, a većinska klasa negativnom klasom [27]. Na uspješnost prepoznavanja pozitivne klase u binarnom klasifikacijskom problemu upućuje stopa stvarno pozitivnih predviđanja (engl. *true positive rate*)

$$\mathrm{TPR} = \frac{\mathrm{T_p}}{\mathrm{T_p} + \mathrm{F_n}}, \tag{2.3}$$

a na uspješnost prepoznavanja negativne klase stopa stvarno negativnih predviđanja

$$\mathrm{TNR} = \frac{\mathrm{T_n}}{\mathrm{T_n} + \mathrm{F_p}} \ . \tag{2.4}$$



(a) Primjer matrice zabune (b) Primjer ROC krivulje

Slika 2.3: Poznati načini izvođenja mjera upješnosti klasifikacije

Prva se još često naziva i mjerom osjetljivosti (engl. *sensitivity*, Sens) ili mjerom odziva (engl. *recall*, Rec), a druga mjerom specifičnosti (engl. *specificity*, Spec). Uz osjetljivost i specifičnost, iz matrice zabune često se izvode i mjera preciznosti (engl. *precision*)

$$\mathrm{Pre} = \frac{\mathrm{T_p}}{\mathrm{T_p} + \mathrm{F_p}} \tag{2.5}$$

te stopa lažno pozitivnih predviđanja (engl. *false positive rate*)

$$\mathrm{FPR} = \frac{\mathrm{F_p}}{\mathrm{F_p} + \mathrm{T_n}} \ . \tag{2.6}$$

Mjera preciznosti pokazuje koliki je udio predviđanja pozitivne klase bio ispravan, dok stopa lažno pozitivnih predviđanja predstavlja udio primjeraka negativne klase koji su pogrešno klasificirani.

S obzirom na to da je vrednovanje izvedbe klasifikatora na temelju uspješnosti prepoznavanja pojedinačne klase složeno i nepregledno zbog velikog broja različitih mjera, u literaturi su predložene mjere koje objedinjuju informacije prikazane u matrici zabune. Jedna od najčešće korištenih takvih mjera je F-mjera (engl. *F-score*), koja je dana kao

$$\mathrm{F}_\beta = (1 + \beta^2) \cdot \frac{\mathrm{Pre} \cdot \mathrm{Rec}}{\beta^2 \cdot \mathrm{Pre} + \mathrm{Rec}} \ . \tag{2.7}$$

Postavljanjem vrijednosti parametra $\beta \in \mathbb{R}^+$ moguće je naglasiti važnost predviđanja pojedine klase. Ipak, on se često postavlja na 1, pa se ta mjera još zove i F1, a u suštini predstavlja harmonijsku sredinu odziva i preciznosti. Uz harmonijsku, geometrijska sredina specifičnosti i osjetljivosti

$$\mathrm{G_{mean}} = \sqrt{\mathrm{Spec} \cdot \mathrm{Sens}} \tag{2.8}$$

također se koristi kao mjera opće uspješnosti klasifikacije. Još jedan rašireni način vredno-

vanja uspješnosti klasifikacije je analiza krivuljom operativnih karakteristika (engl. *receiver operator characteristic*, ROC) koja pruža grafički prikaz ovisnosti TPR o FPR za različite postavke praga diskriminacije između klasa unutar klasifikatora, kao što je prikazano na slici 2.3b. Površina ispod ROC krivulje (engl. *area under the curve*, AUC) jedna je od često korištenih mjera kvalitete izvedbe klasifikatora, koja s jednim brojem opisuje njegovu sposobnost razlikovanja primjeraka različitih klasa.

Iako su sve navedene mjere uspješnosti klasifikacije prvobitno izvedene za probleme binarne klasifikacije, koriste se i na problemima višeklasne klasifikacije. Pri dekompoziciji takvih problema OVO ili OVR shemama, za svaki izvedeni problem binarne klasifikacije njihove vrijednosti lako se određuju. Zatim se uobičajeno provodi mikro ili makro agregacija izvedenih rezultata, pri čemu prva uzima u obzir omjer podataka po klasama, dok se kod druge svakoj klasi dodjeljuje jednaka težina pri računanju težinske aritmetičke sredine izvedenih vrijednosti [28].

2.3 Problem neuravnoteženosti klasa

Vrednovanjem algoritama za klasifikaciju u raznim područjima primjene zamijećene su određene sličnosti u njihovoj izvedbi na problemima s neuravnoteženom raspodjelom primjeraka različitih klasa. Neuravnoteženi skupovi podataka proizlaze iz brojnih područja primjene [6], a neki od njih su prikazani tablicom 2.1. U takvim problemima, uspješno prepoznavanje manjinske klase od primarne je važnosti. Međutim, učenje iz neuravnoteženih podataka u pravilu rezultira klasifikacijskim modelima koji su pristrani većinskoj klasi [10], a ostvaruju nisku razinu uspješnosti prepoznavanja manjinske klase. Većina standardnih klasifikatora u literaturi iskazuje takvo ponašanje [10, 29–31], a osnovni razlog tomu je što njihove procedure treniranja nastoje maksimizirati broj točno klasificiranih primjeraka, neovisno o uspješnosti klasifikacije pojedine klase. Primjerice, uobičajene mjere čistoće kod stabla odluke (engl. *decision tree*, DT) te funkcija gubitka nula-jedan kod višeslojnog perceptrona (engl. *multilayer perceptron*, MLP) jednako kažnjavaju pogrešnu klasifikaciju svakog primjerka, što rezultira treniranim klasifikacijskim modelima pristranima većinskoj klasi [32, 33]. Ipak, funkcija gubitka nije jedini element klasifikatora koji uzrokuje takvo ponašanje. Tako sama formulacija optimizacijskog postupka (tzv. meka margina) kod metode potpornih vektora (engl. *support vector machine*, SVM) dovodi do većeg nesrazmjera u broju potpornih vektora svake klase, čime se smanjuje njegova sposobnost generalizacije i povećava mogućnost klasificiranja neviđenog primjerka kao pripadnika većinske klase [34]. Iako ne provodi učenje raspoloživih podataka, algoritam k-najbližih susjeda (engl. *k-nearest neighbours*, k-NN) također je pogođen problemom neuravnoteženosti klasa jer neviđenom primjerku dodjeljuje oznaku klase s najvećom *a priori* vjerojatnosti klase [35]. Da vrsta klasifikacijskog modela također uvelike određuje izvedbu klasifikatora na neuravnoteženim problemima, pokazuje i

Tablica 2.1: Sažet pregled područja primjene zahvaćenih problemom neuravnoteženosti klasa

Područje primjene	Istaknuti problemi klasifikacije	Autori
Biomedicina	Klasifikacija proteinskih struktura	Zhao et al. [7]
	Analiza genskog izražaja	Anaissi et al. [39]
	Dijagnoza bolesti	Fotouhi et al. [40]
Financije	Otkrivanje financijskih prijevara	Ye et al. [41]
	Utvrđivanje kreditnog rejtinga	Brown i Mues [42]
Informacijska tehnologija	Prepoznavanje grešaka u softveru	Wang i Yao [43]
	Prepoznavanje mrežnih provala	Cieslak et al. [9]
	Kategorizacija teksta	Zheng et al. [44]
	Analiza sentimenta	Ghosh et al. [45]
Industrijska proizvodnja	Otkrivanje nedostataka proizvoda	Wu et al. [46]
	Dijagnoza kvarova na strojevima	Yi et al. [47]
Upravljanje sigurnosti	Prepoznavanje prirodnih nepogoda	Trafalis et al. [48]
	Otkrivanje prijetnji iz nadzornih videozapisa	Franklin et al. [8]

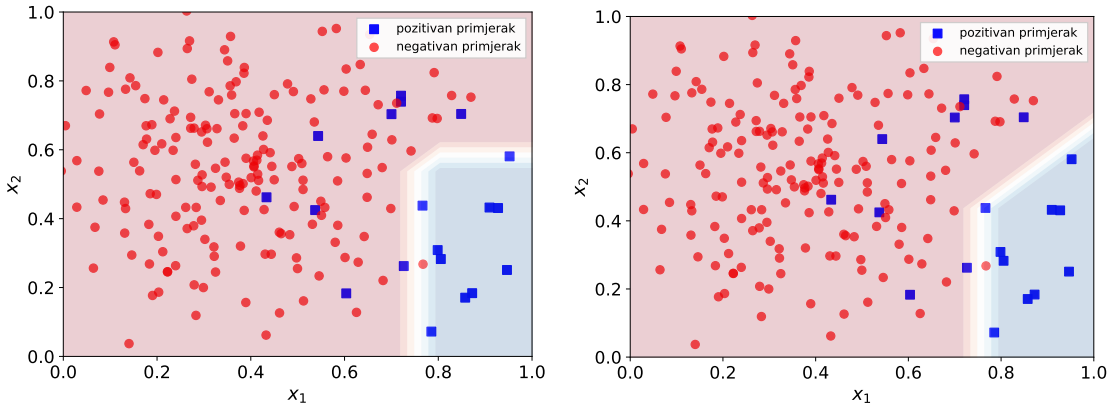
primjer radijalne neuronske mreže (RBFN), koja je ispoljila zadovoljavajuće rezultate na brojnim neuravnoteženim problemima klasifikacije [20, 36, 37] unatoč načinu vrednovanja modela sličnom kao kod klasifikatora MLP. S obzirom na skroman broj primjeraka (posebice manjinskih), metode dubokog učenja rijetko pronalaze svoju primjenu na prikazanim problemima klasifikacije [38]. Detaljnija obrazloženja pristranosti standardnih klasifikatora većinskoj klasi mogu se pronaći u [18].

Većina literature o učenju iz neuravnoteženih podataka posvećena je problemima binarne klasifikacije, pri čemu jedna klasa značajno nadmašuje drugu po brojnosti. U tom slučaju, osnovna mjera kojom se iskazuje stupanj neuravnoteženosti nekog skupa podataka jest omjer neuravnoteženosti (engl. *imbalance ratio*)

$$\mathrm{IR} = \frac{|\mathcal{V}|}{|\mathcal{M}|}, \tag{2.9}$$

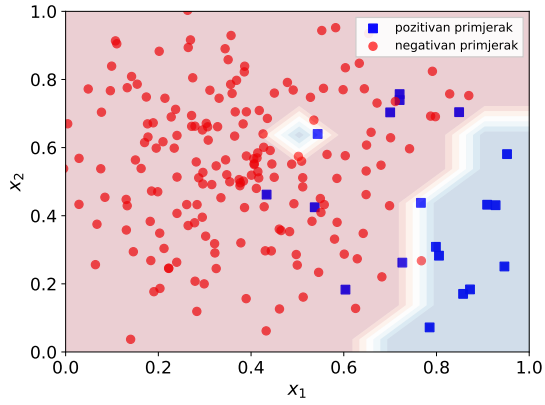
gdje $\mathcal{V}$ označava skup primjeraka većinske, a $\mathcal{M}$ manjinske klase [49]. Slika 2.4 prikazuje primjer binarnog klasifikacijskog problema na dvodimenzionalnom sintetičkom skupu podataka s omjerom neuravnoteženosti $\mathrm{IR} = \frac{200}{20} = 10$. Radi ilustracije utjecaja problema neuravnoteženosti klasa na spomenute algoritme za klasifikaciju, slika također prikazuje i njihove granice odluke, koje su u pravilu značajno bliže primjercima manjinske klase. Prilikom vrednovanja izvedbe klasifikatora na neuravnoteženim problemima, uglavnom se izbjegava uporaba točnosti klasifikacije (CAC), s obzirom na to da čak i trivijalan većinski klasifikator koji svakom primjerku dodjeljuje oznaku većinske klase, može postići izuzetnu točnost klasifikacije [29]. Primjerice, za sintetički skup podataka sa slike 2.4, točnost trivijalnog većinskog klasifikatora nadomak je točnosti ostalih klasifikatora, kao što je prikazano u tablici 2.2. Međutim, takav klasifikator pogrešno klasificira sve primjerke manjinske klase čije je prepoznavanje od presudne važnosti, kao što je ranije objašnjeno.

Stvarni skupovi podataka u pravilu obuhvaćaju razne unutarnje karakteristike koje pove-

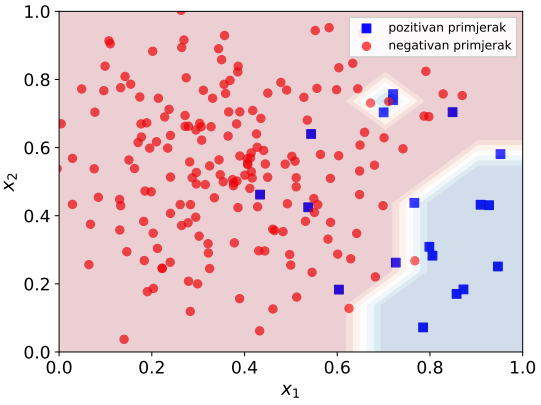


(a) DT

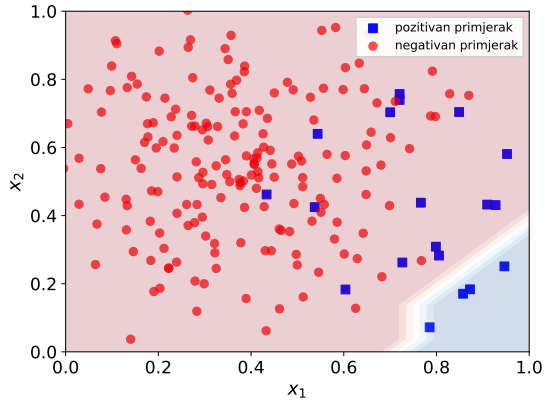
(b) SVM



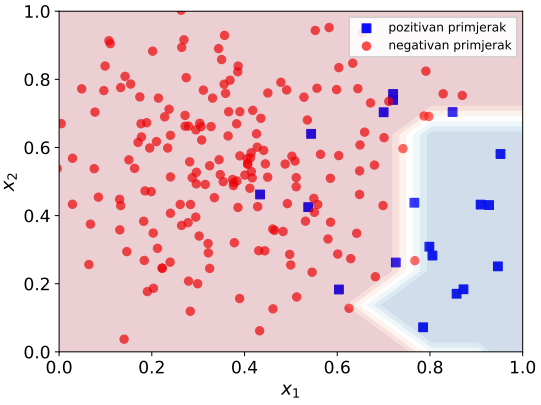
(c) 1-NN



(d) 5-NN



(e) MLP



(f) RBFN

Slika 2.4: Granice odluke standardnih klasifikatora na neuravnoteženom binarnom problemu klasifikacije (IR = 10)

ćavaju njihovu složenost te narušavaju izvedbu klasifikatora. Težina problema neuravnoteženosti klasa odražava se upravo kroz te karakteristike dodatno pogoršavajući navedene učinke. Manifestacija unutarnjih karakteristika skupa podataka postaje još složenija kod neuravnoteženog problema klasifikacije, a određene tehnike za ublažavanje njihovih učinaka prestaju biti djelotvorne jer dodatno povećavaju stupanj neuravnoteženosti te tako otežavaju treniranje učinkovitih klasifikacijskih modela. Prisutnost šuma u neuravnoteženim skupovima

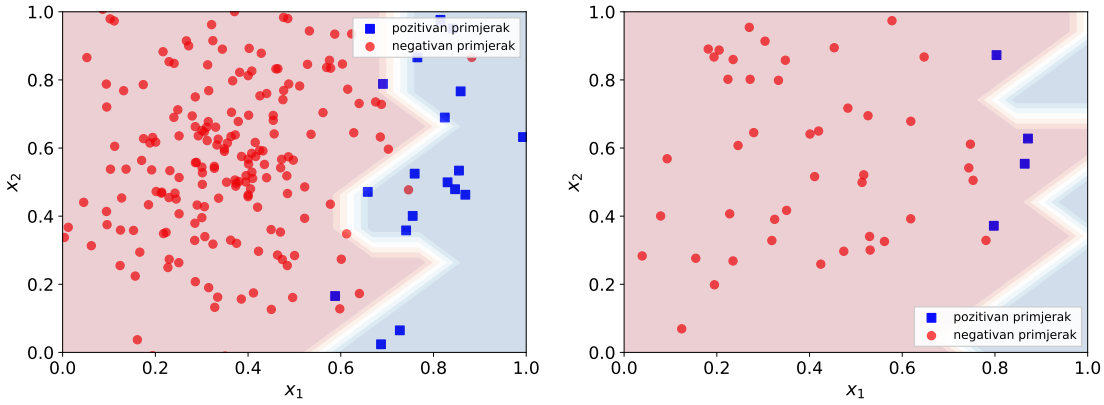
Tablica 2.2: Izvedba standardnih klasifikatora na neuravnoteženom binarnom problemu klasifikacije (IR = 10)

Klasifikator	TPR (%)	TNR (%)	CAC (%)
Trivijalan	0	100	90.9
DT	50	99.5	95
SVM	50	99.5	95
1-NN	55	99.5	95.5
5-NN	65	98.5	95.5
MLP	20	100	92.7
RBFN	55	99	95

podataka često izaziva mnogo problematičnije posljedice nego kod uravnoteženih problema. Primjerice, pogrešno označavanje primjerka manjinske klase rezultirat će povećanjem omjera neuravnoteženosti te uvođenjem većinskih primjeraka u područje manjinske klase. S druge strane, pogrešno označavanje primjerka većinske klase povećava složenost koncepta manjinske klase. U konačnici, klasifikator može naučiti šum kao dio koncepta neke klase, što dovodi do isuviše složenih klasifikacijskih modela koji nemaju dobru sposobnost generalizacije. Iako su u literaturi predložene brojne tehnike za uklanjanje šuma iz skupa podataka [50], treba biti oprezan pri uklanjanju manjinskih primjeraka koji su prepoznati kao šum jer se tako povećava omjer neuravnoteženosti te potencijalno stvara problem apsolutne rijetkosti.

U brojnim neuravnoteženim problemima klasifikacije manjinski primjerci predstavljaju rijetke slučajeve koji su u vrlo malom broju zastupljeni u skupu podataka, što se još naziva i problemom apsolutne rijetkosti. Nedostatak reprezentativnih manjinskih primjeraka dovodi do klasifikacijskih modela koji većinu ulaznog prostora proglašavaju područjem većinske klase. Rijetki manjinski primjerci mogu se pri treniranju klasifikacijskog modela tretirati kao šum u podacima, dok se šum može pogrešno identificirati kao ispravan manjinski primjerak, budući da oba predstavljaju rijetke slučajeve u ulaznom prostoru. Slika 2.5 prikazuje učinak apsolutne rijetkosti na granicu odluke klasifikatora 5-NN. Oba skupa podataka na slici proizlaze iz iste distribucije te im je stupanj neuravnoteženosti klasa jednak. Prikazani utjecaj problema apsolutne rijetkosti daje naslutiti kako omjer neuravnoteženosti skupa podataka nije primjeren pokazatelj njegove složenosti.

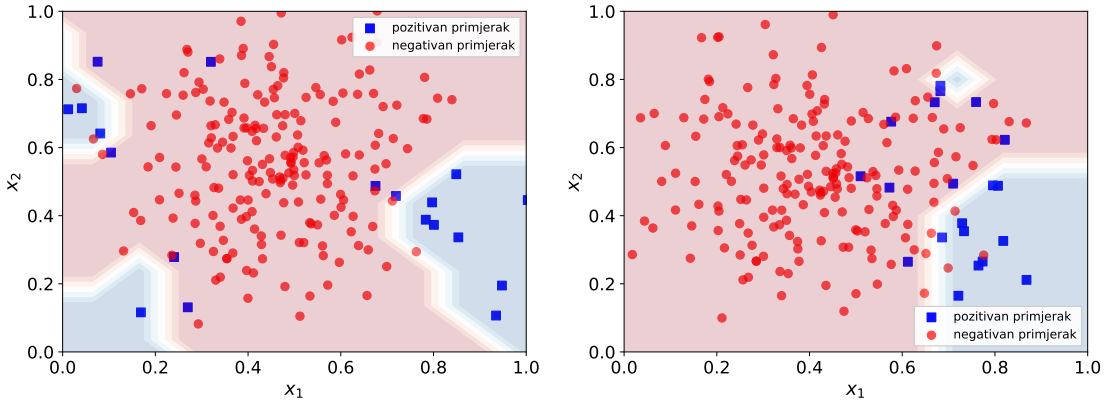
Primjerci unutar skupa podataka često su rasprostranjeni tako da je koncept njihove klase predstavljen kao disjunkcija nekoliko podkoncepata koji povećavaju složenost problema klasifikacije [51]. Posebno je problematično ako je manjinska klasa podijeljena, jer njezini podkoncepti mogu biti zahvaćeni problemom apsolutne rijetkosti. Eksperimentalno je utvrđeno da što su podkoncepti manje veličine to više doprinose rastu greške klasifikacije [51]. Na to upućuju i granice odluke klasifikatora 5-NN sa slike 2.6a, izvedene za skup podataka u kojem je manjinska klasa podijeljena na 3 podkoncepta različite veličine. Stoga je uklanjanje primjeraka koji čine podkoncepte male veličine popularna tehnika za ublažavanje ovog



(a) Neuravnoteženi skup podataka (IR = 10) bez problema apsolutne rijetkosti

(b) Neuravnoteženi skup podataka (IR = 10) s apsolutnom rijetkosti manjinskih primjeraka

Slika 2.5: Primjer utjecaja problema apsolutne rijekosti na granicu odluke klasifikatora 5-NN



(a) Podkoncepti manjinske klase

(b) Preklapanje klasa

Slika 2.6: Primjer utjecaja unutarnjih karakteristika neuravnoteženog skupa podataka na granicu odluke klasifikatora 5-NN

problema u literaturi [51]. Međutim, kod neuravnoteženih skupova podataka ona u prvom redu uklanja primjerke manjinske klase, dodatno otežavajući učenje njezina koncepta.

Iako svaki klasifikator djeluje na jedinstven način, nijedan nije imun na problem preklapanja klasa [52]. Preklapanje klasa nastaje uslijed neznatne razlike u vrijednostima značajki među primjercima različitih klasa, pa ista područja ulaznog prostora sadrže primjerke iz više klasa. Klasifikator u ovom slučaju ne može odrediti jasnu granicu između klasa te je stoga određen broj primjeraka sa svake strane granice pogrešno klasificiran. Povrh toga, sustavna istraživanja ove karakteristike [53–55] potvrđuju da njezini štetni učinci postaju izraženiji kako se povećava omjer neuravnoteženosti skupa podataka. Ako je područje preklapanja klasa neuravnoteženo, standardni klasifikatori ponašaju se slično trivijalnom većinskom klasifikatoru [52], svrstavajući većinu primjeraka u većinsku klasu tog područja, na što upućuje i slika 2.6b.

2.4 Pristupi za ublažavanje problema neuravnoteženosti klasa

Problem neuravnoteženosti klasa odražava se kroz razne unutarnje karakteristike skupova podataka te njihove učinke čini još nepovoljnijima. S obzirom na složenost i rasprostranjenost ovog problema, nije iznenađujuće da su predloženi brojni pristupi koji ga nastoje ublažiti. Primarni zadatak takvih pristupa jest poboljšati uspješnost prepoznavanja manjinske klase, a pri tome ne narušiti izvedbu klasifikatora za ostale klase. Moguće ih je svrstati u četiri glavne skupine prema načinu rada [18].

Pristupi na razini algoritama prilagođavaju procedure treniranja standardnih klasifikatora s ciljem pospješenja njihove sposobnosti prepoznavanja manjinske klase. Ovi pristupi ne ublažavaju izravno stupanj problema neuravnoteženosti klasa u skupu podataka, već su usmjereni na izgradnju klasifikacijskih modela s načelno boljim svojstvima pri učenju iz neuravnoteženih podataka. Ipak, predlaganje novog pristupa na razini algoritma zahtijeva dubinsko razumijevanje odabranog klasifikatora kako bi se identificiralo specifično svojstvo algoritma koje može biti odgovorno za njegovu pristranost većinskoj klasi. U literaturi je predloženo više od 160 izmjena klasifikatora za poboljšanje učenja iz neuravnoteženih podataka, a neki od popularnijih tiču se algoritama SVM, k-NN i DT [6]. Većina pristupa na razini algoritama uvodi dodatne parametre u klasifikacijski model koji u osnovi predstavljaju trošak pogrešnog predviđanja manjinske klase [56] ili pak svakog pojedinog primjerka [57–60]. Kod klasifikatora DT i k-NN, popularne izmjene temelje se na uporabi mjera čistoće stabla, odnosno udaljenosti primjeraka, koje su prikladnije za neuravnotežene skupove podataka [32, 61–63]. No, navedene izmjene uglavnom dodatno otežavaju uporabu standardnih klasifikatora. Stoga su pristupi na razini algoritama rjeđe korištene tehnike za ublažavanje problema neuravnoteženosti klasa u literaturi [64, 65].

Troškovno-osjetljivo učenje (engl. *cost-sensitive learning*) smatra se aspektom pristupa na razini algoritama koji se temelji na izmjeni funkcije gubitka algoritma strojnog učenja. Većina standardnih klasifikatora koristi funkciju gubitka nula-jedan pa takvo treniranje u pravilu rezultira klasifikacijskim modelima koji su pristrani većinskoj klasi. Troškovno-osjetljivo učenje nastoji ublažiti ovaj problem prilagođavajući trošak pogrešnog predviđanja za svaku pojedinu klasu. Troškovi se definiraju pomoću matrice troška (engl. *cost-matrix*) koja ima jednaku strukturu matrici zabune, no vrijednosti predstavljaju težinu točnog i pogrešnog određivanja pojedine klase prilikom treniranja klasifikacijskog modela. Učinkovitost troškovno-osjetljivog učenja uvelike ovisi o definiranoj matrici troška, a nju je moguće pribaviti od strane stručnjaka u području primjene koji mogu preciznije brojčano odrediti stvarni trošak. Ako pak nije moguće tako definirati matricu troška, ona se može odrediti iz skupa podataka pomoću raznih heuristika. Vrlo popularna i jednostavna heuristika jest upotreba omjera neuravnoteženosti za trošak pogrešnog predviđanja manjinske klase [18].

Troškovno-osjetljivo učenje može biti izvedeno i bez matrice troška, tako da se definiraju različiti pragovi diskriminacije klasa unutar klasifikatora [66, 67]. Nedostatak raznih pristupa troškovno-osjetljivog učenja jest taj što, osim omjera neuravnoteženosti, ne uzimaju u obzir druge manisfestacije problema neuravnoteženosti klasa koje povećavaju složenost skupa podataka. Shodno tome, često rezultiraju narušenom uspješnosti prepoznavanja većinske klase.

Pristupi na razini podataka jesu postupci predobrade skupova podataka kojima se nastoji ublažiti stupanj problema neuravnoteženosti klasa. Predstavljaju ih postupak odabira značajki koji smanjuje dimenzionalnost problema klasifikacije te metode uzorkovanja koje uravnotežuju raspodjelu primjeraka različitih klasa. Neuravnoteženi problemi klasifikacije često proizlaze iz područja primjene u kojima su primjerci opisani velikim brojem značajki ili pak značajkama koje otežavaju razlikovanje primjeraka različitih klasa u skupu podataka. U takvim okolnostima, primjerci manjinske klase mogu biti nesmotreno tretirani kao šum ili se mogu nalaziti u području većinske klase. Postupak odabira značajki stoga odbacuje nevažne, redundantne te kontraproduktivne značajke, što dovodi do manje složenosti i kvalitetnije izvedbe treniranih klasifikacijskih modela [27]. Metode uzorkovanja u osnovi provode poduzorkovanje (engl. *undersampling*) većinske ili preuzorkovanje manjinske klase. Prvi postupak stvara podskup izvornog skupa podataka uklanjajući primjerke većinske klase, dok drugi stvara nadskup izvornog skupa podataka stvaranjem novih primjeraka manjinske klase. Najjednostavniji predstavnici metoda uzorkovanja jesu nasumično poduzorkovanje te nasumično preuzorkovanje. Prva metoda nasumično odabire primjerke većinske klase i uklanja ih iz izvornog skupa podataka, dok se u potonjoj metodi primjerci manjinske klase nasumično odabiru i dupliciraju sve dok se ne postigne željena razina uravnoteženosti. Unatoč njihovoj jednostavnosti, obje metode potencijalno povećavaju složenost skupa podataka. Uklanjanjem nasumičnih primjeraka većinske klase, klasifikator može propustiti važne koncepte koji se odnose na tu klasu [10]. S druge strane, umnožavanje postojećih primjeraka iz izvornog skupa podataka može dovesti do prenaučenosti klasifikatora [68]. Radi izbjegavanja ovih nepoželjnih posljedica, unutar obje skupine metoda razvijeni su napredniji algoritmi za uklanjanje postojećih ili stvaranje novih primjeraka [69–73]. Pristupe na razini podataka jednostavnije je implementirati u odnosu na ostale pristupe za ublažavanje problema neuravnoteženosti klasa. Uz to, smanjenje složenosti skupa podataka u konačnici pospješuje izvedbu raznih tipova klasifikatora [74, 75]. Stoga je upotreba ovih pristupa prevladavajući način ublažavanja problema neuravnoteženosti klasa u literaturi [10–12].

Svaki od spomenutih pristupa za ublažavanje problema neuravnoteženosti klasa pruža određene pogodnosti, pa se oni mogu kombinirati u obliku ansambla klasifikatora radi dodatnog poboljšanja izvedbe klasifikacije. Ansambli su poznate metode strojnog učenja koje provode treniranje većeg broja klasifikacijskih modela te zatim kombiniraju njihove rezultate u svrhu donošenja konačne odluke [76]. Najčešći pristup stvaranju ansambla klasifikatora radi ublažavanja problema neuravnoteženosti klasa temelji se na predobradi skupa podataka

nekim od pristupa na razini podataka prethodno treniranju svakog modela [77–79]. Drugi česti pristup odnosi se na uvođenje troškovno-osjetljivog učenja u standardne koncepte konstruiranja ansambla. Prema analizi djelotvornosti ansambla pri učenju iz neuravnoteženih podataka u [80] daje se naslutiti kako ansambli imaju pozitivan učinak na prepoznavanje manjinske klase, no često nauštrb uspješnosti prepoznavanja većinske klase. Učinkovitost ansambla uvjetovana je odabirom odgovarajuće kombinacije ranije spomenutih pristupa, čije utvrđivanje predstavlja dodatan izazov. Osim toga, ansambli su složeniji od pojedinačnih pristupa za ublažavanje problema neuravnoteženosti klasa i obično im je potrebno više vremena za izgradnju klasifikacijskih modela.

Pristupi na razini podataka jedini su od navedenih pristupa koji izravno ublažavaju stupanj problema neuravnoteženosti klasa u skupu podataka, što u konačnici pogoduje izvedbi raznih tipova klasifikatora. Među ovim pristupima, odabir značajki (FS) i preuzorkovanje (OS) mogu se istaknuti kao postupci predobrade skupa podataka koji smanjuju složenost koncepta manjinske klase te u pravilu pospješuju njezino prepoznavanje. Njihova jednostavnost i učinkovitost čine ih prikladnim izborom za ublažavanje problema neuravnoteženosti klasa.

2.4.1 Odabir značajki

Odabir značajki uobičajen je postupak predobrade skupova podataka kojim se smanjuje dimenzionalnost problema klasifikacije te potencijalno doprinosi razdvajanju koncepata različitih klasa u skupu podataka. Opisivanje primjeraka većim brojem značajki čini ih međusobno udaljenijima u ulaznom prostoru, što posebice otežava učenje koncepta manjinske klase. Metode uzorkovanja pokazale su se neučinkovitima za poboljšanje izvedbe klasifikatora na problemima klasifikacije velike dimenzionalnosti [81] te pri apsolutnoj rijetkosti manjinskih primjeraka [82]. U oba slučaja, teško je odrediti prikladno područje ulaznog prostora za uvrštavanje sintetičkih primjeraka, pa oni mogu biti smješteni daleko od postojećih primjeraka ili pak u područje većinske klase. Postupak odabira značajki može se koristiti za smanjenje udaljenosti postojećih manjinskih primjeraka u ulaznom prostoru, što olakšava učenje koncepta manjinske klase i poboljšava učinak metoda uzorkovanja [16, 82]. Odabir značajki također se u literaturi preporuča i za ublažavanje stupnja preklapanja klasa jer se ovim postupkom potencijalno eliminiraju one značajke čije vrijednosti dijeli velik broj primjeraka različitih klasa [17, 83]. Razdvajanje područja preklapanja klasa najviše doprinosi uspješnosti prepoznavanja manjinske klase jer se pretežito njezini primjerci pogrešno klasificiraju u području preklapanja, kako je ranije objašnjeno.

U literaturi su predloženi brojni pristupi kojima se ostvaruje zadatak odabira značajki, a moguće ih je razvrstati u nekoliko skupina, gdje svaka ima svojih prednosti i nedostataka. Ipak, pristupi iz skupine omotača obično su prikladniji za rukovanje neuravnoteženim skupovima podataka [84]. Ovi pristupi sve češće zasnivaju svoj rad na upotrebi bio-inspiriranih

algoritama za iscrpno ispitivanje složenih interakcija između postojećih značajki. Usprkos njihovoj učinkovitosti, ovi algoritmi ponekad uzrokuju i određene nepoželjne učinke zbog svoje stohastičke prirode. Bio-inspirirani omotači stoga su predmet brojnih izmjena ili proširenja koji u izvorne algoritme ugrađuju nove operacije specifične za problem odabira značajki, u svrhu pronalaženja podskupova značajki bolje kvalitete i konzistentnije strukture.

2.4.2 Preuzorkovanje

Korištenje metoda uzorkovanja standardni je postupak za ublažavanje problema neuravnoteženosti klasa u literaturi [18], primarno zbog njihove jednostavnosti i učinkovitosti. Brojna istraživanja pokazuju da preuzorkovanje značajnije doprinosi povećanju uspješnosti prepoznavanja manjinske klase nego poduzorkovanje [19, 52, 85, 86]. Klasifikatoru je obično problem naučiti koncept manjinske klase, s obzirom na to da on može biti podijeljen na podkoncepte, zastupljen s nedovoljnim brojem primjeraka ili pak raspoređen u ulaznom prostoru uz primjerke većinske klase. Od navedenih problema, poduzorkovanje jedino može ublažiti stupanj preklapanja klasa, no uz rizik uklanjanja velikog broja većinskih primjeraka i narušavanja izvedbe klasifikatora za većinsku klasu. S druge strane, preuzorkovanje se preporučuje za povećanje veličine podkoncepata manjinske klase [87, 88], čime se olakšava njihovo prepoznavanje. Slijedom toga, može se zaključiti da je umjetno povećanje skupa podataka stvaranjem sintetičkih primjeraka generalno prikladniji način ublažavanja problema neuravnoteženosti klasa od uklanjanja postojećih primjeraka. Bitno je napomenuti da preuzorkovanje ne uravnotežuje u potpunosti broj primjeraka različitih klasa, nego je optimalna količina sintetičkih primjeraka obično utvrđena eksperimentalno. Najistaknutija zamjerka brojnim algoritmima za preuzorkovanje jest ta što uvode sintetičke primjerke u područje većinske klase i tako dodatno povećavaju stupanj preklapanja klasa, pa se stoga često udružuju s postupkom odabira značajki u svrhu ublažavanja ove nepoželjne posljedice [83, 89].

U literaturi je predloženo pregršt algoritama za preuzorkovanje, a jedan od najistaknutijih i najkorištenijih jest algoritam SMOTE [69]. Tome svjedoči i preko 100 unaprjeđenja, odnosno izmjena tog algoritma predloženih u literaturi [70]. Unatoč njegovoj jednostavnosti i učinkovitosti, neki od njegovih svojstava mogu narušiti kvalitetu izvedbe klasifikatora. S ciljem prevladavanja tih nedostataka, unaprjeđenja tog algoritma zamjenjuju relativno jednostavne procedure izvornog algoritma složenijim procedurama te uvode dodatne parametre koji kontroliraju njihov način rada. Međutim, ovo povećanje složenosti često nije popraćeno i sa značajnim poboljšanjem performansi klasifikacije, na što ukazuju rezultati brojnih eksperimentalnih analiza [74, 90, 91]. Time se dovodi u pitanje valjanost povećanja složenosti relativno jednostavnog algoritma SMOTE kao pristupa njegovu unaprijeđenju.

2.4.3 Zastupljenost pristupa u literaturi

Kako bi se pružio uvid u zastupljenost navedenih pristupa za ublažavanje problema neuravnoteženosti klasa, napravljen je pregled literature. Točnije, analizirano je 130 članaka koji se bave rješavanjem problema klasifikacije navedenih u tablici 2.1, odnosno po 10 članaka za svaki problem. Jedini kriterij pri odabiru članaka bio je da predlažu način rješavanja nekog od istaknutih problema klasifikacije. Pretraga je vođena upisivanjem naziva problema u specijalizirane tražilice znanstvene literature, a redoslijed iščitavanja članaka određen je na temelju njihove relevantnosti. S obzirom na to da su svi promatrani problemi klasifikacije neuravnoteženi, u većini članaka se predlaže uporaba nekog od prethodno navedenih pristupa za ublažavanje problema neuravnoteženosti klasa. Dvije najčešće skupine takvih pristupa izvedene su za svaki problem klasifikacije i istaknute na slici 2.7, uz njihovu zastupljenost u pregledanim člancima. Kako su pristupi na razini podataka uvijek među istaknutim pristupima, dodatno su izražene zastupljenosti preuzorkovanja i odabira značajki kao njihovih primarnih predstavnika.

Bitno je napomenuti kako upotreba jednog pristupa za ublažavanje problema neuravnoteženosti klasa ne isključuje upotrebu ostalih pristupa. Tako se u mnogim člancima udružuje i po nekoliko njih, a u većini takvih kombinacija prevladavaju pristupi na razini podataka. Primjerice, za izvođenje zadatka otkrivanja financijskih prijevara, osam od 10 pregledanih članaka predlaže uporabu pristupa na razini podataka, a četiri članka uporabu troškovno-osjetljivog učenja, pri čemu se u dva takva članka ovi pristupi kombiniraju. Postupci preuzorkovanja i odabira značajki primjenjuju se u ukupno 57, odnosno 50 pregledanih članaka, dok se ostali pristupi primjenjuju znatno rjeđe. Kombinacija ova dva postupka predložena je u 27 pregledanih članaka. Odabir značajki koristi se u većini zadataka kategorizacije teksta te analize genskog izražaja, koji su poznati po velikoj dimenzionalnosti. S druge strane, uporaba preuzorkovanja uobičajena je kod zadataka prepoznavanja mrežnih provala te prepoznavanja grešaka u softveru, u kojima je stupanj neuravnoteženosti klasa izrazito velik. Zbog velikog broja dostupnih članaka izloženi pregled literature nije sveobuhvatan, no predstavlja uzorak koji pruža uvid u zastupljenost pristupa za ublažavanje problema neuravnoteženosti klasa u literaturi. Na temelju izvedenih podataka moguće je potvrditi prepoznatljivost postupaka za odabir značajki i preuzorkovanje u literaturi, koji zbog ranije navedenih odlika olakšavaju učenje iz neuravnoteženih podataka te pospješuju izvedbu raznih algoritama za klasifikaciju.

2.5 Osvrt na problem neuravnoteženosti klasa i njegovo ublažavanje

Prepoznavanje rijetkih događaja od primarne je važnosti u brojnim problemima klasifikacije koji proizlaze iz područja poput medicinske dijagnostike, prepoznavanja lica te otkrivanja upada, grešaka ili prijevara. Međutim, manjkava zastupljenost takvih događaja u skupu

Slika 2.7: Najčešći pristupi za ublažavanje problema neuravnoteženosti klasa u raznim područjima primjene

podataka u odnosu na ostale događaje znatno otežava njihovo prepoznavanje. Štoviše, s povećanjem omjera neuravnoteženosti, standardni algoritmi za klasifikaciju ponašaju se sve sličnije trivijalnom većinskom klasifikatoru. Težina problema neuravnoteženosti klasa proizlazi iz povećanja složenosti skupa podataka s obzirom na to da predstavljanje koncepta klase nedostatnim brojem primjeraka dodatno otežava proces njegova učenja. Povrh toga, neuravnoteženost klasa čini neprikladnima brojne tehnike za ublažavanje nepoželjnih učinaka učestalih unutarnjih karakteristika skupova podataka jer njihovo izvođenje može rezultirati narušenom izvedbom klasifikatora za manjinsku klasu.

Glavni cilj pri ublažavanju problema neuravnoteženosti klasa jest poboljšanje uspješnosti prepoznavanja manjinske klase koje u konačnici dovodi i do poboljšanja opće izvedbe klasifikatora. S obzirom na složenost i rasprostranjenost ovog problema, nije iznenađujuće da su predloženi razni pristupi koji ga nastoje ublažiti. Neki od njih temelje se na izmjeni proce-

dure treniranja klasifikacijskog modela ili načina vrednovanja njegove izvedbe i to primarno uvrštavanjem većeg troška za neprepoznavanje manjinskih primjeraka. S druge strane, pristupi na razini podataka jedini provode predobradu skupa podataka s ciljem ublažavanja stupnja neuravnoteženosti klasa i složenosti koncepta manjinske klase.

Odabir značajki i preuzorkovanje eksperimentalno su dokazali svoju učinkovitost u literaturi nebrojeno puta te se smatraju valjanim postupcima za poboljšanje učenja iz neuravnoteženih podataka. Oni u suštini izmjenjuju strukturu skupa podataka u svrhu postizanja lakše razdvojivosti primjeraka različitih klasa te njihove uravnoteženije raspodjele. Smanjenje složenosti skupa podataka ovim postupcima predobrade u konačnici pospješuje izvedbu raznih tipova klasifikatora, prvenstveno kako bi se olakšalo učenje koncepta manjinske klase. Međutim, iako ovi postupci značajno doprinose ublažavanju problema, ne mogu ga u potpunosti otkloniti. Stoga se pri odabiru klasifikatora ne trebaju zanemariti njegova svojstva pri učenju iz neuravnoteženih podataka.

3

Predobrada neuravnoteženih skupova podataka odabirom značajki

ODABIR značajki važan je postupak predobrade skupa podataka kojim se ostvaruje smanjenje dimenzionalnosti problema klasifikacije te poboljšava izvedba raznih klasifikatora. Ujedno se smatra i bitnim korakom za ublažavanje stupnja preklapanja klasa jer potencijalno uklanja one značajke čije vrijednosti dijeli velik broj primjeraka različitih klasa u skupu podataka. Provedba ovog postupka predobrade može uvelike doprinijeti uspješnosti prepoznavanja manjinske klase kod neuravnoteženih skupova podataka. Ovo poglavlje daje kratak osvrt na postojeće pristupe za odabir značajki, s posebnim naglaskom na omotače zasnovane na bio-inspiriranim algoritmima optimizacije koji su se pokazali valjanim pristupima za ovaj problem. Nakon pregleda literature, opisan je prijedlog proširenja za bio-inspirirane omotače koje nastoji poboljšati njihove performanse i stabilnost. Predloženo proširenje ujedno predstavlja prijedlog prvog izvornog znanstvenog doprinosa, a zasniva se na pohranjivanju raznolikih i kvalitetnih rješenja tijekom pretrage te njihovu naknadnom objedinjavanju. Pozitivan učinak predloženog proširenja eksperimentalno je ispitan za razne afirmirane bio-inspirirane omotače na standardnim skupovima podataka iz literature.

3.1 Uvod u odabir značajki

Jedan od početnih koraka pri izvođenju zadatka klasifikacije jest definiranje i izdvajanje značajki (engl. *feature extraction*) koje opisuju primjerke u skupu podataka. Broj takvih

značajki određuje dimenzionalnost problema klasifikacije, a njihove vrijednosti definiraju položaj primjeraka u ulaznom prostoru. U stvarnim problemima klasifikacije, značajke se obično određuju intuitivno ili pak pomoću raznih postupaka koji izdvajaju niz vrijednosti iz slika, teksta, zvuka, raznih signala i drugih izvora. Idealan postupak izdvajanja značajki izdvojio bi minimalan broj značajki na temelju kojih je moguće odrediti jasnu granicu između klasa. Takva raspodjela primjeraka čini zadatak klasifikatora trivijalnim i omogućuje mu postizanje visoke razine uspješnosti klasifikacije. Međutim, zbog nemogućnosti određivanja relevantnih značajki unaprijed, stvarni skupovi podataka u pravilu uključuju i značajke koje ne doprinose razlikovanju primjeraka različitih klasa ili ga čak i narušavaju uspostavljanjem raznih nepoželjnih unutarnjih karakteristika skupa podataka. Osim toga, nije neuobičajeno da se primjerci opisuju velikim brojem različitih značajki, što može dovesti do problema koji se kolokvijalno naziva "prokletstvo dimenzionalnosti" (engl. *curse of dimensionality*). Ovaj problem predočava činjenicu da povećanje dimenzionalnosti problema povlači potrebu za još većim rastom broja primjeraka u skupu podataka, kako bi se jasnije predstavili koncepti klasa u takvom prostoru. S ciljem izbjegavanja navedenih problema, nakon izdvajanja značajki obično se odabiru samo najprikladnije među njima, što je zadatak raznih pristupa za odabir značajki.

Cilj pristupa za odabir značajki jest odabrati relativno mali podskup dostupnih značajki prema određenom kriteriju, što obično dovodi do manje složenosti, lakše interpretabilnosti i kvalitetnije izvedbe treniranih klasifikacijskih modela [92]. Smanjivanjem dimenzionalnosti problema klasifikacije, smanjuje se i udaljenost postojećih manjinskih primjeraka u ulaznom prostoru, što olakšava učenje koncepta manjinske klase i poboljšava učinak metoda uzorkovanja [16, 82]. Osim toga, eliminiranjem onih značajki čije vrijednosti dijeli velik broj primjeraka različitih klasa ublažava se stupanj preklapanja klasa u skupu podataka, što uvelike doprinosi uspješnosti prepoznavanja manjinske klase, kako je ranije objašnjeno. Prema kriteriju odabira podskupa značajki, pristupe za odabir značajki moguće je podijeliti u četiri skupine [93].

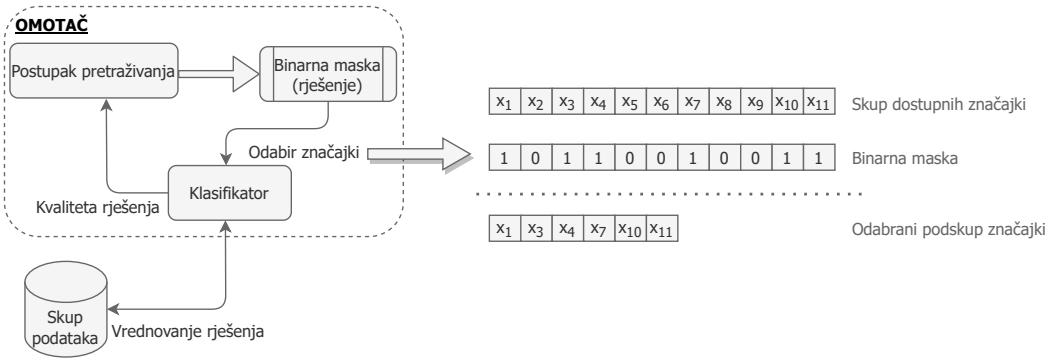
Pristupi iz skupine filtara rangiraju pojedine značajke na temelju mjera općih karakteristika podataka (poput dosljednosti, udaljenosti, količine informacije i korelacije [92]) te odabiru unaprijed zadani broj najbolje rangiranih značajki. Broj odabranih značajki jedan je od parametara takvog pristupa, a njegovo podešavanje nije trivijalan zadatak. Povrh toga, odabiranje isključivo najbolje rangiranih značajki ne podrazumijeva nužno i ostvarivanje najviše razine uspješnosti klasifikacije. Štoviše, rangiranjem značajki potencijalno se mogu odbaciti značajke koje su same po sebi slabo relevantne, ali u kombinaciji s drugima uvelike olakšavaju razlikovanje primjeraka različitih klasa [94]. Filtri odvajaju postupak odabira značajki od postupka treniranja klasifikacijskog modela izbjegavajući pri tome pristranost dobivenog podskupa značajki određenom klasifikatoru. Osim neovisnosti o klasifikatoru, prednost filtara je i manja vremenska složenost u odnosu na ostale pristupe. S druge strane, filtri ne uzimaju u obzir složene interakcije između značajki [95] te zanemaruju učinak dobi-

venog podskupa na uspješnost klasifikacije [96].

Pristupi iz skupine omotača prevladavaju nedostatke filtara koristeći performanse klasifikatora za određivanje kvalitete podskupa značajki. Oni u suštini predstavljaju mehanizam pretraživanja koji tretira klasifikator kao crnu kutiju ("omotava" ga) čije su performanse dio funkcije cilja pretrage. Osnovni koraci koje provodi omotač prikazani su na slici 3.1. Postupak pretraživanja predlaže rješenja koje klasifikator koristi za treniranje klasifikacijskog modela. Trenirani model zatim se vrednuje na odvojenom skupu podataka za vrednovanje (engl. *validation set*) i omotaču se vraća kvaliteta tog rješenja na temelju koje donosi daljnje odluke. Skup za vrednovanje obično se izvodi dodatnom podjelom skupa za treniranje postupkom izdvajanja ili postupkom unakrsne provjere k preklopa. Rješenja u prostoru pretrage uobičajeno se predstavljaju kao binarni vektori koji se rabe kao maska čije vrijednosti određuju zadržavanje ili odbacivanje pojedine značajke. Cilj omotača jest pronaći optimalni binarni vektor

$$\mathbf{q} = \underset{\mathbf{b}\in\{0,1\}^d}{\operatorname{argmax}} f(\mathbf{b}; \mathcal{T}, \mathcal{H}), \tag{3.1}$$

pri čemu f predstavlja funkciju cilja, d dimenzionalnost problema, $\mathcal{T}$ skup ulaznih podataka za treniranje, a $\mathcal{H}$ skup ulaznih podataka za vrednovanje klasifikacijskog modela [97]. Funkciju cilja najčešće čini neka od mjera uspješnosti klasifikacije ili pak složenija mjera koja uključuje i broj odabranih značajki. Iscrpno pretraživanje (engl. *exhaustive search*) jedini je postupak pretraživanja koji jamči pronalazak optimalnog podskupa značajki, ali uglavnom nije provediv zbog velikog broja mogućih rješenja $(2^d - 1)$ [94]. Omotači stoga obavljaju djelomično pretraživanje, koje je usmjereno oko podskupova značajki za koje odabrani klasifikator ostvaruje visoku razinu izvedbe. Jedan od najjednostavnijih omotača jest nasumična pretraga [98] koja uzastopno stvara nasumične podskupove značajki, a najkvalitetniji među njima odabire se kao konačno rješenje. Sustavniju pretragu provode omotači poput slijedne pretrage unaprijed (engl. *sequential forward selection*, SFS) i slijedne pretrage unazad (engl. *sequential backward selection*, SBS), koji u osnovi redom dodaju ili uklanjaju jednu po jednu značajku te popratno vrednuju kvalitetu izvedenog podskupa. Glavno ograničenje ovih omotača jest nemogućnost ponovnog vrednovanja korisnosti određene značajke u kasnijoj fazi, nakon što je donesena odluka o njezinu zadržavanju ili izbacivanju (tzv. "efekt gniježđenja" [99]). Ni složenije kombinacije ovih omotača [100] ne uspijevaju sasvim prevladati nepoželjne učinke spomenutog efekta, s obzirom na to da nije moguće ispitati sve varijacije redoslijeda uklanjanja i dodavanja pojedinih značajki. Omotači stoga sve češće zasnivaju svoj rad na upotrebi bio-inspiriranih algoritama optimizacije koji omogućavaju usmjereno pretraživanje velikog prostora pretrage. Ovi algoritmi smatraju se valjanim izborom za omotače jer zahvaljujući načinu obavljanja pretrage mogu otkriti složene interakcije između značajki [99, 101]. Omotači u pravilu pronalaze kvalitetnije podskupove značajki od ostalih pristupa, no uz rizik njihove pretjerane prilagodbe korištenom skupu za vrednovanje te klasifikatoru [75].



Slika 3.1: Shema rada omotača

Hibridni pristupi predloženi su kako bi se premostio jaz između filtara i omotača te ublažili nedostaci ovih dvaju pristupa [92]. Uobičajeno najprije provode rangiranje značajki prema određenom kriteriju kako bi odabrali unaprijed određen broj kandidata za podskupove značajki. Zatim sužavaju prostor pretrage omotača na rješenja koja uključuju prethodno odabrane značajke, pri čemu je pretraga vođena nekom od mjera uspješnosti klasifikacije. Filtri se unutar bio-inspiriranih omotača također mogu koristiti za inicijalizaciju početne populacije [102, 103], tako da se niti jedna značajka ne odbacuje potpuno, ali se favoriziraju one s većim rangom. Favoriziranjem visoko rangiranih značajki prema kriteriju utemeljenom na općim karakteristikama podataka, hibridni pristupi nastoje izbjeći pretjeranu prilagodbu rješenja klasifikatoru korištenom unutar omotača te tako formirati podskup značajki koji doprinosi izvedbi raznih tipova klasifikatora. Međutim, tijekom njihova rangiranja ne uzimaju se obzir složenije interakcije između značajki na temelju kojih je moguće ustanoviti da su odbačene značajke važne za izgradnju kvalitetnih klasifikacijskih modela. Hibridni pristupi su u pravilu složeniji od pojedinačnih pristupa za odabir značajki, a učinkovitost im je uvjetovana odabirom odgovarajuće kombinacije filtra, omotača i klasifikatora, čije utvrđivanje predstavlja značajan izazov.

Ugrađeni pristupi provode odabir značajki ukorak s treniranjem klasifikacijskog modela, odnosno ugrađuju ga u proces učenja. Primjerice, stabla odluke formiraju skup pravila za klasifikaciju na temelju kombinacije značajki koje maksimiziraju mjeru čistoće stabla. Razne inačice umjetnih neuronskih mreža (engl. *artificial neural networks*, ANNs) implicitno provode isključivanje onih značajki čiji su težinski faktori na sponama ulaznih podataka i čvorova skrivenog sloja postavljeni na nulu. U klasifikatore koji načelno ne provode odabir značajki moguće je ugraditi razne filtre ili omotače, kao što je to često slučaj kod klasifikatora SVM [104]. Ugradnja tih pristupa dodatno komplicira sam proces treniranja klasifikacijskog modela, a odabrani podskup značajki u pravilu je prilagođen skupu za treniranje te odabranom klasifikatoru.

Opsežniji pregled pristupa za odabir značajki može se pronaći u [93, 101]. Učinkovitost spomenutih pristupa ispitana je posebno za neuravnotežene skupove podataka u [105] te je pokazano da omotači ostvaruju bolje performanse klasifikacije od filtara i ugrađenih pristupa.

Razlog tomu je što velik broj filtara koristi mjere za rangiranje značajki koje ne uzimaju u obzir neuravnoteženost klasa [84], dok je kod omotača to jednostavno ostvariti prilagodbom funkcije cilja tako da pridaje veću važnost uspješnosti prepoznavanja manjinske klase. Omotači su jedni od najsloženijih pristupa za odabir značajki, no trošak njihova provođenja obično je opravdan s obzirom na njihovu sposobnost pronalaženja manjih i kvalitetnijih podskupova značajki. Postupak pretraživanja temeljna je komponenta omotača, pri čemu su brojni bio-inspirirani algoritmi optimizacije potvrđeni kao valjani pristupi za usmjereno istraživanje prostora pretrage [99, 106].

3.2 Bio-inspirirani algoritmi kao omotači

Odabir značajki zahtjevan je zadatak prvenstveno zbog velikog broja mogućih rješenja te postojanja složenih interakcija između samih značajki. Bio-inspirirani algoritmi kao omotači u pravilu pronalaze manje i kvalitetnije podskupove značajki od ostalih pristupa, upravo zbog njihove sposobnosti otkrivanja takvih interakcija [107]. Ovi algoritmi predstavljaju računalne implementacije raznih bioloških principa, a prema izvoru motivacije koju crpe moguće ih je grubo podijeliti na evolucijske algoritme (engl. *evolutionary algorithms*) te algoritme zasnovane na inteligenciji rojeva (engl. *swarm intelligence algorithms*) [108]. Ponajviše se primjenjuju za rješavanje složenih problema globalne optimizacije, kod kojih je prostor pretrage velik, a tradicionalne optimizacijske tehnike imaju sklonost zaglavljivanja u lokalnom optimumu. Bio-inspirirani algoritmi su prema svojoj prirodi stohastički algoritmi, a oslanjaju se isključivo na vrijednosti funkcije cilja povezane s rješenjima za usmjeravanje tijeka pretrage [109].

3.2.1 Pregled literature

Razvoj i primjena bio-inspiriranih algoritama vrlo je aktivno područje istraživanja [110], a mnoštvo je predloženih algoritama primijenjeno kao omotač. Siedlecki i Sklansky prvi su primijenili genetski algoritam (engl. *genetic algorithm*, GA) kao omotač u [111], koji je potom našao svoju primjenu u brojnim problemima klasifikacije, poput dijagnosticiranja raznih bolesti [112] te analizi genskog izražaja [113]. Nadalje, Yuan i Chu su predložili uporabu optimizacije rojem čestica (engl. *particle swarm optimisation*, PSO) kao omotača u [114], s ciljem poboljšanja izvedbe klasifikatora SVM koji je korišten za prepoznavanje kvarova na strojevima. Khushaba et al. su primijenili diferencijalnu evoluciju (engl. *differential evolution*, DE) za odabiranje podskupa značajki izvedenih iz elektroencefalograma u [115], dok su Marinakis et al. upotrijebili algoritam umjetne kolonije pčela (engl. *artificial bee colony*, ABC) kao omotač oko klasifikatora k-NN u [116] radi uspješnije procjene kreditnog rizika. Navedeni algoritmi smatraju se primarnim predstavnicima bio-inspiriranih algoritama, a ujedno predstavljaju i jedne od najpopularnijih izbora za omotače [99]. Na njihovu važnost

za problem odabira značajki ukazuje i velik broj problema klasifikacije u kojima su primijenjeni kao omotači [93], od kojih je nekoliko ranije spomenuto. Zasnovani su na populaciji mogućih rješenja, a primjenjuju razne operatore za stvaranje novih rješenja te selekciju za usmjeravanje pretrage prema obećavajućim područjima prostora. Navedeni algoritmi (osim GA) izvorno su predloženi za probleme kontinuirane optimizacije te je stoga potrebno provesti transformaciju njihovih rješenja u binarne vektore da bi bili primjenjivi kao omotači. To se uobičajeno postiže diskretizacijom rješenja na temelju praga ili pomoću sigmoidne funkcije [117].

Zbog svoje popularnosti, navedeni bio-inspirirani algoritmi predmet su brojnih izmjena ili proširenja koji u izvorne algoritme ugrađuju znanje o problemu odabira značajki, s ciljem pronalaženja kvalitetnijih podskupova značajki. Velik dio ovih izmjena odnosi se na sustavniji način inicijalizacije početne populacije, s obzirom da njezina struktura može imati značajan utjecaj na usmjeravanje pretrage. Kako bi početno usmjerili pretragu prema malim podskupovima značajki, Xue et al. su u [118] predložili unaprijeđenu inačicu algoritma PSO za odabir značajki, u kojoj većina (oko dvije trećine) rješenja u početnoj populaciji sadrži oko 10% nasumično odabranih značajki, a ostatak više od 50% značajki. Takav način inicijalizacije početne populacije korišten je i u [119, 120]. Uz to, predložili su i specifičan oblik selekcije prema kojoj se odabiru manji podskupovi značajki u slučaju rješenja jednake kvalitete. Nadalje, brojni omotači koriste filtre za utvrđivanje korisnosti pojedine značajke, na temelju koje određuju vjerojatnost njezina uključivanja u rješenja početne populacije. Time se nastoji ubrzati konvergencija, ali i usmjeriti pretraga prema rješenjima za koje klasifikator ostvaruje bolju sposobnost generalizacije. Apollini et al. su u [102] rangirali značajke prema filtru zasnovanom na informacijskoj dobiti (engl. *information gain*), dok su Hancer et al. koristili filtar *ReliefF* u [121] te Fisherovu mjeru (engl. *Fisher score*) u [122] za omotač zasnovan na algoritmu DE. Zorić et al. dodatno su proširili ovaj koncept u [103], uvodeći drugačiju selekciju značajki na temelju njihova ranga, koja pruža više prilika pojedinoj značajki da bude uključena u rješenje početne populacije. U suštini, predložene tehnike za inicijalizaciju početne populacije uglavnom se zasnivaju na oblikovanju rješenja prema njihovoj veličini te na korištenju filtara za favoriziranje obećavajućih značajki. Sažet pregled ovih tehnika se može naći u [123].

U bio-inspirirane omotače često se ugrađuju dodatni operatori u obliku lokalne pretrage, prvenstveno s ciljem poboljšanja njihove sposobnosti konvergencije [124] te izbjegavanja njihova zaglavljivanja u lokalnom optimumu [125]. Radi ubrzanja konvergencije omotača zasnovanom na GA, Oh et al. su u [124] uveli operatore koji provode lokalnu pretragu nad svim stvorenim rješenjima, a zasnivaju se na operacijama koje uklanjaju značajke koje samostalno najmanje doprinose izvedbi klasifikatora, odnosno uključuju značajke koje samostalno ponajviše doprinose uspješnosti klasifikacije. S istim ciljem, Nguyen et al. su u [126] nakon svake iteracije algoritma PSO provodili slijednu pretragu unazad nad najboljem pronađenom rješenju, dok su Jeong et al. u [127] zamijenili mutaciju algoritma GA plutajućom

slijednom pretragom unaprijed (engl. *sequential floating forward selection*, SFFS) radi poboljšanja svakog rješenja stvorenog tijekom pretrage. Hancer je u [128] nakon svake iteracije algoritma DE provodio lokalnu pretragu nad najboljim rješenjem u populaciji uzastopnim izvođenjem slijedne pretrage unaprijed i slijedne pretrage unazad, pri čemu su doprinosi pojedinih značajki određeni pomoću raznih filtara. U literaturi je predloženo još mnoštvo inačica navedenih bio-inspiriranih omotača koji ugrađuju složenije mehanizme u svoj rad, poput grupiranja podataka [129, 130], koncepte teorije grubih skupova [131] i druge. Detaljniji pregled takvih bio-inspiriranih omotača dan je u [99, 125].

Uvođenje mehanizama specifičnih za odabir značajki u bio-inspirirane omotače u manjoj je mjeri prisutno u literaturi u odnosu na ugrađivanje raznih dodatnih operatora koji su ranije opisani. Ipak, u svrhu izbjegavanja zaglavljivanja u lokalnom optimumu, Chuang et al. su u [132] ponudili pristup resetiranju najboljeg pronađenog rješenja u algoritmu PSO u slučaju da je ono nepromijenjeno nakon nekoliko iteracija, tako da na nulu postavljaju sve komponente vektora koji ga predstavljaju. Yang et al. su proširili ovu ideju u [133] na način da pri istim okolnostima stvaraju novo najbolje rješenje primjenom raznih logičkih operacija između susjednih komponenti binarnog vektora koji predstavlja najbolje nađeno rješenje. Nadalje, Benitez et al. su u [134] predložili novi operator križanja u algoritmu GA, prema kojem broj uključenih značajki oba rješenja između točaka prekida kontrolira proces njihove zamjene. Ovakav način križanja održava veću raznolikost populacije jer se rješenja dobivena njime znatno razlikuju u broju uključenih značajki. Uz utjecanje na njihov unutarnji način rada, bio-inspirirani omotači mogu biti prošireni vanjskim mehanizmima koji služe kao podrška pri izboru konačnog rješenja nakon same pretrage. Martinović et al. su tako u [97] proširili omotač zasnovan na algoritmu DE s arhivom kvalitetnih rješenja prikupljenih tijekom pretrage iz koje se dodatnim vrednovanjem odabire podskup značajki koji najviše doprinosi sposobnosti generalizacije korištenog klasifikatora.

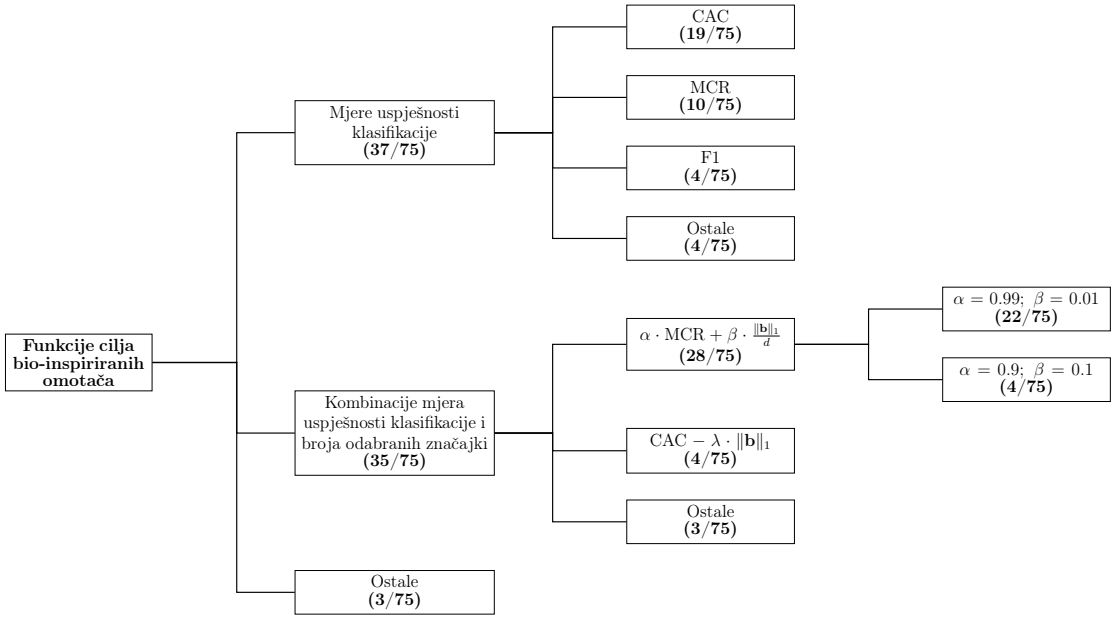
Trend predlaganja novih bio-inspiriranih algoritama jasno je vidljiv u literaturi, a njihov broj je pozamašan [135]. Često se uvode neobične metafore kako bi se opisali postupci pretraživanja (primjerice, imitiranje fenomena crne rupe [136], ponašanje kitova [137], ponašanje pingvina [138] i slično). Ovakav pristup razvoju algoritama usmjeren metaforama naišao je na odgovarajuće kritike [110, 139, 140], budući da većina naglašava svoju inovativnost i nadmoćnost u odnosu na prethodne algoritme, bez iznošenja odgovarajućih popratnih dokaza. Štoviše, velik udio novijih algoritama predstavlja posebne inačice ranije izloženih bio-inspiriranih algoritama te u suštini ne uključuju inovativne operatore za pretraživanje [141]. Ipak, značajan broj radova u literaturi predlaže uporabu ovih algoritama kao omotača, premda u njihov način rada ne uvode ništa specifično za sam problem odabira značajki. Broj takvih radova gotovo je jednak broju predloženih metafora, a iz njih je teško razaznati po čemu se noviji algoritmi ističu u odnosu na starije kada se uzmu u obzir ostvarene performanse i pokazano ponašanje za ovaj problem. Štoviše, rezultati raznih eksperimentalnih usporedbi [106, 142] ne idu u korist novijim bio-inspiriranim algoritmima kao omotačima

za odabir značajki, dok se kao primjereni izbor za omotače potvrđuju afirmirani algoritmi poput DE i GA.

