UNIVERZA V MARIBORU
FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO

Lucija Brezočnik

OPTIMIZACIJA Z ROJEM DELCEV ZA IZBIRO ATRIBUTOV PRI KLASIFIKACIJI

Magistrsko delo

Maribor, avgust 2016

OPTIMIZACIJA Z ROJEM DELCEV ZA IZBIRO ATRIBUTOV PRI KLASIFIKACIJI

Magistrsko delo

Študentka:	Lucija Brezočnik
Študijski program:	Študijski program 2. stopnje Informatika in tehnologije komuniciranja
Mentor:	red. prof. dr. Vili Podgorelec

Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko
Smetanova ulica 17
2000 Maribor, Slovenija

Številka: E5022845
Datum in kraj: 24. 05. 2016, Maribor

Na osnovi 330. člena Statuta Univerze v Mariboru (Ur. l. RS, št. 44/2015)
izdajam

SKLEP O MAGISTRSKEM DELU

1. **Luciji Brezočnik**, študentki študijskega programa 2. stopnje INFORMATIKA IN TEHNOLOGIJE KOMUNICIRANJA, se dovoljuje izdelati magistrsko delo.

2. Tema magistrskega dela je pretežno s področja Inštituta za informatiko.

MENTOR: red. prof. dr. Vili Podgorelec

3. Naslov magistrskega dela:
OPTIMIZACIJA Z ROJEM DELCEV ZA IZBIRO ATRIBUTOV PRI KLASIFIKACIJI

4. Naslov magistrskega dela v angleškem jeziku:
PARTICLE SWARM OPTIMIZATION IN FEATURE SELECTION FOR CLASSIFICATION

5. Magistrsko delo je potrebno izdelati skladno z »Navodili za izdelavo magistrskega dela« in ga do 24. 05. 2017 v 2 vezanih in 1 v spiralo vezanem izvodu oddati v pristojni referat za študentske zadeve.

V skladu z Navodili o pripravi in oddaji e-diplom je potrebno magistrsko delo oddati v Digitalno knjižnico Univerze v Mariboru.

Pravni pouk: Zoper ta sklep je možna pritožba na senat članice v roku 15 dni.

Obvestiti:
1. kandidatko
2. mentorja
3. odložiti v arhiv

UNIVERZA V MARIBORU FAKULTETA ZA ELEKTROTEHNIKO, RAČUNALNIŠTVO IN INFORMATIKO 8

Dekan:
red. prof. dr. Borut Žalik

ZAHVALA

Iskreno se zahvaljujem mentorju, red. prof. dr. Viliju Podgorelcu, za strokovno vodenje pri izdelavi magistrske naloge in vzpodbujanje k dodatnemu raziskovalnemu delu.

Asistentu Saši Karakatiču se zahvaljujem za koristne nasvete in pomoč pri statistični analizi dobljenih rezultatov.

Staršem sem hvaležna, da so v meni vzbudili željo po znanju in me podpirali ves čas mojega izobraževanja.

Nazadnje gre hvala drugim domačim in prijateljem, ki so verjeli vame.

OPTIMIZACIJA Z ROJEM DELCEV ZA IZBIRO ATRIBUTOV PRI KLASIFIKACIJI

Ključne besede: računalniška inteligenca, optimizacija z rojem delcev, metoda izbire atributov, klasifikacija

UDK: 004.89(043.2)

Povzetek

V magistrskem delu smo razvili metodo FS-BPSO, ki združuje postopek izbire atributov z algoritmom optimizacije z rojem delcev. Glavni namen te metode je njena uporabnost pri reševanju naslednjega dobro znanega problema. Ko so v podatkovni množici primerki z ogromnim številom atributov, je med njimi težko najti tiste, ki so najbolj informativni oziroma reprezentativni. Tega problema smo se lotili s predlaganim hibridnim algoritmom binarne optimizacije z rojem delcev v kombinaciji s klasifikacijskimi metodami C4.5, Naive Bayes in SVM v cenitveni funkciji za izbiro informativnih atributov. Dobljeni rezultati so statistično analizirani in razkrivajo, da predlagani hibridni algoritem prekaša znane klasifikacijske metode C4.5, Naive Bayes in SVM.

PARTICLE SWARM OPTIMIZATION IN FEATURE SELECTION FOR CLASSIFICATION

Key words: computational intelligence, particle swarm optimization, feature selection, classification

UDK: 004.89(043.2)

Abstract

In this master's thesis, we have developed an FS-BPSO method that joins a feature selection procedure with a particle swarm optimization algorithm. The main purpose of this method is its usability in addressing the following well-known problem: When there are instances with a huge number of attributes in a data set, it is hard to select the most representative ones among them. In order to cope with this problem, we propose the use of a hybrid binary particle swarm optimization algorithm combined with C4.5, Naive Bayes, and SVM as the classifiers in the fitness function for the selection of informative attributes. The results obtained were statistically analysed and revealed that the proposed algorithm outperformed known classifiers, e.g. C4.5, Naive Bayes, and SVM.

Kazalo

Kazalo slik

Kazalo tabel

Seznam uporabljenih kratic

ABC	Artificial Bee Colony	umetna kolonija čebel
ACO	Ant Colony Optimization	optimizacija s kolonijo mravelj
AIS	Artificial Immune System	umetni imunski sistem
ANN	Artificial Neural Networks	umetne nevronske mreže
AUC	Area Under the ROC Curve	ploščina pod krivuljo ROC
BA	Bat Algorithm	algoritem na osnovi obnašanja netopirjev
BFO	Bacterial Foraging Optimization	optimizacija bakterijskega iskanja hrane
BPSO	Binary Particle Swarm Optimization	binarna optimizacija z rojem delcev
CART	Classification And Regression Trees	klasifikacijska in regresijska drevesa
DE	Differential Evolution	diferencialna evolucija
EC	Evolutionary Computing	evolucijsko računanje
EP	Evolutionary Programming	evolucijsko programiranje
ES	Evolution Strategies	evolucijske strategije
FL	Fuzzy Logic	mehka logika
FS	Feature Selection	metoda izbire atributov
GA	Genetic Algorithm	genetski algoritmi
GP	Genetic Programming	genetsko programiranje
LT	Learning Theory	teorija učenja
NB	Naive Bayes	naivni Bayes
NN	Neural Networks	nevronske mreže
PM	Probabilistic Methods	verjetnostne metode
PSO	Particle Swarm Optimization	optimizacija z rojem delcev
ROC	Receiver Operating Characteristic	karakteristika sprejemnika
SVM	Support Vector Machine	klasifikator po metodi podpornih vektorjev
TS	Tabu Search	tabu iskanje

1 Uvod

Količina zbranih podatkov v najrazličnejših bazah nenehno narašča. S porastom podatkov se povečuje tudi obsežnost njihove obdelave in klasifikacije [37, 46]. Nabori podatkov so lahko sestavljeni iz več sto primerkov, ki imajo tudi do več tisoč lastnosti oziroma atributov. Ker je ročno preverjanje in ugotavljanje tega, kateri atributi primerkov so relevantni za določeni proučevani pojav, praktično nemogoče, so raziskovalci začeli uporabljati umetno inteligenco [65]. Umetna inteligenca se deli na tri smeri – tradicionalne statistične metode, tradicionalno umetno inteligenco in računalniško inteligenco. Slednja, relativno mlada smer, obsega med drugim tudi evolucijsko računanje, katerega del je inteligenca rojev. Inteligenca rojev je ena izmed najbolj uporabljanih tehnik za reševanje omenjenih problemov. Predstavniki inteligence rojev so povzeti iz obnašanja kolonije mravelj, jate ptic in rib ter rasti bakterij. Izmed omenjenih vzorcev obnašanja so nastali različni algoritmi, med katerimi sta najbolj znana algoritma optimizacije s kolonijo mravelj (Ant Colony Optimization – ACO) in optimizacija z rojem delcev (Particle Swarm Optimization – PSO).

Optimizacija z rojem delcev je metoda evolucijskega računanja, ki sta jo razvila Kennedy in Eberhart leta 1995 [31] in izhaja iz simuliranja poenostavljenega socialnega obnašanja jate ptic. Pogosto se uporablja pri reševanju optimizacijskih problemov, sčasoma pa se je razvila v metodo naključnega iskanja optimalne rešitve za izbiro atributov (angl. Feature Selection – FS) [14, 41]. Pri PSO torej opazujemo delce (angl. particles), ki vsebujejo vrednosti o svojem položaju in smeri gibanja. Vsi delci so inicializirani z naključnimi začetnimi vrednostmi, nato pa skušajo z lastnim posodabljanjem položaja v vsaki iteraciji oziroma generaciji najti optimalno rešitev. Iteriranje se navadno konča v dveh primerih – če je doseženo največje dovoljeno število iteracij ali če je dosežena želena vrednost funkcije uspešnosti.

V magistrskem delu smo razvili metodo FS-BPSO (angl. Feature Secetion - Binary Particle Swarm Optimization). Gre za hibridni algoritem BPSO (angl. Binary Particle Swarm Optimi-

zation), ki vključuje metodo izbire atributov. Izbira atributov je proces odkrivanja podmnožic pomembnih atributov iz osnovne učne množice [46]. Če se osredotočimo na klasifikacijo, je cilj izbire posameznih atributov odkritje podmnožice pomembnih, reprezentativnih atributov z namenom zmanjšanja velikosti vhodnih podatkov in na podlagi tega tudi pospešitev procesa klasificiranja, medtem ko se splošna uspešnost klasifikacije ohranja ali celo izboljšuje [65].

1.1 Opis problema

Analiza množic podatkov, ki imajo veliko število atributov, ponavadi ne da najboljših rezultatov zaradi morebitnih nepotrebnih ali nepomembnih atributov, ki zgolj povečajo iskalni prostor. Z namenom odprave takih atributov si lahko pomagamo z že omenjeno metodo izbire atributov. Gre za to, da metoda iz množice atributov X izbere podmnožico $A \subseteq X$ na podlagi določene optimizacijske tehnike. Njen namen je tako zmanjševanje dimenzionalnosti podatkov, krajšanje časa računanja in odprava nepomembnih ter nepotrebnih atributov, kar običajno privede do izboljšave klasifikacijske točnosti.

1.2 Cilj in namen magistrskega dela

Cilj magistrskega dela je razvoj metode, ki obravnava omenjeni problem, kako iz množice več tisoč primerkov (alternativ) izbrati podmnožico njihovih informativnih atributov (lastnosti, genov), pri tem pa ohraniti ali celo izboljšati točnost klasifikacije.

Algoritem BPSO bomo kombinirali z različnimi klasifikacijskimi metodami v cenitveni funkciji in preverjali njihovo uspešnost. V ta namen bomo v sklopu magistrskega dela preverjali pravilnost naslednje hipoteze.

Hipoteza: Izbira reprezentativnih atributov v kombinaciji z algoritmom binarne optimizacije z rojem delcev izboljša točnost klasifikacije.

Hipotezo bomo preverjali s pomočjo raziskovalne metode eksperiment, katerega rezultate bomo še statistično obdelali in iskali statistično značilne razlike med hibridnim algoritmom BPSO in izbranimi obstoječimi klasifikacijskimi metodami.

1.3 Napoved vsebine po poglavjih

V drugem poglavju so predstavljene teoretične osnove klasifikacije in klasifikacijskih metod, ki so uporabljene v sklopu eksperimenta. V tretjem poglavju so zbrane mere za ocenjevanje učenja, kjer se osredotočimo predvsem na klasifikacijsko točnost, mero F in ploščino pod krivuljo ROC. Sledi krajše četrto poglavje, ki razloži smisel in navede pet vrst metod izbire atributov iz osnovnega nabora podatkov. Računalniška inteligenca in njena podmnožica, inteligenca rojev, sta opisani v petem poglavju. V njem prikažemo idejo inteligence rojev, njena izhodišča, povzeta iz narave, in različne načine uporabe v praksi.

Šesto poglavje podrobneje predstavi algoritem optimizacije z rojem delcev. Najprej je opisano obnašanje jate ptic v naravi, po katerem se zgleduje tudi sam algoritem. Sledi teoretični prikaz osnovnega algoritma in njegovih komponent ter pregled njegovih kasnejših različic. Binarna različica algoritma optimizacije z rojem delcev je predstavljena v naslednjem, sedmem poglavju. Na začetku je podan pregled relevantnih raziskav, v katerih je bil omenjeni algoritem uporabljen, nato pa so prikazane še podrobnosti algoritma.

Po tako predstavljenih teoretičnih osnovah, ki so potrebne za razumevanje razvite metode, v osmem poglavju opišemo eksperiment. Najprej opišemo načrt eksperimenta, uporabljene podatkovne množice in programsko opremo Weka, ki smo jo uporabili za strojno učenje, ter nastavitve algoritma BPSO. V zaključku poglavja predstavimo še samo izvedbo algoritma in dobljene rezultate.

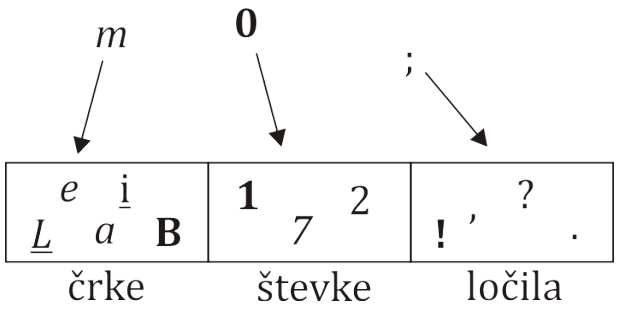
Deveto poglavje je namenjeno statistični obdelavi rezultatov. Rezultati so analizirani glede na uporabljen algoritem v kombinaciji z izbrano klasifikacijsko metodo. V sklepnem, desetem poglavju ovrednotimo razvito metodo optimizacije z rojem delcev za izbiro atributov pri klasifikaciji, podamo omejitve eksperimenta in navedemo priložnosti za nadaljnje delo.

Seznamu uporabljene literature sledi še priloga z zbranimi opisi razredov uporabljenih podatkovnih množic.

2 Klasifikacija

Klasifikacija oziroma uvrščanje primerkov podatkov v razrede je ena izmed nalog strojnega učenja. Mednje prištevamo še regresijo, razvrščanje, učenje asociacij in logičnih relacij ter učenje sistemov (diferencialnih) enačb.

Pri klasifikaciji igra pomembno vlogo klasifikator, katerega namen je določitev razreda posameznemu primerku. Na sliki 2.1 je prikazan primer klasifikacije simbolov, ki vsak simbol uvrsti v enega izmed treh razredov: črke, številke ali ločila.



Slika 2.1: Primer klasifikacije simbolov

Primerek vsebuje množico atributov (lastnosti), ki so lahko neodvisne zvezne ali diskretne spremenljivke. Razred je odvisna spremenljivka z diskretno vrednostjo. Klasifikator pri procesu klasifikacije deluje na podlagi neke vnaprej podane funkcije ali pa uporabi funkcijo, pridobljeno na podlagi prejšnjih rezultatov klasifikacije. Funkcijo klasifikatorja navadno ločimo glede na način njene predstavitve. Med najbolj uporabljene štejemo odločitvena drevesa (angl. decision trees), umetne nevronske mreže (angl. Artificial Neural Networks – ANN), klasifikacijsko metodo naivni Bayes (angl. Naive Bayesian classifier) in klasifikacijsko metodo po metodi podpornih vektorjev (angl. Support Vector Machine – SVM).

Proces klasifikacije navadno vključuje delitev podatkov na učno in testno množico. Omenjena delitev se lahko izvede na različne načine, ki so predstavljeni v podpoglavju 2.1.

V magistrskem delu bomo uporabili klasifikacijske metode C4.5, Naive Bayes in SVM, ki so podrobneje opisane v podpoglavjih 2.2, 2.3 in 2.4.

2.1 Delitev podatkov

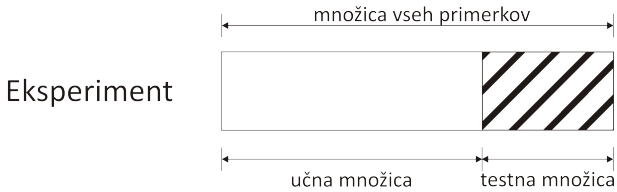
Cilj delitve podatkov je razbitje na učno in testno množico, nekateri avtorji [61] predlagajo še validacijsko (navadno za optimizacijo parametrov). Učna množica podatkov predstavlja vhod za strojno učenje, s testno množico pa vrednotimo rezultate strojnega učenja. Pomembno je, da se testni podatki ne uporabijo pri ustvarjanju klasifikatorja [61]. Težava pa nastane v primeru, ko nimamo na voljo veliko podatkov. S to težavo se spopadamo s pomočjo delitve podatkov.

Delitve podatkov se lahko lotimo na več načinov [3]. Pristopi se razlikujejo glede na število primerkov, ki jih imamo na voljo v osnovni množici, porazdelitev razredov primerkov v osnovni množici, želeno število delov razbitja množice ipd.

Razdelitev (angl. holdout)

Najpreprostejša metoda je razdelitev podatkov. Delitev se izvede tako, da se osnovna množica razbije na dva dela. En del predstavlja učno množico (navadno dve tretjini podatkov), drugi del pa testno (navadno ena tretjina podatkov) [33, 61]. Seveda se lahko zgodi neugodna delitev, kjer množica, namenjena učenju ali testiranju, ni reprezentativna. Načeloma lahko pri preverjanju reprezentativnosti uporabimo preprost trik – preverimo, če je vsak razred iz osnovne podatkovne množice v približno enakih razmerjih prisoten v učni in testni množici. Na sliki 2.2 in tudi na slikah, ki ilustrirajo drugačne delitve, je prikazana razdelitev podatkov, kjer šrafiran del prikazuje testno, nešrafiran pa učno množico. Pri taki naključni delitvi lahko pride do različnih ocen točnosti, saj so ocene odvisne zgolj od tega, kateri primerki so bili vključeni v učno in kateri v testno množico.

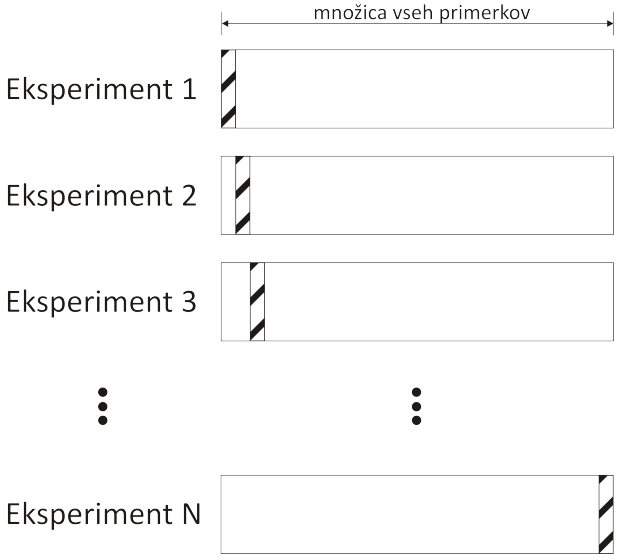
Omejitve te metode ponavadi odpravljamo z uporabo metod prečnega preverjanja in razmnoževanja učnih primerkov.



