Algoritam diferencijalne evolucije za problem odabira značajki

Znaor, Filip

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

ALGORITAM DIFERENCIJALNE EVOLUCIJE ZA PROBLEM ODABIRA ZNAČAJKI

Diplomski rad

Filip Znaor

Osijek, 2022.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 12.09.2022.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime Pristupnika:	Filip Znaor
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	D-1180R, 13.10.2020.
OIB studenta:	07398827546
Mentor:	Doc. dr. sc. Dražen Bajer
Sumentor:	,
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc. dr. sc. Bruno Zorić
Član Povjerenstva 1:	Doc. dr. sc. Dražen Bajer
Član Povjerenstva 2:	Dr. sc. Krešimir Romić
Naslov diplomskog rada:	Algoritam diferencijalne evolucije za problem odabira značajki
Znanstvena grana diplomskog rada:	**Umjetna inteligencija (zn. polje računarstvo)**
Zadatak diplomskog rada:	Opisati problem odabira značajki u smislu problema klasifikacije. Opisati algoritam diferencijalne evolucije kao vrstu evolucijskih algoritama te njegovu primjenu kao omotača za rješavanje problema odabira značajki. Ugraditi barem dvije inačice algoritma diferencijalne evolucije kao omotača s jednostavnim klasifikatorom koje se razlikuju u mehanizmu diskretizacije rješenja. Eksperimentalno ispitati učinkovitost ugrađenih inačica na nekoliko standardnih skupova podataka za klasifikaciju.
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	12.09.2022.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 23.09.2022.

Ime i prezime studenta:	Filip Znaor
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1180R, 13.10.2020.
Turnitin podudaranje [%]:	6

Ovom izjavom izjavljujem da je rad pod nazivom**: Algoritam diferencijalne evolucije za problem odabira značajki**

izrađen pod vodstvom mentora Doc. dr. sc. Dražen Bajer

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD

Odabir značajki (engl. *feature selection*, FS) predstavlja proces biranja podskupa značajki koji će kvalitetno opisati skup podataka te koji neće dovesti do smanjenja performansi klasifikacije nad tim skupom, a u određenim slučajevima i dovesti do poboljšanja performansi. FS je važan korak predobrade skupa podataka, posebice pri korištenju skupova podataka s velikim brojem značajki, gdje FS može dovesti do znatnog skraćenja vremena izvođenja algoritma klasifikacije. FS je NP-težak problem kombinatorne optimizacije za čije su rješavanje razvijene različite metode koje ne jamče pronalazak optimalnog rješenja. Jedna od često korištenih, ali i učinkovitih skupina metoda za FS su metode zasnovane na omotačima koje vrednuju različite podskupove značajki korištenjem prethodno odabranog klasifikatora, time otkrivajući složene interakcije između značajki i prilagođavajući podskupove značajki samome klasifikatoru. Neovisno o korištenoj metodi, pretraživanje cijelog prostora pretrage nije moguće za FS. Evolucijski algoritmi (engl. *evolutionary algorithms*, EAs) predstavljaju jednu grupu mogućih omotača za rješavanje FS, često korištenih zbog sposobnosti opsežnog istraživanja prostora pretrage. Jedan od najznačajnijih predstavnika EAs je algoritam diferencijalne evolucije (engl. *differential evolution*, DE). DE nije izravno primjenjiv za FS jer radi isključivo u realnoj domeni zbog prirode operatora mutacije te je stoga potrebno diskretizirati rješenja dobivena algoritmom DE korištenjem jedne od višestrukih tehnika za diskretizaciju. Također, izazov pri radu s DE predstavlja odabir odgovarajućih vrijednosti parametara algoritma. Osim predloženih često korištenih vrijednosti parametara, razvijeni su i različiti oblici postupaka za podešavanje vrijednosti parametara tijekom izvođenja DE. Odabir tehnike diskretizacije, kao i odabir načina podešavanja vrijednosti parametara DE, može imati utjecaj na postupak FS, a time i performanse klasifikatora u smislu generalizacije.

Drugo poglavlje sadrži opis postupka klasifikacije i FS. Također, detaljno je opisan algoritam DE te njegova primjena za FS, kao i tehnike diskretizacije rješenja i oblici upravljanja parametrima. Treće poglavlje opisuje ostvareno programsko rješenje, dajući detaljan opis njegova načina rada. Uz opis su dani i prikaz te način uporabe programskog rješenja. Četvrto poglavlje opisuje provedenu eksperimentalnu analizu. Opisani su eksperimenti i njihove postavke te su zatim zasebno prikazani i komentirani rezultati obaju provedenih eksperimenata.

2. ODABIR ZNAČAJKI I ALGORITAM DIFERENCIJALNE EVOLUCIJE

Odabir značajki u smislu klasifikacije predstavlja postupak kojim se odabire podskup značajki skupa podataka nad kojim je potrebno izvršiti klasifikaciju, pritom eliminirajući redundantne značajke. Odabir značajki pomaže u razumijevanju samih podataka, smanjenju računalnih zahtjeva te, u idealnom slučaju, poboljšanju performansi klasifikatora [1]. Jedno od mogućih rješenja problema odabira značajki je korištenje evolucijskih algoritama poput algoritma diferencijalne evolucije kao omotača. Navedeni pristup koristi učinkovitost klasifikacije kao mjeru kvalitete određenog rješenja koja usmjerava DE ka pronalasku podskupa značajki koji će rezultirati što je većom mogućom učinkovitošću klasifikacije.

2.1. Klasifikacija

Klasifikacija je postupak kojim se objektima na temelju njihovih značajki dodjeljuje oznaka klase kojoj on pripada. Kako bi taj zadatak bio ostvariv, potrebno je unaprijed poznavati skup postojećih klasa te skup objekata kojima je klasa određena kako bi algoritam za klasifikaciju bio sposoban naučiti razvrstavati objekte u pripadajuće klase. Krajnji cilj klasifikacije je postizanje najveće moguće točnosti pri predviđanju klasa prethodno neviđenih objekata.

Svaki podatak u skupu podataka korištenom za klasifikaciju može se opisati vektorom od n elemenata gdje je $n \in \mathbb{N}$ broj značajki. Iako značajke mogu biti numeričkog i kategoričkog tipa, u ovom radu će fokus biti isključivo na numeričkim značajkama, stoga vrijedi da se skup podataka X sastoji od vektora $\mathbf{x} \in \mathbb{R}^n$. Svakom vektoru $\mathbf{x_i}$ pridružena je odgovarajuća oznaka y_i, pri čemu je $y_i \in \{y_1, y_2, \dots, y_m\}$, gdje je m broj klasa. Cilj klasifikacije je odrediti oznake y_i za one vektore $\mathbf{x_i}$ kojima ta oznaka nije unaprijed poznata.

Klasifikacija se primjenjuje u mnogobrojnim područjima. Neki primjeri uključuju medicinu (npr. detekcija raka dojke na slikama dobivenim mamografijom), biomedicinu (npr. analiza DNA sekvenci u svrhu identificiranja nezdravog tkiva) i bankarstvo (npr. autentifikacija potpisa) [2]. Za svaku primjenu je točnost klasifikacije od iznimne važnosti.

2.1.1. Vrednovanje učinkovitosti

Kako bi klasifikator bio sposoban odrediti klasu za neoznačene objekte, potrebno ga je istrenirati na podskupu korištenog skupa podataka. Navedeni postupak može se ostvariti na brojne načine. Jedan od najčešće korištenih postupaka je pristup izdvajanja (engl. *holdout*)[3]. Pristup izdvajanja podrazumijeva podjelu podataka u sljedeće podskupove:

- Skup za trening (engl. *training set*), odnosno skup označenih objekata koji se koriste za podešavanje parametara klasifikatora
- Skup za validaciju (engl. *validation set*), odnosno skup označenih objekata koji se koristi za vrednovanje modela pri treniranju
- Skup za testiranje (engl. *test set*), odnosno skup objekata koji se koristi za neovisno vrednovanje generalizacije konačnog modela

Konačno vrednovanje učinkovitosti provodi se na skupu za testiranje jer se on sastoji od objekata koje model nije imao na raspolaganju prilikom postupka treniranja te samim time najbolje može ocijeniti sposobnost generalizacije klasifikatora. Osim navedenog pristupa podjele podataka za vrednovanje, postoje i drugi pristupi kao što je vrednovanje pomoću k-preklopa/rezova (engl. *k-fold cross-validation*) [3].

Kako bi se ocijenile performanse klasifikatora na danom skupu podataka, potrebno je upotrijebiti neku od mjera učinkovitosti klasifikacije. Postoji veći broj mjera učinkovitosti te je njihov izbor ovisan o korištenom algoritmu klasifikacije i svojstvima skupa podataka (npr. uravnoteženost klasa). Većina jednostavnijih i češće korištenih mjera učinkovitosti proizlaze iz tzv. matrice zbunjenosti (engl. *confusion matrix*) [3]. Matrica zbunjenosti prikazana je slikom 2.1.

		Stvarno	
		Pozitivna	Negativna
Predviđeno	Pozitivna	TP	FP
	Negativna	FN	TN

Slika 2.1. *Matrica zbunjenosti*

Kao što je vidljivo na slici 2.1., matrica zbunjenosti u slučaju binarne klasifikacije (dvije klase, u ovom slučaju pozitivna i negativna) se sastoji od dva retka i dva stupca, pri čemu reci predstavljaju stvarne klase, a stupci klase dobivene ili predviđene korištenim algoritmom klasifikacije. Vrijednosti dobivene iz matrice zbunjenosti su sljedeće:

- Istiniti pozitivi (engl. *true positives,* TP), odnosno broj objekata ispravno klasificiranih kao pozitivni
- Lažni pozitivi (engl. *false positives,* FP), odnosno broj negativnih objekata neispravno klasificiranih kao pozitivni
- Istiniti negativi (engl. *true negatives,* TN), odnosno broj objekata ispravno klasificiranih kao negativni
- Lažni negativi (engl. *false negatives,* FN), odnosno broj pozitivnih objekata neispravno klasificiranih kao negativni

Cilj klasifikacije je smanjenje količine lažnih pozitiva i lažnih negativa, pri čemu se, ovisno o prirodi klasifikacijskog problema, bira je li bitnije minimizirati lažne pozitive ili lažne negative. Primjerice, pri detekciji neke bolesti bitnije je minimizirati količinu lažnih negativa kako bi se izbjegli slučajevi gdje se bolesnim pacijentima ne detektira oboljenje. U idealnom slučaju vrijednosti FP i FN bile bi jednake nuli.

Iz matrice zbunjenosti proizlaze mnogobrojne mjere učinkovitosti klasifikacije. Preciznost (engl. *precision*) je mjera koja kao rezultat daje udio točno klasificiranih pozitivnih objekata iz skupa podataka. Izračun preciznosti određen je s

$$Preciznost = \frac{TP}{TP + FP}. \tag{2-1}$$

Odziv (engl. *recall*) je mjera koja kao rezultat daje udio pozitivnih objekata koji su klasificirani kao pozitivni. Izračun odziva određen je s

$$Odziv = \frac{TP}{TP + FN}. \tag{2-2}$$

Preciznost klasifikacije (engl. *classification accuracy*) je mjera koja kao rezultat daje udio točno klasificiranih objekata svih klasa. Često je korištena u postupcima klasifikacije, no nije prikladna za neuravnotežene skupove podataka. Izračun preciznosti klasifikacije određen je s

$$Preciznost\ klasifikacije = \frac{TP + TN}{TP + FP + TN + FN}. \tag{2-3}$$

Kvalitetan klasifikator treba imati visoku preciznost i odziv. Kako je rukovanje dvama mjerama istovremeno često nezgrapno, preciznost i odziv mogu se spojiti u jedinstvenu mjeru učinkovitosti klasifikacije poznatu kao F-mjeru (engl. F-score). F-mjera predstavlja ponderiranu harmonijsku sredinu preciznosti i odziva [3]. Za bilo koji $\beta \in \mathbb{R}, \beta > 0$, opći izraz za izračun F-mjere određen je kao

$$F_\beta = \frac{(1+\beta)(preciznost * odziv)}{(\beta * preciznost) + odziv}. \tag{2-4}$$

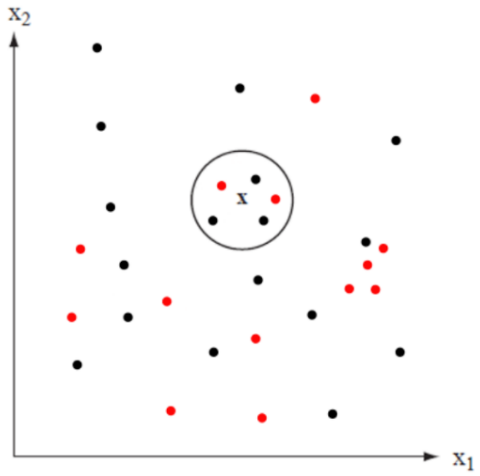
Najčešće korištena F-mjera je F_1-mjera ($\beta = 1$) koja pridaje jednaku važnost preciznosti i odzivu. Izraz za F_1-mjeru dan je kao

$$F_1 = \frac{2(preciznost * odziv)}{preciznost + odziv}. \tag{2-5}$$

Pri računanju F-mjere za problem s više od dviju klasa, postoji više načina za izračun konačne F-mjere [4]. Jedan od tih načina je makro metoda koja računa F-mjeru za svaku klasu te kao konačni rezultat daje aritmetičku sredinu dobivenih vrijednosti mjera. Glavni nedostatak makro metode je ignoriranje potencijalne neuravnoteženosti klasa. Ponderirana metoda (engl. *weighted method*) pri izračunu aritmetičke sredine svakoj klasi daje težinu proporcionalnu zastupljenosti te klase. Nadalje, mikro metoda računa metrike na globalnoj razini brojeći ukupni broj TP, FP i FN, time ublažavajući mogući utjecaj neuravnoteženosti klasa.