3.2.2 Kritički osvrt

Poznato je da bio-inspirirani algoritmi općenito zahtijevaju velik broj vrednovanja za pronalaženje kvalitetnih rješenja, s obzirom na to da se oslanjaju isključivo na vrijednosti funkcije cilja za usmjeravanje tijeka pretrage. Primjetno je međutim kako je bio-inspiriranim omotačima u literaturi često dopušten značajno manji broj vrednovanja u usporedbi s brojem vrednovanja koji je omogućen istim algoritmima kada se koriste za druge probleme optimizacije približno jednake dimenzionalnosti. Ova okolnost posebice je izražena u radovima koji predlažu primjenu novijih bio-inspiriranih algoritama kao omotača. Kao jedan od mogućih razloga iza ove okolnosti može se smatrati činjenica da s povećanjem broja vrednovanja bio-inspiriranog omotača raste mogućnost pretjerane prilagodbe nađenih rješenja skupu podataka korištenom za njihovo vrednovanje [143]. Za pretjerano prilagođeni podskup značajki, klasifikator unutar omotača ostvaruje vrlo kvalitetnu izvedbu na skupu za vrednovanje, ali ujedno ima slabu sposobnost generalizacije. Ipak, uporabu malog broja vrednovanja teško je opravdati nastojanjem ublažavanja pretjerane prilagodbe rješenja, posebice zbog činjenice da prostor pretrage koji istražuju bio-inspirirani omotači u pravilu sadrži pozamašan broj mogućih rješenja ($2^d - 1$). Potencijalni način ublažavanja ove nepoželjne posljedice mogao bi biti korištenje drugačijeg načina vrednovanja unutar omotača, poput unakrsne provjere k preklopa [144]. Međutim, ovaj način vrednovanja ima veću vremensku složenost, a njegovi učinci na izbjegavanje ovog problema su upitni [75]. Pristup za ublažavanje pretjerane prilagodbe ponudili su Martinović et al. u [97], gdje se konačno rješenje odabire na temelju dodatnog vrednovanja arhive kvalitetnih rješenja prikupljenih tijekom pretrage i to drugačijim načinom vrednovanja u odnosu na onaj korišten u omotaču. Dodatan problem koji se može zamijetiti u brojnim radovima je taj što iz skupa podataka ne izdvajaju skup za testiranje prethodno odabiru značajki, nego ispituju kvalitetu konačnog rješenja nad istim skupom koji je korišten za vrednovanje tijekom same pretrage. Ovakav način testiranja onemogućuje uvid u sposobnost generalizacije klasifikatora za nađeni podskup značajki te pruža pretjerano optimističnu predodžbu izvedbe korištenog klasifikatora.

Problem odabira značajki u suštini uključuje dva glavna cilja, a to su poboljšanje uspješnosti klasifikacije i smanjenje broja značajki. Prvi cilj je u pravilu važniji, no kod nekih se problema odabir značajki prvenstveno primjenjuje u svrhu smanjenja dimenzionalnosti (primjerice, kod problema analize genskog izražaja [102]). S obzirom na to da ova dva cilja mogu biti sukobljena [99], odabir značajki može se tretirati i kao problem višeciljne optimizacije sa svrhom pronalaženja skupa nedominiranih rješenja po oba kriterija [145]. Ipak, velik udio pristupa iz skupine omotača svode problem višeciljne optimizacije na jedinstvenu funkciju cilja koja je predstavljena konveksnom kombinacijom uspješnosti klasifikacije i ve-



Slika 3.2: Zastupljenost funkcija cilja bio-inspiriranih omotača

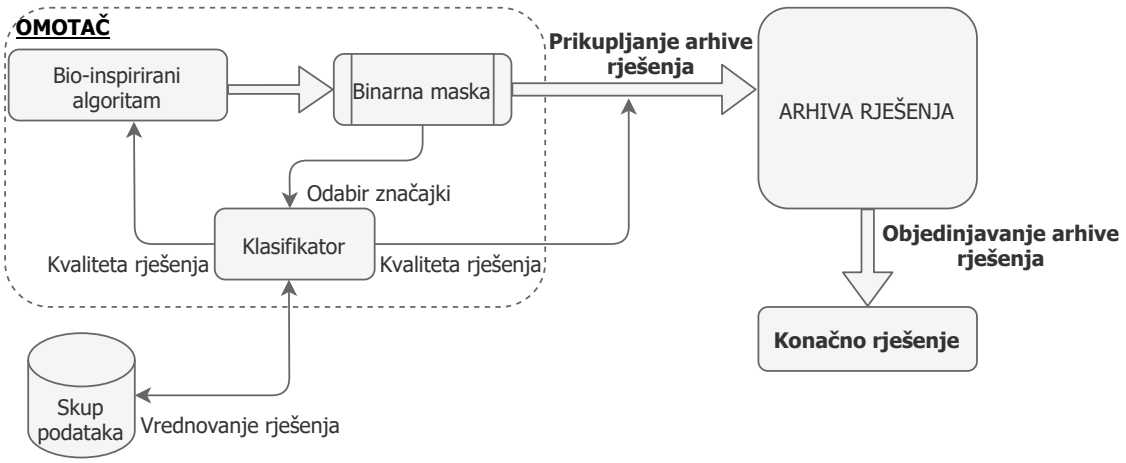
ličine podskupa značajki. Određivanje njihovih težina u konveksnoj kombinaciji izazovan je zadatak te bi se ono trebalo zasebno provoditi za svaki problem. Ipak, pregledom literature moguće je primijetiti da se često koriste iste težine za različite algoritme i skupove podataka. Uvid u to pruža slika 3.2 u kojoj su izvedene najčešće funkcije cilja kod 75 bio-inspiriranih omotača predloženih u literaturi. Uz točnost (CAC) i grešku (MCR) klasifikacije, uvelike se upotrebljava i konveksna kombinacija mjere MCR i normaliziranog broja odabranih značajki s težinama $\alpha = 0.99$ te $\beta = 0.01$, pri čemu se provodi minimizacija takve funkcije cilja. Katkada funkciju cilja predstavlja točnost klasifikacije umanjena za veličinu rješenja (koja se izražava kao $\|\mathbf{b}\|_1$, gdje $\|\cdot\|_1$ predstavlja ℓ_1 normu), prethodno pomnoženim s članom kazne λ čije određivanje pruža jednak izazov kao i određivanje težina α i β. Mjere uspješnosti klasifikacije poput F1, AUC i $\mathrm{G}_{\mathrm{mean}}$, koje su ujedno i prikladnije za rad s neuravnoteženim skupovima podataka, koriste se znatno rijeđe. Stopa smanjenja dimenzionalnosti svakako je važan pokazatelj učinkovitosti bio-inspiriranog omotača, no manje izravan pritisak na pronalaženje malih podskupova značajki moguće je ostvariti drugačijim načinom inicijalizacije početne populacije ili pak posebnim oblikom selekcije koja favorizira manja rješenja, kao što je ranije opisano (primjerice, kao što su predložili Xue et al. u [118]).

Problem odabira značajki ujedno je i multimodalan problem, što znači da je moguća pojava rješenja jednake kvalitete tijekom pretrage, koja se pak mogu značajno razlikovati po svojoj strukturi. U slučaju da na kraju pretrage postoji nekoliko različitih rješenja koja dijele kvalitetu, bio-inspirirani omotači u konačnici vraćaju samo jedno među njima. Osim što se na taj način mogu odabrati rješenja koja slabije doprinose sposobnosti generalizacije klasifikatora, ovakvo stohastičko ponašanje bio-inspiriranih omotača čini ih nestabilnim pristupima za odabir značajki. Stabilnost pristupa za odabir značajki može se promatrati kao konzis-

tentnost u pronalaženju podskupova značajki slične strukture uslijed preslagivanja skupa podataka [146]. Budući da se odabirom značajki nastoji steći uvid u relevantnost značajki i u njihove interakcije, preferiraju se stabilniji pristupi koji neznatno mijenjaju strukturu pronađenog rješenja pri raznim varijacijama ulaznih podataka. S obzirom na njihov deterministički način rada, pristupi iz skupine filtara te heuristički postupci pretraživanja kao omotači (SFS, SBS i njihove kombinacije) za istu podjelu skupa podataka uvijek vraćaju isti podskup značajki, no uslijed malih perturbacija u podacima mogu vratiti značajno različita rješenja [147, 148]. Zbog svoje stohastičke prirode, bio-inspirirani omotači nerijetko pronalaze različita rješenja višestrukim izvođenjem pretrage za istu podjelu skupa podataka [99, 106]. Problem nestabilnosti bio-inspiriranih omotača u literaturi je često spominjan [99, 101, 149], ali slabo dotaknut. Štoviše, većina radova koji razmatraju ovaj problem rabi standardnu devijaciju kvalitete rješenja u svrhu prikazivanja stabilnosti korištenog omotača. S obzirom na multimodalnost problema odabira značajki, ova mjera može pogrešno upućivati na visoku razinu stabilnosti omotača unatoč tome što on može pronaći različita rješenja svakim izvođenjem pretrage. Malobrojne tehnike uvećanja stabilnosti bio-inspiriranih omotača u literaturi temelje se na provedbi objedinjavanja podskupova značajki prikupljenih tijekom više izvođenja, pri čemu je ono zasnovano na operacijama poput presjeka, unije ili glasanja [147, 150, 151]. Međutim, spomenute tehnike podrazumijevaju višestruko izvođenje pretrage, što uvelike povećava trošak upotrebe omotača.

3.3 Prijedlog proširenja bio-inspiriranih omotača zasnovanog na arhivi rješenja

Bez obzira na spomenute nedostatke, bio-inspirirani omotači ističu se kao valjani i učinkoviti pristupi odabiru značajki. Ipak, ublažavanjem njihovih nedostataka moguće im je dodatno poboljšati performanse i stabilnost. S tim ciljem, kao izvorni znanstveni doprinos predlaže se proširenje bio-inspiriranih omotača zasnovano na arhivi rješenja koja se prikupljaju tijekom pretrage. Shema predloženog proširenja prikazana je na slici 3.3, a njegova dva osnovna dijela jesu prikupljanje rješenja te postupak njihova objedinjavanja. U prvom dijelu, rješenja koja se vrednuju tijekom pretrage se po svojoj kvaliteti natječu za ulazak u arhivu ograničene veličine. Arhiviranjem raznolikih i kvalitetnih rješenja izbjegava se oslanjanje isključivo na rješenje nađeno omotačem jer ono može biti pretjerano prilagođeno skupu za vrednovanje. Osim toga, održavanjem arhive takvih rješenja olakšava se naknadno stjecanje uvida u relevantnost pojedinih značajki te u interakcije između njih. Nakon prikupljanja rješenja, izdvajaju se njihove zajedničke značajke te se u tako dobiveni podskup značajki postupno uvode nove značajke prema njihovu doprinosu kvaliteti. Na taj način nastoji se povećati stabilnost omotača te formirati podskup značajki na temelju kojih klasifikator ostvaruje bolju sposobnost generalizacije. Konačno formirano rješenje koje objedinjuje svojstva svih rješe-



Slika 3.3: Shema predloženog proširenja bio-inspiriranih omotača

nja u arhivi može služiti kao alternativa rješenju koje je ponudio sam omotač. Budući da ugradnja ovakvog proširenja ne ometa sam proces pretraživanja izvornog omotača, moguće ga je ugraditi u različite bio-inspirirane omotače.

3.3.1 Prikupljanje arhive rješenja

Kao što je ranije spomenuto, bio-inspirirani algoritmi zahtijevaju velik broj vrednovanja za pronalaženje kvalitetnih rješenja. S obzirom na to da je pretraga bio-inspiriranih omotača vođena kvalitetom rješenja ostvarenom na skupu za vrednovanje, s porastom broja vrednovanja raste i mogućnost pretjerane prilagodbe konačnog rješenja tom skupu. Takvo rješenje može sadržavati nevažne ili kontraproduktivne značajke koje narušavaju sposobnost generalizacije klasifikatora ili pak može odbacivati neke relevantne značajke za promatrani problem klasifikacije. Međutim, na temelju jednog rješenja teško je odrediti koje značajke su relevantne, a koje su uključene ili odbačene kao rezultat njegove pretjerane prilagodbe skupu za vrednovanje. S druge strane, zbog multimodalnosti problema odabira značajki, postoji mogućnost da omotač tijekom pretrage pronađe nekoliko rješenja koja se neznatno razlikuju po kvaliteti, ali imaju različitu strukturu. Prikupljanjem takvih rješenja može se stvoriti određeno znanje o samom problemu klasifikacije, koje se zasniva na pretpostavci da se kvalitetna rješenja podudaraju u većini relevantnih značajki za promatrani problem. Međutim, po završetku pretrage obično nije moguće steći uvid u njezin tijek, odnosno u strukturu i kvalitetu svih rješenja koja su vrednovana. Stoga se u okviru prijedloga njihova proširenja predlaže održavanje arhive različitih rješenja koja se prikupljaju tijekom pretrage, a ističu se po svojoj kvaliteti.

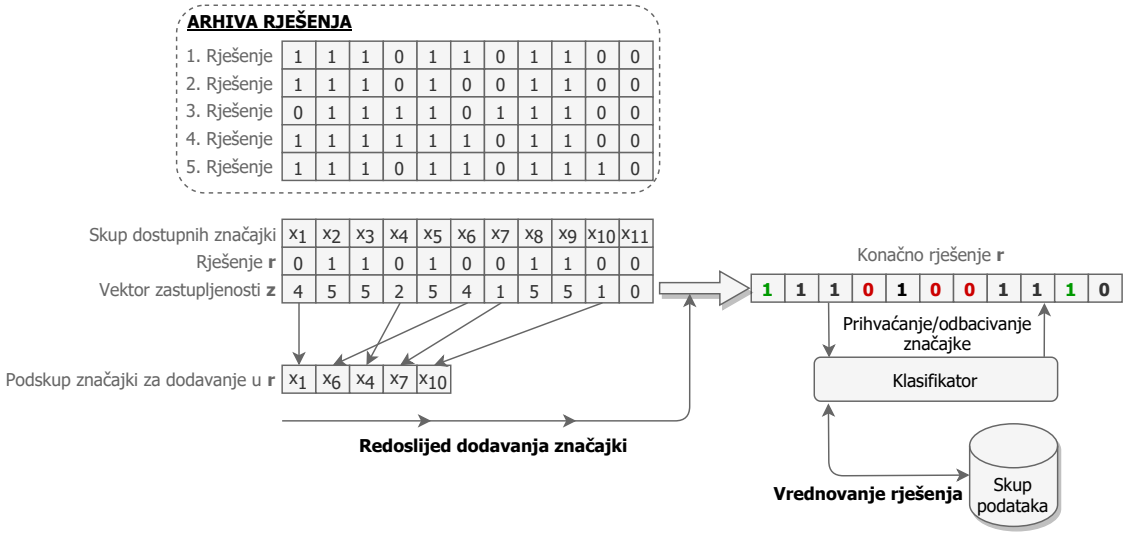
Koncept arhiviranja različitih rješenja tijekom pretrage proizlazi iz područja višeciljne optimizacije, gdje se u arhivu pohranjuju nedominirana rješenja po danim kriterijima. Kod bio-inspiriranih omotača s jedinstvenom funkcijom cilja, arhiviranje kvalitetnih i raznolikih rješenja predloženo je tek u [97], gdje se njihovim dodatnim vrednovanjem pomoću unakrsne

provjere k preklopa izabire ono rješenje koje najviše doprinosi generalizaciji klasifikatora. Ipak, u tom pristupu se ne provodi objedinjavanje arhive rješenja, koje se pak u predloženom proširenju izvršava s ciljem identificiranja relevantnih značajki za problem klasifikacije te povećanja stabilnosti bio-inspiriranog omotača. Osim toga, pri velikom broju vrednovanja unutar omotača, rješenja u arhivi mogu biti pretjerano prilagođena skupu za vrednovanje i neznatno se razlikovati po doprinosu sposobnosti generalizacije klasifikatora. Predloženim postupkom objedinjavanja nastoji se stvoriti novo rješenje koje klasifikatoru omogućuje veću sposobnost generalizacije od ostalih rješenja unutar arhive.

3.3.2 Objedinjavanje arhive rješenja

Nakon prikupljanja arhive rješenja, pristupa se njihovu objedinjavanju s ciljem stvaranja novog rješenja od kojeg se očekuje povećanje sposobnosti generalizacije klasifikatora i stabilnosti omotača. S obzirom na to da arhiva sadrži najkvalitetnija rješenja nađena tijekom pretrage, izvlačenjem njihovih zajedničkih karakteristika nastoje se identificirati relevantne značajke, odnosno njihove kombinacije za promatrani problem klasifikacije. Pri tome, relevantnost značajki određuje se na temelju njihove zastupljenosti u rješenjima arhive. U literaturi je predloženo nekoliko postupaka objedinjavanja rješenja u svrhu povećanja stabilnosti omotača [151], gdje se pomoću operacije presjeka, unije ili glasanja između većeg broja rješenja nađenih omotačem formira konačni podskup značajki. Međutim, ti postupci zahtijevaju višestruko izvođenje pretrage radi prikupljanja tih rješenja, što značajno povećava trošak uporabe omotača. Povrh toga, formirani podskup značajki može biti jednak praznom skupu (u slučaju izvođenja presjeka) ili pak sadržavati sve dostupne značajke (u slučaju izvođenja unije), što u pravilu nisu prihvatljiva rješenja za problem odabira značajki.

Kako bi se u predloženom postupku objedinjavanja izbjegli navedeni problemi, u rješenje koje sadrži zajedničke značajke rješenja arhive postupno se uvode i ostale značajke. Dodavanje svake značajke popraćeno je vrednovanjem novonastalog rješenja, na temelju kojeg se donosi odluka o njezinu zadržavanju. Sličan princip formiranja podskupa značajki provodi i slijedna pretraga unaprijed, koja pak započinje dodavanje od praznog skupa. Uz to, u sklopu predloženog postupka objedinjavanja, značajke se ne uvode redom, već na temelju njihove zastupljenosti u rješenjima arhive. Redoslijed dodavanja preostalih značajki, odnosno onih koje nisu zajedničke svim rješenjima, može utjecati na kvalitetu formiranog rješenja zbog postojanja složenih interakcija između njih. S obzirom na to da pri postupnom dodavanju nije moguće ponovno vrednovati korisnost pojedine značajke nakon što je donesena odluka o njezinu odbacivanju ili zadržavanju, u predloženom načinu objedinjavanja prvotno se uvode zastupljenije značajke s pretpostavkom da je zastupljenost značajke proporcionalna njezinu doprinosu kvaliteti. Da bi se pružio uvid u sposobnost generalizacije klasifikatora nakon dodavanja nove značajke, formirano rješenje vrednuje se drugačijim načinom od onog korištenog unutar omotača. Konkretnije, provodi se unakrsna provjera k preklopa s obzirom na



Slika 3.4: Shema načina objedinjavanja rješenja unutar arhive

to da se ovim načinom vrednovanja potencijalno može ublažiti pretjerana prilagodba rješenja skupu za vrednovanje, kao što je ranije opisano.

Predloženi postupak objedinjavanja arhive rješenja ilustriran je slikom 3.4. Razvidno je da predloženi postupak izvodi dodatna vrednovanja funkcije cilja kako bi utvrdio doprinos pojedinih značajki prilikom njihova dodavanja. U slučaju da operacija presjeka rezultira praznim skupom, pri čemu je svaka značajka sadržana u barem jednom rješenju arhive, potrebno je ispitati uvođenje svih značajki u konačno rješenje, što zahtijeva d dodatnih vrednovanja funkcije cilja. Ipak, s obzirom na objedinjavanje stečenog znanja u obliku arhive rješenja te uporabu drugačijeg načina vrednovanja u odnosu na onaj u omotaču, očekuje se da će predloženo proširenje moći formirati kvalitetniji podskup značajki u odnosu na omotač kojem se omogući dodatnih d vrednovanja funkcije cilja.

3.3.3 Detalji ugradnje

Predloženi način prikupljanja arhive rješenja prikazan je algoritmom 3.1. Sva rješenja koja se vrednuju tijekom pretrage u omotaču se po svojoj kvaliteti natječu za ulazak u arhivu. Pri tome se unutar arhive održavaju rješenja jedinstvene strukture koja su sortirana prema svojoj kvaliteti kako bi se smanjila vremenska složenost dodavanja novog rješenja. Na samom početku pretrage, arhiva sadrži rješenja početne populacije. S obzirom na to da je veličina arhive ograničena na veličinu populacije korištenog bio-inspiriranog algoritma, novo rješenje zamjenjuje najlošije rješenje u arhivi (koje se nalazi na zadnjem mjestu u arhivi, odnosno označeno je kao $A[N_A]$) ako je kvalitetnije od njega. Ako je pak njihova kvaliteta jednaka, u arhivu ulazi (ili ostaje) ono rješenje koje sadrži manje značajki, kako bi se neizravno pridonijelo cilju smanjenja dimenzionalnosti.

Predloženi postupak objedinjavanja arhive rješenja prikazan je algoritmom 3.2. Početno se izvodi operacija presjeka nad komponentama binarnih vektora sadržanih u arhivi, da bi

Algoritam 3.1: Nacrt rada prikupljanja arhive rješenja

```
Funkcija Arhiviraj():
    Ulaz: A[N_A][d]      // Arhiva rješenja
          F[d]           // Kvalitete rješenja u arhivi (silazno sortirane)
          b[d]           // Novo rješenje
          f_b            // Kvaliteta novog rješenja

    ako A ne sadrži b onda
        ako f_b > F[N_A] ILI f_b = F[N_A] I b ima manje značajki od A[N_A] onda
            Ubaci b u A prema kvaliteti f_b
            Ubaci f_b na odgovarajuće mjesto u F
            Izbaci najlošije rješenje iz A
            Izbaci najmanju kvalitetu iz F
        kraj ako
    kraj ako
```

Algoritam 3.2: Nacrt rada objedinjavanja arhive rješenja

```
Funkcija Objedini():
    Ulaz: A[N_A][d]      // Arhiva rješenja
    Izlaz: r[d]          // Konačno rješenje

    r[d] := [1, .., 1]
    z[d] := [0, .., 0]   // Niz zastupljenosti pojedinih značajki u A
    za svaki a u A čini
        r := r · a       // Računanje presjeka arhive rješenja
        z := z + a       // Računanje zastupljenosti značajki
    kraj za svaki
    Sortiraj z i r silazno prema z
    za i := 1, ..., d čini
        ako z[i] ≠ d I z[i] ≠ 0 onda
            s := r
            r[i] := 1
            ako f(s; T*, V*) ≥ f(r; T*, V*) onda
                r[i] := 0
            kraj ako
        kraj ako
    kraj za
```

se utvrdile zajedničke značajke svih rješenja. S obzirom na to da se ovim postupkom mogu odbaciti neke relevantne značajke koje pak nisu sadržane u svim rješenjima, u formirano rješenje se postupno uvode preostale značajke. Pri tome je redoslijed njihova dodavanja određen njihovom zastupljenosti u rješenjima unutar arhive. Ako dodavanje značajke ne uzrokuje povećanje kvalitete formiranog rješenja, ona se odbacuje. Funkcija cilja i klasifikator koji su korišteni za vrednovanje rješenja unutar omotača preuzimaju se i u predloženom postupku objedinjavanja. Vrednovanje formiranog rješenja provodi se podjelom skupa za treniranje postupkom unakrsne provjere korištenjem pet preklopa, pri čemu se preklopi općenito razlikuju u odnosu na one korištene tijekom pretrage, ako je unutar omotača korišten isti broj preklopa. Ovaj postupak je vremenski zahtjevniji od postupka izdvajanja, no uz relativno mali broj vrednovanja njegovo provođenje ne podrazumijeva značajan utrošak vremena. Osim toga, njime se potencijalno može ostvariti bolji uvid u sposobnost generalizacije klasifikatora, posebice jer se radi o općenito drugačijem načinu vrednovanja od onog korištenog unutar omotača.

3.3.4 Procjena vremenske složenosti

S obzirom na činjenicu da se predloženo proširenje nadograđuje na uobičajeni postupak pretrage bio-inspiriranog omotača, moguće je zaključiti da ono povećava trošak njegova izvođenja. Kako bi se stekao uvid u razmjer tog povećanja, u nastavku je dana procjena vremenske složenosti predloženog proširenja. Ona se može razložiti na pojedinačne složenosti funkcija *Arhiviraj()* i *Objedini()*. Složenost obje funkcije može se opisati kao $O(N_A \times d)$. Unutar funkcije *Arhiviraj()* prvotno se vrši provjera nalazi li se novo rješenje već unutar arhive $[O(N_A \times d)]$, što predstavlja najsloženiju operaciju te funkcije. Pri dodavanju novog rješenja u arhivu ono se uspoređuje s najgorim rješenjem u arhivi $[O(d)]$ te se ubacuje na odgovarajuće mjesto prema kvaliteti $[O(N_A)]$. Treba podsjetiti da se ova funkcija poziva unutar bio-inspiriranog omotača nakon svakog vrednovanja rješenja tijekom pretrage. Nadalje, u funkciji *Objedini()* prvotno se izvodi presjek svih rješenja unutar arhive $[O(N_A \times d)]$, a potom se ispituju doprinosi ostalih značajki, gdje je u najgorem slučaju potrebno ispitati ubacivanje svih značajki u konačno rješenje $[O(d)]$.

Iako je vremenska složenost predloženog proširenja na istoj razini kao i složenost samog omotača, vremenski najzahtjevnija operacija koju ono izvodi je postupak vrednovanja rješenja. Vrednovanje rješenja izvodi se najviše d puta, pri čemu se skup za treniranje, prethodno objedinjavanju arhive rješenja, dijeli postupkom unakrsne provjere korištenjem pet preklopa. Ipak, s obzirom na činjenicu da je bio-inspiriranim omotačima potreban znatno veći broj vrednovanja od dimenzionalnosti problema, dodatnih d vrednovanja ne bi trebalo osjetno produžiti njihovo izvođenje, a ono može biti opravdano ako rezultira povećanjem njihovih performansi. Osim toga, moguće je očekivati da će neke značajke biti sadržane ili odbačene u svim rješenjima arhive te se one stoga ne trebaju ispitati za ubacivanje u konačno rješenje.

3.4 Eksperimentalna analiza i rezultati

Da bi se utvrdila korisnost predloženog proširenja bio-inspiriranih omotača, provedena je odgovarajuća eksperimentalna analiza koja je podijeljena u tri dijela. Kao što je ranije rečeno, primarni cilj proširenja jest poboljšati sposobnost generalizacije klasifikatora korištenog u omotaču, dok je dodatan cilj povećati stabilnost samog omotača uslijed višestrukog izvođenja pretrage te preslagivanja skupa podataka. U prvom dijelu analiziran je predloženi način objedinjavanja arhive rješenja kako bi se utvrdio njegov doprinos navedenim performansama proširenja. Drugi dio analize prikazuje utjecaj predloženog proširenja na standardne omotače u literaturi, odnosno one zasnovane na istaknutim predstavnicima bio-inspiriranih algoritama optimizacije korištenim u svom uobičajenom obliku. U trećem dijelu analize razmotren je utjecaj proširenja na unaprijeđene omotače iz literature koji su zasnovani na istim bio-inspiriranim algoritmima kao i standardni omotači, no u koje su ugrađeni mehanizmi

Tablica 3.1: Karakteristike skupova podataka korištenih za potrebe eksperimentalne analize predloženog proširenja bio-inspiriranih omotača

Oznaka	Naziv	Broj primjeraka	Broj značajki	Broj klasa	IR
$\mathcal{D}_1$	QSAR Biodegradation	1055	41	2	1.96
$\mathcal{D}_2$	Breast Cancer Wisconsin	569	30	2	1.68
$\mathcal{D}_3$	Clean2	6598	166	2	5.49
$\mathcal{D}_4$	Climate	540	18	2	10.74
$\mathcal{D}_5$	German Credit Data	1000	61	2	2.33
$\mathcal{D}_6$	Ionosphere	351	34	2	1.79
$\mathcal{D}_7$	Madelon	2600	500	2	1.00
$\mathcal{D}_8$	Image Segmentation	210	19	7	1.00
$\mathcal{D}_9$	Parkinsons	195	22	2	3.06
$\mathcal{D}_{10}$	LSVT Voice Rehabilitation	126	310	2	2.00
$\mathcal{D}_{11}$	Urban Land Cover	675	147	9	2.19
$\mathcal{D}_{12}$	Wine	178	13	3	1.30

specifični za problem odabira značajki.

Eksperimentalna analiza provedena je na standardnim skupovima podataka za vrednovanje novopredloženih postupaka za odabir značajki u literaturi, a njihove karakteristike prikazane su u tablici 3.1. Ovi skupovi podataka preuzeti su s UCI repozitorija [152], a predstavljaju razne probleme klasifikacije opisane u dodatku A koji se, osim po prirodi, razlikuju po dimenzionalnosti te po stupnju neuravnoteženosti klasa. U drugom dijelu analize također je pružen poseban osvrt na korisnost odabira značajki i predloženog proširenja na neuravnoteženim problemima, pri čemu su kao predstavnici takvih problema odabrani oni skupovi podataka koji imaju IR ≥ 2, odnosno skupovi $\mathcal{D}_3$, $\mathcal{D}_4$, $\mathcal{D}_5$, $\mathcal{D}_9$, $\mathcal{D}_{10}$ i $\mathcal{D}_{11}$. Uz to, tri odabrana skupa podataka ($\mathcal{D}_8$, $\mathcal{D}_{11}$ i $\mathcal{D}_{12}$) predstavljaju probleme višeklasne klasifikacije kako bi se ispitala učinkovitost proširenja i na takvim problemima. Stupanj neuravnoteženosti ovih problema izražen je kao prosječan stupanj neuravnoteženosti za $\binom{m}{2}$ nezavisnih binarnih problema koji su izvedeni shemom dekompozicije OVO. Skaliranje značajki, normalizacijom u raspon $[0, 1]$, izvedeno je kao korak predobrade svakog skupa podataka kako bi se ublažio utjecaj različitih raspona vrijednosti, a postupak je opisan u dodatku A.

3.4.1 Postavke eksperimenta

Osnovni preduvjet za provođenje eksperimentalne analize jest nadogradnja nekoliko bio-inspiriranih omotača iz literature predloženim proširenjem s ciljem uspoređivanja njihovih performansi prije i nakon nadogradnje. Kao što je ranije navedeno, algoritmi GA, DE i PSO su se u nekoliko eksperimentalnih analiza pokazali valjanim izborom za omotače zbog zadovoljavajućih performansi. Uz to, ovi algoritmi mogu se smatrati standardnim omotačima u literaturi na osnovu njihove zastupljenosti te su stoga korišteni u prvom i drugom dijelu eksperimentalne analize. Korištene postavke parametara ovih omotača preuzete su iz eksperimentalne analize provedene u [106], a prikazane su u tablici 3.2. Uvjet završetka pretrage svakog omotača bio je izvršavanje zadanog maksimalnog broja vrednovanja funkcije cilja (engl. *maximum number of function evaluations*, $\text{NFEs}_{\max}$), pri čemu je ona predstavljena

Tablica 3.2: Standardni omotači i njihove postavke za eksperimentalnu analizu

Algoritam	Parametri	f	$\text{NFEs}_{\max}$	Klasifikatori
GA [106]	$N_P = 50$ $p_c = 0.9$ $p_m = 0.1$	F1	10^4	1-NN, 5-NN, GNB, SVM
DE [106]	$N_P = 50$ $F = 0.5$ $CR = 0.9$	F1	10^4	1-NN, 5-NN, GNB, SVM
PSO [106]	$N_P = 30$ $w = 0.7298$ $c_1 = c_2 = 1.496$	F1	10^4	1-NN, 5-NN, GNB, SVM

Tablica 3.3: Unaprijeđeni omotači i njihove postavke za eksperimentalnu analizu

Algoritam	Opis specifičnih mehanizama	Parametri	f	$\text{NFEs}_{\max}$	Klasifikator
PSO_D [103]	Inicijalizacija populacije se provodi jednostavnom selekcijom značajki na temelju njihova ranga određenog Fisherovom mjerom.	$N_P = 30$ $w = 0.7298$ $c_1 = c_2 = 1.496$	F1	10^3	1-NN
PSO(4-2) [118]	Inicijalizacija populacije se provodi tako da dvije trećina rješenja sadrži samo 10% nasumično odabranih značajki, a ostala rješenja 50% nasumično odabranih značajki. Selekcijom se odabire manje od dva rješenja iste kvalitete.	$N_P = 30$ $w = 0.7298$ $c_1 = c_2 = 1.496$	MCR	$3 \cdot 10^3$	5-NN
EGAFS [134]	Poseban operator za križanje upravlja zamjenom segmenata rješenja između nekoliko točaka prekida na temelju njihove veličine.	$N_P = 200$ $p_c = 0.9$ $p_m = 0.05$	$0.9 \cdot \text{CAC} +$ $0.1 \cdot \frac{1}{\|\mathbf{b}\|_1}$	$2 \cdot 10^3$	GNB

mjerom F1. Da bi se stekao uvid u izvedbu predloženog proširenja na razini klasifikatora, svaki od tri standardna omotača udružen je sa svakim od četiri klasifikatora prikazanih u tablici 3.2. Uobičajeno se unutar omotača preferiraju jednostavniji klasifikatori, kod kojih izgradnja klasifikacijskog modela ne traje dugo [153, 154]. Stoga su odabrani klasifikatori 1-NN, 5-NN, SVM te naivan Bayesov klasifikator (engl. *Gaussian naive Bayes*, GNB), koji ujedno predstavljaju i jedne od najčešćih izbora za klasifikatore unutar omotača u literaturi [75]. Prilikom treniranja klasifikacijskog modela klasifikatora SVM na višeklasnim problemima, korištena je shema dekompozicije OVO. Pri tome je SVM rabljen s radijalnom funkcijom za jezgru i regularizacijskim parametrom $C = 1$.

U trećem dijelu analize, učinkovitost predloženog proširenja ispitana je za unaprijeđene omotače koji su zasnovani na bio-inspiriranim algoritmima u čiji su način rada ugrađeni mehanizmi specifični za problem odabira značajki. Postavke parametara ovih omotača preuzete su iz radova u kojima su predloženi te su prikazane u tablici 3.3, uz kratak opis rada ovih omotača. Treba napomenuti da su kod algoritma EGAFS izmijenjene težine u funkciji cilja, s obzirom na to da su korištene vrijednosti u izvornom radu ($\alpha = 0.8$ i $\beta = 0.2$) predložene za skupove podataka s iznimno velikim brojem značajki (iznad 10000), koji se pak ne rabe u ovoj eksperimentalnoj analizi.

Tijekom objedinjavanja arhive rješenja korišteni su ista funkcija cilja te klasifikator koji su primjenjeni unutar pojedinog omotača. S obzirom na to da predloženo proširenje izvodi dodatna vrednovanja (u najgorem slučaju d), arhiviranje rješenja tijekom pretrage zaustavljeno je d vrednovanja prije izvršenja ukupnog broja vrednovanja koji je dan omotaču, odnosno nakon $\text{NFEs}_{\max} - d$ vrednovanja. Na ovaj je način svakom omotaču omogućen doda-

tan broj vrednovanja kako bi se pravednije usporedile njegove performanse s performansama proširenog omotača.

3.4.2 Metodologija eksperimentalne analize

Tijek eksperimentalne analize započinje podjelom skupova podataka za potrebe treniranja i testiranja klasifikacijskih modela te se nastavlja izvršavanjem pretrage podešenih omotača i prikupljanjem ostvarenih rezultata. Podjela skupova podataka izvedena je na dva načina, ovisno o korištenom omotaču. Prethodno izvođenju pretrage svakog omotača [osim PSO(4-2)], iz skupa podataka redom su izdvojeni skupovi za treniranje, vrednovanje i testiranje, u omjeru $0.50 : 0.25 : 0.25$. Prva dva skupa daju se na raspolaganju omotaču za izvođenje pretrage, dok se treći koristi za stjecanje uvida u generalizaciju klasifikatora koju on ostvaruje na temelju rješenja pronađenih omotačem te njegovim proširenjem. Pri tome, vrednovanja koje provodi predloženo proširenje izvode se nad objedinjenim skupom za treniranje i vrednovanje, odnosno nad onim podacima koje omotač ima na raspolaganju tijekom pretrage. Kod omotača PSO(4-2), skup podataka podijeljen je na skup za treniranje i skup za testiranje u omjeru $0.7 : 0.3$, kako je i napravljeno u [118]. Skup za treniranje potom se dijeli postupkom unakrsne provjere korištenjem 10 preklopa za vrednovanje rješenja tijekom pretrage. Bitno je napomenuti da su sve izvršene podjele stratificirane da bi se očuvao omjer broja primjeraka različitih klasa u svakom od izvedenih skupova. Kako bi se stekao općenitiji uvid u performanse predloženog proširenja, napravljeno je 10 različitih podjela svakog skupa podataka, a pretraga omotača ponovljena je tri puta za svaku podjelu da bi se ublažio utjecaj stohastičke prirode bio-inspiriranih algoritama na dobivene rezultate. Time svaki omotač pronalazi 30 podskupova značajki za svaki skup podataka, a oni se vrednuju na izdvojenim skupovima za testiranje u svrhu određivanja njihova doprinosa sposobnosti generalizacije klasifikatora.

Kako bi se prikazale performanse korištenih omotača, za svaki skup podataka izračunate su prosječne kvalitete nađenih podskupova značajki ostvarene na skupu za testiranje te prosječno smanjenje dimenzionalnosti

$$\text{red} = \frac{d - \|\mathbf{b}\|_1}{d} \ . \tag{3.2}$$

S ciljem pojednostavljenja usporedbe performansi omotača i njegova proširenja, izvedene su Euklidske udaljenosti (označene s d_{perf}) između izvedbe savršenog klasifikatora i točke čije koordinate čine prosječne kvalitete podskupova značajki nađenih omotačem na svakom skupu podataka, kao u [91, 155]. Osim toga, usporedbom prosječnih kvaliteta njihovih rješenja na korištenim skupovima podataka pomoću Wilcoxonova testa ranga s predznakom [156], izračunati su i rangovi (označeni s R) omotača i njegova proširenja koji ukazuju na općenitu razliku u njihovu doprinosu izvedbi klasifikatora. Nadalje, usporedbom prosječnih veličina rješenja omotača i njegova proširenja primjenom istog testa, izvedeni su i rangovi (označeni s R_{red}) koji ukazuju na opću razliku u njihovom doprinosu smanjenju broja značajki. Konačno,

na temelju 30 podskupova značajki nađenih za svaki skup podataka izračunata je vrijednost mjere ASM (engl. *adjusted stability measure*) [146] koja iskazuje stabilnost omotača uslijed višestrukog izvođenja pretrage i preslagivanja skupa podataka. Postupak računanja ove mjere opisan je u dodatku B, kao i način interpretiranja njezinih vrijednosti koje se nalaze u $[-1, 1]$, pri čemu veće vrijednosti predstavljaju veću stabilnost omotača.

3.4.3 Analiza postupka objedinjavanja u predloženom proširenju

Kao što je pojašnjeno ranije, predloženo proširenje sastoji se od postupka prikupljanja kvalitetnih i raznolikih rješenja te postupka njihova objedinjavanja. Pri tome, objedinjavanje se provodi s pretpostavkom da je moguće stvoriti rješenje koje omogućuje klasifikatoru veću sposobnost generalizacije od pojedinih rješenja unutar arhive. Uz to, izvlačenjem njihovih zajedničkih karakteristika nastoji se povećati stabilnost bio-inspiriranog omotača s očekivanjem da će tako formirana rješenja biti konzistentnije strukture uslijed višestrukog izvođenja pretrage. Kako bi se utvrdila djelotvornost postupka objedinjavanja u ostvarivanju ovih ciljeva, predloženo proširenje uspoređeno je s istim proširenjem bio-inspiriranih omotača koje ne provodi objedinjavanje rješenja nego jednostavno vrednuje sva rješenja u arhivi postupkom unakrsne provjere korištenjem pet preklopa te vraća ono s najvećom kvalitetom, na sličan način kao što je predloženo u [97].

Obje inačice proširenja implementirane su za standardne omotače zasnovane na algoritmima GA, DE i PSO, a usporedbe njihovih performansi prikazane su u tablicama 3.4, 3.5 i 3.6. Uz naziv svakog omotača naznačeno je i proširenje s kojim je nadograđen, pri čemu je predloženo proširenje označeno s nastavkom +A, dok je pojednostavljeno proširenje označeno nastavkom $+\mathrm{A}_{\mathrm{R1}}$. Prema ranije opisanoj metodologiji eksperimentalne analize, izvedene su mjere koje sažeto iskazuju performanse ovih proširenih omotača. U tablicama su također prikazane i prosječne vrijednosti mjere ASM za svaki omotač.

Prikazani rezultati jasno potvrđuju korisnost provođenja objedinjavanja arhive rješenja. Na temelju rangova u smislu kvalitete koje postižu uspoređene inačice proširenja, može se zaključiti da predloženi postupak objedinjavanja općenito stvara rješenje koje omogućuje klasifikatoru veću sposobnost generalizacije od ostalih rješenja unutar arhive. Valjanost ovog zaključka vrijedi za sve razmatrane kombinacije omotača i klasifikatora (osim za omotač DE i klasifikator SVM, gdje nema razlike u rangovima), a podupiru ga i razlike u udaljenostima od savršenog klasifikatora. Uz to, rješenja formirana predloženim postupkom objedinjavanja konzistentnije su strukture uslijed višestrukog izvođenja pretrage, na što upućuju razlike u prosječnoj stabilnosti proširenih omotača. Osim toga, ona su ujedno i manja od najkvalitetnijeg rješenja unutar arhive, što sugeriraju razlike u rangovima u smislu veličine (R_{red}). S obzirom na to da je R_{red} pojednostavljenog proširenja jednak nuli za sve kombinacije omotača i klasifikatora, moguće je tvrditi da ono niti na jednom problemu ne vraća rješenja koja u prosjeku imaju manju veličinu od rješenja predloženog proširenja. Iako dodatno smanje-

Tablica 3.4: Usporedba performansi proširenih omotača GA+A_{R1} i GA+A

	1-NN		5-NN		GNB		SVM	
	GA+A_{R1}	GA+A	GA+A_{R1}	GA+A	GA+A_{R1}	GA+A	GA+A_{R1}	GA+A
R	31.50	**46.50**	27.50	**50.50**	12.50	**53.50**	37.50	**40.50**
d_{perf}	0.78	**0.71**	0.76	**0.66**	0.86	**0.80**	0.75	**0.64**
$\overline{\text{ASM}}$	0.11	**0.20**	0.11	**0.22**	0.17	**0.24**	0.12	**0.24**
R_{red}	0.00	**78.00**	0.00	**78.0**	0.00	**78.00**	0.00	**78.00**

Tablica 3.5: Usporedba performansi proširenih omotača DE+A_{R1} i DE+A

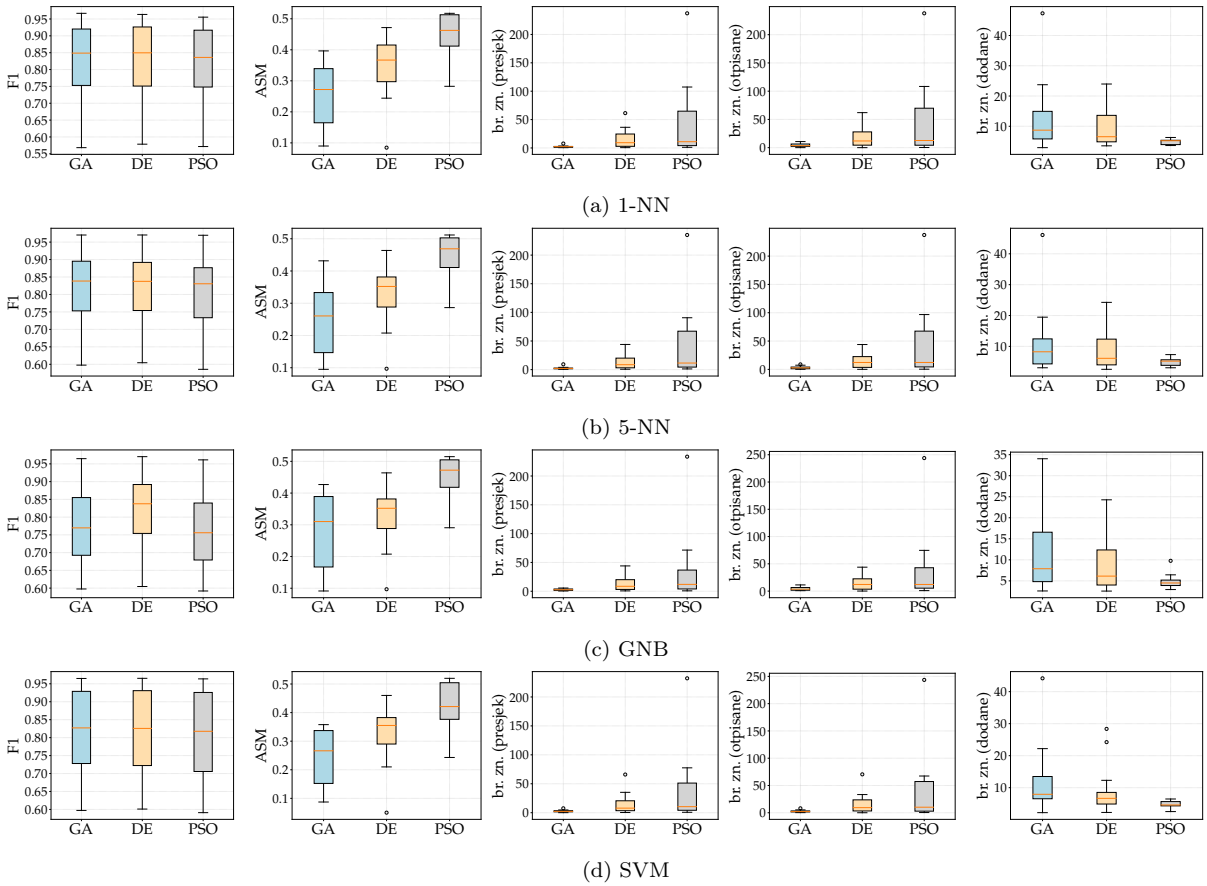
	1-NN		5-NN		GNB		SVM	
	DE+A_{R1}	DE+A	DE+A_{R1}	DE+A	DE+A_{R1}	DE+A	DE+A_{R1}	DE+A
R	20.00	**46.00**	18.50	**47.50**	27.50	**50.50**	33.00	33.00
d_{perf}	0.78	**0.74**	0.76	**0.70**	0.86	**0.80**	0.75	**0.72**
$\overline{\text{ASM}}$	0.09	**0.14**	0.10	**0.16**	0.16	**0.21**	0.12	**0.18**
R_{red}	0.00	**78.00**	0.00	**78.00**	0.00	**78.00**	0.00	**78.00**

Tablica 3.6: Usporedba performansi proširenih omotača PSO+A_{R1} i PSO+A

	1-NN		5-NN		GNB		SVM	
	PSO+A_{R1}	PSO+A	PSO+A_{R1}	PSO+A	PSO+A_{R1}	PSO+A	PSO+A_{R1}	PSO+A
R	28.00	**38.00**	38.50	**39.50**	18.00	**60.00**	26.50	**39.50**
d_{perf}	0.81	0.81	0.80	**0.79**	0.88	**0.84**	0.80	**0.75**
$\overline{\text{ASM}}$	0.05	**0.08**	0.06	**0.09**	0.11	**0.14**	0.08	**0.12**
R_{red}	0.00	**78.00**	0.00	**78.00**	0.00	**78.00**	0.00	**78.00**

nje dimenzionalnosti problema klasifikacije nije prioritet predloženog proširenja, svakako je koristan rezultat postupka objedinjavanja.

Bolje performanse predloženog proširenja u odnosu na pojednostavljeno posljedica su izvlačenja zajedničkih značajki rješenja arhive te postupnog uvođenja preostalih značajki u novo rješenje na temelju njihova doprinosa kvaliteti. Osim toga, rješenje formirano predloženim proširenjem uvijek će imati veću ili jednaku kvalitetu od presjeka rješenja arhive jer je naknadno uvođenje preostalih značajki u rješenje dobiveno presjekom uvjetovano povećanjem njegove kvalitete. Pri tome, broj naknadno uvedenih značajki varira ovisno o korištenom omotaču, na što upućuju dijagrami pravokutnika (engl. *box and whisker plots*) predstavljeni na slici 3.5. Ovi dijagrami za svaki klasifikator u omotačima GA, DE i PSO redom prikazuju usporedbe prosječnih kvaliteta (određenih na skupu za testiranje) arhive rješenja na svim skupovima podataka, sličnosti rješenja u arhivi (izraženih mjerom ASM), veličina presjeka, broja otpisanih (one značajke koje se ne nalaze niti u jednom rješenju arhive te se ne ispituju za dodavanje) te broja naknadno dodanih značajki. Moguće je zamijetiti da s porastom sličnosti rješenja u arhivi raste, očekivano, i veličina presjeka rješenja arhive, dok se broj naknadno dodanih značajki smanjuje. Stoga je većina značajki sadržanih u rješenju proširenog omotača PSO nastala izvođenjem presjeka rješenja arhive, dok se proširenje omotača GA više oslanja na postupno uvođenje preostalih značajki prilikom formiranja konačnog rješenja. Pri tome je redoslijed dodavanja tih značajki određen njihovom zastupljenosti u rješenjima arhive zbog pretpostavke da je zastupljenost značajke proporcionalna njezinu doprinosu kvaliteti. Ako je ta pretpostavka valjana, zastupljenije značajke bi se trebale ranije



(a) 1-NN

(b) 5-NN

(c) GNB

(d) SVM

Slika 3.5: Uvid u strukturu arhive rješenja za korištene omotače i klasifikatore

uvoditi u novo rješenje jer s povećanjem broja uključenih značajki interakcije među njima postaju složenije te se potencijalno umanjuje doprinos relevantnih značajki. Kako bi se utvrdila važnost redoslijeda dodavanja značajki, provedena je usporedba predloženog proširenja s istim proširenjem koji pak uvodi preostale značajke u novo rješenje redom, a ne na temelju njihove zastupljenosti. Tablice 3.7, 3.8 i 3.9 prikazuju usporedbe performansi ovih proširenja, pri čemu je proširenje koje preostale značajke uvodi redom označeno nastavkom $+\mathrm{A}_{\mathrm{R2}}$.

Ostvareni rezultati sugeriraju da redoslijed dodavanja značajki u novo rješenje ima utjecaj na konačne performanse proširenog omotača te se dodavanje prema njihovoj zastupljenosti može smatrati prikladnijim pristupom od dodavanja značajki redom. Predloženo proširenje najčešće formira kvalitetnija rješenja za sve kombinacije korištenih omotača i klasifikatora, što se može vidjeti iz rangova u smislu kvalitete, a potkrijepljeno je i njihovim udaljenostima od savršenog klasifikatora. Štoviše, razlike u tim rangovima još su izraženije u korist predloženog proširenja nego prilikom usporedbe s njegovom inačicom koja ne provodi objedinjavanje rješenja arhive. Također je zanimljivo primijetiti da redoslijed dodavanja značajki čini razliku u performansama uspoređenih inačica proširenja za omotač PSO, unatoč tome što se u rješenje dobiveno presjekom rješenja njegove arhive dodaje vrlo mali broj preostalih značajki, kao što sugeriraju dijagrami prikazani na slici 3.5. S druge strane, iako su razlike u rangovima u smislu veličine uspoređenih inačica proširenja uglavnom neznatne,

Tablica 3.7: Usporedba performansi proširenih omotača GA+A_{R2} i GA+A

	1-NN		5-NN		GNB		SVM	
	GA+A_{R2}	GA+A	GA+A_{R2}	GA+A	GA+A_{R2}	GA+A	GA+A_{R2}	GA+A
R	29.00	**49.00**	14.50	**63.50**	20.00	**58.00**	19.00	**47.00**
d_{perf}	0.77	**0.71**	0.71	**0.66**	0.81	**0.80**	0.69	**0.64**
$\overline{\text{ASM}}$	0.20	**0.20**	**0.27**	0.22	**0.28**	0.24	**0.25**	0.24
R_{red}	34.00	**44.00**	25.50	**52.50**	7.00	**59.00**	9.00	**69.00**

Tablica 3.8: Usporedba performansi proširenih omotača DE+A_{R2} i DE+A

	1-NN		5-NN		GNB		SVM	
	DE+A_{R2}	DE+A	DE+A_{R2}	DE+A	DE+A_{R2}	DE+A	DE+A_{R2}	DE+A
R	12.00	**57.00**	30.50	**47.50**	29.50	**48.50**	38.00	**40.00**
d_{perf}	0.76	**0.74**	0.70	0.70	0.80	0.80	**0.71**	0.72
$\overline{\text{ASM}}$	0.13	**0.14**	**0.18**	0.16	**0.23**	0.21	**0.20**	0.18
R_{red}	20.00	**46.00**	**35.00**	31.00	**39.00**	27.00	24.50	**41.50**

Tablica 3.9: Usporedba performansi proširenih omotača PSO+A_{R2} i PSO+A

	1-NN		5-NN		GNB		SVM	
	PSO+A_{R2}	PSO+A	PSO+A_{R2}	PSO+A	PSO+A_{R2}	PSO+A	PSO+A_{R2}	PSO+A
R	18.00	**48.00**	27.50	**50.50**	28.00	**50.00**	18.00	**60.00**
d_{perf}	0.83	**0.81**	0.80	**0.79**	0.85	**0.84**	0.76	**0.75**
$\overline{\text{ASM}}$	0.07	**0.08**	**0.10**	0.09	**0.15**	0.14	**0.13**	0.12
R_{red}	23.00	**55.00**	24.00	**31.00**	**42.00**	36.00	10.00	**68.00**

ipak upućuju na to da dodavanje značajki prema njihovoj zastupljenosti općenito rezultira manjim rješenjima za većinu korištenih omotača. Razlog tomu je što se vjerojatnost uključivanja manje relevantnih značajki smanjuje ako se prije njih uvedu relevantnije značajke, dok se dodavanjem po redu one mogu uvesti ranije. Konačno, uvođenje značajki u novo rješenje na temelju njihove zastupljenosti većinom rezultira manjom prosječnom stabilnosti nego kada se one uvode redom, što nije začuđujuće s obzirom na to da je potonji redoslijed dodavanja značajki fiksan te ne ovisi o sadržaju arhive rješenja. Ipak, razlike u prosječnim stabilnostima ovih inačica proširenja su neznatne, dok obje ostvaruju znatno veću stabilnost u odnosu na inačicu proširenja koja ne provodi objedinjavanje (što sugeriraju rezultati u tablicama 3.4, 3.5 i 3.6). Povećanje stabilnosti svakako je poželjan učinak proširenja, no njegovo ostvarivanje nauštrb narušavanja kvalitete rješenja ne može biti opravdano.

3.4.4 Utjecaj predloženog proširenja na standardne bio-inspirirane omotače

Standardnim bio-inspiriranim omotačima, kao što je ranije opisano, mogu se smatrati oni omotači koji su zasnovani na bio-inspiriranim algoritmima optimizacije korištenih u svom uobičajenom obliku. Ovi omotači su nadograđeni predloženim proširenjem kako bi se ispitao njegov utjecaj na poboljšanje sposobnosti generalizacije klasifikatora te na stabilnost omotača uslijed višestrukog izvođenja pretrage. Stoga je provedena usporedba performansi standardnih omotača i njihovih proširenih inačica, a ostvareni rezultati u smislu prosječnih

Tablica 3.10: Rezultati za omotač GA i njegovo proširenje

	1-NN				5-NN				GNB				SVM			
	F1		br. zn.		F1		br. zn.		F1		br. zn.		F1		br. zn.	
	prosjek±std. dev.		red (%)		prosjek±std. dev.		red (%)		prosjek±std. dev.		red (%)		prosjek±std. dev.		red (%)	
$\mathcal{D}$	GA	GA+A	GA	GA+A	GA	GA+A	GA	GA+A	GA	GA+A	GA	GA+A	GA	GA+A	GA	GA+A
$\mathcal{D}_1$	0.79±0.03	0.79±0.03	53	**59**	0.83±0.03	0.83±0.03	48	**57**	0.80±0.03	0.80±0.03	58	**66**	0.82±0.02	0.82±0.02	53	**63**
$\mathcal{D}_2$	0.94±0.02	0.94±0.02	51	**64**	0.95±0.02	**0.96**±0.02	46	**65**	0.95±0.03	0.95±0.02	62	**74**	0.96±0.02	0.96±0.02	50	**67**
$\mathcal{D}_3$	0.94±0.01	**0.95**±0.01	55	**70**	0.93±0.01	0.93±0.01	51	**71**	0.78±0.01	0.78±0.02	56	**77**	**0.91**±0.01	0.90±0.01	50	**72**
$\mathcal{D}_4$	0.69±0.09	**0.73**±0.07	55	**66**	0.68±0.11	**0.70**±0.08	53	**69**	0.74±0.06	**0.75**±0.06	45	**65**	0.73±0.08	**0.75**±0.07	58	**66**
$\mathcal{D}_5$	**0.62**±0.03	0.59±0.03	49	**70**	**0.65**±0.03	0.63±0.04	49	**76**	0.67±0.03	0.67±0.03	49	**72**	0.65±0.03	0.65±0.04	51	**74**
$\mathcal{D}_6$	0.86±0.04	**0.87**±0.03	62	**74**	0.83±0.06	**0.84**±0.05	72	**82**	0.87±0.05	**0.89**±0.05	62	**72**	**0.91**±0.03	0.90±0.03	62	**69**
$\mathcal{D}_7$	0.57±0.02	**0.85**±0.04	50	**97**	0.61±0.02	**0.89**±0.01	50	**97**	0.61±0.02	0.61±0.01	51	**95**	0.60±0.02	**0.79**±0.03	51	**95**
$\mathcal{D}_8$	**0.88**±0.05	0.87±0.06	61	**67**	0.84±0.04	0.84±0.06	54	**62**	0.82±0.04	**0.84**±0.03	61	**64**	0.87±0.06	0.87±0.04	53	**58**
$\mathcal{D}_9$	0.87±0.04	**0.89**±0.05	55	**63**	**0.85**±0.04	0.83±0.05	53	**70**	0.73±0.08	**0.74**±0.12	77	**83**	0.78±0.11	**0.79**±0.11	59	**80**
$\mathcal{D}_{10}$	**0.73**±0.09	0.69±0.11	51	**95**	0.77±0.07	**0.81**±0.09	51	**95**	0.56±0.05	**0.70**±0.11	53	**95**	0.74±0.07	**0.81**±0.08	54	**96**
$\mathcal{D}_{11}$	0.76±0.04	**0.79**±0.04	53	**83**	0.79±0.03	**0.81**±0.04	53	**86**	0.84±0.02	0.84±0.03	56	**85**	0.80±0.04	**0.82**±0.03	56	**85**
$\mathcal{D}_{12}$	0.94±0.03	**0.95**±0.03	41	**51**	0.95±0.03	0.95±0.04	42	**51**	0.97±0.02	**0.98**±0.02	45	**47**	0.96±0.02	**0.97**±0.02	39	**44**
R	23.00	**55.00**			23.00	**55.00**			10.00	**68.00**			15.00	**63.00**		
d_{perf}	0.81	**0.71**			0.77	**0.66**			0.88	**0.80**			0.76	**0.64**		

Tablica 3.11: Rezultati za omotač DE i njegovo proširenje

	1-NN				5-NN				GNB				SVM			
	F1		br. zn.		F1		br. zn.		F1		br. zn.		F1		br. zn.	
	prosjek±std. dev.		red (%)		prosjek±std. dev.		red (%)		prosjek±std. dev.		red (%)		prosjek±std. dev.		red (%)	
$\mathcal{D}$	DE	DE+A	DE	DE+A	DE	DE+A	DE	DE+A	DE	DE+A	DE	DE+A	DE	DE+A	DE	DE+A
$\mathcal{D}_1$	0.79±0.03	0.79±0.03	50	**54**	0.82±0.03	**0.83**±0.03	47	**52**	0.79±0.03	0.79±0.04	57	**62**	0.82±0.02	0.82±0.02	53	**59**
$\mathcal{D}_2$	0.94±0.02	0.94±0.02	50	**61**	0.96±0.02	0.96±0.02	49	**68**	0.95±0.03	0.95±0.02	60	**71**	0.96±0.02	0.96±0.02	51	**67**
$\mathcal{D}_3$	0.94±0.01	**0.95**±0.01	56	**65**	0.93±0.01	0.93±0.01	50	**66**	0.79±0.01	0.79±0.02	55	**66**	**0.92**±0.01	0.91±0.01	50	**64**
$\mathcal{D}_4$	0.70±0.07	**0.74**±0.06	55	**66**	0.68±0.10	**0.72**±0.09	50	**69**	0.75±0.06	**0.76**±0.06	43	**62**	0.74±0.09	0.74±0.08	57	**64**
$\mathcal{D}_5$	0.63±0.03	0.63±0.04	50	**52**	0.64±0.03	0.64±0.03	50	**58**	0.67±0.03	0.67±0.02	51	**59**	0.65±0.04	0.65±0.03	50	**57**
$\mathcal{D}_6$	0.87±0.04	0.87±0.04	62	**74**	0.83±0.06	**0.84**±0.05	69	**79**	0.88±0.04	0.88±0.05	62	**73**	**0.92**±0.02	0.91±0.02	61	**70**
$\mathcal{D}_7$	0.57±0.02	**0.65**±0.05	50	**83**	0.61±0.02	**0.72**±0.04	52	**87**	0.60±0.02	**0.61**±0.02	51	**81**	0.60±0.02	**0.64**±0.04	51	**81**
$\mathcal{D}_8$	0.87±0.05	0.87±0.05	60	**66**	**0.84**±0.04	0.83±0.05	52	**63**	0.82±0.03	**0.84**±0.03	60	**63**	**0.88**±0.04	0.87±0.04	53	**57**
$\mathcal{D}_9$	0.88±0.04	0.88±0.05	54	**66**	0.84±0.05	0.84±0.06	51	**68**	0.73±0.07	0.73±0.10	73	**81**	**0.78**±0.09	0.77±0.13	59	**79**
$\mathcal{D}_{10}$	**0.76**±0.09	0.74±0.11	51	**89**	0.76±0.06	**0.79**±0.07	51	**92**	0.56±0.05	**0.71**±0.10	52	**92**	0.73±0.09	**0.81**±0.07	53	**93**
$\mathcal{D}_{11}$	0.77±0.04	0.77±0.04	53	**69**	0.79±0.04	**0.80**±0.03	52	**74**	0.83±0.03	**0.85**±0.02	57	**78**	0.80±0.03	**0.81**±0.03	56	**76**
$\mathcal{D}_{12}$	0.95±0.03	0.95±0.04	41	**44**	0.94±0.03	**0.95**±0.03	44	**51**	0.97±0.02	0.97±0.03	45	**49**	0.96±0.03	**0.97**±0.02	44	44
R	13.00	**65.00**			17.00	**61.00**			7.00	**71.00**			34.00	**43.00**		
d_{perf}	0.79	**0.74**			0.78	**0.70**			0.87	**0.80**			0.76	**0.72**		

kvaliteta rješenja (izraženih mjerom F1 na skupu za testiranje) i njihovih standardnih devijacija prikazani su u tablicama 3.10, 3.11 i 3.12, pri čemu je predloženo proširenje označeno nastavkom +A. Uz prosječnu kvalitetu rješenja, prikazano je i prosječno ostvareno smanjenje broja značajki [koje je dano prema (3.2), a vrijednosti su izražene u obliku postotka]. Na dnu tablica prikazani su rangovi u smislu kvalitete te udaljenosti od savršenog klasifikatora. Pri tome su bolje vrijednosti ovih rezultata podebljane za svaku kombinaciju omotača, skupa podataka i klasifikatora.