Slika 2.2: Razdelitev podatkov

Metoda »izloči enega« (angl. leave-one-out ali Jacknife)

Je zelo potratna metoda, saj zahteva izvedbo N eksperimentov nad množico z N primerki [3, 35]. Pri vsakem eksperimentu se uporabi N-1 primerkov za učenje in en preostali primerek za testiranje (slika 2.3). Pri tej metodi je potrebno izpostaviti, da se uspešnost končnega klasifikacijskega modela ocenjuje z uspešnostjo različnih klasifikacijskih modelov. Gradimo N klasifikacijskih modelov za ocenitev uspešnosti in še en dodatni klasifikacijski model nad vsemi učnimi primerki – vsega skupaj N+1 klasifikacijskih modelov. Vendar, ker se je učni algoritem učil iz zelo podobne učne množice (iz nje je vedno odstranjen zgolj en primerek), so razlike med zgrajenimi klasifikacijskimi modeli majhne in imajo posledično tudi podobno uspešnost.



Slika 2.3: Metoda »izloči enega«

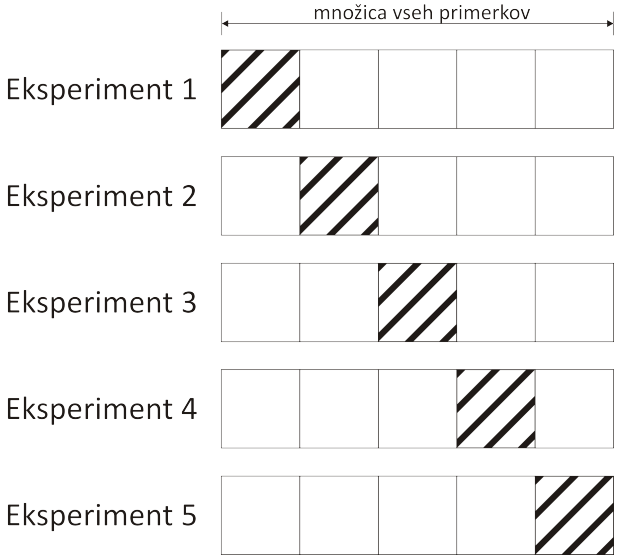
Potratnost metode je najbolj očitna v primeru, ko imamo velik nabor primerkov (velik N), saj moramo namesto enega klasifikacijskega modela zgraditi N+1 klasifikacijskih modelov. V

takih primerih metodo navadno posplošimo na metodo K-kratnega prečnega preverjanja, kjer namesto enega izločamo N/K primerkov.

K-kratno prečno preverjanje (angl. k-fold cross validation)

Ideja metode je ta, da se osnovna množica naključno razbije na K enako velikih delov [3, 35]. V praksi se navadno uporabi $K = 5$ ali $K = 10$, kar predstavlja število klasifikacijskih modelov, ki jih moramo zgraditi.

Na sliki 2.4 je ilustrirano K-kratno prečno preverjanje za $K = 5$. V eksperimentu 1 gradimo klasifikacijski model iz unije množic 2, 3, 4 in 5 ter jo testiramo na množici 1. To ponovimo za vse preostale eksperimente, pri čemer spreminjamo učne množice in testno množico, kot je prikazano na sliki 2.4. Uspešnost končnega klasifikacijskega modela je tako povprečje uspešnosti K klasifikacijskih modelov nad celotno testno množico.



Slika 2.4: K-kratno prečno preverjanje ($K = 5$)

Sorazmerno prečno preverjanje (angl. stratified cross-validation)

Gre za različico metode K-kratnega prečnega preverjanja [3, 35]. Od slednje se razlikuje v tem, da ohranja približno enako distribucijo razredov v učni in testni množici. Če bi izvajali naključno izbiro primerkov iz množice, bi se lahko kaj hitro zgodila nesrečna delitev, ki ne bi zajemala prisotnosti vseh razredov v testni in učni množici. Povedano drugače, če so bili vsi

Slika 2.5: Metoda razmnoževanja učnih primerkov

primerki določenega razreda izvzeti iz učne množice, je težko pričakovati, da bo klasifikator dobro opravil klasifikacijo tega razreda.

Pri vseh omenjenih metodah je potrebno izpostaviti, da je ocena hevristična, saj se testira več klasifikacijskih modelov, ki so bili zgrajeni na manjših učnih množicah, in ne en končni klasifikacijski model.

Metoda razmnoževanja učnih primerkov (angl. bootstrapping)

V primerih, ko je nabor podatkov zelo majhen (30-50 primerkov) [35], se moramo zaradi nezanesljivosti rezultatov metod ocenjevanja uspešnosti zateči k drugim metodam. Leta 1983 sta Diaconis in Efron razvila metodo razmnoževanja učnih primerkov [3, 35]. Postopek metode je prikazan na sliki 2.5.

Naključno se izbere N primerkov s ponavljanjem in se jih uporabi za učno množico. Ponavljanje pomeni, da se lahko isti primerek večkrat pojavi v učni množici. Preostali neizbrani primerki se združijo v testni množici. Število primerkov v testni množici se navadno spreminja od eksperimenta do eksperimenta, kot je predstavljeno na sliki 2.5. Ta postopek se ponovi za vseh K eksperimentov.

2.2 Klasifikacijska metoda C4.5

Klasifikacijsko metodo C4.5 je razvil Ross Quinlan leta 1993 in je naslednik algoritma ID3, ki ga je prav tako izdelal isti avtor. Uvrščamo jo med statistične klasifikacijske metode in je namenjena gradnji odločitvenih dreves. Gradnja drevesa poteka tako, da se najprej izbere atribut, ki najbolje loči osnovno množico podatkov v posamezne podmnožice glede na razred. Delitev se izvede glede na normaliziran informacijski pribitek (angl. information gain), kjer je izbran tisti atribut, ki ima najvišji informacijski pribitek. Postopek se nato nadaljuje na preostalih podseznamih.

Osnovni algoritem klasifikacijske metode C4.5 je predstavljen v psevdokodi algoritma 1 [62].

Algoritem 1 Algoritem C4.5

Vhod: Podatkovna množica D

drevo = $\{\}$
if pogoj_za_zaključek_izpolnjen **then**
    zaključi
end if
for all atribut $a \in D$ **do**
    izračunaj informacijsko-teoretični kriterij, če izvedemo delitev učnih primerkov glede na a
end for
a_{best} = najboljši atribut glede na izračunan kriterij v vrstici 6
drevo = ustvari odločitveno vozlišče, ki testira a_{best} v korenu
D_v = inducirane podmnožice iz nabora D glede na a_{best}
for all D_v **do**
    $drevo_v$ = C4.5(D_v)
    pripni $drevo_v$ na ustrezno vejo drevesa
end for
return drevo

Odločitveno drevo, ki je rezultat klasifikacijske metode C4.5, je ena izmed pogosteje uporabljanih metod pri strojnem učenju. Sestavljeno je iz vozlišč, vej in listov [62]. Vozlišča predstavljajo atribute podatkovne množice, veje vsebujejo vrednosti atributov podatkovne množice, listi pa predstavljajo razred.

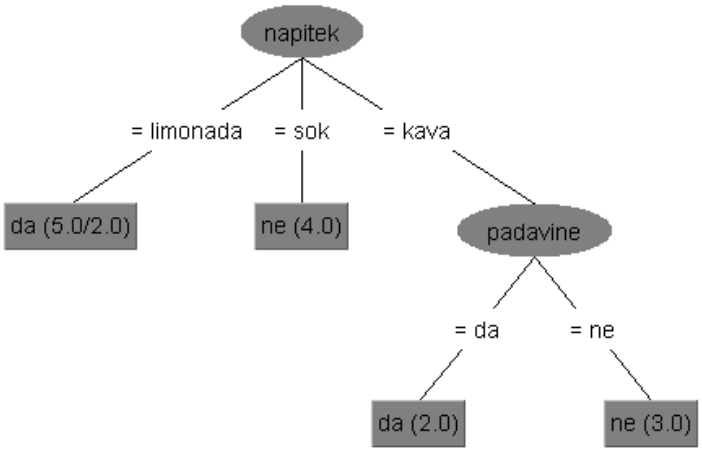
Klasifikacijo s klasifikacijsko metodo C4.5 ilustrirajmo s preprostim primerom. Podatkovna

množica za ta primer je zbrana v tabeli 2.1. Na teh podatkih se s klasifikacijsko metodo C4.5 zgradi odločitveno drevo, predstavljeno na sliki 2.6. Vsaka vrstica tabele predstavlja primerek, ki je opisan z atributi *Napitek*, *Temperatura*, *Padavine* in *Odhod v gostilo?*. Slednji atribut je razred in predstavlja odločitev, ali se na pijačo odpravimo v gostilno ali ne.

Tabela 2.1: Podatkovna množica

	Napitek	**Temperatura**	**Padavine**	**Odhod v gostilno?**
1	limonada	30	ne	da
2	limonada	35	da	da
3	sok	25	ne	ne
4	kava	20	ne	ne
5	kava	24	ne	ne
6	kava	26	da	da
7	sok	28	da	ne
8	limonada	32	ne	da
9	limonada	33	ne	ne
10	kava	22	ne	ne
11	limonada	31	da	ne
12	sok	25	da	ne
13	sok	26	ne	ne
14	kava	23	da	da

Vozlišči v zgrajenem odločitvenem drevesu sta *Napitek* in *Padavine*. Na vejah drevesa so zapisane vrednosti atributov (*limonada*, *sok*, *kava*, *da*, *ne*), razred pa je predstavljen v listih drevesa z vrednostima *da* in *ne*.



Slika 2.6: Rezultat klasifikacije s klasifikacijsko metodo C4.5 (izhod iz programa Weka)

Avtorji članka *Top 10 algorithms in data mining* [62] so klasifikacijsko metodo C4.5 uvrstili na prvo mesto med algoritmi za rudarjenje podatkov.

2.3 Klasifikacijska metoda Naive Bayes

Tudi klasifikacijska metoda Naive Bayes oziroma naivni Bayes je statistična in izračunava pogojne verjetnosti za vsak razred, pri čemer predpostavlja pogojno neodvisnost atributov za podani razred [35].

Bayesova klasifikacijska metoda se izpelje iz Bayesovega pravila, ki je prikazano v enačbi 2.1 [53]. Gre torej za verjetnost, da primerek z vektorjem vrednosti atributov $(x_1, \ldots, x_n)$ pripada razredu r_k. Izračuna se z razmerjem med zmnožkom apriorne verjetnosti razredov r_k in pogojne verjetnosti primerka z atributnim opisom $(x_1, \ldots, x_n)$ pri razredu r_k ter apriorno verjetnostjo primerka z vektorjem vrednosti atributov primerka.

$$P(r_k|x_1, \ldots, x_n) = \frac{P(r_k)\, P(x_1, \ldots, x_n|r_k)}{P(x_1, \ldots, x_n)} \tag{2.1}$$

kjer je:

r_k	k-ti razred podatkovne množice,
$P(r_k)$	apriorna verjetnost razreda r_k,
$(x_1, \ldots, x_n)$	vektor vrednosti atributov primerka,
$P(x_1, \ldots, x_n)$	apriorna verjetnost primerka z atributi $(x_1, \ldots, x_n)$,
$P(x_1, \ldots, x_n\|r_k)$	pogojna verjetnost primerka z atributi $(x_1, \ldots, x_n)$ pri razredu r_k.

Če predpostavimo neodvisnost vrednosti atributov x_i pri danem razredu r_k (vrednosti atributov so glede na vrednost razreda neodvisne), dobimo enačbo 2.2.

$$P(r_k|x_1, \ldots, x_n) = \frac{P(r_k)}{P(x_1, \ldots, x_n)} \prod_{i=1}^{n} P(x_i|r_k) \tag{2.2}$$

Ob ponovni uporabi Bayesovega pravila nad enačbo 2.2 dobimo enačbo 2.3.

$$P(r_k|x_1, \ldots, x_n) = P(r_k)\, \frac{\prod_{i=1}^{n} P(x_i)}{P(x_1, \ldots, x_n)} \prod_{i=1}^{n} \frac{P(r_k|x_i)}{P(r_k)} \tag{2.3}$$

Faktor

$$\frac{\prod_{i=1}^{n} P(x_i)}{P(x_1, \ldots, x_n)}$$

lahko izpustimo, zato je končna enačba klasifikacijske metode prikazana v enačbi 2.4.

$$P(r_k|x_1,\ldots,x_n) = P(r_k)\prod_{i=1}^{n}\frac{P(r_k|x_i)}{P(r_k)} \tag{2.4}$$

Algoritem tako aproksimira pogojne verjetnosti razredov r_k in apriorne verjetnosti razredov $P(r_k)$ za dano verjetnost x_i atributa A_i. Če v podatkovni množici obstaja primer, ki nima znane vrednosti določenega atributa, le-tega ignorira [35].

Čeprav klasifikacijska metoda uporablja naivnost, privzeta je torej pogojna neodvisnost, se v praksi izkaže, da je pogojna neodvisnost pogosto sprejemljiva predpostavka [35]. Navadno daje najboljše rezutate v primerih, kjer je pogojna neodvisnost atributov izpolnjena, na primer pri medicinskih diagnostičnih problemih. Simptomi bolezni bolnika so tako odvisni od diagnoze in so med seboj relativno neodvisni.

Kljub temu obstajajo raziskave, ki dokazujejo, da igra ključno vlogo pri uspešnosti klasifikacijske metode Naive Bayes odvisnostna porazdelitev (angl. dependence distribution) [63].

2.4 Klasifikacijska metoda SVM

SVM (Support Vector Machine) oziroma metoda podpornih vektorjev je še en algoritem za klasifikacijo podatkov. Cilj klasifikacijske metode SVM je iz učnih podatkov generirati model, ki napoveduje ciljne vrednosti testnih podatkov, pri čemer se uporabijo zgolj atributi testnih podatkov [35]. Klasifikacijska metoda iz podatkovne množice vzame vse atribute in jih z linearno kombinacijo uporabi za napovedovanje odvisne spremenljivke [35]. Njena naloga je tako izračunati vrednosti koeficientov diskriminantne funkcije, ki je podana vnaprej in je lahko linearna, kvadratna, polinomska itd. Diskriminantna funkcija tako predstavlja hiperploskev $f(x)$, ki deli dva razreda v prostoru atributov [62].

Metoda podpornih vektorjev uporablja implicitno transformacijo atributov v množico novih atributov s t. i. jedrnimi funkcijami (angl. kernel functions). To transformacijo uporablja zaradi tega, ker mnogi klasifikacijski problemi niso rešljivi z linearno funkcijo. Ravno od izbire ustreznega jedra je odvisna uspešnost klasifikacijske metode SVM [35].

3 Mere za ocenjevanje uspešnosti učenja

Nad testno množico, dobljeno na podlagi razdelitve podatkov (podpoglavje 2.1) podatkovne množice, testiramo hipotezo z namenom ocenitve kvalitete naučenega. Postopek učenja pa se izvaja nad učno množico.

Podpoglavje 3.1 vsebuje predstavitev klasifikacijske točnosti, pojem ploščine pod krivuljo ROC pa je predstavljen v podpoglavju 3.2.

3.1 Klasifikacijska točnost

Vsak primerek v osnovni množici ima enolično določen razred iz končne množice možnih razredov [35].

Klasifikacijsko točnost T predstavimo z enačbo 3.1.

$$T = \frac{N^{(p)}}{N} * 100\% \tag{3.1}$$

kjer je:

$N^{(p)}$ število pravilno rešenih (klasificiranih) primerkov,

N število vseh primerkov.

Takšen T je v realnih primerih praktično nemogoče izračunati, saj bodisi ne poznamo vseh primerkov (N je lahko zelo veliko število) ali pa ne poznamo vseh pravilnih rešitev zanje. Zaradi tega se za določanje klasifikacijske točnosti uporabi testna množica rešenih primerkov n_t. Zaradi tega lahko enačbo 3.1 spremenimo v 3.2.

$$T_t = \frac{n_t^{(p)}}{n_t} * 100\% \tag{3.2}$$

kjer je:

T_t ocena klasifikacijske točnosti nad testnimi primerki,
$n_t^{(p)}$ število pravilno rešenih testnih primerkov,
n_t število vseh testnih primerkov.

Zgornjo mejo ali *optimistično oceno klasifikacijske točnosti* predstavlja klasifikacijska točnost na množici n_u (množica učnih primerkov). Množica učnih primerkov je na voljo algoritmu med učenjem. Enačba zgornje meje klasifikacijske točnosti T_u oziroma klasifikacijske točnosti nad učnimi primerki je 3.3.

$$T_u = \frac{n_u^{(p)}}{n_u} * 100\% \tag{3.3}$$

kjer je:

$n_u^{(p)}$ število pravilno rešenih učnih primerkov,
n_u število vseh učnih primerkov.

Spodnja meja klasifikacijske točnosti se oceni z večinskim razredom. Večinski razred predstavlja tisti razred, ki se v osnovni množici podatkov največkrat pojavi. Kot je bilo že omenjeno, ponavadi ne poznamo vseh primerkov, kot tudi ne poznamo njihove porazdelitve razredov. Zaradi tega si tudi tukaj pomagamo z oceno porazdelitve razredov $n_u^{(i)}$, ki predstavlja število učnih primerkov i-tega razreda. Spodnjo mejo še sprejemljive klasifikacijske točnosti T_v lahko tako izrazimo z enačbo 3.4.

$$T_v = \max_i \frac{n_u^{(i)}}{n_u} \tag{3.4}$$

kjer je:

$n_u^{(i)}$ število učnih primerkov iz i-tega razreda,
n_u število vseh učnih primerkov.

Predstavimo naslednje oznake, ki jih bomo potrebovali za predstavitev problema z dvema razredoma (tabela 3.1) [61].

TP (True Positives) – število pravilno klasificiranih pozitivnih primerkov,

FP (False Positives) – število napačno klasificiranih negativnih primerkov,

TN (True Negatives) – število pravilno klasificiranih negativnih primerkov,

FN (False Negatives) – število napačno klasificiranih pozitivnih primerkov,

PP (Predicted Positives) – število pozitivno klasificiranih primerkov,

PN (Predicted Negatives) – število negativno klasificiranih primerkov,

POS – število pozitivnih primerkov,

NEG – število negativnih primerkov,

n – število vseh primerkov.

V tabeli je s P označen pozitivni razred, z N pa negativni.

Tabela 3.1: Prikaz dvorazrednega problema

		napovedani razred		
		P	N	
pravi razred	P	TP	FN [1]	POS = TP + FN
	N	FP [2]	TN	NEG = FP + TN
		PP = TP + FP	PN = FN + TN	n = TP + FP + PN + TN

[1] napaka druge vrste
[2] napaka prve vrste

Občutljivost (angl. sensitivity) oziroma *priklic* (angl. recall) je definiran(a) kot razmerje med številom pravilno klasificiranih pozitivnih primerkov in številom vseh pozitivnih primerkov. Zapišemo ga z enačbo 3.5.

$$občutljivost = \frac{TP}{TP + FN} = \frac{TP}{POS} \tag{3.5}$$

Občutljivost lahko razumemo kot mero za popolnost klasifikatorja. Nizka vrednost občutljivo-

sti označuje veliko število napačno klasificiranih pozitivnih primerkov.