2.1.2. Algoritam *k*-najbližih susjeda

Algoritam *k*-najbližih susjeda (engl. *k-nearest-neighbours, k*-NN) je jednostavan algoritam klasifikacije. Algoritam svakom nepoznatom objektu **x** pridružuje onu oznaku koja je najčešća među *k* najbližih susjeda objekta **x** [5].



Slika 2.2. *Grafički prikaz rada algoritma k-najbližih susjeda [5]*

Slika 2.2 prikazuje grafički primjer rada algoritma *k*-NN za slučaj binarne klasifikacije i $k = 5$. Za točku označenu simbolom x pronalazi se njezinih pet najbližih susjeda te se promatraju klase kojima oni pripadaju. Kako je na slici vidljivo da tri susjeda pripadaju crnoj klasi, a dva susjeda pripadaju crvenoj klasi, točka x bit će pridružena crnoj klasi jer je ona većinska klasa u susjedstvu točke. Za računanje udaljenosti između točaka najčešće se koristi Euklidska udaljenost

$$d_2(\mathbf{x},\mathbf{y}) = \left[\sum_{j=1}^{d}\left(x_j - y_j\right)^2\right]^{\frac{1}{2}} = [(\mathbf{x}-\mathbf{y})(\mathbf{x}-\mathbf{y})^T]^{\frac{1}{2}}. \tag{2-6}$$

Osim Euklidske udaljenosti, moguće je koristiti i druge udaljenosti poput Manhattan udaljenosti ili Čebiševljeve udaljenosti, no u ovom radu će se razmatrati isključivo Euklidska udaljenost. Pseudokod algoritma *k*-NN prikazan je na slici 2.3.

```
ulaz: testni objekt x, trening objekti T, veličina trening skupa
n, broj susjeda k, broj klasa c

udaljenosti = []
za svaki i=1 do n činiti
   udaljenosti[i] = euklidska_udaljenost(x, T[i])

// vrati indekse k najbližih točaka
indeksi_najbližih = pronađi_k_najbližih_susjeda(x, T, k)

glasovi_klase = []
za svaki i=1 do c činiti
   glasovi_klase[i] = 0

// prebrojavanje kojim klasama pripada susjedstvo
za svaki i=1 do k činiti
   klasa = T[indeksi_najbližih[i]].klasa
   glasovi_klase[klasa] += 1

// pronalaženje klase s najviše pojavljivanja
konačna_klasa = max_index(glasovi_klase)
vrati konačna_klasa
```

Slika 2.3. *Pseudokod algoritma k-najbližih susjeda*

Ključan i jedini parametar algoritma *k*-NN je *k*, odnosno veličina susjedstva. Neke češće korištene vrijednosti su $k \in \{1,3,5\}$. Algoritam *k*-NN je često korišten u postupcima odabira značajki zasnovanim na omotačima, ponajviše zbog jednostavnosti algoritma, njegove brzine, učinkovitosti te izostankom potrebe za treniranjem i znatnim podešavanjem modela [6].

2.2. Problem odabira značajki

Cilj odabira značajki je odabir podskupa značajki koje mogu efikasno opisati ulazne podatke, pritom smanjujući utjecaj šuma u podatcima, preklapanja klasa te često postižući pozitivan utjecaj na kvalitetu rezultata klasifikacije [1]. Skupovi podataka s vrlo velikim brojem značajki nerijetko se mogu svesti na znatno manji podskup značajki bez gubitaka u kvaliteti klasifikacije. Primjerice, neke značajke mogu imati vrlo visoku razinu korelacije s drugim značajkama. U takvom slučaju, jedna takva značajka može biti dovoljna za provođenje klasifikacije jer ostale značajke ne donose nove informacije. Drugu vrstu redundantnih značajki predstavljaju značajke koje nisu korelirane niti s jednom klasom. Osim što takve značajke ne pridonose kvaliteti klasifikacije, u određenim slučajevima mogu i odmoći uvođenjem šuma u podatke [1]. Osim eliminiranja redundantnih značajki, prednost FS je smanjenje vremena izvođenja algoritama klasifikacije koje je direktan rezultat smanjenja broja značajki u skupu podataka. Također, smanjenje broja značajki pomaže u razumijevanju samih podataka. Navedena svojstva čine FS iznimno bitnim aspektom klasifikacije, odnosno korakom predobrade podataka.

Kako je FS NP-težak problem [7], iscrpno vrednovanje svih mogućih podskupova značajki često nije moguće zbog veličine prostora pretrage te je stoga potrebno koristiti postupke koji ne mogu jamčiti optimalna rješenja za eliminaciju značajki [1]. Pritom nije dovoljno zasebno analizirati svaku značajku zbog složenih i teško predvidivih međudjelovanja značajki. Razvijene su mnoge metode FS, a njihova temeljna podjela bazirana je na načinu vrednovanja značajki, odnosno podskupova značajki. Osnovne grupe metoda su metode zasnovane na filtrima, metode zasnovane na omotačima te ugrađene metode [1].

Metode zasnovane na filtrima biraju značajke isključivo na temelju njihovih intrinzičnih svojstava, ne uzimajući u obzir njihove performanse u klasifikaciji. Metode zasnovane na filtrima koriste razne postupke iz statistike i teorije informacije za vrednovanje značajki te odabiru zadani broj najbolje rangiranih značajki. Glavna prednost metoda zasnovanih na filtrima je njihova računska jednostavnost i brzina te smanjenja vjerojatnost prenaučenosti (engl. *overfitting*). Nedostatak metoda zasnovanih na filtrima je zasebno vrednovanje značajki koje onemogućuje uzimanje složenih interakcija između značajki u obzir.

Metode zasnovane na omotačima koriste klasifikator kao crnu kutiju (engl. *black box*) te vrednuju podskupove značajki na temelju performansi korištenog klasifikatora. Kao i kod metoda zasnovanih na filtrima, ključni aspekt cijele metode je strategija pretrage (engl. *search strategy*) podskupova značajki. Osnovna podjela metoda zasnovanih na omotačima s obzirom na strategiju

pretrage je podjela na heuristike i metaheuristike [1]. Osnovni primjer heuristike je algoritam slijedne pretrage unaprijed (engl. *sequential forward selection*, SFS) koji se bazira na iterativnom dodavanju jedne po jedne značajke u skup značajki dok se performanse klasifikatora poboljšavaju. U skup se uvijek dodaje značajka koja rezultira najvećim poboljšanjem performansi klasifikacije. Osim SFS algoritma, postoji i algoritam slijedne pretrage unatrag (engl. *sequential backward selection*) koja kreće sa svim značajkama te odbacuje jednu po jednu značajku. Moguća su daljnja poboljšanja algoritma koji omogućuju vraćanje unazad (engl. *backtracking*) kako bi se uvela dodatna fleksibilnost u rad algoritma. Metaheuristike vrednuju različite podskupove značajki s ciljem što boljih performansi klasifikacije. Primjer metaheuristike su evolucijski algoritmi u kojima se svaka jedinka, odnosno podskup značajki, vrednuje na temelju njezinih performansi u klasifikaciji te se procesima mutacije, križanja i selekcije nastoje generirati rješenja veće kvalitete. Empirijski je dokazano kako metode zasnovane na omotačima daju bolje rezultate od metoda zasnovanih na filtrima zahvaljujući vrednovanju podskupa samim klasifikatorom [8]. Metode zasnovane na omotačima su također sposobne otkriti složenije odnose među značajkama te često proizvode manje podskupove odabranih značajki [9]. Njihov glavni nedostatak je računalna zahtjevnost te veća mogućnost dolaska do prenaučenosti koja nastaje kao rezultat pretjerane prilagodbe podacima za treniranje ili validaciju, time potencijalno smanjujući generalizacijske sposobnosti klasifikatora.

Ugrađene metode izvršavaju algoritam FS tijekom izvođenja algoritma za klasifikaciju tako što su ugrađene u sami algoritam klasifikacije. Najčešće korištene ugrađene metode temeljene su na stablima odluke i logističkoj regresiji, dok neke metode dodjeljuju težinske koeficijente značajkama bazirane na regularizacijskim modelima s funkcijama cilja koje minimiziraju ukupnu pogrešku klasifikacije te postavljaju koeficijente značajki na vrlo male vrijednosti ili na nulu, pritom odbacujući te značajke [8]. Ugrađene metode manje su računalno zahtjevne od metoda zasnovanih na omotačima te se brže izvršavaju s manjim rizikom prenaučenosti, no i dalje su poprilično zahtjevne u usporedbi s metodama zasnovanim na filterima te ograničene isključivo na klasifikator za koji su razvijene, pri čemu je postupak razvoja složen.

Posebnu vrstu metoda FS predstavljaju hibridne metode koje kombiniraju najbolja svojstva metoda zasnovanih na omotačima i filtrima. U jednom od češćih načina stvaranja hibridnih metoda FS, hibridne metode prvo koriste pristup filtra kako bi smanjile prostor pretrage i pritom pronašle potencijalne podskupove značajki, a zatim koriste pristup omotača kako bi pronašle najbolji podskup značajki [8]. Mogući su i drugi pristupi, ovisno o korištenom omotaču. Glavna prednost

hibridnih metoda je kombiniranje računalne efikasnosti filtera s mogućnošću opsežnog istraživanja prostora pretrage omotača.

2.3. Algoritam diferencijalne evolucije

Algoritam diferencijalne evolucije predstavlja stohastičku metodu optimizacije zasnovanu na populaciji rješenja koju su razvili Storn i Price [10]. DE stvara populaciju sastavljenu od nasumično generiranih mogućih rješenja za dani problem optimizacije te zatim iterativnim postupkom nastoji dobiti što kvalitetnija rješenja nastala mutiranjem i rekombiniranjem postojećih rješenja. Novonastala rješenja zadržavaju se u generaciji ako su kvalitetnija od prethodnih, pri čemu se kao mjera kvalitete koristi mjera koja odgovara problemu koji se rješava (npr. F_1-mjera klasifikacije za FS). Primarna razlika algoritma DE u odnosu na ostale evolucijske algoritme se očituje u činjenici da DE koristi informacije o razlikama, odnosno udaljenostima rješenja unutar trenutne populacije kako bi se usmjerio daljnji tijek pretrage. Navedeni postupak se očituje u postupku mutacije. Ovisnost DE o vektorima koji predstavljaju razlike elemenata populacije čine ga primjenjivim za probleme kontinuirane ili numeričke optimizacije dok standardna implementacija DE ne može biti primijenjena za rješavanje problema kombinatorne optimizacije [11]. Standardni koraci DE su inicijalizacija populacije, mutacija, križanje i selekcija. Navedeni koraci opisani su u nastavku.

Populacija $\mathrm{P}_{\mathbf{x}}$ sastoji se od D-dimenzionalnih realnih vektora.. Populacija je određena s

$$\mathrm{P}_{\mathbf{x},g} = \left(\mathbf{x}_{\mathrm{i,g}}\right), \qquad i = 0,1,\dots,NP-1, g = 0,1,\dots,g_{max}, \tag{2-7}$$

pri čemu je g indeks trenutne generacije ili iteracije, g_{max} unaprijed određeni maksimalni broj iteracija, a *NP* veličina populacije. Svaki vektor $\mathbf{x}_{\mathrm{i,g}}$ određen je kao

$$\mathbf{x}_{\mathrm{i,g}} = \left(x_{\mathrm{j,i,g}}\right), \qquad j = 0,1,\dots,D-1, \tag{2-8}$$

pri čemu D označava dimenzionalnost vektora $\mathbf{x}_{\mathrm{i,g}}$.

Korakom inicijalizacije određuje se početna populacija, odnosno njezina nulta generacija. Prije provođenja samog koraka inicijalizacije, potrebno je definirati donje i gornje granice b_L i b_R kojima se ograničavaju početne vrijednosti elemenata vektora u populaciji, odnosno

$$b_L \leq x_{\mathrm{j,i,0}} \leq b_R, \qquad \forall i \in \{0,1,\dots,NP\} \wedge \forall j \in \{0,1,\dots,D-1\}. \tag{2-9}$$

Početna vrijednog j-tog parametra i-tog vektora populacije uobičajeno se postavlja na nasumičnu vrijednost, odnosno

$$x_{\mathrm{j,i,0}} = \mathrm{rand}_j(0,1) * (b_U - b_L) + b_L, \tag{2-10}$$

čime je definirano kako su sve početne vrijednosti u intervalu $[b_L, b_U]$.