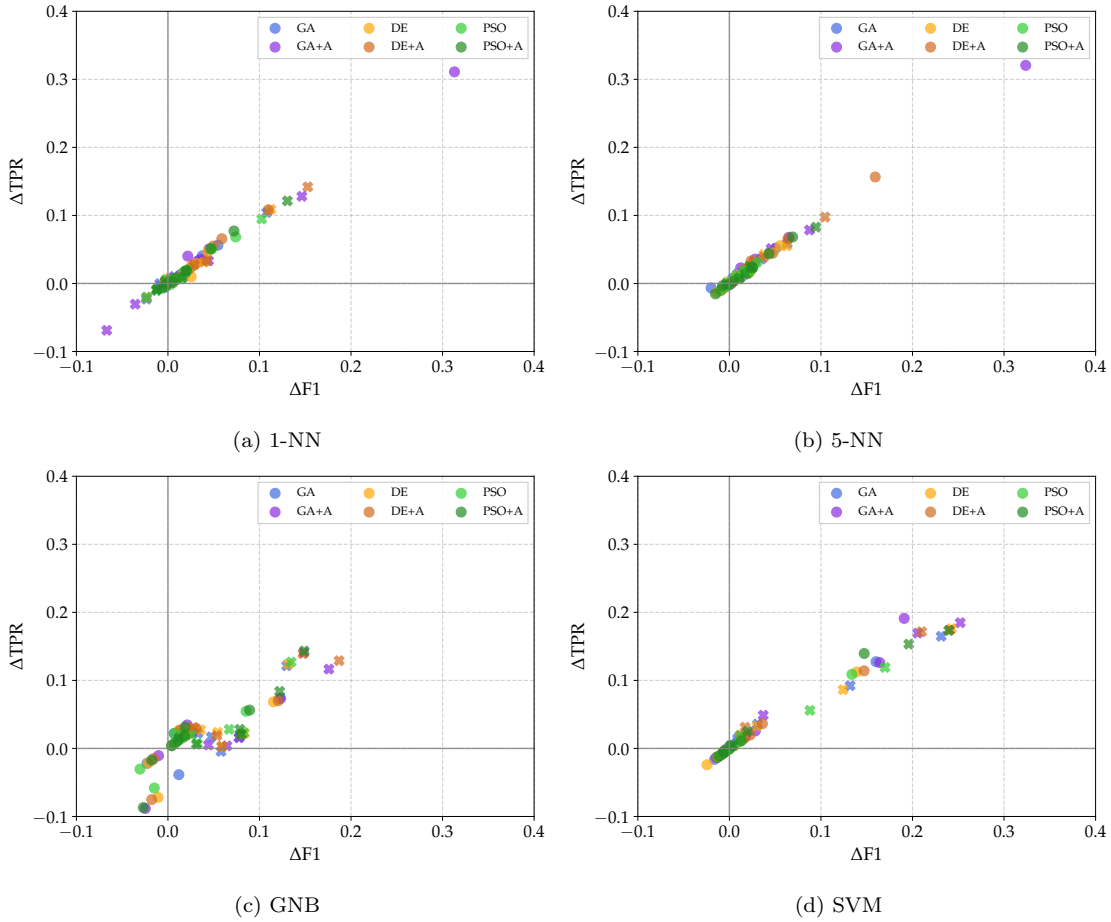
Na temelju prikazanih rangova i udaljenosti od savršenog klasifikatora, može se zaključiti da prošireni omotači, cjelokupno gledano, ostvaruju bolju uspješnost klasifikacije u odnosu na standardne omotače, neovisno o kombinaciji bio-inspiriranog algoritma i klasifikatora. Predloženo proširenje tako se može smatrati korisnom nadogradnjom omotača pomoću koje je moguće ostvariti bolju sposobnost generalizacije klasifikatora na velikom broju promatranih problema klasifikacije. Povrh toga, prošireni omotači uvijek formiraju rješenja manje veličine od neproširenih (osim na skupu podataka $\mathcal{D}_{12}$ za nekolicinu kombinacija omotača i klasifikatora, iako su razlike beznačajne), što je prvenstveno posljedica načina objedinja-

Tablica 3.12: Rezultati za omotač PSO i njegovo proširenje

	1-NN				5-NN				GNB				SVM			
	F1		br. zn.		F1		br. zn.		F1		br. zn.		F1		br. zn.	
	prosjek±std. dev.		red (%)		prosjek±std. dev.		red (%)		prosjek±std. dev.		red (%)		prosjek±std. dev.		red (%)	
$\mathcal{D}$	PSO	PSO+A	PSO	PSO+A	PSO	PSO+A	PSO	PSO+A	PSO	PSO+A	PSO	PSO+A	PSO	PSO+A	PSO	PSO+A
$\mathcal{D}_1$	0.79±0.03	0.79±0.03	50	**51**	0.82±0.03	0.82±0.02	46	**49**	0.76±0.02	0.76±0.02	49	**52**	0.82±0.02	0.82±0.02	49	**55**
$\mathcal{D}_2$	0.94±0.02	0.94±0.01	50	**59**	0.95±0.02	0.95±0.03	46	**58**	0.94±0.03	**0.95**±0.03	55	**66**	0.96±0.02	0.96±0.02	50	**62**
$\mathcal{D}_3$	0.92±0.01	0.92±0.01	52	**54**	0.92±0.01	0.92±0.01	50	**51**	0.77±0.01	0.77±0.01	51	**53**	0.90±0.01	**0.91**±0.01	46	**50**
$\mathcal{D}_4$	0.69±0.07	**0.71**±0.08	52	**65**	0.63±0.08	**0.71**±0.07	48	**66**	0.74±0.06	0.74±0.07	45	**60**	0.67±0.12	**0.74**±0.08	52	**65**
$\mathcal{D}_5$	0.62±0.03	0.62±0.03	49	**51**	0.63±0.03	**0.64**±0.03	48	**52**	0.67±0.03	0.67±0.03	49	**53**	0.64±0.04	0.64±0.04	48	**54**
$\mathcal{D}_6$	0.87±0.04	0.87±0.04	54	**68**	0.81±0.06	**0.82**±0.04	62	**72**	0.87±0.04	**0.89**±0.04	55	**65**	**0.92**±0.03	0.91±0.03	54	**63**
$\mathcal{D}_7$	0.56±0.02	0.56±0.03	50	**51**	0.59±0.02	0.59±0.02	50	**52**	0.60±0.02	0.60±0.02	51	**52**	0.59±0.02	0.59±0.02	51	**53**
$\mathcal{D}_8$	0.86±0.05	0.86±0.05	53	**63**	**0.85**±0.04	0.84±0.05	49	**59**	0.82±0.04	**0.84**±0.03	58	**61**	0.87±0.04	**0.88**±0.04	46	**54**
$\mathcal{D}_9$	0.88±0.04	0.88±0.05	51	**62**	0.82±0.07	**0.83**±0.06	52	**66**	0.74±0.07	**0.75**±0.05	67	**78**	0.76±0.12	**0.77**±0.10	53	**77**
$\mathcal{D}_{10}$	0.73±0.08	**0.75**±0.09	50	**64**	0.76±0.07	0.76±0.09	51	**69**	0.55±0.07	**0.65**±0.12	51	**87**	0.69±0.08	**0.80**±0.08	52	**82**
$\mathcal{D}_{11}$	0.76±0.04	0.76±0.04	51	**54**	0.78±0.03	0.78±0.03	50	**53**	0.83±0.03	0.83±0.03	53	**58**	0.80±0.04	0.80±0.04	53	**57**
$\mathcal{D}_{12}$	0.94±0.04	0.94±0.04	38	**49**	0.95±0.04	0.95±0.03	44	**47**	0.96±0.03	**0.97**±0.03	46	46	0.97±0.03	0.97±0.03	**46**	45
R	19.00	**59.00**			19.00	**59.00**			13.00	**65.00**			16.00	**62.00**		
d_{perf}	0.82	**0.81**			0.83	**0.79**			0.89	**0.84**			0.81	**0.75**		

vanja arhive rješenja, kao što sugeriraju ranije pokazani rezultati. Izvlačenje zajedničkih značajki rješenja u arhivi može rezultirati vrlo malim podskupovima značajki, dok se naknadnim dodavanjem značajki prema njihovu doprinosu kvaliteti u konačno rješenje ubacuje mali broj preostalih značajki, na što upućuju veličine presjeka te brojevi naknadno dodanih značajki prikazani na slici 3.5. S obzirom na to da je smanjenje dimenzionalnosti problema jedan od ciljeva pristupa za odabir značajki, ovakvo ponašanje dodatno povećava korisnost predloženog proširenja. Shodno tome, predloženo proširenje može se smatrati korisnom nadogradnjom omotača i u slučaju kada formira rješenje jednake kvalitete kao ono od omotača, ali manje veličine.

Kao što je ranije navedeno, postupak odabira značajki jedan je od najzastupljenijih pristupa za ublažavanje problema neuravnoteženosti klasa u literaturi jer u pravilu smanjuje složenost koncepta manjinske klase i time pospješuje njezino prepoznavanje. Kako bi se stekao uvid u to, za sve skupove podataka izvedene su razlike u izvedbama korištenih klasifikatora prije i nakon provođenja odabira značajki u smislu ostvarenih vrijednosti mjera F1 i TPR (ΔF1 i ΔTPR) te su njihove ovisnosti prikazane na slici 3.6. Iz slike je jasna proporcionalnost između razlika u vrijednostima mjera F1 i TPR, što upućuje na to da povećanje opće uspješnosti klasifikatora prvenstveno proizlazi iz povećanja uspješnosti prepoznavanja manjinske klase. Pozitivan učinak odabira značajki posebno je vidljiv na neuravnoteženim skupovima podataka ($\mathcal{D}_3$, $\mathcal{D}_4$, $\mathcal{D}_5$, $\mathcal{D}_9$, $\mathcal{D}_{10}$ i $\mathcal{D}_{11}$), koji su na slici označeni pomoću simbola $\times$. S obzirom na to da se prikazane razlike za većinu ovih skupova podataka nalaze u prvom kvadrantu grafova, moguće je zaključiti da postupak odabira značajki za njih uzrokuje povećavanje opće uspješnosti klasifikacije te uspješnosti prepoznavanja manjinske klase. S druge strane, na malom broju skupova podataka provođenje ovog postupka ima obrnuti učinak od prethodno navedenog, što sugeriraju točke na grafovima smještene u trećem kvadrantu. Ipak, negativan učinak odabira značajki na tim skupovima podataka varira ovisno o korištenom omotaču i klasifikatoru te je manje izražen u odnosu na rast vrijednosti mjera F1 i TPR koji ovaj postupak uzrokuje na većini ostalih skupova podataka.



(a) 1-NN

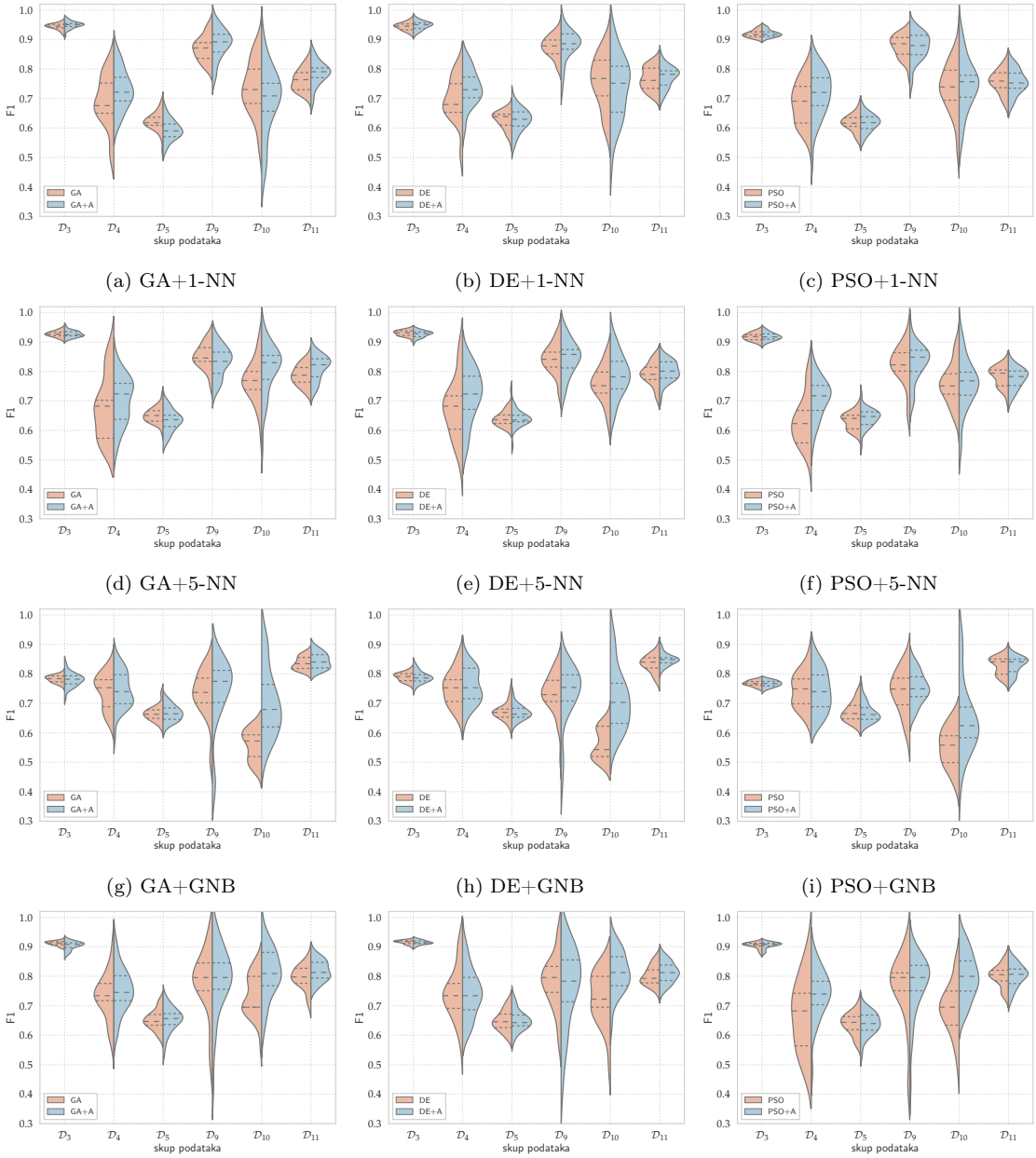
(b) 5-NN

(c) GNB

(d) SVM

Slika 3.6: Razlike u vrijednostima mjera F1 i TPR ostvarenim na smanjenim i punim skupovima značajki

Iako predloženo proširenje nije dizajnirano specifično za neuravnotežene probleme, prosječne vrijednosti mjere F1 prikazane u tablicama 3.10, 3.11 i 3.12 daju naslutiti da za velik broj takvih problema ono nadmašuje neproširenu omotač. Kako bi se dobio jasniji uvid u učinkovitost proširenja na neuravnoteženim skupovima podataka, na slici 3.7 su pomoću violinskih dijagrama prikazane distribucije kvaliteta rješenja koje ostvaruju omotači i njihova proširenja na takvim problemima. Unutar ovih dijagrama označene su vrijednosti prvog i trećeg kvartila te medijana. Iz prikazanih dijagrama može se zaključiti da kvalitete rješenja omotača i njegova proširenja proizlaze iz različitih distribucija, iako je na većini skupova podataka razlika u njihovim prosječnim ostvarenim kvalitetama neznatna (što se vidi iz tablica 3.10, 3.11 i 3.12). Položaji i vrhovi distribucija, kao i oznake kvartila i medijana, pokazuju da prošireni omotači češće pronalaze rješenja veće kvalitete, za većinu kombinacija skupa podataka, omotača i klasifikatora. S obzirom na to da se proširenjem omotača dodatno povećava vrijednost mjere F1 na većini neuravnoteženih problema, a ranije je pokazano da je ono popraćeno i rastom vrijednosti mjere TPR, može se zaključiti da nadogradnja standardnih bio-inspiriranih omotača predloženim proširenjem doprinosi poboljšanju učenja iz neuravnoteženih skupova podataka.



(a) GA+1-NN (b) DE+1-NN (c) PSO+1-NN

(d) GA+5-NN (e) DE+5-NN (f) PSO+5-NN

(g) GA+GNB (h) DE+GNB (i) PSO+GNB

(j) GA+SVM (k) DE+SVM (l) PSO+SVM

Slika 3.7: Distribucije kvaliteta rješenja standardnih omotača za neuravnotežene skupove podataka

Drugi cilj koji se nastoji ostvariti predloženim proširenjem bio-inspiriranih omotača jest povećanje njihove stabilnosti uslijed višestrukog izvođenja pretrage ili drugačije podjele skupa podataka. Prikazane vrijednosti standardnih devijacija u tablicama 3.10, 3.11 te 3.12 daju naslutiti kako je razlika u stabilnosti omotača i njegova proširenja neznatna te da su oni relativno stabilni pristupi za odabir značajki. Međutim, kako je i ranije objašnjeno, ova mjera nije prikladna za ispitivanje stabilnosti pristupa za odabir značajki jer dva rješenja različite strukture mogu imati jednaku kvalitetu. Stoga su u tablici 3.13 izvedene vrijednosti mjere ASM za svaki skup podataka te su vrijednosti koje ostvaruju omotač i njegovo proširenje

Tablica 3.13: Stabilnost standardnih omotača i njihovih proširenja

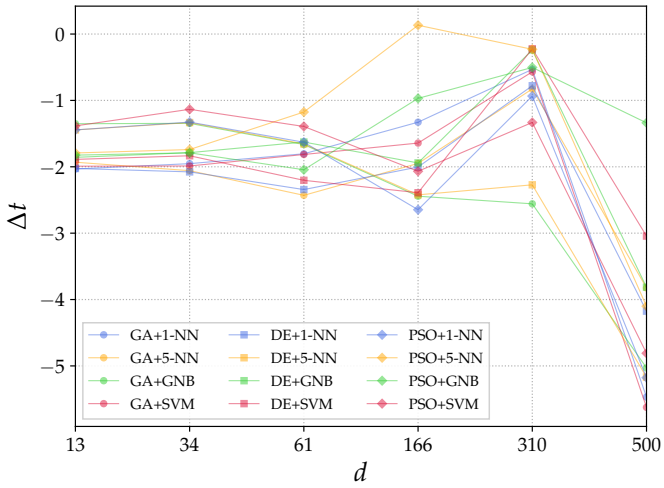
	1-NN			5-NN			GNB			SVM		
$\mathcal{D}$	GA/+A	DE/+A	PSO/+A	GA/+A	DE/+A	PSO/+A	GA/+A	DE/+A	PSO/+A	GA/+A	DE/+A	PSO/+A
$\mathcal{D}_1$	0.06/**0.09**	0.04/**0.05**	0.01/**0.03**	0.06/**0.09**	0.05/**0.07**	0.02/**0.04**	0.19/**0.23**	0.20/**0.23**	0.10/**0.11**	0.09/**0.15**	0.07/**0.12**	0.04/**0.07**
$\mathcal{D}_2$	0.05/**0.12**	0.04/**0.08**	0.02/0.02	0.03/**0.08**	0.03/**0.09**	0.02/**0.03**	0.17/**0.28**	0.19/**0.30**	0.13/**0.19**	0.04/**0.08**	0.05/**0.10**	0.02/**0.05**
$\mathcal{D}_3$	0.10/**0.17**	0.10/**0.15**	0.03/0.03	0.06/**0.13**	0.06/**0.09**	0.02/**0.03**	0.09/**0.17**	0.11/**0.14**	0.04/**0.05**	0.08/**0.12**	0.09/**0.13**	0.02/**0.03**
$\mathcal{D}_4$	0.19/**0.41**	0.20/**0.36**	0.15/**0.34**	0.25/**0.47**	0.25/**0.44**	0.16/**0.34**	0.09/**0.35**	0.09/**0.31**	0.10/**0.24**	0.29/**0.46**	0.28/**0.42**	0.20/**0.41**
$\mathcal{D}_5$	0.05/**0.08**	0.04/**0.05**	0.02/0.02	0.09/**0.15**	0.08/**0.09**	0.04/0.04	0.07/**0.15**	0.07/**0.11**	0.04/**0.06**	0.08/**0.20**	0.08/**0.09**	0.03/**0.05**
$\mathcal{D}_6$	0.07/**0.14**	0.07/**0.16**	0.03/**0.10**	0.13/**0.22**	0.10/**0.18**	0.06/**0.10**	0.13/**0.26**	0.13/**0.23**	0.08/**0.14**	0.11/**0.16**	0.08/**0.14**	0.04/**0.08**
$\mathcal{D}_7$	0.01/**0.65**	0.02/**0.17**	0.01/0.01	0.02/**0.65**	0.02/**0.23**	0.01/0.01	0.01/**0.07**	0.01/**0.03**	0.00/**0.01**	0.01/**0.50**	0.02/**0.07**	0.01/0.01
$\mathcal{D}_8$	0.14/**0.21**	0.13/**0.21**	0.09/**0.19**	0.17/**0.29**	0.16/**0.24**	0.13/**0.19**	0.25/**0.39**	0.24/**0.34**	0.20/**0.32**	0.17/**0.28**	0.16/**0.26**	0.10/**0.19**
$\mathcal{D}_9$	0.05/**0.15**	0.05/**0.15**	0.04/**0.09**	0.04/**0.10**	0.05/**0.10**	0.01/**0.05**	0.28/**0.35**	0.25/**0.27**	0.18/**0.20**	0.09/**0.30**	0.10/**0.29**	0.05/**0.21**
$\mathcal{D}_{10}$	0.01/**0.08**	0.01/**0.07**	0.00/**0.01**	0.00/**0.11**	0.00/**0.06**	0.00/**0.01**	0.01/**0.13**	0.01/**0.12**	0.00/**0.05**	0.02/**0.24**	0.02/**0.17**	0.00/**0.06**
$\mathcal{D}_{11}$	0.04/**0.16**	0.04/**0.05**	0.01/**0.02**	0.04/**0.18**	0.04/**0.09**	0.02/0.02	0.04/**0.15**	0.05/**0.12**	0.02/**0.03**	0.03/**0.17**	0.05/**0.12**	0.02/**0.03**
$\mathcal{D}_{12}$	0.06/**0.19**	0.08/**0.16**	0.06/**0.14**	0.11/**0.20**	0.10/**0.21**	0.11/**0.17**	0.14/**0.30**	0.15/**0.29**	0.12/**0.31**	0.09/**0.23**	0.05/**0.23**	0.05/**0.21**

odvojene znakom /. Prikazane vrijednosti jasno ukazuju na to da se proširenjem standardnih bio-inspiriranih omotača povećava njihova stabilnost, odnosno da prošireni omotači uslijed višestrukog izvođenja pretrage te drugačije podjele skupa podataka pronalaze rješenja konzistentnije strukture. Ovakav rezultat može se pripisati postupku objedinjavanja arhive rješenja u kojem se nastoje odrediti najrelevantnije značajke za promatrani problem klasifikacije. Rast stabilnosti uzrokovan proširenjem omotača općenito varira ovisno o bio-inspiriranom algoritmu te klasifikatoru unutar omotača. Korištenjem klasifikatora GNB nađena rješenja imaju konzistentniju strukturu, dok su omotači najmanje stabilni kada koriste klasifikatore 1-NN i 5-NN. Omotači GA i DE općenito su stabilniji od omotača PSO, unatoč tome što omotač PSO prikuplja arhivu najsličnijih rješenja, što se vidi iz dijagrama na slici 3.5. Ipak, njihovim proširenjem povećava se njihova stabilnost za svaku kombinaciju skupa podataka i klasifikatora, dok se proširenjem omotača PSO na malom broju skupova podataka ona zadržava na istoj razini kao i prije proširenja.

Bitno je podsjetiti kako je u eksperimentalnoj analizi svakom omotaču po završetku arhiviranja omogućeno d dodatnih vrednovanja funkcije cilja, s obzirom na to da se isti broj vrednovanja može potrošiti (u najgorem slučaju) u postupku objedinjavanja arhive rješenja. Unatoč tome, omotač uglavnom ne uspijeva pronaći kvalitetnije rješenje od onoga što je formirano predloženim postupkom objedinjavanja. Stoga se može zaključiti da su vrednovanja tijekom objedinjavanja utrošena na učinkovitiji način. Međutim, ova vrednovanja su i vremenski zahtjevnija, jer se zasnivaju na unakrsnoj provjeri pomoću pet preklopa. Kako bi se demonstrirao vremenski trošak izvođenja predloženog proširenja, slika 3.8 prikazuje relativnu razliku u trajanju omotača i njihovih proširenja

$$\Delta t = \frac{t_{Arhiviraj} + t_{Objedini} - t_{Omotač}}{t_{Omotač}}, \tag{3.3}$$

pri čemu su vrijednosti na slici prikazane u obliku postotka. Pozitivne vrijednosti razlike ukazuju na to da postupak objedinjavanja arhive rješenja traje duže od d vrednovanja u omotaču, dok negativne vrijednosti upućuju na suprotno. Iako postupak objedinjavanja



Slika 3.8: Razlika u trajanju standardnih omotača i njihovih proširenja ovisno o dimenzionalnosti skupa podataka

provodi vremenski zahtjevnija vrednovanja, ono u većini slučajeva traje kraće od dodatnih d vrednovanja unutar omotača, na što upućuju pretežno negativne vrijednosti Δt. Razlog tomu je što rješenja u arhivi ipak sadrže neke zajedničke dijelove pa nije potrebno ispitati uvođenje svih d značajki u konačno rješenje, na što ukazuju veličine presjeka i brojevi otpisanih značajki prikazani na slici 3.5. Razlike u trajanju omotača i njegova proširenja variraju ovisno o dimenzionalnosti problema klasifikacije, sličnosti rješenja u arhivi te o korištenom klasifikatoru unutar omotača. Proširenjem omotača GA potrebno je potrošiti najveći broj vrednovanja u usporedbi s proširenjima ostalih standardnih omotača, s obzirom na to da rješenja u njegovoj arhivi imaju malo zajedničkih dijelova, što pokazuje slika 3.5. Unatoč tome, provođenje tog proširenja ne traje duže od produljenja pretrage omotača GA, što sugeriraju njihove razlike u trajanju prikazane na slici 3.8. Cjelokupno gledano, moguće je zaključiti da se proširenjem bio-inspiriranih omotača poboljšavaju njihove performanse u većoj mjeri nego što je to moguće postići produženjem postupka pretrage te uz zanemariv utrošak vremena.

3.4.5 Utjecaj predloženog proširenja na unaprijeđene bio-inspirirane omotače

Učinkovitost predloženog proširenja također je ispitana i za unaprijeđene omotače koji su zasnovani na bio-inspiriranim algoritmima u koje su ugrađeni mehanizmi specifični za problem odabira značajki. Ostvareni rezultati za te omotače i njihova proširenja dani su u tablicama 3.14, 3.15 i 3.16. Kvalitete rješenja na skupu za testiranje izražene su pomoću mjera koje su korištene kao funkcije cilja tijekom pretrage svakog od razmatranih omotača.

S obzirom na izvedene rangove i udaljenosti od savršenog klasifikatora, moguće je zaključiti da se proširenjem unaprijeđenih omotača također formiraju podskupovi značajki za koje klasifikatori ostvaruju bolju sposobnost generalizacije. Uz to, oni su ujedno manje veličine i

Tablica 3.14: Rezultati za omotač PSO_D i predloženo proširenje

$\mathcal{D}$	F1		br. zn.		ASM	
	PSO_D	PSO_D+A	PSO_D	PSO_D+A	PSO_D	PSO_D+A
	prosjek±std. dev	prosjek±std. dev	red (%)	red (%)		
$\mathcal{D}_1$	0.79±0.03	0.79±0.02	43	**47**	0.04	**0.05**
$\mathcal{D}_2$	0.94±0.02	0.94±0.02	43	**56**	0.07	**0.09**
$\mathcal{D}_3$	0.91±0.00	0.91±0.00	84	**89**	0.03	**0.04**
$\mathcal{D}_4$	0.67±0.08	**0.74**±0.08	50	**66**	0.17	**0.36**
$\mathcal{D}_5$	**0.62**±0.03	0.61±0.03	43	**49**	**0.07**	0.04
$\mathcal{D}_6$	0.85±0.05	**0.87**±0.05	64	**74**	0.06	**0.11**
$\mathcal{D}_7$	0.55±0.02	0.55±0.03	40	**61**	**0.02**	0.01
$\mathcal{D}_8$	0.86±0.06	**0.87**±0.06	58	**65**	0.14	**0.24**
$\mathcal{D}_9$	**0.89**±0.04	0.86±0.07	47	**65**	0.10	**0.13**
$\mathcal{D}_{10}$	0.73±0.07	**0.74**±0.07	49	**68**	0.02	**0.03**
$\mathcal{D}_{11}$	0.74±0.04	**0.75**±0.04	49	**54**	**0.05**	0.04
$\mathcal{D}_{12}$	0.94±0.03	**0.95**±0.03	34	**46**	0.12	**0.16**
R	23	**55**				
d_{perf}	0.84	**0.82**				

Tablica 3.15: Rezultati za omotač PSO(4-2) i predloženo proširenje

$\mathcal{D}$	MCR		br. zn.		ASM	
	PSO(4-2)	PSO(4-2)+A	PSO(4-2)	PSO(4-2)+A	PSO(4-2)	PSO(4-2)+A
	prosjek±std. dev	prosjek±std. dev	red (%)	red (%)		
$\mathcal{D}_1$	0.15±0.02	0.15±0.02	31	**36**	0.05	0.05
$\mathcal{D}_2$	0.04±0.02	0.04±0.02	42	**52**	**0.07**	0.06
$\mathcal{D}_3$	0.04±0.01	0.04±0.01	28	**29**	**0.04**	0.03
$\mathcal{D}_4$	0.07±0.01	0.07±0.01	40	**62**	0.19	**0.30**
$\mathcal{D}_5$	0.28±0.02	0.28±0.02	37	**41**	0.03	0.03
$\mathcal{D}_6$	**0.11**±0.04	0.12±0.03	85	**86**	0.14	**0.17**
$\mathcal{D}_7$	0.40±0.03	**0.39**±0.03	43	**46**	0.02	0.02
$\mathcal{D}_8$	0.13±0.05	0.13±0.06	46	**55**	0.10	**0.13**
$\mathcal{D}_9$	0.11±0.03	**0.10**±0.03	48	**59**	0.07	**0.14**
$\mathcal{D}_{10}$	0.20±0.05	0.20±0.04	60	**66**	0.01	**0.02**
$\mathcal{D}_{11}$	0.20±0.02	0.20±0.03	45	**47**	0.02	**0.03**
$\mathcal{D}_{12}$	0.04±0.02	0.04±0.02	34	**50**	0.20	**0.21**
R	28	**38**				
d_{perf}	0.63	**0.62**				

konzistentnije strukture u odnosu na podskupove značajki nađene pretragom omotača. Ipak, mogu se primijetiti neznatna odstupanja u rezultatima, ovisno o načinu rada ovih omotača. Stabilnosti omotača PSO_D i PSO(4-2) na nekim su skupovima podataka veće ili jednake od stabilnosti njihova proširenja, s obzirom na to da oba koriste posebne načine inicijalizacije populacije koji usmjeravaju pretragu pa njezinim višestrukim izvođenjem pronalaze rješenja konzistentnije strukture. Nadalje, omotač PSO(4-2) tijekom pretrage potencijalno ima bolji uvid u sposobnost generalizacije klasifikatora s obzirom na činjenicu da vrednuje rješenja unakrsnom provjerom pomoću 10 preklopa. Ipak, prema ostvarenoj prosječnoj kvaliteti rješenja na skupu za testiranje, ovaj omotač nadmašuje svoje proširenje samo na jednom skupu podataka. Osim toga, iako u početku usmjerava pretragu oko rješenja male veličine te koristi poseban oblik selekcije koji favorizira takva rješenja, njegovo proširenje ga svejedno nadmašuje po stupnju smanjenja dimenzionalnosti. Isto vrijedi i za omotač EGAFS koji pomoću funkcije cilja kažnjava rješenja ovisno o broju uključenih značajki, no unatoč tome

Tablica 3.16: Rezultati za omotač EGAFS i predloženo proširenje

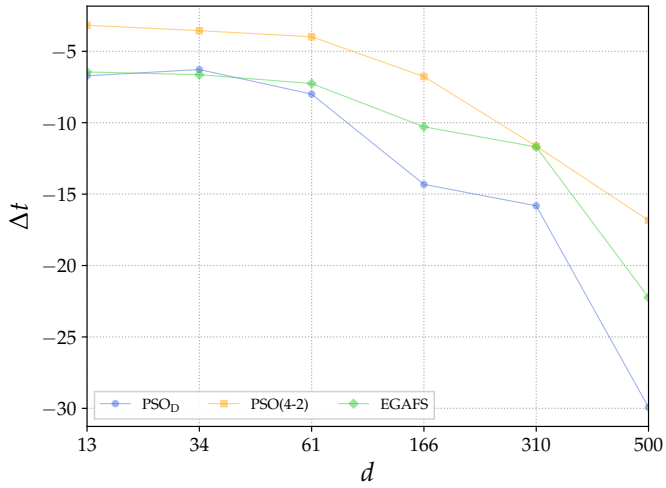
$\mathcal{D}$	CAC		br. zn.		ASM	
	EGAFS	EGAFS+A	EGAFS	EGAFS+A	EGAFS	EGAFS+A
	prosjek±std. dev	prosjek±std. dev	red (%)	red (%)		
$\mathcal{D}_1$	0.78±0.02	**0.79**±0.03	57	**64**	0.11	**0.21**
$\mathcal{D}_2$	0.95±0.02	0.95±0.02	63	**77**	0.14	**0.32**
$\mathcal{D}_3$	0.86±0.01	**0.88**±0.03	51	**94**	0.04	**0.13**
$\mathcal{D}_4$	0.93±0.02	**0.94**±0.02	**74**	62	0.14	**0.36**
$\mathcal{D}_5$	0.72±0.02	0.72±0.02	53	**85**	0.05	**0.20**
$\mathcal{D}_6$	0.88±0.03	**0.90**±0.04	62	**78**	0.10	**0.27**
$\mathcal{D}_7$	0.61±0.02	0.61±0.02	50	**97**	0.01	**0.08**
$\mathcal{D}_8$	0.83±0.04	0.83±0.04	65	65	0.23	**0.39**
$\mathcal{D}_9$	0.80±0.04	**0.84**±0.04	77	**87**	0.17	**0.34**
$\mathcal{D}_{10}$	0.57±0.06	**0.70**±0.10	62	**95**	0.00	**0.12**
$\mathcal{D}_{11}$	0.83±0.02	**0.84**±0.03	55	**87**	0.02	**0.11**
$\mathcal{D}_{12}$	0.95±0.03	**0.97**±0.02	**64**	52	0.19	**0.32**
R	5	**73**				
d_{perf}	0.78	**0.69**				

uglavnom pronalazi veća, a manje kvalitetna rješenja od svog proširenja.

Unaprijeđenim omotačima je, kao i standardnim omotačima, također omogućeno d naknadnih vrednovanja nakon završetka arhiviranja rješenja. Relativne razlike u trajanju unaprijeđenih omotača i njihovih proširenja (Δt) prikazane su na slici 3.9. S obzirom na to da ovi omotači koriste manji broj vrednovanja tijekom pretrage u odnosu na standardne omotače, razlika u trajanju još je izraženija u korist njihova proširenja. Primjerice, na skupu podataka koji sadrži 500 značajki ($\mathcal{D}_5$) omotač PSO_D prekida arhiviranje nakon 500 vrednovanja te nastavlja pretragu s još 500 vrednovanja. Međutim, u postupku objedinjavanja očito se potroši znatno manji broj vrednovanja s obzirom na izraženiju razliku u trajanju izvođenja tog omotača i njegova proširenja. Povrh toga, vrednovanja u postupku objedinjavanja su i vremenski manje zahtjevna od vrednovanja koje izvodi omotač PSO(4-2), pa se njegovim proširenjem umjesto produljenja pretrage mogu povećati performanse te smanjiti vremenski trošak izvođenja tog omotača.

3.5 Osvrt na odabir značajki i predloženo proširenje bio-inspiriranih omotača

Odabir značajki važan je postupak predobrade skupova podataka kojim je moguće smanjiti udaljenost primjeraka u ulaznom prostoru te ublažiti složenost problema klasifikacije. Smanjenje udaljenosti između primjeraka te razdvajanje područja preklapanja klasa posebice olakšava učenje koncepta manjinske klase, pa se ovaj postupak pokazuje korisnim za poboljšanje učenja iz neuravnoteženih skupova podataka. Kao što pokazuju rezultati eksperimentalne analize, upotreba bio-inspiriranih omotača za odabir značajki uglavnom rezultira poboljšanjem uspješnosti prepoznavanja manjinske klase te opće uspješnosti klasifikacije. Ipak, uslijed velikog broja potrebnih vrednovanja, ovi omotači imaju sklonost pretjerane



Slika 3.9: Razlika u trajanju unaprijeđenih omotača i njihovih proširenja ovisno o dimenzionalnosti skupa podataka

prilagodbe rješenja korištenom skupu za vrednovanje. Osim toga, zbog svoje stohastičke prirode, ponavljanjem pretrage nerijetko pronalaze različita rješenja što otežava stjecanje uvida u relevantnost pojedinih značajki za promatrani problem klasifikacije. S obzirom na multimodalnost problema odabira značajki, tijekom pretrage omotača moguće je prikupiti arhivu raznolikih rješenja podjednake kvalitete koji čine određeno znanje o samom problemu klasifikacije. Objedinjavanjem tako stečenog znanja mogu se prepoznati relevantne značajke za promatrani problem te formirati kvalitetnija i manja rješenja od onih koje nudi omotač. Rezultati provedene eksperimentalne analize pokazuju da se predloženim proširenjem bio-inspiriranih omotača povećava njihova stabilnost uslijed višestrukog izvođenja pretrage te ostvaruje bolja sposobnost generalizacije klasifikatora na većini korištenih skupova podataka. Također, zamijećeno je da povećanje opće uspješnosti klasifikatora prvenstveno proizlazi upravo iz poboljšanja uspješnosti prepoznavanja manjinske klase.

Pokazano je kako glavnina učinka predloženog proširenja proizlazi iz postupka objedinjavanja, koji uz relativno mali broj vrednovanja uspijeva formirati novo rješenje sastavljeno od malog broja relevantnih značajki. Provođenje ovog postupka pretežno je korisnije te vremenski manje zahtjevno od produljenja pretrage omotača. No, preduvjet za njegovo provođenje jest prikupljanje arhive kvalitetnih i raznolikih rješenja, koja se može razlikovati po strukturi i veličini, ovisno o korištenom bio-inspiriranom algoritmu. Ipak, učinkovitost predloženog proširenja ispitana je za razne često korištene bio-inspirirane algoritme te načine vrednovanja rješenja tijekom pretrage. Za sve upotrijebljene omotače pokazano je da ugradnja predloženog proširenja donosi povoljne rezultate u smislu kvalitete i veličine formiranih rješenja te povećava stabilnost omotača u smislu višestrukog izvođenja pretrage te preslagivanja skupa podataka. Time ono predstavlja ispunjenje prijedloga prvog izvornog znanstvenog doprinosa ove disertacije.

4

Predobrada neuravnoteženih skupova podataka preuzorkovanjem

PREUZORKOVANJE je važan postupak predobrade neuravnoteženih skupova podataka kojim se nastoji povećati broj manjinskih primjeraka te olakšati učenje koncepta manjinske klase. S obzirom na njegovu zastupljenost u literaturi, ovaj postupak može se smatrati uobičajenim i primjerenim pristupom za ublažavanje problema neuravnoteženosti klasa. Jedan od najistaknutijih algoritama za preuzorkovanje u literaturi jest algoritam SMOTE, koji je ujedno podvrgnut brojnim izmjenama s ciljem prevladavanja njegovih nedostataka. Ovo poglavlje daje kratak osvrt na algoritme za preuzorkovanje, s posebnim naglaskom na unaprijeđene inačice algoritma SMOTE. Nakon pregleda literature, opisan je prijedlog novog unaprjeđenja algoritma SMOTE kojim se nastoji pojednostaviti uporaba izvornog algoritma te održati ili nadmašiti kvaliteta njegove izvedbe. Predloženi algoritam ujedno predstavlja prijedlog drugog izvornog znanstvenog doprinosa, a zasniva se na uklanjanju parametara algoritma te novom pristupu stvaranja sintetičkih primjeraka manjinske klase prema unutarnjim karakteristikama skupa podataka. Performanse predloženog algoritma eksperimentalno su ispitane na standardnim skupovima podataka iz literature te su uspoređene s performansama algoritma SMOTE i nekoliko njegovih unaprijeđenih inačica.

4.1 Uvod u preuzorkovanje manjinske klase

Učenje iz neuravnoteženih skupova podataka obično rezultira klasifikacijskim modelima koji su pristrani većinskoj klasi, što znači da imaju lošu sposobnost prepoznavanja manjinske klase. Jedan od glavnih razloga zbog kojeg dolazi do navedenog jest nedostatan broj primjeraka koji bi na odgovarajući način predstavili koncept manjinske klase. Stoga se preuzorkovanje nameće kao prikladan pristup za olakšavanje prepoznavanja koncepta manjinske klase jer ima za cilj povećati broj manjinskih primjeraka u skupu podataka. Osim toga, stvaranje (sintetičkih ili umjetnih) manjinskih primjeraka općenito se smatra pogodnijim pristupom za ublažavanje problema neuravnoteženosti klasa u odnosu na uklanjanje većinskih primjeraka [157], s obzirom na to da se na taj način mogu ukloniti važni primjerci iz skupa podataka [158].

Distribucija manjinskih primjeraka u ulaznom prostoru određuje koncept manjinske klase promatranog problema, a povećanje broja primjeraka koji čine ovaj koncept olakšava njegovo učenje. Idealan algoritam preuzorkovanja stvorio bi sintetičke primjerke prema distribuciji postojećih manjinskih primjeraka, no ona u pravilu nije poznata. Određeni algoritmi pokušavaju približno odrediti ovu distribuciju na temelju dostupnih primjeraka i koristiti je za stvaranje sintetičkih primjeraka koji su dobri predstavnici postojećeg koncepta manjinske klase [159]. Međutim, ovi algoritmi zahtijevaju velik broj primjeraka da bi se približno odredila njihova distribucija (u [159] su korišteni skupovi podataka s nekoliko tisuća manjinskih primjeraka), kojih nema mnogo u uobičajenim neuravnoteženim skupovima podataka. Stoga se najpopularniji algoritmi preuzorkovanja oslanjaju na prikupljanje lokalnih informacija o manjinskim primjercima i koriste ih za stvaranje novih primjeraka [69, 160].

Najjednostavniji predstavnik algoritama za preuzorkovanje jest metoda nasumičnog preuzorkovanja koja nasumično odabire i umnožava postojeće manjinske primjerke sve dok se ne postigne željena razina uravnoteženosti. Međutim, umnožavanje nasumičnih primjeraka može dovesti do povećanja šuma u skupu podataka [161] te do prenaučenosti klasifikatora [68]. S ciljem izbjegavanja ovih neželjenih učinaka preuzorkovanja, Chawla et al. su u [69] predložili algoritam SMOTE koji se zasniva na ideji stvaranja sintetičkih primjeraka kao konveksnih kombinacija (bliskih) parova postojećih manjinskih primjeraka. Iako se može očekivati da će izvođenje ovog algoritma promijeniti postojeću distribuciju manjinskih primjeraka, uvrštavanje takvih sintetičkih primjeraka u skup podataka ipak u većini slučajeva pridonosi poboljšanju uspješnosti prepoznavanja manjinske klase. Razlog tome je što je klasifikatorima uglavnom lakše naučiti koncept manjinske klase ako se poveća broj primjeraka koji ju predstavljaju, a pritom se ne razlikuju bitno od postojećih. Algoritam SMOTE se u literaturi afirmirao kao predstavnik algoritama preuzorkovanja, a velik broj novopredloženih algoritama usvaja određene koncepte njegova načina rada.

Algoritam 4.1: Nacrt rada algoritma SMOTE

```
Izdvoji skup manjinskih primjeraka M iz skupa podataka za treniranje;
Postavi vrijednosti parametara k i q;
Definiraj skup sintetičkih primjeraka S = ∅;
za svaki x ∈ M čini
    Odredi k-susjedstvo N_k(x) od x;
    za i := 1, ..., q čini
        Nasumično odaberi x^r ∈ N_k(x);
        Stvori sintetički primjerak s prema (4.1) ili (4.2);
        S := S ∪ s;
    kraj za
kraj za svaki
```

4.1.1 Algoritam SMOTE

U odnosu na trivijalnu metodu nasumičnog preuzorkovanja, algoritam SMOTE ne stvara duplikate postojećih manjinskih primjeraka nego uvodi mehanizam za stvaranje sintetičkih primjeraka koji se nalaze u njihovu susjedstvu. Ovaj algoritam se u suštini sastoji od dviju osnovnih procedura koje se opetovano provode nad skupom manjinskih primjeraka $\mathcal{M} \subset \mathcal{X}$, a to su određivanje susjedstva te stvaranje sintetičkih primjeraka. Za svaki primjerak manjinske klase $\mathbf{x} \in \mathcal{M}$, najprije se pronalazi k manjinskih primjeraka koji su mu najbliži te oni čine njegovo k-susjedstvo $\mathcal{N}_k(\mathbf{x})$. Za određivanje susjedstva mogu se koristiti različite mjere udaljenosti, pri čemu se uobičajeno koristi Euklidska udaljenost [91]. Nakon određivanja susjedstva odabranog manjinskog primjerka $\mathbf{x}$, slijedi stvaranje q sintetičkih primjeraka kao konveksnih kombinacija

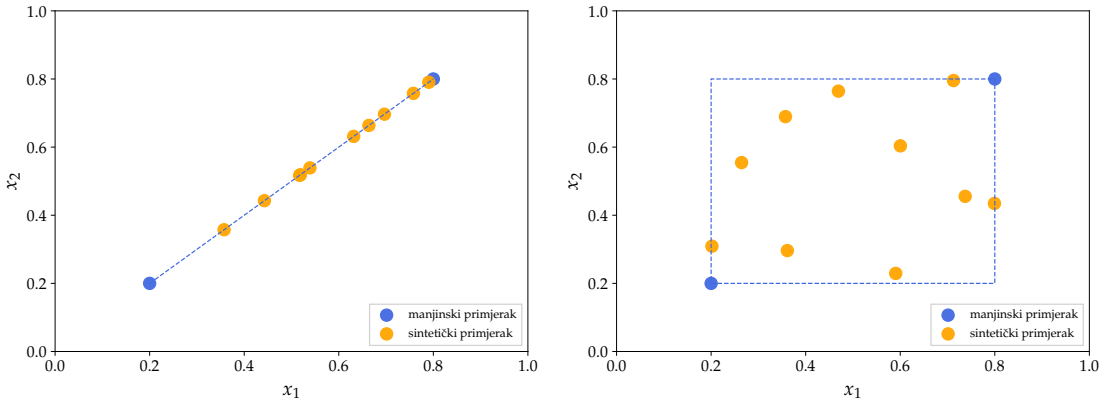
$$\mathbf{s}^i = \mathbf{x} + U_i(0,1) \cdot (\mathbf{x}^{r(i)} - \mathbf{x}), \quad i = 1, \ldots, q, \tag{4.1}$$

gdje $U_i(0,1)$ predstavlja uniformnu slučajnu varijablu iz $[0,1]$, a $\mathbf{x}^{r(i)}$ nasumično odabrani primjerak iz $\mathcal{N}_k(\mathbf{x})$. Kao što se može primijetiti, algoritam ima dva korisnički definirana parametra, veličinu susjedstva k te broj novostvorenih sintetičkih primjeraka za svaki postojeći manjinski primjerak q. Najprikladnije vrijednosti parametara k i q ovise o karakteristikama skupa podataka na kojem se primjenjuje algoritam, a njihovo podešavanje predstavlja dodatan izazov.

Opisani način rada algoritma SMOTE prikazan je algoritmom 4.1. Iako je ovaj algoritam u literaturi često korišten, još postoje nedoumice oko njegova načina stvaranja sintetičkih primjeraka, kao što je istaknuto u [91]. Naime, prema opisu algoritma izloženom u izvornom radu [69], svaki sintetički primjerak stvara se na liniji između promatranog manjinskog primjerka i njegova nasumično odabranog susjeda iz k-susjedstva, kao što je i definirano s (4.1). Međutim, prema pseudokodu danom u istom radu, sintetički primjerci stvaraju se unutar hiperkvadra određenim tim primjercima kao

$$\mathbf{s}^i = \mathbf{x} + \mathbf{U}_i(0,1) \odot (\mathbf{x}^{r(i)} - \mathbf{x}), \quad i = 1, \ldots, q, \tag{4.2}$$

gdje $\mathbf{U}_i(0,1)$ predstavlja d-dimenzionalan vektor uniformnih slučajnih varijabli iz $[0,1]$, a $\odot$



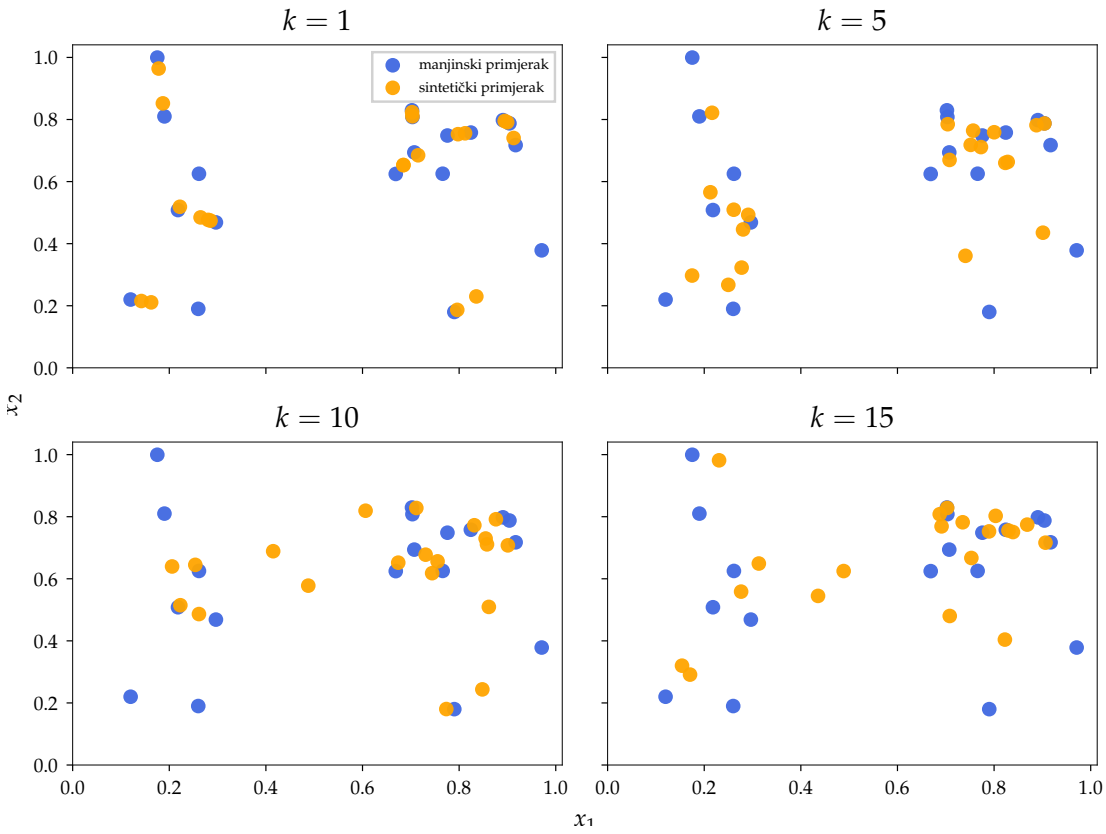
(a) Stvaranje prema (4.1) (b) Stvaranje prema (4.2)

Slika 4.1: Načini stvaranja sintetičkih primjeraka u algoritmu SMOTE

označava Hadamardov umnožak. Ovakav način stvaranja bolje odgovara tvrdnji autora u [70] da je algoritam orijentiran na prostor značajki, a ne na prostor primjeraka. Razlika u značenju načina stvaranja određenih prema (4.1) i (4.2) ilustrirana je slikom 4.1. S obzirom na to da ovi načini stvaranja rezultiraju drugačijim položajem sintetičkih primjeraka, mogu se očekivati i njihovi različiti utjecaji na izvedbu klasifikatora. Oba načina eksperimentalno su uspoređena u [91], a rezultati te analize daju naslutiti da stvaranje prema (4.2) više doprinosi poboljšanju izvedbe klasifikatora. Unatoč tome što je stvaranje prema (4.1) često kritizirano [162, 163], ono se zbog svoje zastupljenosti može smatrati uvriježenim načinom stvaranja sintetičkih primjeraka u algoritmu SMOTE. Pri tome, inačica algoritma SMOTE koja ga usvaja u pravilu se koristi u usporedbama s njegovim unaprijeđenim inačicama u literaturi.

4.1.2 Nedostaci algoritma SMOTE

Iako je algoritam SMOTE iznimno popularan zbog svoje učinkovitosti i jednostavnosti [70], valja spomenuti da ima određene nedostatke koji mogu štetno utjecati na kvalitetu izvedbe klasifikatora. Ovi nedostaci izvornog algoritma često se koriste za motiviranje prijedloga njegovih unaprijeđenih inačica. Najistaknutiji nedostatak algoritma jest taj što on razmatra samo manjinske primjerke prilikom određivanja njihova k-susjedstva, a zanemaruje većinske. Uzimajući u obzir činjenicu da stvarni neuravnoteženi skupovi podataka mogu imati vrlo mali broj manjinskih primjeraka te određeni stupanj preklapanja klasa, za očekivati je da će područja ulaznog prostora u kojem se nalaze tako definirana susjedstva sadržavati i primjerke većinske klase. Stoga, stvaranje sintetičkih primjeraka u tom području može dodatno povećati stupanj preklapanja klasa [164, 165] te tako povećati složenost skupa podataka. Ovakvo ponašanje algoritma izraženije je pri velikim vrijednostima parametra k jer će veća susjedstva vjerojatno sadržavati više većinskih primjeraka. Osim toga, rizik dodatnog povećanja stupnja preklapanja klasa značajniji je kada je manjinska klasa podijeljena u



Slika 4.2: Utjecaj veličine susjedstva na položaj sintetičkih primjeraka u algoritmu SMOTE

podkoncepte [166] jer područja susjedstva lako mogu nadilaziti veličine ovih podkoncepata. S druge strane, smanjenjem susjedstva, učinak algoritma SMOTE postaje sve sličniji učinku nasumičnog preuzorkovanja jer se sintetički uzorci postavljaju bliže postojećim manjinskim uzorcima, što je ilustrirano slikom 4.2. Općenito, stvaranje sintetičkih primjeraka u području preklapanja klasa može doprinijeti poboljšanju uspješnosti prepoznavanja manjinske klase (jer se broj manjinskih primjeraka povećava), ali često naušrtb uspješnosti prepoznavanja većinske klase, a ponekad i na štetu opće izvedbe klasifikatora.

Drugi ključni nedostatak algoritma SMOTE jest taj što on jednakomjerno preuzorkuje susjedstva postojećih manjinskih primjeraka, unatoč tome što nisu svi primjerci jednako važni za učenje koncepta manjinske klase [167, 168]. Većini klasifikatora je lakše naučiti koncept manjinske klase u području ulaznog prostora koji ima najveću gustoću manjinskih primjeraka. Ipak, s obzirom na to da je broj manjinskih primjeraka općenito oskudan te da oni mogu biti okruženi većinskim primjercima, uspješno prepoznavanje ovih primjeraka zahtjevan je zadatak čak i u području njihove najveće gustoće. S druge strane, klasifikator generalno u većoj mjeri pogrešno klasificira manjinske primjerke koji su udaljeniji od područja njihove najveće gustoće (stršćeći primjerci) te one koje se nalaze uz granicu s većinskom klasom [169]. Iako preuzorkovanje susjedstva takvih primjeraka potencijalno može olakšati njihovo prepoznavanje kao pripadnika manjinske klase, ono nosi rizik povećanja stupnja preklapanja klasa u skupu podataka. Osim toga, udaljeniji primjerci mogu predstavljati šum u

skupu podataka i time ne odražavati koncept manjinske klase. Međutim, SMOTE preuzorkuje takve primjerke u jednakoj mjeri kao i ostale, povećavajući šum u skupu podataka, a time i mogućnost prenaučenosti klasifikatora [170].

Treba spomenuti i manju djelotvornost algoritma SMOTE na problemima klasifikacije velike dimenzionalnosti [81] te pri apsolutnoj rijetkosti manjinskih primjeraka [82]. U oba slučaja, susjedi su vrlo udaljeni pa sintetički primjerci mogu biti smješteni daleko od postojećih primjeraka. Postupak odabira značajki može se koristiti za smanjenje udaljenosti postojećih manjinskih primjeraka u ulaznom prostoru, što olakšava učenje koncepta manjinske klase i poboljšava učinak metoda uzorkovanja [16, 82].

4.2 Unaprijeđene inačice algoritma SMOTE

S ciljem prevladavanja spomenutih nedostataka algoritma SMOTE, u literaturi je predloženo preko 100 njegovih unaprijeđenih inačica [70] te se ovaj trend i dalje nastavlja. Većina ovih algoritama prvenstveno preoblikuje osnovne procedure izvornog algoritma i to najčešće izmjenom sadržaja susjedstva manjinskih primjeraka te novim načinom stvaranja sintetičkih primjeraka. Ipak, poneki algoritmi uključuju i dodatne metode poput grupiranja podataka, filtriranja šuma, poduzorkovanja i druge. S obzirom na njihov pozamašan broj, u nastavku nije moguće detaljno opisati sva predložena unaprijeđenja. Pregled literature stoga je usmjeren na one algoritme koji mijenjaju osnovne procedure algoritma SMOTE, dok su složenija unaprijeđenja sažeto opisana. Detaljniji pregled većine unaprijeđenih inačica algoritma SMOTE može se pronaći u [70].

4.2.1 Pregled literature

Jedna od istaknutijih skupina unaprijeđenih inačica algoritma SMOTE u literaturi temelji se na jednom od najstarijih i najpopularnijih takvih unaprijeđenja, algoritmu Borderline-SMOTE, koji su predložili Han et al. u [169]. Ovaj algoritam preuzorkuje susjedstva samo onih manjinskih primjeraka koji su pretežno okruženi većinskim primjercima, kako bi olakšao prepoznavanje primjeraka manjinske klase koji se nalaze uz granicu s većinskom klasom. Valja napomenuti da ovaj algoritam prilikom formiranja susjedstva uzima u obzir sve primjerke, a ne samo one manjinske. S druge strane, algoritam izbjegava preuzorkovanje onih primjeraka čija se susjedstva sastoje većinom od manjinskih primjeraka ili pak isključivo od većinskih primjeraka (takve primjerke proglašava šumom). Bunkhumpornpat et al. su u [171] predložili algoritam Safe-Level-SMOTE kao preinaku algoritma Borderline-SMOTE koja uvodi dodatna pravila za kategoriziranje manjinskih primjeraka. Uz razmatranje susjedstva odabranog manjinskog primjerka, algoritam Safe-Level-SMOTE također razmatra i susjedstvo njegovog nasumično odabranog susjeda iste klase, kako bi bolje razlikovao šum od graničnih primjeraka. Algoritam postavlja sintetički primjerak bliže onom primjerku

koji ima više manjinskih susjeda, a preuzorkovanje se ne izvodi ako promatrani primjerak u svom susjedstvu nema manjinskih primjeraka. Za razliku od algoritma Borderline-SMOTE, algoritam Safe-Level-SMOTE postavlja sintetičke primjerke bliže području veće gustoće postojećih manjinskih primjeraka, kako bi ublažio povećanje stupnja preklapanja klasa u skupu podataka. Nadalje, Maciejewski i Stefanowski su u [166] predložili algoritam LN-SMOTE, kao proširenje algoritma Safe-Level-SMOTE, u kojem se nasumično odabrani manjinski primjerak odbacuje iz susjedstva promatranog primjerka te ga zamjenjuje zadnji susjed u $(k+1)$-susjedstvu. Na ovaj način se strože tretira situacija kada su dva promatrana manjinska primjerka jedan drugome jedini susjedi iz manjinske klase te ih algoritam LN-SMOTE percipira kao šum i ne uzima u obzir za preuzorkovanje, za razliku od algoritma Safe-Level-SMOTE. Kako bi također ublažili uvođenje sintetičkih primjeraka u područje većinske klase, Das et al. su u [172] predložili algoritam Reverse-SMOTE koji preuzorkuje samo one primjerke koji imaju većinu manjinskih primjeraka u svom susjedstvu, pri čemu je broj sintetičkih primjeraka stvorenih u okolini svakog primjerka približno jednak recipročnom omjeru neuravnoteženosti njegovog susjedstva. Moguće je primijetiti da kod iznimno neuravnoteženih skupova podataka postoji rizik da ovaj algoritam stvori vrlo malo sintetičkih primjeraka. Također je zanimljivo istaknuti da algoritam Reverse-SMOTE ne koristi susjede iz prethodno spomenutog susjedstva pri stvaranju sintetičkih primjeraka, već pronalazi obrnuto susjedstvo promatranog primjerka (sve manjinske primjerke kojima je promatrani primjerak najbliži susjed) te potom stvara sintetičke primjerke na linijama između njega i tih susjeda. Općenito, ova skupina algoritama nastoji ublažiti ranije spomenute nedostatke algoritma SMOTE odlučivanjem o provedbi preuzorkovanja za određeni primjerak na temelju omjera većinskih i manjinskih primjeraka u njegovu susjedstvu.

Izbjegavanje jednakomjernog uzorkovanja moguće je postići prilagođavanjem vrijednosti parametra q za svaki manjinski primjerak u skupu podataka. Jedan od istaknutijih algoritama koji primjenjuje ovu ideju jest ADASYN [173], u kojem se vrijednost parametra q množi s normaliziranim (na temelju svih k-susjedstava) udjelom većinskih primjeraka u susjedstvu promatranog manjinskog primjerka. Algoritam stoga stvara više sintetičkih primjeraka uz granicu s većinskom klasom, kako bi pridonio njihovu uspješnijem prepoznavanju. Ipak, ovakvo ponašanje algoritma može rezultirati povećanjem šuma jer će se najviše preuzorkovati oni primjerci koji nemaju niti jednog susjeda iz manjinske klase. Na sličan način, Torres et al. su u [174] predložili algoritam SMOTE-D, u kojem se vrijednost parametra q za svaki manjinski primjerak množi s normaliziranim prosjekom te standardnom devijacijom njegovih udaljenosti od svojih susjeda iz manjinske klase. Tako se u najvećoj mjeri preuzorkuju susjedstva onih manjinskih primjeraka koji su najudaljeniji od ostalih primjeraka iste klase, a najmanje područja ulaznog prostora s najvećom gustoćom manjinskih primjeraka. S druge strane, Prusty et al. su u [175] predložili algoritam Weighted-SMOTE, u kojem se za svaki manjinski primjerak vrijednost parametra q množi s brojem obrnuto proporcionalnom normaliziranom zbroju njegovih udaljenosti od manjinskih primjeraka iz njegova susjedstva. Na

ovaj način se u području najveće gustoće manjinskih primjeraka stvara najviše sintetičkih primjeraka, što se suprostavlja idejama iz algoritama ADASYN i SMOTE-D.

Iako je način kreiranja sintetičkih primjeraka iz algoritma SMOTE [definiran prema (4.1)] zadržan u većini njegovih unaprijeđenih inačica, određeni algoritmi uvode nove načine njihova stvaranja. Calleja i Fuentes su u [176] predložili algoritam Distance-SMOTE koji stvara sintetičke primjerke na liniji između promatranog manjinskog primjerka te točke koja predstavlja središte konveksne ljuske koju omeđuju primjerci u njegovu susjedstvu. Iako se na ovaj način svi susjedi koriste pri stvaranju novih sintetičkih primjeraka, područje ulaznog prostora u kojem se oni mogu smjestiti je ograničeno u odnosu na ono kod algoritma SMOTE jer se ovi primjerci opetovano (q puta) stvaraju na liniji između dva ista primjerka. Kako bi dodatno proširili koncept manjinske klase, Dong i Wang su u [162] predložili algoritam Random-SMOTE koji stvara sintetičke primjerke kao konveksne kombinacije promatranog manjinskog primjerka i dva nasumično odabrana manjinska primjerka iz cijelog skupa podataka. Ovaj algoritam u suštini predstavlja pojednostavljenje algoritma SMOTE jer otklanja potrebu za određivanjem susjedstava postojećih primjeraka. Nadalje, Zheng et al. su u [163] predložili algoritam SNOCC koji stvara sintetičke primjerke kao konveksne kombinacije onih susjeda promatranog primjerka čija je udaljenost od tog primjerka manja od njegove prosječne udaljenosti od svojih susjeda. Tako je novostvoreni sintetički primjerak rezultat konveksne kombinacije minimalno dva, a maksimalno k manjinskih primjeraka. Primarni cilj algoritama Distance-SMOTE, Random-SMOTE i SNOCC jest dodatno proširiti područje za uvrštavanje sintetičkih primjerka. Međutim, niti jedan od njih ne sprječava jednakomjerno uzorkovanje svih manjinskih primjeraka niti uzima u obzir položaj većinskih primjeraka prilikom određivanja njihova susjedstva što može dovesti do povećanja složenosti skupa podataka, kao što je ranije opisano.

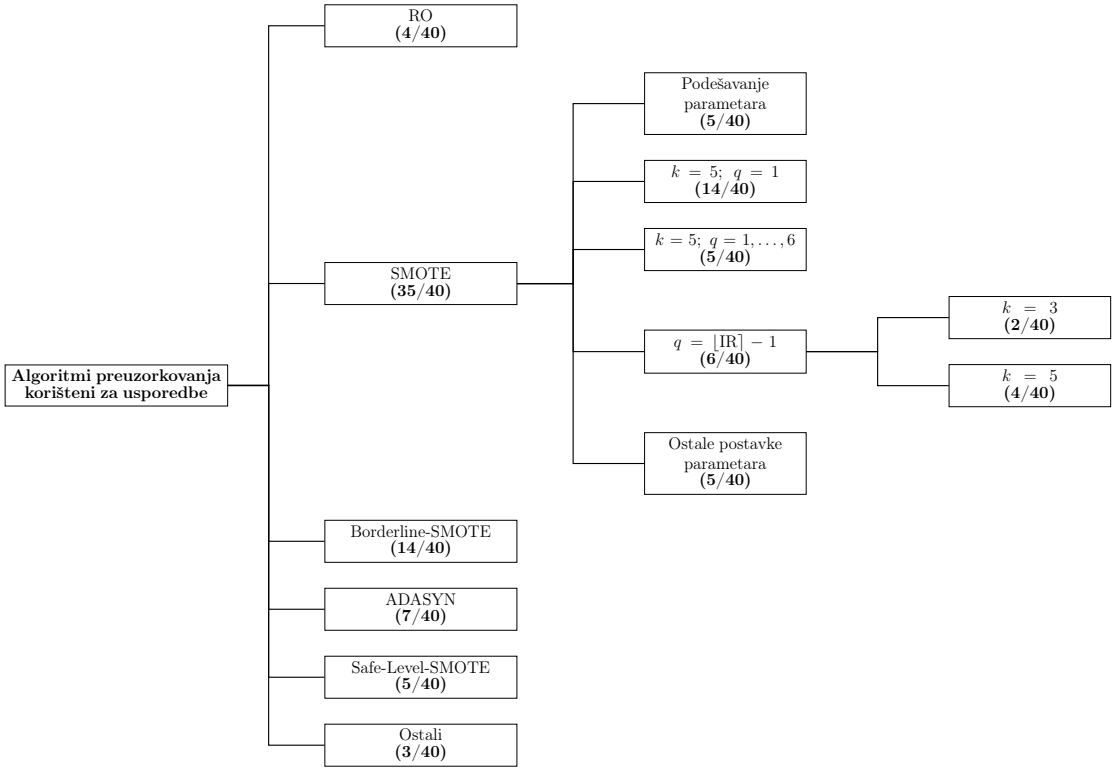
Nekolicina algoritama predstavlja generalizaciju ranije spomenutih algoritama preuzorkovanja jer uvode više različitih strategija za određivanje susjedstva i stvaranje sintetičkih primjeraka, kojima se upravlja dodatnim parametrima. Gazzah i Amara su u [177] predložili novi algoritam za preuzorkovanje koji stvara sintetičke primjerke uvrštavanjem većeg broja manjinskih primjeraka u razne polinome čije je koeficijente potrebno prethodno podesiti. Douzas i Bacao su u [178] predložili algoritam G-SMOTE, koji stvara sintetičke primjerke u d-sferi (pri čemu d označava dimenzionalnost skupa podataka) određenoj promatranim manjinskim primjerkom i njegovim susjedom iz jedne od klasa, ovisno o parametru koji određuje strategiju odabira susjeda. Pri tome, d-sfera uvijek ima središte u promatranom manjinskom primjerku, a može biti omeđena nasumično odabranim manjinskim primjerkom iz njegova susjedstva ili pak najbližim primjerkom iz većinske klase. Na sam položaj stvorenog sintetičkog primjerka unutar d-sfere utječu vrijednosti dodatnih parametara koje je potrebno podesiti prije upotrebe algoritma. Koncept d-sfere određene promatranim manjinskim primjerkom i njegovim najbližim susjedom iz većinske klase koriste i Pradipta et al. u algoritmu Radius-SMOTE [179]. Prednost ovog algoritma jest ta što eliminira parametar k

iz izvornog SMOTE algoritma jer se svi sintetički primjerci stvaraju unutar d-sfere na liniji koja prolazi njezinim centrom, a omeđena je najbližim susjedom iz većinske klase. Međutim, ovaj algoritam provodi i poduzorkovanje svih manjinskih primjeraka koji su pogrešno klasificirani klasifikatorom 5-NN, što može dovesti do nastajanja problema apsolutne rijetkosti manjinskih primjeraka.

Konačno, brojni algoritmi predstavljaju proširenja algoritma SMOTE koja uvode dodatne, u pravilu složene, procedure u algoritam s ciljem prevladanja njegovih nedostataka. Primjerice, razne unaprijeđene inačice algoritma SMOTE oslanjaju se na upotrebu grupiranja podataka radi pronalaženja područja ulaznog prostora s najvećom gustoćom manjinskih primjeraka u koje potom uvrštavaju sintetičke primjerke [180–182]. Osim toga, znatan broj algoritama preuzorkovanja nastoji utvrditi šum korištenjem raznih metoda filtriranja šuma zasnovanih na grupiranju podataka [183], ansamblima klasifikatora [184], konceptima teorije grubih skupova [185, 186] te korištenju bio-inspiriranih algoritama optimizacije [187]. Velik broj algoritama u svoj rad ugrađuje i postupak poduzorkovanja nastojeći ukloniti one većinske primjerke koji se nalaze u području preklapanja klasa [188, 189]. Povrh toga, određene unaprijeđene inačice algoritma SMOTE uvode različite postupke za smanjenje dimenzionalnosti [190–192] s ciljem smanjivanja udaljenosti postojećih manjinskih primjeraka u ulaznom prostoru. Općenito, navedeni složeni algoritmi preuzorkovanja mogu se smatrati proširenjima algoritma SMOTE jer uvode dodatne procedure koje povećavaju vremensku složenost izvornog algoritma te otežavaju njegovu uporabu zbog velikog broja dodatnih parametara koji su potrebni za upravljanje načinom rada dodanih procedura.