Natančnost oziroma *preciznost* (angl. precision) je definirana kot razmerje med številom pravilno klasificiranih pozitivnih primerkov in številom vseh pozitivno klasificiranih primerkov. Zapišemo ga z enačbo 3.6.

$$natančnost = \frac{TP}{TP + FP} = \frac{TP}{PP} \tag{3.6}$$

Preciznost lahko razumemo kot mero natančnosti klasifikatorja. Nizka vrednost označuje veliko število napačno klasificiranih negativnih primerkov.

Mera F (angl. F-measure ali F-score) je harmonična sredina občutljivosti in natančnosti. Definirana je z enačbo 3.7.

$$mera\ \ F = \frac{2 * občutljivost * natančnost}{občutljivost + natančnost} = \frac{2 * TP}{2 * TP + FP + FN} \tag{3.7}$$

Z drugimi besedami lahko povemo, da mera F izraža ravnotežje med občutljivostjo in natančnostjo.

Mera F je poseben primer splošne mere F_β, ki v formuli vsebuje tudi faktor β. Če je $\beta = 1$, je mera F_1 enaka meri F.

Zaloga vrednosti vseh treh mer (občutljivost, natančnost in mera F) se nahaja na intervalu $[0, 1]$.

Klasifikacijsko točnost (angl. classification accuracy – ACC) lahko izrazimo z enačbo 3.8.

$$ACC = \frac{TN + TP}{TN + FP + FN + TP} = \frac{TN + TP}{n} \tag{3.8}$$

Je razmerje med pravilno klasificiranimi primerki in vsemi primerki.

3.2 Krivulja ROC

Specifičnost (angl. specificity) je definirana z enačbo 3.9 in podaja razmerje med številom pravilno klasificiranih negativnih primerkov in številom vseh negativnih primerkov.

$$specifičnost = \frac{TN}{TN + FP} = \frac{TN}{NEG} \tag{3.9}$$

Izpad (angl. fall-out) je definiran kot razmerje med številom napačno klasificiranih negativnih primerkov in številom vseh negativnih primerkov (enačba 3.10).

$$izpad = \frac{FP}{FP + TN} = \frac{FP}{NEG} \tag{3.10}$$

Zlahka lahko izpeljemo, da je izpad povezan s specifičnostjo po enačbi 3.11.

$$izpad = 1 - specifičnost \tag{3.11}$$

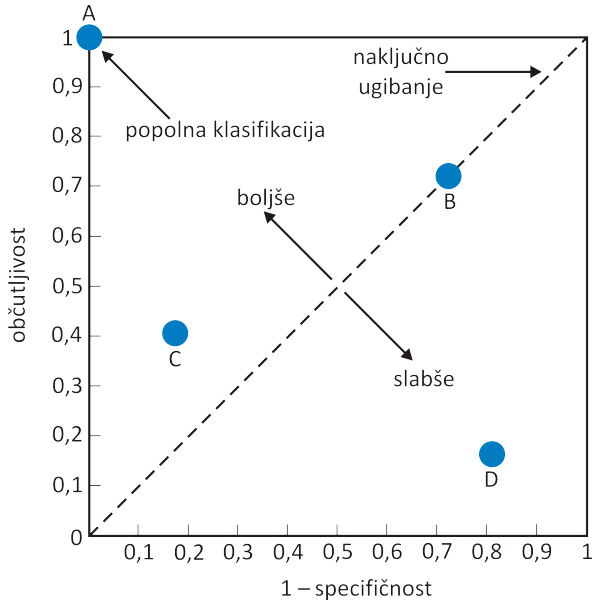
V statistiki je krivulja *ROC* (Receiver operating characteristic) graf, ki ilustrira kakovost klasifikatorja za problem z dvema razredoma. Krivulja predstavlja odvisnost občutljivosti od izpada. Kriuljo ROC včasih imenujemo tudi graf občutljivost proti (1 – specifičnost). Vsak klasificirani primerek predstavlja eno točko v prostoru ROC (slika 3.1).

Najboljša možna klasifikacijska metoda bi dala točko v zgornjem levem kotu (točka A) s koordinatama (0,1), ki predstavlja 100 % občutljivost in 100 % specifičnost. Čisto naključno ugibanje bi dalo točko na diagonali od levega spodnjega do desnega zgornjega kota (na primer točka B). Ta diagonala deli prostor na dva dela. Točke nad diagonalo (na primer točka C) predstavljajo dobre rezultate klasifikacije (boljše od naključnih), točke pod njo (na primer točka D) pa slabe rezultate (slabše od naključnih).

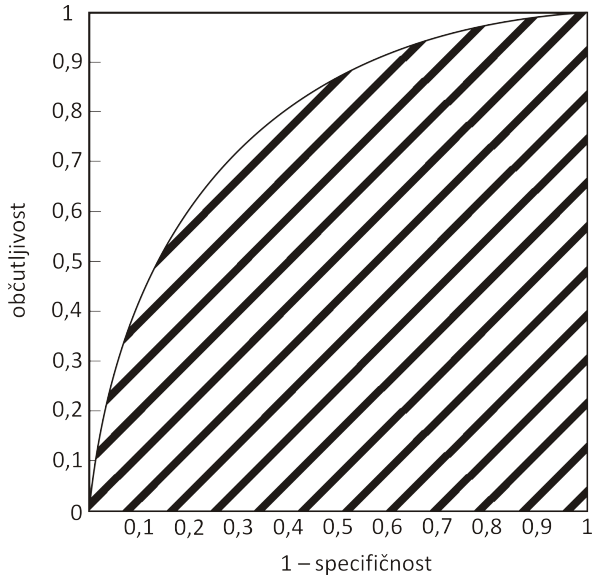
Če zvezno spreminjamo prag[1] odločitvenega pravila, nam točke v prostoru ROC izrišejo celotno krivuljo. Kakovost posameznega klasifikatorja odraža ploščina pod krivuljo ROC,

[1] Pri problemih z dvema razredoma napoved razreda za vsak primerek pogosto temelji na zvezni spremenljivki X, ki je neki »rezultat«, izračunan za dani primerek. Za dani prag T je primerek klasificiran kot pozitiven, če $X > T$, sicer pa kot negativen.

imenovano *AUC* (Area Under the ROC Curve). AUC je na sliki 3.2 prikazan s šrafiranim delom. Če primerjamo med seboj dva klasifikatorja, je boljši tisti, ki ima večjo vrednost AUC.



Slika 3.1: Prostor ROC



Slika 3.2: Ilustracija AUC

4 Izbira atributov

Glavna naloga izbire atributov je, kot pove že ime, izbira podmnožice reprezentativnih atributov, tj. atributov, ki predstavljajo večino relevantnih informacij [27]. Dobro izvedene metode izbire atributov naj bi bile zmožne odkriti in izločiti zavajajoče atribute ter šum v njih. Cilj izbire atributov je torej izboljšanje kakovosti osnovnega nabora podatkov.

V knjigi *Computational Intelligence and Feature Selection* [27] so izpostavili dve lastnosti, ki ju je treba upoštevati v metodah izbire atributov. To sta *pomembnost* oziroma relevanca in *redundanca*. Atribut je pomemben, če ima visoko napovedno vrednost, sicer je nepomemben. Če ima atribut visoko korelacijo z drugimi atributi, pravimo, da je redundanten. O reprezentativnem atributu govorimo tedaj, ko je visoko koreliran s klasifikacijo, vendar je zelo nepovezan z drugimi atributi.

Avtorji v [28] so opredelili dva pojma pomembnosti. Gre za močno in šibko pomembnost. Če je atribut označen kot močno pomemben, bi to v primeru njegove odstranitve iz množice povzročilo poslabšanje točnosti klasifikacije. Če iz množice odstranimo atribut s šibko pomembnostjo, njegova odsotnost v večini primerov ne bo vplivala na točnost, vendar je vse to odvisno od preostalih atributov v množici.

Potrebno je izpostaviti, da lahko pride do slučaja, kjer sta dva atributa sama po sebi nepomembna, vendar imata visoko napovedno vrednost, če ju obravnavamo paroma.

Prednosti metode izbire atributov so [27]:

- bolj zgoščena predstavitev podatkov zaradi zmanjšanega števila atributov,
- zmanjšanje potrebnega prostora za shranjevanje podatkov in njihovo obdelavo,
- skrajšanje časa učenja in
- izboljšanje točnosti klasifikacije.

V splošnem lahko izbiro atributov izvedemo na tri različne načine [40]. Prvi način so metode notranje optimizacije (angl. Wrapper Methods). Pri teh metodah se za optimizacijo uporablja prečno preverjanje brez testnih primerov, kar imenujemo tudi notranje prečno preverjanje. Primer te metode je algoritem rekurzivne odprave atributov.

Drugi način so filtrirne metode (angl. Filter Methods), ki predstavljajo najpreprostejši način izbire atributov. Metoda oceni kakovost vsakega atributa in izbere podmnožico k najboljših atributov. Vrednost k določimo sami ali pa do nje pridemo z izbiro podmnožice atributov na podlagi vnaprej določenega praga. Primera filtrirnih metod sta test hi-kvadrat (angl. Chi-squared test) in informacijski pribitek (angl. Information Gain).

Zadnji, tretji način so vgrajene metode (angl. Embedded Methods). Metode analizirajo, kateri atributi najbolj prispevajo k natančnosti modela, ki je bil zgrajen. V bistvu integrirajo metodo izbire atributov v proces gradnje modela. V to vrsto sodijo metode LASSO, Elastic Net in regresija Ridge.

Metoda izbire atributov se v aplikacijah navadno uporablja na naslednjih področjih [27]:

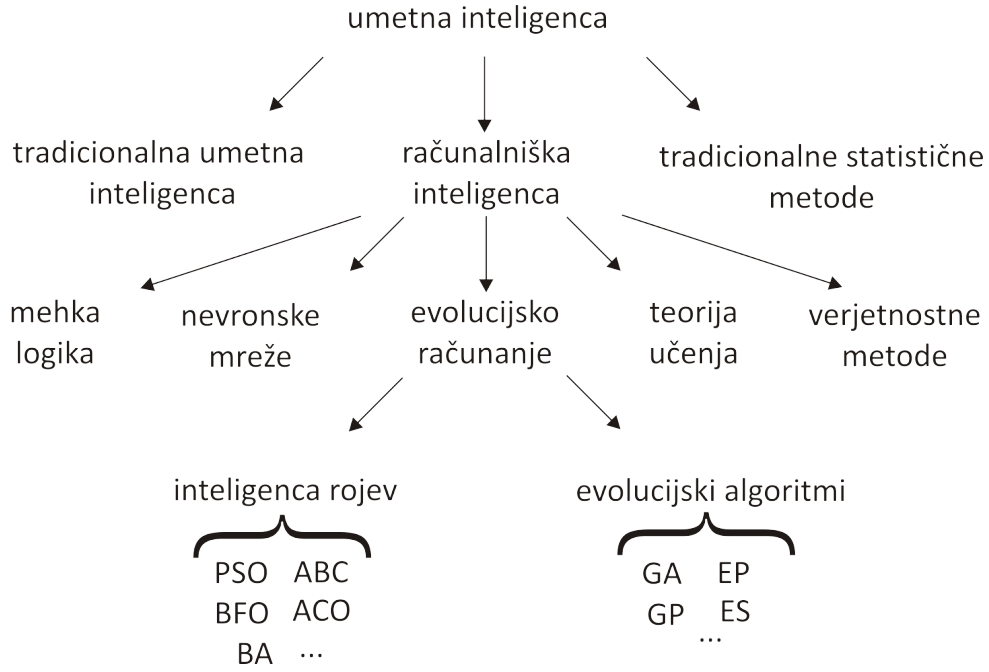
- nadzor sistemov,
- prepoznavanje slik,
- kategorizacija besedil,
- razvrščanje v skupine,
- bio-informatika in
- indukcija pravil.

Ta področja so zanimiva za uporabo izbire atributov predvsem zaradi tega, ker navadno ponujajo velike podatkovne množice za obdelavo.

V magistrskem delu bomo uporabili način metode notranje optimizacije, saj bomo implementirali hibridni algoritem BPSO s klasifikacijskimi metodami v cenitveni funkciji.

5 Računalniška inteligenca

Mnogo ljudi enači umetno in računalniško inteligenco, vendar je slednja v bistvu podmnožica umetne inteligence [5]. Do enačitve verjetno prihaja zaradi tega, ker obe stremita k podobnim ciljem. Kot prikazuje slika 5.1, se umetna inteligenca deli na tri podpodročja. Ta so: tradicionalna umetna inteligenca, računalniška inteligenca in tradicionalne statistične metode.



Slika 5.1: Delitev področja umetne inteligence

Engelbrecht [20, 54] je opredelil, da računalniška inteligenca zajema naslednjih pet pristopov:

- mehka logika (angl. Fuzzy Logic),
- nevronske mreže (angl. Neural Networks),
- evolucijsko računanje (angl. Evolutionary Computing),

- teorija učenja (angl. Learning Theory) in
- verjetnostne metode (angl. Probabilistic Methods).

Evolucijsko računanje zajema predvsem metahevristične optimizacijske tehnike, kot so genetsko programiranje, evolucijski algoritmi ipd. V nadaljevanju se bomo osredotočili na inteligenco rojev (angl. Swarm Intelligence), ki je relativno novo interdisciplinarno podpodročje evolucijskega računanja.

V inteligenco rojev sodijo optimizacija z rojem delcev (PSO – Particle Swarm Optimization), algoritem umetne kolonije čebel (ABC – Artificial Bee Colony), optimizacija bakterijskega iskanja hrane (BFO – Bacterial Foraging Optimization), optimizacija s kolonijami mravelj (ACO – Ant Colony Optimization), algoritem na osnovi obnašanja netopirjev (BA – Bat Algorithm) in podobni pristopi.

5.1 Inteligenca rojev

Izraz inteligenca rojev sta leta 1989 predstavila avtorja Beni in Wang v kontekstu celične robotike [18]. Inteligenca rojev obravnava skupinsko obnašanje, ki izhaja iz medsebojnih lokalnih interakcij posameznih delcev in njihove interakcije z okoljem [7]. Roj tako sestavlja rahlo sklopljena zbirka delcev, ki medsebojno sodelujejo.

Paradigmo inteligence rojev definira pet osnovnih načel, ki jih je že leta 1994 opredelil Milonas [43].

- *Načelo bližine*. Roj mora biti sposoben opravljati enostavne prostorske in časovne izračune.
- *Načelo kakovosti*. Roj mora biti sposoben odziva na dejavnike kakovosti v iskalnem prostoru.
- *Načelo raznolikega odziva*. Roj ne sme izvesti vseh aktivnosti vzdolž preveč ozkih kanalov.
- *Načelo stabilnosti*. Roj ne sme spremeniti svojega obnašanja vsakokrat, ko se spremeni okolje.

- *Načelo prilagodljivosti*. Roj mora biti sposoben spreminjanja svojega obnašanja, ko je to vredno računske cene.

Vidimo, da sta si zadnji načeli v nasprotju.

Inteligenca rojev je zaradi svojih mnogih pozitivnih lastnosti ena izmed najobetavnejših tehnik umetne inteligence s konstantno naraščajočo znanstveno pozornostjo [65]. Njene glavne prednosti so [2]:

- *Razširljivost*. Vsi sistemi inteligence rojev so zelo razširljivi. Nadzorni mehanizmi rojev niso odvisni od velikosti roja, dokler roj ni premajhen [4]. En sam implicitni nadzorni mehanizem lahko upravlja roj s samo nekaj delci ali pa velike roje s tisoči delcev.

- *Prilagodljivost*. Sistemi inteligence rojev se hitro odzivajo na spremembe okolja, pri čemer uporabljajo svoje podedovane zmožnosti samooblikovanja in samoorganizacije [4].

- *Skupinska robustnost*. Sistemi inteligence rojev so zelo robustni, saj delajo skupinsko, brez centralnega nadzora, kar pomeni, da ni nobenega ključnega delca, od katerega bi bilo odvisno delovanje roja. Toleranca pri okvarah je zelo visoka, saj sistem nima kakšne kritične točke okvare. Povedano drugače, okvara enega delca ima zelo majhen vpliv na delovanje sistema [17].

- *Individualna enostavnost*. Sistemi inteligence rojev so sestavljeni iz več enostavnih delcev, ki nimajo veliko zmožnosti sami po sebi, vendar hkrati tako enostavno obnašanje zadostuje za izvajanje prefinjenega obnašanja roja [17].

Tako kot vsak pristop ima tudi inteligenca rojev nekaj omejitev. Te so naslednja [2]:

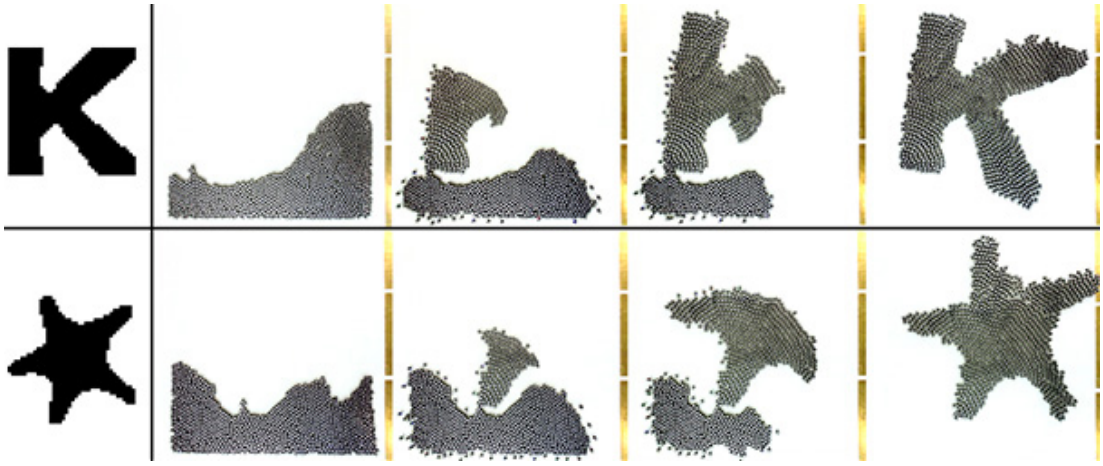
- *Časovno-kritične aplikacije* [4]. Poti do rešitev v sistemih inteligence rojev niso niti vnaprej definirane niti predprogramirane, pač pa se pojavljajo sproti. Zaradi tega inteligenca rojev ni primerna za časovno kritične aplikacije, ki zahtevajo:
  - sprotni nadzor sistemov,
  - časovno kritične odločitve,
  - zadovoljive rešitve v zelo restriktivnih časovnih okvirjih, kot je na primer nadzor

delovanja dvigal ali nadzor temperature v nuklearnem reaktorju.

- *Nastavitve parametrov*. Ena največjih slabosti inteligence rojev je nastavljanje parametrov, tako kot pri vseh stohastičnih optimizacijskih metodah za razliko od deterministčnih optimizacijskih metod, ki teh težav nimajo.