Korak mutacije ključni je element DE smatran glavnim faktorom performansi algoritma [12]. Njime se linearnim kombinacijama članova trenutne populacije dobiva nova populacija sačinjena od *NP* mutiranih vektora, odnosno mutanata. Navedeno se ostvaruje nasumičnim uzorkovanjem dvaju vektora populacije, izračunom njihove razlike te zbrajanjem dobivene razlike s trećim, odnosno baznim vektorom (engl. *base vector*). Zbog njezine važnosti, razvijene su mnogobrojne strategije mutacije u pravilu označene notacijom ρ/δ, gdje ρ predstavlja metodu odabira baznog vektora, a δ predstavlja broj razlika dvaju vektora s kojima se množi bazni vektor. Dvije najčešće korištene strategije mutacije su *rand/1* strategija dana kao

$$\mathbf{v}_{i,g} = \mathbf{x}_{r0,g} + F * \left(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}\right) \tag{2-11}$$

i *best/1* strategija dana kao

$$\mathbf{v}_{i,g} = \mathbf{x}_{najbolji,g} + F * \left(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}\right), \tag{2-12}$$

gdje su *r0*, *r1*, i *r2* nasumično odabrani međusobno različiti indeksi elemenata populacije, a $\mathbf{x}_{najbolji,g}$ najbolji vektor trenutne populacije na temelju vrijednosti funkcije cilja. Parametar *F* naziva se faktorom skaliranja (engl. *scale factor*) čija se vrijednost postavlja na vrijednost u intervalu $[0, \infty >$, pri čemu vrijednosti veće od 1 rijetko daju kvalitetne rezultate [13]. Ako je prilikom izvođenja DE potrebno da sve vrijednosti budu unutar nekog intervala $[b_L, b_U]$, navedeno se postiže implementacijom rukovanja granicama prostora pretrage u korak mutacije. Najjednostavniji primjer rukovanja granicama je pristup projekcijom opisan kao

$$v_{j,i,g} = \begin{cases} b_L & ako\ v_{j,i,g} > b_L \\ b_U & ako\ v_{j,i,g} < b_U \\ v_{j,i,g} & inače \end{cases}. \tag{2-13}$$

Korakom križanja ili rekombinacije kreiraju se novi pokusni vektori (engl. *trial vector*) $\mathbf{u}_{i,g}$ dijelom sastavljeni od elemenata ciljnog vektora (engl. *target vector*) $\mathbf{x}_{\mathrm{i,g}}$, a dijelom od elemenata njima pripadnih mutanata $\mathbf{v}_{i,g}$. Korak križanja zadan je kao

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & ako\ \mathrm{rand}_j(0,1) \leq CR\ ili\ j = j_{\mathrm{rand}} \\ x_{j,i,g} & u\ suprotnom \end{cases}, \tag{2-14}$$

gdje *CR* predstavlja vrijednost križanja (engl. *crossover-rate*) u rasponu [0,1], odnosno vjerojatnost da će se pojedina komponenta ciljnog vektora $\mathbf{x}_{i,g}$ zamijeniti komponentom mutanta $\mathbf{v}_{i,g}$, a j_{rand} predstavlja nasumično odabran indeks elementa iz populacije u intervalu [0, *NP*-1] koji osigurava da dobiveni pokusni vektor nije u potpunosti jednak ciljnom vektoru $\mathbf{x}_{i,g}$.

Korakom selekcije određuju se vektori koji će sačinjavati sljedeću generaciju populacije. Navedeno se ostvaruje usporedbom ciljnog vektora $\mathbf{x}_{i,g}$ s njemu pripadnim pokusnim vektorom $\mathbf{u}_{i,g}$ dobivenim postupcima mutacije i križanja na temelju vrijednosti ciljne funkcije. Postupak je određen kao

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g} & ako\ f(\mathbf{u}_{i,g}) \leq f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & u\ suprotnom \end{cases} \tag{2-15}$$

iz čega je vidljivo da se u sljedeću generaciju prenosi vektor koji ostvaruje višu vrijednost ciljne funkcije, dok se u slučaju jednakih vrijednosti ciljne funkcije uzima pokusni vektor. Navedeni postupak osigurava opstanak boljih jedinki što kroz iterativno izvođenje algoritma omogućuje kontinuirano poboljšanje kvalitete populacije. Koraci mutacije, križanja i selekcije ponavljaju se dok se ne zadovolji unaprijed zadani kriterij stajanja. Slika 2.3. prikazuje pseudokod DE s korištenom *rand/1* mutacijom. Zamjena *rand/1* s *best/1* mutacijom lako je izvediva bez izmjene ostatka algoritma.

```
postavi vrijednosti NP, F, CR i gmax
inicijaliziraj populaciju P = [x1,0,...,xNP,0] //po jednadžbi (2-10)
g = 0
sve dok g < gmax činiti
   za svaki i=1 do NP činiti
      generiraj nasumične različite indekse r0,r1,r2 u rasponu
[1,NP]
      vi,g = xr0,g + F(xr1,g – xr2,g)
      generiraj nasumični indeks jrand u rasponu [1,D]
      za svaki j=1 do D činiti
         generiraj nasumični broj r u rasponu [0,1]
         ako r <= CR ili j == jrand onda
            uj,i,g = vj,i,g
         u suprotnom
            uj,i,g = xj,i,g
      ako f(ui,g) >= f(xi,g) onda
         xi,g+1 = ui,g
      u suprotnom
         xi,g+1 = xi,g
   g = g+1
```

Slika 2.4. *Pseudokod DE*

2.3.1. Upravljanje parametrima

Rad algoritma DE uvelike je ovisan o odabiru prethodno spomenutih parametara *NP*, *CR* i *F*. Jedna od najčešće korištenih postavki DE koristi vrijednosti *NP* = 100, *CR* = 0.9 i *F* = 0.5 [12], pri čemu se navedene vrijednosti *CR* i *F* koriste na razini cijele populacije tijekom cijelog izvođenja DE. Međutim, odabir vrijednosti parametara DE nerijetko ovisi o svojstvima problema koji se rješava te samim time ne postoji optimalna kombinacija vrijednosti parametara koja bi vrijedila za sve moguće slučajeve. Kako bi se uklonila potreba za ručnom pretragom optimalne kombinacije vrijednosti parametara, istraživane su mogućnosti upravljanja parametrima (engl. *parameter control*) *F* i/ili *CR* u DE.

Osnovne kategorije postupaka upravljanja parametrima su determinističke metode, metode podešavanja i metode samopodešavanja [14]. Determinističke metode određuju nove vrijednosti parametara na temelju predodređenog rasporeda, metode podešavanja nove vrijednosti određuju na temelju povratne informacije dobivene od DE, dok metode samopodešavanja dodjeljuju zasebne vrijednosti parametara svim rješenjima u populaciji te ih podešavaju zajedno s rješenjima.

Primjer determinističke metode upravljanja parametrima [14] temeljen je na linearnom smanjivanju parametra *F* opisanog izrazom

$$F_g = (F_{max} - F_{min}) * \frac{(g_{max} - g)}{g_{max}} + F_{min}, \tag{2-16}$$

gdje su F_{max} i F_{min} unaprijed zadana maksimalna i minimalna vrijednost *F*, g trenutna generacija, a g_{max} maksimalni broj generacija. Parametar *F* linearno se smanjuje s vrijednosti F_{max} u prvoj generaciji do vrijednosti F_{min} u posljednjoj generaciji. Korištenjem viših vrijednosti *F* u ranijim fazama izvođenja DE potiče se šire istraživanje prostora pretrage, dok se korištenjem nižih vrijednosti *F* u kasnijim fazama izvođenja DE rješenja podešavaju na manjoj skali kako bi se istražio uži prostor pretrage u kojem leži pretpostavljeni globalni optimum [14].

Metode samopodešavanja vrijednosti parametara *F* i *CR* zasnivaju se na njihovoj izmjeni tijekom izvođenja algoritma. Naime, svakom članu populacije pridružuje se vlastita vrijednost *F* i *CR* iz čega proizlazi da će članovi s boljim vrijednostima parametara imati veću vjerojatnost preživljavanja u sljedeću generaciju, a samim time i propagiranja tih vrijednosti parametara u daljnje generacije. Primjer izračuna novih vrijednosti parametra *F* [15] dan je izrazom

$$F_{i,g+1} = \begin{cases} F_l + rand_1 * F_u & ako\ rand_2 < \tau_1 \\ F_{i,g} & u\ suprotnom \end{cases}, \tag{2-17}$$

gdje su F_l i F_u donja, odnosno gornja granica vrijednosti parametra F, $rand_1$ i $rand_2$ nasumične vrijednosti iz intervala [0,1], a τ_1 vrijednost praga, odnosno vjerojatnost promjene vrijednosti F. Autori u [15] koriste vrijednosti $\tau_1 = 0.1$, $F_l = 0.1$ i $F_u = 0.9$ čime se vrijednost F ograničava na interval [0.1, 1]. Nova vrijednost parametra *CR* računa se prema izrazu

$$CR_{i,g+1} = \begin{cases} rand_3 & ako\ rand_4 < \tau_2 \\ CR_{i,g} & u\ suprotnom \end{cases}, \tag{2-18}$$

gdje je $rand_4$ nasumična vrijednost iz intervala [0,1], a τ_2 vrijednost praga, odnosno vjerojatnost promjene vrijednosti *CR*. Autori koriste vrijednost $\tau_2 = 0.1$, a izraz (2-18) ograničava vrijednost *CR* na interval [0,1]. Novostvorene vrijednosti *F* i *CR* prosljeđuju se u sljedeću generaciju isključivo u slučajevima kada pokusni vektor bude odabran za prijenos u sljedeću generaciju, dok se u suprotnome slučaju zadržavaju vrijednosti iz prethodne generacije.

2.3.2. Tehnike za diskretizaciju rješenja

Kao što je prethodno napomenuto, zbog prirode postupka mutacije, algoritam DE radi isključivo u realnoj domeni i nije izravno primjenjiv na probleme diskretne ili kombinatorne optimizacije, primjerice FS. Primjena DE za probleme diskretne optimizacije moguća je uvođenjem odgovarajućeg postupka diskretizacije rješenja. Diskretizacijom se svi elementi vektora u populaciji pretvaraju u diskretne vrijednosti (primjerice, za FS na vrijednosti 0 i 1). Diskretizacija se provodi isključivo u svrhu vrednovanja rješenja, ali ne utječe na vrijednosti rješenja u populaciji koja ostaju u realnoj domeni. Diskretizacijom se stvara zasebno binarno rješenje koje se zatim vrednuje odgovarajućim algoritmom klasifikacije. Postoje različite tehnike diskretizacije pri čemu svaka tehnika može imati drugačiji utjecaj na ponašanje i konačne rezultate DE [11]. U nastavku su opisane tri tehnike diskretizacije rješenja korištene za stvaranje binarnih vektora.

Prva tehnika je preuzeta iz algoritma *binDE* [11]. Tehnika koristi realne vrijednosti elemenata vektora kao vjerojatnosti postavljanja elemenata diskretiziranih vektora na vrijednost 1. U svrhu postavljanja realnih vrijednosti elemenata vektora u raspon [0,1] koristi se sigmoidna funkcija. Stvaranje diskretiziranih rješenja ovom tehnikom dano je s

$$y_{i,j,g} = \begin{cases} 1 & ako\ U(0,1) < s\left(x_{i,j,g}\right) \\ 0 & u\ suprotnom \end{cases}, \tag{2-19}$$

gdje $U(0,1)$ predstavlja nasumično odabranu vrijednost iz intervala [0,1] a s standardnu logističku funkciju

$$s(x) = \frac{1}{1 + e^{-x}} \tag{2-20}$$

Druga tehnika je preuzeta iz algoritma *normDE* [11]. Tehnika je zasnovana na postupku normalizacije, odnosno linearnom skaliranju elemenata vektora u raspon [0,1]. Normalizacija je određena s

$$x'_{i,j,g} = \frac{x_{i,j,g} - x_i^{min}}{x_i^{max} - x_i^{min}}, \tag{2-21}$$

gdje x_i^{max} i x_i^{min} predstavljaju najveću i najmanju komponentu *i*-tog vektora. Nakon normalizacije, konačne binarne vrijednosti dobivaju se prema

$$y_{i,j,g} = \begin{cases} 0 & ako\ x'_{i,j,g} < 0.5 \\ 1 & u\ suprotnom \end{cases}, \tag{2-22}$$

odnosno diskretizacijom vrijednosti manjih od 0.5 na vrijednost 0 i diskretizacijom vrijednosti većih ili jednakih od 0.5 na vrijednost 1.

Treća tehnika [16] zasniva se na jednostavnom postupku usporedbe vrijednosti sa zadanom vrijednosti praga (engl. *threshold*) u kojem se sve vrijednosti veće ili jednake nekom zadanom pragu postavljaju na vrijednost 1, a u suprotnome se postavljaju na vrijednost 0, odnosno

$$y_{i,j,g} = \begin{cases} 0 & ako\ x_{i,j,g} < \theta \\ 1 & u\ suprotnom \end{cases}, \tag{2-23}$$

gdje je θ vrijednost praga koja je u pravilu postavljena na 0.5. Pri izvođenju DE s navedenim postupkom diskretizacije, potrebno je držati vrijednosti elemenata vektora u intervalu [0,1] prethodno spomenutim postupkom rukovanja granicama prostora pretrage u koraku mutacije.