4.2.2 Kritički osvrt

Popriličan broj algoritama preuzorkovanja dostupnih u literaturi čini odabir prikladnog algoritma za dani problem iznimno teškim. Moglo bi se očekivati da novije unaprijeđene inačice algoritma SMOTE uspijevaju prevladati nedostatke svojih prethodnika i više pridonijeti poboljšanju izvedbe klasifikatora. Međutim, detaljniji pregled literature otkriva određene nedostatke radova u kojima se ovi algoritmi predlažu. Kao prvo, rijetko se osim generičke motivacije pruža dublje obrazloženje predloženih izmjena te se njihova učinkovitost uglavnom potkrepljuje boljim performansama unaprijeđenih algoritama u odnosu na nekolicinu drugih algoritama korištenih u eksperimentalnoj analizi. Pri tome, za usporedbu se najčešće rabe jednostavniji i stariji algoritmi poput nasumičnog preuzorkovanja, SMOTE, Borderline-SMOTE i ADASYN, dok se novije unaprijeđene inačice koriste u vrlo maloj mjeri. Uvid u to daje slika 4.3 koja prikazuje najčešće algoritme preuzorkovanja korištene za potrebe eksperimentalnih analiza novopredloženih unaprijeđenih inačica algoritma SMOTE u ukupno 40 radova u literaturi. S obzirom na to da je algoritam SMOTE najzastupljeniji algoritam u tim analizama, na slici su također prikazane i najčešće postavke njegovih parametara koje se uglavnom preuzimaju iz prijašnjih radova. Uz navedene nedostatke eksperimentalnih



Slika 4.3: Zastupljenost algoritama preuzorkovanja korištenih za usporedbe s unaprijeđenim inačicama algoritma SMOTE

analiza u pregledanim radovima, treba napomenuti da se one često temelje na skromnom broju skupova podataka (primjerice, u [169] su korištena četiri skupa podataka, a u [171] tek dva) što otežava stjecanje uvida u općenitu učinkovitost ovih algoritama. S druge strane, načini rada novopredloženih algoritama uglavnom su motivirani njihovim djelovanjem na proizvoljno konstruiranim dvodimenzionalnim sintetičkim skupovima podataka. Stoga nije iznenađujuće da su strategije preuzorkovanja unutar brojnih algoritama često oprečne te je moguće zaključiti da su one pristrane određenim strukturama neuravnoteženih skupova podataka.

Zbog velikog broja unaprijeđenih inačica algoritma SMOTE te spomenutih manjkavosti radova u kojima su predložene, u literaturi je provedeno nekoliko opsežnijih analiza ovih algoritama [74, 90, 91]. Rezultati ovih eksperimentalnih analiza sugeriraju da ne postoje statistički značajne razlike između unaprijeđenih inačica algoritma SMOTE te se općenito ne mogu istaknuti napredniji algoritmi prema njihovu doprinosu izvedbi klasifikatora. Ovi rezultati potvrđuju tvrdnje određenih istraživanja [158, 193] koja ukazuju na to da nije moguće očekivati da pojedinačni algoritam preuzorkovanja ostvari najbolje performanse na svim skupovima podataka. Štoviše, najbolje rangirani algoritmi u eksperimentalnoj analizi provedenoj u [90] u prosjeku nadmašuju SMOTE do 1% u iznosu standardnih mjera uspješnosti klasifikacije (AUC i F1). Uz to, zajednička svojstva koja se mogu izdvojiti najbolje rangiranim algoritmima u [90] jesu jednostavnost te robusnost na razne unutarnje karakteristike

skupova podataka (kao primjerice, preklapanje klasa, šum te apsolutnu rijetkost manjinskih primjeraka). S druge strane, složeniji algoritmi pokazali su se osjetljivijima na velik omjer neuravnoteženosti, apsolutnu rijetkost manjinskih primjeraka i veliku dimenzionalnost skupova podataka. U slučaju ovih karakteristika skupa podataka, grupiranje podataka postaje nepouzdano, većina manjinskih primjeraka može se proglasiti šumom, a sposobnost konvergencije određenih pristupa za smanjenje dimenzionalnosti korištenih unutar ovih algoritama je smanjena. Ovi rezultati dovode u pitanje korisnost uvođenja složenijih mehanizama u relativno jednostavnu proceduru preuzorkovanja koju provodi algoritam SMOTE, budući da ono ne pridonosi značajnom povećanju performansi, a uzrokuje povećanje vremenske složenosti te smanjenje robusnosti izvornog algoritma.

Povrh toga, uvođenje dodatnih procedura u algoritam SMOTE najčešće povlači potrebu za dodavanjem novih parametara koji kontroliraju njihov način rada, a čije podešavanje predstavlja značajan izazov. Kao što je prethodno spomenuto, za izvođenje algoritma SMOTE potrebno je podesiti vrijednosti dvaju parametara, broja sintetičkih primjeraka q i veličinu susjedstva k. Kako bi olakšali ovaj zadatak, Chawla et al. su u [194] predložili automatizirani pristup podešavanju parametra q, koji se temelji na opetovanom vrednovanju izvedbe klasifikatora postupkom unakrsne provjere korištenjem 10 preklopa. Vrijednost parametra q se slijedno povećava (u koracima po jedan) od početne vrijednosti (predlažu $q = 1$) do vrijednosti za koju se izvedba klasifikatora počinje narušavati. Nadalje, učinkovitost bioinspiriranih algoritama za problem podešavanja parametara algoritma SMOTE ispitali su Zorić et al. u [195]. Pokazano je da ovi algoritmi optimizacije mogu pronaći kvalitetne postavke parametara, ali obično zahtijevaju velik broj vrednovanja rješenja (u [195] je rabljeno 1000 vrednovanja). Rezultati eksperimentalne analize u tom radu daju uvid u značajne razlike u kvaliteti pojedinih kombinacija parametara za određeni problem i sugeriraju da niti jedna kombinacija nije općenito najbolja za više skupova podataka. Ipak, spomenuti postupci podešavanja parametara algoritma SMOTE rijetko su korišteni u literaturi, ponajviše zbog dugog trajanja. Štoviše, uvođenje novih parametara u algoritam preuzorkovanja čini postupak njihova podešavanja još složenijim te stoga nije iznenađujuće da se uobičajeno preuzimaju postavke parametara iz ranijih radova. Ipak, izostanak podešavanja parametara može znatno umanjiti performanse algoritma preuzorkovanja te se velik broj parametara stoga može smatrati nepovoljnom karakteristikom ovih algoritama.

4.3 Prijedlog unaprijeđenog algoritma za preuzorkovanje

Unatoč navedenim nedostacima, algoritam SMOTE ističe se kao jedan od najčešće korištenih algoritama preuzorkovanja u literaturi, prvenstveno zbog svoje jednostavnosti i korisnosti. Kao što je prethodno spomenuto, u nastojanju ublažavanja tih nedostataka, većina njegovih unaprijeđenih inačica uvodi složenije procedure u izvorni algoritam koje povećavaju njegovu

vremensku složenost i otežavaju njegovo korištenje zbog velikog broja dodatnih parametara koji su potrebni za upravljanje načinom rada tih procedura. Međutim, rezultati opsežnijih eksperimentalnih analiza [74, 90, 91] sugeriraju da uvođenje dodatnih procedura ne pridonosi značajnom povećanju performansi, a potencijalno uzrokuje smanjenje robusnosti algoritma na razne unutarnje karakteristike skupova podataka. S obzirom na ove probleme složenih unaprjeđenja algoritma SMOTE, kao izvorni znanstveni doprinos predlaže se nova unaprijeđena inačica kojom se nastoji pojednostaviti uporaba izvornog algoritma uz zadržavanje njegovih performansi. Predloženi algoritam izmjenjuje način rada osnovnih koraka algoritma SMOTE s ciljem izbjegavanja jednakomjernog preuzorkovanja te povećanja stupnja preklapanja klasa u skupu podataka. Pri tome, algoritam ne uvodi dodatne procedure, a odlikuje ga izostanak parametara što značajno olakšava njegovu uporabu u odnosu na SMOTE i njegove unaprijeđene inačice. Predloženi algoritam određuje susjedstva postojećih manjinskih primjeraka te stvara sintetičke primjerke uzimajući u obzir unutarnje karakteristike skupa podataka. Pri određivanju susjedstva, ne vodi se vrijednošću nekog parametra već položajem tog primjerka u odnosu na većinsku klasu, kako bi se izbjeglo povećanje stupnja preklapanja klasa. Osim toga, broj stvorenih primjeraka određen je sadržajem susjedstva svakog manjinskog primjerka, čime se izbjegava jednakomjerno preuzorkovanje te potreba za podešavanjem odgovarajućeg parametra.

4.3.1 Opis predloženog algoritma

Predloženi algoritam sastoji se u suštini od dva koraka koji se opetovano provode nad skupom manjinskih primjeraka $\mathcal{M} \subset \mathcal{X}$, a to su određivanje susjedstva te stvaranje sintetičkih primjeraka. Za svaki primjerak manjinske klase $\mathbf{x} \in \mathcal{M}$, formira se njegovo susjedstvo

$$\mathcal{N} = \{\mathbf{x}_i \in \mathcal{M} : \|\mathbf{x} - \mathbf{x}_i\| \leq \|\mathbf{x} - \mathbf{t}\|, \mathbf{x}_i \neq \mathbf{x}, \mathbf{t} = \underset{\tau \in \mathcal{V}}{\operatorname{argmin}} \|\mathbf{x} - \tau\|\}, \tag{4.3}$$

gdje $\|\cdot\|_2$ predstavlja ℓ_2 (Euklidsku) normu, a $\mathcal{V} \subset X$ skup većinskih primjeraka. Prema tome, susjedstvo $\mathcal{N}$ promatranog manjinskog primjerka $\mathbf{x}$ čine svi manjinski primjerci $\mathbf{x}_i \in \mathcal{M}$, $\mathbf{x}_i \neq \mathbf{x}$, čija je udaljenost od $\mathbf{x}$ manja ili jednaka njegovoj udaljenosti od svog najbližeg susjeda iz većinske klase, odnosno $\mathbf{t} \in \mathcal{V}$. S obzirom na to da se kao funkcija udaljenosti koristi Euklidska udaljenost, moguće je primijetiti da se susjedstvo promatranog manjinskog primjerka nalazi unutar d-sfere [196] (pri čemu d označava dimenzionalnost skupa podataka), koja je određena tim primjerkom $\mathbf{x}$ i njegovim najbližim susjedom iz većinske klase $\mathbf{t}$. Stvaranjem sintetičkih primjeraka unutar tog područja izbjegava se povećanje stupnja preklapanja klasa u skupu podataka. Osim toga, definiranjem susjedstva pomoću najbližeg primjerka iz većinske klase uklanja se potreba za parametrom koji predstavlja veličinu susjedstva jer se ona prilagođava svakom manjinskom primjerku. Prema tome, veličina susjedstva svakog manjinskog primjerka nalazi se u $\{0, \ldots, N_M - 1\}$, gdje N_M predstavlja broj manjinskih

primjeraka u skupu podataka. S druge strane, algoritam SMOTE i većina njegovih unaprjeđenja zahtijevaju podešavanje veličine susjedstva (koja je kod njih univerzalna za sve primjerke) kako bi se postigao kompromis između smanjenja stupnja preklapanja klasa te proširenja koncepta manjinske klase, s obzirom na to da primjerci nisu ravnomjerno raspoređeni u ulaznom prostoru. Nekolicina unaprjeđenja algoritma SMOTE u literaturi [178, 179] pak stvara primjerke unutar d-sfere, no pritom ne koriste susjedne manjinske primjerke već uvode dodatne parametre koji upravljaju položajem sintetičkih primjeraka unutar nje.

Nakon određivanja susjedstva $\mathcal{N}$ promatranog manjinskog primjerka $\mathbf{x}$, slijedi stvaranje sintetičkih primjeraka u tom području, pri čemu je njihov broj jednak $\max\{1, |\mathcal{N}|\}$. Ako susjedstvo $\mathcal{N}$ ne sadrži niti jedan manjinski primjerak, stvara se samo jedan sintetički primjerak konveksnom kombinacijom

$$\mathbf{s} = \mathbf{x} + \alpha \cdot U(0,1) \cdot (\mathbf{x}^r - \mathbf{x}), \tag{4.4}$$

gdje $\mathbf{x}^r$ predstavlja nasumično odabrani primjerak iz $\mathcal{M}$, a $\alpha = \frac{\|\mathbf{t}-\mathbf{x}\|_2}{\|\mathbf{x}^r-\mathbf{x}\|_2}$ faktor skaliranja kako bi se osiguralo da je stvoreni primjerak unutar spomenute d-sfere. Kao i u algoritmu SMOTE, sintetički primjerak stvara se na liniji između $\mathbf{x}$ i $\mathbf{x}^r$, ali ostaje unutar d-sfere da bi se izbjeglo njegovo uvođenje u područje većinske klase. Iako susjedstvo $\mathcal{N}$ ne sadrži manjinske primjerke, ovaj primjerak stvara se zbog činjenice da u stvarnim skupovima podataka manjinski primjerci često imaju najbližeg susjeda iz većinske klase, prvenstveno zbog velikog omjera neuravnoteženosti te stupnja preklapanja klasa. Izbjegavanje njihova preuzorkovanja može rezultirati vrlo malim brojem stvorenih primjeraka te značajno smanjenim učinkom algoritma preuzorkovanja. S druge strane, ako susjedstvo promatranog manjinskog primjerka $\mathbf{x}$ sadrži manjinske primjerke, slijedi stvaranje $|\mathcal{N}|$ sintetičkih primjeraka. Prije stvaranja svakog od njih, nasumično se generira broj susjeda $\kappa \in \{1, \ldots, |\mathcal{N}|\}$ koji se koriste tijekom stvaranja novog primjerka. Nakon toga, formira se prošireno podsusjedstvo $\hat{\mathcal{N}}_\kappa$ koje uz promatrani manjinski primjerak $\mathbf{x}$ sadrži κ nasumično odabranih susjeda iz $\mathcal{N}$, odnosno

$$\hat{\mathcal{N}}_\kappa = \{\mathbf{x}\} \cup \{\mathbf{x}^{r(j)} \in \mathcal{N} : j = 1, \ldots, \kappa\} \ . \tag{4.5}$$

Prošireno podsusjedstvo iznova se definira prethodno stvaranju svakog sintetičkog primjerka, uz ponovno generiranje nasumičnog broja susjeda koje to podsusjedstvo sadržava (κ). Za promatrani manjinski primjerak $\mathbf{x}$ tako se definira $|\mathcal{N}|$ proširenih podsusjedstava, pri čemu se sintetički primjerci stvaraju kao konveksne kombinacije, odnosno težinske aritmetičke sredine

$$\mathbf{s}^i = \frac{\sum\limits_{a_j \in \hat{\mathcal{N}}_{\kappa(i)}} \omega_j \cdot a_j}{\sum \omega_j}, \quad i = 1, \ldots, |\mathcal{N}|, \tag{4.6}$$

gdje je ω_j nasumično stvorena nenegativna težina pridružena primjerku a_j odabranom iz i-tog proširenog podsusjedstva $\hat{\mathcal{N}}_{\kappa(i)}$ za $j = 1, \ldots, |\hat{\mathcal{N}}_{\kappa(i)}|$. Uz nasumične težine, radi pos-

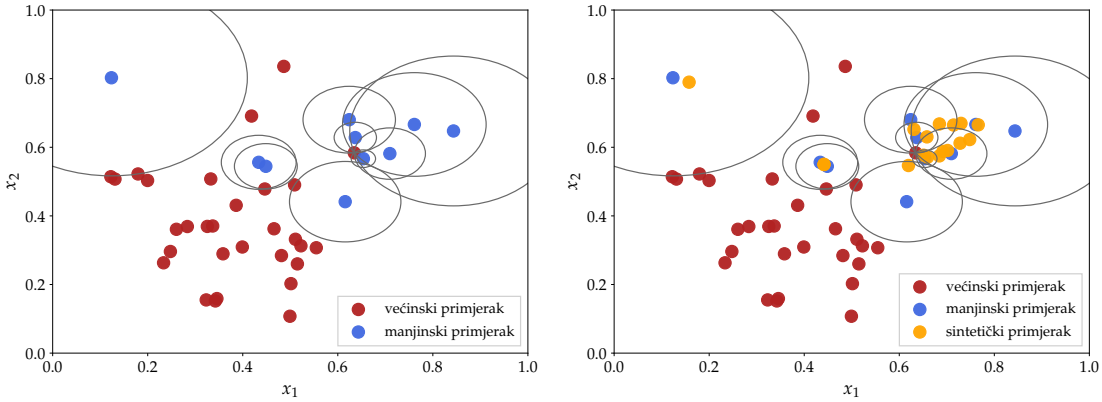
tizanja veće raznolikosti među sintetičkim primjercima, ne koriste se uvijek svi susjedi iz $\mathcal{N}$ tijekom njihova stvaranja, nego nasumično odabran podskup zadane nasumične veličine. Prema navedenom opisu, svaki sintetički primjerak u susjedstvu promatranog primjerka $\mathbf{x}$ predstavlja konveksnu kombinaciju od najmanje dva i najviše $|\mathcal{N}|+1$ manjinskih primjeraka (uključujući i $\mathbf{x}$) te se nalazi unutar konveksne ljuske omeđene korištenim primjercima.

Predloženim načinom stvaranja uklanja se potreba za parametrom iz algoritma SMOTE koji predstavlja univerzalni broj primjeraka stvorenih za pojedini manjinski primjerak (q) jer se broj novostvorenih primjeraka zasebno određuje za svaki manjinski primjerak na temelju veličine njegova susjedstva. Broj stvorenih primjeraka u algoritmu SMOTE uvijek je jednak $q \cdot N_M$, gdje N_M označava broj manjinskih primjeraka u skupu podataka. S druge strane, broj primjeraka stvorenih predloženim algoritmom nalazi se u $\{N_M, \ldots, N_M \cdot (N_M - 1)\}$. Najmanje sintetičkih primjeraka nastaje kada svi manjinski primjerci imaju nula ili jednog susjeda, dok se najviše primjeraka stvara kada susjedstvo svakog manjinskog primjerka čine svi preostali manjinski primjerci. Broj primjeraka stvorenih predloženim algoritmom nije moguće znati unaprijed, a utvrđuje se tijekom izvođenja algoritma uzimajući u obzir unutarnje karakteristike skupa podatka. Nadalje, predloženi algoritam također izbjegava jednakomjerno uzorkovanje s obzirom na to da se u najvećoj mjeri preuzorkuju gusto raspoređena područja manjinske klase, što se u opsežnijim eksperimentalnim analizama [74, 91] pokazalo učinkovitijom strategijom preuzorkovanja. S druge strane, manjinski primjerci koji su primarno okruženi većinskom klasom potencijalno predstavljaju šum u skupu podataka te se u njihovoj okolini stvara tek po jedan sintetički primjerak. Također, u manjoj se mjeri preuzorkuju primjerci koji se nalaze uz granicu s većinskom klasom kako bi se izbjeglo povećanje stupnja preklapanja klasa u skupu podataka. Međutim, očekuje se da će predloženi algoritam ove primjerke višestruko koristiti za stvaranje novih sintetičkih primjeraka jer će oni vjerojatno biti sadržani u susjedstvima drugih manjinskih primjeraka koji su udaljeniji od većinske klase. Osim smanjenja stupnja neuravnoteženosti klasa, predloženi algoritam tako nastoji olakšati učenje koncepta manjinske klase bez povećanja složenosti skupa podataka.

Prijedlog je predstavljen na visokoj razini algoritmom 4.2, a primjer njegova načina rada ilustriran je slikom 4.4. Nije moguće očekivati da će predloženi algoritam biti općenito najbolji na svim skupovima podataka, što daju naslutiti istraživanja u [158, 193], gdje je jasno da ne postoji jedinstveni algoritam preuzorkovanja koji je najprikladniji za svaki skup podataka. No, s obzirom na izostanak parametara, on je trivijalan za korištenje u odnosu na algoritam SMOTE, a posebno u odnosu na mnoge njegove unaprijeđene inačice. Uz to, dodatna odlika predloženog algoritma jest minimalno preuzorkovanje šuma te izbjegavanje povećanja stupnja preklapanja klasa. S druge strane, primarni nedostatak predloženog algoritma može biti relativno mali broj stvorenih primjeraka u slučaju apsolutne rijetkosti manjinskih primjeraka ili velike količine šuma u konceptu manjinske klase. Ostali algoritmi preuzorkovanja u takvim slučajevima mogu imati prednost jer imaju mogućnost podešavanja broja stvorenih primjeraka putem parametara. Međutim, podešavanje njihovih parametara zahtijeva doda-

Algoritam 4.2: Prijedlog unaprijeđene inačice algoritma SMOTE na visokoj razini

Izdvoji skup manjinskih primjeraka $\mathcal{M}$ i skup većinskih primjeraka $\mathcal{V}$ iz skupa podataka za treniranje;
Definiraj skup sintetičkih primjeraka $\mathcal{S} = \emptyset$;
za svaki $\mathbf{x} \in \mathcal{M}$ **čini**
  Pronađi najbližeg susjeda $\mathbf{t} \in \mathcal{V}$;
  $\mathcal{N} = \{\forall \mathbf{x}_i \in \mathcal{M} : \|\mathbf{x} - \mathbf{x}_i\|_2 \leq \|\mathbf{t} - \mathbf{x}\|_2, \mathbf{x}_i \neq \mathbf{x}\}$;
  ako $\mathcal{N} = \emptyset$ **onda**
    Nasumično odaberi $\mathbf{x}^r \in \mathcal{M}$;
    Stvori sintetički primjerak $\mathbf{s}$ prema (4.4);
    $\mathcal{S} := \mathcal{S} \cup \mathbf{s}$;
  kraj ako
  inače
    za $i := 1, ..., |\mathcal{N}|$ **čini**
      Nasumično odaberi $\kappa \in \{1, \ldots, |\mathcal{N}|\}$;
      $\hat{\mathcal{N}}_\kappa := \{\mathbf{x}\} \cup \{\mathbf{x}^{r(j)} \in \mathcal{N} : j = 1, \ldots, \kappa\}$;
      Stvori sintetički primjerak $\mathbf{s}$ prema (4.6);
      $\mathcal{S} := \mathcal{S} \cup \mathbf{s}$;
    kraj za
  kraj ako
kraj za svaki



(a) Određivanje susjedstva (b) Stvaranje sintetičkih primjeraka

Slika 4.4: Primjer načina rada predloženog algoritma

tan napor i vrijeme zbog potrebe za višestrukim provođenjem postupaka preuzorkovanja i treniranja klasifikacijskog modela.

4.3.2 Detalji ugradnje

Detaljniji prikaz ranije izloženog načina rada predloženog algoritma dan je algoritmom 4.3. Za svaki manjinski primjerak u skupu podataka provode se koraci određivanja susjedstva te stvaranja sintetičkih primjeraka u njegovu susjedstvu. Način djelovanja ovih koraka moguće je jasno interpretirati pomoću algoritma 4.3, no postoji nekoliko detalja ugradnje koje valja dodatno protumačiti. Za razliku od algoritma SMOTE, koji radi isključivo s manjinskim primjercima, predloženom algoritmu potreban je cijeli skup podataka kako bi odredio susjedstva manjinskih primjeraka. Pri tome računa (Euklidsku) udaljenost promatranog manjinskog primjerka od svih primjeraka u skupu podataka da bi odredio najbližeg većinskog susjeda i sve susjede iz manjinske klase. Ako susjedstvo sadrži manjinske primjerke, sintetički

Algoritam 4.3: Nacrt rada predložene unaprijeđene inačice algoritma SMOTE

```
Funkcija Preuzorkuj():
    Ulaz: M[N_M][d]      // Manjinski primjerci
          V[N_V][d]      // Većinski primjerci
    Izlaz: S[N_S][d]     // Sintetički primjerci

    S[N_S][d] := []
    za svaki x u M čini
        Pronađi najbližeg susjeda t iz V
        N[N_M][d] := []
        p := 0                  // Veličina susjedstva N
        za svaki x_i u M čini
            ako x ≠ x_i I d(x, x_i) ≤ d(x, t)   // gdje d(·,·) označava Euklidsku udaljenost
             onda
                Ubaci x_i u N
                p := p + 1
            kraj ako
        kraj za svaki
        ako p = 0 onda
            Nasumično odaberi x^r iz M
            Stvori s prema (4.4)
            Ubaci s u S
        kraj ako
        inače
            za i := 1, ..., p čini
                Nasumično odaberi 1 ≤ κ ≤ p
                N̂_κ[κ][d] := [x]
                za j := 1, ..., κ čini
                    Nasumično odaberi n iz N bez zamjene
                    Ubaci n u N̂_κ
                kraj za
                Stvori s prema (4.6)
                Ubaci s u S
            kraj za
        kraj ako
    kraj za svaki
```

primjerci se stvaraju prema (4.6) te oni predstavljaju konveksne kombinacije promatranog primjerka i njegovih nasumično odabranih susjeda. Ovdje je važno naglasiti da se prije stvaranja svakog primjerka, broj susjeda u proširenom podsusjedstvu određenom prema (4.5) nasumično određuje, a susjedi se nasumično biraju iz susjedstva bez zamjene. Drugim riječima, tijekom stvaranja jednog sintetičkog primjerka uvijek se koriste različiti susjedi jer nije moguće uključiti niti jednog susjeda u isto prošireno podsusjedstvo više od jednom. Nadalje, pri stvaranju primjeraka prema (4.6), težine pridružene susjedima nasumično se generiraju za svako podsusjedstvo i mogu predstavljati brojeve iz $[0, R_+\rangle$. Pri tome je poželjno da se one generiraju unutar istog raspona kako bi se ublažio utjecaj različitih raspona vrijednosti, odnosno da bi se izbjeglo pridavanje znatno veće važnosti pojedinom primjerku. Radi pojednostavljenja ugradnje ovog načina stvaranja i bez smanjenja općenitosti, one se nasumično generiraju iz $[0, 1]$. Broj tako stvorenih primjeraka jednak je veličini susjedstva promatranog manjinskog primjerka. U slučaju da susjedstvo ne sadrži manjinske primjerke, stvara se po jedan sintetički primjerak prema (4.4), kako je prethodno objašnjeno i detaljnije prikazano algoritmom 4.3.

4.3.3 Procjena vremenske složenosti

S obzirom na činjenicu da predloženi algoritam predstavlja unaprjeđenje algoritma SMOTE koje uzima u obzir i većinsku klasu prilikom određivanja susjedstva, moguće je zaključiti da ono povećava trošak izvođenja izvornog algoritma. Kako bi se stekao uvid u razmjer tog povećanja, u nastavku je dana procjena vremenske složenosti predloženog algoritma preuzorkovanja. Za svaki manjinski primjerak, prvo se određuje njegovo susjedstvo $[O(N)]$, pri čemu treba izračunati njegovu udaljenost od svih (ukupno N) primjeraka u skupu podataka. Zatim slijedi stvaranje sintetičkih primjeraka, gdje u najgorem slučaju (u smislu računanja vremenske složenosti) susjedstvo svakog manjinskog primjerka mogu činiti preostali manjinski primjerci te se ukupno stvara oko N_M^2 sintetičkih primjeraka $[O(N_M^2)]$. Opisani postupak ponavlja se za svaki manjinski primjerak u skupu podataka, pa je složenost ovog algoritma $O[N_M \times (N + N_M^2)]$. S druge strane, s obzirom na ranije izloženi način rada algoritma SMOTE, njegova vremenska složenost može se izraziti kao $O[N_M \times (N_M \times k + q)]$, pri čemu k i q predstavljaju parametre algoritma. Bitno je napomenuti da prilikom izražavanja složenosti nije uzeta u obzir dimenzionalnost problema, s obzirom na činjenicu da ona jednako utječe na složenost oba algoritma.

Iako predloženi algoritam uzima u obzir i većinsku klasu prilikom određivanja susjedstva, vremenska složenost ove procedure može biti niža od složenosti iste procedure u algoritmu SMOTE, ako je omjer neuravnoteženosti skupa podataka manji od vrijednosti parametra k. S druge strane, postupak stvaranja sintetičkih primjeraka u predloženom algoritmu je složeniji od istog postupka u algoritmu SMOTE, budući da može koristiti više primjeraka u konveksnoj kombinaciji u odnosu na dva u izvornom algoritmu. Međutim, važno je podsjetiti da je vremenska složenost predloženog algoritma izvedena za najgori slučaj (kada susjedstva svih primjeraka čine svi ostali primjerci) te da na razliku u trajanju izvođenja ovih algoritama ponajviše utječu postavke parametara algoritma SMOTE i unutarnje karakteristike skupa podataka.

4.4 Eksperimentalna analiza i rezultati

Kako bi se utvrdila korisnost predložene unaprijeđene inačice algoritma SMOTE, provedena je odgovarajuća eksperimentalna analiza koja je podijeljena u dva dijela. Kao što je ranije spomenuto, primarni cilj predloženog unaprjeđenja jest pojednostaviti korištenje izvornog algoritma uz održavanje ili poboljšanje njegova učinka. Stoga se u prvom dijelu uspoređuju performanse predloženog unaprjeđenja s performansama algoritma SMOTE, uzimajući u obzir utjecaj različitih postavki parametara na kvalitetu izvedbe izvornog algoritma. U drugom dijelu prikazana je usporedba predloženog algoritma s nekoliko unaprijeđenih inačica algoritma SMOTE iz literature, s ciljem stjecanja uvida u doseg njegova poboljšanja izvornog algoritma.

Tablica 4.1: Karakteristike skupova podataka korištenih za potrebe eksperimentalne analize predložene unaprijeđene inačice algoritma SMOTE

Oznaka	Naziv	Izvor podataka	Broj primjeraka	Broj značajki	Broj klasa	IR
$\mathcal{D}_1$	Connectionist Bench	UCI	208	60	2	1.14
$\mathcal{D}_2$	Congressional Voting Records	UCI	435	16	2	1.59
$\mathcal{D}_3$	Ionosphere	UCI	351	34	2	1.79
$\mathcal{D}_4$	Blood Transfusion	UCI	748	4	2	3.20
$\mathcal{D}_5$	New Thyroid1	KEEL	215	5	2	5.14
$\mathcal{D}_6$	Hepatitis	UCI	80	19	2	5.15
$\mathcal{D}_7$	Ecoli3	KEEL	336	7	2	8.60
$\mathcal{D}_8$	Climate	UCI	540	18	2	10.74
$\mathcal{D}_9$	Led7digit1	KEEL	443	7	2	10.97
$\mathcal{D}_{10}$	Zoo3	KEEL	101	16	2	19.20
$\mathcal{D}_{11}$	Abalone-3vs11	KEEL	502	8	2	32.47
$\mathcal{D}_{12}$	Yeast6	KEEL	1484	10	2	41.40

Eksperimentalna analiza provedena je na standardnim skupovima podataka za vrednovanje postupaka preuzorkovanja u literaturi, a njihove karakteristike prikazane su u tablici 4.1. Polovica odabranih skupova podataka ($\mathcal{D}_1$ – $\mathcal{D}_4$, $\mathcal{D}_6$ i $\mathcal{D}_8$) u svom izvornom obliku predstavlja binarne probleme klasifikacije koji su preuzeti s UCI repozitorija. Preostali skupovi podataka ($\mathcal{D}_5$, $\mathcal{D}_7$, $\mathcal{D}_9$ – $\mathcal{D}_{12}$) preuzeti su s KEEL repozitorija [197], a dobiveni su dekompozicijom višeklasnih problema u binarne probleme, tako da primjerci određenih klasa predstavljaju primjerke manjinske klase, a preostali primjerci imaju oznaku većinske klase. Detaljniji opis načina formiranja ovih skupova podataka dan je u dodatku A. Ovi skupovi podataka često se koriste kod usporedbi raznih algoritama preuzorkovanja [52, 90, 91] jer ih je jednostavno prilagoditi da imaju traženi omjer neuravnoteženosti. Odabrani skupovi podataka stoga se znatno razlikuju prema omjeru neuravnoteženosti koji imaju kako bi se ispitala učinkovitost preuzorkovanja za razne stupnjeve neuravnoteženosti klasa. Pri tome, skupovi $\mathcal{D}_1$ – $\mathcal{D}_3$ predstavljaju blago neuravnotežene probleme (IR < 2), skupovi $\mathcal{D}_4$ – $\mathcal{D}_9$ iskazuju značajnu neuravnoteženost klasa ($2 \leq$ IR < 11), dok je u skupovima $\mathcal{D}_{10}$ – $\mathcal{D}_{12}$ omjer neuravnoteženosti izrazito velik (IR > 19). Skaliranje značajki, normalizacijom u raspon $[0, 1]$, izvedeno je kao korak predobrade svakog skupa podataka, kao i u poglavlju 3.

4.4.1 Postavke eksperimenta

Performanse algoritma preuzorkovanja obično se predstavljaju kvalitetom izvedbe klasifikatora na skupu podataka koji je predobrađen tim algoritmom. S obzirom na činjenicu da nisu svi klasifikatori jednaki, za očekivati je da su neki osjetljiviji na problem neuravnoteženosti klasa od drugih. Osim toga, moguće je da način rada nekog algoritma preuzorkovanja više pogoduje izvedbi određenog klasifikatora. Sukladno tome, u eksperimentalnoj analizi korištena su četiri klasifikatora, radi bolje procjene korisnosti i svestranosti predložene unaprijeđene inačice algoritma SMOTE. Odabrani su klasifikatori 1-NN, 5-NN, SVM te MLP koji se često koriste u literaturi za usporedbu raznih algoritama preuzorkovanja [74, 91]. Pri tome, SVM je rabljen s radijalnom funkcijom za jezgru i regularizacijskim parametrom

Tablica 4.2: Postavke parametara algoritma SMOTE korištene za eksperimentalnu analizu

	1-NN	5-NN	SVM	MLP
Podešene postavke parametara	$k = 7$; $q = 3$	$k = 7$; $q = 1$	$k = 7$; $q = 2$	$k = 3$; $q = 2$
Postavke parametara iz literature	$k = 5$; $q = 1$		[169, 186, 198, 199]	
	$k = 3$; $q = \lfloor IR \rceil - 1$		[162, 200]	
	$k = 5$; $q = \lfloor IR \rceil - 1$		[166, 174, 179]	

$C = 1$, a MLP sa 100 čvorova u jedinom skrivenom sloju te zglobnicom (engl. *rectified linear unit*) kao aktivacijskom funkcijom.

U prvom dijelu eksperimentalne analize izvršena je usporedba predloženog algoritma s algoritmom SMOTE, pri čemu je korištena inačica izvornog algoritma koja stvara sintetičke primjerke prema (4.1), što je i uobičajeno u literaturi [70]. Za razliku od predloženog algoritma, algoritmu SMOTE je prije njegova izvođenja potrebno postaviti vrijednosti parametara, odnosno veličinu susjedstva (k) te broj stvorenih primjeraka za svaki postojeći manjinski primjerak (q). Budući da vrijednosti ovih parametara imaju značajan utjecaj na performanse algoritma SMOTE, nekoliko različitih postavki parametara primijenjeno je pri njegovoj usporedbi s predloženim algoritmom. U prvom redu, primijenjene su tri najčešće postavke parametara iz literature, odnosno $(k, q) \in \{(5, 1), (3, \lfloor \mathrm{IR} \rceil - 1), (5, \lfloor \mathrm{IR} \rceil - 1)\}$, pri čemu potonje dvije postavke nastoje ostvariti uravnoteženu raspodjelu primjeraka različitih klasa u skupu podataka. Algoritam SMOTE s ovim postavkama parametara udružen je sa svakim od klasifikatora korištenim u eksperimentalnoj analizi. Povrh toga, za svaki klasifikator provedeno je ograničeno, ali vremenski zahtjevno traženje prikladnih vrijednosti navedenih parametara, a tako dobivene postavke parametara prikazane su u tablici 4.2.

U drugom dijelu eksperimentalne analize predloženi je algoritam uspoređen s nekolicinom unaprijeđenih inačica algoritma SMOTE iz literature koje su navedene u tablici 4.3. Kao i predloženi algoritam, unaprjeđenja odabrana za usporedbu izmijenjuju osnovne korake algoritma SMOTE, bez uvođenja dodatnih složenih procedura u izvorni algoritam. Složenija unaprjeđenja algoritma SMOTE (primjerice, algoritmi koji uključuju metode poput grupiranja podataka, poduzorkovanja i smanjenja dimenzionalnosti) nisu uključena u usporedbu jer su njihova vremenska složenost i velik broj parametara jasni nedostaci u usporedbi s predloženim unaprjeđenjem, a većina ih ne uspijeva nadmašiti izvorni algoritam prema kvaliteti izvedbe, kao što je ranije navedeno. Nasumično preuzorkovanje i Borderline-SMOTE ujedno su i najčešći algoritmi preuzorkovanja koji se rabe za usporedbu s novijim unaprjeđenjima algoritma SMOTE. S druge strane, Weighted-SMOTE i Random-SMOTE nalaze se među najbolje rangiranim algoritmima u opširnijim eksperimentalnim analizama [52, 90, 91]. Kao što je prethodno spomenuto, algoritam Reverse-SMOTE novije je unaprjeđenje algoritma SMOTE koje, slično kao i predloženi algoritam, određuje broj novostvorenih primjeraka zasebno za svaki manjinski primjerak na temelju sadržaja njegova susjedstva. Svakom od

Tablica 4.3: Postavke parametara unaprijeđenih inačica algoritma SMOTE korištene za eksperimentalnu analizu

Algoritam	Oznaka	Opis	1-NN	5-NN	SVM	MLP
Nasumično preuzorkovanje	RO	Nasumično odabire i umnožava postojeće manjinske primjerke.	$q = 3$	$q = 2$	$q = 3$	$q = 3$
Borderline-SMOTE [169]	BSMOTE	Preuzorkuje samo one manjinske primjerke koji u susjedstvu imaju većinu većinskih primjeraka.	$k = 5$ $q = 3$	$k = 5$ $q = 1$	$k = 5$ $q = 2$	$k = 5$ $q = 2$
Random-SMOTE [162]	RASMOTE	Stvara sintetičke primjerke kao konveksne kombinacije promatranog primjerka i dva nasumično odabrana manjinska primjerka.	$q = 3$	$q = 1$	$q = 3$	$q = 3$
Weighted-SMOTE [175]	WSMOTE	Prilagođava broj sintetičkih primjeraka u susjedstvu promatranog manjinskog primjerka na temelju njegove udaljenosti od svojih manjinskih susjeda.	$k = 5$ $q = 2$	$k = 5$ $q = 1$	$k = 5$ $q = 1$	$k = 5$ $q = 2$
Reverse-SMOTE [172]	RESMOTE	Za svaki postojeći manjinski primjerak zasebno određuje broj novostvorenih primjeraka na temelju sadržaja njegova susjedstva te koristi koncept obrnutog susjedstva pri stvaranju sintetičkih primjeraka.	$k = 5$	$k = 5$	$k = 5$	$k = 5$

odabranih algoritama (osim Random-SMOTE) potrebno je definirati veličinu susjedstva (k), dok je za Borderline-SMOTE, Random-SMOTE i Weighted-SMOTE potrebno definirati i broj stvorenih primjeraka za svaki postojeći manjinski primjerak (q). Za ove algoritme je također provedeno traženje prikladnih vrijednosti navedenih parametara, a tako dobivene postavke parametara prikazane su u tablici 4.3.

4.4.2 Metodologija eksperimentalne analize

Tijek eksperimentalne analize započinje podjelom skupova podataka za potrebe treniranja i testiranja klasifikacijskih modela te se nastavlja njihovom predobradom korištenim algoritmima preuzorkovanja i prikupljanjem ostvarenih rezultata o njihovoj izvedbi. Iz svakog skupa podataka redom su stratificirano izdvojeni skupovi za treniranje i testiranje, u omjeru $0.75 : 0.25$. Kako bi se omogućilo testiranje izvedbe korištenih algoritama preuzorkovanja, svaki od njih je primijenjen na izdvojenom skupu za treniranje. Uspješnost treniranog modela na skupu za testiranje tada predstavlja i kvalitetu izvedbe korištenog algoritma preuzorkovanja. Da bi se stekao općenitiji uvid u performanse predloženog algoritma, napravljeno je 30 različitih podjela svakog skupa podataka. Time se za svaku kombinaciju korištenih skupova podataka, klasifikatora i algoritama preuzorkovanja bilježi 30 rezultata o uspješnosti treniranih modela. Treba napomenuti da su za svaki klasifikator također testirani i klasifikacijski modeli koji su trenirani na izvornim skupovima za treniranje (koji nisu predobrađeni postupkom preuzorkovanja) kako bi se stekao uvid u uspješnost klasifikatora kada se ne provodi ovaj postupak ublažavanja neuravnoteženosti klasa.

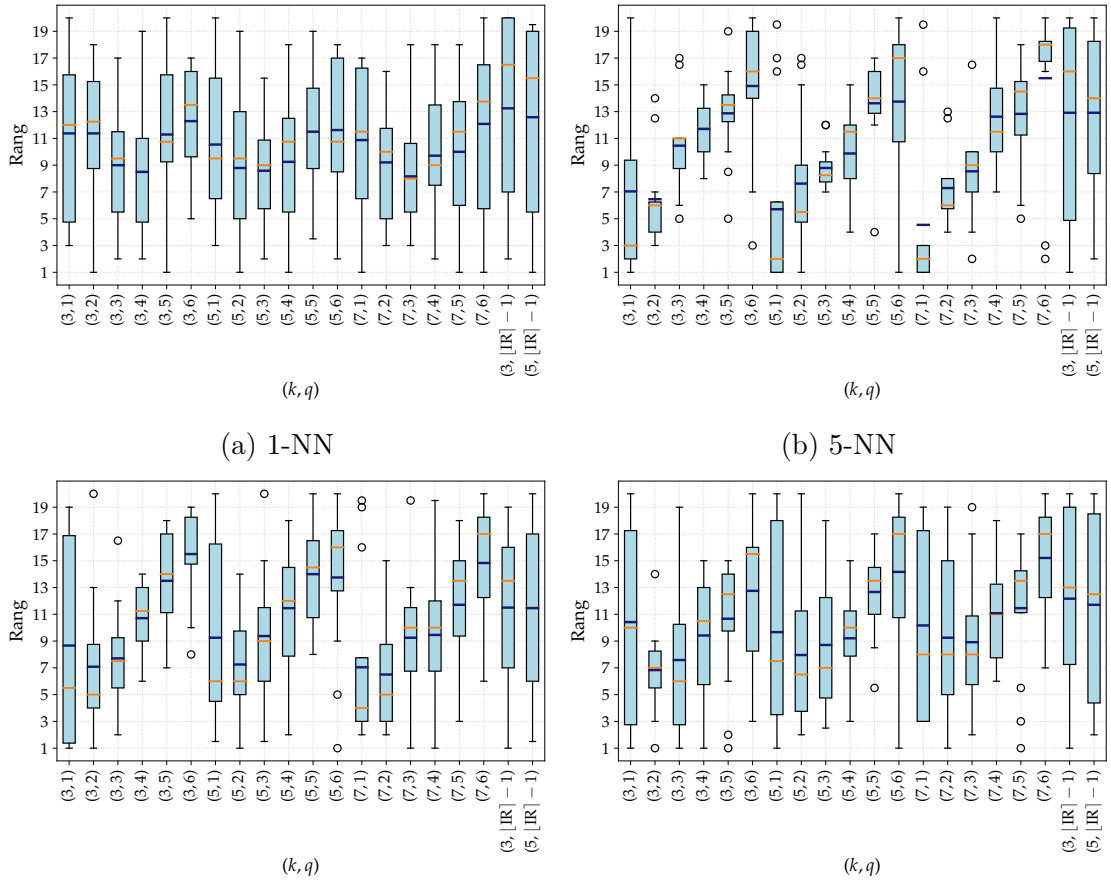
Prije same usporedbe različitih algoritama preuzorkovanja, provedeno je podešavanje vrijednosti njihovih parametara. U prvom dijelu analize, za algoritam SMOTE provedeno je traženje pogodnih parova (k, q) nad skupovima $\Omega_K = \{3, 5, 7\}$ i $\Omega_Q = \{1, 2, 3, 4, 5, 6\}$, odnosno nad njihovim Kartezijevim produktom $\Omega_K \times \Omega_Q$, što je uobičajeni način podešavanja

ovih parametara u literaturi [171, 175, 181, 201]. Učinak svake kombinacije parametara predstavljen je kvalitetom izvedbe klasifikacijskog modela treniranog na predobrađenom skupu podataka, iskazanom pomoću mjere F1. Ova mjera često se koristi u literaturi za iskazivanje uspješnosti izvedbe raznih pristupa za ublažavanje problema neuravnoteženosti klasa [91]. Odabir najbolje kombinacije parametara algoritma SMOTE izvršen je na razini korištenih klasifikatora, tako da su na svakom skupu podataka rangirani učinci ispitanih kombinacija parametara te je za svaku od njih izračunat prosječni rang. Kombinacije parametara s najboljim prosječnim rangom potom su korištene prilikom usporedbe izvornog algoritma s drugim algoritmima preuzorkovanja, a odabrane postavke prikazane su u tablici 4.2. U drugom dijelu analize provedeno je podešavanje parametra q razmatranih unaprijeđenih inačica algoritma SMOTE, pri čemu je veličina njihova susjedstva postavljena na $k = 5$ jer je ta vrijednost korištena u izvornim radovima u kojima su predloženi te je ujedno jedna od najčešćih postavki tog parametara u literaturi [69, 70, 169]. Stoga je za sve algoritme (osim RESMOTE koji nema parametar q) provedeno traženje pogodnih vrijednosti parametra q unutar skupa $\{1, 2, 3, 4, 5, 6\}$, na isti način kao i tijekom podešavanja parametara izvornog algoritma. Za svaku kombinaciju klasifikatora i algoritma preuzorkovanja odabrana je vrijednost parametra q s najboljim prosječnim rangom te je prikazana u tablici 4.3.

Nakon podešavanja parametara odabranih algoritama, provedena je usporedba njihovih performansi s performansama predloženog algoritma. Kako bi se sažeto prikazale performanse uspoređenih algoritama preuzorkovanja, za svaki skup podataka izračunate su prosječne izvedbe korištenih klasifikatora u smislu mjere F1. S ciljem pojednostavljenja njihove usporedbe, izvedene su Euklidske udaljenosti (označene s d_{perf}) između izvedbe savršenog klasifikatora i točke čije koordinate čine prosječne izvedbe korištenih klasifikatora na svakom skupu podataka, kao u poglavlju 3. Osim toga, usporedbom prosječnih izvedbi korištenih klasifikatora na svim skupovima podataka pomoću Friedmanova testa ranga [156], izračunati su i prosječni rangovi (označeni s FR) razmatranih algoritama preuzorkovanja koji ukazuju na općenitu razliku u njihovu doprinosu izvedbi klasifikatora.

4.4.3 Usporedba predloženog algoritma s algoritmom SMOTE

Kako bi se ispitala korisnost predloženog algoritama, provedena je usporedba s algoritmom SMOTE. Prije ove usporedbe, izvršeno je podešavanje parametara izvornog algoritma, kao što je ranije objašnjeno. Rangovi pojedinih kombinacija parametara ostvareni na svim skupovima podataka prikazani su sažeto na slici 4.5 pomoću dijagrama pravokutnika, uz oznaku medijana (narančastom bojom) i prosjeka (plavom bojom). Vidljivo je kako se niti jedna od ispitanih kombinacija parametara ne može proglasiti generalno najprikladnijom, što je i ranije istaknuto. Štoviše, rangovi većine kombinacija parametara protežu se od najboljeg do najlošijeg ovisno o skupu podataka, što upućuje na nužnost podešavanja parametara algoritma SMOTE za pojedini problem. Ipak, u literaturi se ovaj postupak vrlo rijetko provodi



(a) 1-NN (b) 5-NN

(c) SVM (d) MLP

Slika 4.5: Rangovi ostvareni podešavanjem parametara algoritma SMOTE

Tablica 4.4: Prosječni rangovi ostvareni podešavanjem parametara algoritma SMOTE

Klas.	$k=3$						$k=5$						$k=7$						$q=\lfloor \text{IR} \rfloor - 1$	
	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$	$q=6$	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$	$q=6$	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$	$q=6$	$k=3$	$k=5$
1-NN	11.38	11.38	9.00	8.50	11.29	12.29	10.54	8.79	8.58	9.25	11.50	11.63	10.88	9.21	**8.17**	9.71	10.00	12.08	13.25	12.58
5-NN	7.04	6.46	10.46	11.71	12.88	14.92	5.71	7.63	8.79	9.88	13.63	13.75	**4.54**	7.29	8.54	12.63	12.83	15.50	12.92	12.92
SVM	8.67	7.08	7.71	10.71	13.50	15.50	9.25	7.25	9.38	11.46	14.00	13.75	7.04	**6.50**	9.25	9.46	11.71	14.83	11.50	11.46
MLP	10.42	**6.83**	7.58	9.42	10.67	12.75	9.67	7.96	8.71	9.21	12.67	14.17	10.17	9.25	8.92	11.08	11.46	15.21	12.17	11.71

zbog svoje dugotrajnosti i složenosti, a performanse algoritama preuzorkovanja najčešće se uspoređuju za jedinstvene postavke parametara. U tablici 4.4 prikazani su prosječni rangovi razmatranih kombinacija parametara za svaki klasifikator, a najbolje rangirane kombinacije (podebljane u tablici) korištene su pri usporedbi algoritma SMOTE s predloženim algoritmom, uz najčešće postavke parametara u literaturi. Ostvarene performanse uspoređenih algoritama izražene su prosjekom te standardnom devijacijom kvalitete izvedbe korištenih klasifikatora te su prikazane u tablicama 4.5, 4.6, 4.7 i 4.8. Pri tome je kvaliteta izvedbe klasifikatora prije provedbe preuzorkovanja označena oznakom NO. Na dnu tablica prikazani su rangovi u smislu kvalitete te udaljenosti od savršenog klasifikatora. Pri tome su najbolje vrijednosti prikazanih rezultata podebljane za svaki skup podataka.

Na temelju prikazanih rangova i udaljenosti od savršenog klasifikatora, moguće je zaključiti da predloženi algoritam, cjelokupno gledano, više doprinosi izvedbi klasifikatora u

Tablica 4.5: Usporedba performansi algoritma SMOTE i predloženog algoritma za klasifikator 1-NN u smislu mjere F1

$\mathcal{D}$	NO	SMOTE ($k=5;\ q=1$)	SMOTE ($k=3;\ q=\lfloor \mathrm{IR} \rceil - 1$)	SMOTE ($k=5;\ q=\lfloor \mathrm{IR} \rceil - 1$)	SMOTE ($k=7;\ q=3$)	Predloženi
$\mathcal{D}_1$	0.86±0.04	0.87±0.04	0.86±0.04	0.86±0.04	**0.88**±0.04	**0.88**±0.04
$\mathcal{D}_2$	**0.93**±0.02	**0.93**±0.01	**0.93**±0.02	**0.93**±0.02	0.93±0.02	0.92±0.02
$\mathcal{D}_3$	0.84±0.04	0.87±0.04	0.87±0.04	0.87±0.03	**0.89**±0.03	**0.89**±0.04
$\mathcal{D}_4$	0.57±0.03	0.57±0.02	**0.58**±0.03	**0.58**±0.02	0.57±0.03	0.57±0.02
$\mathcal{D}_5$	0.97±0.03	0.97±0.03	0.97±0.03	0.97±0.03	0.97±0.03	0.97±0.03
$\mathcal{D}_6$	0.69±0.14	0.69±0.14	0.69±0.10	0.67±0.10	**0.70**±0.12	0.68±0.13
$\mathcal{D}_7$	0.73±0.06	0.75±0.07	**0.78**±0.05	**0.78**±0.06	0.77±0.05	0.73±0.06
$\mathcal{D}_8$	0.61±0.05	**0.64**±0.05	0.59±0.03	0.60±0.03	0.62±0.04	0.63±0.05
$\mathcal{D}_9$	0.82±0.09	0.82±0.09	0.83±0.07	0.83±0.06	0.83±0.05	0.83±0.07
$\mathcal{D}_{10}$	**0.85**±0.20	0.81±0.19	0.79±0.17	0.79±0.17	0.81±0.17	0.83±0.18
$\mathcal{D}_{11}$	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00
$\mathcal{D}_{12}$	**0.73**±0.06	**0.73**±0.05	0.64±0.03	0.65±0.03	**0.73**±0.05	**0.73**±0.06
FR	3.92	3.42	3.79	3.88	**2.67**	3.33
d_{perf}	0.83	0.81	0.86	0.86	**0.80**	0.81

Tablica 4.6: Usporedba performansi algoritma SMOTE i predloženog algoritma za klasifikator 5-NN u smislu mjere F1

$\mathcal{D}$	NO	SMOTE ($k=5;\ q=1$)	SMOTE ($k=3;\ q=\lfloor \mathrm{IR} \rceil - 1$)	SMOTE ($k=5;\ q=\lfloor \mathrm{IR} \rceil - 1$)	SMOTE ($k=7;\ q=1$)	Predloženi
$\mathcal{D}_1$	0.81±0.05	**0.83**±0.04	0.81±0.05	0.81±0.05	**0.83**±0.05	**0.83** ±0.05
$\mathcal{D}_2$	**0.93**±0.02	**0.93**±0.02	**0.93**±0.02	**0.93**±0.02	**0.93**±0.02	0.92±0.02
$\mathcal{D}_3$	0.82±0.04	0.88±0.04	0.87±0.04	0.88±0.04	0.88±0.04	**0.89**±0.04
$\mathcal{D}_4$	0.62±0.03	**0.64**±0.03	0.63±0.03	0.62±0.03	**0.64**±0.03	**0.64**±0.03
$\mathcal{D}_5$	0.93±0.06	0.96±0.03	**0.97**±0.03	**0.97**±0.03	**0.97**±0.04	**0.97**±0.03
$\mathcal{D}_6$	0.71±0.12	**0.73**±0.12	0.68±0.11	0.69±0.11	**0.73**±0.11	0.72±0.12
$\mathcal{D}_7$	0.78±0.06	0.80±0.06	0.76±0.04	0.77±0.05	**0.81**±0.06	0.80±0.05
$\mathcal{D}_8$	0.65±0.06	**0.68**±0.05	0.57±0.03	0.58±0.03	**0.68**±0.05	**0.68**±0.05
$\mathcal{D}_9$	**0.86**±0.07	0.84±0.07	0.80±0.06	0.80±0.05	0.84±0.08	0.85±0.07
$\mathcal{D}_{10}$	0.65±0.23	**0.81**±0.19	0.75±0.15	0.75±0.16	0.80±0.19	**0.82**±0.19
$\mathcal{D}_{11}$	0.95±0.06	0.96±0.06	**0.99**±0.02	**0.99**±0.03	0.96±0.06	0.97±0.06
$\mathcal{D}_{12}$	**0.78**±0.06	**0.78**±0.06	0.64±0.03	0.64±0.02	0.77±0.05	0.77±0.05
FR	4.33	2.71	4.54	4.25	2.63	**2.54**
d_{perf}	0.82	0.72	0.88	0.87	**0.71**	**0.71**

odnosu na algoritam SMOTE, gotovo neovisno o korištenoj kombinaciji parametara. Štoviše, predloženi algoritam uvijek nadmašuje izvedbu algoritma SMOTE primijenjenog s najčešćim postavkama parametara iz literature, dok je izvorni algoritam s podešenim postavkama parametara bolje rangiran od predloženog algoritma jedino za klasifikator 1-NN. S obzirom na to da su razlike u njihovim rangovima općenito neznatne, valja podsjetiti da predloženi algoritam ne zahtijeva podešavanje parametara koje je pak provedeno za SMOTE. Osim toga, nije za očekivati značajne razlike u njihovim performansama jer ne postoji univerzalno najprikladniji algoritam preuzorkovanja za sve skupove podataka, kao što sugeriraju ranije spomenuta istraživanja [158, 193]. Ipak, predloženi algoritam može se smatrati korisnim unaprjeđenjem algoritma SMOTE jer iznimno pojednostavljuje njegovo korištenje, ostvarujući pri tome jednake ili bolje prosječne performanse. Kako bi se dobio detaljniji uvid u izvedbu predloženog algoritma pored njegovih prosječnih performansi, na slici 4.6 su po-

Tablica 4.7: Usporedba performansi algoritma SMOTE i predloženog algoritma za klasifikator SVM u smislu mjere F1

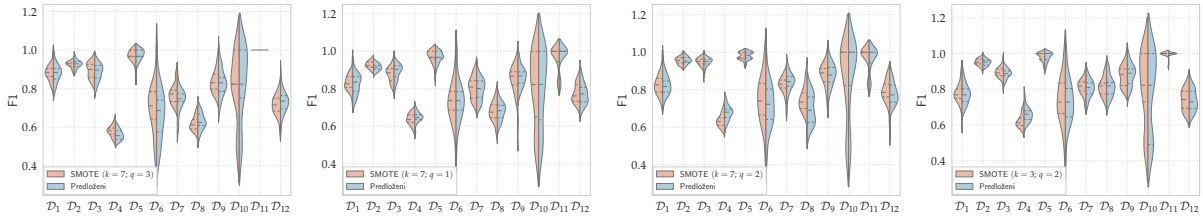
$\mathcal{D}$	NO	SMOTE ($k=5;\ q=1$)	SMOTE ($k=3;\ q=\lfloor \mathrm{IR} \rceil - 1$)	SMOTE ($k=5;\ q=\lfloor \mathrm{IR} \rceil - 1$)	SMOTE ($k=7;\ q=2$)	Predloženi
$\mathcal{D}_1$	**0.84**±0.03	**0.84**±0.06	**0.84**±0.03	**0.84**±0.03	0.83±0.05	0.83±0.05
$\mathcal{D}_2$	0.95±0.02	0.95±0.02	0.95±0.02	0.95±0.02	0.95±0.02	**0.96**±0.02
$\mathcal{D}_3$	0.93±0.03	**0.95**±0.02	**0.95**±0.02	0.94±0.03	**0.95**±0.02	**0.95**±0.02
$\mathcal{D}_4$	0.50±0.03	0.67±0.04	0.63±0.03	0.63±0.03	0.64±0.03	**0.68**±0.04
$\mathcal{D}_5$	0.95±0.04	**0.98**±0.02	**0.98**±0.02	0.97±0.02	**0.98**±0.02	**0.98**±0.02
$\mathcal{D}_6$	0.60±0.14	0.73±0.14	0.73±0.13	0.72±0.13	**0.75**±0.13	0.71±0.12
$\mathcal{D}_7$	0.75±0.07	**0.84**±0.05	0.77±0.04	0.77±0.05	0.83±0.05	**0.84**±0.05
$\mathcal{D}_8$	0.56±0.07	0.72±0.07	**0.75**±0.06	**0.75**±0.06	0.74±0.07	0.71±0.08
$\mathcal{D}_9$	**0.88**±0.06	**0.88**±0.06	0.83±0.06	0.82±0.06	**0.88**±0.06	**0.88**±0.06
$\mathcal{D}_{10}$	0.51±0.09	0.86±0.20	0.85±0.20	0.85±0.20	0.85±0.20	**0.88**±0.20
$\mathcal{D}_{11}$	0.95±0.06	0.95±0.06	**0.98**±0.03	**0.98**±0.03	0.97±0.06	0.97±0.06
$\mathcal{D}_{12}$	0.49±0.00	0.71±0.08	0.66±0.03	0.66±0.03	**0.79**±0.06	0.78±0.06
FR	5.21	2.88	3.29	3.92	2.96	**2.75**
d_{perf}	1.10	0.66	0.72	0.73	**0.64**	**0.64**

Tablica 4.8: Usporedba performansi algoritma SMOTE i predloženog algoritma za klasifikator MLP u smislu mjere F1

$\mathcal{D}$	NO	SMOTE ($k=5;\ q=1$)	SMOTE ($k=3;\ q=\lfloor \mathrm{IR} \rceil - 1$)	SMOTE ($k=5;\ q=\lfloor \mathrm{IR} \rceil - 1$)	SMOTE ($k=3;\ q=2$)	Predloženi
$\mathcal{D}_1$	0.80±0.06	0.78±0.05	**0.81**±0.05	0.80±0.05	0.78±0.06	0.78±0.05
$\mathcal{D}_2$	0.95±0.02	0.95±0.02	0.95±0.02	0.95±0.02	0.95±0.02	0.95±0.02
$\mathcal{D}_3$	0.89±0.04	**0.90**±0.03	**0.90**±0.03	**0.90**±0.03	0.89±0.03	0.89±0.03
$\mathcal{D}_4$	0.55±0.03	**0.67**±0.04	0.63±0.04	0.62±0.04	0.62±0.03	0.66±0.04
$\mathcal{D}_5$	0.78±0.07	0.87±0.07	**0.98**±0.02	**0.98**±0.02	**0.98**±0.02	**0.98**±0.02
$\mathcal{D}_6$	0.71±0.14	0.72±0.14	0.73±0.12	0.73±0.15	**0.74**±0.13	0.73±0.13
$\mathcal{D}_7$	0.49±0.05	0.78±0.07	0.77±0.05	0.77±0.05	**0.81**±0.04	**0.81**±0.05
$\mathcal{D}_8$	0.63±0.10	**0.82**±0.06	0.78±0.04	0.79±0.05	0.81±0.05	0.81±0.06
$\mathcal{D}_9$	0.88±0.05	**0.89**±0.05	0.84±0.06	0.84±0.06	0.87±0.06	0.88±0.06
$\mathcal{D}_{10}$	0.63±0.22	0.77±0.22	0.77±0.19	0.78±0.19	**0.81**±0.20	0.78±0.22
$\mathcal{D}_{11}$	0.98±0.05	0.99±0.04	0.96±0.03	0.96±0.04	**1.00**±0.02	**1.00**±0.01
$\mathcal{D}_{12}$	0.51±0.03	0.72±0.06	0.64±0.03	0.63±0.03	**0.74**±0.06	**0.74**±0.06
FR	4.96	3.08	3.67	3.75	2.79	**2.75**
d_{perf}	1.09	0.70	0.76	0.76	0.68	**0.67**

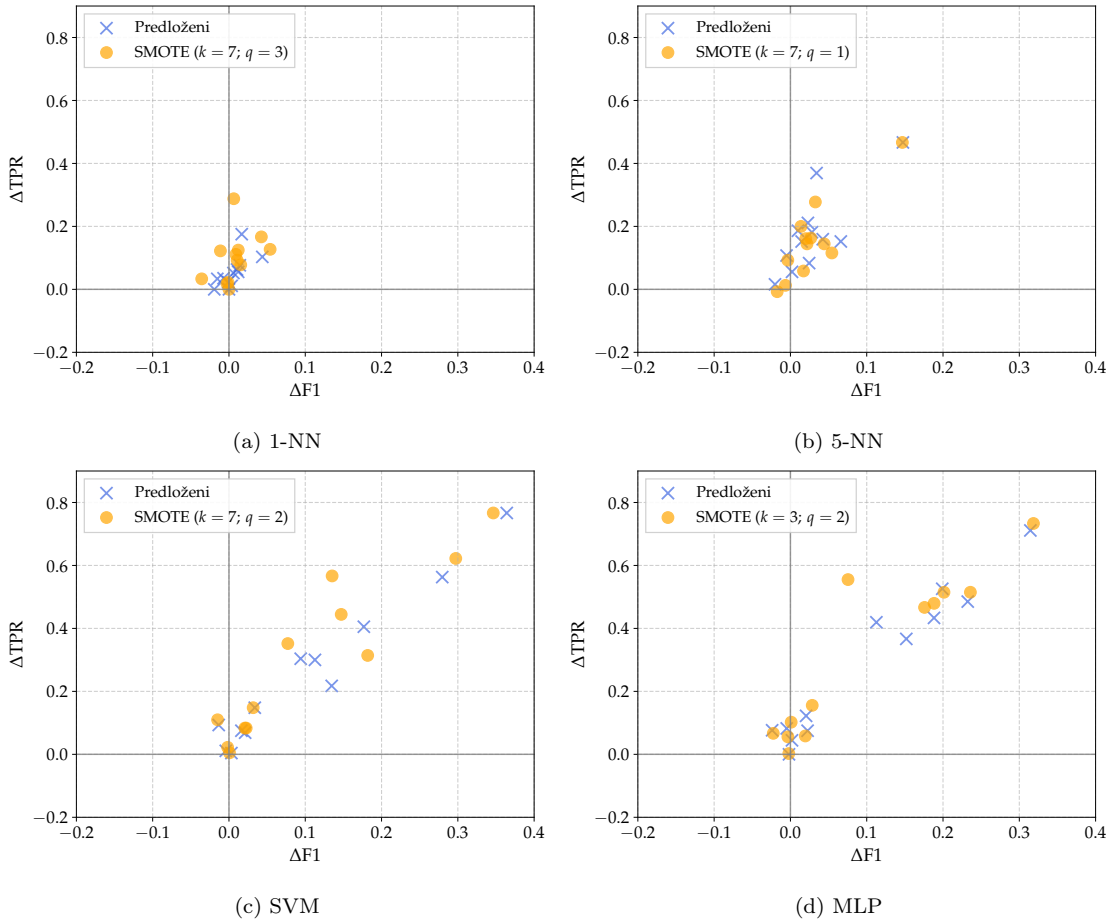
moću violinskih dijagrama prikazane distribucije kvalitete izvedbe klasifikatora za svih 30 podjela skupova podataka nakon njihova preuzorkovanja predloženim algoritmom, odnosno algoritmom SMOTE s podešenim postavkama parametara. Unutar ovih dijagrama označene su vrijednosti prvog i trećeg kvartila te medijana. Položaji i vrhovi distribucija, kao i oznake kvartila i medijana, sugeriraju da ne postoji primjetna razlika u performansama uspoređenih algoritama. Stoga je moguće zaključiti da je način rada predloženog algoritma robustan na različite podjele skupova podataka te da unatoč izostanku parametara pokazuje stabilnost u smislu kvalitete svoje izvedbe.