- *Stagnacija*. Ker inteligenca rojev nima centralnega nadzora koordinacije delcev, se lahko zgodi, da sistem preide v stanje stagnacije ali prehitre konvergence v lokalni optimum.

Inteligenca rojev se uporablja za reševanje mnogih problemov v različnih domenah [4]. Njeni principi so bili uspešno vpeljani za reševanje problemov iskanja optimalnih poti, sestavljanje urnikov, analizo slik in podatkov [65] ter celo na področje robotike (npr. projekt univerze Harvard, katerega rezultat so avtonomni 3 cm veliki roboti, ki se znajo urediti v kompleksne oblike (slika 5.2)) [47]. Več primerov njene uporabe je zbranih v podpoglavju 5.3.



Slika 5.2: Avtonomno razvrščanje več kot tisoč robotov v kompleksne oblike [47]

5.2 Primeri inteligence rojev v naravi

Obnašanje skupinske inteligence živali v naravi so pred nekaj desetletji začeli proučevati biologi in drugi naravoslovci ravno zaradi njene izjemne učinkovitosti [13].

Že na koncu 50-ih let prejšnjega stoletja je Pierre-Paul Grassé predstavil pojem *stigmergy*, ki zajema posredno komunikacijo, za katero se zdi, da predstavlja podlago za sodelovanje med socialnimi insekti [13]. Znano namreč je, da imajo insekti, kot so na primer mravlje, čebele in termiti, kompleksne socialne strukture. Gre torej za komunikacijo v obliki znakov ali pokazateljev v okolju s strani enega subjekta, ki vpliva na obnašanje drugih subjektov, če ga srečajo.

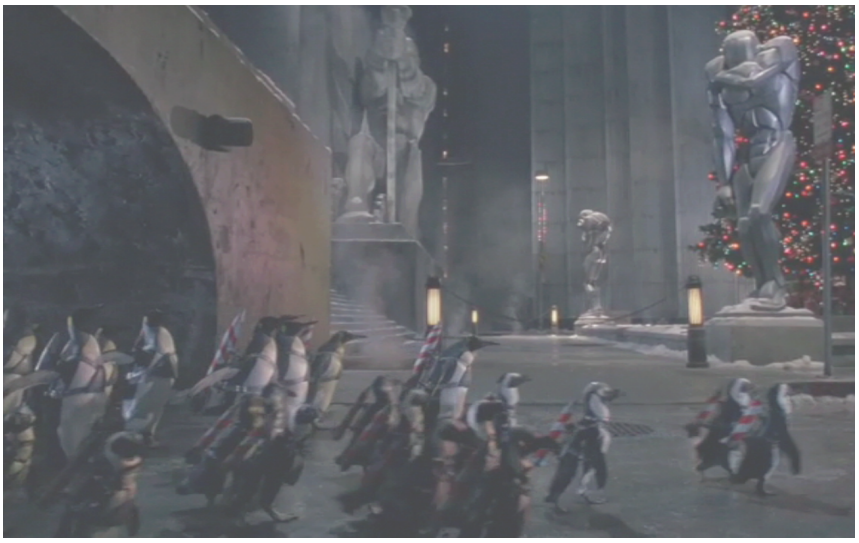
Ravno na podlagi proučevanja obnašanja insektov so se začeli razvijati algoritmi po njihovih vzorih. Obnašanje insektov lahko razvrstimo v nekaj različnih skupin [13]. Naravno gručenje je najbolje vidno pri gručenju trupel mrtvih mravelj v t. i. pokopališčih in urejanju ličink v starostno ločene gruče. Druga skupina je naravna navigacija, kjer je gravitiranje k izvoru hrane odvisno od kemičnega signala – feromona, ki ga v izredno majhnih količinah izločajo posamezne mravlje. Rezultat tretje skupine, naravna gradnja, so gnezda os, panji čebel in termitnjaki (slika 5.3). Pri četrti skupini, iskanje hrane, pa zaznavamo dva različna principa. Prvi je zaznan pri bakterijah, ki se gibljejo proti izvoru hrane s pomočjo kemotakse. Drugi pa je opažen pri čebelah, kjer v primeru, ko čebela najde oddaljen izvor hrane, sporoči njegovo lokacijo, razdaljo in kvaliteto hrane preostalim čebelam v obliki t. i. plesa (gibanja v obliki številke 8 nad izvorom hrane). Nazadnje omenimo še zbiranje živali v jate ali črede, kar je najverjetneje primarna asociacija ljudi ob omembi inteligence rojev.

Slika 5.3: Primer termitnjaka [49]

5.3 Uporaba inteligence rojev v praksi

V nadaljevanju je predstavljena uporaba inteligence rojev na različnih področjih.

Boids [51] je programska oprema, ki je bila razvita leta 1986 in simulira gibanje jate ptic. Leta 1992 so omenjeno programsko opremo uporabili v filmu Tima Burtona z naslovom *Batmanova vrnitev* za simulacijo jate netopirjev in vojske pingvinov. Gre za prvi film, pri katerem je bila uporabljena inteligenca rojev. Prikaz simulacije vojske pingvinov je na sliki 5.4.



Slika 5.4: Vojska pingvinov v filmu *Batmanova vrnitev* [12]

MASSIVE (Multiple Agent Simulation System in Virtual Environment) [42] je programska oprema umetne inteligence za vizualne učinke, povezane z animacijo množic in avtonomno animacijo akterjev. Nastala je z namenom uporabe pri trilogiji filmov *Gospodar prstanov*, saj je Peter Jackson, direktor teh filmov, izrazil željo po programski opremi, ki bi omogočala bojevanje več sto tisoč vojakov. Od takrat dalje se je razvila v celostni produkt in postala vodilna programska oprema za omenjene vizualne učinke. Uporabljajo jo praktično vsi večji filmski studii, kot so Pixar, Sony Pictures Imageworks, ImageMovers Digital, Rhythm & Hues, Digital Domain, Framestore CFC, The Mill idr.

Slika 5.5 prikazuje scene z množico objektov zgolj iz štirih filmov (*Vesele nogice*, *Mumija – Grobnica zmajskega cesarja*, *Vitez teme* in *King Kong*), pri katerih je bila uporabljena programska oprema MASSIVE.

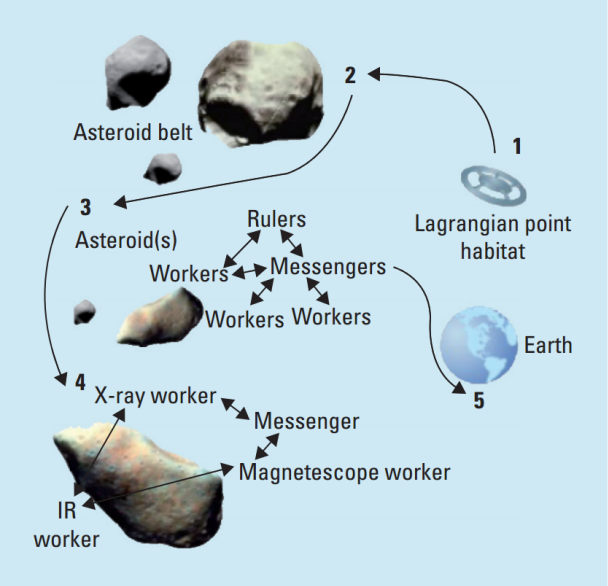
Slika 5.5: Primer uporabe orodja MASSIVE v filmih [42]

Tudi NASA proučuje različne možnosti uporabe inteligence rojev. Članek z naslovom *Swarm Technology at NASA: Building Resilient Systems* [60] zajema prikaz tehnologij, ki jih NASA že uporablja ali pa jih ima namen razvijati. Izpostavljeno je posvečanje razvoju na roju temelječih sistemov vesoljskih plovil, ki obsegajo množico samoorganizirajočih in avtonomnih vesoljskih plovil.

V povezavi s tem je bila predstavljena misija ANTS (Autonomous Nano Technology Swarm), katere cilj je raziskati tehnike umetne inteligence in njene paradigme v prihodnjih raziskovanjih vesolja. Uporabila bo tehnike umetne inteligence tako za vesoljska plovila kot tudi za vozila, namenjena za vožnjo po površini. Misijo ANTS sestavljajo tri konceptne misije. Te so SARA (The Saturn Autonomous Ring Array), PAM (Prospecting Asteroid Mission) in LARA (ANTS Application Lunar Base Activities). Primer misije iskanja rudnin v asteroidnem pasu (PAM), kamor želijo poslati 1000 pico-razrednih vesoljskih plovil, je prikazan na sliki 5.6.

Cilj NASE je v sodelovanju z inštitutom Virginia Institute of Technology izdelati nizkocenovni sistem za planetarno raziskovanje, ki bi deloval samostojno več let v krutih okoljih, kot so na primer atmosferski pogoji Venere ali Titana. Za usmerjanje podatkov po komunikacijskih omrežnih poteh bodo razvili usmerjevalne algoritme, temelječe na inteligenci rojev.

Ameriška vojska proučuje tehnike rojev za usmerjanje vozil brez posadke [57]. Avgusta leta 2014 je organizacija Office of Naval Research predstavila konfiguracijo trinajstih avtonomnih robotskih plovil, povezanih v roj, ki so namenjena varovanju (slika 5.7). Izvedli so akcijo,

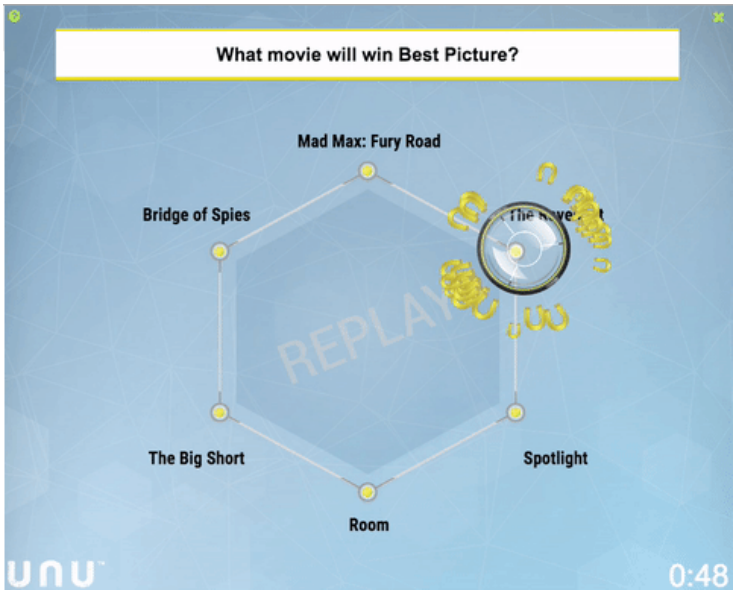


Slika 5.6: Primer uporabe inteligence rojev v NASI [60]

v kateri so robotska plovila varovala veliko ladjo (HVU – High Value Unit). Helikoptrska ekipa, ki je situacijo spremljala, je obvestila robotska plovila, da se neka sumljiva »sovražna« ladja nahaja preblizu HVU-ja. Sumljivo plovilo so roboti zaznali z radarskimi in infrardečimi senzorji in jo obkrožili. Obkrožitev je bila tako natančna, da so lahko s sovražno ladjo stopili v stik in le-ta ni mogla pobegniti. Po obkrožitvi so plovila začela oddajati opozorila prek zvočnikov z utripajočimi lučmi.

Slika 5.7: Avtonomna robotska plovila, ki služijo varovanju [57]

Razvijajo tudi program LOCUST (Low-Cost UAV Swarming Technology), kjer se posvečajo avtonomnemu obnašanju roja dronov [39].



Slika 5.8: Platforma Unu [58]

V začetku maja 2016 je prišla novica, da je podatkovna platforma UNU pravilno napovedala štiri zmagovalne konje na tekmi Kentucky Derby, pri tem pa je bil uporabljen t. i. človeški roj [15]. Platforma UNU [58] omogoča združevanje ljudi v realnem času in reševanje težav po principu umetne inteligence rojev.

Platforma omogoča zastavljanje vprašanj, na katera sodelujoči odgovorijo. Predstavimo primer s slike 5.8. Na zastavljeno vprašanje »Kateri film bo na oskarjih osvojil nagrado za najboljši film?« imamo na voljo šest odgovorov, ki so zapisani v vozliščih. Prvoten položaj t. i. lupe je na sredini šestkotnika, okoli nje pa so magneti, ki predstavljajo ljudi. Vsak človek, ki odgovarja na vprašanje, skuša s svojim magnetom potegniti lupo na izbran odgovor v vozlišču. Pri takem načinu se torej lupa premika po šestkotniku in se na koncu izbere rešitev, ki predstavlja skupinsko odločitev vseh sodelujočih. Za razliko od navadne ankete tukaj ljudje skupaj rešujejo zastavljeni problem in uporabljajo nastajajočo inteligenco.

6 Optimizacija z rojem delcev

Na algoritem PSO sta imeli vpliv predvsem dve metodologiji [31]. Prva je v splošnem umetna inteligenca, znotraj nje pa zlasti teorija rojev, povzeta iz obnašanja jate ptic in rib. Druga metodologija je evolucijsko računanje in njegovi podpodročji, genetski algoritmi in evolucijsko programiranje. PSO ne uporablja križanja ali mutacije, kot ga zahtevajo genetski algoritmi, pač pa omenjeni metodi zamenja z naključnostjo realnih števil in globalno komunikacijo med delci roja.

Pred avtorjema Kennedyjem in Eberhartom so teorijo gibanja jate ptic proučevali tudi drugi avtorji. Leta 1987 se je avtor Reynolds v članku *Flocks, herds and schools: a distributed behavioral model* posvetil predvsem estetiki gibanja jate ptic [51]. Predstavil je program, napisan v Common Lispu, ki simulira gibanje ptic v jati. Tri leta kasneje sta se avtorja Heppner in Grenander v članku *A Stochastic Nonlinear Model for Coordinate Bird Flocks* [25] posvetila odkrivanju osnovnih pravil, ki omogočajo veliki jati ptic, da se giblje sinhrono, pogosto nenadno spremeni smer gibanja, se razkropi in ponovno združi.

Pri razvoju optimizacije z rojem delcev pa je bistveni pomen imela hipoteza sociobiologa E. O. Wilsona, ki je podal naslednjo izjavo [31]: *»Vsaj v teoriji lahko posamezni člani jate napredujejo na podlagi odkritij in dosedanjih izkušenj vseh preostalih članov med iskanjem hrane. Ta prednost lahko postane odločilna in odtehta slabosti tekmovanja za hrano, kadar je le-ta mestoma nepredvidljivo porazdeljena.«* Ta izjava kaže, da socialna izmenjava informacij med posamezniki populacije ponuja evolucijsko prednost.

V naslednjih podpoglavjih je podrobneje predstavljena povezava med razvitim algoritmom in sociološkim vedenjem ptic ter sam algoritem PSO, vključno z njegovimi različicami.

6.1 Jate ptic v naravi

Po eni strani si lahko skupinsko obnašanje razlagamo z vzorci iz narave, kot so gradnja gnezd, iskanje hrane, zbiranje živali v čredo, roj ali jato ipd. Po drugi strani pa skupinsko obnašanje iz inženirskega vidika predstavlja zasnovo porazdeljenih sistemov od spodaj navzgor, kjer se na globalni ravni prikazuje obnašanje medsebojno povezanih enot na lokalni ravni [18].

Glavne značilnosti skupinskega obnašanja, z dodanimi obrazložitvami iz sveta ptic, so [43]:

- *Homogenost*. Vsaka ptica v roju ima enak model obnašanja. Jata se premika brez vodje, čeprav se morda zdi, da se pojavljajo začasni vodje.
- *Vid*. Vid se šteje za najpomembnejši čut organiziranja jate. Ptice imajo zaradi položaja oči sposobnost, da vidijo na obe strani glave hkrati in imajo zaradi tega široko vidno polje.
- *Izogibanje trkom*. Izogibanje trkom z bližnjimi pticami v jati.
- *Usklajevanje hitrosti*. Prizadevanje, da se ptica ujema s hitrostjo bližnjih ptic v jati.
- *Centriranje jate*. Prizadevanje, da ptica ostane blizu sosednjih ptic v jati.

Vidimo, da nekatere značilnosti delujejo kot medsebojna dopolnitev. Na primer, tretja značilnost (izogibanje trkom) se uporablja za določitev minimalne razdalje med pticami v jati, četrta značilnost (usklajevanje hitrosti) pa poskrbi, da se takšna razdalja ohranja med samim letom. Zadnja značilnost (centriranje jate) je bila potrjena na podlagi več študij, ki so izpostavile, da se posamezne živali izogibajo izolaciji [36].

Poleg omenjenih glavnih značilnosti skupinskega obnašana ptic lahko opazimo, da imajo ptice izjemno sposobnost sinhronizacije gibanja v jati in letenja na dolge razdalje. Zaradi njihovega socialnega obnašanja so zmožne [36]:

- letenja brez trčenja, tudi ko nenadoma spremenijo smer leta,
- razpršitve in hitre prerazporeditve pri odzivanju na zunanje grožnje,
- izogibanja plenilcem.

Gibanje ptic v jati sicer izgleda zelo enostavno in tekoče, vendar je jata sestavljena iz diskretne

množice ptic 6.1.

Slika 6.1: Jata ptic [1]

6.2 Algoritem optimizacije z rojem delcev

PSO je metoda evolucijskega računanja, ki sta jo razvila Kennedy in Eberhart leta 1995 [31]. Izhaja iz simuliranja poenostavljenega socialnega obnašanja jate ptic. Pogosto se uporablja pri reševanju optimizacijskih problemov, sčasoma pa se je razvila v metodo naključnega iskanja optimalne rešitve za izbiro atributov [14, 41, 56]. Pri omenjeni metodi opazujemo delce (angl. particles), ki imajo znane vrednosti o hitrosti in smeri svojega gibanja, vrednosti cenitvene funkcije in se gibljejo v raziskovanem prostoru. Vsi delci so inicializirani z naključnimi začetnimi vrednostmi položaja in hitrosti, nato pa skušajo z lastnim posodabljanjem položaja v vsaki iteraciji oziroma generaciji najti optimalno rešitev. Iteriranje se konča v dveh primerih – če je doseženo največje dovoljeno število iteracij ali če je dosežena želena vrednost cenitvene funkcije.

Algoritem PSO deluje tako, da hkrati ohranja več možnih rešitev v iskalnem prostoru. V splošnem lahko rečemo, da je sestavljen iz štirih glavnih delov, ki se ponavljajo tako dolgo, dokler pogoj za zaključek ni izpolnjen. Ti deli so:

- evalvacija cenitvene funkcije vsakega delca,
- posodobitev osebno najboljše vrednosti cenitvene funkcije vsakega delca in njihovega

položaja,

- posodobitev globalno najboljše vrednosti cenitvene funkcije in položaja globalno najboljšega delca ter
- posodobitev hitrosti in položaja vsakega delca.

Poglejmo algoritem PSO podrobneje. Inicializacija delcev v iskalnem prostoru je naključna, število ustvarjenih delcev pa določi uporabnik.