2.3.3. Primjena za odabir značajki

Algoritam DE može se upotrijebiti za rješavanje problema FS odgovarajućom prilagodbom standardnog DE. Ako se provodi FS za skup podataka s *n* značajki, tada će svaki član populacije u DE biti vektor od *n* elemenata, gdje svaki element predstavlja jednu značajku. Kako je za vrednovanje (klasifikaciju) potrebno odrediti podskup značajki koji će se koristiti, potrebno je ugraditi i postupak diskretizacije rješenja kojim se elementi vektora postavljaju na vrijednost 1 (značajka se uzima) ili 0 (značajka se odbacuje). Kvaliteta pojedinog rješenja iz populacije dobiva se klasifikacijom skupa podataka s odabranim podskupom značajki nekim od algoritama klasifikacije (npr. *k*-NN) te računanjem prikladne mjere kvalitete (npr. F1-mjera). U slučaju da se diskretizacijom dobije prazan skup odabranih značajki, takvom rješenju se kvaliteta postavlja na

nulu čime se osigurava brza zamjena tog rješenja valjanim rješenjem u narednoj iteraciji. Slika 2.5. prikazuje pseudokod DE korištenog za FS. Podebljanim tekstom istaknute su potrebne preinake u kodu kako bi se standardni DE prilagodio za FS, odnosno korak diskretizacije rješenja i vrednovanja diskretiziranog rješenja postupkom klasifikacije.

```
postavi vrijednosti NP, F, CR i gmax
inicijaliziraj populaciju P = [x1,0,...,xNP,0] //po jednadžbi (2-10)

//vrednovanje početne populacije
za svaki i=1 do NP činiti
  x'i,0 = diskretizacija(xi,0)
  F1xi = kNN(dataset, x'i,0)

g = 0
sve dok g < gmax činiti
   za svaki i=1 do NP činiti
      generiraj nasumične različite indekse r0,r1,r2 u rasponu
[1,NP]
      vi,g = xr0,g + F(xr1,g – xr2,g)
      generiraj nasumični indeks jrand u rasponu [1,D]
      za svaki j=1 do D činiti
         generiraj nasumični broj r u rasponu [0,1]
         ako r <= CR ili j == jrand onda
            uj,i,g = vj,i,g
         u suprotnom
            uj,i,g = xj,i,g

      u'i,g = diskretizacija(ui,g)

      // F1-mjera dobivena klasifikacijom zadanog
      // skupa podataka korištenjem podskupa značajki
      F1u = kNN(dataset, u'i,g)

      ako F1u >= F1xi onda
         xi,g+1 = ui,g
         F1xi = F1u
      u suprotnom
         xi,g+1 = xi,g
   g = g+1
```

Slika 2.5. *Pseudokod DE korištenog za FS*

2.4. Pregled postupaka za odabir značajki zasnovanih na omotačima

Korištenje postupaka za FS zasnovanih na omotačima proučavano je u brojnim radovima. Razvijene su mnogobrojne heuristike i metaheuristike, pri čemu je jedan od popularnijih pristupa uporaba metaheuristika kao što su evolucijski algoritmi ili algoritmi inteligencije rojeva (engl.

swarm intelligence). U nastavku je dan pregled odabranih radova u kojima su korišteni evolucijski algoritmi ili slični pristupi za FS zasnovani na omotačima.

Rad dostupan na [17] razmatra korištenje modificiranog algoritma mravljeg lava (engl. *ant lion optimizer*, ALO), algoritma zasnovanog na ponašanju mravljih lavova prilikom lova na plijen, u svrhu FS. Kako su rješenja algoritma kontinuirana, upotrijebljena je metoda diskretizacije rješenja za koju su korišteni različiti oblici prijenosnih funkcija koje definiraju vjerojatnost promjene binarnog elementa iz 0 u 1 i obratno. Dobiveni rezultati pokazuju kako određeni oblici prijenosnih funkcija (tzv. V-oblici) u prosjeku daju bolju točnost klasifikacije i manji broj odabranih značajki od standardnog ALO algoritma i algoritam gravitacijske pretrage (engl. *gravitational search algorithm*) te optimizacije rojem čestica (engl. *particle swarm optimization*).

Rad dostupan na [16] razmatra korištenje algoritma diferencijalne evolucije u svrhu FS za klasifikaciju. Korištena je inačica algoritma DE istovjetna inačici opisanoj u poglavlju 2.3. s *rand/1* mutacijom u kombinaciji s *k*-NN klasifikatorom, a konačno rješenje dobiveno je iz arhive prethodno pronađenih kvalitetnih rješenja korištenjem metode k-rezova. Vektori populacije diskretizirani su u svrhu provođenja *k*-NN algoritma, a ukupna mjera dobrote rješenja ovisna je o točnosti klasifikacije i broju odabranih značajki. Rezultati klasifikacije (točnost) s korištenim algoritmom DE u većini slučajeva bili su bolji u odnosu na klasifikaciju sa svim značajkama i u odnosu na druge testirane metode zasnovane na omotačima.

Rad dostupan na [18] uspoređuje neke od češće korištenih bio-inspiriranih algoritama optimizacije za FS. Svi razmatrani algoritmi (osim genetskog algoritma koji radi s diskretnim vrijednostima) koriste postupak diskretizacije zasnovan na pragu u svrhu dobivanja binarnih rješenja. Gledajući F_1-mjere klasifikacije, DE algoritam daje najbolje rezultate, a algoritam Jaya i algoritam umjetne kolonije pčela (engl. *artificial bee colony*) daju sljedeće najbolje rezultate, dok su ostali algoritmi dali slabije rezultate. DE algoritam također je rezultirao i najvećom stabilnošću u smislu višestrukih izvođenja algoritma.

Rad dostupan na [19] daje temeljit pregled češće korištenih metaheuristika, prvenstveno evolucijskih algoritama, za FS. Kao osnovne tehnike FS zasnovane na evolucijskim algoritmima navode se genetski algoritmi, genetsko programiranje, optimizacija rojem čestica i algoritam kolonije mrava (engl. *ant colony optimization*). Kao osnovne primjene takvih algoritama navode se obrada slika i signala, biološke i biomedicinske primjene, poslovne i financijske primjene te primjene za web servise i mrežne usluge. Naposljetku je dan pregled izazova i potencijalnih problema koji se pojavljuju kod navedenih pristupa FS, konkretno navodeći skalabilnost,

računalne zahtjeve, korištene mjere kvalitete rješenja, način reprezentacije rješenja, višeciljni FS, izgradnju novih značajki temeljenih na postojećima te utjecaj broja instanci u skupu podataka na performanse algoritma.

3. OSTVARENO PROGRAMSKO RJEŠENJE

Programsko rješenje ostvareno je u programskom jeziku Python. Ono omogućuje učitavanje skupa podataka u CSV (engl. *comma separated values*) formatu te provođenje FS korištenjem DE kao omotača na učitanom skupu podataka. Korisniku je omogućeno postavljanje vrijednosti parametara DE i broja ponavljanja njegovog izvođenja, odabir metode diskretizacije, odabir metode upravljanja vrijednostima parametara F i CR u DE te postavljanje vrijednosti parametra k u algoritmu k-NN. Nakon što algoritam završi s izvršavanjem, korisniku su dostupni podatci o učinkovitosti algoritma.

3.1. Način rada programskog rješenja

Ulazi programskog rješenja su skup podataka u CSV formatu, veličina populacije NP, broj iteracija jednog izvođenja DE g_{max}, ukupni broj izvođenja DE $n_{izvođenja}$ te parametar k algoritma k-NN. Osim navedenih podataka, korisnik također odabire postupak diskretizacije koji će biti korišten za vrednovanje rješenja u DE te metodu upravljanja parametrima F i CR. Mogući postupci diskretizacije su postupak zasnovan na logističkoj funkciji (*binDE*) određen izrazima (2-19) i (2-20), postupak zasnovan na normalizaciji (*normDE*) određen izrazima (2-21) i (2-22) te postupak zasnovan na usporedbi s vrijednosti praga (*pragDE*) određen izrazom (2-23), dok su mogući postupci upravljanja parametrima F i CR postavljanje istih na stalnu vrijednost za cijelo vrijeme izvršavanja (fiksno postavljene na jednu od najčešće korištenih postavki, $CR = 0.9$ i $F = 0.5$ [12]), nasumično generiranje novih vrijednosti u svakoj generaciji na razini populacije te pristup samopodešavanjem opisan u potpoglavlju 2.3.1. Učitani skup podataka dijeli se na podskupove za treniranje, validaciju i testiranje u fiksno definiranom omjeru 60/20/20.

Nakon što su svi ulazi definirani, inicijalizira se populacija rješenja koja se odmah zatim i vrednuje te se postupci mutacije, križanja i selekcije ponavljaju dok se ne dosegne zadani broj iteracija g_{max}. Pri koraku inicijalizacije, svi elementi rješenja se postavljaju na vrijednosti u intervalu [0,1] osim u slučaju odabira *binDE* diskretizacije, kada se svi elementi rješenja postavljaju na vrijednosti u intervalu [-1,1]. Također, u slučaju odabira *pragDE* diskretizacije, vrijednosti elemenata populacije se za cijelo vrijeme izvršavanja algoritma održavaju unutar intervala [0,1]. Pri koraku selekcije, generira se novi vektor dobiven diskretizacijom pokusnog vektora koji se zatim koristi za klasifikaciju učitanog skupa podataka algoritmom k-NN na validacijskom podskupu, pri čemu se kao mjera kvalitete klasifikacije koristi F_1-mjera određena izrazom (2-5). Izvršavanje algoritma se ponavlja dok se ne dosegne zadani broj ponavljanja $n_{izvođenja}$ te se spremaju najbolja rješenja

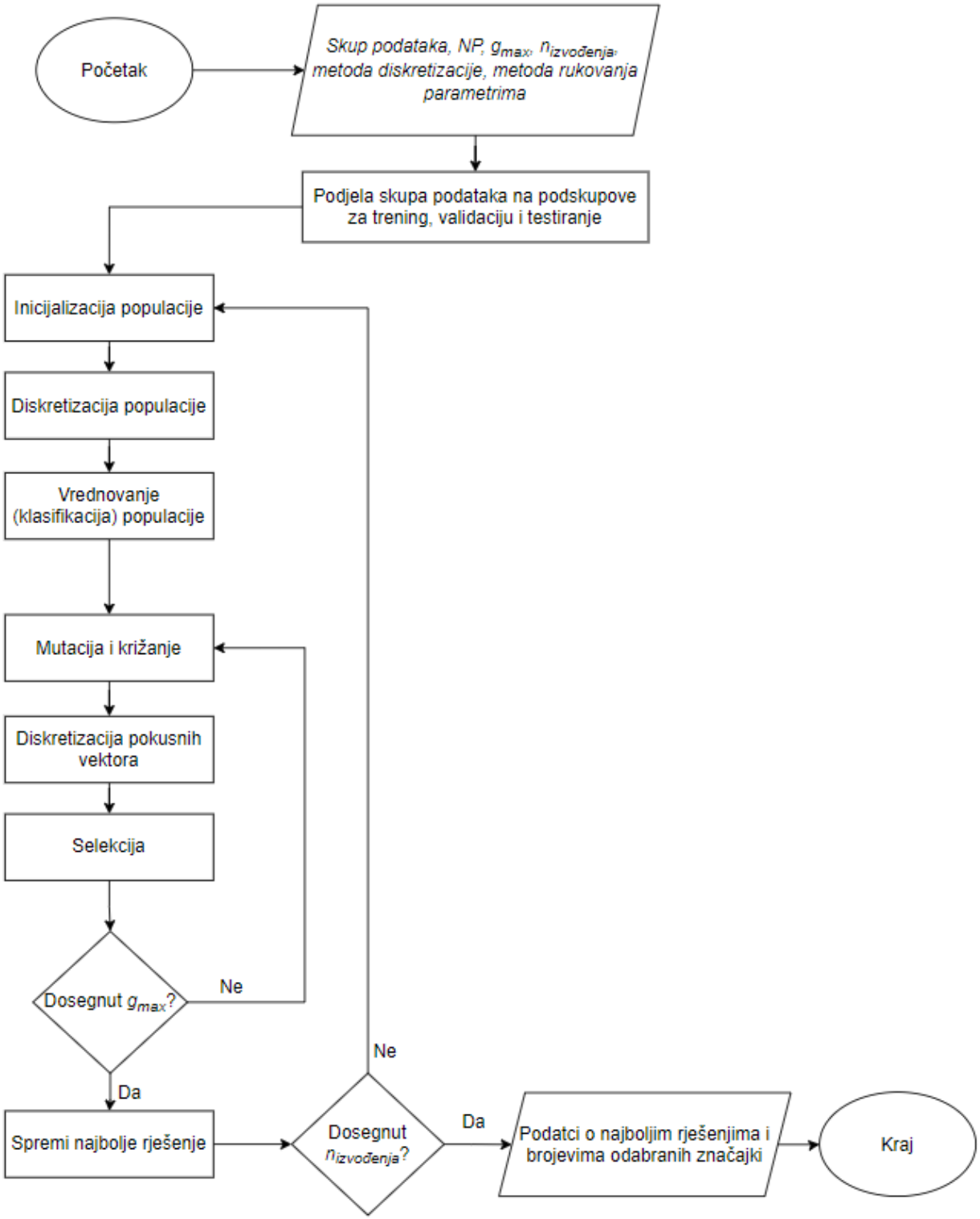
svakog izvođenja. Naposljetku se najbolja dobivena rješenja klasificiraju na podskupu za testiranje. F_1-mjere najboljih rješenja (na validacijskom i testnom podskupu) i broj odabranih značajki predstavljaju izlaz programskog rješenja te se računa njihova maksimalna i minimalna vrijednost, aritmetička sredina te standardna devijacija za uzorak

$$s = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2\,, \tag{2-24}$$

gdje je N broj uzoraka, x_i i-ti uzorak, a $\bar{x}$ aritmetička sredina svih uzoraka. Nadalje, računa se korelacija F_1-mjera dobivenih na validacijskom podskupu i F_1-mjera dobivenih na testnom podskupu korištenjem Pearsonovog koeficijenta korelacije

$$r_{xy} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}, \tag{2-25}$$

gdje je N broj uzoraka u skupovima podataka X i Y, a $\bar{x}$ i $\bar{y}$ aritmetička sredina svih vrijednosti skupa X, odnosno Y. Uz navedene podatke, računa se i F_1-mjera dobivena klasifikacijom skupa podataka s korištenjem svih značajki kako bi se mogla vrednovati učinkovitost FS. Dijagram toka programskog rješenja prikazan je na slici 3.1.