Kao što je ranije navedeno, postupak preuzorkovanja jedan je od najzastupljenijih pristupa u literaturi za ublažavanje problema neuravnoteženosti klasa jer proširuje koncept manjinske klase i time pospješuje njezino prepoznavanje. Rezultati u tablicama 4.5, 4.6, 4.7 i 4.8 sugeriraju da također doprinosi poboljšanju opće izvedbe klasifikatora jer predobrada



(a) 1-NN (b) 5-NN (c) SVM (d) MLP

Slika 4.6: Distribucije kvalitete izvedbe klasifikatora nakon preuzorkovanja



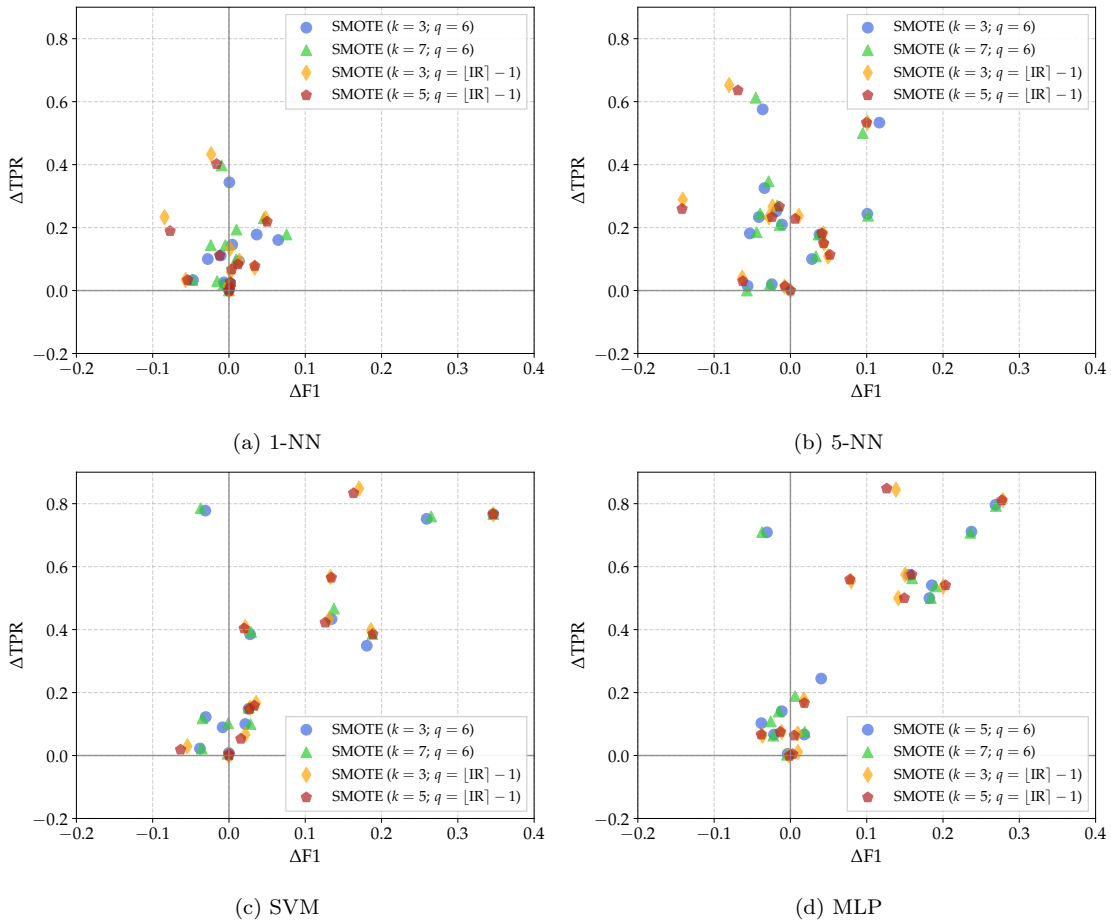
(a) 1-NN (b) 5-NN

(c) SVM (d) MLP

Slika 4.7: Razlike u vrijednostima mjera F1 i TPR ostvarenim nakon i prije provedbe preuzorkovanja

skupa podataka uspoređenim algoritmima poboljšava iznos mjere F1 na većini promatranih problema. Poboljšanje ove mjere prvenstveno proizlazi iz poboljšanja uspješnosti predviđanja manjinske klase, na što ukazuju ovisnosti razlika mjera F1 i TPR prije i nakon provedbe preuzorkovanja (ΔF1 i ΔTPR) koje su prikazane na slici 4.9 za predloženi algoritam te algoritam SMOTE s podešenim postavkama parametara. Iz slike je jasna proporcionalnost između navedenih razlika, posebice za klasifikatore SVM i MLP. Na većini skupova podataka, oba algoritma doprinose poboljšanju opće izvedbe klasifikatora te poboljšanju njegove sposobnosti prepoznavanja manjinske klase.

Performanse predloženog algoritma slične su performansama algoritma SMOTE koji je



(a) 1-NN

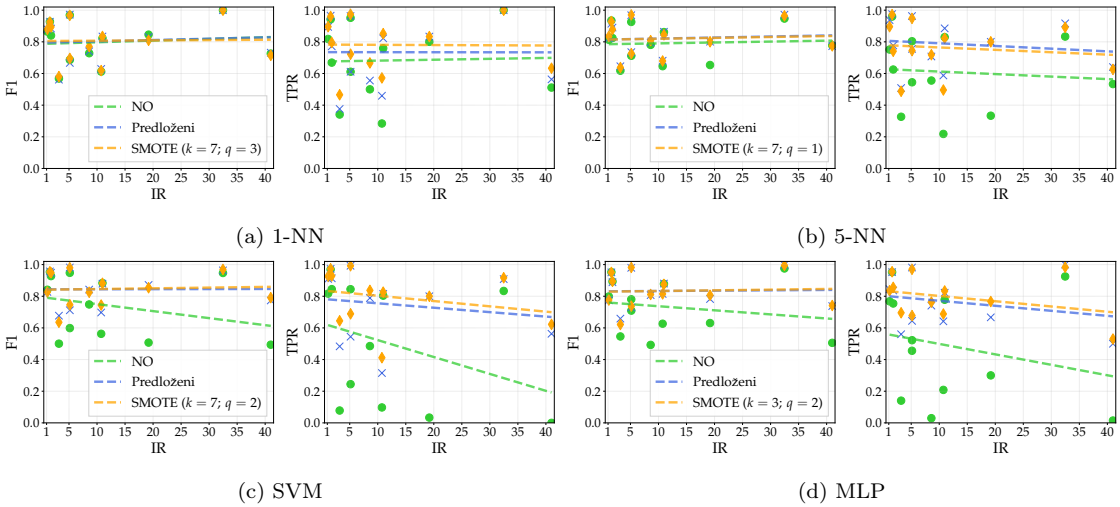
(b) 5-NN

(c) SVM

(d) MLP

Slika 4.8: Razlike u vrijednostima mjera F1 i TPR ostvarenim nakon i prije provedbe algoritma SMOTE s raznim postavkama parametara

primijenjen s prosječno najkvalitetnijim ispitanim kombinacijama parametara. Međutim, iako SMOTE u pravilu poboljšava prepoznavanje manjinske klase, kvaliteta njegove izvedbe uvelike ovisi o korištenim postavkama parametara. Na slici 4.8 prikazane su ovisnosti razlika u vrijednostima mjera F1 i TPR kada se za preuzorkovanje koristi algoritam SMOTE s prosječno najgore rangiranim kombinacijama parametara. Za ove postavke parametara, provedba algoritma SMOTE na većem broju skupova podataka rezultira narušavanjem opće izvedbe klasifikatora. Razlog tomu je što ovaj algoritam može uzrokovati povećanje složenosti skupa podataka zbog svojih nedostataka koji su izraženiji pri neprikladnim postavkama parametara. Štoviše, dvije kombinacije parametara $(k, q) \in \{(3, \lfloor \text{IR} \rceil - 1), (5, \lfloor \text{IR} \rceil - 1)\}$ prikazane na slici 4.8 jedne su od najzastupljenijih u literaturi te se njima nastoji postići posve uravnotežena raspodjela primjeraka različitih klasa u skupu podataka. Iako intuitivno ovakav ishod preuzorkovanja može biti najsmisleniji, očito je da nema najpogodniji učinak na uspješnost klasifikatora. Podešavanje parametara algoritma SMOTE stoga je neophodan korak pri korištenju ovog algoritma, dok je pri korištenju predloženog algoritma ono nepotrebno s obzirom na mogućnost prilagođavanja njegova načina rada unutarnjim karakteristikama skupa podataka.



(a) 1-NN (b) 5-NN

(c) SVM (d) MLP

Slika 4.9: Vrijednosti mjera F1 i TPR ostvarene prije i nakon provedbe preuzorkovanja

Težina problema neuravnoteženosti klasa prvenstveno se izražava pomoću omjera neuravnoteženosti skupa podataka te je primarni zadatak postupka preuzorkovanja smanjiti ga stvaranjem novih sintetičkih primjeraka. Kako bi se dobio jasniji uvid u učinkovitost uspoređenih algoritama ovisno o stupnju neuravnoteženosti klasa, na slici 4.9 prikazani su iznosi mjera F1 i TPR prije i nakon provedbe preuzorkovanja ovisno o omjeru neuravnoteženosti skupa podataka. Također su ilustrirani pravci koji najbolje opisuju ovisnost ovih mjera o omjeru neuravnoteženosti, pri čemu su parametri pravaca utvrđeni metodom najmanjih kvadrata. Može se primijetiti da korisnost preuzorkovanja raste s povećanjem omjera neuravnoteženosti, s obzirom na to da je poboljšanje mjera F1 i TPR uglavnom najizraženije na iznimno neuravnoteženim skupovima podataka. Međutim, doseg pozitivnog učinka preuzorkovanja također varira ovisno o korištenom klasifikatoru te se može smatrati važnim postupkom predobrade skupa podataka prije upotrebe klasifikatora SVM i MLP. Pri tome, nije moguće uočiti značajne razlike u utjecaju stupnja neuravnoteženosti klasa na izvedbe uspoređenih algoritama, što daje naslutiti da je predloženi algoritam jednako robustan na ovu karakteristiku skupa podataka kao i algoritam SMOTE s podešenim postavkama parametara.

4.4.4 Usporedba predloženog algoritma s unaprijeđenim inačicama algoritma SMOTE

Predloženi algoritam također je uspoređen s nekolicinom unaprjeđenja algoritma SMOTE iz literature, kako bi se stekao uvid u njihov doprinos poboljšanju izvornog algoritma. Prije njihove usporedbe, provedeno je podešavanje parametara odabranih algoritama, kao što je ranije opisano. Ostvarene performanse uspoređenih algoritama izražene su prosjekom te standardnom devijacijom kvalitete izvedbe korištenih klasifikatora (izraženom pomoću mjere F1) te su prikazane u tablicama 4.9, 4.10, 4.11 i 4.12. Uz svaki od odabranih algoritama,

Tablica 4.9: Usporedba performansi unaprijeđenih inačica algoritma SMOTE i predloženog algoritma za klasifikator 1-NN u smislu mjere F1

$\mathcal{D}$	NO	RO ($q = 3$)	SMOTE ($k = 7$; $q = 3$)	BSMOTE ($k = 5$; $q = 3$)	RASMOTE ($q = 3$)	WSMOTE ($k = 5$; $q = 2$)	RESMOTE ($k = 5$)	Predloženi
$\mathcal{D}_1$	0.86±0.04	0.86±0.04	**0.88**±0.04	0.86±0.04	0.87±0.04	**0.88**±0.05	0.87±0.04	**0.88**±0.04
$\mathcal{D}_2$	**0.93**±0.02	**0.93**±0.01	**0.93**±0.02	**0.93**±0.02	0.92±0.02	**0.93**±0.02	**0.93**±0.02	0.92±0.02
$\mathcal{D}_3$	0.84±0.04	0.84±0.04	0.89±0.04	0.88±0.04	0.89±0.04	0.89±0.04	0.84±0.04	**0.90**±0.04
$\mathcal{D}_4$	0.57±0.03	0.56±0.02	**0.58**±0.03	**0.58**±0.03	**0.58**±0.03	**0.58**±0.03	0.57±0.02	0.57±0.02
$\mathcal{D}_5$	0.97±0.03	0.97±0.03	0.97±0.03	0.97±0.03	0.97±0.03	0.97±0.03	0.97±0.03	0.97±0.03
$\mathcal{D}_6$	0.69±0.14	0.69±0.14	**0.70**±0.12	**0.70**±0.12	0.69±0.12	**0.70**±0.12	0.69±0.14	0.68±0.13
$\mathcal{D}_7$	0.73±0.06	0.73±0.06	**0.77**±0.05	0.74±0.05	**0.77**±0.05	0.76±0.06	0.73±0.06	0.73±0.06
$\mathcal{D}_8$	0.61±0.05	0.61±0.05	0.62±0.04	**0.63**±0.04	0.62±0.04	**0.63**±0.04	0.61±0.05	**0.63**±0.05
$\mathcal{D}_9$	0.82±0.09	**0.83**±0.06	**0.83**±0.05	0.77±0.11	0.80±0.11	**0.83**±0.07	0.82±0.09	**0.83**±0.07
$\mathcal{D}_{10}$	0.85±0.20	**0.86**±0.18	0.81±0.17	0.82±0.18	0.82±0.18	0.82±0.18	0.85±0.20	0.83±0.18
$\mathcal{D}_{11}$	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00
$\mathcal{D}_{12}$	**0.73**±0.06	**0.73**±0.06	0.71±0.05	**0.73**±0.06	0.71±0.05	**0.73**±0.05	**0.73**±0.06	**0.73**±0.06
FR	5.25	5.05	3.79	4.38	4.83	**3.21**	5.04	4.46
d_{perf}	0.83	0.83	**0.80**	0.81	0.81	**0.80**	0.83	0.81

Tablica 4.10: Usporedba performansi unaprijeđenih inačica algoritma SMOTE i predloženog algoritma za klasifikator 5-NN u smislu mjere F1

$\mathcal{D}$	NO	RO ($q = 2$)	SMOTE ($k = 7$; $q = 1$)	BSMOTE ($k = 5$; $q = 1$)	RASMOTE ($q = 1$)	WSMOTE ($k = 5$; $q = 1$)	RESMOTE ($k = 5$)	Predloženi
$\mathcal{D}_1$	0.81±0.05	**0.83**±0.05	**0.83**±0.05	**0.83**±0.05	**0.83**±0.06	**0.83**±0.04	**0.83**±0.05	**0.83**±0.05
$\mathcal{D}_2$	**0.93**±0.02	0.92±0.02	**0.93**±0.02	0.92±0.02	0.91±0.02	**0.93**±0.02	0.92±0.02	0.92±0.02
$\mathcal{D}_3$	0.82±0.04	0.88±0.04	0.88±0.04	0.88±0.04	0.88±0.04	0.88±0.04	0.85±0.04	**0.89**±0.04
$\mathcal{D}_4$	0.62±0.03	0.60±0.03	**0.64**±0.03	0.63±0.03	**0.64**±0.03	**0.64**±0.03	0.62±0.04	**0.64**±0.03
$\mathcal{D}_5$	0.93±0.06	**0.97**±0.03	**0.97**±0.04	0.96±0.03	0.96±0.04	0.96±0.04	0.93±0.06	**0.97**±0.03
$\mathcal{D}_6$	0.71±0.12	0.70±0.10	**0.73**±0.11	**0.73**±0.11	0.72±0.11	**0.73**±0.10	0.71±0.12	0.72±0.12
$\mathcal{D}_7$	0.78±0.06	0.76±0.06	**0.81**±0.06	0.79±0.05	0.80±0.05	0.80±0.06	0.78±0.06	0.80±0.05
$\mathcal{D}_8$	0.65±0.06	0.64±0.05	0.68±0.05	0.68±0.05	**0.70**±0.04	0.69±0.05	0.65±0.06	0.68±0.05
$\mathcal{D}_9$	**0.86**±0.07	0.81±0.06	0.84±0.08	0.79±0.08	0.82±0.08	0.84±0.08	**0.86**±0.07	**0.86**±0.07
$\mathcal{D}_{10}$	0.65±0.23	**0.82**±0.19	0.80±0.19	0.81±0.20	0.79±0.19	0.81±0.19	0.65±0.23	0.82±0.19
$\mathcal{D}_{11}$	0.95±0.06	**0.99**±0.04	0.96±0.06	0.97±0.06	0.96±0.06	0.96±0.06	0.95±0.06	0.97±0.06
$\mathcal{D}_{12}$	**0.78**±0.06	0.74±0.05	0.77±0.05	0.76±0.05	**0.78**±0.05	0.77±0.05	**0.78**±0.06	0.77±0.05
FR	5.83	5.42	3.42	4.63	4.25	3.54	5.71	**3.21**
d_{perf}	0.82	0.79	**0.71**	0.73	0.72	0.72	0.81	**0.71**

navedene su i upotrijebljene vrijednosti njegovih parametara.

S obzirom na izvedene rangove i udaljenosti od savršenog klasifikatora, moguće je zaključiti da predloženi algoritam općenito više doprinosi poboljšanju izvedbe klasifikatora u odnosu na ostale razmatrane unaprijeđene inačice algoritma SMOTE. Pri tome, razlike u ostvarenim rangovima predloženog algoritma te algoritama WSMOTE, SMOTE i RASMOTE su neznatne, dok algoritmi RESMOTE, RO i BSMOTE manje doprinose poboljšanju izvedbe klasifikatora. S obzirom na njegovu jednostavnost upotrebe i generalno najbolje performanse, predloženi algoritam može se smatrati najkorisnijom unaprijeđenom inačicom algoritma SMOTE od razmatranih. Uz predloženo unaprjeđenje, algoritam WSMOTE jedini nadmašuje izvedbu izvornog algoritma za većinu korištenih klasifikatora. Povoljne performanse ovog algoritma vjerojatno su posljedica njegova načina izbjegavanja jednakomjernog preuzorkovanja stvaranjem većine sintetičkih primjeraka u području najveće gustoće ma-

Tablica 4.11: Usporedba performansi unaprijeđenih inačica algoritma SMOTE i predloženog algoritma za klasifikator SVM u smislu mjere F1

$\mathcal{D}$	NO	RO ($q = 3$)	SMOTE ($k = 7$; $q = 2$)	BSMOTE ($k = 5$; $q = 2$)	RASMOTE ($q = 3$)	WSMOTE ($k = 5$; $q = 1$)	RESMOTE ($k = 5$)	Predloženi
$\mathcal{D}_1$	0.84±0.03	0.83±0.06	0.83±0.05	0.83±0.06	0.84±0.06	0.84±0.05	**0.85**±0.05	0.83±0.05
$\mathcal{D}_2$	0.95±0.02	0.95±0.02	0.95±0.02	0.95±0.02	**0.96**±0.02	0.95±0.02	0.95±0.02	**0.96**±0.02
$\mathcal{D}_3$	0.93±0.03	**0.95**±0.02	**0.95**±0.02	0.94±0.03	0.92±0.03	**0.95**±0.02	0.93±0.03	**0.95**±0.02
$\mathcal{D}_4$	0.50±0.03	0.60±0.03	0.64±0.03	0.63±0.05	0.61±0.03	0.67±0.04	0.55±0.05	**0.68**±0.04
$\mathcal{D}_5$	0.95±0.04	**0.98**±0.02	**0.98**±0.02	0.97±0.02	**0.98**±0.02	**0.98**±0.02	0.95±0.04	**0.98**±0.02
$\mathcal{D}_6$	0.60±0.14	0.71±0.12	**0.75**±0.13	0.72±0.14	0.72±0.13	0.71±0.12	0.61±0.14	0.71±0.12
$\mathcal{D}_7$	0.75±0.07	0.80±0.05	0.83±0.05	0.80±0.05	0.81±0.05	**0.84**±0.05	0.75±0.07	**0.84**±0.05
$\mathcal{D}_8$	0.56±0.07	**0.76**±0.06	0.74±0.07	0.73±0.08	0.72±0.07	0.73±0.06	0.56±0.07	0.71±0.08
$\mathcal{D}_9$	**0.88**±0.06	0.86±0.06	**0.88**±0.06	0.83±0.06	0.87±0.06	**0.88**±0.06	**0.88**±0.06	**0.88**±0.06
$\mathcal{D}_{10}$	0.51±0.09	0.86±0.20	0.85±0.20	0.86±0.20	0.86±0.20	0.87±0.20	0.51±0.09	**0.88**±0.20
$\mathcal{D}_{11}$	0.95±0.06	**0.98**±0.04	0.97±0.06	0.96±0.06	**0.98**±0.05	0.97±0.06	0.95±0.06	0.97±0.06
$\mathcal{D}_{12}$	0.49±0.00	0.78±0.05	0.79±0.06	**0.80**±0.05	0.79±0.04	0.71±0.07	0.49±0.00	0.78±0.06
FR	6.58	4.33	3.50	4.79	3.83	3.42	6.25	**3.29**
d_{perf}	1.10	0.68	**0.64**	0.68	0.68	0.66	1.07	**0.64**

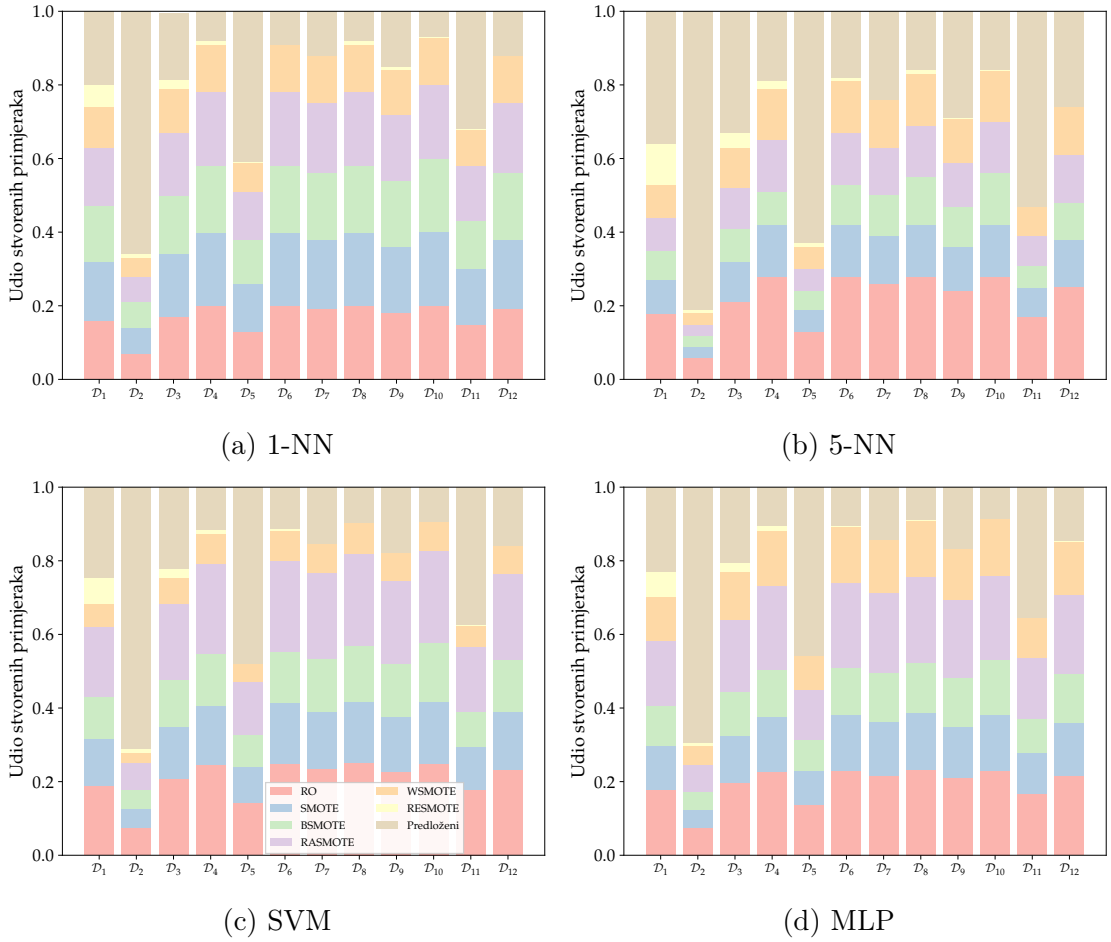
Tablica 4.12: Usporedba performansi unaprijeđenih inačica algoritma SMOTE i predloženog algoritma za klasifikator MLP u smislu mjere F1

$\mathcal{D}$	NO	RO ($q = 3$)	SMOTE ($k = 3$; $q = 2$)	BSMOTE ($k = 5$; $q = 2$)	RASMOTE ($q = 3$)	WSMOTE ($k = 5$; $q = 2$)	RESMOTE ($k = 5$)	Predloženi
$\mathcal{D}_1$	**0.80**±0.06	0.78±0.06	0.78±0.06	0.78±0.06	0.78±0.06	0.78±0.06	0.78±0.08	0.78±0.05
$\mathcal{D}_2$	0.95±0.02	0.95±0.02	0.95±0.02	0.95±0.02	0.95±0.02	0.95±0.02	0.95±0.02	0.95±0.02
$\mathcal{D}_3$	0.89±0.04	**0.91**±0.03	0.89±0.03	0.89±0.03	0.90±0.03	0.89±0.03	0.90±0.04	0.89±0.03
$\mathcal{D}_4$	0.55±0.03	0.59±0.03	0.62±0.03	0.63±0.04	0.62±0.03	0.62±0.03	0.55±0.04	**0.66**±0.04
$\mathcal{D}_5$	0.78±0.07	0.98±0.02	0.98±0.02	0.98±0.02	**0.99**±0.02	0.98±0.02	0.78±0.08	0.98±0.02
$\mathcal{D}_6$	0.71±0.14	0.72±0.12	**0.74**±0.13	**0.74**±0.12	0.73±0.12	0.73±0.11	0.72±0.14	0.73±0.13
$\mathcal{D}_7$	0.49±0.05	0.78±0.05	**0.81**±0.04	0.79±0.06	0.78±0.05	**0.81**±0.05	0.51±0.05	**0.81**±0.05
$\mathcal{D}_8$	0.63±0.10	0.80±0.05	0.81±0.05	0.81±0.06	0.80±0.05	**0.82**±0.05	0.62±0.08	0.81±0.06
$\mathcal{D}_9$	**0.88**±0.05	0.86±0.06	0.87±0.06	0.82±0.06	0.86±0.05	**0.88**±0.05	**0.88**±0.05	**0.88**±0.06
$\mathcal{D}_{10}$	0.63±0.22	**0.84**±0.19	0.81±0.20	0.77±0.21	0.83±0.19	0.80±0.21	0.62±0.22	0.78±0.22
$\mathcal{D}_{11}$	0.98±0.05	**1.00**±0.00	**1.00**±0.02	0.98±0.05	**1.00**±0.01	0.98±0.05	0.97±0.06	**1.00**±0.01
$\mathcal{D}_{12}$	0.51±0.03	**0.74**±0.05	**0.74**±0.06	**0.74**±0.06	**0.74**±0.05	**0.74**±0.06	0.51±0.04	**0.74**±0.06
FR	6.04	4.29	3.67	4.46	3.88	3.88	6.21	**3.58**
d_{perf}	1.09	0.71	0.68	0.71	0.69	0.68	1.08	**0.67**

njinskih primjeraka jer sve ostale korake preuzima iz izvornog algoritma. Međutim, također zadržava i oba njegova parametra čije vrijednosti imaju značajan utjecaj na kvalitetu izvedbe algoritma. S druge strane, RASMOTE jedini je algoritam (uz predloženi) koji pojednostavljuje upotrebu izvornog algoritma uklanjanjem jednog od njegovih parametra, a zadržava njegove performanse na velikom broju skupova podataka. Ipak, prije njegova provođenja potrebno je definirati broj sintetičkih primjeraka koje će ovaj algoritam stvoriti, dok predloženi algoritam samostalno određuje njihov broj te postiže bolje performanse od njega na većini skupova podataka. Nadalje, algoritmi RESMOTE i BSMOTE ostvaruju znatno lošije performanse od izvornog algoritma, unatoč tome što su to jedina razmatrana unaprjeđenja (uz predloženo) koja uzimaju u obzir većinsku klasu pri određivanju susjedstva manjinskih primjeraka. Zanimljivo je primijetiti da su ovi algoritmi suprostavljeni prema strategiji preuzorkovanja koju provode jer BSMOTE preuzorkuje samo one primjerke čije susjedstvo

većinom čine većinski primjerci, dok RESMOTE upravo takve primjerke ne preuzorkuje. Konačno, iako je zbog svoje trivijalne složenosti te jednostavnosti upotrebe jedan od najčešće korištenih postupaka preuzorkovanja u literaturi, algoritam RO ne uspijeva konkurirati izvornom algoritmu ni njegovim unaprijeđenim inačicama po ostvarenim performansama. Štoviše, prikazani rezultati sugeriraju da ovaj algoritam ima zanemariv učinak na izvedbu klasifikatora 1-NN, a pokazao se i neprikladnim za klasifikator DT [202]. Stoga se ne može preporučiti kao valjan pristup ublažavanju neuravnoteženosti klasa.

Kvaliteta izvedbe algoritma preuzorkovanja uvelike ovisi o broju sintetičkih primjeraka koje on stvara [194]. Kod većine korištenih algoritama moguće je utjecati na broj stvorenih primjeraka podešavanjem odgovarajućeg parametra, što je i učinjeno prije njihove usporedbe. Predloženi i RESMOTE jedini su od razmatranih algoritama koji samostalno određuju broj stvorenih primjeraka na temelju sadržaja susjedstva manjinskih primjeraka. Kako bi se stekao dublji uvid u njihov učinak, na slici 4.10 su pomoću naslaganih stupčastih dijagrama za svaki skup podataka prikazani brojevi sintetičkih primjeraka stvorenih uspoređenim algoritmima preuzorkovanja, pri čemu su vrijednosti u svakom stupcu normalizirane. Vidljivo je da predloženi algoritam na većini skupova podataka stvara približno isti broj sintetičkih primjeraka kao i ostale razmatrane unaprijeđene inačice algoritma SMOTE. Tek za skupove podataka $\mathcal{D}_2$, $\mathcal{D}_5$ i $\mathcal{D}_{11}$ broj primjeraka stvorenih ovim algoritmom je značajno veći u odnosu na ostale algoritme, što je vjerojatno posljedica dobre razdvojenosti većinskih i manjinskih primjeraka u tim skupovima podataka. Naime, ako su manjinski primjerci relativno blizu jedni drugima, a znatno udaljeniji od većinskih primjeraka, njihova susjedstva će biti velika te će predloženi algoritam stvoriti velik broj sintetičkih primjeraka. Pretpostavku da navedeni skupovi podataka imaju mali stupanj preklapanja klasa sugeriraju velike vrijednosti mjere F1 koju ostvaruju odabrani klasifikatori na tim skupovima podataka prije provedbe preuzorkovanja, što se vidi u tablicama 4.9, 4.10, 4.11 i 4.12. Unatoč tome, predobrada tih skupova podataka preuzorkovanjem uspijeva pridonijeti poboljšanju izvedbe korištenih klasifikatora, pri čemu predloženi algoritam uglavnom polučuje bolje rezultate od većine razmatranih algoritama. S druge strane, algoritam RESMOTE stvara najmanje sintetičkih primjeraka na svim skupovima podataka. Ovakav ishod vjerojatno je posljedica toga što ovaj algoritam ne preuzorkuje one manjinske primjerke koji su pretežito okruženi većinskim primjercima, a kojih u neuravnoteženim skupovima podataka uobičajeno ima puno. Zanemariv broj primjeraka stvorenih algoritmom RESMOTE može se smatrati razlogom iza njegova najlošijeg ranga u usporedbi s ostalim algoritmima preuzorkovanja i njegova beznačajna doprinosa poboljšanju izvedbe klasifikatora. Treba napomenuti da u literaturi postoji više unaprjeđenja algoritma SMOTE koji izbjegavaju preuzorkovanje onih manjinskih primjeraka koji imaju više susjeda iz većinske klase (primjerice, algoritmi predloženi u [166, 171]) ili čak uklanjaju takve primjerke iz skupa podataka (primjerice, algoritam predložen u [179]). Naravno, korištena veličina susjedstva u algoritmu RESMOTE i sličnim algoritmima ima značajan utjecaj na omjer većinskih i manjinskih primjeraka u susjedstvu, a time i na broj stvorenih



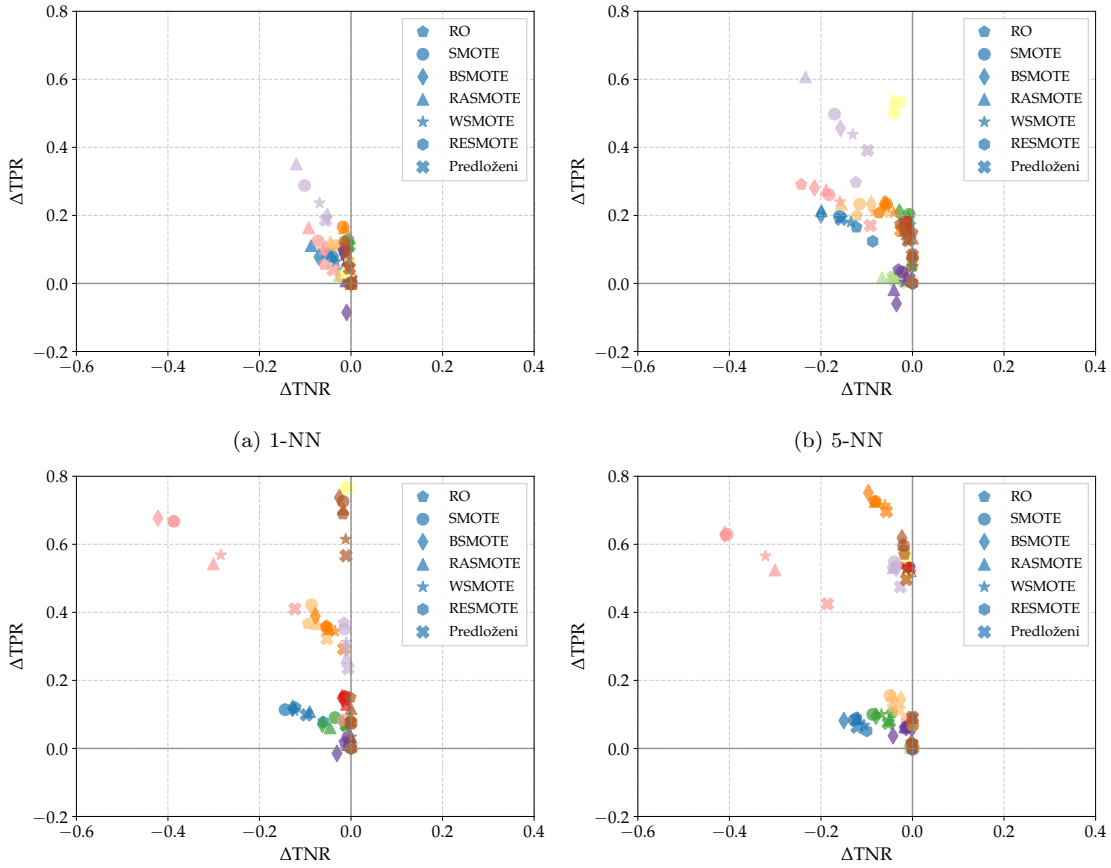
(a) 1-NN (b) 5-NN

(c) SVM (d) MLP

Slika 4.10: Usporedba broja stvorenih primjeraka korištenim algoritmima preuzorkovanja

sintetičkih primjeraka, pa je njezino podešavanje neophodno kako bi se izbjegao beznačajan učinak preuzorkovanja. Ipak, valja podsjetiti da je u ovoj eksperimentalnoj analizi korištena veličina susjedstva koja je predložena u izvornom radu [172], u kojem algoritam RESMOTE pokazuje povoljne performanse na tamo odabranim skupovima podataka.

Iako uspoređeni algoritmi preuzorkovanja na većini skupova podataka stvaraju približno jednak broj sintetičkih primjeraka, njihove performanse se često znatno razlikuju. Ove razlike u performansama upućuju na važnost položaja stvorenih sintetičkih primjeraka. Kako bi se detaljnije analizirale razlike u izvedbi odabranih algoritama, uspoređen je njihov doprinos uspješnosti klasifikatora za svaku klasu. Općenito, svi razmatrani algoritmi poboljšavaju prepoznavanje manjinske klase te zadržavaju ili narušavaju uspješnost prepoznavanja većinske klase, na što ukazuju razlike mjera TPR i TNR prije i nakon provedbe preuzorkovanja koje su prikazane na slici 4.11 različitom bojom za svaki skup podataka. Pri tome, povećanje mjere TPR u pravilu je značajnije od narušenja mjere TNR, što u konačnici rezultira i poboljšanjem opće izvedbe klasifikatora, na što ukazuju vrijednosti mjere F1 prikazane u tablicama 4.9, 4.10, 4.11 i 4.12. Međutim, uspoređeni algoritmi općenito se razlikuju prema svom utjecaju na uspješnost prepoznavanja pojedine klase, što sugeriraju njihovi prosječni



(a) 1-NN

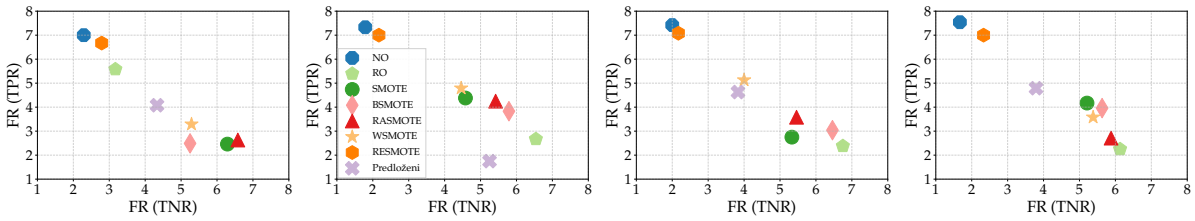
(b) 5-NN

(c) SVM

(d) MLP

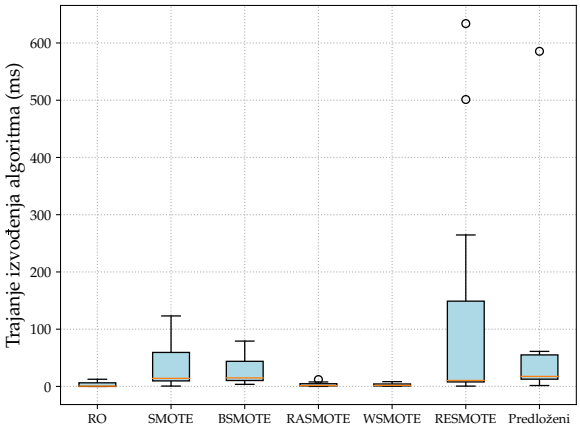
Slika 4.11: Razlike u vrijednostima mjera TNR i TPR ostvarenim nakon i prije provedbe preuzorkovanja

rangovi u smislu postignutih vrijednosti mjera TPR i TNR koji su prikazani na slici 4.12. Moguće je primijetiti da algoritmi koji najviše doprinose poboljšanju prepoznavanja manjinske klase (RO, BSMOTE i RASMOTE) ujedno i najviše narušavaju izvedbu klasifikatora za većinsku klasu. S obzirom na to da algoritam BSMOTE stvara sintetičke primjerke na granici s većinskom klasom, a RO i RASMOTE postavljaju sintetičke primjerke u ulaznom prostoru bez obzira na njihov položaj u odnosu na većinske primjerke, ne iznenađuje što značajnije narušavaju prepoznavanje većinske klase. Valja podsjetiti da ovi algoritmi ne uspijevaju nadmašiti izvorni algoritam po iznosu mjere F1, što daje naslutiti da povećanje stupnja preklapanja klasa radi poboljšanja prepoznavanja manjinskih primjeraka nije najprikladnija strategija preuzorkovanja. S druge strane, predloženi algoritam za većinu klasifikatora ostvaruje najbolji rang u smislu mjere TNR (nakon algoritma RESMOTE koji ima zanemariv učinak na uspješnost prepoznavanja obje klase), a ujedno doprinosi osjetnom poboljšanju mjere TPR što se može uočiti na slikama 4.7 i 4.11. Kao što je ranije rečeno, predloženi algoritam nastoji izbjeći povećanje stupnja preklapanja klasa u skupu podataka stvaranjem sintetičkih primjeraka unutar d-sfere određene promatranim manjinskim primjerkom i njegovim najbližim susjedom iz većinske klase, što je vjerojatno i razlog nje-



(a) 1-NN (b) 5-NN (c) SVM (d) MLP

Slika 4.12: Rangovi korištenih algoritama preuzorkovanja u smislu ostvarenih vrijednosti mjera TPR i TNR



Slika 4.13: Usporedba prosječnog trajanja unaprijeđenih inačica algoritma SMOTE

gova neznatna narušavanja mjere TNR. Štoviše, moguće je zaključiti da predloženi algoritam postiže određeni kompromis između pogoršanja prepoznavanja većinske klase te poboljšanja prepoznavanja manjinske klase (što se vidi na slici 4.12) te stoga ostvaruje najbolje rangove u smislu mjere F1.

Kao što je ranije objašnjeno, predloženo unaprjeđenje ima veću vremensku složenost od algoritma SMOTE te je moguće očekivati duže trajanje izvođenja ovog algoritma. Kako bi se demonstrirao vremenski trošak njegova izvođenja, slika 4.13 prikazuje usporedbu prosječnog trajanja svih razmatranih algoritama preuzorkovanja. Iako predloženi algoritam pri određivanju susjedstva uzima u obzir većinsku klasu te provodi složeniji postupak stvaranja sintetičkih primjeraka, na većini problema traje podjednako kao i izvorni algoritam te njegova razmatrana unaprjeđenja. Trajanje izvođenja svakog algoritma varira ovisno o karakteristikama skupa podataka, no prikazani medijani se neznatno razlikuju. Cjelokupno gledano, moguće je zaključiti da predloženi algoritam poboljšava performanse izvornog algoritma u većoj mjeri nego što to postižu ostala razmatrana unaprjeđenja te ispoljava približno jednako trajanje izvođenja kao i SMOTE unatoč većoj vremenskoj složenosti (u najgorem slučaju). Osim toga, prikazano trajanje se može smatrati ukupnim troškom njegova korištenja jer on nema parametara koje je potrebno postaviti, dok je ostale algoritme potrebno opetovano izvoditi uz popratno treniranje klasifikacijskih modela kako bi se utvrdile prikladne vrijednosti njihovih parametara.

4.5 Osvrt na preuzorkovanje i predloženu unaprijeđenu inačicu algoritma SMOTE

Preuzorkovanje je važan postupak predobrade neuravnoteženih skupova podataka koji nastoji povećati broj manjinskih primjeraka te tako olakšati prepoznavanje koncepta manjinske klase. Kao što pokazuju rezultati eksperimentalne analize, poboljšanje prepoznavanja manjinske klase uzrokovano provedbom preuzorkovanja također doprinosi poboljšanju opće izvedbe klasifikatora. Većina algoritama preuzorkovanja oslanja se na prikupljanje lokalnih informacija o manjinskim primjercima radi stvaranja sintetičkih primjeraka u njihovoj okolini, a njihov glavni predstavnik jest algoritam SMOTE. Iako je ovaj algoritam iznimno popularan u literaturi zbog svoje učinkovitosti i jednostavnosti, ima određene nedostatke koji mogu uzrokovati povećanje složenosti skupa podataka te narušavanje kvalitete izvedbe klasifikatora. Nedostaci ovog algoritma izraženiji su pri neprikladnim postavkama njegovih parametara te se njihovo podešavanje može smatrati neophodnim korakom pri korištenju ovog algoritma. Kako bi se ostvarile povoljne performanse algoritma SMOTE te izbjegao dugotrajan i složen proces podešavanja njegovih parametara, predložena je njegova unaprijeđena inačica koja izvodi osnovne korake izvornog algoritma uzimajući u obzir unutarnje karakteristike skupa podataka. Za razliku od izvornog algoritma, predloženi algoritam samostalno određuje veličinu susjedstava manjinskih primjeraka te broj sintetičkih primjeraka koje treba stvoriti. Rezultati eksperimentalne analize sugeriraju da ostvaruje i bolje performanse od algoritma SMOTE na većini skupova podataka, neovisno o korištenim postavkama parametara izvornog algoritma.

Predloženi algoritam također je uspoređen s nekolicinom unaprjeđenja algoritma SMOTE iz literature, a pokazano je da generalno najviše doprinosi poboljšanju opće uspješnosti klasifikacije, neovisno o korištenom klasifikatoru. Povoljne performanse predloženog algoritma posljedica su njegove strategije preuzorkovanja prema kojoj stvara najviše sintetičkih primjeraka u području najveće gustoće manjinskih primjeraka te izbjegava njihovo uvođenje u područje većinske klase. S druge strane, brojne unaprijeđene inačice algoritma SMOTE zanemaruju položaj većinskih primjeraka pri stvaranju sintetičkih primjeraka (ili ih čak namjerno uvode u područje većinske klase), što u konačnici može rezultirati povećanom složenosti skupa podataka te znatno narušenom uspješnosti prepoznavanja većinske klase. Razmjer ovakvog nepovoljnog učinka uvelike je uvjetovan vrijednostima njihovih parametara te je njihovo podešavanje potrebno provesti za svaki problem kako bi se on umanjio. S druge strane, upotreba predloženog algoritma za ublažavanje problema neuravnoteženosti klasa pretežno je korisnija te vremenski manje zahtjevnija od upotrebe drugih algoritama preuzorkovanja jer nema parametara koje je potrebno postaviti. Time ono predstavlja ispunjenje prijedloga drugog izvornog znanstvenog doprinosa ove disertacije.

5

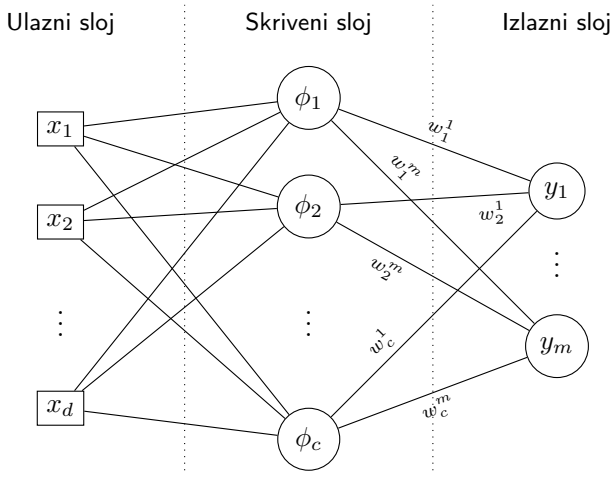
Izgradnja klasifikacijskih modela radijalne neuronske mreže

PRILIKOM učenja iz neuravnoteženih podataka izbor prikladnog klasifikatora od iznimne je važnosti, s obzirom na činjenicu da većina klasifikatora iskazuje pristranost većinskoj klasi te ostvaruje nisku razinu uspješnosti prepoznavanja manjinske klase. Radijalna neuronska mreža (RBFN) primjer je klasifikatora koji je na raznim neuravnoteženim problemima klasifikacije iskazao zadovoljavajuće ponašanje te polučio povoljne performanse, prvenstveno zbog svoje sposobnosti lokaliziranog djelovanja u ulaznom prostoru. Međutim, na ponašanje ovog klasifikatora uvelike utječu složenost klasifikacijskog modela i postavke njegovih parametara, čije određivanje predstavlja zahtjevan i dugotrajan zadatak. Ovo poglavlje daje osvrt na postojeće postupke za određivanje strukture RBFN, s posebnim naglaskom na postupke koji su primarno namijenjeni za izgradnju i treniranje klasifikacijskih modela. Nakon pregleda literature, opisan je prijedlog novog postupka izgradnje klasifikacijskih modela RBFN koji nastoji pronaći slijed mreža povećane složenosti koje postižu zadovoljavajuću uspješnost klasifikacije, osobito na neuravnoteženim problemima. Predloženi postupak ujedno predstavlja prijedlog trećeg izvornog znanstvenog doprinosa, a zasniva se na ideji postupnog povećanja složenosti prethodnih klasifikacijskih modela. Učinkovitost predloženog postupka eksperimentalno je ispitana na standardnim skupovima podataka različitih omjera neuravnoteženosti iz literature te je uspoređena s učinkovitošću nekoliko drugih postupaka za izgradnju klasifikacijskih modela RBFN.

5.1 Uvod u radijalne neuronske mreže

Nakon predobrade neuravnoteženih skupova podataka obično se pristupa izgradnji i treniranju klasifikacijskih modela, što je zadatak odabranog klasifikatora. U literaturi je predloženo pregršt klasifikatora koji se generalno razlikuju prema klasifikacijskom modelu koji treniraju, optimizacijskom postupku korištenom za treniranje te načinu vrednovanja modela tijekom treniranja [22]. U suštini, odabir prikladnog klasifikatora uvelike ovisi o složenosti i prirodi problema koji se nastoji naučiti. Kao što je pojašnjeno u poglavlju 2, većina standardnih klasifikatora iskazuje pristranost većinskoj klasi na neuravnoteženim skupovima podataka te ostvaruje nisku razinu uspješnosti prepoznavanja manjinske klase. Iako se odabirom značajki i preuzorkovanjem smanjuje složenost neuravnoteženih skupova podataka te pospješuje opća izvedba raznih tipova klasifikatora, ovi postupci ne ublažavaju u potpunosti problem neuravnoteženosti klasa. Stoga se pri odabiru klasifikatora ne smiju zanemariti njegova svojstva u pogledu učenja iz neuravnoteženih skupova podataka. Primjer klasifikatora koji je demonstrirao zadovoljavajuće ponašanje i ostvario povoljne performanse na brojnim neuravnoteženim problemima klasifikacije jest RBFN [13–15, 36, 37]. Uz to, ovaj klasifikator se doima posebice prikladnim za upotrebu na problemima u kojima je manjinska klasa podijeljena na podkoncepte (što je čest slučaj, kao što je istaknuto u poglavlju 2) zbog svoje sposobnosti lokaliziranog djelovanja u ulaznom prostoru [37, 52]. Povrh toga, često se udružuje s postupkom preuzorkovanja [20, 203, 204] s ciljem uspješnijeg prepoznavanja takve raspodjele primjeraka manjinske klase.

Radijalne neuronske mreže su, uz višeslojni perceptron (MLP), najčešće korištena vrsta umjetnih neuronskih mreža (ANNs) za klasifikaciju. Umjetne neuronske mreže jedna su od temeljnih paradigmi u području računalne inteligencije, a inspirirane su biološkim živčanim sustavom, prvenstveno mozgom. U širem smislu, ANNs predstavljaju kompozicije raznih funkcija koje se primjenjuju nad linearnom kombinacijom ulaznih podataka. U pogledu strukture, sastoje se od čvorova koji su predstavljeni tim funkcijama (koji po analogiji odgovaraju biološkom neuronu) i težinskih veza između čvorova (koje po analogiji odgovaraju sinapsama). Njihova upotreba podrazumijeva izgradnju odgovarajuće topologije mreže za promatrani problem, dok se treniranje mreže svodi na traženje prikladnih vrijednosti parametara čvorova i težina na vezama između njih. U literaturi je predložen pozamašan broj različitih vrsta ANNs, a RBFNs su karakteristične prema tome što u čvorovima skrivenog sloja ugrađuju radijalne funkcije (engl. *radial basis functions*), odnosno funkcije realne varijable čija vrijednost ovisi samo o udaljenosti argumenta od ishodišta ili neke centralne točke [205]. Radijalne funkcije svoju prvotnu primjenu pronalaze pri rješavanju problema aproksimacije funkcija gdje se rabe za interpolaciju skupa točaka u višedimenzionalnom prostoru [205]. Njihova učinkovitost na ovom problemu poslužila je kao motivacija Broomheadu i Loweu u [206] te Moodyiju i Darkenu u [207] za predlaganje RBFN. Primjenom radijalnih funkcija kao čvorova u ANNs cilj je bio ostvariti mreže s dobrom sposobnosti generalizacije



Slika 5.1: Uobičajena struktura RBFN [215]

i relativno malom složenosti u smislu broja čvorova kako bi se izbjegli dugotrajni postupci treniranja koje imaju ostale ANNs (primjerice, MLP) [208]. Osim aproksimacije funkcija te klasifikacije, ova mreža se često primjenjuje i za potrebe regresije [209], predviđanja vremenskih serija [210], obrade slika [211], prepoznavanja govora [212], obrade raznih oblika signala [213] i brojne druge [36].

5.1.1 Struktura radijalne neuronske mreže

U pogledu strukture, RBFN predstavlja umjetnu neuronsku mrežu s propagacijom signala prema naprijed (engl. *feed forward neural network*) koja se sastoji od ulaznog, skrivenog i izlaznog sloja. Struktura standardne RBFN prikazana je na slici 5.1, a određena je brojem čvorova u skrivenom sloju te odgovarajućim parametrima radijalnih funkcija u tim čvorovima, kao i težinama veza čvorova skrivenog i izlaznog sloja. Broj čvorova u ulaznom sloju jednak je broju značajki koje opisuju svaki primjerak u skupu podataka $\mathbf{x} = (x_1, \ldots, x_d) \in \mathcal{X}$, dok je broj čvorova u izlaznom sloju jednak broju oznaka klasa $m = |\mathcal{L}|$ promatranog problema. S druge strane, broj čvorova skrivenog sloja c određuje se tijekom izgradnje klasifikacijskog modela RBFN. Veze između čvorova ulaznog i skrivenog sloja su jedinične, dok se težine veza između čvorova skrivenog i izlaznog sloja $w_i^j \in \mathbb{R}$, $i = 1, \ldots, c$, $j = 1, \ldots, m$ definiraju tijekom treniranja klasifikacijskog modela. Svaki čvor skrivenog sloja ugrađuje radijalnu funkciju $\phi_i : \mathbb{R}^d \to \mathbb{R}$, $i = 1, \ldots, c$ kao svoju aktivacijsku funkciju. Prema tome, izlaz RBFN može se definirati kao linearna kombinacija [214]

$$y_j = \sum_{i=1}^{c} w_i^j \cdot \phi_i(\mathbf{x}), \qquad j = 1, ..., m \ . \tag{5.1}$$

U literaturi je razmotreno nekoliko oblika radijalnih funkcija za upotrebu u čvorovima skrivenog sloja, poput višekvadratne i njezine obrnute inačice, kubne, linearne i drugih [214, 216]. Pri tome, uvjerljivo najčešće korištena je Gaussova funkcija [214, 217, 218]

$$\phi(\mathbf{x}) = e^{-\frac{||\mathbf{x}-\mathbf{z}||^2}{2\sigma^2}}, \tag{5.2}$$

gdje $\mathbf{z} = (z_1, \ldots, z_d) \in \mathbb{R}^d$ predstavlja centar funkcije, $\sigma > 0$ pripadajuću širinu, a $\|\cdot\|$ ℓ_2 (Euklidsku) normu. Ova funkcija jest lokalizirana radijalna funkcija, sa svojstvom da joj vrijednost opada s udaljenosti ulaznog primjerka $\mathbf{x}$ od centra $\mathbf{z}$. Funkcija tako postiže istu vrijednost za sve točke jednako udaljene od centra. Ugradnjom odgovarajućeg broja lokaliziranih radijalnih funkcija u čvorove RBFN, ova mreža ima sposobnost univerzalne aproksimacije bilo koje kontinuirane funkcije [219]. Uz navedeno svojstvo, Gaussova funkcija ima razna pogodna analitička svojstva koja olakšavaju treniranje RBFN [214]. S Gaussovim radijalnim funkcijama ugrađenim u čvorove skrivenog sloja, izlaz mreže definiran s (5.1) može se iznova napisati kao

$$y_j = \sum_{i=1}^{c} w_i^j \cdot e^{-\frac{||\mathbf{x}-\mathbf{z}^i||^2}{2\sigma_i^2}}, \qquad j = 1, \ldots, m \ . \tag{5.3}$$

Pri tome, centri $\mathbf{z}^1, \ldots, \mathbf{z}^c$ određuju položaje radijalnih funkcija, dok pripadajuće širine $\sigma_1, \ldots, \sigma_c$ određuju područje ulaznog prostora koji ove funkcije pokrivaju. Vrijednosti funkcija u čvorovima skrivenog sloja utječu na vrijednosti izlaza mreže, odnosno na odluku klasifikacijskog modela, pri čemu težine w_i^j, $i = 1, \ldots, c$, $j = 1, \ldots, m$ određuju razmjer tog utjecaja. Vrijednost na svakom od izlaza y_j, $j = 1, \ldots, m$ određuje pripadnost ulaznog primjerka $\mathbf{x}$ jednoj od klasa, a odluka o njegovoj oznaci klase donosi se prema najvećoj vrijednosti izlaza.

S obzirom na prikazanu strukturu RBFN, jasno je kako ovaj klasifikator koristi preklapajuća lokalizirana područja ulaznog prostora formirana jednostavnim radijalnim funkcijama za stvaranje složenih granica odluke [208]. Prvenstveno zbog ovog svojstva, često postiže povoljne performanse na neuravnoteženim skupovima podataka jer formiranje takvih granica odluke olakšava prepoznavanje koncepta manjinske klase koji je općenito zastupljen s neznatnim brojem primjeraka te podijeljen na podkoncepte [37, 52]. Pri tome, klasifikator RBFN izvodi nelinearnu transformaciju ulaznih primjeraka u prostor veće dimenzije nego što je dimenzija ulaznog prostora [220]. Motivacija za ovaj postupak leži u Coverovom teoremu o razdvojivosti primjeraka, prema kojem je veća vjerojatnost da primjerci budu linearno razdvojivi u prostoru veće dimenzionalnosti [221]. Da bi se demonstriralo ponašanje RBFN u postupku donošenja odluke o pripadnosti primjerka nekoj klasi, djelovanje svakog skrivenog čvora s ugrađenom Gaussovom funkcijom [danom prema (5.2)] u ulaznom prostoru može se predočiti d-sferom smještenom u centru čvora i polumjerom jednakim odgovarajućoj širini. Ulazni primjerci koji se nalaze unutar d-sfere dovest će do visoke aktivacije odgovarajućeg čvora. Uz pretpostavku dobro definirane mreže, očekuje se da ove d-sfere pokrivaju područja ulaznog prostora koji uglavnom sadrže primjerke iz iste klase [222]. Ipak, s obzirom na činjenicu da se djelovanja čvorova međusobno preklapaju u ulaznom prostoru, moguće je

tvrditi da između njih postoje složene interakcije koje utječu na konačnu odluku treniranog klasifikacijskog modela.

Dakako, odgovarajuću strukturu RBFN potrebno je odrediti za svaki problem da bi ovaj klasifikator postigao povoljnu razinu izvedbe. Struktura i parametri mreže određuju se tijekom izgradnje klasifikacijskog modela te njegova treniranja, za što se mogu koristiti razni postupci predloženi u literaturi. Treba napomenuti da je pri izgradnji klasifikacijskog modela RBFN prikladnije koristiti normalizirane izlaze čvorova skrivenog sloja. Prema tome, izlaz mreže može se umjesto (5.1) napisati kao [207]

$$y_j = \frac{\sum_{i=1}^{c} w_i^j \cdot \phi_i(\mathbf{x})}{\sum_{i=1}^{c} \phi_i(\mathbf{x})}, \qquad j = 1, \ldots, m \ . \tag{5.4}$$

Ovakve mreže uobičajeno imaju bolju sposobnost generalizacije te rezultiraju manjim brojem čvorova skrivenog sloja u odnosu na stadnardne RBFN [223, 224].

5.1.2 Treniranje RBFN i izgradnja klasifikacijskog modela

Izgradnja klasifikacijskog modela RBFN podrazumijeva određivanje broja čvorova skrivenog sloja, što ujedno predstavlja hiperparametar klasifikacijskog modela. Odgovarajući parametri tih čvorova (centri i pripadajuće širine radijalnih funkcija), kao i težine veza između čvorova skrivenog i izlaznog sloja, određuju se tijekom treniranja klasifikacijskog modela. Prikladan broj čvorova skrivenog sloja nije jednostavno odrediti te se on u pravilu zasebno utvrđuje za svaki problem. Prilikom primjene RBFN za aproksimacije raznih funkcija, većina primjeraka iz skupa podataka obično se postavlja za centre radijalnih funkcija [206]. Međutim, za potrebe klasifikacije ovakav dizajn mreže može dovesti do prenaučenosti, odnosno do slabe generalizacije klasifikatora [225]. S druge strane, upotreba relativno malog broja čvorova u skrivenom sloju obično rezultira podnaučenosti klasifikatora. S obzirom na složenost problema utvrđivanja odgovarajućeg broja čvorova skrivenog sloja, nije iznenađujuće što je predloženo pregršt postupaka za izgradnju klasifikacijskog modela RBFN koji nastoje ostvariti što bolju sposobnost generalizacije klasifikatora uz minimalnu složenost mreže [226].

Nakon odabira odgovarajuće strukture mreže, uobičajeno se provodi treniranje klasifikacijskog modela RBFN na skupu označenih primjeraka $\mathcal{T} = \{(\mathbf{x}^i, l_{t(i)}) : \mathbf{x}^i \in \mathcal{X}, l_{t(i)} \in \mathcal{L}, i = 1, \ldots, N\}$. Cilj treniranja jest definirati parametre mreže takve da vrijednosti na izlaznim čvorovima budu što bliže željenim vrijednostima. U pogledu klasifikacije, željena vrijednost svakog izlaznog čvora mreže jedna je od dvije moguće vrijednosti $o_j \in \{0, 1\}$, pri čemu o_j označava pripada li ulazni primjerak $\mathbf{x}$ klasi s oznakom l_j, $j = 1, \ldots, m$. Određivanje željenih vrijednosti o_i^j, $j = 1, \ldots, m$ na izlaznim čvorovima za ulazni primjerak $\mathbf{x}^i$ s oznakom klase $l_{t(i)}$, $i = 1, \ldots, N$ može se opisati funkcijom

$$g((\mathbf{x}^i, l_{t(i)}), j) = \begin{cases} 1, & \text{ako } l_{t(i)} = j \\ 0, & \text{u suprotnom} \end{cases} . \tag{5.5}$$

Proces treniranja klasifikacijskog modela RBFN uobičajeno se provodi u dvije faze [207]. U prvoj fazi definiraju se parametri aktivacijskih funkcija unutar čvorova, dok se u drugoj fazi određuju vrijednosti težina veza čvorova skrivenog i izlaznog sloja. Za provođenje obje faze predloženi su brojni postupci u literaturi. Pri tome, provođenje prve faze treniranja može biti relativno jednostavno i brzo te se oslanjati na položaje ulaznih primjeraka za određivanje centara i širina radijalnih funkcija. S druge strane, prilikom primjene RBFN na problemima klasifikacije, parametri radijalnih funkcija često se definiraju na temelju njihova doprinosa performansama klasifikacijskog modela, što uobičajeno uključuje upotrebu određenog algoritma optimizacije kao postupka za treniranje. Iako je ovakav način provođenja prve faze treniranja složeniji i dugotrajniji, obično rezultira mrežama sa znatno boljom kvalitetom izvedbe. Nakon što su parametri skrivenog sloja definirani, slijedi faza određivanja težina koja je obično jednostavnija od prve faze jer se težine mogu estimirati na temelju prethodno definiranih centara i širina. Uključivanjem skupa od N označenih uzoraka $\mathbf{x}^1, \ldots, \mathbf{x}^N$ i željenih izlaza za svaki od njih, (5.3) može se zapisati u matričnom obliku kao sustav linearnih jednadžbi

$$\mathbf{o}^j = \mathbf{\Phi} \cdot \mathbf{w}^j, \qquad j = 1, \ldots, m, \tag{5.6}$$

gdje je $\mathbf{o}^j = [o_1^j, \ldots, o_N^j]^T$ vektor željenih vrijednosti na j-tom izlaznom čvoru, $\mathbf{w}^j = [w_1^j, \ldots, w_c^j]^T$ vektor težina veza čvorova skrivenog i izlaznog sloja, dok je $\mathbf{\Phi} \in \mathbb{R}^{N \times c}$ matrica s komponentama $\Phi_{r,s} = e^{-||\mathbf{x}^r - \mathbf{z}^s||^2/(2\sigma_s^2)}$, $r = 1, ..., N$, $s = 1, ..., c$, koji predstavljaju izlaze iz čvorova skrivenog sloja. S obzirom na to da $\mathbf{\Phi}$ u pravilu nije kvadratna matrica (odnosno, vrijedi $c < N$), težine je moguće odrediti metodom najmanjih kvadrata uz uporabu generaliziranog inverza (Moore-Penroseov inverz) $\mathbf{\Phi}^{-1}$ matrice $\mathbf{\Phi}$ [227]. Rješavanjem linearnog problema najmanjih kvadrata

$$\mathbf{w}^j = \mathbf{\Phi}^{-1} \cdot \mathbf{o}^j, \qquad j = 1, \ldots, m, \tag{5.7}$$

dobivaju se težine koje minimiziraju grešku $||\mathbf{\Phi} \cdot \mathbf{w}^j - \mathbf{o}^j||^2$ [214, 228]. Mogućnost analitičkog određivanja težina jedna je od glavnih prednosti RBFN u odnosu na većinu ostalih ANNs jer se time značajno ubrzava treniranje. Stoga ovaj postupak predstavlja učestao odabir za drugu fazu treniranja klasifikacijskog modela RBFN. Ipak, u praksi ponekad nije jednostavno odrediti generalizirani inverz $\mathbf{\Phi}^{-1}$ zbog raznih numeričkih poteškoća (primjerice, determinanta matrice je blizu nuli, matrica je velikih dimenzija i slično [229]) pa se optimalne vrijednosti težina pokušavaju odrediti iterativnim metodama optimizacije, poput metode gradijentnog spusta (engl. *gradient descent*, GD). Ove metode nastoje pronaći

vrijednosti težina koje minimiziraju zbroj kvadrata greški (engl. *sum of squared errors*, SSE)

$$\frac{1}{2} \cdot \sum_{i=1}^{N} \sum_{j=1}^{m} (o_i^j - y_{j,i})^2, \tag{5.8}$$

gdje je o_i^j željeni izlaz na j-tom čvoru za i-ti primjerak, a $y_{j,i}$ izlaz j-tog izlaznog čvora mreže za i-ti primjerak. Ponekad se umjesto SSE koristi srednja kvadratna greška (engl. *mean squared error*, MSE)

$$\frac{\sum_{i=1}^{N} \sum_{j=1}^{m} (o_i^j - y_{j,i}^2)}{N \cdot m}, \tag{5.9}$$

koja je prikladnija od SSE ako se želi napraviti usporedba učinkovitosti mreže na različitim skupovima primjeraka. Obje mjere predstavljaju ukupnu grešku mreže u smislu dobivenih vrijednosti na izlaznim čvorovima u odnosu na željene, dok je kvadrat greške (engl. *squared error*, SE) za svaki pojedini primjerak $\mathbf{x}_i$ jednak

$$\sum_{j=1}^{m} (o_i^j - y_{j,i})^2 \ . \tag{5.10}$$

Navedene mjere se u pravilu koriste za iskazivanje kvalitete klasifikacijskog modela RBFN i tijekom druge faze njegova treniranja [214], primarno zbog mogućnosti analitičkog određivanja težina te derivabilnosti mjera SSE i MSE. S obzirom na to da stavljaju jednaku važnost na ispravno prepoznavanje svih primjeraka neovisno o oznaci njihove klase, moguće je naslutiti da nisu prikladne za iskazivanje kvalitete klasifikacijskog modela na neuravnoteženim skupovima podataka. Međutim, u literaturi je mnoštvo puta pokazano (primjerice, u [215, 230, 231]) da smanjenje greške mreže koju ona ostvaruje na odvojenom skupu za testiranje najčešće rezultira i poboljšanjem iznosa drugih mjera uspješnosti klasifikacije koje su pogodnije za neuravnotežene skupove podataka (primjerice, F1). Stoga je moguće zaključiti da se korištenjem spomenutih mjera može zadržati jednostavnost postupka treniranja te postići zadovoljavajuća kvaliteta izvedbe klasifikatora i u smislu drugih mjera uspješnosti klasifikacije.