Hitrost delcev se posodablja znotraj vsake iteracije po enačbi 6.1.

$$v_i(t+1) = \omega v_i(t) + c_1 r_1[oNajboljsi_i(t) - x_i(t)] + c_2 r_2[gNajboljsi(t) - x_i(t)] \quad (6.1)$$

kjer je:

i	indeks delca,
t	čas,
$v_i(t)$	hitrost delca i v času t,
$x_i(t)$	položaj delca i v času t,
$oNajboljsi_i(t)$	najboljša osebna rešitev delca i v času t,
$gNajboljsi(t)$	najboljša globalna rešitev roja v času t,
ω	vztrajnostna utež,
c_1	kognitivni koeficient,
c_2	socialni koeficient,
r_1, r_2	naključno generirani števili na intervalu [0, 1].

Enačba 6.1 ima tri člene.

Člen $\omega v_i(t)$ predstavlja komponento vztrajnostne uteži, ki skrbi za to, da se delec giblje v zastavljeni smeri. Vrednost ω torej vpliva na zmanjševanje ali zviševanje hitrosti delca.

Drugi člen predstavlja kognitivno komponento $c_1 r_1[oNajboljsi_i(t) - x_i(t)]$. Lahko si jo predstavljamo kot spomin delca, saj teži k temu, da se delci nagibajo k vrnitvi na položaje v raziskovalnem prostoru, v katerih so imeli visoko vrednost cenitvene funkcije. Komponenta je sestavljena iz omenjenega kognitivnega koeficienta c_1, katerega vrednost se navadno giblje okoli 2. Koeficient vpliva na velikost koraka, ki ga delec opravi proti svoji dosedanji najboljši

vrednosti $oNajboljsi_i(t)$.

Člen $c_2 r_2[gNajboljsi(t) - x_i(t)]$ predstavlja socialno komponento in teži k raziskovanju najboljše regije, ki so jo delci roja do sedaj odkrili. Tudi vrednost socialnega koeficienta c_2 se navadno giblje okoli 2. Koeficient vpliva na velikost koraka, ki ga delec opravi proti globalno najboljši rešitvi $gNajboljsi(t)$, ki jo je do sedaj našel roj delcev.

Pri kognitivni in socialni komponenti sta vpletena koeficienta r_1 in r_2 z definicijskim območjem na intervalu $[0, 1]$ z namenom stohastičnega vpliva na spremembo hitrosti delca.

Po enačbi 6.1 izračunana nova hitrost posameznega delca se pred dokončno potrditvijo preveri. Gre za omejevanje najvišje hitrosti, ki se izvede z enačbo 6.2 v primerih, ko je nova izračunana hitrost $v_i(t + 1)$ izven meja največje dovoljene hitrosti v iskalnem prostoru $[-x_{max}, x_{max}]$.

$$v_{max} = k \times x_{max} \tag{6.2}$$

kjer je:

$k \in [0{,}1,\ 1]$ koeficient omejevanja hitrosti.

Omejevanje hitrosti je torej potrebno zato, da se delci roja ne premaknejo predaleč izven iskalnega prostora. Ker iskalni prostor v večini primerov ni centriran okoli 0, se omejevanje iskalnega prostora zapiše kot $[x_{min}, x_{max}]$, omejevanje hitrosti pa se izvede z enačbo 6.3.

$$v_{max} = k \times \frac{(x_{max} - x_{min})}{2} \tag{6.3}$$

Po tako posodobljeni hitrosti se posodobi tudi položaj delca. To se zgodi po enačbi 6.4.

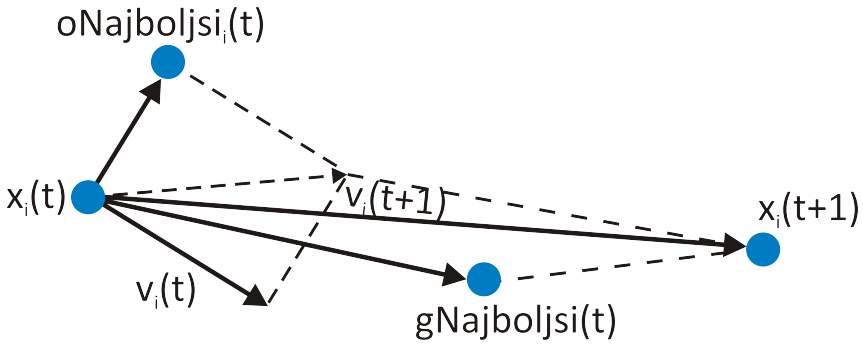
$$x_i(t + 1) = x_i(t) + v_i(t + 1) \tag{6.4}$$

kjer je:

i	indeks delca,
t	čas,
$v_i(t)$	hitrost delca i v času t,
$x_i(t)$	položaj delca i v času t.

Pri tem se vsaka pozicija delca posodobi na podlagi upoštevanja nove hitrosti in prejšnjega položaja delca.

Primer premika delca je prikazan na sliki 6.2.



Slika 6.2: Premik delca *i* pri algoritmu PSO

Omenjeni koraki se ponavljajo v vsaki iteraciji, dokler pogoj za zaključitev ni izpolnjen. Pogoji za zaključitev so lahko različni, največkrat pa je eden izmed naslednjih:

- doseženo največje število iteracij,
- dosežena želena vrednost cenitvene funkcije ali
- doseženo največje število dovoljenih iteracij, v katerih ni prišlo do spremembe globalno najboljšega delca.

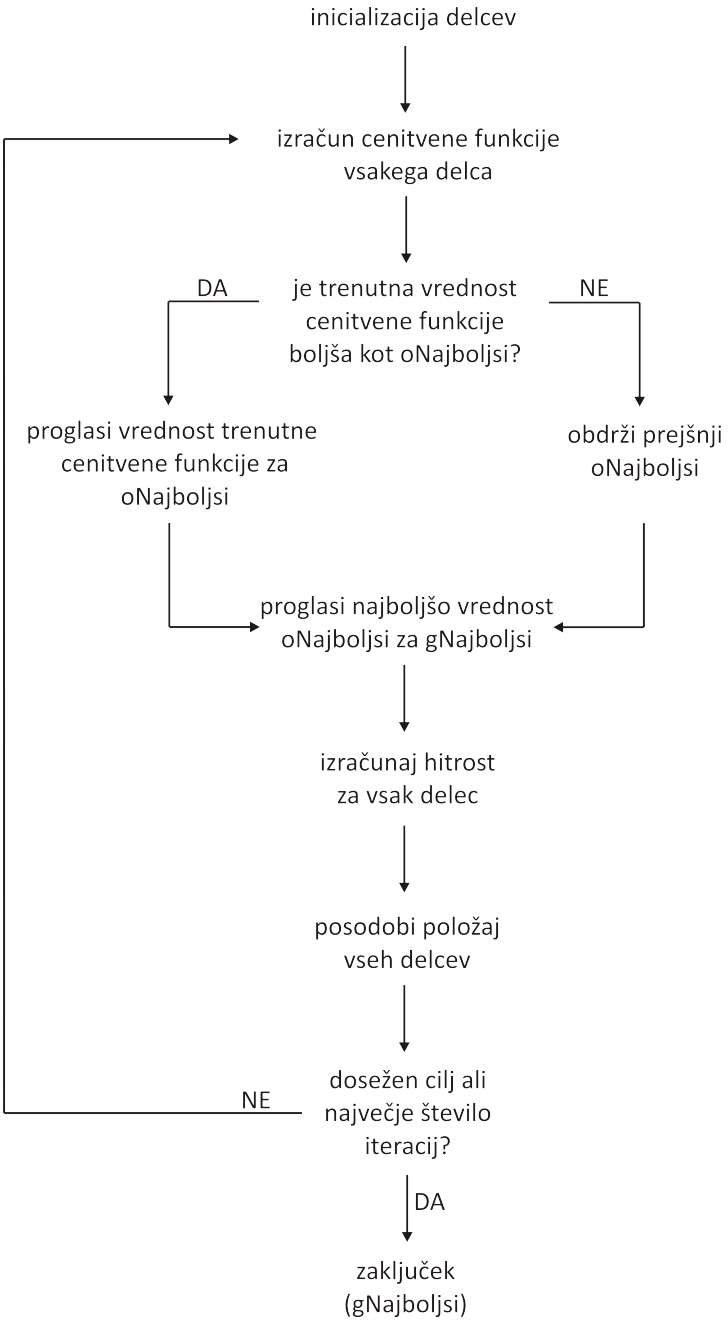
Prikaz splošnega algoritma PSO je prikazan na diagramu na sliki 6.3.

Po predstavljenem algoritmu PSO lahko sedaj na podlagi definiranih načel v podpoglavju 5.1 ugotovimo, da jih algoritem PSO izpolnjuje [65].

- *Načelo bližine*. Delec roja izvede večdimenzionalno preračunavanje prostora v zaporedju časovnih korakov.
- *Načelo kakovosti*. Delec roja se odziva na dejavnike kakovosti lokalno in globalno najboljše vrednosti.
- *Načelo raznolikega odziva*. Odzivi delca roja se gibljejo med lokalno in globalno najboljšo vrednostjo.
- *Načelo stabilnosti*. Delec roja spremeni svoje stanje samo takrat, kadar se spremeni

vrednost globalno najboljšega delca.

- *Načelo prilagodljivosti*. Delec roja spremeni svoje stanje vsakokrat, kadar se spremeni vrednost globalno najboljšega delca.

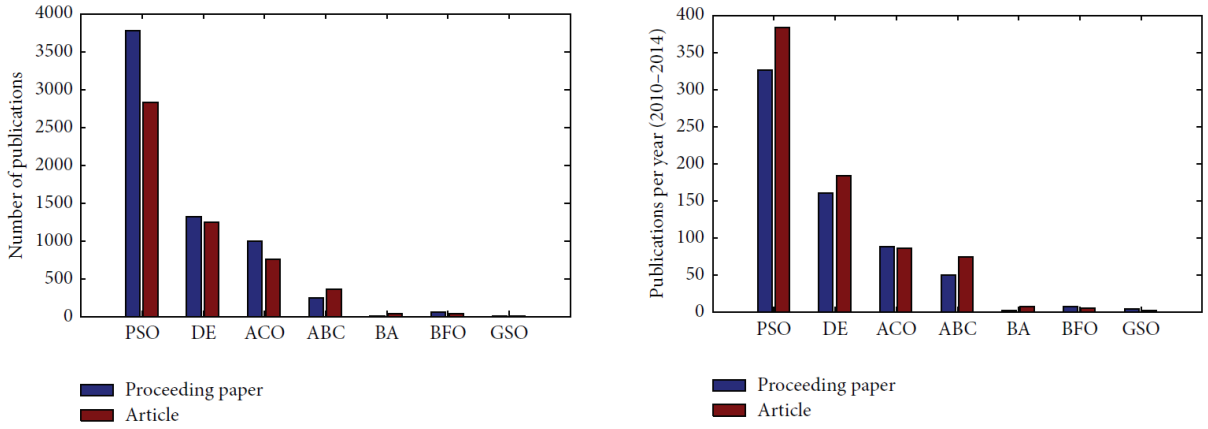


Slika 6.3: Algoritem BPSO

6.3 Različice algoritma optimizacije z rojem delcev

Osnovno različico algoritma PSO, predstavljenega v poglavju 6.2, so različni avtorji preizkušali na mnogih področjih in ga tako spreminjali z namenom optimizacije reševanja specifičnih problemov. Preden predstavimo nekaj različic algoritma PSO, je potrebno izpostaviti, da parameter ”vztrajnostna utež” ni bil prisoten v prvotni različici algoritma, vendar je z leti postal splošno sprejet.

Priljubljenost algoritma PSO so predstavili avtorji Zhang, Wang in Ji, ki so v preglednem članku [65] raziskovali algoritem PSO in njegovo uporabo v različnih aplikacijah. Na sliki 6.4, ki so jo objavili v tem članku, je jasno razvidna priljubljenost algoritma PSO. Levi graf prikazuje število publikacij (ločeno za prispevke na konferencah in članke v revijah), desni pa število publikacij med letoma 2010 in 2014 v odvisnosti od algoritmov, ki temeljijo na inteligenci rojev.



Slika 6.4: Rast popularnosti algoritma PSO glede na število objav na konferencah in v revijah [65]

Aplikacije in študije algoritma PSO v objavah lahko predstavimo s petih vidikov [65]:

- spremembe algoritma PSO (npr. Fuzzy PSO, topologije, bare-bones PSO, izbira atributov v PSO [16, 24, 29, 41, 50, 64]),
- hibridizacija algoritma PSO z drugimi metahevrističnimi metodami (npr. z genetskimi algoritmi [22], tabu iskanjem (angl. Tabu Search – TS) [11], ACO, diferencialno evolucijo (DE – Differential Evolution), umetnim imunskim sistemom (AIS – Artificial

Immune System), nevronskimi mrežami [44]),

- razširitev algoritma PSO na druga področja optimizacije (npr. binarna in diskretna optimizacija [19]),
- teoretična analiza algoritma PSO [30] in
- paralelna implementacija algoritma PSO.

PSO se uporablja na mnogih domenah, kot so medicina, biologija, kemija, mehanika, gradbeništvo in elektronika, energetika, avtomatika, telekomunikacije in druge [65].

7 Binarna optimizacija z rojem delcev

Binarna optimizacija z rojem delcev (BPSO) je binarna različica PSO-ja, ki sta jo leta 1997 predstavila Kennedy in Eberhart [32]. Glavna razlika med PSO in BPSO je ta, da se pri BPSO delci premikajo v binarnem prostoru in lahko njihovi vektorji položaja zavzamejo binarni vrednosti 0 ali 1. Druga razlika je pri interpretaciji pomena hitrosti delcev. Čeprav tudi pri BPSO govorimo o hitrosti, imamo pri tem v mislih pravzaprav verjetnost, da bo položaj delca zavzel vrednost 0 ali 1.

V podpoglavju 7.1 je predstavljen algoritem BPSO, v podpoglavjih 7.2 in 7.3 pa so podrobne informacije o sigmoidalni in cenitveni funkciji, ki se v algoritmu uporabljata. Poglavje sklenemo s pregledom relevantnih raziskav, ki se najbolj navezujejo na zastavljeni problem magistrskega dela.

7.1 Algoritem binarne optimizacije z rojem delcev

BPSO je inicializiran s populacijo delcev. Delci predstavljajo potencialno podmnožico rešitev (tj. informativnih atributov) v n-dimenzionalnem prostoru in se skozi iteracije posodabljajo s sledenjem dvema ekstremoma. Prvi je optimalni položaj delca, ki ga je kdaj dosegel, imenovan tudi individualni ekstrem. Individualni ekstrem i-tega delca je $oNajboljsi_{\mathrm{i}} = (pb_i^1, pb_i^2, \ldots, pb_i^n)$. Drugi ekstrem predstavlja trenutno najboljši delec roja $gNajboljsi = (gb^1, gb^2, \ldots, gb^n)$ in ga zato imenujemo globalni ekstrem. Na gibanje delcev vplivajo vektorji položaja in hitrosti. Za i-ti delec je njegov vektor položaja $X_i = [x_i^1, x_i^2, \ldots, x_i^n]$ in hitrosti $V_i = [v_i^1, v_i^2, \ldots, v_i^n]$, kjer je $x_i^d \in \{0, 1\}$, $d = 1, 2, \ldots, n$ (n predstavlja število atributov) in $i = 1, 2, \ldots, m$ (m predstavlja število delcev).

Na gibanje delcev vplivata koeficienta c_1 in c_2, ki predstavljata pozitivni kognitivni in socialni koeficient, razložena že v podpoglavju 6.2. Če je $c_1 > c_2$, bo iskalno obnašanje težilo k

najboljšim osebnim vrednostim delcev. Če pa je $c_1 < c_2$, bo iskalno obnašanje težilo k najboljšemu do sedaj najdenemu delcu.

Omejitev hitrosti je definirana z v_{min} in v_{max}, parameter ω pa predstavlja vztrajnostno utež, ki zagotavlja ravnovesje med globalnim in lokalnim raziskovanjem. Vsak delec se ovrednoti s cenitveno funkcijo f, ki je dobljena s preračunom točnosti klasifikacije delca. Cenitvena funkcija je podrobneje predstavljena v podpoglavju 7.3.

Psevdokoda hibridnega algoritma BPSO je prikazana na sliki 7.1. Na levi strani slike je zapisan potek algoritma, na desni strani pa so nazorno prikazani glavni elementi algoritma. Čeprav smo že omenili, da je hitrost v BPSO praktično verjetnost, da bo bit spremenil vrednost iz 0 v 1 oziroma iz 1 v 0, smo za lažje razumevanje algoritma v psevdokodi ohranili izraz hitrost.

V magistrskem delu smo kombinirali algoritem BPSO s klasifikacijskimi metodami v cenitveni funkciji. Zaradi lažjega razumevanja omenjene kombinacije, smo le-to vključili v psevdokodo.