Slika 3.1. *Dijagram toka programskog rješenja*

3.2. Prikaz i način uporabe programskog rješenja

Pokretanjem programskog rješenja u konzoli, korisniku se nudi odabir željene datoteke s podatcima, parametara DE (veličina populacije i broj iteracija), broj ponavljanja izvršavanja DE, odabir postupka diskretizacije i postupka upravljanja parametrima *F* i *CR* te parametra *k* u algoritmu *k*-NN. Ako je odabrana neispravna vrijednost parametra, od korisnika se zahtjeva ponovni unos. Primjer unosa ulaznih parametara prikazan je na slici 3.2.

```
C:\Users\korisnik\Desktop\Diplomski>python main.py
Enter csv name: biodeg
Enter population size: 50
Enter number of iterations per single DE algorithm run: 100
Enter number of DE algorithm runs: 5
Choose discretization method:
1 - sigmoid
2 - normalization
3 - threshold: 1
Choose parameter control method:
1 - constant values
2 - random values for each generation
3 - self-adapting values: 1
Choose value of k for k-NN: 5
```

Slika 3.2. *Unos ulaznih parametara programskog rješenja*

Nakon definiranja potrebnih parametara, izvršava se algoritam DE. Korisniku je omogućeno praćenje stanja izvršavanja na način da je u konzoli vidljivo trenutno izvođenje i trenutna iteracija, kao i očekivano preostalo vrijeme izvođenja.

```
Run 1 of 5
100%|████████████████████████████████████████████████████████| 100/100 [00:49<00:00,  2.02it/s]
Run 2 of 5
100%|████████████████████████████████████████████████████████| 100/100 [00:49<00:00,  1.93it/s]
Run 3 of 5
 30%|████████████                                             | 30/100 [00:15<00:34,  2.04it/s]
```

Slika 3.3. *Ispis tijekom izvođenja programskog rješenja*

Po završetku izvršavanja programskog rješenja, F_1-mjera dobivena klasifikacijom skupa podataka s korištenjem svih značajki, podatci o kvaliteti najboljih rješenja, odnosno F_1-mjerama dobivenima klasifikacijom najboljih rješenja na validacijskom i testnom podskupu, podatci o odabranom broju značajki te podatci o korelaciji rezultata klasifikacije na validacijskom i testnom podskupu spremaju se u tekstualnu (*.txt*) datoteku istog imena kao i korišteni skup podataka. Primjer ispisa rezultata izvođenja programskog rješenja prikazan je na slici 3.4.

```
F1 score with all features selected: 0.8436018957345972

Results on validation dataset:
Max. F1 score: 0.9052132701421801
Min. F1 score: 0.8862559241706162
Average F1 score: 0.8947867298578199
F1 score standard deviation: 0.006993979027228621

Results on test dataset:
Max. F1 score: 0.8625592417061612
Min. F1 score: 0.8104265402843602
Average F1 score: 0.8393364928909953
F1 score standard deviation: 0.014556271089781287

Max. number of selected features: 30
Min. number of selected features: 18
Average number of selected features: 25.7
Standard deviation of selected features: 4.110960958218893

Pearson correlation of validation and test F1 scores:
[[ 1.         -0.36281049]
 [-0.36281049  1.        ]]
```

Slika 3.4. *Primjer rezultata izvođenja programskog rješenja*

4. EKSPERIMENTALNA ANALIZA

Cilj eksperimentalne analize je ispitivanje razlika između tehnika diskretizacije rješenja te ispitivanje utjecaja odabira vrijednosti parametara *CR* i *F* u algoritmu DE korištenom za FS. Mjere korištene za usporedbu pojedinih metoda su F_1-mjera klasifikacije i smanjenje broja značajki. Algoritam *k*-NN korišten je kao klasifikator u svim eksperimentima, dok je za strategiju mutacije u DE odabrana *rand/1* strategija.

U svrhu eksperimentalne analize odabrano je deset skupova podataka namijenjenih problemu klasifikacije s isključivo numeričkim značajkama. Odabrani su skupovi iz različitih domena koji se razlikuju po broju uzoraka, značajki i klasa. Svi skupovi podataka preuzeti su s UCI repozitorija [20]. Korišteni skupovi podataka prikazani su u tablici 4.1.

Tablica 4.1. *Korišteni skupovi podataka*

Oznaka	Ime skupa	Broj uzoraka	Broj značajki	Broj klasa
S1	QSAR biodegradation	1055	41	2
S2	Ionosphere	351	34	2
S3	LSVT voice rehabilitation	126	309	2
S4	Libras movement	360	91	15
S5	Musk (Version 1)	476	168	2
S6	Parkinsons	197	23	2
S7	Image Segmentation	210	19	7
S8	Sonar	208	60	2
S9	Urban Land Cover	168	148	9
S10	Wine	178	13	3

4.1. Postavke eksperimenta

Korišteni skupovi podataka podijeljeni su na podskupove za trening, validaciju i testiranje u omjeru 60/20/20, pri čemu je kao korak predobrade podataka provedena standardizacija podataka, odnosno postupak svođenja svih značajki na značajke s aritmetičkom sredinom $\mu = 0$ i standardnom devijacijom $\sigma = 1$. Navedeno je ostvareno oduzimanjem aritmetičke sredine od vrijednosti značajke te dijeljenjem sa standardnom devijacijom. Podatci su u svakom izvođenju podijeljeni u identične podskupove kako bi se dobio bolji uvid u razlike performansi DE s različitim tehnikama diskretizacije rješenja, odnosno različitim vrijednostima parametara DE. Parametar *k* algoritma *k*-NN koji određuje veličinu susjedstva postavljen je na *k*=5, jednu od češće korištenih vrijednosti u literaturi [21].

Veličina populacije *NP* u DE postavljena je na vrijednost 50, često korištenu u literaturi [22], dok je broj iteracija jednog izvođenja algoritma DE g_{max} postavljen na vrijednost 100. Kako je DE stohastički algoritam, potrebno je provesti veću količinu izvođenja kako bi se dobio općeniti uvid u njegovu učinkovitost. U tu svrhu je broj izvođenja DE ($n_{izvođenja}$) postavljen na vrijednost 30. Korišteni parametri DE prikazani su u tablici 4.2.

Tablica 4.2. *Korišteni parametri DE*

Parametar	**Vrijednost**
NP	50
g_{max}	100
$n_{izvođenja}$	30

U svrhu analize rezultata spremani su najbolji rezultati dobiveni svakim izvođenjem algoritma DE, pri čemu je kao mjera kvalitete korištena F_1-mjera. Također, spremane su količina odabranih značajki najboljih rješenja kako bi se moglo usporediti smanjenje broja značajki između različitih tehnika diskretizacije rješenja, odnosno različitih vrijednosti parametara DE.

4.2. Rezultati

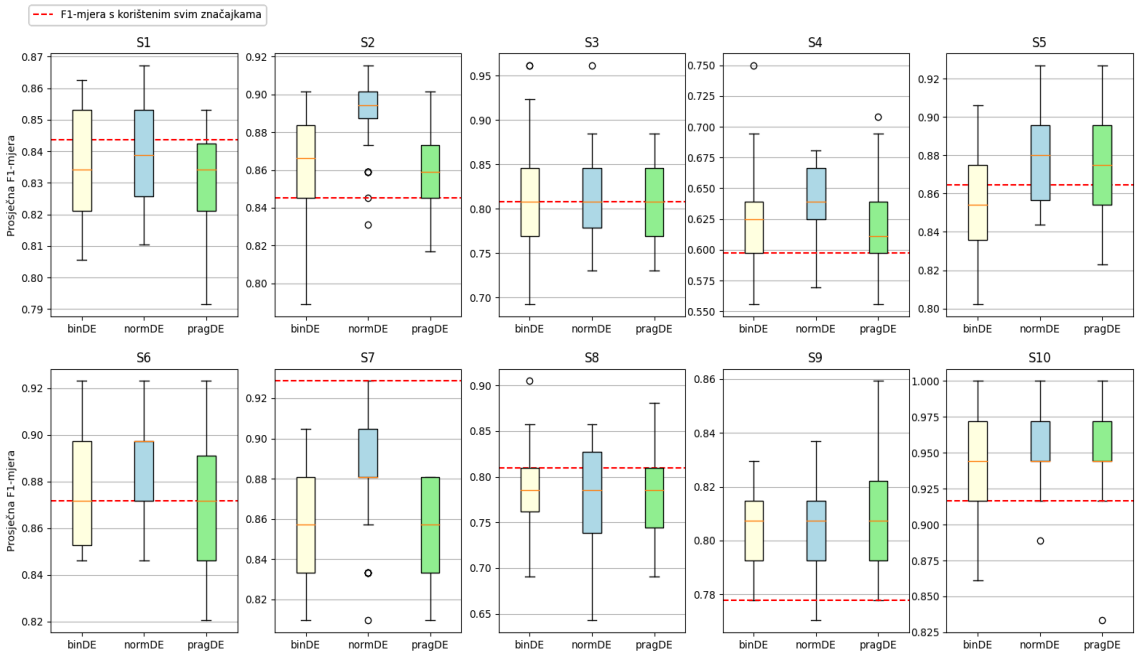
U nastavku su dani rezultati eksperimentalne analize. Prvo potpoglavlje prikazuje rezultate usporedbe tehnika diskretizacija rješenja, dok drugo potpoglavlje prikazuje rezultate ispitivanja utjecaja odabira vrijednosti parametara DE.

4.2.1. Utjecaj tehnike diskretizacije rješenja

Cilj prvog dijela eksperimentalne analize je ispitivanje utjecaja odabira tehnike diskretizacije rješenja na performanse algoritma DE korištenog za FS. Kao testirane tehnike diskretizacije rješenja odabrane su prethodno opisane tehnike *binDE*, *normDE* i *pragDE*. Ostvareni rezultati temeljeni na F_1-mjerama prikazani su u tablici 4.3 te grafički na slici 4.1.

Tablica 4.3. *Ostvarene F_1-mjere u usporedbi tehnika za diskretizaciju rješenja*

Skup	**F_1-mjera sa svim značajkama**	**Diskretizacija**	$\overline{F_1}$	F_{1max}	F_{1min}	σ
S1	0.84360	*binDE*	0.83428	0.86256	0.80569	0.01806
		normDE	**0.83949**	**0.86730**	**0.81043**	0.01543
		pragDE	0.83065	0.85308	0.79147	**0.01534**
S2	0.84507	*binDE*	0.86056	0.90141	0.78873	0.02777
		normDE	**0.88732**	**0.91549**	**0.83099**	**0.02026**
		pragDE	0.85822	0.90141	0.81690	0.02424
S3	0.80769	*binDE*	0.81154	**0.96154**	0.69231	0.06802
		normDE	0.81282	**0.96154**	**0.73077**	**0.04599**
		pragDE	**0.81667**	0.88462	**0.73077**	0.04811
S4	0.59722	*binDE*	0.62731	**0.75000**	0.55556	0.04047
		normDE	**0.63565**	0.68056	**0.56944**	**0.03276**
		pragDE	0.62361	0.70833	0.55556	0.03814
S5	0.86458	*binDE*	0.85625	0.90625	0.80208	0.02741
		normDE	**0.87813**	**0.92708**	**0.84375**	**0.02434**
		pragDE	0.87743	**0.92708**	0.82292	0.02619
S6	0.87179	*binDE*	0.87436	**0.92308**	**0.84615**	**0.02166**
		normDE	**0.88120**	**0.92308**	**0.84615**	0.02180
		pragDE	0.86667	**0.92308**	0.82051	0.02465
S7	0.92857	*binDE*	0.85635	0.90476	**0.80952**	0.02760
		normDE	**0.88095**	**0.92857**	**0.80952**	0.03249
		pragDE	0.85556	0.88095	**0.80952**	**0.02334**
S8	0.80952	*binDE*	**0.78968**	**0.90476**	**0.69048**	0.04847
		normDE	0.77540	0.85714	0.64286	0.05855
		pragDE	0.77698	0.88095	**0.69048**	**0.04699**
S9	0.77778	*binDE*	0.80519	0.82963	**0.77778**	**0.01418**
		normDE	0.80543	0.83704	0.77037	0.01695
		pragDE	**0.80840**	**0.85926**	**0.77778**	0.01991
S10	0.91667	*binDE*	0.93981	**1.00000**	0.86111	0.04067
		normDE	0.95000	**1.00000**	**0.88889**	**0.02670**
		pragDE	**0.95370**	**1.00000**	0.83333	0.03672