Kao što je vidljivo iz prikazanog, ovu mrežu odlikuje relativno velik broj parametara koje je potrebno odrediti tijekom izgradnje i treniranja klasifikacijskog modela. Pronalazak parametara mreže koji će rezultirati mrežom poželjnih svojstava, prvenstveno dobrom sposobnosti generalizacije te malom složenosti, nije jednostavno. S obzirom na široku primjenjivost RBFN, nije iznenađujuće što je predloženo mnoštvo postupaka za izgradnju strukture mreže i određivanje njezinih parametara.

5.2 Postupci za izgradnju i treniranje klasifikacijskih modela RBFN

Većina postupaka za izgradnju klasifikacijskih modela RBFN u literaturi pretpostavlja da je unaprijed poznata ili zadana topologija mreže, odnosno broj čvorova u skrivenom sloju. Ovi postupci stoga se fokusiraju isključivo na treniranje mreže zadane topologije, odnosno traženje parametara radijalnih funkcija te težina veza čvorova skrivenog i izlaznog sloja. Tek nekolicina postupaka objedinjuje traženje broja čvorova skrivenog sloja i traženje preostalih parametara mreže te tako provode automatski način izgradnje klasifikacijskog modela RBFN. U nastavku je izložen sažet pregled postupaka za izgradnju i treniranje klasifikacijskih modela RBFN predloženih u literaturi. Valja napomenuti da neki od izloženih postupaka nisu izvorno predloženi za potrebe klasifikacije nego za potrebe aproksimacije funkcija, no to ne predstavlja ograničavajući čimbenik u smislu njihove primjene za traženje parametara RBFN.

5.2.1 Pregled literature

Jedan od najjednostavnijih postupaka za treniranje RBFN predložili su Broomhead i Lowe u [206] koji nasumično odabire unaprijed određen broj primjeraka iz skupa podataka (uobičajeno sve primjerke) za centre radijalnih funkcija, dok se težine određuju metodom najmanjih kvadrata (kao što je ranije opisano). Nedostatak ovog postupka jest taj što se za centre mogu odabrati primjerci koji nemaju velik utjecaj na odluku klasifikatora ili pak primjerci koji predstavljaju šum, što obično dovodi do slabe generalizacije klasifikatora RBFN [219]. Stoga su Yousef i El-Hindi u [232] predložili primjenu tehnike za uklanjanje šuma iz skupa podataka nakon koje se preostali primjerci odabiru za centre radijalnih funkcija, dok se širine određuju nasumično. Ovakva mreža daje značajno bolje performanse od mreže koja koristi sve primjerke kao centre jer potonji dizajn može dovesti do prenaučenosti klasifikatora. Međutim, općeniti nedostatak ovih jednostavnih postupaka treniranja jest taj što obično moraju odabrati velik broj primjeraka za centre da bi mreža mogla na odgovarajući način (s relativno malim iznosom greške) naučiti preslikavanje ulaznih podataka u izlazne jer ne rade nikakve pretpostavke o njihovoj distribuciji u ulaznom prostoru.

Unatoč brzini i jednostavnosti navedenih postupaka izgradnje, u [219, 233] je pokazano da se najbolje lokacije centara ne moraju nužno podudarati s lokacijama primjeraka iz skupa podataka te da je prikladnije razmatrati cijeli ulazni prostor pri traženju parametara radijalnih funkcija. U tom pogledu, brojni postupci treniranja RBFN oslanjanju se na upotrebu algoritama za grupiranje podataka koji nastoje pronaći predstavnike područja ulaznog prostora s najvećom gustoćom primjeraka [207, 208, 234]. Primjerice, Moody i Darken su u [207] upotrijebili algoritam k-means za traženje centara radijalnih funkcija, što se postiže raspodjelom dostupnih primjeraka u k grupa, a predstavnici grupa se postavljaju za cen-

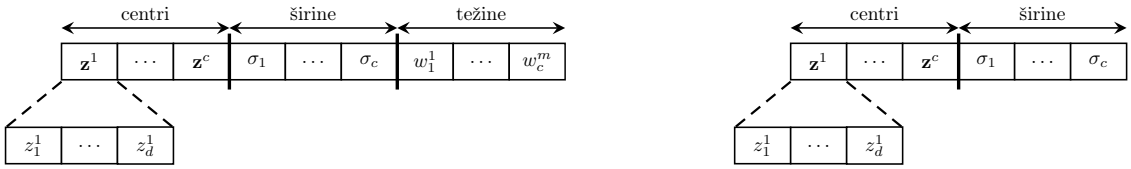
tre. Pripadajuće širine računaju se kao prosječna udaljenost najbližih parova centara te su jednake za sve čvorove, a težine veza skrivenog i izlaznog sloja računaju se pomoću metode GD. Međutim, jedno od glavnih ograničenja pri upotrebi grupiranja podataka za treniranje klasifikacijskog modela RBFN jest to što se ne uzimaju u obzir oznake primjeraka pri njihovu grupiranju te je moguće da radijalne funkcije pokrivaju područja ulaznog prostora s velikim stupnjem preklapanja klasa (primjerci različitih klasa često se neznatno razlikuju u vrijednostima značajki, kao što je objašnjeno u poglavlju 2) što može umanjiti kvalitetu izvedbe klasifikatora (posebice za manjinsku klasu). S obzirom na to da postupci grupiranja ne uzimaju u obzir uspješnost klasifikacije, a opetovano treniranje klasifikacijskog modela RBFN može biti vremenski zahtjevno, nekolicina postupaka koristi druge klasifikatore za pronalazak prikladnih lokacija centara [222, 235, 236]. Primjerice, Vogt je u [222] predložio uporabu klasifikacijskog algoritma kvantizacija vektora učenja (engl. *learning vector quantization*) za određivanje centara, dok su pripadajuće širine postavljene kao

$$\tau \cdot u_{min,i}, \tag{5.11}$$

gdje je $u_{min,i} = \min\{\|\mathbf{z}^i - \mathbf{z}^j\| : j = 1, \ldots, c \wedge j \neq i\}$, $i = 1, \ldots, c$, pri čemu se vrijednost $\tau = 1.2$ empirijski pokazala kao najbolji izbor.

Konačno, pozamašan broj postupaka odabire parametre RBFN na temelju performansi same mreže, pri čemu se u suštini razlikuju u optimizacijskom algoritmu korištenom za treniranje te broju parametara koje podešavaju. Osim što se često koristi za traženje težina veza čvorova skrivenog i izlaznog sloja, metoda GD je u [237, 238] korištena i za određivanje parametara radijalnih funkcija s ciljem minimizacije SSE, uz unaprijed poznat broj čvorova skrivenog sloja. Na sličan način, Wettsscherek i Dietterich su u [239] koristili metodu GD za traženje povoljnih lokacija centara koje su inicijalno postavljene na pozicije nasumično odabranih primjeraka ili predstavnike grupa nađenih algoritmom k-means, pri čemu su širine postavljene na unaprijed zadane nepromjenjive vrijednosti. Ipak, valja istaknuti da iterativne metode optimizacije (poput metode GD) imaju sklonost zaglavljivanja u lokalnom optimumu kojih može biti mnogo zbog multimodalnosti problema treniranja RBFN [216].

S obzirom na ograničenost tradicionalnih metoda optimizacije i veliku dimenzionalnost problema treniranja klasifikacijskog modela RBFN, bio-inspirirani algoritmi iskazuju se kao obećavajuće metode za pretragu prostora mogućih rješenja. Ovi postupci treniranja uobičajeno pronalaze mreže s istaknutom sposobnosti generalizacije [214, 233], no zahtijevaju velik broj vrednovanja funkcije cilja koja je najčešće predstavljena mjerom MSE. Pri tome, rješenja mogu biti predstavljena na različite načine, a dva najčešća načina su ilustrirani slikom 5.2. U prvom načinu, svi parametri mreže izravno su kodirani u vektor dimenzionalnosti $c \cdot (d + 1 + m)$. S obzirom na činjenicu da se težine veza čvorova izlaznog i skrivenog sloja mogu estimirati na temelju centara i pripadajućih širina metodom najmanjih kvadrata, njihovo kodiranje može se izbjeći te je stoga češći način predstavljanja rješenja onaj prikazan

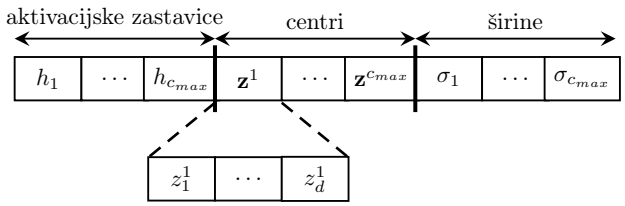


(a) S kodiranjem težina (b) Bez kodiranja težina

Slika 5.2: Načini predstavljanja rješenja u bio-inspiriranim algoritmima pri zadanom broju čvorova u skrivenom sloju [215]

na slici 5.2b. Ovaj način predstavljanja rješenja koristili su Korürek i Doğan u [230], gdje su predložili primjenu algoritma PSO za traženje parametara unaprijed zadanog broja čvorova skrivenog sloja. Pokazano je da mreže trenirane algoritmom PSO daju znatno bolje performanse od mreža treniranih pomoću algoritma k-means na različitim problemima klasifikacije. Nadalje, Kurban i Beşdok u [240] primijenili su algoritam ABC za traženje svih parametara mreže (koristili su način predstavljanja rješenja sa slike 5.2a) koji je pronašao mreže boljih performansi od mreža treniranih pomoću algoritma GA te metode GD. Povrh toga, u navedene bio-inspirirane algoritme ponekad se uvode dodatni operatori ili se postojeći unaprjeđuju s ciljem poboljšanja sposobnosti konvergencije ovih algoritama. Tako su Lu et al. u [241] predložili unaprijeđenu strategiju podešavanja parametra inercije (engl. *inertia weight*) u algoritmu PSO zasnovanu na upotrebi eksponencijalne funkcije, pomoću koje se nastoji djelotvornije kontrolirati odnos pretraživanja i iskorištavanja rješenja. Nadalje, Yu et al. su u [242] predložili unaprijeđeni algoritam ABC za potrebe treniranje klasifikacijskog modela RBFN koji periodički lošiju (u smislu kvalitete) polovicu pčela radilica pretvara u pčele skaute, dok pčele radilice mogu tražiti rješenja samo na određenoj udaljenosti od najboljeg pronađenog rješenja. Učinkovitost nekoliko bio-inspiriranih algoritama za potrebe treniranja klasifikacijskih modela RBFN eksperimentalno je uspoređena u [243], pri čemu su se algoritmi PSO i DE pokazali najprikladnijim izborom.

Svi prethodno opisani postupci treniranja RBFN pretpostavljaju da je broj čvorova skrivenog sloja unaprijed određen ili poznat. Međutim, to u pravilu nije slučaj te je prikladan broj potrebno odrediti za svaki problem. Stoga su predloženi razni postupci koji provode automatsku izgradnju klasifikacijskog modela RBFN, gdje se broj čvorova skrivenog sloja određuje usporedno s parametrima klasifikacijskog modela. Jednostavniji postupci automatske izgradnje RBFN prvotno su predloženi za rješavanje problema aproksimacije funkcija, a zasnivaju se na postupnom povećanju ili smanjenju složenosti prethodnih mreža. Kao i ranije spomenuti jednostavni postupci treniranja, ovi postupci izgradnje također uzimaju primjerke u skupu podataka kao kandidate za centre radijalnih funkcija, no njihov odabir nije nasumičan već je vođen performansama mreže. Osim toga, kriterij dodavanja ili uklanjanja čvorova u skrivenom sloju također je zasnovan na performansama mreže ili pak njezinoj složenosti. Jednostavan primjer takvog postupka izgradnje predložio je Platt u [244], gdje se mreža postupno gradi počevši od nasumično odabranog primjerka kao centra radijalne



Slika 5.3: Način predstavljanja rješenja u bio-inspiriranim algoritmima pri nepoznatom broju čvorova u skrivenom sloju [215]

funkcije te se u skriveni sloj opetovano dodaju čvorovi s centrima postavljenim u primjerke za koji je SE veći od unaprijed određene vrijednosti. Pri tome, širine dodanih čvorova su izračunate slično kao u [222], odnosno kao $\tau \cdot u_{min,i}$ (pri čemu je $\tau = 0.87$), a po svakom dodavanju iznova se računaju težine veza čvorova skrivenog i izlaznog sloja pomoću metode GD. Nadalje, Chen et al. su u [245] izmijenili spomenuti postupak na način da u svakom koraku za centar radijalne funkcije postavljaju primjerak s najvećim SE, dok se širine za sve čvorove postavljaju na istu vrijednost (empirijski određenu za svaki problem). Postupak se ponavlja sve dok mreža ne postigne željeni iznos mjere SSE. S druge strane, nekolicina postupaka iz literature (kao primjerice oni u [208, 246]) postupno grade mrežu uklanjanjem čvorova skrivenog sloja, pri čemu uobičajeno započinju proces uklanjanja od mreže s brojem čvorova u skrivenom sloju jednakim broju primjeraka iz skupa za treniranje, a završavaju ga dok se ne dosegne željena razina složenosti mreže. Jedan od nedostataka navedenih postupaka automatske izgradnje RBFN jest nemogućnost ponovna vrednovanja korisnosti određenog čvora u kasnijoj fazi, nakon što je donesena odluka o njegovu zadržavanju ili odbacivanju. Stoga su Yingwei et al. u [247] te Han et al. u [248] predložili postupke koji grade mreže slično kao i postupci predloženi u [244, 245], no periodički preispituju hoće li uklanjanje nekog od čvorova skrivenog sloja rezultirati povećanjem performansi mreže. Cjelokupno gledano, ovi postupci automatske izgradnje RBFN u suštini predstavljaju različite tehnike za odabir podskupa primjeraka iz skupa podataka za centre radijalnih funkcija s obzirom na to da se prvotno postavljene vrijednosti ne mijenjaju u kasnijoj fazi treniranja. Tek su Yu et al. u [249] predložili postupak izgradnje RBFN koji postupno gradi mreže postavljajući u svakom koraku centar novog čvora inicijalno na poziciju primjerka s najvećim SE (a njegovu širinu i težine veza s čvorovima izlaznog sloja na vrijednost 1), ali koji potom provodi treniranje cijele mreže pomoću algoritma Levenberg–Marquardt. Iako ovaj postupak traje duže od prethodno spomenutih postupaka izgradnje RBFN jer u svakom koraku provodi treniranje cijele mreže, pokazano je da rezultira boljim performansama nego da se vrijednosti parametara prethodno definiranih čvorova ostave nepromijenjenima.

Prethodno razmatranje načina predstavljanja rješenja u bio-inspiriranim algoritmima korištenim za treniranje RBFN (ilustrirano slikom 5.2) također je zasnovano na pretpostavci da je broj čvorova c u skrivenom sloju unaprijed zadan ili poznat. Slikom 5.3 ilustriran je mogući način predstavljanja rješenja koji uz parametre radijalnih funkcija određuje i broj čvorova.

Prikazani vektor rješenja uključuje i aktivacijske zastavice $h_j \in [0, 1]$, $j = 1, \ldots, c_{max}$ koje određuju hoće li se koristiti odgovarajući centar i pripadajuća širina u rješenju kao parametar mreže. Točnije, ako je $h_r > 0.5$, odgovarajući centar $\mathbf{z}^r$ i širina σ_r uzimaju se kao dio rješenja, dok ih se u suprotnom zanemaruje. Dimenzionalnost vektora rješenja, jednaka $c_{max} \cdot (d + 2)$, ovisi i o najvećem dozvoljenom broju čvorova u skrivenom sloju c_{max}, koji je potrebno podesiti. Ovakav način predstavljanja rješenja rabili su Qin et al. u [231] za algoritam PSO te Bajer et al. u [215] za algoritam DE. Pri tome, oba postupka dodatno uključuju mehanizme specifične za problem treniranja RBFN koji pospješuju konvergenciju korištenih bio-inspiriranih algoritama. Tako je u [231] predložena metoda inicijalizacije populacije gdje se kao centri nasumično odabiru primjerci iz skupa podataka te se za svaki centar računa pripadajuća širina kao njegova prosječna udaljenost do preostalih centara. S druge strane, u [215] je korišten algoritam k-means za generiranje dijela početne populacije te je dinamički sužavan prostor pretrage periodičkim povećanjem granice c_{min}, odnosno smanjenjem granice c_{max}. Međutim, kodiranje broja čvorova u vektor rješenja usmjeruje ove algoritme na traženje većih mreža, s obzirom na činjenicu da greška mreže na skupu za treniranje opada s povećanjem njezine složenosti [250]. No, suviše velik broj čvorova u skrivenom sloju može dovesti do prenaučenosti klasifikatora, kao što je ranije objašnjeno. Stoga se u funkciju cilja obično uvodi član kazne koji je proporcionalan broju čvorova skrivenog sloja c, odnosno $\lambda \cdot c$, gdje je $\lambda \in [0, +\infty)$ težina koja skalira parametar c [215, 231]. Prema tome, kazna je veća za veće mreže i obratno, čime se nastoji ostvariti ravnoteža između sposobnosti generalizacije mreže i njezine složenosti.

5.2.2 Kritički osvrt

Na temelju izloženog pregleda literature moguće je zaključiti kako postoji velik broj različitih postupaka za treniranje klasifikacijskih modela RBFN, no tek nekolicina postupaka koji provode automatski način izgradnje u kojem se složenost mreže određuje zajedno s ostalim parametrima. Postupci izgradnje klasifikacijskih modela RBFN u suštini imaju dva glavna cilja, a to su ostvarivanje dobre sposobnosti generalizacije klasifikatora te održavanje relativno jednostavnog modela u smislu broja čvorova u skrivenom sloju. Oni se mogu podijeliti na jednostavnije postupke koji postupno povećavaju složenost prethodnih modela i složenije bio-inspirirane algoritme koji u rješenja kodiraju i broj čvorova skrivenog sloja. Pri tome, prednost jednostavnijih postupaka u odnosu na potonje jest ta što po završetku izgradnje daju na raspolaganje velik broj treniranih klasifikacijskih modela koje je potom jednostavno vrednovati na odvojenom skupu za ispitivanje te odabrati onaj s najboljom sposobnosti generalizacije. Međutim, ovi postupci su izvorno predloženi za rješavanje problema aproksimacije funkcija koji se razlikuju od problema klasifikacije u nekoliko bitnih karakteristika. Primjerice, mreže primijenjene za potrebe aproksimacije funkcija imaju samo jedan čvor u izlaznom sloju, a primjerci u skupu podataka korišteni za treniranje mreže nisu svrstani u

klase. S obzirom na navedene razlike ovih problema, moguće je identificirati nekoliko nedostataka jednostavnijih postupaka automatske izgradnje RBFN iz literature u slučaju njihove primjene na problemima klasifikacije. Kao prvo, većina ovih postupaka se oslanja isključivo na primjerke u skupu podataka kao kandidate za centre radijalnih funkcija te upotrebu raznih jednostavnih heuristika za određivanje širina, unatoč tome što je u literaturi pokazano [219, 233] da se povoljnije performanse klasifikatora RBFN ostvaruju traženjem ovih parametara u cijelom ulaznom prostoru. Nadalje, pri dodavanju novog čvora u skriveni sloj ovi postupci ne provode treniranje cijele mreže, već isključivo dodanog čvora, iako svi čvorovi međudjeluju pri donošenju odluke klasifikacijskog modela, kao što je ranije opisano. Unatoč tome što mreže veće složenosti u pravilu iskazuju manju grešku na skupu za treniranje, dodavanje novog čvora u skriveni sloj s neprikladnim postavkama parametara u odnosu na ostale čvorove potencijalno može povećati SSE, zbog postojanja složenih interakcija između ovih čvorova.

Treniranjem cijele mreže nakon dodavanja svakog čvora mogu se prevladati spomenuti nedostaci, no ono se provodi tek u [249]. Ipak, valja napomenuti da svakim dodavanjem čvora problem treniranja poprima veću dimenzionalnost te postaje složeniji i zbog spomenutih interakcija, što podrazumijeva upotrebu velikog broja vrednovanja funkcije cilja. Stoga je moguće naslutiti da inicijalne postavke parametara nadodanog čvora mogu biti od iznimne važnosti za daljnji tijek treniranja cijele mreže jer bi prikladne postavke njegovih parametara mogle pospješiti konvergenciju korištenog algoritma optimizacije te samim time i umanjiti potreban broj vrednovanja funkcije cilja. U [249] centar novog čvora se inicijalno postavlja u poziciju primjerka s najvećim SE (a njegova širina na vrijednost 1), što se intuitivno čini kao primjeren odabir jer je primarni cilj treniranja smanjiti SSE mreže. Međutim, u pogledu klasifikacije, ovakav primjerak može predstavljati šum ili se pak nalaziti u području preklapanja klasa, što je posebice izgledno za manjinske primjerke u neuravnoteženim skupovima podataka. Stoga je moguće pretpostaviti da će djelovanje nadodanog čvora opet biti nepovoljno za taj primjerak, odnosno da će i dalje biti pogrešno klasificiran s obzirom na svoju okolinu. Osim toga, ovaj način odabira parametara novog čvora ne uzima u obzir njegovu interakciju s prethodno definiranim čvorovima, koja, ako je nepovoljna, može otežati pronalazak mreže zadovoljavajućih performansi.

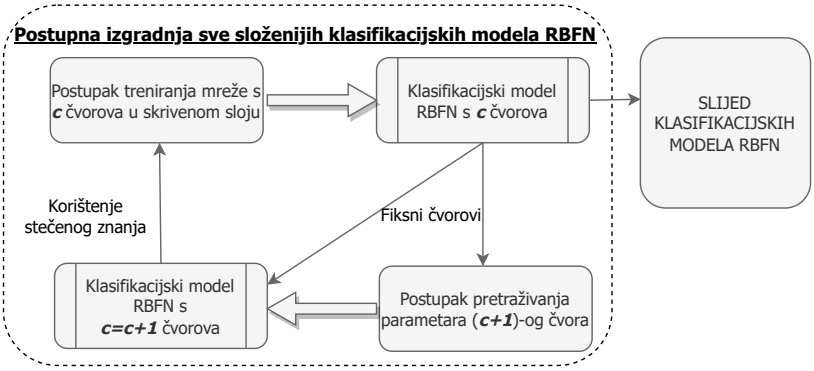
Iz predstavljenog je moguće zaključiti kako traženje prikladnog broja čvorova skrivenog sloja i vrijednosti parametara koji opisuju mrežu predstavlja iznimno zahtjevan problem globalne optimizacije. S obzirom na postojanje složenih interakcija između čvorova skrivenog sloja te veliku dimenzionalnost problema, nije iznenađujuća popularnost bio-inspiriranih algoritama za potrebe izgradnje klasifikacijskih modela RBFN. Poznato je da ovi algoritmi općenito zahtijevaju velik broj vrednovanja za pronalaženje kvalitetnih rješenja jer se oslanjaju isključivo na vrijednosti funkcije cilja za usmjeravanje tijeka pretrage. Osim toga, dimenzionalnost problema dodatno raste kodiranjem broja čvorova u obliku aktivacijskih zastavica u vektore rješenja što za sobom izravno povlači poveće računalne zahtjeve. Dodatno,

potrebno je uvesti i član kazne (koji je uobičajeno potrebno parametrizirati) u funkciju cilja pretrage kako bi se izbjegla prekomjerna složenost klasifikacijskog modela koja može dovesti do prenaučenosti klasifikatora. Za razliku od ranije spomenutih automatskih postupaka izgradnje, ovi postupci vraćaju samo jednu mrežu nakon završetka pretrage koja minimizira navedenu funkciju cilja. Unatoč utrošku velikog broja vrednovanja funkcije cilja, moguće je da pronađena mreža nema dobru sposobnost generalizacije, no uvid u ponašanje mreža različitih složenosti nije omogućen ovim postupkom.

Dakako, kao alternativu automatskom načinu izgradnje, velik broj postupaka iz literature predlaže jednostavno treniranje više mreža različitih složenosti (s unaprijed zadanim brojem čvorova u skrivenom sloju) te njihovo naknadno vrednovanje na skupu za testiranje kako bi se odabrala ona s najboljom sposobnosti generalizacije. Pri tome, prva faza treniranja najčešće se provodi upotrebom algoritama za grupiranje podataka ili bio-inspiriranih algoritama. Unatoč brzini i jednostavnosti algoritama za grupiranje podataka, mreže trenirane tim postupcima obično nisu dostatne kvalitete, s obzirom na činjenicu da ne uzimaju u obzir performanse mreže te oznake klasa primjeraka pri njihovu grupiranju. Štoviše, intuitivno se može naslutiti da treniranje mreže ovim postupcima može biti posebice nepovoljno za izvedbu klasifikatora na manjinskoj klasi jer je ona često podijeljena u podkoncepte te su njezini primjerci primarno okruženi primjercima iz većinske klase. S druge strane, iako se treniranjem pomoću bio-inspiriranih algoritama obično pronalaze kvalitetnije mreže, ovi algoritmi se oslanjaju na izvođenje velikog broja vrednovanja funkcije cilja, što je uglavnom računalno zahtjevno, kao što je i ranije spomenuto. Treba napomenuti da su kod ovih postupaka izgradnje RBFN iz literature koraci treniranja mreža različitih složenosti međusobno neovisni te da se pri treniranju mreža veće složenosti ne iskorištava znanje stečeno treniranjem mreža manje složenosti. Stoga je moguće zaključiti da se na ovaj način zahtijeva suvišno puno računalnog vremena za izgradnju klasifikacijskih modela RBFN.

5.3 Prijedlog postupka izgradnje klasifikacijskih modela RBFN

Glavni ciljevi prilikom izgradnje klasifikacijskih modela RBFN jesu postizanje visoke razine generalizacije klasifikatora te održavanje relativno jednostavne složenosti modela u smislu broja čvorova u skrivenom sloju. S obzirom na činjenicu da se radi o složenom multimodalnom problemu optimizacije, smanjivanje vremena potrebnog za pronalazak ovakvih mreža može biti od praktične važnosti. S tim ciljem, kao izvorni znanstveni doprinos predlaže se novi postupak izgradnje klasifikacijskih modela RBFN koji se temelji na postupnom treniranju mreža sve većeg stupnja složenosti, ali uz korištenje znanja iz prethodno treniranih mreža manje složenosti. U literaturi je predloženo nekoliko različitih postupaka koji grade strukturu RBFN postupnim povećanjem složenosti prethodnih mreža, kao što je ranije izlo-



Slika 5.4: Shema rada predloženog postupka izgradnje klasifikacijskih modela RBFN

ženo. Međutim, ovi postupci nisu izvorno namijenjeni za potrebe klasifikacije. Osim toga, izuzev postupka predloženog u [249], ne provode treniranje cijele mreže nakon dodavanja novog čvora u skriveni sloj, što je pak preporučljivo zbog postojanja složenih interakcija između tih čvorova. U odnosu na postupak izložen u [249], predloženi postupak izgradnje ulaže značajan napor u traženje prikladnog čvora koji se nadodaje (u smislu njegovih interakcija s postojećim čvorovima) kako bi se olakšalo i ubrzalo treniranje cijele mreže. Način dodavanja ovog čvora predstavlja primarnu sastavnicu doprinosa predloženog postupka koja ga izdvaja od ostalih postupaka izgradnje RBFN u literaturi. Uzastopnim ponavljanjem koraka treniranja mreža te dodavanja novog čvora, predloženi postupak izgradnje u konačnici pronalazi više mreža različitih složenosti što omogućava odabir one s najboljom sposobnosti generalizacije.

5.3.1 Opis predloženog postupka

Predloženi postupak nastoji izgraditi slijed klasifikacijskih modela RBFN povećane složenosti koji ostvaruju zadovoljavajuću uspješnost klasifikacije. Kao i ostali postupci izgradnje RBFN iz literature, također traži mreže unutar zadanih granica u smislu broja čvorova u skrivenom sloju. Pri tome, početno se trenira mreža s c_{min} čvorova u skrivenom sloju te se njezina struktura prenosi u naredni korak treniranja mreže veće složenosti, a ovaj postupak prenošenja se ponavlja sve dok nije trenirana mreža s c_{max} čvorova. Općenito gledajući, struktura trenirane mreže s c čvorova u skrivenom sloju se koristi pri treniranju mreže s $c+1$ čvorem. Očekuje se da će korištenje strukture prethodno trenirane mreže značajno pospješiti pronalaženje mreže veće složenosti u smislu kvalitete i potrebnog vremena. Pri tome, u prethodno treniranu mrežu potrebno je naknadno dodati novi čvor kako bi se ona mogla koristiti pri treniranju mreže veće složenosti. U sklopu predloženog postupka, dodavanje novog čvora predstavlja problem optimizacije u kojem je cilj pronaći čvor koji će u interakciji s preostalim čvorovima prethodno trenirane mreže rezultirati minimalnom greškom mreže povećane složenosti. Iako se potom provodi treniranje cijele mreže, prethodno trenirana mreža s optimiziranim nadodanim čvorom može poslužiti kao kvalitetno početno rješenje

Algoritam 5.1: Prijedlog postupka izgradnje klasifikacijskih modela RBFN na visokoj razini

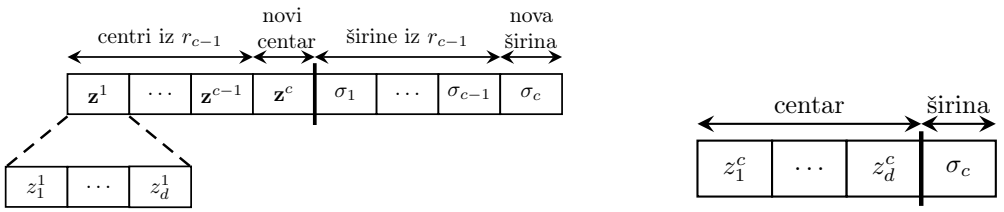
Izdvoji skup označenih primjeraka za treniranje $\mathcal{T}$;
Postavi c_{min} i c_{max};
Odaberi algoritam A_1 za treniranje mreže s populacijom P veličine N_P;
Definiraj skup treniranih mreža $\mathcal{R} = \emptyset$;
$c := c_{min}$;
Inicijaliziraj populaciju $\mathcal{P}_c = (\mathbf{p}^j \in \mathbb{R}^{c\cdot(d+1)} : j = 1, \ldots, N_P)$;
Pronađi $\mathbf{r}_c$ pomoću $\mathsf{A}_1(P_c)$;
$\mathcal{R} := \mathcal{R} \cup \mathbf{r}_c$;
za $c := c_{min} + 1, \ldots, c_{max}$ **čini**
  Pronađi rješenje ρ_c pomoću algoritma 5.2;
  Inicijaliziraj populaciju $\mathcal{P}_c = (\mathbf{p}^j \in \mathbb{R}^{c\cdot(d+1)} : j = 1, \ldots, N_P)$, uz $\mathbf{p}^1 = \rho_c$;
  Pronađi $\mathbf{r}_c$ pomoću $\mathsf{A}_1(P_c)$;
  $\mathcal{R} := \mathcal{R} \cup \mathbf{r}_c$;
kraj za

Algoritam 5.2: Prijedlog postupka dodavanja novog čvora u prethodno treniranu mrežu na visokoj razini

Odaberi algoritam A_2 za traženje parametara c-tog čvora s populacijom Q veličine N_Q;
Pronađi primjerak $\mathbf{x}^e$ za koji prethodno trenirana mreža daje najveći SE;
Izračunaj širinu σ_e prema (5.12);
Stvori rješenje $\theta \in \mathbb{R}^{d+1}$ iz $\mathbf{x}^e$ i σ_e;
Inicijaliziraj populaciju $Q = (\mathbf{q}^j \in \mathbb{R}^{d+1} : j = 1, \ldots, N_Q)$, uz $\mathbf{q}^1 = \theta$;
Pronađi $\mathbf{q}_c$ pomoću $\mathsf{A}_2(Q)$;
Stvori rješenje $\rho_c \in \mathbb{R}^{c\cdot(d+1)}$ iz $\mathbf{r}_{c-1}$ i $\mathbf{q}_c$;

na temelju kojeg algoritam za treniranje može lakše pronaći mrežu s povoljnom izvedbom klasifikacije. S druge strane, dodavanje čvora koji ima nepovoljne interakcije s preostalim čvorovima može otežati pronalazak mreže zadovoljavajućih performansi, kao što je ranije pojašnjeno. Tako se u okviru predloženog postupka te interakcije uzimaju u obzir, što nije slučaj u ostalim postupcima izgradnje RBFN u literaturi. Način rada predloženog postupka izgradnje na visokoj razini ilustriran je slikom 5.4.

Razni algoritmi optimizacije mogu se koristiti za treniranje mreža, pri čemu se bio-inspirirani algoritmi ističu kao jedni od najpogodnijih zbog svojih dobrih performansi. Bez smanjenja općenitosti, predloženi postupak izgradnje je stoga opisan s idejom upotrebe ovih algoritama za treniranje klasifikacijskih modela RBFN. Prijedlog postupka izgradnje predstavljen je na visokoj razini algoritmom 5.1. Budući da navedeni algoritmi obično koriste populaciju rješenja, predviđeno je da se strukture prethodno treniranih mreža prenose upravo kroz njihovu populaciju. Pri tome, rješenja u populaciji odabranog algoritma za treniranje (koji je u nastavku označen s A_1) su predstavljena vektorima u koje su kodirani parametri čvorova skrivenog sloja (kao što je ranije prikazano na slici 5.2b). Uz to, pretpostavlja se da se težine veza čvorova skrivenog i izlaznog sloja određuju metodom najmanjih kvadrata. Cilj algoritma za treniranje A_1 jest pronaći mrežu koja minimizira funkciju cilja predstavljenu mjerom MSE [dana s (5.9)]. Za treniranje mreže s c čvorova u skrivenom sloju, algoritam A_1 koristi populaciju rješenja $P_c = (\mathbf{p}^j \in \mathbb{R}^{c\cdot(d+1)} : j = 1, \ldots, N_P)$ veličine N_P, pri čemu je struktura prethodno trenirane mreže uvrštena u prvo rješenje početne populacije. Konkretno, $\mathbf{p}^1$ sadrži sve komponente najboljeg rješenja $\mathbf{r}_{c-1}$ pronađenog treniranjem mreže s



(a) Prvo rješenje početne populacije A_1 (b) Opće rješenje u populaciji algoritma A_2

Slika 5.5: Načini predstavljanja rješenja u populacijama algoritama A_1 i A_2

$c-1$ čvorova u skrivenom sloju te dodatno komponente koje predstavljaju parametre novodanog (c-tog) čvora, kao što je ilustrirano slikom 5.5a. Određivanje parametara novododanog čvora istaknut je korak predloženog postupka kojemu se ne pridaje posebna pažnja u ostalim postupcima izgradnje u literaturi, iako može imati značajan utjecaj na tijek treniranja, posebice u slučaju problema klasifikacije.

Prijedlog postupka dodavanja novog čvora u prethodno treniranu mrežu predstavljen je na visokoj razini algoritmom 5.2. Kao što je ranije rečeno, predviđeno je da se parametri ovog čvora traže odgovarajućim algoritmom optimizacije (u nastavku označen s A_2). S obzirom na relativno malu složenost problema traženja jednog čvora, algoritmi lokalne pretrage predstavljaju prikladan izbor za ovu namjenu (primjerice, Nelder-Mead [251]). Opis primjene algoritma A_2 također se temelji na pretpostavci da on koristi populaciju rješenja, iako to ne predstavlja ograničavajući čimbenik pri odabiru ovog algoritma. Pri tome, algoritam A_2 nastoji pronaći čvor koji će imati povoljne interakcije s čvorovima prethodno trenirane mreže. U vektore rješenja su tako kodirani samo parametri tog čvora, kao što pokazuje slika 5.5b. Cilj algoritma A_2 jest minimizirati iznos mjere MSE za mrežu s c čvorova u skrivenom sloju, pri čemu su parametri $c-1$ čvorova definirani u strukturi prethodno trenirane mreže te se drže fiksnima. Nadalje, bitno je napomenuti kako algoritam A_2 također koristi ranije stečeno znanje o problemu treniranja u određenom obliku. Naime, uz pretpostavku da ovaj algoritam koristi populaciju rješenja $Q = (\mathbf{q}^j \in \mathbb{R}^{d+1} : j = 1, \ldots, N_Q)$ veličine N_Q, prvo rješenje njegove početne populacije $\mathbf{q}^1$ je formirano na temelju ponašanja prethodno trenirane mreže. Preciznije, centar radijalne funkcije u $\mathbf{q}^1$ je postavljen u poziciju primjerka $\mathbf{x}^e \in \mathcal{X}$, odnosno primjerka za koji ta mreža daje najveći SE. Nadalje, širina radijalne funkcije u $\mathbf{q}^1$ računa se kao

$$\sigma_e = \tau \cdot u_{min,c}, \tag{5.12}$$

gdje je $\tau = 1.2$, a $u_{min,c} = \min\{\|\mathbf{x}^e - \mathbf{z}^j\| : j = 1, \ldots, c-1\}$, pri čemu su centri $\mathbf{z}^j$ sadržani u rješenju $\mathbf{r}_{c-1}$. Kao što je ranije spomenuto, ova jednostavna heuristika za određivanje širine radijalne funkcije se pokazala povoljnom u brojnim postupcima treniranja RBFN u literaturi [222, 244] te je stoga i ovdje preuzeta. Ovo rješenje se formira kako bi se ubrzalo pronalaženje prikladnog novog čvora koji se dodaje u prethodno treniranu mrežu, što će zauzvrat olak-

šati algoritmu A_1 pronalaženje mreže sa zadovoljavajućom izvedbom klasifikacije. Međutim, također je moguće da novododani čvor definiran na ovaj način nema povoljne interakcije s preostalim čvorovima skrivenog sloja iz prethodno trenirane mreže. Stoga je izvođenjem algoritma A_2 omogućeno traženje prikladnijih postavki njegovih parametara. Nakon završetka pretrage algoritma A_2, komponente najboljeg pronađenog rješenja uvrštavaju se na pozicije parametara c-tog čvora u rješenju $\mathbf{p}^1$ u populaciji P_c algoritma A_1, kao što je ilustrirano slikom 5.5a. Na temelju ovog kvalitetnog rješenja, algoritam A_1 može usmjeriti pretragu prema mrežama s povoljnom izvedbom klasifikacije. Pri tome, pretpostavlja se da će utrošak računalnih resursa potrebnih za izvođenje algoritma A_2 značajnije pomoći algoritmu A_1 u postizanju navedenog cilja nego da mu se ti resursi pridodaju za treniranje koje je usmjereno oko prethodno trenirane mreže s novododanim čvorom definiranim nasumično ili pomoću jednostavnih heuristika (primjerice, fiksnim postavljanjem centra radijalne funkcije u primjerak s najvećim SE kao u [249]).

Vidljivo je da predloženi postupak u konačnici daje na raspolaganje $c_{max} - c_{min} + 1$ treniranih klasifikacijskih modela RBFN uzastopnim ponavljanjem koraka treniranja mreža te dodavanja novog čvora. S obzirom na činjenicu da umjesto jedne mreže, pronalazi slijed mreža različitih složenosti, ne može se smatrati potpuno automatskim postupkom izgradnje RBFN. Ipak, naknadno vrednovanje ovih mreža na skupu za ispitivanje nije vremenski zahtjevno, a omogućava odabir mreže s najboljom sposobnosti generalizacije. S druge strane, u brojnim automatskim postupcima izgradnje u literaturi pronađena mreža ne mora imati dobra svojstva na skupu za ispitivanje, a uvid u ponašanje mreža različitih složenosti nije moguć.

5.3.2 Detalji ugradnje

Djelovanje svih koraka predloženog postupka izgradnje moguće je jasno interpretirati iz ranije izloženog opisa, no postoji nekoliko detalja ugradnje koje valja dodatno protumačiti. Detaljniji prikaz opisanog načina rada predloženog postupka izgradnje dan je algoritmima 5.3 i 5.4 u kojima je detaljnije izložen način formiranja prvih rješenja početnih populacija algoritama A_1 i A_2. Nadalje, valja istaknuti da predloženi postupak, kao i ostali postupci izgradnje RBFN iz literature, također ima dva parametra koje je potrebno postaviti prije njegova provođenja, odnosno c_{min} i c_{max} koji redom predstavljaju najmanji i najveći dozvoljeni broj čvorova u skrivenom sloju. Pri tome, maksimalan broj vrednovanja funkcije cilja $\mathrm{NFEs}_{\max}$ koji je omogućen predloženom postupku izgradnje dijeli se za potrebe treniranja svih $c_{max} - c_{min} + 1$ mreža. Način raspodjele ovih vrednovanja na pojedine korake treniranja nije propisan u sklopu predloženog postupka te ga je moguće pravodobno definirati u skladu s ostalim postavkama. Dakako, on uvelike ovisi o izboru odgovarajućih algoritama A_1 i A_2 te o postavkama parametara c_{min} i c_{max}, pri čemu je najjednostavniji način odrediti da se $\mathrm{NFEs}_{\max}$ vrednovanja ravnomjerno dijele za potrebe treniranja svih mreža. Nadalje, dobi-

Algoritam 5.3: Nacrt rada predloženog postupka izgradnje klasifikacijskih modela RBFN

Funkcija `Izgradi()`:

Ulaz: $T[N_T][d]$ `// Primjerci za treniranje`
$I[N_T]$ `// Oznake primjeraka za treniranje`
c_{min} `// Najmanji dozvoljeni broj čvorova u skrivenom sloju`
c_{max} `// Najveći dozvoljeni broj čvorova u skrivenom sloju`
A_1 `// Odabrani algoritam optimizacije`
N_P `// Veličina populacije algoritma` A_1
Izlaz: $R[c_{max} - c_{min} + 1][]$ `// Trenirane mreže`

$R[c_{max} - c_{min} + 1][] := []$
$c := c_{min}$
Inicijaliziraj populaciju $P_c[N_P][d]$
$r_c := \mathsf{A}_1(P_c)$
Ubaci r_c u R;
za $c := c_{min} + 1, ..., c_{max}$ **čini**
  $\rho_c :=$ Nadogradi(T, I, r_{c-1})
  Inicijaliziraj populaciju $P_c[N_P][d]$
  $P_c[1] := \rho_c$
  $r_c := \mathsf{A}_1(P_c)$
  Ubaci r_c u R;
kraj za

Algoritam 5.4: Nacrt rada predloženog postupka za dodavanje čvora

Funkcija `Nadogradi()`:

Ulaz: $T[N_T][d]$ `// Primjerci za treniranje`
$I[N_T]$ `// Oznake primjeraka za treniranje`
$r_{c-1}[c-1][d+1]$ `// Prethodno trenirana mreža s` $c-1$ `čvorova u skrivenom sloju`
A_2 `// Odabrani algoritam optimizacije`
N_Q `// Veličina populacije algoritma` A_2
Izlaz: $\rho_c[c \cdot (d+1)]$ `// Prethodno trenirana mreža nadograđena s` c`-tim čvorom`

$\rho_c[1 : (c-1) \cdot d] := r_{c-1}[1 : (c-1) \cdot d]$ `// Pridruživanje centara iz` r_{c-1}
$\rho_c[c \cdot d + 1 : c \cdot (d+1) - 1] := r_{c-1}[(c-1) \cdot d + 1 : (c-1) \cdot (d+1)]$ `// Pridruživanje širina iz` r_{c-1}
Pronađi primjerak x^e za koji prethodno trenirana mreža daje najveći SE
Izračunaj širinu σ_e prema (5.12)
$\theta[1 : d] := x^e$
$\theta[d+1] := \sigma_e$
Inicijaliziraj populaciju $Q[N_Q][d]$
$Q[1] := \theta$
$q_c := \mathsf{A}_2(Q)$
$\rho_c[(c-1) \cdot d + 1 : c \cdot d] := q_c[1 : d]$ `// Pridruživanje centra iz` q_c
$\rho_c[c \cdot (d+1)] := q_c[d+1]$ `// Pridruživanje širine iz` q_c

vena vrednovanja za treniranje pojedine mreže dijele se između algoritama A_1 i A_2. Način raspodjele ovih vrednovanja također je prepušten naknadnom utvrđivanju u skladu s ostalim postavkama. Međutim, treba obratiti pozornost na značajne razlike u dimenzionalnosti prostora koje ovi algoritmi pretražuju, pa se stoga može pretpostaviti da je algoritmu A_2 potreban značajno manji broj vrednovanja za traženje parametara jednog čvora u odnosu na broj vrednovanja potreban algoritmu A_1 za treniranje cijele mreže. Kako bi se učinkovitije rukovalo raspodjelom vrednovanja između ovih algoritama, u algoritam A_2 je moguće ugraditi mehanizam ranog zaustavljanja u smislu neznatnih razlika u pronađenim rješenjima ili njihovim kvalitetama, pri čemu se preostali broj vrednovanja može pridodati algoritmu A_1.

5.3.3 Procjena vremenske složenosti

Kao što je ranije navedeno, u sklopu predloženog postupka izgradnje trenira se ukupno $c_{max} - c_{min} + 1$ klasifikacijskih modela RBFN različitih složenosti. U tom pogledu, vremenska složenost svakog postupka treniranja može se razložiti na pojedinačne složenosti algoritma A_1 i funkcije *Nadogradi()* koja se koristi za dodavanje novog čvora u prethodno treniranu mrežu. Prema tome, svaki korak treniranja ima vremensku složenost $O(\mathsf{A}_1) + O(\mathsf{A}_2) + O(N)$, gdje je N broj primjeraka u skupu za treniranje. Vremenske složenosti odabranih algoritama A_1 i A_2 ovise o njihovu načinu izvođenja pretrage, no važno je naglasiti da su vrednovanja funkcije cilja njihove najdugotrajnije operacije. Osim izvođenja ovih algoritama, pri svakom treniranju traži se primjerak iz skupa podataka za koji prethodno trenirana mreža daje najveći SE te se računa širina radijalne funkcije s tim primjerkom kao centrom $[O(N + d)]$.

S obzirom na to da postoji nekoliko postupaka izgradnje klasifikacijskih modela RBFN u literaturi koji se znatno razlikuju prema načinu rada, teško ih je sve usporediti s predloženim postupkom u smislu vremenske složenosti. U odnosu na postupke izgradnje koji također treniraju više mreža različitih složenosti (pri čemu su koraci treniranja međusobno neovisni), predloženi postupak dodatno izvodi tek funkciju *Nadogradi()* neposredno prije treniranja svake mreže. Štoviše, uz pretpostavke da ovi postupci izgradnje treniraju mreže pomoću istog algoritma A_1 te da se vremenske složenosti algoritama A_1 i A_2 ne razlikuju značajno, razlika u njihovoj vremenskoj složenosti u odnosu na predloženi postupak svodi se samo na složenost operacija traženja primjerka s najvećim SE i računanja odgovarajuće širine $[O(N+d)]$ koje u pravilu zahtijevaju neznatan vremenski trošak u odnosu na ostale operacije. Nadalje, uz valjanost ovih pretpostavki, moguće je primijetiti da predloženi postupak ima jednaku vremensku složenost kao i postupak predložen u [249] koji se također temelji na postupnom treniranju mreža sve većeg stupnja složenosti.

5.4 Eksperimentalna analiza i rezultati

Kako bi se utvrdila učinkovitost predloženog postupka izgradnje klasifikacijskih modela RBFN, provedena je odgovarajuća eksperimentalna analiza koja je podijeljena u dva dijela. Kao što je ranije spomenuto, primarni cilj predloženog postupka jest izgraditi slijed kvalitetnih klasifikacijskih modela RBFN povećane složenosti. Dodatan cilj jest pronaći ove mreže u manje vremena u odnosu na druge postupke izgradnje iz literature. Oba cilja se nastoje ostvariti korištenjem strukture prethodno trenirane mreže pri treniranju mreže veće složenosti, pri čemu način dodavanja novog čvora u prethodno treniranu mrežu predstavlja ključnu sastavnicu doprinosa predloženog postupka izgradnje. Stoga je u prvom dijelu analiziran predloženi način dodavanja ovog čvora kako bi se utvrdio njegov doprinos performansama predloženog postupka. U drugom dijelu eksperimentalne analize ostvarena je usporedba predloženog postupka s nekoliko različitih postupaka izgradnje klasifikacijskih mo-

Tablica 5.1: Karakteristike skupova podataka korištenih za potrebe eksperimentalne analize predloženog postupka izgradnje klasifikacijskih modela RBFN

Oznaka	Naziv	Broj primjeraka	Broj značajki	Broj klasa	IR
$\mathcal{D}_1$	QSAR biodegradation	1055	41	2	1.96
$\mathcal{D}_2$	Breast Cancer Wisconsin	569	30	2	1.68
$\mathcal{D}_3$	Clean2	6598	166	2	5.49
$\mathcal{D}_4$	Climate	540	18	2	10.74
$\mathcal{D}_5$	Glass Identification	214	9	6	3.59
$\mathcal{D}_6$	Heart Disease	270	13	2	1.25
$\mathcal{D}_7$	Hill-Valley	1212	100	2	1.00
$\mathcal{D}_8$	Ionosphere	351	34	2	1.79
$\mathcal{D}_9$	MuskV1	476	166	2	1.30
$\mathcal{D}_{10}$	Parkinsons	195	22	2	3.06
$\mathcal{D}_{11}$	Urban Land Cover	675	147	9	2.19
$\mathcal{D}_{12}$	Wine	178	13	3	1.30

dela RBFN iz literature, s ciljem stjecanja uvida u razlike klasifikacijskih modela izgrađenih ovim postupcima u smislu njihove kvalitete i veličine.

Eksperimentalna analiza provedena je na standardnim skupovima podataka koji se uobičajeno koriste za vrednovanje novopredloženih postupaka za izgradnju klasifikacijskih modela RBFN u literaturi, a njihove karakteristike prikazane su u tablici 5.1, dok je njihov opis dan u dodatku A. Ovi skupovi podataka preuzeti su s UCI repozitorija te, kao i u poglavljima 3 i 4, predstavljaju razne probleme klasifikacije koji se, osim po prirodi, razlikuju po dimenzionalnosti i stupnju neuravnoteženosti klasa. Uz to, tri odabrana skupa podataka ($\mathcal{D}_5$, $\mathcal{D}_{11}$ i $\mathcal{D}_{12}$) predstavljaju višeklasne probleme kako bi se ispitala učinkovitost predloženog postupka izgradnje i na takvim problemima. Skaliranje značajki, normalizacijom u raspon $[0, 1]$, izvedeno je kao korak predobrade svakog skupa podataka, jednako kao i u poglavljima 3 i 4.

5.4.1 Postavke eksperimenta

Kako bi se analizirao utjecaj predloženog načina dodavanja novog čvora u prethodno treniranu mrežu, u prvom dijelu predloženi postupak izgradnje uspoređen je s dva postupka izgradnje koji na isti način postupno treniraju mreže sve većeg stupnja složenosti, ali se razlikuju u načinu dodavanja ovog čvora. U prvom postupku (označenom s $\mathrm{I_R}$), centar novododanog čvora postavlja se u poziciju nasumično odabranog primjerka iz skupa podataka, dok se širina postavlja također nasumično unutar $[0, 1]$. U drugom postupku (označenom s $\mathrm{I_E}$), centar novododanog čvora postavlja se u poziciju primjerka za koji prethodno trenirana mreža daje najveću grešku (SE), a njegova širina na vrijednost 1. Bitno je naglasiti da je potonji postupak izgradnje istovjetan postupku predloženom u [249], koji je ranije opisan. Prema tome, oba postupka koriste jednostavne heuristike za određivanje parametara novododanog čvora te se tako definirana mreža povećane složenosti daje algoritmu za treniranje cijele mreže. S obzirom na to da je u prvom dijelu eksperimentalne analize naglasak na ispitivanju utjecaja načina dodavanja novog čvora, u sva tri postupka korišten je isti algoritam za

Tablica 5.2: Postavke predloženog postupaka za izgradnju klasifikacijskih modela RBFN korištene za potrebe eksperimentalne analize

Oznaka	Algoritam za treniranje		Algoritam za traženje novododanog čvora	
	Naziv	Parametri	Naziv	Parametri
Predloženi	PSO	$N_P = 30$ $w = 0.724$ $c_1 = c_2 = 1.468$ $\text{NFEs}_{\max} = 0.9 \cdot 10^4$	Nelder-Mead	$\gamma^s = \delta^{oc} = 0.5$ $\delta^{ic} = -0.5$ $\delta^r = 1$ $\delta^e = 2$ $\text{NFEs}_{\max} = 0.1 \cdot 10^4$

Tablica 5.3: Postavke postupaka za izgradnju klasifikacijskih modela RBFN korištenih za potrebe prvog dijela eksperimentalne analize

Oznaka	Opis	Algoritam za treniranje	Parametri
I_R	Centar novododanog čvora se postavlja u poziciju nasumično odabranog primjerka iz skupa podataka, dok se širina određuje unutar $[0, 1]$.	PSO	$N_P = 30$ $w = 0.724$ $c_1 = c_2 = 1.468$ $\text{NFEs}_{\max} = 10^4$
I_E [249]	Centar novododanog čvora se postavlja u poziciju primjerka za koji prethodno trenirana mreža daje najveći SE, a njegova širina na vrijednost 1.		

treniranje mreža kako bi se otklonio njegov utjecaj na razlike u performansama uspoređenih postupaka. Konkretno, odabran je algoritam PSO [252] koji se mnoštvo puta u literaturi iskazao kao prikladan algoritam za treniranje klasifikacijskih modela RBFN [230, 231, 241]. Postavke parametara ovog algoritma preuzete su iz [253]. U skladu s ranije izloženim opisom predloženog postupka izgradnje, za funkciju cilja algoritma PSO korištena je mjera MSE, a način predstavljanja rješenja bio je istovjetan onome prikazanom na slici 5.2a. Prije provođenja ovih postupaka potrebno je odrediti najmanji i najveći dozvoljeni broj čvorova u skrivenom sloju. U tom pogledu, za donju granicu odabrano je $c_{min} = 2$, a $c_{max} = 20$ za gornju granicu te je za treniranje svake pojedine mreže bilo dozvoljeno 10^4 vrednovanja funkcije cilja ($\text{NFEs}_{\max}$). Nadalje, za traženje novododanog čvora u predloženom postupku izgradnje korišten je algoritam Nelder-Mead [251] zbog njegove jednostavnosti te činjenice da se svi čvorovi naknadno korigiraju pomoću algoritma PSO. Postavke parametara ovog algoritma preuzete su iz [254]. Pri tome, 10% vrednovanja dozvoljenih za treniranje svake pojedine mreže dano je algoritmu Nelder-Mead za traženje novog čvora. Uz to, u ovaj algoritam je ugrađen mehanizam ranog zaustavljanja u slučaju da je razlika u kvalitetama novog najboljeg i prethodnog najboljeg rješenja manja od 10^{-4}, pri čemu se preostali broj vrednovanja pridodaje algoritmu PSO za treniranje cijele mreže. Korištene postavke parametara uspoređenih postupaka izgradnje prikazane su u tablicama 5.2 i 5.3.

U drugom dijelu eksperimentalne analize, predloženi postupak uspoređen je s nekolicinom postupaka izgradnje klasifikacijskih modela RBFN iz literature koji su navedeni u tablici 5.4. Za usporedbu su odabrana dva jednostavna postupka izgradnje koji treniraju više mreža različitih složenosti, pri čemu su koraci treniranja međusobno neovisni. Prvi postupak (označen s J_PSO) provodi treniranje upotrebom algoritma PSO, dok drugi (označen s $\text{J}_{k\text{-means}}$) rabi algoritam k-means za određivanje centara radijalnih funkcija, pri čemu se širine

Tablica 5.4: Postavke postupaka za izgradnju klasifikacijskih modela RBFN korištenih za potrebe drugog dijela eksperimentalne analize

Oznaka	Opis	Algoritam za treniranje	Parametri
$\mathrm{J_{PSO}}$	Trenira više mreža različitih složenosti pomoću algoritma PSO, pri čemu su koraci treniranja međusobno neovisni.	PSO	$N_P = 30$ $w = 0.724$ $c_1 = c_2 = 1.468$ $\mathrm{NFEs_{max}} = 2 \cdot 10^5$
$\mathrm{J}_{k\text{-means}}$	Trenira više mreža različitih složenosti pomoću algoritma k-means, pri čemu su koraci treniranja međusobno neovisni.	k-means	$t_{max} = 50$ $\epsilon = 0.01$ $\mathrm{NFEs_{max}} = 2 \cdot 10^5$
$\mathrm{A_{DE}}$ [215]	Automatski postupak izgradnje RBFN zasnovan na unaprijeđenom algoritmu DE.	DE	$N_P = 30$ $CR = 0.9$ $F = 0.5$ $\lambda = \frac{1}{2m}0.01$ $\mathrm{NFEs_{max}} = 2 \cdot 10^5$
$\mathrm{A_{PSO}}$ [231]	Automatski postupak izgradnje RBFN zasnovan na unaprijeđenom algoritmu PSO.	PSO	$N_P = 30$ $w = 0.9 \rightarrow 0.4$ $c_1 = c_2 = 2$ $\lambda = 4$ $\mathrm{NFEs_{max}} = 2 \cdot 10^5$

računaju prema (5.11), kao što je predloženo u [222]. Kako bi se istaknula korisnost znanja iz prethodno treniranih mreža koje se odvija u predloženom postupku, ovim postupcima je za treniranje svake pojedine mreže dan veći broj vrednovanja funkcije cilja (konkretno, $2 \cdot 10^5$) nego što je omogućen predloženom postupku za treniranje svih mreža. Pri tome, oba postupka imaju istu funkciju cilja i način predstavljanja rješenja kao i predloženi postupak izgradnje. Osim toga, za $\mathrm{J_{PSO}}$ su korištene iste postavke parametara kao i za predloženi postupak, dok su za $\mathrm{J}_{k\text{-means}}$ preuzete postavke parametara iz [228]. Nadalje, predloženi postupak također je uspoređen s dva postupka automatske izgradnje RBFN iz literature, pri čemu je prvi (označen s $\mathrm{A_{DE}}$) zasnovan na algoritmu DE [215], a drugi (označen s $\mathrm{A_{PSO}}$) na algoritmu PSO [231]. Postavke parametara ovih algoritama su preuzete iz radova u kojima su ovi postupci predloženi [215, 231]. U oba algoritma su dodatno uključeni mehanizmi specifični za problem treniranja RBFN koji pospješuju njihovu učinkovitost, kao što je ranije objašnjeno. Ovim postupcima je za cjelokupno izvođenje dozvoljeno $2 \cdot 10^5$ vrednovanja funkcije cilja. Valja podsjetiti da automatski postupci izgradnje vraćaju samo jednu mrežu, pri čemu kodiraju broj čvorova skrivenog sloja u vektore rješenja te ugrađuju u funkciju cilja član kazne koji je proporcionalan broju čvorova skrivenog sloja ($\lambda \cdot c$) čime se nastoji ostvariti ravnoteža između sposobnosti generalizacije mreže i njezine složenosti. Za sve postupke izgradnje navedene u tablici 5.4 također je odabrano $c_{min} = 2$ za donju te $c_{max} = 20$ za gornju granicu broja čvorova u skrivenom sloju. Osim toga, u svim su postupcima težine veza čvorova skrivenog i izlaznog sloja estimirane metodom najmanjih kvadrata.

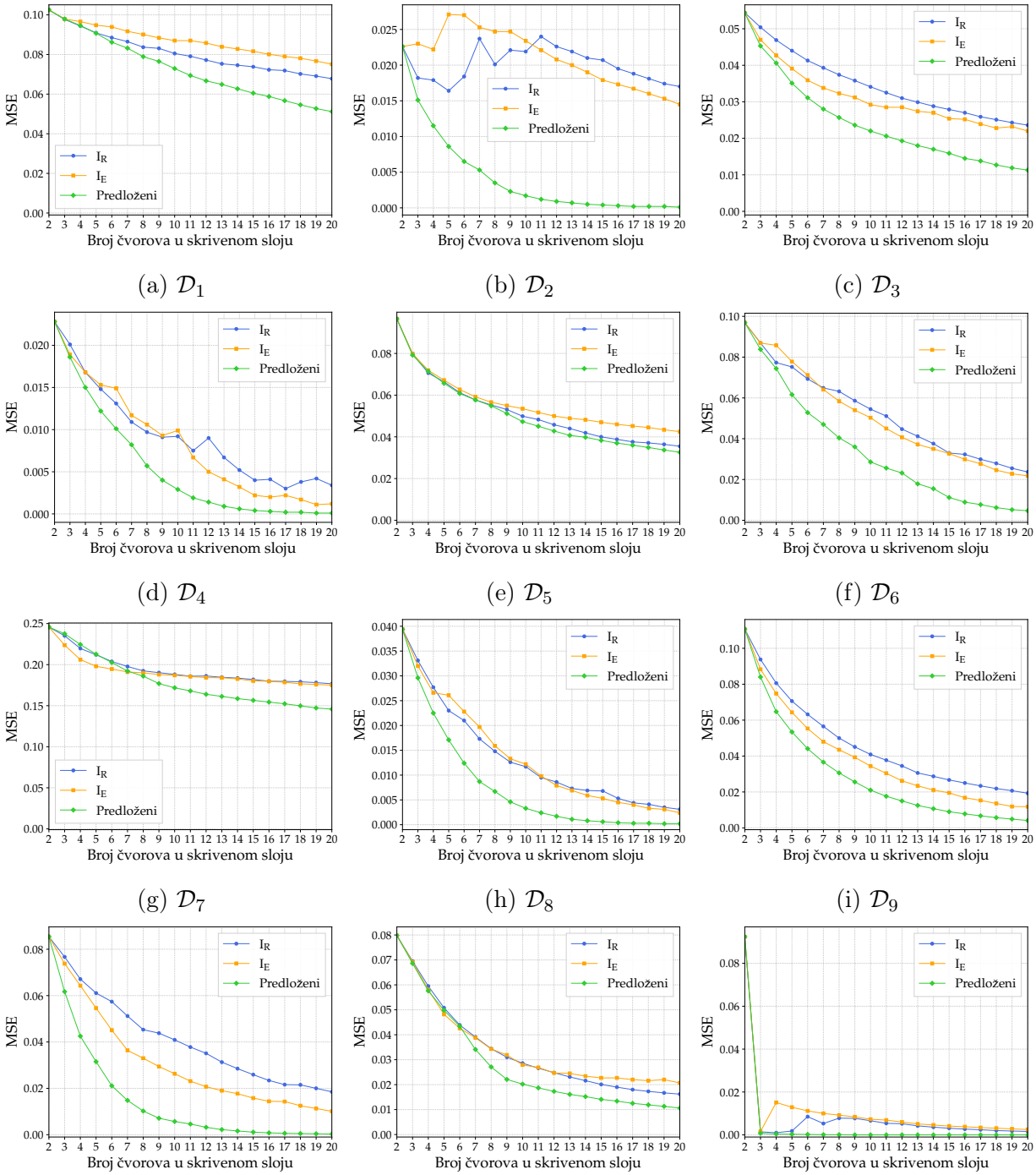
5.4.2 Metodologija eksperimentalne analize

Tijek eksperimentalne analize započinje podjelom skupova podataka. Iz svakog skupa podataka redom su stratificirano izdvojeni skupovi za treniranje i testiranje, u omjeru $0.75 : 0.25$. Da bi se stekao općenitiji uvid u performanse predloženog postupka izgradnje, napravljeno je 10 različitih podjela svakog skupa podatka, a postupak izgradnje je ponovljen tri puta za svaku podjelu kako bi se ublažio utjecaj stohastičke prirode algoritama korištenih za treniranje na dobivene rezultate, kao što je to napravljeno u poglavlju 3. Time automatski postupci izgradnje pronalaze 30 mreža za svaki skup podataka, dok preostali postupci izgradnje korišteni u eksperimentalnoj analizi pronalaze 30 mreža za svaku kombinaciju skupa podataka i broja čvorova u skrivenom sloju RBFN.

Nakon podjele skupova podataka te izvođenja odabranih postupaka izgradnje RBFN, provedena je usporedba njihovih performansi. Za sve postupke izgradnje prvotno su zabilježene kvalitete mreža pronađenih svakim korakom treniranja u smislu vrijednosti funkcije cilja, s ciljem analize njihova ponašanja tijekom izgradnje. Nadalje, iako većina uspoređenih postupaka izgradnje pronalazi slijed mreža različitih složenosti, u konačnici je poželjno odabrati jednu koja će predstavljati taj postupak. U pogledu klasifikacije, primarni cilj jest odabrati mrežu s najboljom sposobnosti generalizacije. Stoga je za svaku podjelu skupa podataka odabrana ona mreža koja daje najveći iznos mjere F1 na skupu za testiranje. Kako bi se sažeto prikazale performanse uspoređenih postupaka izgradnje, za svaki skup podataka su izračunate prosječne izvedbe mreža s najboljom sposobnosti generalizacije. S ciljem pojednostavljenja njihove usporedbe, izvedene su Euklidske udaljenosti (označene s d_{perf}) između izvedbe savršenog klasifikatora i točke čije koordinate čine prosječne izvedbe najboljih mreža na svakom skupu podataka, kao u poglavljima 3 i 4. Osim toga, usporedbom prosječnih izvedbi najboljih mreža na svim skupovima podataka pomoću Friedmanova testa ranga [156], izračunati su i prosječni rangovi (označeni s FR) razmatranih postupaka koji ukazuju na općenitu razliku u njihovim performansama. Uz to, za svaki skup podataka izračunata je i prosječna složenost najboljih mreža u smislu broja čvorova u skrivenom sloju.

5.4.3 Analiza utjecaja načina dodavanja čvora u predloženom postupku

Predloženim načinom dodavanja čvora prethodno treniranoj mreži nastoji se pronaći čvor koji će imati povoljne interakcije s postojećim čvorovima te mreže. Pri tome, očekuje se da će korištenje tako formirane mreže uvelike pomoći algoritmu za treniranje u pronalaženju kvalitetnih mreža. Kako bi se utvrdila djelotvornost ovakvog načina dodavanja novog čvora, predloženi postupak izgradnje uspoređen je s dva postupka izgradnje RBFN koji se od predloženog razlikuju samo po načinu provođenja ovog koraka. Oba postupka određuju parametre novog čvora pomoću jednostavnih heuristika, kao što je ranije objašnjeno.