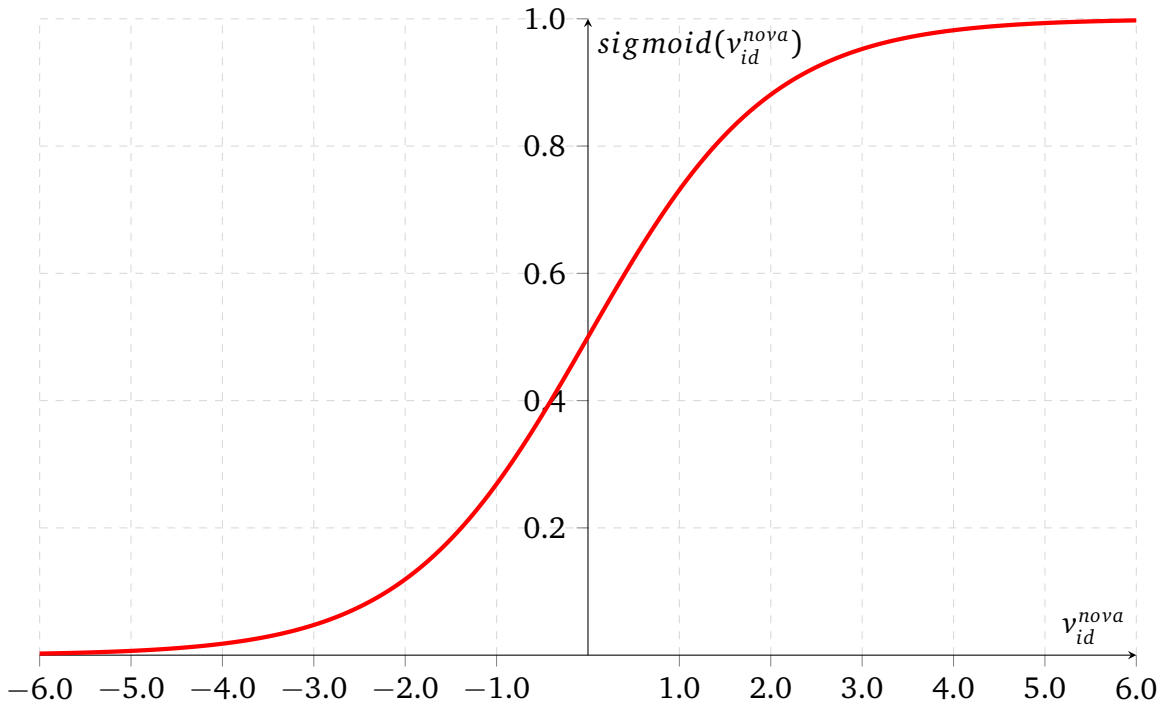
```
Vhodi
  c1, c2,                                                                          ┐
  vmax, vmin,                                                                      │
  STEVILO_ITERACIJ, STEVILO_DELCEV,                                                │
  ω,                                                                               │ nastavitve
  r1, r2                                                                           │ parametrov
                                                                                   │ algoritma
01 začetek                                                                         │
02   Inicializacija populacije                                                     │
03   while STEVILO_ITERACIJ ali konvergenčni kriterij ni izpolnjen do              ┐
04     for i = 1 do števila vseh delcev                                            │ evalvacija cenitvene
05       evalviraj vrednost cenitvene funkcije delca z izbranim klasifikatorjem    ┘ funkcije delca
06       if vrednost cenitvene funkcije delca Xi > vrednost cenitvene funkcije oNajboljsii  ┐
07         then oNajboljsii = Xi                                                   │ posodobi lokalno
08       end if                                                                    ┘ najboljši položaj
09       if vrednost cenitvene funkcije delca Xi > vrednost cenitvene funkcije gNajboljsi  ┐
10         then gNajboljsi = Xi                                                    │ posodobi globalno
11       end if                                                                    ┘ najboljši položaj
12       for d = 1 do števila vseh atributov                                       ┐
13         v_id^nova = ω * v_id^stara + c1r2(oNajboljsi_id^star - x_id^star) + c2r2(gNajboljsi_d^star - x_id^star)  │ posodobi hitrosti
14         if v_id^nova > vmax                                                     │
15           then v_id^nova = vmax                                                 │
16         end if                                                                  │
17         if v_id^nova < vmin                                                     │
18           then v_id^nova = vmin                                                 │
19         end if                                                                  ┘
20         if sigmoid(v_id^nova) > U(0, 1)                                         ┐ posodobi položaje
21           then X_id^novi = 1                                                    │
22           else X_id^novi = 0                                                    │
23         end if                                                                  │
24       next d                                                                    │
25     next i                                                                      ┘
26   end while
27 konec

Izhodi
  gNajboljsi
  vrednost cenitvene funkcije gNajboljsi
```

Slika 7.1: Psevdokoda hibridnega algoritma BPSO

7.2 Sigmoidna funkcija

Algoritem BPSO uporablja za posodabljanje hitrosti (vrstica 13 psevdokode) isto enačbo kot PSO (enačba 6.1), vendar so zdaj vrednosti za x_{id} diskretne in binarne. Če je tako izračunana nova hitrost v_{id}^{nova} manjša (večja) od minimalne (maksimalne) v algoritmu nastavljene hitrosti v_{min} (v_{max}), se nova hitrost postavi na v_{min} (v_{max}).



Slika 7.2: Sigmoidna funkcija

Za posodabljanje položaja delca se posodobljena hitrost v_{id}^{nova} najprej pretvori v vrednost z intervala $[0,\ 1]$ z uporabo sigmoidne funkcije (slika 7.2) [52], definirane z enačbo 7.1.

$$sigmoid({v_{id}}^{nova}) = \frac{1}{1 + e^{-v_{id}^{nova}}} \tag{7.1}$$

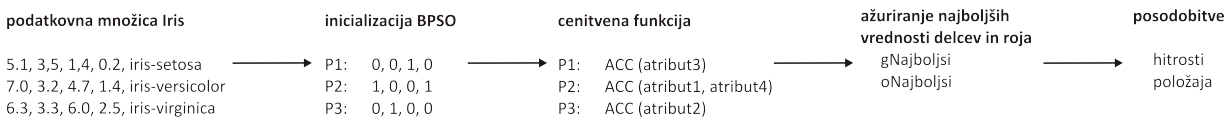
kjer je:

- e osnova naravnega logaritma.

Zdaj se ta vrednost primerja z naključnim številom med 0 in 1 (funkcija $U(0,1)$), ki se generira po enakomerni verjetnostni porazdelitvi (vrstica 20 psevdokode). Odločitev o novem položaju delca x_{id} v vrsticah 21 in 22 je sedaj verjetnostna, kar pomeni, da večja kot je vrednost v_{id}^{nova}, večja je vrednost sigmoidne funkcije, zato je večja verjetnost, da bo za x_{id}^{novi} dodeljena vrednost 1. Če se v_{id}^{nova} povečuje, funkcija $sigmoid(v_{id}^{nova})$ limitira proti 1. Na primer, če je $v_{id}^{nova} > 6$, je verjetnost, da bo $x_{id}^{novi} = 1$, že skoraj enaka 1, ne pa eksaktno enaka 1. Tako za $v_{id}^{nova} = 6$ obstaja verjetnost 0,998, da bo $x_{id}^{novi} = 1$, še vedno pa majhna verjetnost 0,002, da bo $x_{id}^{novi} = 0$.

7.3 Cenitvena funkcija

Cilj cenitvene funkcije je čimbolj objektivna ocenitev sposobnosti posameznika v populaciji [34]. Pri našem hibridnem algoritmu BPSO cenitveno funkcijo predstavlja točnost klasifikacije, pri čemer so ločeno uporabljene klasifikacijske metode C4.5, Naive Bayes in SVM. Cenitvena funkcija igra praktično najpomembnejšo vlogo v samem algoritmu, saj preračunava t. i. kakovost posameznega delca.



Slika 7.3: Shema posplošenega hibridnega algoritma BPSO

Na sliki 7.3 prikažimo shemo poteka hibridnega algoritma BPSO z izpostavljeno cenitveno funkcijo. Delci se inicializirajo z naključnimi vrednostmi 0 ali 1 za atribute in se nato ovrednostijo s cenitveno funkcijo. Vhod v cenitveno funkcijo tako predstavlja delec z izbranimi atributi, to so atributi, ki so označeni z 1. Z izbrano klasifikacijsko metodo se nato zgradi klasifikator, pri čemer se uporabijo atributi, ki so v delcu označeni z 1. Rezultat cenitvene funkcije je točnost klasifikacije posameznega delca, izračunana z izbrano klasifikacijsko metodo. Točnost klasifikacije je ena izmed najbolj uporabljenih metrik za ocenjevanje klasifikacijskih metod [61]. Sledijo morebitna ažuriranja osebnih najboljših vrednosti posameznih delcev in globalno najboljše vrednosti roja. Na koncu se posodobita še hitrost in položaj vsakega delca.

7.4 Pregled relevantnih raziskav

V tem podpoglavju bomo predstavili pet raziskav, ki se še najbolj navezujejo na zastavljeni problem magistrskega dela. Članke smo iskali v bazah podatkov IEEExplorer, ScienceDirect, ACM Digital Library in BioMed Central.

Chen in drugi so v članku [9] izpostavili problem, kako iz množice več tisoč genov izbrati podmnožico informativnih genov, ki lahko nakazujejo na pojav raka. Predstavili so metodo, ki algoritem BPSO kombinira z odločitvenim drevesom v obliki klasifikatorja. Uspešnost predlagane metode so potrdili z znanimi referenčnimi klasifikacijskimi metodami, kot so metoda podpornih vektorjev, samoorganizirajoče karte (angl self-organizing map), nevronske mreže Backpropagation, C4.5, naivni Bayes, klasifikacijska in regresijska drevesa (angl. Classification And Regression Trees – CART) in umetni imunski sistem za razpoznavo (angl. Artificial Immune Recognition System – AIRS). Eksperiment so izvedli nad enajstimi nabori podatkov in dobili nad vsemi nabori podatkov boljše rezultate kot z referenčnimi klasifikacijskimi metodami. Pri tem so uporabili predprocesiranje podatkov s tremi različnimi algoritmi in s tem zmanjšali velikost podatkovnih množic. Nad vsako podatkovno množico so uporabili drugačno število delcev v populaciji. Kot rezultat so podali izbrane gene, ki so bili v procesu izbrani več kot štirikrat, in jih označili kot reprezentativne. V članku je več napak v psevdokodi algoritma in v opisih uporabljenih podatkovnih množic.

Nad enakim naborom podatkov je raziskovalna skupina z istim vodjem in z nekaj manj raziskovalci kot v [9] izvedla praktično enak eksperiment v [10]. Tudi tukaj so uporabili klasifikacijsko metodo C4.5, vendar so uspešnost predlagane metode primerjali s štirimi klasifikacijskimi metodami – SVM, SOM, BPNN in C4.5. Podatke so predprocesirali s tremi različnimi algoritmi z namenom zmanjševanja variance in kot reprezentativne označili tiste atribute, ki so bili izbrani več kot štirikrat. Napake v opisih uporabljenih množic in v psevdokodi algoritma so ostale tudi v tem članku.

Nad tremi nabori medicinskih podatkov o črevesnem in limfnem raku ter levkemiji sta eksperiment izvedla Dara in Banka [16]. Predlagala sta hibridno metodo BPSO, ki uporabi sestavljeno cenitveno funkcijo. Rezultate eksperimenta sta primerjala z algoritmoma GA in NSGA-II na podlagi klasifikacijske metode KNN.

J. Li in drugi [37] so predlagali hibridno metodo BPSO-NB. Pri tej metodi se za kriterijsko funkcijo algoritma BPSO uporabi klasifikacijsko metodo Naive Bayes. Predlagano metodo so primerjali z rezultati klasifikacijskih metod Cfs-BestFirst, NB, C4.5 in KNN nad devetnajstimi nabori različnih podatkov. Podali so zgolj točnost klasifikacije in ne števila izbranih atributov.

Izboljšano metodo BPSO, imenovano EPSO, so predlagali raziskovalci Mohamad in drugi [45]. Uvedli so hibridno cenitveno funkcijo in spremenili potek spreminjanja hitrosti in sigmoide funkcije. Tudi oni so izvedli predprocesiranje in izbrali 500 najvišje rangiranih genov. Nad temi so nato izvedli algoritem BPSO.

Vseh teh eksperimentov ni mogoče neposredno primerjati med seboj, saj uporabljajo različne nabore podatkov in različne nastavitve algoritma BPSO. Vsi eksperimenti tudi nimajo navedenih podatkov glede izvedbe, zato je njihova notranja veljavnost vprašljiva.

8 Eksperiment

Delovanje hibridnega algoritma BPSO, predstavljenega v prejšnjem poglavju, smo preverili s pomočjo eksperimenta.

Eksperiment [6] je raziskovalna metoda, katere primarna naloga je preverjanje vpliva neodvisnih spremenljivk na odvisne spremenljivke. Pri izvedbi eksperimenta je potrebno paziti na nadzor spremenljivk, ki se jih morda pri načrtovanju eksperimenta še niti ne zavedamo in lahko potencialno vplivajo na končni rezultat. Če tega ne storimo, lahko pridemo do napačnih sklepov.

V naslednjih podpoglavjih je predstavljeno načrtovanje eksperimenta in njegova izvedba.

Metoda FS-BPSO obsega razvit hibridni algoritem BPSO, napisan v progamskem jeziku java, kombiniran s klasifikacijskimi metodami znotraj cenitvene funkcije. Pri razvoju smo uporabili še zunanji knjižnici Weka [14] in LibSVM [8]. Knjižnico Weka uporabljamo za izračun točnosti klasifikacije, vrednosti mere F in AUC-ja, LibSVM pa za implementacijo klasifikacijske metode SVM.

8.1 Načrtovanje eksperimenta

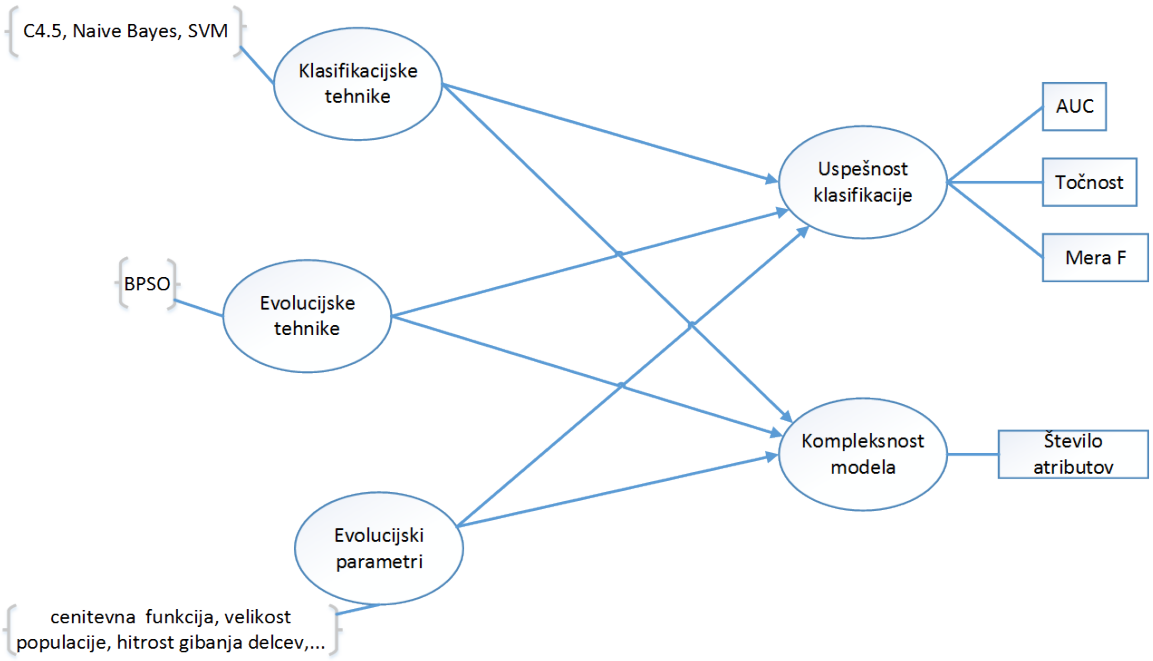
V modelu (slika 8.1) smo definirali pet latentnih spremenljivk, od tega so tri eksogene (neodvisne) in dve endogeni (odvisni). Ker je latentna spremenljivka abstraktna ideja, ki je ni moč neposredno meriti, smo za vsako definirali indikatorje in za vsak indikator še mersko lestvico. Eksogene spremenljivke imajo samo izhodne povezave, endogene pa zgolj vhodne. Vrednosti eksogenih spremenljivk imenujejo stopnje, endogenih pa indikatorji.

Prva latentna eksogena spremenljivka z imenom *Klasifikacijske tehnike* zajema nabor klasifikacijskih metod C4.5, Naive Bayes in SVM, ki bodo uporabljene v razviti metodi za imple-

mentacijo cenitvene funkcije. *Evolucijske tehnike* so druga latentna eksogena spremenljivka. Izmed evolucijskih tehnik bomo v okviru naloge izbrali BPSO. Zadnja eksogena spremenljivka so *Evolucijski parametri*, pri katerih bomo spreminjali cenitveno funkcijo. Pri slednjih bi lahko spreminjali še druge parametre, kot so velikost populacije, hitrost gibanja delcev ipd., kar smo izpostavili v modelu, vendar tega v tem eksperimentu nismo zajeli.

Endogena spremenljivka *Uspešnost klasifikacije* ima indikatorje AUC, mero F in točnost. Vsi trije indikatorji zavzemajo vrednosti na intervalu $[0, 1]$, le da točnost ponavadi izražamo v odstotkih.

Zadnja latentna endogena spremenljivka je *Kompleksnost modela*. Merili jo bomo z diskretno spremenljivko, ki meri število atributov.



Slika 8.1: Teoretični model eksperimenta

8.2 Uporabljene podatkovne množice

V sklopu eksperimenta smo uporabili 11 podatkovnih množic, ki so predstavljene v tabeli 8.1 in urejene po naraščajočem številu atributov primerkov. Pomen stolpcev tabele je naslednji: zaporedna številka, ime, opis in domena množice, število atributov primerkov množice, število razredov množice in število primerkov množice. Kot je razvidno iz četrtega stolpca, so v eksperimentu uporabljene pretežno množice s področja medicine. Temu je botrovalo predvsem dejstvo, da smo potrebovali množice z večjim številom atributov primerkov, kar pa omenjene množice ponujajo. Od vseh podatkovnih množic imata zgolj dve manjkajoče podatke. To sta množici Primary_Tumor in Soybean.

Tabela 8.1: Uporabljeni nabori podatkovnih množic

	Podatkovna množica	**Opis**	**Domena**	**Število**		
				atributov	**razredov**	**primerkov**
1	Primary_Tumor	primarni tumor	Medicina	18	22	339
2	Ionosphere	klasifikacija povratnih informacij radarja iz ionosfere	Fizika	35	2	351
3	Soybean	bolezni soje	Biologija	36	19	683
4	Movement-libras	vrsta giba roke v LIBRAS[1] (brazilski znakovni jezik)	Medicina	91	15	360
5	SRBCT[2]	majhne okrogle modre celice tumorja	Medicina	2309	4	83
6	Leukemia1	akutna mieloblastna levkemija[3], akutna limfoblastna levkemija[4] B-celic, akutna limfoblastna levkemija[3] T-celic	Medicina	5328	3	72
7	DLBCL[5]	difuzni velikocelični limfom B	Medicina	5470	2	77
8	CNS[6]	osrednje živčevje	Medicina	7130	2	60
9	Brain_Tumor2	4 maligne vrste gliom	Medicina	10368	4	50
10	Prostate_Tumor	tumor prostate	Medicina	10510	2	102
11	Leukemia2	AML[3], ALL[4], levkemija mešanega izvora[7]	Medicina	11226	3	72

[1] língua brasileira de sinais
[2] small round blue cell tumor
[3] acute myelogenous leukemia (AML)
[4] acute lympboblastic leukemia (ALL)
[5] diffuse large B-cell lymphomas
[6] central nervous system
[7] mixed-lineage leukemia (MLL)

Vse uporabljene množice so prosto dostopne na internetu. Sneli smo jih z dveh različnih spletnih virov. Večino množic s področja medicine smo pridobili na strani univerze Plymouth

iz Velike Britanije [59]. Drugo uporabljeno spletno mesto pa je UCI Machine Learning Repository [38]. Gre za stran, ki obsega nabor množic, primernih za empirično analizo algoritmov strojnega učenja z namenom klasifikacije, grozdenja in regresije ter jo raziskovalci opisujejo kot primarni vir naborov podatkov za strojno učenje.

Tabela 8.2 prikazuje razporeditev klasificiranih primerkov po razredih. Na primer, primerki v naboru podatkov Leukemia2 se klasificirajo v tri različne razrede: AML, ALL in MLL. Ti imajo v omenjenih razredih 28, 24 in 20 primerkov. Razporeditev klasificiranih primerkov po razredih je pomembna zaradi tega, ker premajhno število klasificiranih primerkov v nekem razredu lahko poslabša učenje in zmanjša kasnejšo točnost klasifikacije. Opis kategorij oziroma razredov, v katere se klasificirajo primerki, je predstavljen v prilogi A.