Slika 4.1. *Dijagram pravokutnika ostvarenih F_1-mjera u usporedbi tehnika za diskretizaciju rješenja*

Iz rezultata je vidljivo kako je *normDE* tehnika ostvarila najvišu prosječnu vrijednost F_1-mjere na šest od deset testiranih skupova podataka, a najnižu prosječnu vrijednost F_1-mjere na samo jednome skupu (S8). Tehnika *pragDE* ostvarila je najbolje prosječne rezultate na trima skupovima i najgore prosječne rezultate na pet skupova dok je tehnika *binDE* ostvarila najbolje prosječne rezultate na jednome skupu i najgore prosječne rezultate na četirima skupovima. Razlika prosječnih F_1-mjera dobivenih različitim tehnikama diskretizacije u većini slučajeva je vrlo mala (manja od 1%) s jedinim značajnim iznimkama na skupovima S2 i S7, na kojima je najuspješnija tehnika (*normDE*) ostvarila oko 2.5% višu F_1-mjeru od drugih tehnika. Tehnika *normDE* istakla se i po količini ostvarenih najviših vrijednosti F_1-mjere, ne uspjevši ju ostvariti na samo trima skupovima (S4, S8, S9). Također, u osam od deset slučajeva ostvarila je najvišu minimalnu vrijednost F_1-mjere od svih testiranih tehnika diskretizacije. Za razliku od prosječnih F_1-mjera, kod maksimalnih i minimalnih vrijednosti primjetljive su povremene veće oscilacije između različitih tehnika diskretizacija koje na određenim skupovima (S3, S4, S7, S8) dosežu raspon od 5% do čak 8%. Rezultatski najstabilnijom tehnikom pokazala se tehnika *normDE*, ostvarujući najnižu vrijednost standardne devijacije u pet slučajeva, dok su *pragDE* i *binDE* najnižu vrijednost standardne devijacije ostvarili u tri, odnosno dva navrata. Najmanje stabilnom tehnikom pokazala se tehnika *binDE*, ostvarujući najvišu vrijednost standardne devijacije u šest od deset slučajeva,

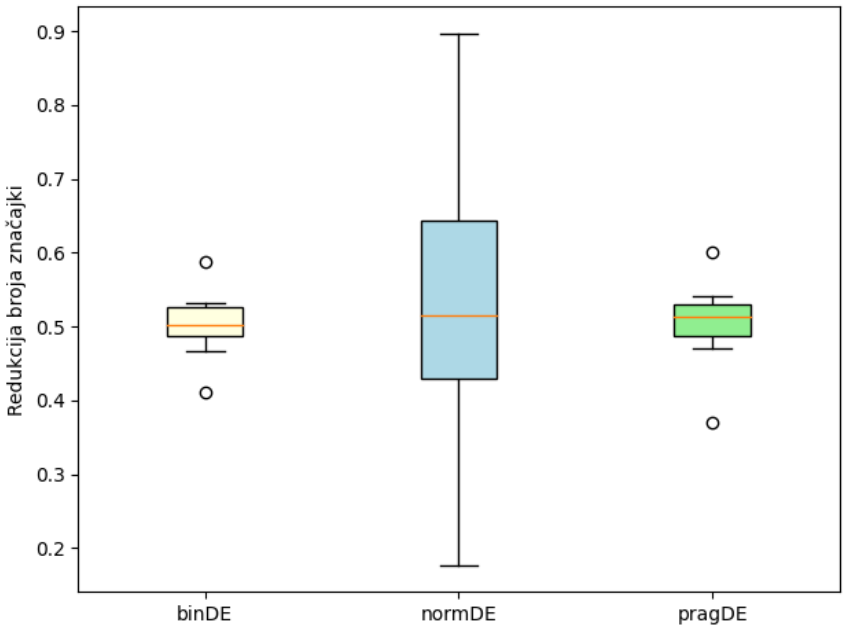
no treba naglasiti kako su razlike standardnih devijacija u većini slučajeva bile male (sve tri tehnike unutar 1%). Promatrajući F_1-mjere ostvarene korištenjem svih značajki i F_1-mjere ostvarene korištenjem FS, vidljivo je kako FS ostvaruje bolje rezultate u čak sedam od deset slučajeva, dok u preostala tri slučaja rezultira neznatnim smanjenjem F_1-mjere (uz iznimku skupa S7 gdje dolazi do smanjenja za 5-7% ovisno o korištenoj tehnici diskretizacije).

Osim F_1-mjera, promatrana je i redukcija broja značajki dobivena pojedinom tehnikom diskretizacije rješenja. Ostvareni rezultati prikazani su u tablici 4.4.

Tablica 4.4. *Ostvarene redukcije broja značajki u usporedbi tehnika za diskretizaciju rješenja*

Skup	**Diskretizacija**	**Prosječna redukcija br. značajki**	**Max. redukcija br. značajki**	**Min. redukcija br. značajki**	$\boldsymbol{\sigma}$
S1	*binDE*	46.75%	**60.98%**	31.71%	**6.18%**
	normDE	41.06%	**60.98%**	21.95%	11.67%
	pragDE	**48.29%**	58.54%	**34.15%**	6.19%
S2	*binDE*	58.82%	73.53%	41.18%	8.64%
	normDE	**89.71%**	**94.12%**	**70.59%**	**5.11%**
	pragDE	60.00%	79.41%	47.06%	8.23%
S3	*binDE*	49.86%	55.16%	**45.16%**	2.58%
	normDE	49.28%	**93.55%**	24.19%	17.16%
	pragDE	**49.94%**	56.77%	44.84%	**2.52%**
S4	*binDE*	53.15%	62.22%	**45.56%**	**4.53%**
	normDE	**67.07%**	**84.44%**	34.44%	11.31%
	pragDE	53.44%	63.33%	44.44%	5.18%
S5	*binDE*	51.14%	57.23%	**41.57%**	4.05%
	normDE	48.47%	**68.07%**	28.31%	9.23%
	pragDE	**51.41%**	57.83%	40.96%	**3.22%**
S6	*binDE*	48.33%	68.18%	27.27%	9.95%
	normDE	**53.79%**	**77.27%**	31.82%	15.57%
	pragDE	51.97%	**77.27%**	**36.36%**	**8.91%**
S7	*binDE*	**50.35%**	**68.42%**	26.32%	8.82%
	normDE	31.40%	52.63%	10.53%	13.99%
	pragDE	47.02%	57.89%	**36.84%**	**8.12%**
S8	*binDE*	53.06%	61.67%	**41.67%**	5.76%
	normDE	**65.67%**	**85.00%**	**41.67%**	11.49%
	pragDE	54.17%	66.67%	40.00%	**5.48%**
S9	*binDE*	49.91%	61.90%	**42.18%**	**4.48%**
	normDE	**60.09%**	**85.03%**	26.53%	14.36%
	pragDE	51.13%	61.22%	**42.18%**	4.65%
S10	*binDE*	**41.03%**	**61.54%**	**23.08%**	11.13%
	normDE	17.69%	53.85%	7.69%	10.91%
	pragDE	36.92%	53.85%	15.38%	**10.18%**

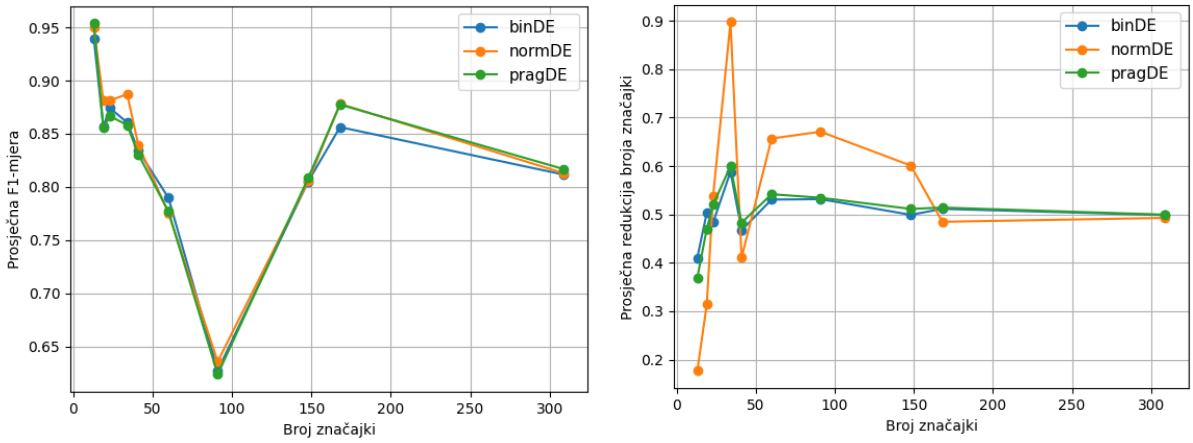
Tehnike *binDE* i *pragDE* ostvarile su podjednake stope redukcije na svim skupovima podataka, uvijek dajući redukciju u rasponu od 40 do 60 posto. Tehnika *normDE* ponovno se istakla, ostvarivši najvišu prosječnu stopu redukcije broja značajki u pet od deset slučajeva te ostvarivši najvišu maksimalnu dobivenu redukciju broja značajki u osam od deset slučajeva. Međutim, ista metoda ostvarila je najnižu minimalnu dobivenu redukciju broja značajki u sedam od deset slučajeva. Razlog tomu vidljiv je u iznosima standardne devijacije redukcije broja značajki, gdje je *normDE* ostvarila najveću vrijednost u osam od deset slučajeva, pritom većinom dosežući vrijednosti veće od 10% dok su preostale dvije metode ostvarivale standardnu devijaciju nižu od 10% na svim skupovima podataka izuzev skupa S10. Na slici 4.2 prikazan je dijagram pravokutnika prosječnih vrijednosti redukcije broja značajki.



Slika 4.2. *Dijagram pravokutnika ostvarenih prosječnih vrijednosti redukcije broja značajki u usporedbi tehnika za diskretizaciju rješenja*

Dijagram prikazuje već spomenutu znatno veću nestabilnost rezultata dobivenih tehnikom *normDE*. Navedeno čini tehniku korisnom za slučajeve s višestrukim pokretanjima algoritma, no manje efikasnu pri jednostrukim izvođenjima, u kojem slučaju tehnike *binDE* i *pragDE* ostvaruju znatno konzistentnije iznose redukcije broja značajki, većinom uklanjajući oko polovice značajki.

Osim samih vrijednosti F_1-mjera i redukcija broja značajki, promatran je i utjecaj broja značajki korištenih skupova podataka na dobivene vrijednosti. Grafovi koji prikazuju dobivene ovisnosti prikazani su na slici 4.3.



Slika 4.3. *Ovisnost prosječne F_1-mjere i prosječne redukcije broja značajki o broju značajki skupa podataka u usporedbi tehnika za diskretizaciju rješenja*

Na slici 4.3. može se primijetiti kako *normDE* tehnika ostvaruje uvjerljivo najviše F_1-mjere od svih tehnika ponajviše pri skupovima s niskim brojem značajki, dok se pri skupovima s većim brojem značajki tehnika *pragDE* pokazuje podjednako dobrom, a u određenim slučajevima i neznatno boljom. Graf ovisnosti prosječne redukcije broja značajki o broju značajki skupa podataka pokazuje kako povećavanjem broja značajki stopa redukcije značajki svih tehnika diskretizacije teži k 50%. Pritom je vidljivo kako su oscilacije pri manjim brojevima značajki znatno više za tehniku *normDE* u odnosu na preostale tehnike. Međutim, potrebno je naglasiti kako većina korištenih skupova podataka sadrži manje od 100 značajki te bi za daljnju analizu bilo potrebno koristiti veću količinu skupova podataka sa znatno većim brojem značajki.

Kao što je prethodno spomenuto, najbolja rješenja u pojedinim izvođenjima algoritma DE birana su na temelju ostvarenih F_1-mjera na validacijskom podskupu podataka te su ista rješenja zatim vrednovana na testnom podskupu podataka. Kako bi se utvrdila moguća veza između performansi pojedinog rješenja DE na validacijskom i testnom podskupu, mjeren je Pearsonov koeficijent korelacije između F_1-mjeri dobivenih na validacijskom i testnom podskupu za svaki skup izvođenja algoritma DE na istome skupu podataka korištenjem iste tehnike diskretizacije rješenja. Dobivene vrijednosti korelacije prikazane su u tablici 4.5. Reci bez vrijednosti predstavljaju slučajeve u kojima nije bilo moguće izračunati koeficijent korelacije jer je jedna od varijabli imala standardnu devijaciju jednaku nuli.