(a) $\mathcal{D}_1$ (b) $\mathcal{D}_2$ (c) $\mathcal{D}_3$

(d) $\mathcal{D}_4$ (e) $\mathcal{D}_5$ (f) $\mathcal{D}_6$

(g) $\mathcal{D}_7$ (h) $\mathcal{D}_8$ (i) $\mathcal{D}_9$

(j) $\mathcal{D}_{10}$ (k) $\mathcal{D}_{11}$ (l) $\mathcal{D}_{12}$

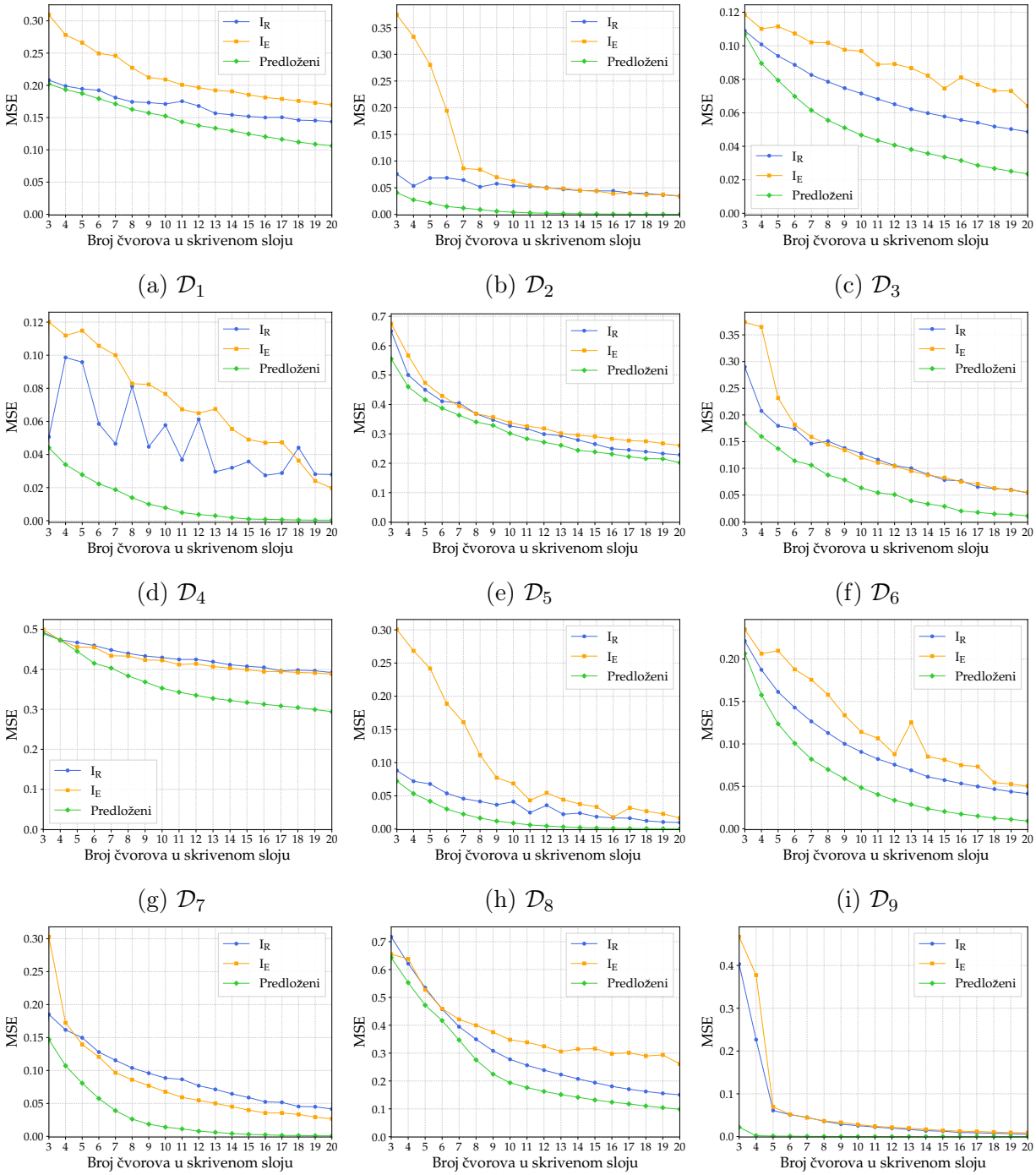
Slika 5.6: Kvalitete izgrađenih mreža na skupu za treniranje

Na slici 5.6 su za svaki skup podataka prikazani medijani izvođenja u smislu mjere MSE na skupu za treniranje redom za mreže rastućih složenosti. S obzirom na to da mjera MSE predstavlja funkciju cilja kojom je vođeno treniranje svake mreže, vidljivo je da predloženi postupak uspijeva pronaći kvalitetnija rješenja od onih pronađenih pomoću druga dva postupka izgradnje, gotovo za svaku kombinaciju skupa podataka i broja čvorova u skrivenom sloju mreža. S obzirom na činjenicu da se uspoređeni postupci razlikuju jedino u načinu

dodavanja čvora u prethodno treniranu mrežu, jasno je kako je to posljedica djelotvornosti predloženog načina izvođenja ovog koraka. Bitno je napomenuti da sva tri postupka izgradnje uvijek kreću od istih mreža s dva čvora u skrivenom sloju te se postupnim izvođenjem koraka dodavanja novog čvora i treniranja cijele mreže počinju razlikovati. S povećanjem broja čvorova prednost predloženog postupka postaje sve izraženija, što se može pokušati objasniti tvrdnjom da se razlike u kvalitetama treniranih mreža akumuliraju. Štoviše, predloženi postupak uspio je pronaći mreže koje daju grešku jednaku nuli gotovo na polovici skupova podataka ($\mathcal{D}_2$, $\mathcal{D}_4$, $\mathcal{D}_8$, $\mathcal{D}_{10}$ i $\mathcal{D}_{12}$), dok druga dva postupka to nisu postigla niti jednom.

Valja podsjetiti da predloženi postupak izgradnje troši dio vrednovanja funkcije cilja za traženje novododanog čvora, za razliku od postupaka $\mathrm{I_R}$ i $\mathrm{I_E}$. Međutim, sa slike 5.6 vidljivo je kako prethodno trenirana mreža s prikladnim nadodanim čvorom značajnije olakšava pronalazak kvalitetnih mreža u odnosu na povećanje broja vrednovanja za treniranje uz parametre novododanog čvora definirane pomoću jednostavnih heuristika iz literature. Štoviše, ubacivanjem novog čvora u prethodno treniranu mrežu pomoću ovih heuristika ponekad se pogorša kvaliteta te mreže, što se može primijetiti na slici 5.7 gdje su dani medijani kvaliteta treniranih mreža neposredno nakon dodavanja novog čvora. Ove mreže redom su uvrštavane u početnu populaciju korištenog algoritma za treniranje te su utjecale na njegovo ponašanje tijekom treniranja. Valja istaknuti kako na slici 5.7 nije primjetna razlika u kvalitetama mreža formiranih postupcima $\mathrm{I_R}$ i $\mathrm{I_E}$. Štoviše, na velikom broju skupova podataka $\mathrm{I_R}$ formira kvalitetnije mreže. Iz toga je moguće zaključiti da heuristika za definiranje novododanog čvora predložena u [249] ne rezultira značajnijim doprinosom performansama algoritma za treniranje na problemima klasifikacije u odnosu na heuristiku koja taj čvor određuje nasumično. No, ovaj rezultat ne bi trebao biti veliko iznenađenje s obzirom na činjenicu da je postupak izgradnje dan u [249] predložen za rješavanje problema aproksimacije funkcija koji se razlikuju od problema klasifikacije u nekoliko bitnih karakteristika, kao što je ranije pojašnjeno. S druge strane, predloženim načinom dodavanja novog čvora uvijek se poboljšava kvaliteta prethodno trenirane mreže na razmatranim problemima, što se također može primijetiti na slici 5.7.

Iz prikazanog je moguće zaključiti kako predloženi postupak izgradnje općenito pronalazi kvalitetnije mreže na skupu za treniranje u odnosu na postupke $\mathrm{I_R}$ i $\mathrm{I_E}$. Štoviše, na većini skupova podataka uspijeva pronaći nekoliko mreža različitih složenosti koje su kvalitetnije od najkvalitetnijih mreža pronađenih pomoću preostala dva postupka (koje su u pravilu i najveće složenosti). Stoga je moguće očekivati da barem neka od mreža izgrađenih predloženim postupkom ima i bolju sposobnost generalizacije. Kako bi se utvrdila njegova uspješnost u ostvarivanju ovog cilja, za svaku podjelu skupa podataka je zabilježen broj mreža izgrađenih predloženim postupkom koje daju veći iznos mjere F1 na skupu za testiranje od najbolje (također u smislu mjere F1 na skupu za testiranje) mreže pronađene postupkom $\mathrm{I_R}$, odnosno $\mathrm{I_E}$. Na slici 5.8 su prikazani prosječni brojevi takvih mreža za svaki skup podataka. Moguće



(a) $\mathcal{D}_1$ (b) $\mathcal{D}_2$ (c) $\mathcal{D}_3$

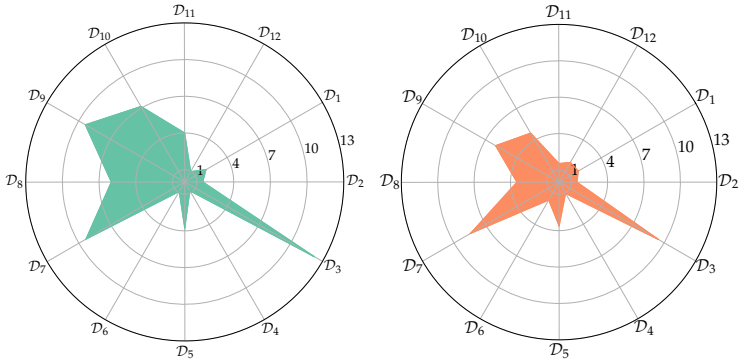
(d) $\mathcal{D}_4$ (e) $\mathcal{D}_5$ (f) $\mathcal{D}_6$

(g) $\mathcal{D}_7$ (h) $\mathcal{D}_8$ (i) $\mathcal{D}_9$

(j) $\mathcal{D}_{10}$ (k) $\mathcal{D}_{11}$ (l) $\mathcal{D}_{12}$

Slika 5.7: Kvalitete izgrađenih mreža na skupu za treniranje nakon dodavanja novog čvora

je primijetiti da je predloženi postupak na većini skupova podataka pronašao barem jednu mrežu s boljom sposobnosti generalizacije od obje najbolje mreže izgrađene postupcima $\mathrm{I_R}$ i $\mathrm{I_E}$, pri čemu je na nekolicini skupova podataka broj takvih mreža pozamašan (> 4). Kako bi se stekao uvid u razlike mreža izgrađenih uspoređenim postupcima, u tablici 5.5 su prikazane prosječne kvalitete i prosječne veličine za njihove tri najbolje mreže. Na dnu tablice su prikazani rangovi u smislu kvalitete te udaljenosti od savršenog klasifikatora, a najbolje



(a) U odnosu na I_R (b) U odnosu na I_E

Slika 5.8: Prosječan broj mreža s boljom sposobnosti generalizacije izgrađenih predloženim postupkom u odnosu na najbolje mreže izgrađene postupcima I_R i I_E

Tablica 5.5: Usporedba performansi predloženog postupka za izgradnju i performansi postupaka I_R i I_E

$\mathcal{D}$	1. najbolje mreže						2. najbolje mreže						3. najbolje mreže					
	I_R		I_E		Predloženi		I_R		I_E		Predloženi		I_R		I_E		Predloženi	
	F1	c	F1	c	F1	c	F1	c	F1	c	F1	c	F1	c	F1	c	F1	c
$\mathcal{D}_1$	0.86±0.02	8.83	0.86±0.02	11.57	0.86±0.02	**7.73**	0.85±0.02	10.23	0.85±0.02	8.80	0.85±0.02	**7.70**	0.85±0.02	11.90	0.85±0.02	9.97	0.85±0.02	**8.47**
$\mathcal{D}_2$	0.98±0.01	4.57	0.98±0.01	4.03	0.98±0.01	**3.97**	0.98±0.01	**4.53**	0.98±0.01	5.23	0.98±0.01	4.57	**0.98**±0.01	6.63	0.97±0.01	**6.23**	0.97±0.01	6.40
$\mathcal{D}_3$	0.94±0.01	17.19	0.95±0.01	**15.42**	**0.97**±0.01	16.65	0.93±0.01	17.73	0.94±0.01	**15.92**	**0.97**±0.01	16.73	0.93±0.01	16.69	0.94±0.01	**15.08**	**0.97**±0.01	16.77
$\mathcal{D}_4$	**0.87**±0.04	5.10	0.86±0.04	4.50	0.86±0.05	**3.57**	**0.85**±0.04	5.20	**0.85**±0.04	4.97	0.84±0.05	**4.17**	**0.84**±0.04	7.17	**0.84**±0.04	5.73	0.83±0.05	**5.07**
$\mathcal{D}_5$	0.64±0.06	**12.53**	0.64±0.05	12.77	**0.65**±0.07	13.90	0.63±0.06	**13.03**	0.62±0.06	13.50	**0.64**±0.07	14.60	0.62±0.06	13.73	0.62±0.06	**13.30**	**0.63**±0.07	15.47
$\mathcal{D}_6$	**0.84**±0.03	5.57	**0.84**±0.04	**4.63**	0.83±0.04	6.17	**0.83**±0.03	6.93	0.82±0.04	**4.57**	0.82±0.03	6.80	**0.82**±0.04	6.67	0.81±0.04	**5.90**	0.81±0.04	5.97
$\mathcal{D}_7$	0.75±0.03	11.37	0.75±0.03	**7.40**	**0.78**±0.04	15.03	0.74±0.03	11.20	0.75±0.02	**10.13**	**0.78**±0.04	14.63	0.74±0.03	12.57	0.74±0.02	**12.37**	**0.77**±0.05	15.10
$\mathcal{D}_8$	0.93±0.02	8.40	**0.95**±0.02	**7.87**	**0.95**±0.02	11.27	0.93±0.02	**7.27**	0.94±0.02	7.50	**0.95**±0.02	12.63	0.92±0.02	**6.50**	0.93±0.02	7.20	**0.94**±0.02	13.53
$\mathcal{D}_9$	0.90±0.03	**12.07**	0.91±0.03	13.63	**0.93**±0.03	12.57	0.89±0.03	14.20	0.91±0.03	12.97	**0.92**±0.03	**12.60**	0.89±0.03	**12.73**	0.90±0.03	12.93	**0.92**±0.03	12.80
$\mathcal{D}_{10}$	0.89±0.05	9.83	0.91±0.04	9.77	**0.92**±0.04	**9.63**	0.88±0.05	9.87	0.90±0.04	10.27	**0.91**±0.04	**9.77**	0.87±0.05	**10.17**	0.90±0.04	**10.17**	**0.91**±0.04	11.40
$\mathcal{D}_{11}$	0.82±0.03	16.20	**0.83**±0.03	15.43	**0.83**±0.03	**13.13**	0.81±0.03	16.53	**0.82**±0.03	16.10	**0.82**±0.03	**14.10**	0.80±0.03	17.17	**0.81**±0.03	16.03	**0.81**±0.03	**13.20**
$\mathcal{D}_{12}$	0.99±0.01	4.67	0.99±0.01	4.63	0.99±0.01	**4.37**	0.99±0.01	5.93	0.99±0.01	5.57	0.99±0.01	**5.53**	**0.99**±0.01	7.50	0.98±0.01	**6.23**	**0.99**±0.01	6.73
FR	2.38		2.00		**1.63**		2.38		2.04		**1.58**		2.25		2.17		**1.58**	
d_{perf}	0.56		0.55		**0.53**		0.59		0.59		**0.56**		0.61		0.60		**0.58**	

vrijednosti prikazanih rezultata su podebljane za svaki skup podataka.

Na temelju prikazanih rangova i udaljenosti od savršenog klasifikatora, moguće je zaključiti da predloženi postupak, cjelokupno gledano, pronalazi mreže s boljom sposobnosti generalizacije od druga dva postupka izgradnje. Štoviše, na skupovima podataka na kojima ne uspijeva naći mreže s boljom sposobnosti generalizacije, one su obično manje veličine što je također poželjan ishod postupka za izgradnju RBFN. Kao što je ranije navedeno, RBFN se u literaturi mnoštvo puta iskazao kao prikladnim klasifikatorom za neuravnotežene skupove podataka. Iako predloženi postupak izgradnje klasifikacijskih modela RBFN nije dizajniran specifično za neuravnotežene skupove podataka, prosječne vrijednosti mjere F1 prikazane u tablici 5.5 daju naslutiti da na većini takvih problema (primjerice, na skupovima podataka $\mathcal{D}_3$, $\mathcal{D}_5$, $\mathcal{D}_{10}$, $\mathcal{D}_{11}$) nadmašuje preostala dva postupka izgradnje. Da bi se stekao uvid u učinkovitost klasifikacijskih modela izgrađenih predloženim postupkom specifično za manjinsku klasu, u tablici 5.6 prikazane su prosječne vrijednosti mjere TPR koje ostvaruju najbolje mreže (one koje postižu najbolji iznos mjere F1 na skupu za testiranje)

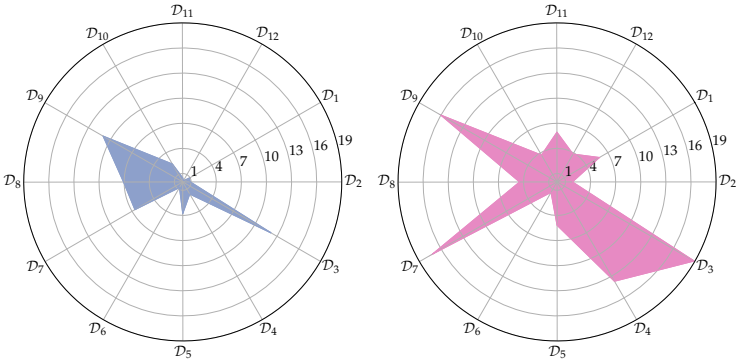
Tablica 5.6: Usporedba performansi predloženog postupka za izgradnju i performansi postupaka I_R i I_E u smislu mjere TPR

$\mathcal{D}$	I_R	I_E	Predloženi
$\mathcal{D}_1$	0.87±0.02	0.87±0.02	0.87±0.02
$\mathcal{D}_2$	**0.99**±0.01	0.98±0.01	0.98±0.01
$\mathcal{D}_3$	0.96±0.01	0.97±0.01	**0.99**±0.00
$\mathcal{D}_4$	0.93±0.05	0.93±0.04	0.93±0.05
$\mathcal{D}_5$	0.66±0.06	0.66±0.05	**0.69**±0.10
$\mathcal{D}_6$	**0.85**±0.03	0.84±0.04	0.84±0.04
$\mathcal{D}_7$	0.78±0.03	0.78±0.03	**0.81**±0.04
$\mathcal{D}_8$	0.94±0.02	**0.95**±0.02	**0.95**±0.02
$\mathcal{D}_9$	0.90±0.03	0.92±0.03	**0.93**±0.03
$\mathcal{D}_{10}$	0.89±0.05	0.92±0.04	**0.93**±0.05
$\mathcal{D}_{11}$	0.84±0.03	**0.85**±0.03	**0.85**±0.02
$\mathcal{D}_{12}$	0.99±0.01	0.99±0.01	0.99±0.01
FR	2.33	2.08	**1.58**
d_{perf}	0.51	0.50	**0.46**

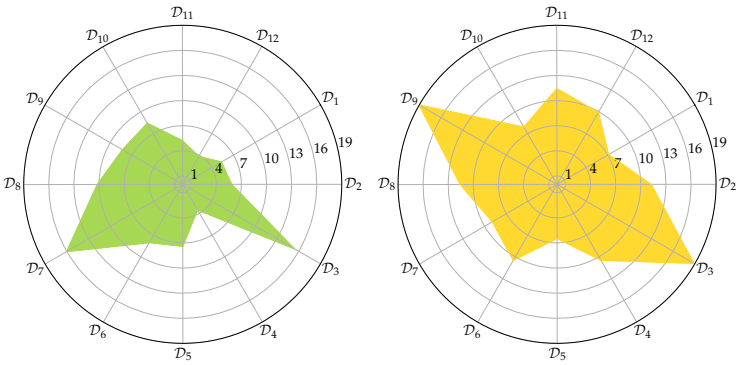
izgrađene uspoređenim postupcima. Iz prikazanih rezultata vidljivo je da najbolje mreže izgrađene predloženim postupkom ujedno ostvaruju i veći iznos mjere TPR na većini skupova podataka. S obzirom na činjenicu da predloženi postupak općenito pronalazi mreže koje imaju veću uspješnost prepoznavanja manjinske klase, može se zaključiti da je prikladniji od druga dva postupka izgradnje za učenje iz neuravnoteženih skupova podataka. Ovakav rezultat posljedica je predloženog načina dodavanja čvora u prethodno treniranu mrežu kojim se pospješuje pronalazak kvalitetnijih mreža tijekom treniranja, što u konačnici dovodi do klasifikacijskih modela RBFN koji imaju bolju sposobnost generalizacije te veću uspješnost prepoznavanja manjinske klase.

5.4.4 Usporedba predloženog s postupcima izgradnje RBFN iz literature

Predloženi postupak također je uspoređen s nekolicinom postupaka izgradnje RBFN iz literature, kako bi se stekao uvid u razlike klasifikacijskih modela izgrađenih ovim postupcima u smislu njihove kvalitete i veličine. S obzirom na to da je primarni cilj predloženog postupka izgraditi slijed mreža koje imaju bolju sposobnost generalizacije od mreža izgrađenih ostalim razmatranim postupcima, za svaku podjelu skupa podataka zabilježen je broj mreža izgrađenih ovim postupkom koje daju veći iznos mjere F1 na skupu za testiranje od svake najbolje mreže izgrađene ostalim postupcima, kao i u prvom dijelu eksperimentalne analize. Prosječni brojevi ovih mreža za svaki skup podataka prikazani su na slici 5.9. Moguće je primijetiti da predloženi postupak na većini skupova podataka u prosjeku pronalazi barem jednu mrežu s boljom sposobnosti generalizacije od najboljih mreža izgrađenih preostalim postupcima. Štoviše, broj takvih mreža je uglavnom velik (> 4), osim pri usporedbi s postup-



(a) U odnosu na J_{PSO} (b) U odnosu na $J_{k\text{-means}}$



(c) U odnosu na A_{DE} (d) U odnosu na A_{PSO}

Slika 5.9: Prosječan broj mreža s boljom sposobnosti generalizacije izgrađenih predloženim postupkom u odnosu na najbolje mreže izgrađene postupcima iz literature

kom J_{PSO}. Nadalje, kako bi se stekao uvid u razlike najboljih mreža izgrađenih uspoređenim postupcima, u tablici 5.7 su prikazane njihove prosječne kvalitete te prosječne veličine.

Na temelju prikazanih rangova i udaljenosti od savršenog klasifikatora, moguće je zaključiti da predloženi postupak izgradnje, cjelokupno gledano, pronalazi mreže s boljom sposobnosti generalizacije od ostalih postupaka. Po kvaliteti ostvarenih mreža uvjerljivo nadmašuje postupke A_{DE} i A_{PSO}. Valja podsjetiti da je ovim postupcima za traženje prikladne strukture mreže i vrijednosti njenih parametara omogućen veći broj vrednovanja funkcije cilja nego što je dan predloženom postupku za izgradnju svih 19 mreža. Unatoč tome, niti na jednom skupu podataka nisu pronašli mreže koje su u prosjeku kvalitetnije od prosjeka najboljih mreža izgrađenih predloženim postupkom, što je prvenstveno posljedica velike složenosti problema automatske izgradnje RBFN. Osim toga, kod ovih postupaka je problem podesiti član kazne u funkciji cilja na način da pogoduje svim skupovima podataka. Pri tome, postupak A_{DE} postiže bolji rang od postupka A_{PSO} te pronalazi manje mreže što nije iznenađujuće jer tijekom pretrage smanjuje granice u smislu broja čvorova. Nadalje, postupcima J_{PSO} i $J_{k\text{-means}}$ je za treniranje svake pojedine mreže omogućen jednak broj vrednovanja kao i automatskim postupcima za cjelokupno izvođenje. Na taj način imaju na raspolaganju veći broj mreža od kojih je jednostavno odabrati onu s najboljom sposobnosti generalizacije. Stoga ovi postupci također općenito pronalaze mreže s boljom sposobnosti generalizacije u

Tablica 5.7: Usporedba performansi predloženog postupka za izgradnju i performansi postupaka iz literature

$\mathcal{D}$	J_{PSO}		$J_{k\text{-means}}$		A_{DE}		A_{PSO}		Predloženi	
	F1	c	F1	c	F1	c	F1	c	F1	c
$\mathcal{D}_1$	**0.86**±0.01	6.17	0.85±0.02	15.00	0.85±0.02	**3.90**	0.84±0.03	12.00	**0.86**±0.02	7.73
$\mathcal{D}_2$	**0.98**±0.01	6.37	**0.98**±0.01	9.00	0.97±0.02	**2.40**	0.96±0.01	14.90	**0.98**±0.01	3.97
$\mathcal{D}_3$	0.93±0.01	15.67	0.85±0.01	15.80	0.90±0.02	**3.50**	0.64±0.11	18.57	**0.97**±0.01	16.90
$\mathcal{D}_4$	**0.86**±0.04	5.47	0.74±0.03	15.80	0.83±0.05	**2.53**	0.77±0.06	11.67	**0.86**±0.05	3.57
$\mathcal{D}_5$	0.64±0.05	13.67	0.62±0.07	16.00	0.58±0.10	14.27	0.58±0.08	**12.37**	**0.65**±0.07	13.90
$\mathcal{D}_6$	**0.84**±0.04	**5.27**	**0.84**±0.04	7.60	0.79±0.04	6.43	0.78±0.04	9.80	0.83±0.04	6.17
$\mathcal{D}_7$	0.77±0.02	**4.23**	0.56±0.02	6.40	0.55±0.05	2.53	0.75±0.03	11.37	**0.78**±0.04	15.03
$\mathcal{D}_8$	0.93±0.02	**6.77**	0.94±0.02	14.30	0.92±0.02	6.97	0.91±0.03	15.43	**0.95**±0.02	11.27
$\mathcal{D}_9$	0.89±0.02	11.53	0.85±0.03	15.00	0.91±0.03	**8.50**	0.64±0.06	18.77	**0.93**±0.03	12.57
$\mathcal{D}_{10}$	**0.93**±0.03	9.17	0.92±0.03	16.80	0.87±0.06	**6.20**	0.88±0.04	10.23	0.92±0.04	9.63
$\mathcal{D}_{11}$	**0.84**±0.03	14.17	0.81±0.03	16.60	0.81±0.03	**11.97**	0.65±0.16	18.67	0.83±0.03	13.13
$\mathcal{D}_{12}$	**1.00**±0.01	4.50	0.99±0.01	5.50	0.98±0.02	4.63	0.96±0.03	16.67	0.99±0.01	**4.37**
FR	1.79		2.96		3.58		5.00		**1.67**	
d_{perf}	0.55		0.74		0.74		0.88		**0.53**	

odnosu na automatske postupke izgradnje, no uz daleko veći utrošak računalnih resursa. Pri tome, mreže nađene postupkom $J_{k\text{-means}}$ su uglavnom osjetno lošije kvalitete od mreža nađenih postupkom J_{PSO}, što ne čudi s obzirom na to da je treniranje kod potonjeg vođeno uspješnosti klasifikacijskih modela, dok je algoritam k-means postupak nenadziranog učenja. Osim toga, mreže nađene postupkom $J_{k\text{-means}}$ ujedno su i veće složenosti.

Nadalje, za očekivati je da najbolje mreže izgrađene predloženim postupkom imaju i bolju uspješnost prepoznavanja manjinske klase, s obzirom na prethodno naznačenu međuovisnost mjera F1 i TPR. Da bi se potvrdilo ovo očekivanje, u tablici 5.8 su prikazane prosječne vrijednosti mjere TPR koje ostvaruju najbolje mreže izgrađene uspoređenim postupcima. Iz prikazanih rezultata je moguće zaključiti da najbolje mreže izgrađene predloženim postupkom ujedno ostvaruju i veći iznos mjere TPR na većini skupova podataka. Predloženi postupak stoga se može smatrati prikladnijim postupkom izgradnje klasifikacijskih modela RBFN za neuravnotežene skupove podataka u odnosu na ostale razmatrane postupke.

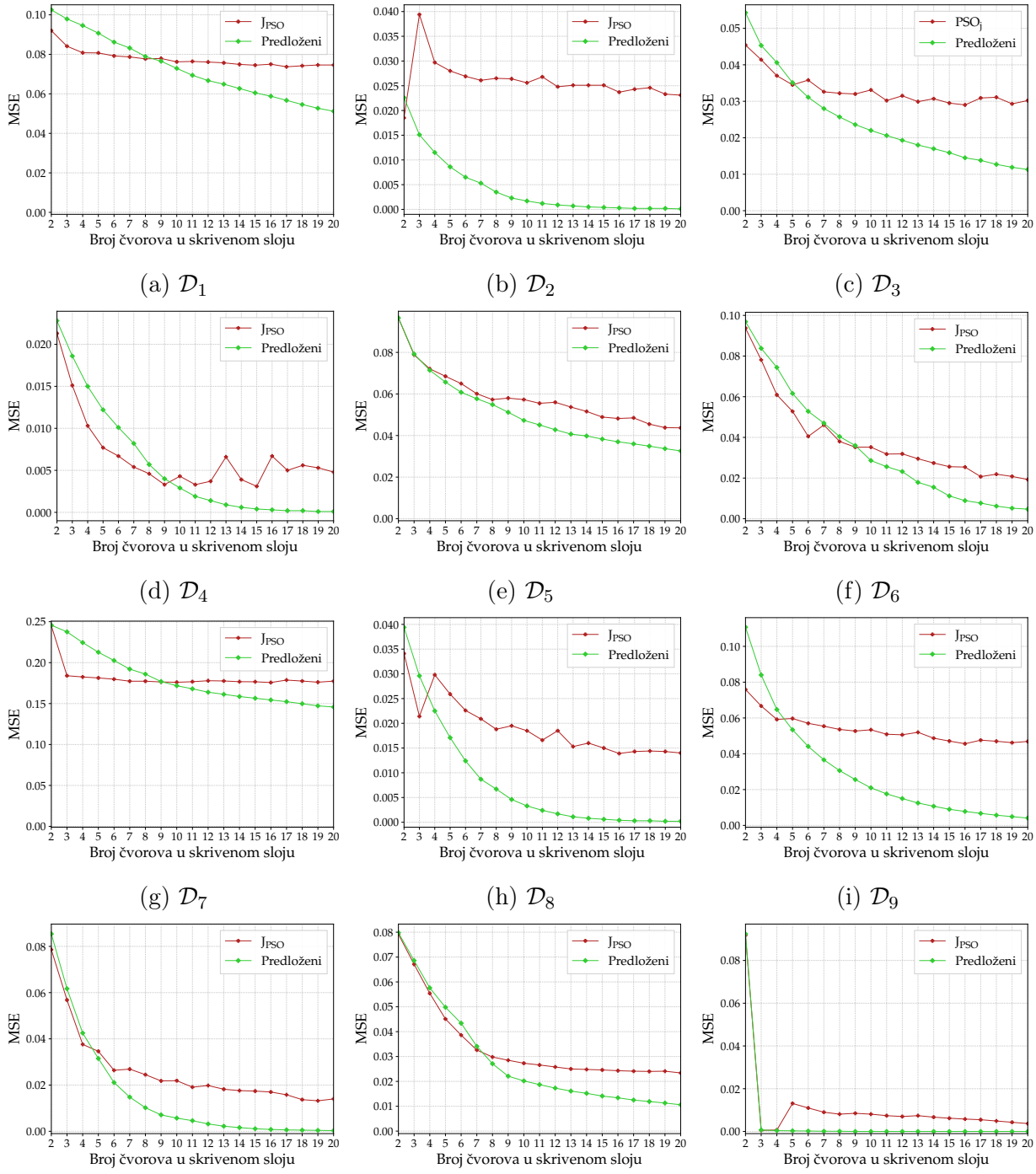
Rezultati prikazani u tablicama 5.7 i 5.8 sugeriraju da je prema uspješnosti izgrađenih klasifikacijskih modela najveći konkurent predloženom postupku izgradnje postupak J_{PSO}, koji se od njega razlikuje samo po tome što ne koristi strukturu prethodno trenirane mreže pri treniranju mreže veće složenosti. Unatoč utrošku značajno većeg broja vrednovanja za treniranje svake mreže, ipak ostvaruje nešto lošiji rang od predloženog postupka izgradnje što nedvojbeno potvrđuje korisnost prethodno stečenog znanja. Da bi se dodatno analizirala korisnost ovog koraka, na slici 5.10 prikazani su medijani izvođenja u smislu mjere MSE na skupu za treniranje za mreže izgrađene predloženim postupkom te postupkom J_{PSO}. Vidljivo je kako predloženi postupak izgradnje općenito pronalazi veći broj kvalitetnijih mreža na skupu za treniranje. Štoviše, na većini problema uspijeva pronaći nekoliko mreža

Tablica 5.8: Usporedba performansi predloženog postupka za izgradnju i performansi postupaka iz literature u smislu mjere TPR

$\mathcal{D}$	$\mathrm{J_{PSO}}$	$\mathrm{J}_{k\text{-means}}$	$\mathrm{A_{DE}}$	$\mathrm{A_{PSO}}$	Predloženi
$\mathcal{D}_1$	**0.87**±0.02	0.85±0.02	0.86±0.02	0.85±0.03	**0.87**±0.02
$\mathcal{D}_2$	**0.98**±0.01	**0.98**±0.01	0.97±0.02	0.96±0.01	**0.98**±0.01
$\mathcal{D}_3$	0.97±0.00	0.95±0.01	0.95±0.01	0.77±0.19	**0.99**±0.00
$\mathcal{D}_4$	0.93±0.05	**0.96**±0.00	0.90±0.07	0.83±0.06	0.93±0.05
$\mathcal{D}_5$	0.66±0.05	0.63±0.07	0.59±0.12	0.60±0.09	**0.69**±0.10
$\mathcal{D}_6$	0.84±0.04	**0.85**±0.04	0.79±0.04	0.79±0.03	0.84±0.04
$\mathcal{D}_7$	0.80±0.02	0.57±0.02	0.61±0.11	0.77±0.03	**0.81**±0.04
$\mathcal{D}_8$	0.94±0.02	0.94±0.02	0.93±0.02	0.92±0.03	**0.95**±0.02
$\mathcal{D}_9$	0.89±0.02	0.86±0.03	0.91±0.03	0.65±0.06	**0.93**±0.03
$\mathcal{D}_{10}$	0.92±0.03	**0.94**±0.03	0.87±0.06	0.88±0.05	0.93±0.05
$\mathcal{D}_{11}$	**0.86**±0.03	0.83±0.04	0.83±0.03	0.66±0.17	0.85±0.02
$\mathcal{D}_{12}$	**1.00**±0.01	0.99±0.01	0.98±0.02	0.96±0.03	0.99±0.01
FR	2.08	2.79	3.88	4.58	**1.67**
d_{perf}	0.50	0.65	0.68	0.79	**0.46**

različitih složenosti koje su kvalitetnije od najkvalitetnijih mreža nađenih postupkom $\mathrm{J_{PSO}}$. Pri tome, moguće je primijetiti kako postupak $\mathrm{J_{PSO}}$ uglavnom pronalazi kvalitetnije mreže za manje brojeve čvorova u skrivenom sloju, dok povećanjem složenosti prednost predloženog postupka postaje sve izraženija. Razlog tomu je manji utjecaj korištenja ranije stečenog znanja u početku izgradnje te uvjerljivo manji broj vrednovanja. Razliku u performansama ovih postupaka u početku izgradnje potencijalno je moguće smanjiti drugačijom raspodjelom vrednovanja između pojedinih koraka treniranja u predloženom postupku. No, iz prikazanih kvaliteta mreža izgrađenih ovim postupcima moguće je zaključiti kako utrošak značajno većeg broja vrednovanja ne uspijeva nadomjestiti izostanak korištenja prethodno stečenog znanja, posebice pri treniranju mreža veće složenosti.

Kao što je ranije spomenuto, dodatan cilj predloženog postupka jest pronaći slijed kvalitetnih mreža u manje vremena u odnosu na ostale postupke. Prethodno prikazani rezultati ukazuju na to da uspijeva pronaći kvalitetnije mreže uz korištenje manjeg broja vrednovanja funkcije cilja. Kako bi se dobio jasniji uvid u razlike u trajanju ovih postupaka, slika 5.11 prikazuje usporedbu prosječnog trajanja svih razmatranih postupaka izgradnje pomoću linijskog dijagrama s logaritamskom skalom. Trajanje izvođenja svakog postupka varira ovisno o karakteristikama skupa podataka, no vidljivo je kako predloženi postupak traje približno kao i automatski postupci izgradnje, iako za razliku od njih pronalazi slijed mreža različitih složenosti. Prikazana prednost u trajanju prvenstveno je posljedica predloženog načina iskorištenja znanja iz prethodno treniranih mreža te potvrđuje ispunjenje dodatnog cilja predloženog postupka izgradnje.



(a) $\mathcal{D}_1$ (b) $\mathcal{D}_2$ (c) $\mathcal{D}_3$

(d) $\mathcal{D}_4$ (e) $\mathcal{D}_5$ (f) $\mathcal{D}_6$

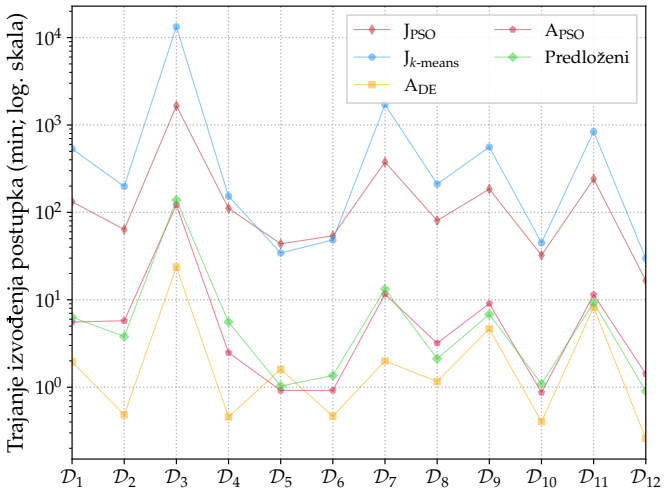
(g) $\mathcal{D}_7$ (h) $\mathcal{D}_8$ (i) $\mathcal{D}_9$

(j) $\mathcal{D}_{10}$ (k) $\mathcal{D}_{11}$ (l) $\mathcal{D}_{12}$

Slika 5.10: Kvalitete mreža izgrađenih predloženim postupkom te postupkom J_{PSO} na skupu za treniranje

5.5 Osvrt na RBFNs i predloženi postupak izgradnje

Radijalna neuronska mreža primjer je klasifikatora koji općenito ispoljava zadovoljavajuće performanse na neuravnoteženim skupovima podataka, prvenstveno zbog svoje sposobnosti lokaliziranog djelovanja u ulaznom prostoru. No, izgradnja odgovarajućeg klasifikacijskog modela RBFN za promatrani problem predstavlja zahtjevan problem globalne optimizacije,



Slika 5.11: Usporedba prosječnog trajanja postupaka za izgradnju RBFN

a podrazumijeva traženje prikladnog broja čvorova u skrivenom sloju mreže te vrijednosti parametara koji opisuju mrežu. Tek nekolicina postupaka izgradnje iz literature objedinjuje traženje broja čvorova skrivenog sloja i traženje preostalih parametara te tako provode automatski način izgradnje klasifikacijskog modela RBFN. Međutim, kao što je pokazano u eksperimentalnoj analizi, ovi postupci uglavnom ostvaruju lošije performanse od jednostavnijih postupaka koji treniraju više mreža različitih složenosti. Razlog tomu je što jednostavni postupci pronalaze zadani broj mreža od kojih najčešće barem jedna ima bolju sposobnost generalizacije od jedine mreže izgrađene automatskim postupkom izgradnje. No, za traženje slijeda mreža obično je potrebno utrošiti više računalnih resursa nego za provedbu automatske izgradnje. Kako bi se smanjilo vrijeme potrebno za pronalazak ovakvih mreža, predložen je postupak izgradnje klasifikacijskih modela RBFN koji se temelji na postupnom treniranju mreža sve veće složenosti, ali uz korištenje znanja iz prethodno treniranih mreža manje složenosti. Za razliku od jednostavnih postupaka izgradnje, u predloženom postupku struktura prethodno trenirane mreže se prenosi u naredni korak treniranja mreže veće složenosti. Rezultati eksperimentalne analize ukazuju na to da korištenje prethodno stečenog znanja uvelike olakšava i ubrzava pronalazak mreža zadovoljavajuće uspješnosti klasifikacije. Konkretno, predloženi postupak uspijeva izgraditi mreže s boljom sposobnosti generalizacije od mreža izgrađenih nekolicinom jednostavnih i automatskih postupaka izgradnje RBFN iz literature. Osim toga, pokazano je da te mreže ujedno ostvaruju bolju uspješnost prepoznavanja manjinske klase pa se predloženi postupak izgradnje stoga može smatrati prikladnijim za učenje iz neuravnoteženih skupova podataka.

Korištenje strukture prethodno trenirane mreže pri treniranju mreže veće složenosti predloženo je samo u [249]. Međutim, u odnosu na taj postupak, predloženi postupak izgradnje ulaže značajan napor u traženje prikladnog čvora koji se nadodaje (u smislu njegovih interakcija s postojećim čvorovima) kako bi se olakšalo i ubrzalo treniranje cijele mreže. Eks-

perimentalnom analizom je pokazano kako glavnina učinka predloženog postupka proizlazi upravo iz njegova načina dodavanja ovog čvora. Iako troši određenu količinu računalnih resursa, provođenje ovog načina dodavanja pretežno je korisnije te vremenski manje zahtjevno od produljenja treniranja cijele mreže. Štoviše, pokazano je i da zanemarivanje interakcija novododanog čvora s preostalim čvorovima prethodno trenirane mreže ponekad uzrokuje pogoršanje njezine kvalitete, što na neki način umanjuje doprinos prethodno stečenog znanja.

Kao što je prethodno spomenuto, učinkovitost predloženog postupka potvrđena je njegovom usporedbom s nekolicinom različitih postupaka izgradnje RBFN koji koriste drugačije algoritme za treniranje mreža. U usporedbi sa svim razmatranim postupcima, predloženi postupak općenito uspijeva izgraditi kvalitetnije klasifikacijske modele uz manji utrošak vremena. Time on predstavlja ispunjenje prijedloga trećeg izvornog znanstvenog doprinosa ove disertacije.

6 Zaključak

Učenje iz neuravnoteženih podataka izazovan je zadatak koji je zastupljen u brojnim problemima klasifikacije koji proizlaze iz raznih područja primjene, poput medicinske dijagnostike te otkrivanja upada, grešaka ili prijevara. Pozamašan broj pristupa ublažavanju problema neuravnoteženosti klasa u literaturi otežava odabir odgovarajućeg pristupa za dani problem. Ipak, postupci predobrade skupova podataka koji smanjuju njihovu složenost, poput odabira značajki i preuzorkovanja, te klasifikator radijalne neuronske mreže (RBFN) zbog sposobnosti lokaliziranog djelovanja u ulaznom prostoru, doimaju se prikladnim pristupima za ublažavanje ovog problema. Iako su brojne izvedbe ovih postupaka predložene u literaturi, prostora za njihova poboljšanja ne nedostaje, kao što je pokazano predloženim unaprjeđenjima. Također, postoji prostor za daljnje dorade predloženih unaprjeđenja, što bi moglo dodatno pridonijeti njihovoj učinkovitosti i otpornosti na različite manifestacije problema neuravnoteženosti klasa.

6.1 Zaključci

Problem neuravnoteženosti klasa u pravilu se manifestira povećanom složenosti skupa podataka, što zauzvrat narušava izvedbu većine klasifikatora. Posljedica učenja iz neuravnoteženih podataka jest pristranost klasifikatora većinskoj klasi, zbog čega ostvaruje lošu uspješnost prepoznavanja manjinske klase. Kao što je istaknuto u poglavlju 2, brojni su problemi klasifikacije po prirodi neuravnoteženi, pri čemu su upravo događaji čije je prepoznavanje od primarne važnosti najčešće predstavljeni nedostatnim brojem primjeraka. S obzirom na složenost i rasprostranjenost ovog problema, nije iznenađujuće da su predloženi razni pristupi

za njegovo ublažavanje. Pristupi na razini podataka jedini su koji izravno ublažavaju stupanj problema neuravnoteženosti klasa u skupu podataka, što u konačnici pogoduje izvedbi raznih tipova klasifikatora. Među ovim pristupima, odabir značajki i preuzorkovanje mogu se istaknuti kao postupci predobrade skupa podataka koji smanjuju složenost koncepta manjinske klase te u pravilu pospješuju njezino prepoznavanje. Njihova jednostavnost i učinkovitost čine ih najzastupljenijim postupcima za ublažavanje ovog problema u literaturi. Učinkovitost ovih postupaka potvrđena je i ostvarenim rezultatima eksperimentalnih analiza izloženih u ovoj disertaciji, koji ukazuju na to da njihova provedba uzrokuje poboljšanje opće izvedbe raznih klasifikatora te da ono prvenstveno proizlazi iz povećanja uspješnosti prepoznavanja manjinske klase. Iako se odabirom značajki i preuzorkovanjem smanjuje složenost neuravnoteženih skupova podataka te pospješuje izvedba raznih tipova klasifikatora, ovi postupci ne uklanjaju problem. Stoga se pri odabiru klasifikatora ne smiju zanemariti njegova svojstva u pogledu učenja iz neuravnoteženih skupova podataka. Primjer klasifikatora koji je demonstrirao zadovoljavajuće ponašanje i ostvario povoljne performanse na brojnim neuravnoteženim problemima klasifikacije u literaturi jest RBFN. Odabir značajki, preuzorkovanje te izgradnja klasifikacijskih modela RBFN stoga su u središtu istraživanja ove disertacije u kojoj su predloženi unaprijeđeni postupci za njihovo provođenje.

U literaturi su predloženi brojni pristupi odabiru značajki, pri čemu se omotači zasnovani na bio-inspiriranim algoritmima optimizacije ističu po svojoj sposobnosti otkrivanja složenih interakcija između značajki te postizanja povoljnih performansi u skladu s tim. No, skloni su pretjerano prilagoditi rješenja skupu za vrednovanje, što umanjuje njihov doprinos sposobnosti generalizacije klasifikatora. Osim toga, zbog svoje stohastičke prirode, ponavljanjem pretrage često pronalaze različita rješenja što ih čini nestabilnim pristupima odabiru značajki. S ciljem ublažavanja ovih nedostataka, u disertaciji je predloženo proširenje bio-inspiriranih omotača koje se zasniva na prikupljanju kvalitetnih i raznolikih rješenja tijekom pretrage te njihovu naknadnom objedinjavanju. Prikupljanjem ovakvih rješenja izbjegava se oslanjanje isključivo na rješenje nađeno omotačem koje može biti pretjerano prilagođeno skupu za vrednovanje te se olakšava naknadno stjecanje uvida u relevantnost pojedinih značajki za promatrani problem i u njihove interakcije. Nakon prikupljanja rješenja, izdvajaju se njihove zajedničke značajke, a preostale značajke postupno uvode u tako dobiveni podskup prema njihovu doprinosu kvaliteti. Učinkovitost predloženog proširenja ispitana je za tri standardna omotača te za tri unaprijeđena omotača iz literature. Analizom postupka objedinjavanja u predloženom proširenju utvrđeno je da on čini glavninu njegova učinka. Predloženi način objedinjavanja uspijeva stvoriti rješenje koje klasifikatoru općenito omogućuje veću sposobnost generalizacije od ostalih rješenja unutar arhive, što samo dodatno ukazuje na to da su ona u arhivi pretjerano prilagođena skupu za vrednovanje. Nadalje, za sve razmatrane omotače pokazano je da ugradnja predloženog proširenja donosi povoljne rezultate u smislu kvalitete i veličine formiranih rješenja te povećava stabilnost omotača u smislu višestrukog izvođenja pretrage te preslagivanja skupa podataka. Treba napome-

nuti da se navedene povoljnije performanse predloženog proširenja ogledaju u poboljšanju uspješnosti prepoznavanja manjinske klase.

Preuzorkovanje primjeraka manjinske klase standardni je postupak za ublažavanje problema neuravnoteženosti klasa u literaturi, a jedan od najistaknutijih i najkorištenijih algoritama preuzorkovanja jest algoritam SMOTE. Iako je ovaj algoritam iznimno popularan u literaturi zbog svoje korisnosti i jednostavnosti, ima određene nedostatke koji mogu uzrokovati povećanje složenosti skupa podataka te narušavanje kvalitete izvedbe klasifikatora. Nedostaci ovog algoritma izraženiji su pri neprikladnim postavkama njegovih parametara te se njihovo podešavanje može smatrati neophodnim korakom pri korištenju ovog algoritma. U ovoj disertaciji, predloženo je novo unaprjeđenje algoritma SMOTE kojim se nastoji pojednostaviti uporaba izvornog algoritma te održati ili nadmašiti kvaliteta njegove izvedbe. Predloženi algoritam sastoji se u suštini od dva koraka koji se provode za svaki manjinski primjerak u skupu podataka, a to su određivanje njegova susjedstva te stvaranje sintetičkih primjeraka konveksnom kombinacijom promatranog primjerka i njegovih nasumično odabranih susjeda. Susjedstvo svakog manjinskog primjerka čine oni manjinski primjerci koji su od njega manje ili jednako udaljeni kao njegov najbliži susjed iz većinske klase. Uzimajući u obzir većinsku klasu pri određivanju susjedstva te veličinu susjedstva pri određivanju broja stvorenih sintetičkih primjeraka, izbjegava se povećanje stupnja preklapanja klasa i minimalno preuzorkuje šum, čemu se u algoritmu SMOTE ne pridaje pozornost. Osim toga, uklonjena je potreba za parametrima jer je način rada predloženog algoritma određen unutarnjim karakteristikama skupa podataka, što ga čini trivijalnim za korištenje u odnosu na algoritam SMOTE, a posebno u odnosu na mnoge njegove unaprijeđene inačice. Učinak predloženog algoritma eksperimentalno je uspoređen s učinkom algoritma SMOTE i nekoliko njegovih unaprijeđenih inačica. Pokazano je da on generalno najviše doprinosi poboljšanju opće uspješnosti klasifikacije, neovisno o korištenom klasifikatoru. Također, primijećeno je da od uspoređenih algoritama predloženi najmanje narušava uspješnost prepoznavanja većinske klase, a podjednako povećava uspješnost prepoznavanja manjinske klase kao i ostali, što je prvenstveno posljedica njegova izbjegavanja uvođenja sintetičkih primjeraka u prostor većinske klase.

Ponašanje i performanse klasifikatora RBFN ovise o složenosti klasifikacijskog modela te o postavkama njegovih parametara. Oni se određuju u sklopu izgradnje klasifikacijkog modela RBFN koji u suštini predstavlja složen problem globalne optimizacije. Postupci izgradnje klasifikacijskih modela RBFN imaju dva glavna cilja, a to su ostvarivanje dobre sposobnosti generalizacije klasifikatora te održavanje relativno jednostavnog modela u smislu broja čvorova u skrivenom sloju. Mogu se podijeliti na jednostavnije postupke koji treniraju veći broj mreža različitih složenosti te na složenije postupke automatske izgradnje koji objedinjuju traženje broja čvorova skrivenog sloja i traženje preostalih parametara mreže. Iako je treniranje većeg broja mreža vremenski zahtjevnije od postupka automatske izgradnje, prednost jednostavnijih postupaka jest ta što daju na raspolaganje određeni broj treniranih

mreža od kojih je potom jednostavno odabrati onu s najboljom sposobnosti generalizacije. Smanjivanje vremena potrebnog za pronalazak ovakvih mreža može biti od izvjesne praktične važnosti. S tim ciljem, u ovoj disertaciji predložen je novi postupak izgradnje klasifikacijskih modela RBFN koji se temelji na postupnom treniranju mreža sve većeg stupnja složenosti, ali uz korištenje znanja iz prethodno treniranih mreža manje složenosti. Pri tome, prethodno treniranoj mreži nadodaje se novi čvor prije treniranja mreže veće složenosti. Određivanje parametara ovog čvora istaknut je korak predloženog postupka izgradnje koji ima za cilj pronaći čvor s povoljnim interakcijama s postojećim čvorovima te mreže. Korisnost ovakvog načina dodavanja čvora u prethodno treniranu mrežu utvrđena je u prvom dijelu eksperimentalne analize. Pokazano je da korištenje prethodno trenirane mreže s povoljnim interakcijama između čvorova uvelike pospješuje treniranje mreže povećane složenosti, što u konačnici olakšava izgradnju slijeda mreža povoljne kvalitete. Nadalje, usporedbom s nekolicinom automatskih i jednostavnih postupaka izgradnje iz literature utvrđeno je kako predloženi postupak uspijeva izgraditi po nekoliko mreža s boljom sposobnosti generalizacije od najboljih mreža izgrađenih ostalim postupcima. S obzirom na to da je jednostavnim postupcima izgradnje u eksperimentalnoj analizi omogućeno osjetno više računalnih resursa za izgradnju slijeda mreža, ostvareni rezultati ukazuju na to da korištenje prethodno stečenog znanja uvelike olakšava i ubrzava pronalazak mreža zadovoljavajuće uspješnosti klasifikacije. Ustanovljeno je da te mreže ujedno imaju i bolju uspješnost prepoznavanja manjinske klase, što predloženi postupak čini prikladnijim postupkom izgradnje klasifikacijskih modela RBFN za neuravnotežene skupove podataka.

Konačno, predložena unaprjeđenja, kao što sugeriraju predstavljeni rezultati, pružaju osjetno povećanje učinkovitosti pri učenju iz neuravnoteženih podataka u odnosu na uobičajene izvedbe postupaka za odabir značajki, preuzorkovanje te izgradnju klasifikacijskih modela RBFN. Uz to, značajno pojednostavljuju primjenu preuzorkovanja te klasifikatora RBFN, što je posljedica uklanjanja parametara iz algoritma SMOTE te korištenja prethodno stečenog znanja pri treniranju klasifikacijskih modela RBFN povećane složenosti. Također, predloženo proširenje bio-inspiriranih omotača pokazalo se povoljnijim za performanse i stabilnost omotača od produljenja njegove pretrage. Osim u odnosu na standardne izvedbe ovih postupaka, postignuti su bolji ili usporedivi rezultati i u odnosu na nekolicinu njihovih unaprjeđenja iz literature. Na temelju navedenog, može se izvesti zaključak da su ciljevi disertacije, a time i očekivani izvorni znanstveni doprinosi, ispunjeni.

6.2 Budući rad

Iako ugradnja i primjena predloženih unaprjeđenja općenito pruža bolju uspješnost klasifikacije pri učenju iz neuravnoteženih skupova podataka, prostora za njihovu daljnju doradu ne nedostaje. Osim toga, njihove moguće kombinacije pružaju potencijal za daljnje poboljšanje

učinkovitosti pri ublažavanju problema neuravnoteženosti klasa.

Predloženo proširenje bio-inspiriranih omotača sastoji se u osnovi od koraka prikupljanja rješenja te koraka njihova objedinjavanja. Oba koraka su usko povezana te izmjene prvog moraju uzeti u obzir posljedice na izvedbu drugog. Rješenja se u arhivu uvrštavaju prema kvaliteti, no osim kvalitete moguće je definirati i dodatne kriterije prikupljanja, poput stupnja njihove raznolikosti u odnosu na postojeća rješenja u arhivi. S obzirom na multimodalnost problema odabira značajki, velik broj rješenja značajno različitih struktura može imati približno istu kvalitetu. Stoga bi možda bilo korisno razmotriti prikupljanje raznovrsnijih rješenja radi boljeg uvida u relevantne značajke, ali treba pripaziti jer se prevelika raznolikost može negativno odraziti na performanse rješenja dobivenog objedinjavanjem. Nadalje, razmatranje veličine arhive kao parametra predloženog proširenja predstavlja valjanu smjernicu za budući rad. Iako je poželjno imati što više kvalitetnih i raznolikih rješenja u arhivi, treba paziti da se zbog prevelikog broja rješenja ne izgubi istaknutost važnosti nekih značajki, što bi dovelo do negativnog učinka objedinjavanja takvih rješenja. Radi lakšeg podešavanja vrijednosti parametra za pojedinačni problem, poželjno je ispitati razne vrijednosti ovisno o postavkama omotača (primjerice, proporcionalno veličini arhive i/ili broju vrednovanja) i dimenzionalnosti skupa podataka. Konačno, može se istražiti ponašanje drugih načina objedinjavanja rješenja unutar arhive koji su također zasnovani na logičkim (primjerice, presjek i unija) i aritmetičkim operacijama (primjerice, glasovanje) između binarnih vektora. U predloženom načinu objedinjavanja sva rješenja u arhivi imaju jednaku važnost zbog pretpostavke da će arhiva sadržavati rješenja približno jednake kvalitete. No, moguće je da se njihove kvalitete bitno razlikuju što varira o postavkama omotača, veličini arhive te samom problemu. Kako bi se dala veća važnost značajkama koje su sadržane u kvalitetnijim rješenjima, umjesto operacije presjeka svih rješenja u arhivi, moguće je istražiti učinak težinskog glasovanja, gdje bi težina glasa svakog rješenja odgovarala njegovoj kvaliteti.

Daljnja istraživanja predloženog algoritma preuzorkovanja uključivala bi razmatranje različitih načina stvaranja sintetičkih primjeraka. Iako je stvaranje sintetičkih primjeraka konveksnom kombinacijom postojećih manjinskih primjeraka najčešći način njihova stvaranja u literaturi, ne nedostaje prostora za daljnja istraživanja ovog koraka, pogotovo jer su u literaturi predloženi razni načini stvaranja ovih primjeraka. Nadalje, u slučaju kada promatrani manjinski primjerak nema susjeda, predloženi algoritam stvara jedan sintetički primjerak na liniji između njega te nasumično odabranog manjinskog primjerka. U svrhu očuvanja postojećih koncepata manjinske klase, određivanje njegova položaja na temelju položaja ostalih manjinskih primjeraka treba dodatno analizirati. Primjerice, umjesto nasumičnog odabira manjinskog primjerka, moguće je osmisliti heuristiku za odabir tog primjerka ili čak uključiti više primjeraka u stvaranje sintetičkog primjerka. Konačno, predloženi algoritam preuzorkovanja stvara relativno malo primjeraka u slučaju apsolutne rijetkosti manjinskih primjeraka ili velike količine šuma u konceptu manjinske klase. Stoga, ako je pak potrebno zadržati određenu razinu kontrole u smislu količine novostvorenih primjeraka, moguće ga je proširiti

parametrom koji bi utjecao na broj novostvorenih sintetičkih primjeraka. Ovaj parametar može se koristiti za skaliranje veličine susjedstava kako bi se stvorio željeni broj primjeraka, a da preuzorkovanje ne bude jednakomjerno.

Predloženi postupak za izgradnju klasifikacijskih modela RBFN vraća slijed mreža različitih složenosti unutar zadanih granica u smislu broja čvorova u skrivenom sloju, pri čemu se u konačnici odabire jedna od tih mreža za korištenje. Na nekolicini skupova podataka korištenih u eksperimentalnoj analizi moglo se primijetiti da se složenije mreže u izgrađenom slijedu mreža neznatno razlikuju po ostvarenim performansama na skupu za treniranje. Njihovim treniranjem stoga su nepotrebno utrošeni računalni resursi te bi se u budućem radu moglo razmisliti o zaustavljanju postupka izgradnje i prije nego što dosegne najveći dozvoljeni broj čvorova u skrivenom sloju. Prilagodba metode lakta (engl. *elbow method*), koja se koristi za određivanje prikladnog broja grupa pri primjeni algoritma k-means za grupiranje podataka, doima se prikladnim načinom za određivanje granice u smislu broja čvorova, pri čemu bi postupak izgradnje mogao vratiti posljednju treniranu mrežu prije nego što se premaši ta granica. Time bi nalikovao automatskom postupku izgradnje, s bitnom razlikom da su prethodno trenirane mreže i dalje na raspolaganju. Nadalje, u predloženom postupku izgradnje troši se određeni broj vrednovanja pri dodavanju novog čvora u prethodno treniranu mrežu. U budućem radu potrebno je detaljnije analizirati broj vrednovanja potreban za traženje ovog čvora te razmisliti o njegovu podešavanju ovisno o dimenzionalnosti problema te ukupnom dozvoljenom broju vrednovanja. U eksperimentalnoj analizi moglo se primijetiti kako prednost predloženog postupka u odnosu na ostale razmatrane postupke izgradnje (u smislu kvalitete treniranih mreža) postaje sve izraženija s povećanjem složenosti mreže. Razlika u performansama ovih postupaka u početku izgradnje potencijalno se može smanjiti drugačijom raspodjelom vrednovanja između pojedinih koraka treniranja u predloženom postupku. Jedan od mogućih načina bio bi da se predloženom postupku u početku izgradnje omogući nešto veći broj vrednovanja, a da se u kasnijoj fazi smanji. Ovo bi posebice bilo prikladno ako bi se ugradio neki način zaustavljanja cjelokupnog postupka izgradnje, kao što je prethodno navedeno.

Ponašanje i učinkovitost predloženih unaprijeđenih postupaka za odabir značajki, preuzorkovanje te izgradnju klasifikacijskih modela RBFN testirani su i analizirani zasebno. Međutim, provedba jednog od ovih postupaka ne isključuje provedbu drugih. Dapače, u literaturi je uobičajeno kombinirati ih (primjerice, odabir značajki i preuzorkovanje u [83, 89] te preuzorkovanje i RBFN u [20, 203, 204]), posebice pri učenju iz neuravnoteženih skupova podataka. Sukladno tome, razmatranje njihovih kombinacija nameće se kao budući rad. Iako pojedinačno pružaju povećanje učinkovitosti, teško je predvidjeti ponašanje njihovih kombinacija. No, vodeći se literaturom, da se pretpostaviti da bi kombinacije ovih postupaka dale povoljne performanse. U tom smislu, potrebna su opsežnija ispitivanja i moguća usklađivanja predloženih postupaka.

Literatura

[1] D. Lu i Q. Weng. A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 28(5):823–870, 2007.

[2] L. Deng i X. Li. Machine learning paradigms for speech recognition: An overview. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(5):1060–1089, 2013.

[3] B. Agarwal i N. Mittal. Text classification using machine learning methods-a survey. *Proceedings of the 2nd International Conference on Soft Computing for Problem Solving (SocProS)*, stranice 701–709, 2014.

[4] N. Ortiz, R. D. Hernández, R. Jimenez, M. Mauledeoux i O. Avilés. Survey of biometric pattern recognition via machine learning techniques. *Contemporary Engineering Sciences*, 11(34):1677–1694, 2018.

[5] S. Theodoridis, A. Pikrakis, K. Koutroumbas i D. Cavouras. *Introduction to pattern recognition: a matlab approach*. Academic Press, 2010.

[6] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue i G. Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017.

[7] X.-M. Zhao, X. Li, L. Chen i K. Aihara. Protein classification with imbalanced data. *Proteins: Structure, Function, and Bioinformatics*, 70(4):1125–1132, 2008.

[8] R. J. Franklin, V. Dabbagol et al. Anomaly detection in videos for video surveillance applications using neural networks. *Proceedings of the 4th International Conference on Inventive Systems and Control (ICISC)*, stranice 632–637, 2020.

[9] D. A. Cieslak, N. V. Chawla i A. Striegel. Combating imbalance in network intrusion datasets. *Proceedings of the 2006 International Conference on Granular Computing*, stranice 732–737, 2006.

[10] H. He i E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[11] V. López, A. Fernández, S. García, V. Palade i F. Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113–141, 2013.

[12] P. Skryjomski i B. Krawczyk. Influence of minority class instance types on smote imbalanced data oversampling. *Proceedings of the 1st International Workshop on Learning with Imbalanced Domains: Theory and Applications*, stranice 7–21, 2017.

[13] G. Foody. Supervised image classification by mlp and rbf neural networks with and without an exhaustively defined set of classes. *International Journal of Remote Sensing*, 25(15):3091–3104, 2004.

[14] A. Raad, A. Kalakech i M. Ayache. Breast cancer classification using neural network approach: Mlp and rbf. *Proceedings of the 13th International Arab Conference on Information Technology*, stranica 105, 2012.

[15] D. Tripathi, D. R. Edla, V. Kuppili i R. Dharavath. Binary bat algorithm and rbfn based hybrid credit scoring model. *Multimedia Tools and Applications*, 79(43):31889–31912, 2020.

[16] M. Wasikowski i X.-w. Chen. Combating the small sample class imbalance problem using feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 22 (10):1388–1400, 2009.

[17] A. Fernández, M. J. del Jesus i F. Herrera. Addressing overlapping in classification with imbalanced datasets: A first multi-objective approach for feature and instance selection. *Proceedings of the 16th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, stranice 36–44, 2015.

[18] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk i F. Herrera. *Learning from imbalanced data sets*, svezak 11. Springer, 2018.

[19] V. Garcia, J. S. Sanchez, R. A. Mollineda, R. Alejo i J. M. Sotoca. The class imbalance problem in pattern classification and learning. *Data Engineering*, 1:283–291, 2007.

[20] M. Gao, X. Hong, S. Chen i C. J. Harris. On combination of smote and particle swarm optimization based radial basis function classifier for imbalanced problems. *Proceedings of the 2011 International Joint Conference on Neural Networks (IJCNN)*, stranice 1146–1153, 2011.

[21] E. Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[22] S. B. Kotsiantis, I. Zaharakis i P. Pintelas. Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, 160(1):3–24, 2007.

[23] D. Gómez i A. Rojas. An empirical overview of the no free lunch theorem and its effect on real-world machine learning classification. *Neural Computation*, 28(1):216–228, 2016.

[24] S. Wang i X. Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4): 1119–1130, 2012.

[25] Q. Li, Y. Song, J. Zhang i V. S. Sheng. Multiclass imbalanced learning with one-versus-one decomposition and spectral clustering. *Expert Systems with Applications*, 147:113152, 2020.

[26] S. H. Ebenuwa, M. S. Sharif, A. Al-Nemrat, A. H. Al-Bayatti, N. Alalwan, A. I. Alzahrani i O. Alfarraj. Variance ranking for multi-classed imbalanced datasets: A case study of one-versus-all. *Symmetry*, 11(12):1504, 2019.

[27] L. Yijing, G. Haixiang, L. Xiao, L. Yanan i L. Jinling. Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data. *Knowledge-Based Systems*, 94:88–104, 2016.

[28] M. Sokolova i G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[29] X. Guo, Y. Yin, C. Dong, G. Yang i G. Zhou. On the class imbalance problem. *Proceedings of the 4th International Conference on Natural Computation (ICNC)*, svezak 4, stranice 192–201, 2008.

[30] R. Longadge i S. Dongre. Class imbalance problem in data mining review, 2013.

[31] A. Ali, S. M. Shamsuddin i A. L. Ralescu. Classification with class imbalance problem. *International Journal of Advances in Soft Computing and its Applications*, 5(3), 2013.

[32] D. A. Cieslak, T. R. Hoens, N. V. Chawla i W. P. Kegelmeyer. Hellinger distance decision trees are robust and skew-insensitive. *Data Mining and Knowledge Discovery*, 24(1):136–158, 2012.

[33] C. L. Castro i A. P. Braga. Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 24(6):888–899, 2013.

[34] R. Batuwita i V. Palade. Class imbalance learning methods for support vector machines. *Imbalanced learning: Foundations, Algorithms, and Applications*, stranice 83–99, 2013.

[35] W. Liu i S. Chawla. Class confidence weighted knn algorithms for imbalanced data sets. *Proceedings of the 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, stranice 345–356, 2011.

[36] M. D. Pérez-Godoy, A. Fernández, A. J. Rivera i M. J. del Jesus. Analysis of an evolutionary rbfn design algorithm, co2rbfn, for imbalanced data sets. *Pattern Recognition Letters*, 31(15):2375–2388, 2010.

[37] M. D. Pérez-Godoy, A. J. Rivera, C. J. Carmona i M. J. del Jesus. Training algorithms for radial basis function networks to tackle learning processes with imbalanced datasets. *Applied Soft Computing*, 25:26–39, 2014.

[38] J. M. Johnson i T. M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[39] A. Anaissi, P. J. Kennedy i M. Goyal. Feature selection of imbalanced gene expression microarray data. *Proceedings of the 12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, stranice 73–78, 2011.

[40] S. Fotouhi, S. Asadi i M. W. Kattan. A comprehensive data level analysis for cancer diagnosis on imbalanced data. *Journal of Biomedical Informatics*, 90:103089, 2019.