Tabela 8.2: Razporeditev klasificiranih primerkov po razredih

	Podatkovna množica	**Klasificirani primerki po razredih**
1	Primary_Tumor	84/20/9/14/39/1/14/6/0/2/28/16/7/24/2/1/10/29/6/2/1/24
2	Ionosphere	126/225
3	Soybean	20/20/20/88/44/20/20/92/20/20/20/44/20/91/91/15/14/16/8
4	Movement-libras	24/24/24/24/24/24/24/24/24/24/24/24/24/24/24/
5	SRBCT	29/11/18/25
6	Leukemia1	38/9/25
7	DLBCL	58/19
8	CNS	21/39
9	Brain_Tumor2	14/7/14/15
10	Prostate_Tumor	52/50
11	Leukemia2	28/24/20

Izmed uporabljenih podatkovnih množic ima optimalno razporeditev klasificiranih primerkov po razredih podatkovna množica Movement-libras, ki ima v vsakem razredu enako število primerkov, tj. 24.

Množice, dostopne na strani [59], so na voljo v formatu MAT (format podatkov, zapisanih s pomočjo programa MATLAB). Zaradi tega smo jih morali pretvoriti v želen format ARFF. Format MAT smo shranili kot besedilno datoteko z vrednostmi, ločenimi z vejicami (angl. Comma-Separated Values – CSV) in ga nato s kodo, prikazano na sliki 8.2, pretvorili v format ARFF.

```
1   try {
2        String imeDatoteke = "ImeDatoteke";
3
4        //naloži CSV
5        CSVLoader csvLoader = new CSVLoader();
6        csvLoader.setSource(new File("pot"+imeDatoteke+".csv"));
7        Instances data = csvLoader.getDataSet();
8
9        // shrani ARFF
10       ArffSaver arffSaver = new ArffSaver();
11       arffSaver.setInstances(data);
12       arffSaver.setFile(new File("pot"+imeDatoteke+".arff"));
13       arffSaver.writeBatch();
14
15  } catch (Exception e) {
16       e.printStackTrace();
17  }
```

Slika 8.2: Pretvorba datoteke tipa CSV v datoteko tipa ARFF

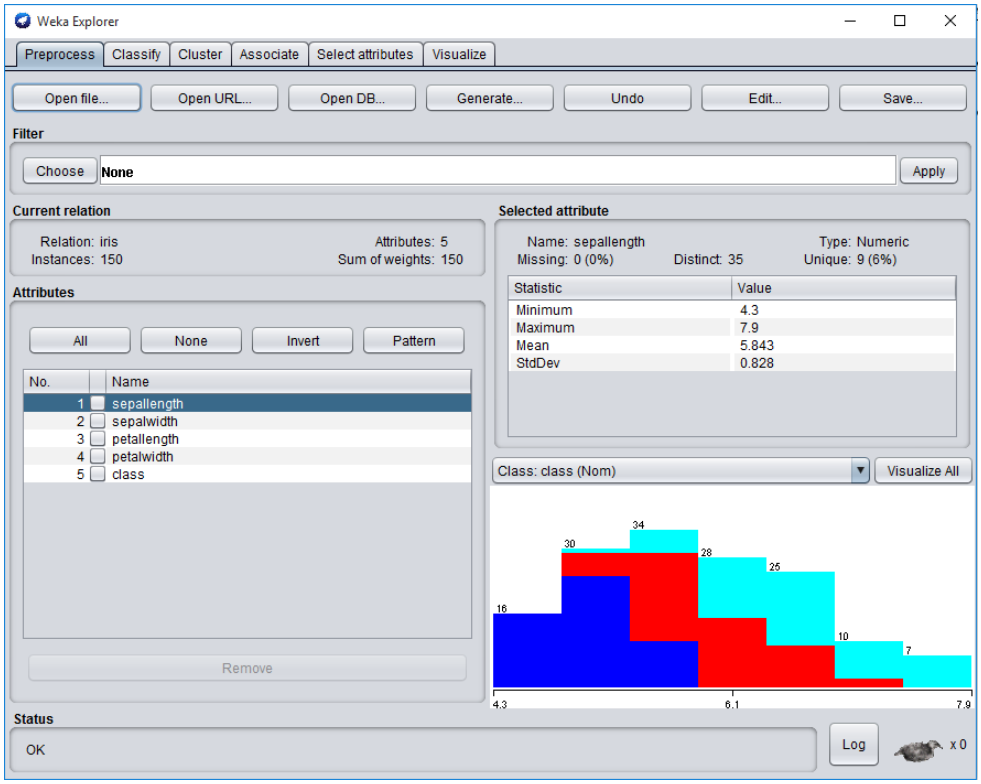
8.3 Programska oprema Weka za strojno učenje

Weka (Waikato Environment for Knowledge Analysis) [23] je odprtokodna programska oprema za podatkovno rudarjenje, napisana v programskem jeziku java, ki so jo razvili na univerzi Waikato na Novi Zelandiji. Logo Weke je ptič weiko, ki je novozelandski endemit (slika 8.3).

Slika 8.3: Logo programske opreme Weka

Izdana je tako namizna aplikacija kot tudi javanska knjižnica, ki omogoča razvijalcem vključitev nabora metod učnih algoritmov v lastne projekte. Prikaz uporabniškega vmesnika Weke je prikazan na sliki 8.4.

ARFF (Attribute-Relation File Format) [61] je tekstovna datoteka, ki vsebuje seznam primerkov. Slika 8.5 prikazuje primer datotečne strukture ARFF, ki jo lahko razdelimo na dva dela – na glavo in na podatke. V glavi so zapisani ime relacije, imena atributov z njihovimi podatkovnimi



Slika 8.4: Uporabniški vmesnik namizne aplikacije Weka z včitano podatkovno množico Iris

tipi in ponavadi v obliki komentarjev še dodatne informacije o množici podatkov, kot so datum oblikovanja, avtor podatkov in razlaga razredov primerkov ter njihovih atributov. Razred množice je v datoteki ARFF predstavljen z zapisom *@ATTRIBUTE class* in je navadno naveden na zadnjem mestu v spisku atributov. V drugem delu so zbrani podatki primerkov. Vsak primerek je zapisan v svoji vrstici, njegove vrednosti atributov pa so med seboj ločene z vejico ali tabulatorjem. Posamezne vrednosti atributov primerkov se morajo ujemati z vrstnim redom definiranih atributov.

Weka podpira naslednje podatkovne tipe: NUMERIC, INTEGER, REAL, STRING, DATE [<oblika-datuma>] in <nominalna-specifikacija>.

Na sliki 8.6 je prikaz vsebine začetnega dela datoteke ARFF, ki vsebuje množico podatkov za cvetlico iris.

```
%
% <komentar>
%
%
%
@RELATION <relation-name>

@ATTRIBUTE <ime-atributa> <podatkovni-tip>
@ATTRIBUTE <ime-atributa> <podatkovni-tip>
@ATTRIBUTE <ime-atributa> <podatkovni-tip>
...
@ATTRIBUTE <ime-atributa> <podatkovni-tip>
@ATTRIBUTE class {<nominalno-ime><nominalno-ime> ... <nominalno-ime>}

@DATA
<vrednost-atributa1>, <vrednost-atributa2>, <vrednost-atributa3>, ..., <vrednost-atributaN>, <nominalno-ime>
<vrednost-atributa1>, <vrednost-atributa2>, <vrednost-atributa3>, ..., <vrednost-atributaN>, <nominalno-ime>
<vrednost-atributa1>, <vrednost-atributa2>, <vrednost-atributa3>, ..., <vrednost-atributaN>, <nominalno-ime>
...
<vrednost-atributa1>, <vrednost-atributa2>, <vrednost-atributa3>, ..., <vrednost-atributaN>, <nominalno-ime>
```

Glava

Podatki

Slika 8.5: Sestava datoteke ARFF

```
%  1. Title: Iris Plants Database
%
%  2. Sources:
%       (a) Creator: R.A. Fisher
%       (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%       (c) Date: July, 1988
%
@RELATION iris

@ATTRIBUTE sepallength NUMERIC
@ATTRIBUTE sepalwidth  NUMERIC
@ATTRIBUTE petallength NUMERIC
@ATTRIBUTE petalwidth  NUMERIC
@ATTRIBUTE class       {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
...
```

Glava

Podatki

Slika 8.6: Primer začetnega dela datoteke ARFF za cvetlico iris

8.4 Nastavitve algoritma BPSO

Nastavitve, ki smo jih uporabili pri eksperimentu, so prikazane v tabeli 8.3.

Tabela 8.3: Nastavitve algoritma BPSO

Spremenljivka	**Vrednost**
STEVILO_DELCEV	200
STEVILO_ITERACIJ	100
ω	0,8
c_1	2
c_2	2
r_1	$[0,1]$
r_2	$[0,1]$
HITROST_MIN	-4
HITROST_MAX	4

Spremenljivka *STEVILO_DELCEV* predstavlja število generiranih delcev v populaciji. Za vsak delec se pri inicializaciji uporabi generirana naključna vrednost, ki se jo pridobi na podlagi generatorja naključnih števil. S tem se vzpostavi začetna populacija delcev, ki se gibljejo v raziskovalnem prostoru.

HITROST_MIN in *HITROST_MAX* sta parametra, ki omejujeta hitrost, oziroma v BPSO verjetnost spremembe bita iz 0 v 1 oziroma iz 1 v 0. Gre torej za verjetnost, da se bo neizbran atribut označil za izbranega ali obratno. Navadno se parametra nastavita na vrednosti $HITROST\_MIN = -4$ in $HITROST\_MAX = 4$ zaradi sigmoidne funkcije, predstavljene v podpoglavju 7.2. V metodi FS-BPSO smo uporabili javansko knjižnico Math in metodo exp(), ki vrne vrednost e^a, pri čem je e osnova naravnega algoritma, a pa predstavlja eksponent. V našem primeru je $a = -hitrost$.

Parametra c_1 in c_2 sta pospeševalna koeficienta. Prvi predstavlja kognitivni pospeševalni koeficient, ki ponazarja moč približevanja delca k njegovi najboljši vrednosti ($oNajboljsi_i$), drugi pa je socialni pospeševalni koeficient, ki ponazarja moč približevanja delcev h globalno najboljšemu delcu (gNajboljsi). Obema koeficientoma smo dodelili vrednost 2, kar pomeni, da ima vsak delec enako moč približevanja tako svoji lokalno najboljši vrednosti kot tudi približevanja globalno najboljši vrednosti roja.

Parameter ω predstavlja vztrajnostno utež. Namenjena je uravnavanju ravnovesja med

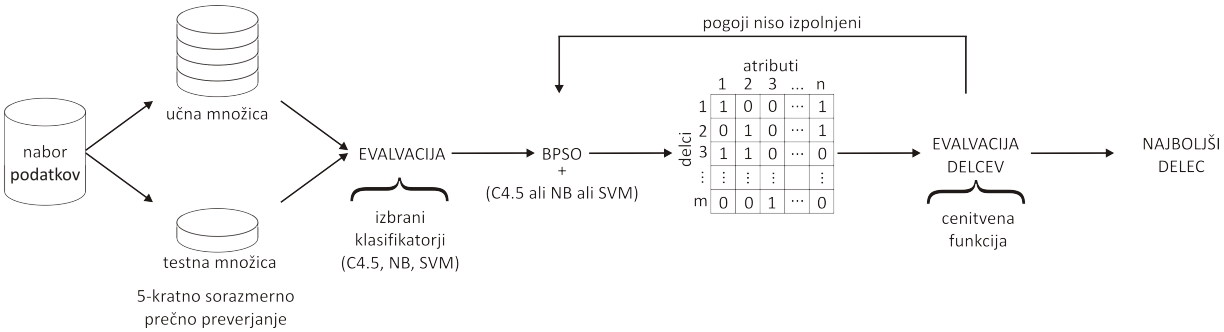
lokalnim in globalnim raziskovanjem. V splošnem nižje vrednosti parametra ω pospešijo konvergenco roja k optimumu, višje vrednosti parametra ω pa spodbujajo raziskovanje celotnega iskalnega prostora. Nekateri avtorji so zaradi tega vpeljali spreminjajočo se vztrajnostno utež, katere vrednost je v začetnih iteracijah višja, s čimer se vzpodbuja globalno raziskovanje raziskovalnega prostora, nato pa z naraščanjem števila iteracij upada, da se s tem izboljša iskanje optimuma.

Parametra r_1 in r_2 predstavljata naključno generirani števili na intervalu $[0, 1]$ za zagotavljanje naključnosti pri iskanju rešitev.

8.5 Izvedba in rezultati eksperimenta

Nad predstavljenimi podatkovnimi množicami v podpoglavju 8.2 smo izvedli eksperiment.

Vsako množico smo razdelili na 5 stratificiranih delov s pomočjo 5-kratnega sorazmernega prečnega preverjanja. Testno množico smo evalvirali z izbranimi klasifikacijskimi metodami C4.5, Naive Bayes in SVM. Dobljene rezultate smo shranili za kasnejšo primerjavo z rezultati hibridnega algoritma BPSO. Po tako shranjenih referenčnih rezultatih se požene algoritem BPSO. Najprej se generirajo delci, ki imajo toliko atributov, kot jih ima uporabljena podatkovna množica. Delci se evalvirajo s cenitveno funkcijo in se na podlagi njene vrednosti in algoritma, opisanega v poglavju 7.1, gibljejo po raziskovalnem prostoru. Rezultat algoritma so najboljši delec in njegova vrednost cenitvene funkcije, mere F in AUC. Shema poteka eksperimenta nad eno podatkovno množico je na sliki 8.7. Po tej shemi se eksperiment ponovi za vsako izmed 11 izbranih podatkovnih množic.



Slika 8.7: Shema poteka eksperimenta nad eno podatkovno množico

Najprej smo za cenitveno funkcijo uporabili točnost klasifikacijske metode C4.5. To pomeni, da smo vseh 10 izbranih podatkovnih množic evalvirali z omenjeno cenitveno funkcijo. Rezultati eksperimenta so predstavljeni v tabeli 8.4. Prvi stolpec vsebuje ime podatkovne množice. Drugi stolpec je številka dela (angl. fold), uporabljenega za testno množico, nato pa sledijo stolpci, ki vsebujejo točnost klasifikacije za klasifikacijske metode C4.5, NB in SVM ter hibridni algoritem BPSO+C4.5.

Za vsako množico smo klasifikacijo zagnali petkrat in v tabelo vnesli povprečno vrednost točnosti posameznega dela. Na primer, pri množici Ionosphere smo v petem delu dobili v petih zaporednih zagonih naslednje točnosti klasifikacije: 97,14, 98,57, 95,71, 95,71 in 98,57. Povprečje vrednosti tako znaša (97,14 + 98,57 + 95,71 + 95,71 + 98,57) / 5 = 97,14.

Krepko so označene najboljše vrednosti posameznega dela, najboljše povprečne vrednosti in najmanjši standardni odklon.

Delni prikaz generiranega izhoda metode FS-BPSO je na sliki 8.8. Pomen stolpcev v generiranem izpisu je naslednji: ime podatkovne množice, zaporedna številka generacije, številka delca, ki je trenutno globalno najboljši, število uporabljenih atributov, točnost klasifikacije, mera F in AUC.

```
Movement-libras; 0; 181;46; 0.7777777777777778; 0.7733632879466212; 0.8878874621988098
Movement-libras; 1; 50;42; 0.8055555555555556; 0.7963464005130672; 0.9063384060091698
Movement-libras; 2; 149;40; 0.8333333333333334; 0.8269270081770084; 0.909955431177446
Movement-libras; 3; 149;40; 0.8333333333333334; 0.8269270081770084; 0.909955431177446
Movement-libras; 4; 149;40; 0.8333333333333334; 0.8269270081770084; 0.909955431177446
Movement-libras; 5; 196;38; 0.8333333333333334; 0.8317169713003048; 0.9081979806848113
Movement-libras; 6; 196;38; 0.8333333333333334; 0.8317169713003048; 0.9081979806848113
Movement-libras; 7; 196;38; 0.8333333333333334; 0.8317169713003048; 0.9081979806848113
Movement-libras; 8; 196;38; 0.8333333333333334; 0.8317169713003048; 0.9081979806848113
Movement-libras; 9; 31;43; 0.8472222222222222; 0.8403198653198652; 0.9164075456053068
Movement-libras; 10; 31;43; 0.8472222222222222; 0.8403198653198652; 0.9164075456053068
Movement-libras; 11; 31;43; 0.8472222222222222; 0.8403198653198652; 0.9164075456053068
Movement-libras; 12; 12;38; 0.8472222222222222; 0.8439574314574314; 0.9308924129353235
Movement-libras; 13; 12;38; 0.8472222222222222; 0.8439574314574314; 0.9308924129353235
Movement-libras; 14; 12;38; 0.8472222222222222; 0.8439574314574314; 0.9308924129353235
Movement-libras; 15; 12;38; 0.8472222222222222; 0.8439574314574314; 0.9308924129353235
Movement-libras; 16; 12;38; 0.8472222222222222; 0.8439574314574314; 0.9308924129353235
Movement-libras; 17; 12;38; 0.8472222222222222; 0.8439574314574314; 0.9308924129353235
Movement-libras; 18; 12;38; 0.8472222222222222; 0.8439574314574314; 0.9308924129353235
Movement-libras; 19; 12;38; 0.8472222222222222; 0.8439574314574314; 0.9308924129353235
Movement-libras; 20; 12;38; 0.8472222222222222; 0.8439574314574314; 0.9308924129353235
Movement-libras; 21; 12;38; 0.8472222222222222; 0.8439574314574314; 0.9308924129353235
Movement-libras; 22; 197;43; 0.8611111111111112; 0.8537708433541767; 0.9388626719344454
Movement-libras; 23; 197;43; 0.8611111111111112; 0.8537708433541767; 0.9388626719344454
Movement-libras; 24; 197;43; 0.8611111111111112; 0.8537708433541767; 0.9388626719344454
Movement-libras; 25; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 26; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 27; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 28; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 29; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 30; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 31; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 32; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 33; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 34; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 35; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 36; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 37; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 38; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 39; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 40; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 41; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 42; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 43; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 44; 177;44; 0.8888888888888888; 0.881749037999038; 0.9406307921178421
Movement-libras; 45; 152;43; 0.9166666666666666; 0.9132996632996632; 0.9559585650180469
Movement-libras; 46; 152;43; 0.9166666666666666; 0.9132996632996632; 0.9559585650180469
Movement-libras; 47; 152;43; 0.9166666666666666; 0.9132996632996632; 0.9559585650180469
Movement-libras; 48; 152;43; 0.9166666666666666; 0.9132996632996632; 0.9559585650180469
Movement-libras; 49; 152;43; 0.9166666666666666; 0.9132996632996632; 0.9559585650180469
Movement-libras; 50; 152;43; 0.9166666666666666; 0.9132996632996632; 0.9559585650180469
```