Tablica 4.5. *Pearsonovi koeficijenti korelacije između F_1-mjera ostvarenih na validacijskom i testnom podskupu*

Skup	Diskretizacija	Korelacija
S1	*binDE*	-0.02939
	normDE	-0.33364
	pragDE	0.18709
S2	*binDE*	0.18594
	normDE	-0.14482
	pragDE	0.10792
S3	*binDE*	0.00000
	normDE	-0.02106
	pragDE	0.00000
S4	*binDE*	0.46458
	normDE	0.20175
	pragDE	0.60893
S5	*binDE*	0.15995
	normDE	0.00591
	pragDE	-0.17596
S6	*binDE*	-
	normDE	0.17943
	pragDE	0.23576
S7	*binDE*	0.00000
	normDE	0.00000
	pragDE	0.00000
S8	*binDE*	-0.14778
	normDE	-0.05515
	pragDE	-
S9	*binDE*	0.07272
	normDE	0.10968
	pragDE	0.10937
S10	*binDE*	-
	normDE	-
	pragDE	-

Iz tablice je vidljivo kako ne postoji značajna korelacija između rezultata ostvarenih na validacijskom i testnom podskupu neovisno o korištenoj tehnici diskretizacije. Navedeno pokazuje kako uzimanje rješenja DE s najboljim performansama na validacijskom podskupu ne rezultira nužno najboljim mogućim performansama na testnome podskupu. Međutim, nije moguće isključiti mogućnost da bi se drugačijim načinom vrednovanja performansi klasifikacije, primjerice

metodom k-preklopa/rezova, dobili drugačiji rezultati. Odgovarajući eksperimenti trebali bi biti provedeni u budućim istraživanjima.

4.2.2. Utjecaj odabira vrijednosti parametara diferencijalne evolucije

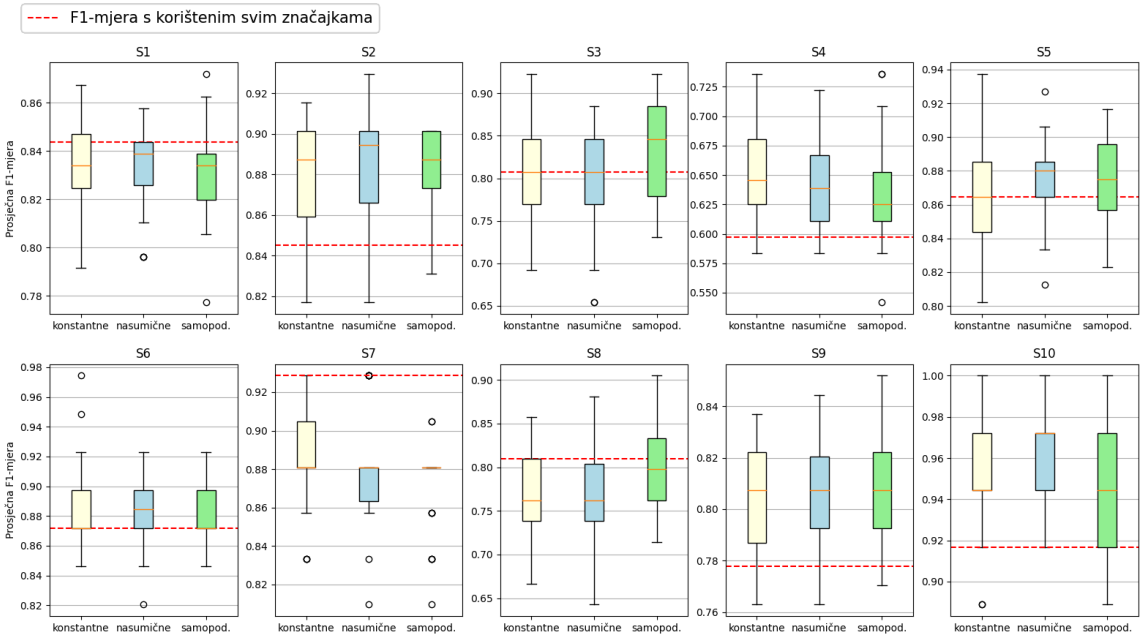
Performanse algoritma DE, kao i svih evolucijskih algoritama, su osjetljive na postavke parametara [15]. Cilj drugog dijela eksperimentalne analize je ispitivanje utjecaja odabira vrijednosti parametara *F* i *CR* algoritma DE korištenog za FS. U tu svrhu uspoređivani su sljedeći pristupi postavljanja vrijednosti parametara:

- postavljanje parametara na konstantne vrijednosti (odabrane su često korištene vrijednosti *CR* = 0.9 i *F* = 0.5 [12])
- nasumično generiranje vrijednosti $CR \in [0.1,1]$ i $F \in [0,1]$ u svakoj novoj iteraciji DE (identična vrijednost za sve članove populacije)
- metoda samopodešavanja vrijednosti parametara opisana u potpoglavlju 2.3.1. i dana izrazima (2-17) i (2-18) s vrijednostima parametara $\tau_1 = 0.1$, $\tau_2 = 0.1$, $F_l = 0.1$ i $F_u = 0.9$

Kao tehnika diskretizacije rješenja odabrana je tehnika *normDE* na temelju rezultata usporedbe tehnika diskretizacije u kojemu se pokazala najboljom po pitanju ostvarenih F_1-mjera. Ostali parametri DE postavljeni su na vrijednosti dane u tablici 4.2. Ostvareni rezultati u smislu F_1-mjera prikazani su u tablici 4.6 te grafički na slici 4.4.

Tablica 4.6. *Ostvarene F_1-mjere u analizi utjecaja odabira vrijednosti parametara DE*

Skup	**F_1-mjera sa svim značajkama**	**Odabir vrijednosti parametara**	$\overline{F_1}$	F_{1max}	F_{1min}	σ
S1	0.84360	*konstantne*	**0.83397**	0.86730	0.79147	0.01771
		nasumične	0.83302	0.85782	**0.79621**	**0.01650**
		samopodešavanje	0.83112	**0.87204**	0.77725	0.02013
S2	0.84507	*konstantne*	0.87887	0.91549	0.81690	0.02579
		nasumične	**0.88685**	**0.92958**	0.81690	0.02705
		samopodešavanje	0.88404	0.90141	**0.83099**	**0.01946**
S3	0.80769	*konstantne*	0.80897	**0.92308**	0.69231	0.06017
		nasumične	0.79359	0.88462	0.65385	0.06427
		samopodešavanje	**0.82821**	**0.92308**	**0.73077**	**0.05779**
S4	0.59722	*konstantne*	**0.65324**	**0.73611**	**0.58333**	**0.03638**
		nasumične	0.64167	0.72222	**0.58333**	0.03727
		samopodešavanje	0.63426	**0.73611**	0.54167	0.04546
S5	0.86458	*konstantne*	0.86563	**0.93750**	0.80208	0.03218
		nasumične	**0.87569**	0.92708	0.81250	**0.02368**
		samopodešavanje	0.87465	0.91667	**0.82292**	0.02439
S6	0.87179	*konstantne*	**0.88718**	**0.97436**	**0.84615**	0.02744
		nasumične	0.88205	0.92308	0.82051	**0.02293**
		samopodešavanje	0.88291	0.92308	**0.84615**	0.02491
S7	0.92857	*konstantne*	0.88254	**0.92857**	**0.83333**	0.02573
		nasumične	**0.88333**	**0.92857**	0.80952	0.03021
		samopodešavanje	0.87302	0.90476	0.80952	**0.02105**
S8	0.80952	*konstantne*	0.77063	0.85714	0.66667	0.05174
		nasumične	0.76746	0.88095	0.64286	0.05404
		samopodešavanje	**0.79603**	**0.90476**	**0.71429**	**0.04451**
S9	0.77778	*konstantne*	0.80617	0.83704	**0.76296**	**0.01936**
		nasumične	0.80519	0.84444	**0.76296**	0.02228
		samopodešavanje	**0.80716**	**0.85185**	0.77037	0.02370
S10	0.91667	*konstantne*	0.95278	**1.00000**	0.88889	0.03108
		nasumične	**0.95741**	**1.00000**	**0.91667**	**0.02894**
		samopodešavanje	0.95278	**1.00000**	0.88889	0.03510



Slika 4.4. *Dijagram pravokutnika ostvarenih F_1-mjera u analizi utjecaja odabira vrijednosti parametara DE*

Nijedan pristup se nije značajno istaknuo po pitanju postignutih prosječnih F_1-mjera. Nasumično biranje vrijednosti rezultiralo je najvišom prosječnom F_1-mjerom na četiri skupa, dok su oba preostala dva pristupa ostvarili najbolje rezultate u tri slučaja. Također, nijedan pristup nije ostvario najnižu prosječnu F_1-mjeru u više od četiri navrata. Razlike između prosječnih F_1-mjera ostvarenih različitim pristupima u velikoj većini slučajeva su vrlo male (manje od 1%) s jedinim značajnim iznimkama na skupovima S3 i S8 gdje je samopodešavanje ostvarilo barem 2% višu prosječnu F_1-mjeru od ostalih pristupa. Postavljanje parametara na konstantne vrijednosti i samopodešavanje parametara u većini slučajeva su ostvarili najviše vrijednosti F_1-mjere, dok je nasumičnim biranjem vrijednosti parametara to ostvareno u samo dva slučaja. Nijedan pristup se nije istaknuo po pitanju ostvarenih minimalnih vrijednosti F_1-mjera. Razlike u postignutim maksimalnim i minimalnim vrijednostima F_1-mjera nešto su izraženije u odnosu na razlike između prosječnih vrijednosti, ali su u velikoj većini slučajeva unutar 3% s najznačajnijim iznimkama na skupovima S6, gdje korištenje konstantnih vrijednosti ostvaruje 5% višu maksimalnu F_1-mjeru od ostalih pristupa, i S8, gdje samopodešavanje ostvaruje 5% višu minimalnu F_1-mjeru od ostalih pristupa. Svi pristupi su ostvarili minimalnu vrijednost standardne devijacije F_1-mjere u podjednakom broju slučaja te se vrijednosti standardnih devijacija razlikuju u vrlo malim količinama (unutar 1% u svih deset slučajeva).

Osim F_1-mjera, promatrana je i redukcija broja značajki dobivena pojedinim pristupom odabira vrijednosti parametara DE. Ostvareni rezultati prikazani su u tablici 4.7.

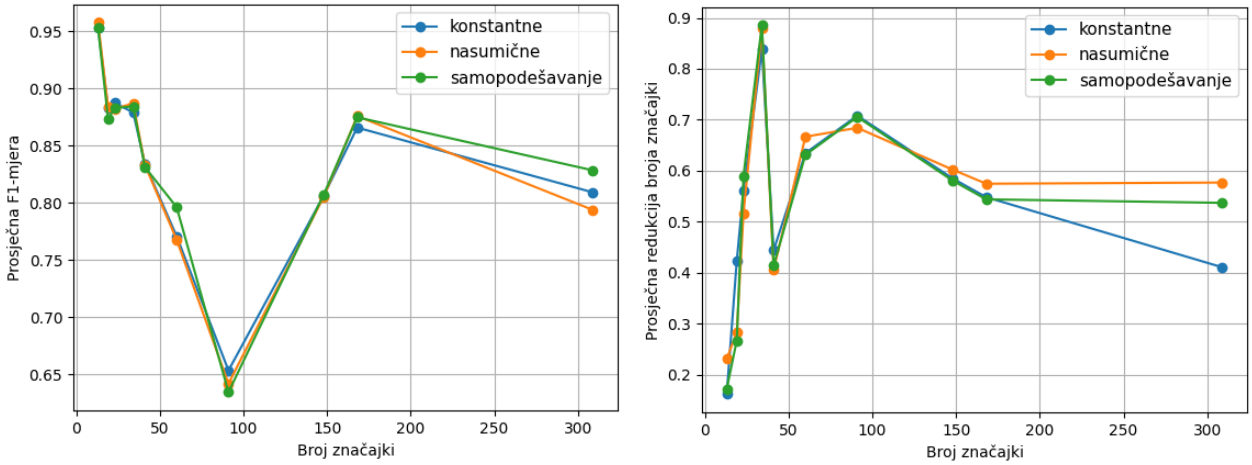
Tablica 4.7. *Ostvarene redukcije broja značajki u analizi utjecaja odabira vrijednosti parametara DE*