[41] H. Ye, L. Xiang i Y. Gan. Detecting financial statement fraud using random forest with smote. *Proceedings of the IOP Conference Series: Materials Science and Engineering*, stranica 052051, 2019.

[42] I. Brown i C. Mues. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39(3):3446–3453, 2012.

[43] S. Wang i X. Yao. Using class imbalance learning for software defect prediction. *IEEE Transactions on Reliability*, 62(2):434–443, 2013.

[44] Z. Zheng, X. Wu i R. Srihari. Feature selection for text categorization on imbalanced data. *ACM Sigkdd Explorations Newsletter*, 6(1):80–89, 2004.

[45] K. Ghosh, A. Banerjee, S. Chatterjee i S. Sen. Imbalanced twitter sentiment analysis using minority oversampling. *Proceedings of the 10th International Conference on Awareness Science and Technology (iCAST)*, stranice 1–5, 2019.

[46] Y. Wu, D. Guo, H. Liu i Y. Huang. An end-to-end learning method for industrial defect detection. *Assembly Automation*, 40(1):31–39, 2019.

[47] H. Yi, Q. Jiang, X. Yan i B. Wang. Imbalanced classification based on minority clustering smote with wind turbine fault detection application. *IEEE Transactions on Industrial Informatics*, 17(9):5867 – 5875, 2020.

[48] T. B. Trafalis, I. Adrianto, M. B. Richman i S. Lakshmivarahan. Machine-learning classifiers for imbalanced tornado data. *Computational Management Science*, 11(4): 403–418, 2014.

[49] A. Orriols-Puig i E. Bernadó-Mansilla. Evolutionary rule-based systems for imbalanced data sets. *Soft Computing*, 13(3):213–225, 2009.

[50] S. Gupta i A. Gupta. Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science*, 161:466–474, 2019.

[51] R. C. Holte, L. Acker, B. W. Porter et al. Concept learning and the problem of small disjuncts. *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI)*, svezak 89, stranice 813–818, 1989.

[52] M. Dudjak i G. Martinović. An empirical study of data intrinsic characteristics that make learning from imbalanced data difficult. *Expert Systems with Applications*, 182: 115297, 2021.

[53] V. García, J. Sánchez i R. Mollineda. An empirical study of the behavior of classifiers on imbalanced and overlapped data sets. *Proceedings of the 12th Iberoamerican Congress on Pattern Recognition (CIARP)*, stranice 397–406, 2007.

[54] M. Denil i T. Trappenberg. Overlap versus imbalance. *Proceedings of the 23th Canadian Conference on Artificial Intelligence (Canadian AI)*, stranice 220–231, 2010.

[55] R. C. Prati, G. E. Batista i M. C. Monard. Class imbalances versus class overlapping: an analysis of a learning system behavior. *Proceedings of the 3rd Mexican International Conference on Artificial Intelligence (MICAI)*, stranice 312–321, 2004.

[56] T. Imam, K. M. Ting i J. Kamruzzaman. z-svm: An svm for improved classification of imbalanced data. *Proceedings of the 19th Australian Joint Conference on Artificial Intelligence*, stranice 264–273, 2006.

[57] M. R. Smith, T. Martinez i C. Giraud-Carrier. An instance level analysis of data complexity. *Machine Learning*, 95(2):225–256, 2014.

[58] C.-F. Lin i S.-D. Wang. Fuzzy support vector machines. *IEEE Transactions on Neural Networks*, 13(2):464–471, 2002.

[59] S. S. Mullick, S. Datta i S. Das. Adaptive learning-based k-nearest neighbor classifiers with resilience to class imbalance. *IEEE Transactions on Neural Networks and Learning Systems*, 29(11):5713–5725, 2018.

[60] H. Han i B. Mao. Fuzzy-rough k-nearest neighbor algorithm for imbalanced data sets learning. *Proceedings of the 7th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, svezak 3, stranice 1286–1290, 2010.

[61] A. Cano, A. Zafra i S. Ventura. Weighted data gravitation classification for standard and imbalanced data. *IEEE Transactions on Cybernetics*, 43(6):1672–1687, 2013.

[62] P. Lenca, S. Lallich, T.-N. Do i N.-K. Pham. A comparison of different off-centered entropies to deal with class imbalance for decision trees. *Proceedings of the 12th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, stranice 634–643, 2008.

[63] K. Boonchuay, K. Sinapiromsaran i C. Lursinsap. Decision tree induction based on minority entropy for the class imbalance problem. *Pattern Analysis and Applications*, 20(3):769–782, 2017.

[64] P. Branco, L. Torgo i R. P. Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys*, 49(2):1–50, 2016.

[65] B. Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[66] C. Ferri, P. Flach i J. Hernández-Orallo. Learning decision trees using the area under the roc curve. *Proceedings of the 19th International Conference on Machine Learning (ICML)*, svezak 2, stranice 139–146, 2002.

[67] H. Narasimhan i S. Agarwal. Support vector algorithms for optimizing the partial area under the roc curve. *Neural Computation*, 29(7):1919–1963, 2017.

[68] D. Mease, A. J. Wyner i A. Buja. Boosted classification trees and class probability/quantile estimation. *Journal of Machine Learning Research*, 8(3), 2007.

[69] N. V. Chawla, K. W. Bowyer, L. O. Hall i W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[70] A. Fernández, S. Garcia, F. Herrera i N. V. Chawla. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, 61:863–905, 2018.

[71] P. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14(3):515–516, 1968.

[72] I. Tomek. Two modifications of cnn. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(11):769–772, 1976.

[73] J. Laurikkala. Improving identification of difficult small classes by balancing class distribution. *Proceedings of the 8th Conference on Artificial Intelligence in Medicine in Europe (AIME)*, stranice 63–66, 2001.

[74] M. Dudjak i G. Martinović. In-depth performance analysis of smote-based oversampling algorithms in binary classification. *International Journal of Electrical and Computer Engineering Systems*, 11(1):13–23, 2020.

[75] D. Bajer, M. Dudjak i B. Zorić. Wrapper-based feature selection: how important is the wrapped classifier? *Proceedings of the 2020 International Conference on Smart Systems and Technologies (SST)*, stranice 97–105, 2020.

[76] L. I. Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014.

[77] J. Błaszczyński, M. Deckert, J. Stefanowski i S. Wilk. Integrating selective preprocessing of imbalanced data with ivotes ensemble. *Proceedings of the 7th International Conference on Rough Sets and Current Trends in Computing (RSCTC)*, stranice 148–157, 2010.

[78] N. V. Chawla, A. Lazarevic, L. O. Hall i K. W. Bowyer. Smoteboost: Improving prediction of the minority class in boosting. *Proceedings of the 7th European Conference on Principles and Practice on Knowledge Discovery in Databases (PKDD)*, stranice 107–119, 2003.

[79] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse i A. Napolitano. Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1):185–197, 2009.

[80] S. Wang i X. Yao. Relationships between diversity of classification ensembles and single-class performance measures. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):206–219, 2011.

[81] R. Blagus i L. Lusa. Smote for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 14(106), 2013.

[82] G. Forman i I. Cohen. Learning from little: Comparison of classifiers given little training. *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery (PKDD)*, stranice 161–172, 2004.

[83] G.-H. Fu, Y.-J. Wu, M.-J. Zong i L.-Z. Yi. Feature selection and classification by minimizing overlap degree for class-imbalanced data in metabolomics. *Chemometrics and Intelligent Laboratory Systems*, 196:103906, 2020.

[84] M. Makrehchi i M. S. Kamel. Impact of term dependency and class imbalance on the performance of feature ranking methods. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(07):953–983, 2011.

[85] S.-J. Yen i Y.-S. Lee. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. *Intelligent Control and Automation*, stranice 731–740. Springer, 2006.

[86] D. Devi, S. K. Biswas i B. Purkayastha. A review on solution to class imbalance problem: Undersampling approaches. *Proceedings of the 2020 International Conference on Computational Performance Evaluation (ComPE)*, stranice 626–631, 2020.

[87] T. Jo i N. Japkowicz. Class imbalances versus small disjuncts. *ACM Sigkdd Explorations Newsletter*, 6(1):40–49, 2004.

[88] R. C. Prati, G. E. Batista i M. C. Monard. Learning with class skews and small disjuncts. *Proceedings of the 17th Brazilian Symposium on Artificial Intelligence (SBIA)*, stranice 296–306, 2004.

[89] A. Hamdy i A. El-Laithy. Smote and feature selection for more effective bug severity prediction. *International Journal of Software Engineering and Knowledge Engineering*, 29(06):897–919, 2019.

[90] G. Kovács. An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. *Applied Soft Computing*, 83:105662, 2019.

[91] D. Bajer, B. Zorić, M. Dudjak i G. Martinović. Performance analysis of smote-based oversampling techniques when dealing with data imbalance. *Proceedings of the 2019 International Conference on Systems, Signals and Image Processing (IWSSIP)*, stranice 265–271, 2019.

[92] J. Tang, S. Alelyani i H. Liu. Feature selection for classification: A review. *Data Classification: Algorithms and Applications*, stranica 37, 2014.

[93] A. Jović, K. Brkić i N. Bogunović. A review of feature selection methods with applications. *Proceedings of the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, stranice 1200–1205, 2015.

[94] I. Guyon i A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[95] N. Sánchez-Maroño, A. Alonso-Betanzos i M. Tombilla-Sanromán. Filter methods for feature selection–a comparative study. *Proceedings of the 8th International Conference on Intelligent Data Engineering and Automated Learning*, stranice 178–187, 2007.

[96] R. Kohavi i G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[97] G. Martinović, D. Bajer i B. Zorić. A differential evolution approach to dimensionality reduction for classification needs. *International Journal of Applied Mathematics and Computer Science*, 24(1), 2014.

[98] L. C. Molina, L. Belanche i À. Nebot. Feature selection algorithms: A survey and experimental evaluation. *Proceedings of the 2002 International Conference on Data Mining (ICDM)*, stranice 306–313, 2002.

[99] B. Xue, M. Zhang, W. N. Browne i X. Yao. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20 (4):606–626, 2015.

[100] P. Pudil, J. Novovičová i J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119–1125, 1994.

[101] G. Chandrashekar i F. Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[102] J. Apolloni, G. Leguizamón i E. Alba. Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. *Applied Soft Computing*, 38:922–932, 2016.

[103] B. Zorić, D. Bajer i G. Martinović. Utilising filter inferred information in nature-inspired hybrid feature selection. *Proceedings of the 2018 International Conference on Smart Systems and Technologies (SST)*, stranice 117–123, 2018.

[104] I. Guyon, J. Weston, S. Barnhill i V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, 2002.

[105] B. H. Cho, H. Yu, K.-W. Kim, T. H. Kim, I. Y. Kim i S. I. Kim. Application of irregular and unbalanced data to predict diabetic nephropathy using visualization and feature selection methods. *Artificial Intelligence in Medicine*, 42(1):37–53, 2008.

[106] D. Bajer, B. Zorić, M. Dudjak i G. Martinović. Evaluation and analysis of bio-inspired optimization algorithms for feature selection. *Proceedings of the 15th International Scientific Conference on Informatics*, stranice 285–292, 2019.

[107] J. Cai, J. Luo, S. Wang i S. Yang. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79, 2018.

[108] D. Karaboga i B. Akay. A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 214(1):108–132, 2009.

[109] S. Swayamsiddha. Bio-inspired algorithms: principles, implementation, and applications to wireless communication. *Nature-Inspired Computation and Swarm Intelligence*, stranice 49–63. Elsevier, 2020.

[110] J. Del Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P. N. Suganthan, C. A. C. Coello i F. Herrera. Bio-inspired computation: Where we stand and what's next. *Swarm and Evolutionary Computation*, 48:220–250, 2019.

[111] W. Siedlecki i J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recogniton Letters*, 10(5):335–347, 1989.

[112] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn i A. K. Jain. Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4(2):164–171, 2000.

[113] F. Tan, X. Fu, Y. Zhang i A. G. Bourgeois. A genetic algorithm-based method for feature subset selection. *Soft Computing*, 12(2):111–120, 2008.

[114] S.-F. Yuan i F.-L. Chu. Fault diagnostics based on particle swarm optimisation and support vector machines. *Mechanical Systems and Signal Processing*, 21(4):1787–1798, 2007.

[115] R. N. Khushaba, A. Al-Ani i A. Al-Jumaily. Differential evolution based feature subset selection. *Proceedings of the 19th International Conference on Pattern Recognition*, stranice 1–4, 2008.

[116] Y. Marinakis, M. Marinaki, N. Matsatsinis i C. Zopounidis. Discrete artificial bee colony optimization algorithm for financial classification problems. *Trends in Developing Metaheuristics, Algorithms, and Optimization Approaches*, stranice 44–58, 2013.

[117] B. Zorić, D. Bajer i M. Dudjak. Wrapper-based feature selection via differential evolution: benchmarking different discretisation techniques. *Proceedings of the 2020 International Conference on Smart Systems and Technologies (SST)*, stranice 89–96, 2020.

[118] B. Xue, M. Zhang i W. N. Browne. Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing*, 18:261–276, 2014.

[119] E. Emary, H. M. Zawbaa, C. Grosan i A. E. Hassenian. Feature subset selection approach by gray-wolf optimization. *Proceedings of the 1st International Afro-European Conference for Industrial Advancement (AECIA)*, stranice 1–13, 2015.

[120] E. Emary, H. M. Zawbaa i A. E. Hassanien. Binary ant lion approaches for feature selection. *Neurocomputing*, 213:54–65, 2016.

[121] E. Hancer, B. Xue i M. Zhang. A differential evolution based feature selection approach using an improved filter criterion. *Proceedings of the 2017 Symposium Series on Computational Intelligence (SSCI)*, stranice 1–8, 2017.

[122] E. Hancer, B. Xue i M. Zhang. Differential evolution for filter feature selection based on information theory and feature ranking. *Knowledge-Based Systems*, 140:103–119, 2018.

[123] A. Deniz i H. E. Kiziloz. On initial population generation in feature subset selection. *Expert Systems with Applications*, 137:11–21, 2019.

[124] I.-S. Oh, J.-S. Lee i B.-R. Moon. Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1424–1437, 2004.

[125] B. Tran, B. Xue i M. Zhang. Overview of particle swarm optimisation for feature selection in classification. *Proceedings of the 10th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL)*, stranice 605–617, 2014.

[126] H. B. Nguyen, B. Xue, I. Liu i M. Zhang. Filter based backward elimination in wrapper based pso for feature selection in classification. *Proceedings of the 2014 Congress on Evolutionary Computation (CEC)*, stranice 3111–3118, 2014.

[127] Y.-S. Jeong, K. S. Shin i M. K. Jeong. An evolutionary algorithm with the partial sequential forward floating search mutation for large-scale feature selection problems. *Journal of The Operational Research Society*, 66(4):529–538, 2015.

[128] E. Hancer. Differential evolution for feature selection: a fuzzy wrapper–filter approach. *Soft Computing*, 23(13):5233–5248, 2019.

[129] M. C. Lane, B. Xue, I. Liu i M. Zhang. Particle swarm optimisation and statistical clustering for feature selection. *Proceedings of the 26th Australian Joint Conference on Artificial Intelligence (AI)*, stranice 214–220, 2013.

[130] M. Rostami, K. Berahmand i S. Forouzandeh. A novel community detection based genetic algorithm for feature selection. *Journal of Big Data*, 8(1):1–27, 2021.

[131] A. K. Das, S. Sengupta i S. Bhattacharyya. A group incremental feature selection for classification using rough set theory based genetic algorithm. *Applied Soft Computing*, 65:400–411, 2018.

[132] L.-Y. Chuang, H.-W. Chang, C.-J. Tu i C.-H. Yang. Improved binary pso for feature selection using gene expression data. *Computational Biology and Chemistry*, 32(1): 29–38, 2008.

[133] C.-S. Yang, L.-Y. Chuang, C.-H. Ke i C.-H. Yang. Boolean binary particle swarm optimization for feature selection. *Proceedings of the 2008 Congress on Evolutionary Computation (CEC)*, stranice 2093–2098, 2008.

[134] I. P. Benitez, A. M. Sison i R. P. Medina. An improved genetic algorithm for feature selection in the classification of disaster-related twitter messages. *Proceedings of the 2018 Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, stranice 238–243, 2018.

[135] J. Rajpurohit, T. K. Sharma, A. Abraham i A. Vaishali. Glossary of metaheuristic algorithms. *International Journal of Computer Information Systems and Industrial Management Applications*, 9:181–205, 2017.

[136] M. Nemati, H. Momeni i N. Bazrkar. Binary black holes algorithm. *International Journal of Computer Applications*, 79(6), 2013.

[137] S. Mirjalili i A. Lewis. The whale optimization algorithm. *Advances in Engineering Software*, 95:51–67, 2016.

[138] G. Dhiman i V. Kumar. Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. *Knowledge-Based Systems*, 159:20–50, 2018.

[139] M. A. Lones. Metaheuristics in nature-inspired algorithms. *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation (GECCO)*, stranice 1419–1422, 2014.

[140] K. Sörensen. Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*, 22(1):3–18, 2015.

[141] C. L. Camacho-Villalón, M. Dorigo i T. Stützle. Why the intelligent water drops cannot be considered as a novel algorithm. *Proceedings of the 11th International Conference on Swarm Intelligence (ANTS)*, stranice 302–314, 2018.

[142] D. Bajer, B. Zorić, M. Dudjak i G. Martinović. Benchmarking bio-inspired computation algorithms as wrappers for feature selection. *Acta Electrotechnica et Informatica*, 20: 35–43, 2020.

[143] J. Loughrey i P. Cunningham. Overfitting in wrapper-based feature subset selection: The harder you try the worse it gets. *Proceedings of the 24th International Conference on Innovative Techniques and Applications of Artificial Intelligence (SGAI)*, stranice 33–43, 2004.

[144] J. Loughrey i P. Cunningham. Using early-stopping to avoid overfitting in wrapper-based feature selection employing stochastic search. Technical report, Trinity College Dublin, Department of Computer Science, 2005.

[145] Q. Al-Tashi, S. J. Abdulkadir, H. M. Rais, S. Mirjalili i H. Alhussian. Approaches to multi-objective feature selection: A systematic literature review. *IEEE Access*, 8: 125076–125096, 2020.

[146] U. M. Khaire i R. Dhanalakshmi. Stability of feature selection algorithm: A review. *Journal of King Saud University-Computer and Information Sciences*, 2019.

[147] K. Dunne, P. Cunningham i F. Azuaje. Solutions to instability problems with sequential wrapper-based approaches to feature selection. *Journal of Machine Learning Research*, stranice 1–22, 2002.

[148] A. Kalousis, J. Prados i M. Hilario. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and Information Systems*, 12(1):95–116, 2007.

[149] S. S. Mousaabadi. *Evolutionary computation-based feature selection for finding a stable set of features in high-dimensional data.* Nottingham Trent University, United Kingdom, 2019.

[150] V. Bolón-Canedo, N. Sánchez-Maroño i A. Alonso-Betanzos. Distributed feature selection: An application to microarray data classification. *Applied Soft Computing*, 30: 136–150, 2015.

[151] V. Bolón-Canedo i A. Alonso-Betanzos. Ensembles for feature selection: A review and future trends. *Information Fusion*, 52:1–12, 2019.

[152] K. Bache i M. Lichman. Uci machine learning repository, 2013.

[153] M. S. Wibawa, H. A. Nugroho i N. A. Setiawan. Performance evaluation of combined feature selection and classification methods in diagnosing parkinson disease based on voice feature. *Proceedings of the 2015 International Conference on Science in Information Technology (ICSITech)*, stranice 126–131, 2015.

[154] N. Suchetha, A. Nikhil i P. Hrudya. Comparing the wrapper feature selection evaluators on twitter sentiment classification. *Proceedings of the 2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, stranice 1–6, 2019.

[155] R. Scitovski, M. Vinković, K. Sabo i A. Kozić. A research project ranking method based on independent reviews by using the principle of the distance to the perfectly assessed project. *Croatian Operational Research Review*, stranice 429–442, 2017.

[156] J. Derrac, S. García, D. Molina i F. Herrera. A practical tutorial on the use of non-parametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.

[157] J. Hernandez, J. A. Carrasco-Ochoa i J. F. Martínez-Trinidad. An empirical study of oversampling and undersampling for instance selection methods on imbalance datasets. *Proceedings of the 18th Iberoamerican Congress on Pattern Recognition (CIARP)*, stranice 262–269, 2013.

[158] M. Koziarski. Radial-based undersampling for imbalanced data classification. *Pattern Recognition*, 102:107262, 2020.

[159] B. Das, N. C. Krishnan i D. J. Cook. Racog and wracog: Two probabilistic oversampling techniques. *IEEE Transactions on Knowledge and Data Engineering*, 27(1): 222–234, 2014.

[160] V. García, J. S. Sánchez, A. Marqués, R. Florencia i G. Rivera. Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data. *Expert Systems with Applications*, 158:113026, 2020.

[161] P. Kaur i A. Gosain. Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise. *ICT Based Innovations*, svezak 653, stranice 23–30. Springer, 2018.

[162] Y. Dong i X. Wang. A new over-sampling approach: random-smote for learning from imbalanced data sets. *Proceedings of the 5th International Conference on Knowledge Science, Engineering and Management (KSEM)*, stranice 343–352, 2011.

[163] Z. Zheng, Y. Cai i Y. Li. Oversampling method for imbalanced classification. *Computing and Informatics*, 34(5):1017–1037, 2015.

[164] J. Stefanowski i S. Wilk. Improving rule based classifiers induced by modlem by selective pre-processing of imbalanced data. *Proceedings of the 2007 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, stranice 54–65, 2007.

[165] C. Bellinger, C. Drummond i N. Japkowicz. Beyond the boundaries of smote. *Proceedings of the 2016 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, stranice 248–263, 2016.

[166] T. Maciejewski i J. Stefanowski. Local neighbourhood extension of smote for mining imbalanced data. *Proceedings of the 2011 Symposium on Computational Intelligence and Data Mining (CIDM)*, stranice 104–111, 2011.

[167] K. Napierala i J. Stefanowski. Types of minority class examples and their influence on learning classifiers from imbalanced data. *Journal of Intelligent Information Systems*, 46(3):563–597, 2016.

[168] M. Koziarski, B. Krawczyk i M. Woźniak. Radial-based approach to imbalanced data oversampling. *Proceedings of the 12th International Conference on Hybrid Artificial Intelligence Systems (HAIS)*, stranice 318–327, 2017.

[169] H. Han, W.-Y. Wang i B.-H. Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. *Proceedings of the 2005 International Conference on Intelligent Computing (ICIC)*, stranice 878–887, 2005.

[170] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse i A. Folleco. An empirical study of the classification performance of learners on imbalanced and noisy software quality data. *Information Sciences*, 259:571–595, 2014.

[171] C. Bunkhumpornpat, K. Sinapiromsaran i C. Lursinsap. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. *Proceedings of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, stranice 475–482, 2009.

[172] R. Das, S. K. Biswas, D. Devi i B. Sarma. An oversampling technique by integrating reverse nearest neighbor in smote: Reverse-smote. *Proceedings of the 2020 International Conference on Smart Electronics and Communication (ICOSEC)*, stranice 1239–1244, 2020.

[173] H. He, Y. Bai, E. A. Garcia i S. Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. *Proceedings of the 2008 International Joint Conference on Neural Networks (IJCNN)*, stranice 1322–1328, 2008.

[174] F. R. Torres, J. A. Carrasco-Ochoa i J. F. Martínez-Trinidad. Smote-d a deterministic version of smote. *Proceedings of the 8th Mexican Conference on Pattern Recognition (MCPR)*, stranice 177–188, 2016.

[175] M. R. Prusty, T. Jayanthi i K. Velusamy. Weighted-smote: A modification to smote for event classification in sodium cooled fast reactors. *Progress in Nuclear Energy*, 100: 355–364, 2017.

[176] J. De La Calleja i O. Fuentes. A distance-based over-sampling method for learning from imbalanced data sets. *Proceedings of the 20th International Florida Artificial Intelligence Research Society Conference*, stranice 634–635, 2007.

[177] S. Gazzah i N. E. B. Amara. New oversampling approaches based on polynomial fitting for imbalanced data sets. *Proceedings of the 8th International Workshop on Document Analysis Systems (IAPR)*, stranice 677–684, 2008.

[178] G. Douzas i F. Bacao. Geometric smote a geometrically enhanced drop-in replacement for smote. *Information Sciences*, 501:118–135, 2019.

[179] G. A. Pradipta, R. Wardoyo, A. Musdholifah i I. N. H. Sanjaya. Radius-smote: A new oversampling technique of minority samples based on radius distance for learning from imbalanced data. *IEEE Access*, 9:74763–74777, 2021.

[180] G. Cohen, M. Hilario, H. Sax, S. Hugonnet i A. Geissbuhler. Learning from imbalanced data in surveillance of nosocomial infection. *Artificial Intelligence in Medicine*, 37(1): 7–18, 2006.

[181] C. Bunkhumpornpat, K. Sinapiromsaran i C. Lursinsap. Dbsmote: density-based synthetic minority over-sampling technique. *Applied Intelligence*, 36(3):664–684, 2012.

[182] F. Last, G. Douzas i F. Bacao. Oversampling for imbalanced learning based on k-means and smote. *Information Sciences*, 465:1–20, 2017.

[183] K. Puntumapon, T. Rakthamamon i K. Waiyamai. Cluster-based minority over-sampling for imbalanced datasets. *IEICE Transactions on Information and Systems*, 99(12):3101–3109, 2016.

[184] J. De La Calleja, O. Fuentes i J. González. Selecting minority examples from misclassified data for over-sampling. *Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, stranice 276–281, 2008.

[185] E. Ramentol, Y. Caballero, R. Bello i F. Herrera. Smote-rs b*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory. *Knowledge and Information Systems*, 33(2):245–265, 2012.

[186] F. Hu i H. Li. A novel boundary oversampling algorithm based on neighborhood rough set model: Nrsboundary-smote. *Mathematical Problems in Engineering*, 2013, 2013.

[187] V. López, I. Triguero, C. J. Carmona, S. García i F. Herrera. Addressing imbalanced classification with instance generation techniques: Ipade-id. *Neurocomputing*, 126: 15–28, 2014.

[188] H. Al Majzoub, I. Elgedawy, Ö. Akaydın i M. K. Ulukök. Hcab-smote: A hybrid clustered affinitive borderline smote approach for imbalanced data binary classification. *Arabian Journal for Science and Engineering*, 45(4):3205–3222, 2020.

[189] Y. Yan, Y. Jiang, Z. Zheng, C. Yu, Y. Zhang i Y. Zhang. Ldas: Local density-based adaptive sampling for imbalanced data classification. *Expert Systems with Applications*, stranica 116213, 2021.

[190] P. Cao, X. Liu, J. Zhang, D. Zhao, M. Huang i O. Zaiane. l2, 1 norm regularized multi-kernel based joint nonlinear feature selection and over-sampling for imbalanced data classification. *Neurocomputing*, 234:38–57, 2017.

[191] Z. Xie, L. Jiang, T. Ye i X. Li. A synthetic minority oversampling method based on local densities in low-dimensional space for imbalanced learning. *Proceedings of the 20th International Conference on Database Systems for Advanced Applications (DASFAA)*, stranice 3–18, 2015.

[192] T. Deepa i M. Punithavalli. An e-smote technique for feature selection in high-dimensional imbalanced dataset. *Proceedings of the 3rd International Conference on Electronics Computer Technology*, svezak 2, stranice 322–324, 2011.

[193] N. Moniz i H. Monteiro. No free lunch in imbalanced learning. *Knowledge-Based Systems*, stranica 107222, 2021.

[194] N. V. Chawla, D. A. Cieslak, L. O. Hall i A. Joshi. Automatically countering imbalance and its empirical relationship to cost. *Data Mining and Knowledge Discovery*, 17(2): 225–252, 2008.

[195] B. Zorić, D. Bajer i G. Martinović. Employing different optimisation approaches for smote parameter tuning. *Proceedings of the 2016 International Conference on Smart Systems and Technologies (SST)*, stranice 191–196, 2016.

[196] J. D. Pascual-Triana, D. Charte, M. A. Arroyo, A. Fernández i F. Herrera. Revisiting data complexity metrics based on morphology for overlap and imbalance: snapshot, new overlap number of balls metrics and singular problems prospect. *Knowledge and Information Systems*, 63:1961–1989, 2021.

[197] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez i F. Herrera. Keel data-mining software tool: data set repository, integration of algorithms and

experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17, 2011.

[198] S. Hu, Y. Liang, L. Ma i Y. He. Msmote: Improving classification performance when training data is imbalanced. *Proceedings of the 2nd International Workshop on Computer Science and Engineering*, svezak 2, stranice 13–17, 2009.

[199] S. Barua, M. M. Islam i K. Murase. A novel synthetic minority oversampling technique for imbalanced data set learning. *Proceedings of the 18th International Conference on Neural Information Processing (ICONIP)*, stranice 735–744, 2011.

[200] L. Zhang i W. Wang. A re-sampling method for class imbalance learning with credit data. *Proceedings of the 2011 International Conference of Information Technology, Computer Engineering and Management Sciences (ICM)*, svezak 1, stranice 393–397, 2011.

[201] F. Koto. Smote-out, smote-cosine, and selected-smote: An enhancement strategy to handle imbalance in data level. *Proceedings of the 2014 International Conference on Advanced Computer Science and Information System (ICACSIS)*, stranice 280–284, 2014.

[202] C. Drummond, R. C. Holte et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. *Proceedings of the 2nd Workshop on Learning from Imbalanced Datasets (ICML)*, svezak 11, stranice 1–8, 2003.

[203] M. Gao, X. Hong, S. Chen i C. J. Harris. A combined smote and pso based rbf classifier for two-class imbalanced problems. *Neurocomputing*, 74(17):3456–3466, 2011.

[204] H. Li, D. Pi i C. Wang. The prediction of protein-protein interaction sites based on rbf classifier improved by smote. *Mathematical Problems in Engineering*, 2014, 2014.

[205] M. J. Powell. Radial basis functions for multivariable interpolation: a review. *Algorithms for Approximation*, 1987.

[206] D. Broomhead i D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2, 1988.

[207] J. Moody i C. Darken. *Learning with localized receptive fields*. Yale University, Department of Computer Science, 1988.

[208] M. T. Musavi, W. Ahmed, K. H. Chan, K. B. Faris i D. M. Hummels. On the training of radial basis function classifiers. *Neural Networks*, 5(4):595–603, 1992.

[209] S. W. Choi, D. Lee, J. H. Park i I.-B. Lee. Nonlinear regression using rbfn with linear submodels. *Chemometrics and Intelligent Laboratory Systems*, 65(2):191–208, 2003.

[210] B. A. Whitehead i T. D. Choate. Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction. *IEEE Transactions on Neural Networks*, 7(4):869–880, 1996.

[211] T. Poggio i S. Edelman. A network that learns to recognize three-dimensional objects. *Nature*, 343(6255):263–266, 1990.

[212] Y. Lee i R. P. Lippmann. Practical characteristics of neural network and conventional pattern classifiers on artificial and speech problems. *Proceedings of the 2nd International Conference on Neural Information Processing Systems (NIPS)*, stranice 168–177, 1989.

[213] S. A. Kassam i I. Cha. Radial basis function networks in nonlinear signal processing applications. *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, stranice 1021–1025, 1993.

[214] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.

[215] D. Bajer, B. Zorić i G. Martinović. Automatic design of radial basis function networks through enhanced differential evolution. *Proceedings of the 10th International Conference on Hybrid Artificial Intelligence Systems (HAIS)*, stranice 244–256, 2015.

[216] S. A. Billings i G. L. Zheng. Radial basis function network configuration using genetic algorithms. *Neural Networks*, 8(6):877–890, 1995.

[217] R. Kruse, C. Borgelt, C. Braune, S. Mostaghim, M. Steinbrecher, F. Klawonn i C. Moewes. *Computational intelligence*, svezak 1. Springer London, 2013.

[218] S.-H. Yoo, S.-K. Oh i W. Pedrycz. Optimized face recognition algorithm using radial basis function neural networks and its practical applications. *Neural Networks*, 69: 111–125, 2015.

[219] E. J. Hartman, J. D. Keeler i J. M. Kowalski. Layered neural networks with gaussian hidden units as universal approximations. *Neural Computation*, 2(2):210–215, 1990.

[220] P. Strumiłło i W. Kamiński. Radial basis function neural networks: theory and applications. *Neural Networks and Soft Computing*, stranice 107–119. Springer, 2003.

[221] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, 1965.

[222] M. Vogt. Combination of radial basis function neural networks with optimized learning vector quantization. *Proceedings of the 1993 International Conference on Neural Networks*, stranice 1841–1846, 1993.

[223] G. Bugmann. Normalized gaussian radial basis function networks. *Neurocomputing*, 20(1-3):97–110, 1998.

[224] M. Visani, C. Garcia i J.-M. Jolion. Normalized radial basis function networks and bilinear discriminant analysis for face recognition. *Proceedings of the 2005 Conference on Advanced Video and Signal Based Surveillance (AVSS)*, stranice 342–347, 2005.

[225] J. Qiao, X. Meng i W. Li. An incremental neuronal-activity-based rbf neural network for nonlinear system modeling. *Neurocomputing*, 302:1–11, 2018.

[226] O. Buchtala, M. Klimek i B. Sick. Evolutionary optimization of radial basis function classifiers for data mining applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(5):928–947, 2005.

[227] A. Albert. *Regression and the Moore-Penrose pseudoinverse*, svezak 94. Elsevier, 1972.

[228] D. Bajer. *Unaprjeđenja algoritma diferencijalne evolucije podešavanjem parametara i izborom početne populacije*. PhD thesis, Josip Juraj Strossmayer University of Osijek, 2017.

[229] D. R. Hush i B. G. Horne. Progress in supervised neural networks. *IEEE Signal Processing Magazine*, 10(1):8–39, 1993.

[230] M. Korürek i B. Doğan. Ecg beat classification using particle swarm optimization and radial basis function neural network. *Expert Systems with Applications*, 37(12): 7563–7569, 2010.

[231] Z. Qin, J. Chen, Y. Liu i J. Lu. Evolving rbf neural networks for pattern classification. *Proceedings of the International Conference on Computational and Information Science (CIS)*, stranice 957–964, 2005.

[232] R. M. Yousef i K. El-Hindi. *Locating Center Points for Radial Basis Function Networks Using Instance Reduction Techniques*. University of Jordan, 2004.

[233] M. W. Mak i K. W. Cho. Genetic evolution of radial basis function centers for pattern classification. *Proceedings of the 1998 International Joint Conference on Neural Networks Proceedings (IJCNN)*, svezak 1, stranice 669–673, 1998.

[234] G. Zheng i S. Billings. Radial basis function network training using a fuzzy clustering scheme. Technical report, University of Sheffield, 1994.

[235] B. Burdsall i C. Giraud-Carrier. Ga-rbf: a self-optimising rbf network. *Proceedings of the 1998 International Conference on Artificial Neural Nets and Genetic Algorithms*, stranice 346–349, 1998.

[236] D. Casasent i X.-W. Chen. New training strategies for rbf neural networks for x-ray agricultural product inspection. *Pattern Recognition*, 36(2):535–547, 2003.

[237] D. Lowe. Adaptive radial basis function nonlinearities, and the problem of generalisation. *Proceedings of the 1st International Conference on Artificial Neural Networks*, stranice 171–175, 1989.

[238] N. B. Karayiannis. Reformulated radial basis neural networks trained by gradient descent. *IEEE Transactions on Neural Networks*, 10(3):657–671, 1999.

[239] D. Wettschereck i T. Dietterich. Improving the performance of radial basis function networks by learning center locations. *Proceedings of the 4th International Conference on Neural Information Processing Systems (NIPS)*, svezak 4, stranice 1133–1140, 1991.

[240] T. Kurban i E. Beşdok. A comparison of rbf neural network training algorithms for inertial sensor based terrain classification. *Sensors*, 9(8):6312–6329, 2009.

[241] J. Lu, H. Hu i Y. Bai. Generalized radial basis function neural network based on an improved dynamic particle swarm optimization and adaboost algorithm. *Neurocomputing*, 152:305–315, 2015.

[242] W. Yu, L. Liu i W. Zhang. Traffic prediction method based on rbf neural network with improved artificial bee colony algorithm. *Proceedings of the 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, stranice 141–144, 2015.

[243] D. Bajer, B. Zorić i G. Martinović. Effectiveness of differential evolution in training radial basis function networks for classification. *Proceeding of the 2016 International Conference on Smart Systems and Technologies (SST)*, stranice 179–184, 2016.

[244] J. Platt. A resource-allocating network for function interpolation. *Neural Computation*, 3(2):213–225, 1991.

[245] S. Chen, C. Cowan i P. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2(2):302–309, 1991.

[246] A. Leonardis i H. Bischof. An efficient mdl-based construction of rbf networks. *Neural Networks*, 11(5):963–973, 1998.

[247] L. Yingwei, N. Sundararajan i P. Saratchandran. Performance evaluation of a sequential minimal radial basis function (rbf) neural network learning algorithm. *IEEE Transactions on Neural Networks*, 9(2):308–318, 1998.

[248] H.-G. Han, Q.-l. Chen i J.-F. Qiao. An efficient self-organizing rbf neural network for water quality prediction. *Neural Networks*, 24(7):717–725, 2011.

[249] H. Yu, P. D. Reiner, T. Xie, T. Bartczak i B. M. Wilamowski. An incremental design of radial basis function networks. *IEEE Transactions on Neural Networks and Learning Systems*, 25(10):1793–1803, 2014.

[250] T. Poggio i F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[251] J. A. Nelder i R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[252] J. Kennedy i R. Eberhart. Particle swarm optimization. *Proceedings of 1995 International Conference on Neural Networks (ICNN)*, svezak 4, stranice 1942–1948, 1995.

[253] K. R. Harrison, B. M. Ombuki-Berman i A. P. Engelbrecht. A parameter-free particle swarm optimization algorithm using performance classifiers. *Information Sciences*, 503:381–400, 2019.

[254] A. R. Conn, K. Scheinberg i L. N. Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.

[255] J. L. Lustgarten, V. Gopalakrishnan i S. Visweswaran. Measuring stability of feature selection in biomedical datasets. *Proceedings of the 2009 American Medical Informatics Association Annual Symposium*, svezak 2009, stranica 406, 2009.

[256] R Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

Sažetak

Učenje iz neuravnoteženih podataka odnosi se na problem nepovoljne izvedbe standardnih algoritama za klasifikaciju pri kategorizaciji manjinskih primjeraka, koji obično predstavljaju rijetke događaje od ključne važnosti u području iz kojeg proizlaze. S obzirom na složenost i rasprostranjenost ovog problema, nije iznenađujuće da postoje različite vrste pristupa za njegovo ublažavanje. Najprikladniji jesu postupci predobrade skupa podataka, poput odabira značajki i preuzorkovanja, koji smanjuju složenost problema te klasifikator radijalne neuronske mreže (RBFN). Iako se u literaturi mogu naći brojne učinkovite inačice ovih postupaka, u okviru svake postoje određeni nedostaci. U ovoj disertaciji predložena su unaprjeđenja tih postupaka. Predloženo je proširenje bio-inspiriranih omotača koje se zasniva na prikupljanju kvalitetnih rješenja tijekom pretrage te njihovu naknadnom objedinjavanju prema doprinosu kvaliteti. Prikupljanjem takvih rješenja izbjegava se oslanjanje isključivo na jedno rješenje nađeno omotačem koje može biti pretjerano prilagođeno skupu korištenom za vrednovanje, a njihovim objedinjavanjem nastoje se prepoznati relevantne značajke za promatrani problem. Također, predstavljeno je unaprjeđenje algoritma SMOTE koje određuje susjedstva manjinskih primjeraka te broj stvorenih sintetičkih primjeraka uzimajući u obzir unutarnje karakteristike skupa podataka. Time je uklonjena potreba za parametrima koji kontroliraju njegov rad, što ga čini trivijalnim za korištenje u odnosu na algoritam SMOTE, a posebno u odnosu na mnoge njegove unaprijeđene inačice. Konačno, predložen je novi postupak izgradnje klasifikacijskih modela RBFN koji nastoji pronaći slijed mreža povoljnih performansi klasifikacije postupnim treniranjem mreža sve većeg stupnja složenosti, uz korištenje znanja iz prethodno treniranih mreža manje složenosti. S ciljem što učinkovitijeg iskorištenja tog znanja, prethodnoj mreži nadodaje se čvor koji ima povoljne interakcije s postojećim čvorovima. Predložena unaprjeđenja opsežno su testirana na standardnim skupovima podataka koji predstavljaju razne probleme klasifikacije različitog stupnja neuravnoteženosti klasa. Analiza ostvarenih rezultata dovela je do zaključka da predložena unaprjeđenja pospješuju prepoznavanje manjinske klase i time poboljšavaju opću uspješnost klasifikacije. Osim toga, utvrđeno je da u tom pogledu nadmašuju uobičajene, ali i unaprijeđene inačice tih postupaka iz literature.

Ključne riječi: bio-inspirirani algoritmi optimizacije, klasifikacija, neuravnoteženost klasa, odabir značajki, preuzorkovanje, radijalna neuronska mreža, SMOTE

Abstract

Learning from imbalanced data through improved approaches for feature selection, oversampling and designing radial basis function networks

Learning from imbalanced data refers to the problem of inadequate performance of standard classification algorithms when categorising minority instances, which usually represent rare events of critical importance in the problem domain. Given the complexity and prevalence of this problem, it is not surprising that various approaches to alleviating it are available. The most appropriate are preprocessing procedures, such as feature selection and oversampling, that reduce the problem complexity and the radial basis function network (RBFN) classifier. Although many effective variants of these procedures can be found in the literature, there are certain shortcomings within each. Improvements to these procedures are proposed in the thesis. An extension of bio-inspired wrappers that is based on storing solutions found during search and their subsequent combination according to the impact on classification performance is proposed. Reliance solely on one solution found by the wrapper, which may be overly fitted to the data used for validation, is avoided by storing multiple solutions and combining them in order to identify relevant features for the problem at hand. Also, an improvement of the SMOTE algorithm which determines the neighbourhoods of minority instances and the number of synthetic instances to be created by considering the intrinsic data characteristics is presented. This eliminates the need for parameters that control its operation, rendering it trivial for use compared to the SMOTE algorithm, and especially compared to many of its improved versions. Finally, a new procedure for designing classification models based on RBFN is proposed which aims to find a sequence of networks of favourable classification performance by gradually training networks of increasing complexity using knowledge from previously trained, less complex, networks. To make the most of this knowledge, a node that has beneficial interactions with existing ones is added to the network obtained in the previous step. The proposed improvements were extensively tested on standard datasets that represent various classification problems of different degrees of class imbalance. The analysis of the obtained results led to the conclusion that the proposed improvements enhance the recognition of the minority class and thus improve overall classification performance. Furthermore, it was shown that they outperform common, but also improved variants of these procedures available in the literature.

Keywords: bio-inspired optimisation algorithms, classification, class imbalance, feature selection, oversampling, radial basis function network, SMOTE

Životopis

Mario Dudjak rođen je 25.3.1995. godine u Našicama. Ondje završava osnovnu i srednju školu nakon čega upisuje Elektrotehnički fakultet u Osijeku (sada Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek). Akademski naziv prvostupnik inženjer računarstva stječe na istoj instituciji 2016., a akademski naziv magistar inženjer računarstva 2018. godine. Tijekom studija sudjeluje na raznim natjecanjima u području računarstva, poput IEEE Extreme, Elektrijada i IEEE MADC. U dva navrata nagrađen je dekanovim priznanjem za uspjeh tijekom studija. Uz fakultet, polazi i završava PHP akademiju organiziranu od strane fakulteta i tvrtke Inchoo, te *Microsoft Professional Degree Data Science* program. Na završnoj godini diplomskog studija radi kao razvojni programski inženjer u tvrtki Mono Software. Od studenog 2018. godine zaposlen je na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek kao koordinator-analitičar na projektu "Istraživanje beacona u svrhu izgradnje mreže kretanja - razvoj platforme za urbanu mobilnost", a od lipnja 2020. godine kao mlađi istraživač na projektu DATACROSS. Održava nastavu na stručnom i preddiplomskom studiju računarstva te sudjeluje kao sumentor na završnim radovima. Suautor je nekoliko znanstvenih radova u znanstvenim časopisima i zbornicima međunarodnih skupova. Glavno područje znanstvenog interesa i istraživanja mu je nadzirano strojno učenje, s težištem na učenje iz neuravnoteženih podataka. Aktivno se služi engleskim i slovačkim jezikom.

U Osijeku, 2022.

Mario Dudjak

A

Skupovi podataka i njihova predobrada

A.1 Opisi skupova podataka

U nastavku su sažeto opisani skupovi podataka korišteni za potrebe eksperimentalnih analiza u različitim poglavljima ove disertacije. Skupovi podataka preuzeti su s repozitorija za strojno učenje UCI [152] te KEEL [197]. Prvi repozitorij kreiran je na sveučilištu Irvine iz Kalifornije, a drugi zajedničkim naporom istraživača sa sveučilišta Granada, Jaén te Oviedo iz Španjolske. Skupovi podataka donirani su od strane različitih istraživača ili institucija. Uz ime skupa, u zagradama su redom navedeni broj primjeraka koje sadrži, broj značajki koje opisuju svaki primjerak, broj oznaka klasa te stupanj neuravnoteženosti skupa podataka.

A.1.1 Skupovi podataka s UCI repozitorija

1. Blood Transfusion (748/4/2/3.20)

   Skup podataka sadrži informacije o darivateljima krvi Centra za transfuziju krvi u Tajvanu koji služe za predviđanje spremnosti darivatelja na ponovno darivanje krvi. Svaki primjerak predstavlja informacije o jednom darivatelju koje su opisane pomoću 4 značajke, a koje predstavljaju njegove navike darivanja krvi. Skup podataka ukupno sadrži 748 primjeraka koji su razvrstani u 2 klase ovisno o tome je li darivatelj pristao dati krv u sklopu određenog poziva (da ili ne).

2. Breast Cancer Wisconsin (569/30/2/1.68)

Skup podataka sadrži podatke koji se koriste za identifikaciju malignih i benignih tumora dojke prikazanih na slikama. Svaki primjerak predstavlja jednu sliku mase dojke iz koje je izdvojeno 30 značajki koje opisuju njena geometrijska obilježja. Skup podataka ukupno sadrži 569 primjeraka koji su razvrstani u 2 klase ovisno o vrsti tumora (maligni ili benigni).

3. Clean2 (6598/166/2/5.49)

   Skup podataka sadrži opise struktura različitih molekula od kojih neke pripadaju skupini mosuša (korištenog, primjerice, kod kreiranje parfema). Svaki primjerak predstavlja jednu molekulu čija je struktura opisana pomoću 166 značajki. Skup podataka ukupno sadrži 6598 primjeraka koji su razvrstani u 2 klase ovisno o tome pripada li molekula skupini mosuša (da ili ne).

4. Climate (540/18/2/10.74)

   Skup podataka sadrži zapise o provedenim simulacijama klimatskih modela i njihovoj uspješnosti. Svaki primjerak predstavlja jednu simulaciju koja je opisana pomoću 18 značajki, a koje predstavljaju parametre simulacije. Skup podataka ukupno sadrži 540 primjeraka koji su razvrstani u 2 klase ovisno o njihovoj uspješnosti (uspješna ili neuspješna).

5. Congressional Voting Records (435/16/2/1.59)

   Skup podataka sadrži zapise o rezultatima glasovanja svakog zastupnika u zastupničkom domu Sjedinjenih Američkih Država. Svaki primjerak predstavlja rezultate glasovanja jednog zastupnika koji su opisani pomoću 16 značajki, a koje predstavljaju odgovore na 16 postavljenih pitanja. Skup podataka ukupno sadrži 435 primjeraka koji su razvrstani u 2 klase ovisno o stranci kojoj zastupnik pripada (republikanska ili demokratska).

6. Connectionist Bench (208/60/2/1.14)

   Skup podataka sadrži informacije prikupljene signalom sonara koji služe za identifikaciju sastava tijela od kojih se signali odbijaju. Svaki primjerak predstavlja jedan put signala koji je opisan pomoću 60 značajki. Skup podataka ukupno sadrži 208 primjeraka koji su razvrstani u 2 klase ovisno o materijalu od kojeg je signal odbijen (metal ili kamen).

7. German Credit Data (1000/61/2/2.33)

   Skup podataka sadrži zapise o kreditima koji se koriste za raspoznavanje dobrih i loših kreditnih rizika. Svaki primjerak predstavlja jedan kredit koji je opisan pomoću

61 značajke, a koje predstavljaju informacije o klijentu te parametre kredita. Skup podataka ukupno sadrži 1000 primjeraka koji su razvrstani u 2 klase ovisno o kvaliteti kreditnog rizika (dobar ili loš).

8. Glass Identification (214/9/6/3.59)

   Skup podataka sadrži zapise o kemijskim strukturama različitih vrsta stakla koji se rabe u forezničke svrhe. Svaki primjerak predstavlja rezultate jedne kemijske analize stakla koji su opisani pomoću 9 značajki. Skup podataka ukupno sadrži 214 primjeraka koji su razvrstani u 6 klasa ovisno o vrsti stakla (obrađeni građevinski prozor, neobrađeni građevinski prozor, obrađeno staklo vozila, neobrađeno staklo vozila, kontejneri, posuđe i prednja svijetla).

9. Heart Disease (270/13/2/1.25)

   Skup podataka sadrži zdravstvene podatke o pacijentima koji služe za otkrivanje srčanih bolesti. Svaki primjerak predstavlja rezultate specifičnog zdravstvenog pregleda srca koji su opisani pomoću 13 značajki. Skup podataka ukupno sadrži 270 primjeraka koji su razvrstani u 2 klase ovisno o tome ima li pacijent srčanu bolest (da ili ne).

10. Hepatitis (80/19/2/5.15)

    Skup podataka sadrži zdravstvene podatke o pacijentima koji služe za otkrivanje hepatitisa jetre. Svaki primjerak predstavlja rezultate specifičnog zdravstvenog pregleda jetre koji su opisani pomoću 19 značajki. Skup podataka ukupno sadrži 80 primjeraka koji su razvrstani u 2 klase ovisno o tome ima li pacijent hepatitis (da ili ne).

11. Hill-Valley (1212/100/2/1.00)

    Skup podataka sadrži uzorke krivulja prikazanih pomoću dvodimenzionalnih grafova koji se rabe za određivanje oblika krivulje. Svaki primjerak predstavlja jednu krivulju iz koje su izdvojene koordinate 100 točaka na y osi, a koje predstavljaju njegove značajke. Skup podataka ukupno sadrži 1212 primjeraka koji su razvrstani u 2 klase ovisno o obliku kojeg prikazuju na grafu (izbočina ili uron).

12. Image Segmentation (210/19/7/1.00)

    Skup podataka sadrži podatke koji se koriste za identifikaciju različitih materijala prikazanih na slikama. Svaki primjerak predstavlja jednu regiju slike od 3×3 piksela, koja je opisana pomoću 19 značajki. Skup podataka ukupno sadrži 210 primjeraka koji su razvrstani u 7 klasa ovisno o materijalu kojeg prikazuju (cigla, nebo, zelenilo, beton, prozor, staza ili trava).

13. Ionosphere (351/34/2/1.79)

Skup podataka sadrži informacije iz povratnih signala radara iz ionosfere. Svaki primjerak predstavlja jedan put signala radara u ionosferu koji je opisan pomoću 34 značajke, a koje predstavljaju primljene signale obrađene autokorelacijskom funkcijom s dva ulaza (vrijeme i broj pulsa). Skup podataka ukupno sadrži 351 primjeraka koji su razvrstani u 2 klase ovisno o tome je li signal našao neku strukturu u ionosferi (da ili ne).

14. LSVT Voice Rehabilitation (126/310/2/2.00)

    Skup podataka sadrži zapise o tretmanima rehabilitacije glasa koji služe za utvrđivanje njihove učinkovitosti. Svaki primjerak predstavlja jedan zvučni zapis s jednog tretmana koji je opisan pomoću 310 značajki. Skup podataka ukupno sadrži 126 primjeraka koji su razvrstani u 2 klase ovisno o učinkovitosti tretmana (učinkovit ili neučinkovit).

15. Madelon (2600/500/2/1.00)

    Umjetno stvoreni skup podataka koji sadrži podatkovne točke grupirane u 32 grupe u peterodimenzionalnom prostoru te nasumično označene oznakama $+1$ te -1. Izrađen je za potrebe natjecanja u odabiru značajki NIPS 2003. Svaki primjerak predstavlja jednu podatkovnu točku opisanu pomoću 500 značajki koje predstavljaju vrijednosti raznih linearnih kombinacija njezinih koordinata. Skup podataka ukupno sadrži 2600 primjeraka koji su razvrstani u 2 klase ovisno o oznakama primjeraka ($+1$ ili -1).

16. MuskV1 (476/166/2/1.30)

    Skup podataka sadrži opise struktura različitih molekula od kojih neke pripadaju skupini mosuša (korištenog, primjerice, kod kreiranje parfema). Svaki primjerak predstavlja jednu molekulu čija je struktura opisana pomoću 166 značajki. Skup podataka ukupno sadrži 476 primjeraka koji su razvrstani u 2 klase ovisno o tome pripada li molekula skupini mosuša (da ili ne).

17. Parkinsons (195/22/2/3.06)

    Skup podataka sadrži podatke iz zvučnih zapisa ljudskog glasa koji pomažu u prepoznavanju Parkinsonove bolesti. Svaki primjerak predstavlja jedan zvučni zapis koji je opisan pomoću 22 značajke. Skup podataka ukupno sadrži 195 primjeraka koji su razvrstani u 2 klase ovisno o tome imaju li pacijenti Parkinsonovu bolest (da ili ne).

18. QSAR Biodegradation (1055/41/2/1.96)

    Skup podataka sadrži podatke koji se koriste za proučavanje odnosa između kemijske strukture i biorazgradnje molekula. Svaki primjerak predstavlja jednu molekulu čija je struktura opisana pomoću 41 značajke. Skup podataka ukupno sadrži 1055 primjeraka

koji su razvrstani u 2 klase ovisno o spremnosti molekule na biorazgradnju (spremna ili nespremna).

19. Urban Land Cover (675/147/9/2.19)

    Skup podataka sadrži zračne snimke urbanog zemljišnog pokrivača koje se koriste kao pomoć pri urbanističkom planiranju. Svaki primjerak predstavlja jednu zračnu snimku koja je opisana pomoću 147 značajki. Skup podataka ukupno sadrži 675 primjeraka koji su razvrstani u 9 klasa ovisno o terenu kojeg prikazuju (drveće, trava, tlo, beton, asfalt, zgrade, auti, bazeni ili sjene).

20. Wine (178/13/3/1.30)

    Skup podataka sadrži rezultate kemijske analize različitih vrsta vina koji služe za utvrđivanje njihova porijekla. Svaki primjerak predstavlja rezultate analize jednog vina koji su opisani pomoću 13 značajki, a koji predstavljaju količine 13 različitih sastojaka vina. Skup podataka ukupno sadrži 178 primjeraka koji su razvrstani u 3 klase (tri kulture vina s istog područja u Italiji).

A.1.2 Skupovi podataka s KEEL repozitorija

1. Abalone-3vs11 (502/8/2/32.47)

    Skup podataka sadrži zapise fizičkih mjerenja puža puzlatka koji služe za utvrđivanje njegove starosti. Svaki primjerak predstavlja rezultate jednog mjerenja koji su opisani pomoću 8 značajki, a koje predstavljaju informacije o njegovoj težini i veličini. Izvorni skup podataka ukupno sadrži 502 primjerka koji su razvrstani u 29 klasa ovisno o starosti (0 – 29), pri čemu su klase odvojene na temelju razlike od 1.5 godine starosti. Skup podataka Abalone-3vs11 dobiven je dekompozicijom izvornog višeklasnog problema u binarni problem na način da klasa s oznakom "3" predstavlja manjinsku klasu, a klasa s oznakom "11" većinsku klasu.

2. Ecoli3 (336/7/2/8.60)

    Skup podataka sadrži podatke o strukturama bakterije E. coli koji služe za predviđanje mjesta lokalizacije proteina. Svaki primjerak predstavlja jednu strukturu bakterije Escherichia coli koja je opisana pomoću 7 značajki. Izvorni skup podataka ukupno sadrži 336 primjeraka koji su razvrstani u 8 klasa ovisno o mjestu lokalizacije proteina (cp, im, pp, imU, om, omL, imL ili imS). Skup podataka Ecoli3 dobiven je dekompozicijom izvornog višeklasnog problema u binarni problem na način da klasa s oznakom "imU" predstavlja manjinsku klasu, dok je svim ostalim primjercima dodijeljena zajednička oznaka nove (većinske) klase.

3. LED7digit1 (443/7/2/10.97)

   Skup podataka sadrži zapise o stanjima dioda na sedam segmenentnom pokazivaču koji služe za utvrđivanje znamenke prikazane na pokazivaču. Svaki primjerak predstavlja jedan prikaz pokazivača koji je opisan pomoću 7 značajki, a koje predstavljaju stanja dioda. Izvorni skup podataka ukupno sadrži 443 primjeraka koji su razvrstani u 10 klasa ovisno o prikazanoj znamenci (0 – 9). Skup podataka LED7digit1 dobiven je dekompozicijom izvornog višeklasnog problema u binarni problem na način da klasa s oznakom "1" predstavlja manjinsku klasu, dok je svim ostalim primjercima dodijeljena zajednička oznaka nove (većinske) klase.

4. New Thyroid1 (215/5/2/5.14)

   Skup podataka sadrži zdravstvene podatke o pacijentima koji služe za otkrivanje bolesti štitnjače. Svaki primjerak predstavlja rezultate specifične krvne slike jednog pacijenta koji su opisani pomoću 5 značajki. Izvorni skup podataka ukupno sadrži 215 primjeraka koji su razvrstani u 3 klase ovisno o stanju pacijenta (normalno, hipertireoza ili hipotireoza). Skup podataka New Thyroid1 dobiven je dekompozicijom izvornog višeklasnog problema u binarni problem na način da klasa s oznakom "hipertireoza" predstavlja manjinsku klasu, dok je primjercima s oznakama klase "normalno" i "hipotireoza" dodijeljena zajednička oznaka nove (većinske) klase.

5. Yeast6 (1484/10/2/41.40)

   Skup podataka sadrži zapise o strukturama kvasca radi utvrđivanja mjesta stanične lokalizacije njegovih proteina. Svaki primjerak predstavlja jednu strukturu kvasca koja je opisana pomoću 10 značajki. Izvorni skup podataka ukupno sadrži 1484 primjeraka koji su razvrstani u 10 klasa ovisno o mjestu lokalizacije proteina (CYT, NUC, MIT, ME3, ME2, ME1, EXC, VAC, POX ili ERL). Skup podataka Yeast6 dobiven je dekompozicijom izvornog višeklasnog problema u binarni problem na način da klasa s oznakom "EXC" predstavlja manjinsku klasu, dok je svim ostalim primjercima dodijeljena zajednička oznaka nove (većinske) klase.

6. Zoo3 (101/16/2/19.20)

   Skup podataka sadrži podatke o raznim životinja koji služe za njihovo razvrstavanje u vrste. Svaki primjerak predstavlja jednu životinju koja je opisana pomoću 16 značajki, a koje predstavljaju njezine prehrambene navike te biološka obilježja. Izvorni skup podataka ukupno sadrži 101 primjerak koji su razvrstani u 7 klasa ovisno o vrsti kojoj životinja pripada (gmazovi, sisavci, ptice, ribe, vodozemci, insekti ili beskičmenjaci). Skup podataka Zoo3 dobiven je dekompozicijom izvornog višeklasnog problema u binarni problem na način da klasa s oznakom "gmazovi" predstavlja manjinsku klasu, dok je svim ostalim primjercima dodijeljena zajednička oznaka nove (većinske) klase.

Tablica A.1: Zastupljenost skupova podataka u eksperimentalnim analizama

Naziv	Izvor	Broj primjeraka	Broj značajki	Broj klasa	IR	Poglavlje 3	Poglavlje 4	Poglavlje 5
Blood Transfusion	UCI	748	4	2	3.20	✗	✓	✗
Breast Cancer Wisconsin	UCI	569	30	2	1.68	✓	✗	✓
Clean2	UCI	6598	166	2	5.49	✓	✗	✓
Climate	UCI	540	18	2	10.74	✓	✓	✓
Congressional Voting Records	UCI	435	16	2	1.59	✗	✓	✗
Connectionist Bench	UCI	208	60	2	1.14	✗	✓	✗
German Credit Data	UCI	1000	61	2	2.33	✓	✗	✗
Glass Identification	UCI	214	9	6	3.59	✗	✗	✓
Heart Disease	UCI	270	13	2	1.25	✗	✗	✓
Hepatitis	UCI	80	19	2	5.15	✗	✓	✗
Hill-Valley	UCI	1212	100	2	1.00	✗	✗	✓
Image Segmentation	UCI	210	19	7	1.00	✓	✗	✗
Ionosphere	UCI	351	34	2	1.79	✓	✓	✓
LSVT Voice Rehabilitation	UCI	126	310	2	2.00	✓	✗	✗
Madelon	UCI	2600	500	2	1.00	✓	✗	✗
MuskV1	UCI	476	166	2	1.30	✗	✗	✓
Parkinsons	UCI	195	22	2	3.06	✓	✗	✓
QSAR Biodegradation	UCI	1055	41	2	1.96	✓	✗	✓
Urban Land Cover	UCI	675	147	9	2.19	✓	✗	✓
Wine	UCI	178	13	3	1.30	✓	✗	✓
Abalone-3vs11	KEEL	502	8	2	32.47	✗	✓	✗
Ecoli3	KEEL	336	7	2	8.60	✗	✓	✗
LED7digit1	KEEL	443	7	2	10.97	✗	✓	✗
New Thyroid1	KEEL	215	5	2	5.14	✗	✓	✗
Yeast6	KEEL	1484	10	2	41.40	✗	✓	✗
Zoo3	KEEL	101	16	2	19.20	✗	✓	✗

A.1.3 Zastupljenost skupova podataka u eksperimentalnim analizama

Korišteni skupovi podataka predstavljaju standardne probleme klasifikacije koji se uobičajeno koriste za vrednovanje novopredloženih postupaka za odabir značajki, preuzorkovanje te izgradnju klasifikacijskih modela RBFN. Stoga su neki od njih višestruko korišteni, a njihova zastupljenost u različitim poglavljima disertacije prikazana je tablicom A.1. Općenito, pri odabiru skupova podataka cilj je bio predstaviti razne probleme klasifikacije koji se, osim po prirodi, razlikuju po dimenzionalnosti te po stupnju neuravnoteženosti klasa. Također, nekolicina skupova podataka predstavlja probleme višeklasne klasifikacije kako bi se ispitala učinkovitost predloženih unaprjeđenja i na takvim problemima. Tako su za potrebe eksperimentalnih analiza u poglavljima 3 i 5 korišteni gotovo isti skupovi podataka, pri čemu su skupovi velike dimenzionalnosti iz poglavlja 3 zamijenjeni skupovima manje dimenzionalnosti u poglavlju 5, zbog iznimne složenosti i dugotrajnosti postupka izgradnje RBFN. S druge strane, skupovi podataka preuzeti s repozitorija KEEL korišteni su samo za potrebe analize predloženog algoritma preuzorkovanja s obzirom na mogućnost jednostavnog podešavanja njihova omjera neuravnoteženosti.

A.2 Normalizacija podataka

U svakom od korištenih skupova podataka primjerci su opisani značajkama koje imaju cjelobrojne ili realne vrijednosti. Međutim, rasponi njihovih vrijednosti često se razlikuju, što može umanjiti učinkovitost ili otežati podešavanje parametara klasifikacijskih modela. Stoga je poželjno svesti vrijednosti pojedinih značajki na jednak interval, primjerice $[0, 1]$. To se može postići normalizacijom pomoću koje se vrijednost svake značajke x_i skalira u njenu normaliziranu vrijednost

$$x'_i = \frac{x_i - x_{i,min}}{x_{i,max} - x_{i,min}}, \quad i = 1, \ldots, d, \tag{A.1}$$

gdje je $x_{i,min}$ najmanja, a $x_{i,max}$ najveća vrijednost značajke x_i u skupu podataka, dok d predstavlja broj značajki, odnosno dimenzionalnost skupa podataka.

B

Mjera ASM

Stabilnost pristupa za odabir značajki može se promatrati kao konzistentnost u pronalaženju podskupova značajki slične strukture uslijed preslagivanja skupa za treniranje ili višestrukog izvođenja pretrage u slučaju stohastičkih omotača [147]. U literaturi postoji niz mjera za utvrđivanje stabilnosti pristupa za odabir značajki, pri čemu je mjera ASM (engl. *adjusted stability measure*) jedna od najrobusnijih za usporedbu stabilnosti različitih pristupa te procjenu sličnosti podskupova značajki različitih veličina [146]. Ova mjera izražava sličnost između c podskupova značajki, a dana je prema [255] kao

$$\text{ASM} = \frac{2}{c \cdot (c-1)} \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} S_a(S_i, S_j), \tag{B.1}$$

gdje je

$$S_a(S_i, S_j) = \frac{|S_i \cap S_j| - \frac{|S_i| \cdot |S_j|}{d}}{\min(|S_i|, |S_j|) - \max(0, |S_i| + |S_j| - d)} \tag{B.2}$$

mjera sličnosti između podskupova značajki S_i i S_j, a d dimenzionalnost skupa podataka. Vrijednosti mjere ASM nalaze se u $[-1, 1]$, pri čemu vrijednost 1 označava da je pristup za odabir značajki potpuno stabilan jer su svi podskupovi značajki koje on vraća jednake strukture. S druge strane, vrijednost 0 ukazuje na to da su podskupovi značajki nasumično stvoreni, dok vrijednost -1 sugerira da niti jedan par podskupova značajki nema zajedničkih dijelova.

C

Programski jezici i računalno sklopovlje

U nastavku su navedeni programski jezici korišteni za ugradnju algoritama i provođenje statističkih testova te računala korištena za izvođenje algoritama u svrhu njihova testiranja.

- Za potrebe testiranja algoritama u poglavljima 3 i 4 razvijena je okolina u programskom jeziku Python uz uporabu radnog okvira Scikit-learn, dok je za iste potrebe u poglavlju 5 razvijena okolina u programskom jeziku C# uz uporabu radnog okvira Accord.NET.
- S obzirom na vrlo velik broj testiranja koje je bilo nužno provesti, korištena su tri računala, a njihove karakteristike sažeto su prikazane tablicom C.1. Pri bilježenju vremena izvođenja algoritama, u svim eksperimentalnim analizama uvijek je korišteno računalo označeno u tablici s Rač. 1.
- Za provođenje statističkih testova, odnosno Wilcoxonova testa ranga s predznakom te Friedmanova testa ranga, korišteni su programski jezik R [256] te programski alat KEEL [197].

Tablica C.1: Karakteristike računala korištenih za potrebe eksperimentalnih analiza

Oznaka	Vrsta	Procesor	Radna mem.	OS
Rač. 1	Stolno	Ryzen^{TM} Threadripper 2950X @ 3.50 GHz	32GB	Windows® 10 x64
Rač. 2	Prijenosno	Core^{TM} i5-8300H @ 2.30 GHz	8GB	Windows® 10 x64
Rač. 3	Stolno	Core^{TM} i3-8100 @ 3.60 GHz	8GB	Windows® 10 x64