Skup	Odabir vrijednosti parametara	Prosječna redukcija br. značajki	Max. redukcija br. značajki	Min. redukcija br. značajki	σ
S1	*konstantne*	**44.39%**	**68.29%**	**21.95%**	11.63%
	nasumične	40.57%	63.41%	**21.95%**	12.33%
	samopodešavanje	41.38%	53.66%	**21.95%**	**9.58%**
S2	*konstantne*	83.82%	**94.12%**	**44.12%**	12.83%
	nasumične	88.04%	**94.12%**	73.53%	5.62%
	samopodešavanje	**88.53%**	**94.12%**	76.47%	**4.72%**
S3	*konstantne*	41.04%	80.58%	17.80%	17.95%
	nasumične	**57.68%**	**98.71%**	**15.53%**	23.07%
	samopodešavanje	53.69%	94.50%	15.86%	**16.31%**
S4	*konstantne*	**70.81%**	83.52%	35.16%	12.42%
	nasumične	68.42%	91.21%	**32.97%**	13.47%
	samopodešavanje	70.55%	**93.41%**	46.15%	**12.06%**
S5	*konstantne*	54.80%	77.98%	**16.67%**	14.46%
	nasumične	**57.44%**	**79.76%**	36.31%	9.51%
	samopodešavanje	54.38%	69.64%	38.69%	**7.84%**
S6	*konstantne*	56.09%	**91.30%**	**34.78%**	17.88%
	nasumične	51.59%	78.26%	**34.78%**	**14.88%**
	samopodešavanje	**58.99%**	**91.30%**	**34.78%**	16.42%
S7	*konstantne*	**42.28%**	**63.16%**	15.79%	**12.93%**
	nasumične	28.25%	**63.16%**	**10.53%**	14.40%
	samopodešavanje	26.67%	52.63%	**10.53%**	14.02%
S8	*konstantne*	63.44%	85.00%	**40.00%**	13.52%
	nasumične	**66.67%**	**86.67%**	46.67%	**9.59%**
	samopodešavanje	63.11%	83.33%	**40.00%**	10.39%
S9	*konstantne*	58.47%	82.43%	**28.38%**	14.35%
	nasumične	**60.20%**	**93.24%**	39.19%	12.33%
	samopodešavanje	58.04%	77.70%	33.78%	**10.59%**
S10	*konstantne*	16.15%	46.15%	**7.69%**	**8.89%**
	nasumične	**23.08%**	**61.54%**	**7.69%**	14.00%
	samopodešavanje	17.18%	38.46%	**7.69%**	9.62%

Nasumičan odabir vrijednosti parametara ostvario je najveću prosječnu redukciju broja značajki u pet od deset slučajeva, dok su konstantne vrijednosti i metoda samopodešavanje to uspjele u tri, odnosno dva navrata. Kao i u slučaju prosječnih F_1-mjera, prosječne vrijednosti redukcije broja

značajki nisu se znatno razlikovale između različitih pristupa odabira vrijednosti parametara. Na šest od deset skupova razlike su unutar 5%, dok su na skupovima S3 i S7 razlike najboljeg i najgoreg pristupa veće od 15%. Konstantne i nasumične vrijednosti parametara rezultirale su najvećim maksimalnim i minimalnim redukcijama broja značajki u većini slučajeva, dok je metoda samopodešavanja po tom pitanju ostvarivala nešto slabije rezultate. Međutim, metoda samopodešavanja pokazala se najstabilnijim pristupom ostvarivši najmanju standardnu devijaciju redukcije broja značajki na šest od deset skupova. Razlike između vrijednosti standardnih devijacija u većini slučajeva nisu prelazile iznos od 5%.

Kao i pri ispitivanju utjecaja tehnike diskretizacije rješenja, osim samih vrijednosti F_1-mjera i redukcija broja značajki, promatran je i utjecaj broja značajki korištenih skupova podataka na dobivene vrijednosti. Grafovi koji prikazuju dobivene ovisnosti prikazani su na slici 4.5.



Slika 4.5. *Ovisnost prosječne F_1-mjere i prosječne redukcije broja značajki o broju značajki skupa podataka u analizi utjecaja odabira vrijednosti parametara DE*

Na slici 4.5 vidljivo je kako nasumičan odabir vrijednosti parametara ostvaruje najbolje rezultate na skupovima s manje od 50 značajki po pitanju prosječnih F_1-mjera, dajući najvišu vrijednost na tri od pet takvih skupova, dok na preostalih pet skupova s većim brojem značajki niti jednom ne ostvaruje najvišu prosječnu F_1-mjeru. Relativne performanse nasumičnog odabira i metode samopodešavanja po pitanju ostvarenih F_1-mjera, ali i prosječne redukcije broja značajki, ne pokazuju značajnu korelaciju s brojem značajki korištenih skupova podataka. Međutim, nasumičan odabir vrijednosti parametara ostvaruje najbolje rezultate u vidu redukcije broja značajki na skupovima s više od 50 značajki, ostvarujući najvišu stopu prosječne redukcije broja značajki na četiri od pet takvih skupova.

5. ZAKLJUČAK

Odabir značajki predstavlja bitan korak predobrade skupa podataka za klasifikaciju. Popularna skupina metoda korištenih za rješavanje problema FS su metode zasnovane na omotačima, među kojima je i algoritam DE. Zadatak rada bio je opisati primjenu algoritma DE kao omotača za rješavanje problema FS te realizirati više inačica algoritma DE koje se razlikuju u mehanizmu diskretizacije rješenja, potrebnom zbog operatora mutacije koji izvorno ograničava DE na probleme kontinuirane optimizacije. Kako je algoritam DE osjetljiv na postavke parametara, promatran je i utjecaj odabira vrijednosti parametara DE. U tu svrhu razvijeno je programsko rješenje koje, uz ispis detaljnih rezultata izvođenja algoritma DE korištenog kao omotač za FS, korisniku omogućuje učitavanje željenog skupa podataka i odabir tehnike diskretizacije rješenja te načina biranja vrijednosti parametara DE, omogućujući jednostavniju provedbu eksperimentalne analize.

Eksperimentalnom analizom pokazano je kako odabir tehnike diskretizacije rješenja utječe na performanse algoritma DE. Uspoređene su tri tehnike za diskretizaciju rješenja, od kojih se, na temelju ostvarenih F_1-mjera klasifikacije, s neznatno boljim rezultatima, posebice na skupovima podataka s niskim brojem značajki, istaknula tehnika *normDE* zasnovana na normalizaciji, odnosno linearnom skaliranju svih elemenata članova populacije, pritom ostvarivši najmanje stabilne iznose redukcije broja značajki. Usporedba korištenja konstantnih vrijednosti parametara DE s korištenjem nasumično generiranih vrijednosti i vrijednosti dobivenih metodom samopodešavanja parametara pokazala je kako između navedenih pristupa ne postoji značajna razlika po pitanju ostvarenih F_1-mjera klasifikacije i redukcije broja značajki.

U budućem radu mogli bi se provesti statistički testovi za preciznije utvrđivanje razlika u performansama između pojedinih tehnika diskretizacije rješenja, odnosno pristupa odabira vrijednosti parametara DE. Također, moglo bi se razmotriti više tehnika diskretizacije i metoda upravljanja parametrima u svrhu pronalaska najkvalitetnijeg pristupa implementacije DE za FS. Osim toga, moguća je i detaljna analiza utjecaja drugih parametara DE, kao što su veličina populacije i broj iteracija na performanse algoritma. Testiranjem različitih metoda vrednovanja performansi klasifikacije moglo bi se doći do drugačijih zaključaka o performansama pojedinih tehnika diskretizacije rješenja, odnosno metoda upravljanja parametrima, ali i drugih mjera poput korelacije između ostvarenih rezultata na validacijskom i testnom skupu podataka.

LITERATURA

[1] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Computers & Electrical Engineering, No. 1, Vol. 40, pp. 16–28, siječanj 2014.

[2] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Fourth Edition, Academic Press, Cambridge, 2008.

[3] N. Japkowicz, M. Shah, Evaluating Learning Algorithms: A Classification Perspective, Cambridge University Press, Cambridge, 2011.

[4] Multi-Class Metrics Made Simple, Part II: the F1-score, dostupno na: https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1 [11.7.2022.]

[5] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, Wiley-Interscience, Hoboken, 2000.

[6] D. Bajer, B. Zorić, M. Dudjak, Wrapper-based feature selection: how important is the wrapped classifier?, 2020 International Conference on Smart Systems and Technologies, pp. 97–105, listopad 2020.

[7] Y. Chen, D. Miao, R. Wang, A rough set approach to feature selection based on ant colony optimization, Pattern Recognition Letters, No. 3, Vol. 31, pp. 226-233, veljača 2010.

[8] A. Jović, K. Brkić, N. Bogunović, A review of feature selection methods with applications, 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, pp. 1200–1205, svibanj 2015.

[9] D. Bajer, B. Zorić, M. Dudjak, G. Martinović, Benchmarking bio-inspired computation algorithms as wrappers for feature selection, Acta Electrotechnica et Informatica, No.2, Vol. 20, pp. 35–43, srpanj 2020.

[10] A. P. Engelbrecht, Computational Intelligence: An Introduction, Second Edition, Wiley Publishing, Pretoria, rujan 2007.

[11] A. P. Engelbrecht, G. Pampara, Binary differential evolution strategies, Proceedings of the 2007 IEEE Congress on Evolutionary Computation Conference, pp. 1942–1947, rujan 2007.

[12] D. Bajer, Adaptive k-tournament mutation scheme for differential evolution, Applied Soft Computing, Vol. 85, prosinac 2019.

[13] K. Price, R. Storn, J. A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer-Verlag New York, Inc., New York, 2005.

[14] S. Das, A. Konar, U. K. Chakraborty, Two improved differential evolution schemes for faster global search, In Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation (GECCO), pp. 991–998, lipanj 2005.

[15] J. Brest, S. Greiner, B. Bošković, M. Mernik, V. Žumer, Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems, IEEE Transactions on Evolutionary Computation, No. 6, Vol. 10, pp. 646-657, prosinac 2006.

[16] G. Martinović, D. Bajer, B. Zorić, A differential evolution approach to dimensionality reduction for classification needs, International Journal of Applied Mathematics and Computer Science, No. 1, Vol. 24, pp. 111–122, ožujak 2014.

[17] S. Mirjalili, A. Lewis, S-shaped versus v-shaped transfer functions for binary particle swarm optimization, Swarm and Evolutionary Computation, Vol. 9, pp. 1–14, travanj 2013.

[18] D. Bajer, B. Zorić, M. Dudjak, G. Martinović, Evaluation and analysis of bioinspired optimisation algorithms for feature selection, 2019 IEEE 15th International Scientific Conference on Informatics, pp. 18–25, studeni 2019.

[19] B. Xue, M. Zhang, W. N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Transactions on Evolutionary Computation, No. 4, Vol. 20, pp. 606-626, kolovoz 2016.

[20] UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets.php [14.8.2022.]

[21] I. Paryudi, What Affects K Value Selection In K-Nearest Neighbor, International Journal of Scientific & Technology Research, No. 7, Vol. 8, pp. 86-92, srpanj 2019.

[22] B. Zorić, D. Bajer, M. Dudjak, Wrapper-based feature selection via differential evolution: benchmarking different discretisation techniques, 2020 International Conference on Smart Systems and Technologies, pp. 89–96, listopad 2020.

SAŽETAK

U radu je opisan problem klasifikacije i odabira značajki (FS) te algoritam diferencijalne evolucije (DE), popularan evolucijski algoritam koji može biti korišten kao omotač za FS. Opisane su tri tehnike diskretizacije rješenja u DE korištenog za FS te oblici pristupa odabira vrijednosti parametara DE. Dan je opis ostvarenog programskog rješenja koje omogućuje izvođenje algoritma DE u svrhu FS s mogućnošću podešavanja parametara DE i korištene tehnike diskretizacije rješenja. Korištenjem ostvarenog programskog rješenja provedena je eksperimentalna analiza na nekoliko često korištenih skupova podataka kojom su uspoređivane tehnike diskretizacije rješenja i pristupi odabira vrijednosti parametara DE u smislu performansi klasifikacije i redukcije broja značajki. Rezultati prikazuju važnost odabira tehnike diskretizacije rješenja, ali i gotovo nepostojeći utjecaj pristupa odabira vrijednosti parametara DE na performanse algoritma.

Ključne riječi: diferencijalna evolucija, klasifikacija, odabir vrijednosti parametara, odabir značajki, tehnike diskretizacije rješenja

ABSTRACT

A differential evolution algorithm for feature selection

The paper describes classification and feature selection (FS), as well as the differential evolution algorithm (DE), a popular evolutionary algorithm that can be used as a wrapper for FS. Three solution discretization techniques used in DE for FS are described, as are types of approaches to parameter value selection in DE. A description is given of the implemented software solution which allows for the execution of the DE algorithm used for FS with the possibility of DE parameter adjustment and the selection of a solution discretization technique. By using the implemented software solution, an experimental analysis was carried out on several frequently used data sets, which compared the solution discretization techniques and approaches to DE parameter value selection based on classification performance and feature reduction. The results show the importance of selecting the solution discretization technique, but also the almost non-existent influence of the DE parameter value selection approach on the performance of the algorithm.

Keywords: differential evolution, classification, parameter value selection, feature selection, solution discretization techniques

ŽIVOTOPIS

Filip Znaor rođen je 4. ožujka 1998. godine u Osijeku gdje je pohađao Osnovnu školu Jagode Truhelke. 2013. godine upisuje III. gimnaziju Osijek koju završava 2017. godine. Iste godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek. Tijekom prve godine studija pohađa preddiplomski studij Elektrotehnika i informacijska tehnologija nakon čega prelazi na preddiplomski sveučilišni studij Računarstvo kojeg završava 2020. godine te upisuje diplomski sveučilišni studij Računarstvo, izborni blok Informacijske i podatkovne znanosti. Predstavljao je Fakultet na natjecanju *IEEEXtreme* u trima navratima te na *Stem Games* 2018. godine, natječući se u *Technology* areni. Tijekom studiranja bio je zaposlen kao demonstrator na kolegijima Digitalna elektronika, Programiranje 1, Programiranje 2 i Objektno orijentirano programiranje. Dobitnik je Dekanove nagrade, Rektorove nagrade i priznanja za uspješnost u studiranju na preddiplomskom i diplomskom studiju na temelju postignutog uspjeha tijekom cjelokupnog studija.