Slika 8.8: Delni prikaz generiranega izhoda metode FS-BPSO

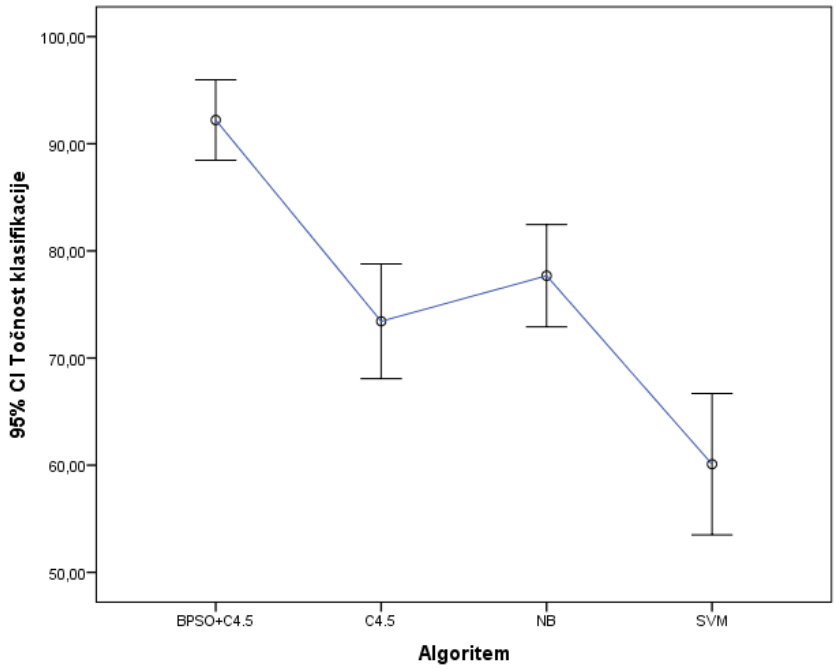
Tabela 8.4: Točnost klasifikacije C4.5, Naive Bayes, SVM in BPSO+C4.5

Podatkovna množica	**Del**	**C4.5** [%]	**Naive Bayes** [%]	**SVM** [%]	**BPSO+C4.5** [%]
Primary_Tumor	1	39,71	51,47	47,06	**52,65**
	2	47,06	45,59	41,18	**52,94**
	3	39,71	51,47	36,76	**52,94**
	4	41,18	48,53	39,71	**50,00**
	5	46,27	50,75	40,30	**53,73**
	$\bar{x}$	42,79	49,56	41,00	**52,45**
	σ	3,60	2,53	3,77	**1,43**
Ionosphere	1	94,37	91,55	98,59	**100**
	2	91,43	84,29	92,86	**97,14**
	3	91,43	78,57	90,00	**96,85**
	4	90,00	84,29	88,57	**97,43**
	5	87,14	74,29	91,43	**97,14**
	$\bar{x}$	90,87	82,60	92,29	**97,71**
	σ	2,62	6,50	3,87	**1,30**
Soybean	1	93,43	93,43	89,05	**97,08**
	2	90,51	94,16	89,05	**96,64**
	3	86,13	88,32	82,48	**94,74**
	4	94,85	93,38	90,44	**97,06**
	5	91,18	94,12	89,71	**97,94**
	$\bar{x}$	91,22	92,68	88,15	**96,69**
	σ	3,33	2,47	3,22	**1,19**
Movement-libras	1	63,89	54,17	25,00	**78,34**
	2	70,83	63,89	37,50	**83,33**
	3	59,72	59,72	33,33	**77,22**
	4	68,06	65,28	33,33	**91,67**
	5	66,67	63,89	16,66	**82,22**
	$\bar{x}$	65,83	61,39	29,16	**82,56**
	σ	**4,24**	4,54	8,34	5,70
SRBCT	1	82,35	**100**	94,12	**100**
	2	82,35	**100**	**100**	**100**
	3	58,82	**100**	**100**	**100**
	4	87,50	93,75	93,75	**100**
	5	87,50	**100**	**100**	**100**
	$\bar{x}$	79,70	98,75	97,57	**100**
	σ	11,96	2,80	3,32	**0**
Leukemia1	1	86,67	93,33	53,33	**100**
	2	**100**	93,33	53,33	**100**
	3	**100**	**100**	57,14	**100**
	4	**100**	92,86	50,00	**100**
	5	85,71	92,86	50,00	**100**
	$\bar{x}$	94,48	94,48	52,60	**100**
	σ	7,57	3,10	2,96	**0**

Tabela 8.4: Točnost klasifikacije C4.5, Naive Bayes, SVM in BPSO+C4.5 (nadaljevanje)

Podatkovna množica	**Del**	**C4.5** [%]	**Naive Bayes** [%]	**SVM** [%]	**BPSO+C4.5** [%]
DLBCL	1	56,25	75,00	75,00	**100**
	2	87,50	75,00	75,00	**97,50**
	3	73,33	**100**	80,00	**100**
	4	66,67	73,33	73,33	**98,67**
	5	80,00	86,67	73,33	**100**
	$\bar{x}$	72,75	82,00	75,33	**99,23**
	σ	12,04	11,39	2,74	**1,13**
CNS	1	33,33	66,67	58,33	**100**
	2	41,67	41,67	66,67	**85,00**
	3	33,33	66,67	66,67	**95,00**
	4	66,67	91,67	66,67	**100**
	5	50,00	58,33	66,67	**100**
	$\bar{x}$	45,00	65,00	65,00	**96,00**
	σ	13,95	18,07	**3,73**	6,52
Brain_Tumor2	1	40,00	80,00	30,00	**100**
	2	40,00	70,00	30,00	**90,00**
	3	60,00	70,00	30,00	**90,00**
	4	70,00	60,00	30,00	**90,00**
	5	70,00	80,00	30,00	**100**
	$\bar{x}$	56,00	72,00	30,00	**94,00**
	σ	15,17	8,37	**0**	5,48
Prostate_Tumor	1	85,71	66,67	52,38	**99,05**
	2	85,71	66,67	52,38	**100**
	3	85,00	60,00	50,00	**97,00**
	4	70,00	70,00	50,00	**100**
	5	75,00	45,00	50,00	**95,00**
	$\bar{x}$	80,28	61,67	50,95	**98,21**
	σ	7,33	10,00	**1,30**	2,17
Leukemia2	1	**100**	**100**	40,00	**100**
	2	86,67	93,33	40,00	**100**
	3	92,86	92,86	35,71	**100**
	4	85,71	92,86	35,71	**100**
	5	78,57	**92,86**	42,86	**92,86**
	$\bar{x}$	88,76	94,38	38,86	**92,86**
	σ	8,07	3,15	**3,10**	3,48

Povprečna točnost klasifikacijskih metod C4.5, Naive Bayes, SVM in algoritma BPSO+C4.5 nad vsemi enajstimi množicami je prikazana na sliki 8.9.



Slika 8.9: Povprečje klasifikacijske točnosti algoritmov BPSO+C4.5, C4.5, NB in SVM na intervalu zaupanja 95 %

Povprečne vrednosti mere F in vrednosti AUC pri vseh enajstih podatkovnih množicah so prikazane v tabeli 8.5.

Tabela 8.5: Povprečne vrednosti mere F in vrednosti AUC pri algoritmu BPSO+C4.5

	Podatkovna množica	**mera F**	**AUC**
1	Primary_Tumor	0,454	0,788
2	Ionosphere	0,980	0,982
3	Soybean	0,969	0,991
4	Movement-libras	0,823	0,917
5	SRBCT	1	1
6	Leukemia1	1	1
7	DLBCL	1	1
8	CNS	0,949	0,938
9	Brain_Tumor2	0,938	0,969
10	Prostate_Tumor	0,990	0,990
11	Leukemia2	0,973	0,981

Rezultati eksperimenta, kjer je za cenitveno funkcijo uporabljena klasifikacijska metoda Naive Bayes, so predstavljeni v tabeli 8.6.

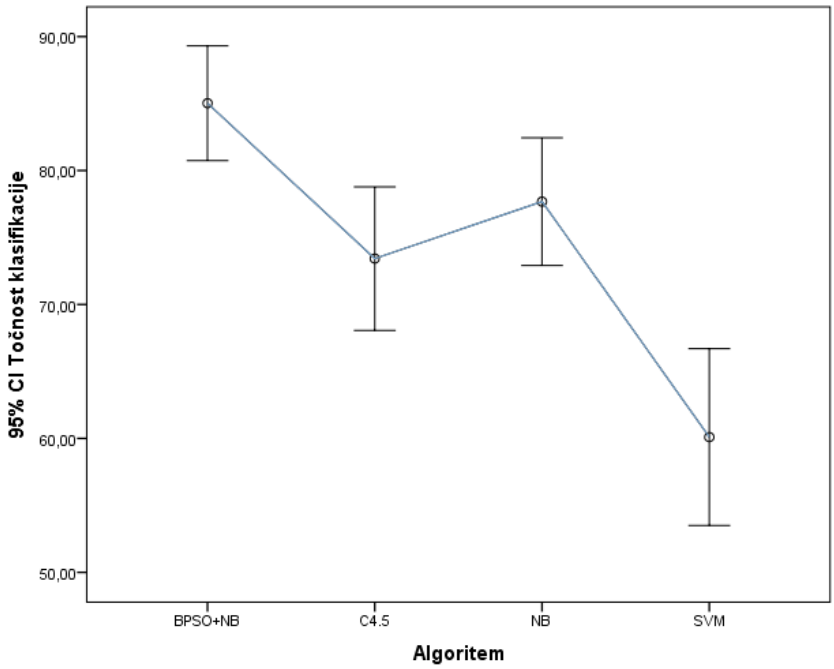
Tabela 8.6: Točnost klasifikacije C4.5, Naive Bayes, SVM in BPSO+NB

Podatkovna množica	**Del**	**C4.5** [%]	**Naive Bayes** [%]	**SVM** [%]	**BPSO+NB** [%]
Primary_Tumor	1	39,71	51,47	47,06	**57,35**
	2	47,06	45,59	41,18	**55,59**
	3	39,71	51,47	36,76	**57,35**
	4	41,18	48,53	39,71	**54,41**
	5	46,27	50,75	40,30	**58,21**
	$\bar{x}$	42,79	49,56	41,00	**56,58**
	σ	3,60	2,53	3,77	**1,54**
Ionosphere	1	94,37	91,55	**98,59**	**98,59**
	2	91,43	84,29	92,86	**96,00**
	3	91,43	78,57	90,00	**94,86**
	4	90,00	84,29	88,57	**95,71**
	5	87,14	74,29	91,43	**95,71**
	$\bar{x}$	90,87	82,60	92,29	**96,17**
	σ	2,62	6,50	3,87	**1,42**
Soybean	1	93,43	93,43	89,05	**96,35**
	2	90,51	94,16	89,05	**96,64**
	3	86,13	88,32	82,48	**92,99**
	4	94,85	93,38	90,44	**96,76**
	5	91,18	94,12	89,71	**95,74**
	$\bar{x}$	91,22	92,68	88,15	**95,70**
	σ	3,33	2,47	3,22	**1,56**
Movement-libras	1	63,89	54,17	25,00	**68,34**
	2	70,83	63,89	37,50	**77,22**
	3	59,72	59,72	33,33	**70,27**
	4	68,06	65,28	33,33	**74,71**
	5	66,67	63,89	16,66	**75,83**
	$\bar{x}$	65,83	61,39	29,16	**73,17**
	σ	4,24	4,54	8,34	**3,75**
SRBCT	1	82,35	**100**	94,12	**100**
	2	82,35	**100**	**100**	**100**
	3	58,82	**100**	**100**	**100**
	4	87,50	93,75	93,75	**100**
	5	87,50	**100**	**100**	**100**
	$\bar{x}$	79,70	98,75	97,57	**100**
	σ	11,96	2,80	3,32	**0**
Leukemia1	1	86,67	93,33	53,33	**100**
	2	**100**	93,33	53,33	97,33
	3	**100**	**100**	57,14	**100**
	4	**100**	92,86	50,00	**100**
	5	85,71	92,86	50,00	**98,57**
	$\bar{x}$	94,48	94,48	52,60	**99,18**
	σ	7,57	3,10	2,96	**1,21**

Tabela 8.6: Točnost klasifikacije C4.5, Naive Bayes, SVM in BPSO+NB (nadaljevanje)

Podatkovna množica	**Del**	**C4.5** [%]	**Naive Bayes** [%]	**SVM** [%]	**BPSO+NB** [%]
DLBCL	1	56,25	75,00	75,00	**93,75**
	2	**87,50**	75,00	75,00	**87,50**
	3	73,33	**100**	80,00	**100**
	4	66,67	73,33	73,33	**86,67**
	5	80,00	86,67	73,33	**93,33**
	$\bar{x}$	72,75	82,00	75,33	**92,25**
	σ	12,04	11,39	**2,74**	5,41
CNS	1	33,33	66,67	58,33	**91,67**
	2	41,67	41,67	**66,67**	**66,67**
	3	33,33	**66,67**	**66,67**	**66,67**
	4	66,67	**91,67**	66,67	**91,67**
	5	50,00	58,33	**66,67**	**66,67**
	$\bar{x}$	45,00	65,00	65,00	**76,67**
	σ	13,95	18,07	**3,73**	6,52
Brain_Tumor2	1	40,00	80,00	30,00	**80,00**
	2	40,00	70,00	30,00	**80,00**
	3	60,00	70,00	30,00	**90,00**
	4	70,00	60,00	30,00	**70,00**
	5	70,00	80,00	30,00	**90,00**
	$\bar{x}$	56,00	72,00	30,00	**82,00**
	σ	15,17	8,37	**0**	8,37
Prostate_Tumor	1	**85,71**	66,67	52,38	71,43
	2	**85,71**	66,67	52,38	66,67
	3	**85,00**	60,00	50,00	65,00
	4	70,00	70,00	50,00	**75,00**
	5	**75,00**	45,00	50,00	45,00
	$\bar{x}$	**80,28**	61,67	50,95	64,62
	σ	7,33	10,00	**1,30**	11,66
Leukemia2	1	**100**	**100**	40,00	**100**
	2	86,67	93,33	40,00	**94,66**
	3	92,86	92,86	35,71	**100**
	4	85,71	92,86	35,71	**100**
	5	78,57	92,86	42,86	**100**
	$\bar{x}$	88,76	94,38	38,86	**98,93**
	σ	8,07	3,15	3,10	**2,39**

Povprečna točnost klasifikacijskih metod C4.5, Naive Bayes, SVM in algoritma BPSO+NB nad vsemi enajstimi množicami je prikazana na sliki 8.10.



Slika 8.10: Povprečje klasifikacijske točnosti algoritmov BPSO+NB, C4.5, NB in SVM na intervalu zaupanja 95 %

Povprečne vrednosti mere F in vrednosti AUC pri vseh enajstih podatkovnih množicah so prikazane v tabeli 8.7.

Tabela 8.7: Povprečne vrednosti mere F in vrednosti AUC pri algoritmu BPSO+NB

	Podatkovna množica	**mera F**	**AUC**
1	Primary_Tumor	0,511	0,842
2	Ionosphere	0,960	0,971
3	Soybean	0,956	0,996
4	Movement-libras	0,732	0,951
5	SRBCT	1	1
6	Leukemia1	0,985	0,986
7	DLBCL	0,919	0,893
8	CNS	0,758	0,683
9	Brain_Tumor2	0,796	0,872
10	Prostate_Tumor	0,617	0,646
11	Leukemia2	0,986	0,989

Rezultati eksperimenta, kjer je za cenitveno funkcijo uporabljen klasifikacijska metoda SVM, so predstavljeni v tabeli 8.8.

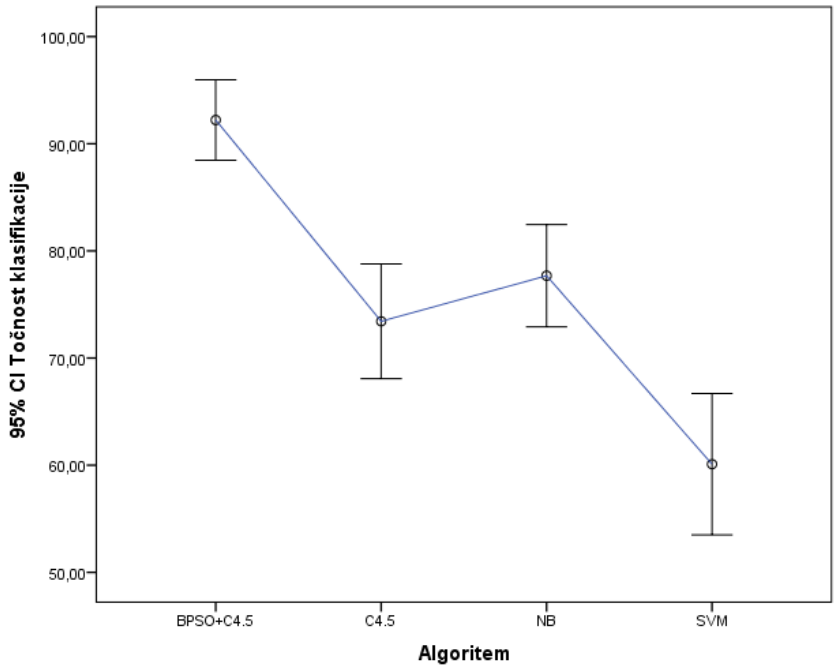
Tabela 8.8: Točnost klasifikacije C4.5, Naive Bayes, SVM in BPSO+SVM

Podatkovna množica	**Del**	**C4.5** [%]	**Naive Bayes** [%]	**SVM** [%]	**BPSO+SVM** [%]
Primary_Tumor	1	39,71	**51,47**	47,06	50,00
	2	47,06	45,59	41,18	**48,53**
	3	39,71	**51,47**	36,76	50,00
	4	41,18	**48,53**	39,71	47,35
	5	46,27	**50,75**	40,30	**50,75**
	$\bar{x}$	42,79	**49,56**	41,00	49,33
	σ	3,60	2,53	3,77	**1,37**
Ionosphere	1	94,37	91,55	**98,59**	**98,59**
	2	91,43	84,29	92,86	**98,00**
	3	91,43	78,57	90,00	**95,71**
	4	90,00	84,29	88,57	**98,57**
	5	87,14	74,29	91,43	**96,85**
	$\bar{x}$	90,87	82,60	92,29	**97,54**
	σ	2,62	6,50	3,87	**1,24**
Soybean	1	93,43	93,43	89,05	**94,89**
	2	90,51	**94,16**	89,05	**94,16**
	3	86,13	88,32	82,48	**93,58**
	4	94,85	93,38	90,44	**94,12**
	5	91,18	**94,12**	89,71	93,53
	$\bar{x}$	91,22	92,68	88,15	**94,06**
	σ	3,33	2,47	3,22	**0,55**
Movement-libras	1	**63,89**	54,17	25,00	48,05
	2	**70,83**	63,89	37,50	48,05
	3	**59,72**	**59,72**	33,33	48,33
	4	**68,06**	65,28	33,33	41,67
	5	**66,67**	63,89	16,66	30,28
	$\bar{x}$	**65,83**	61,39	29,16	43,28
	σ	**4,24**	4,54	8,34	7,79
SRBCT	1	82,35	**100**	94,12	**100**
	2	82,35	**100**	**100**	**100**
	3	58,82	**100**	**100**	**100**
	4	87,50	93,75	93,75	**100**
	5	87,50	**100**	**100**	**100**
	$\bar{x}$	79,70	98,75	97,57	**100**
	σ	11,96	2,80	3,32	**0**
Leukemia1	1	86,67	**93,33**	53,33	53,33
	2	**100**	93,33	53,33	53,33
	3	**100**	**100**	57,14	57,14
	4	**100**	92,86	50,00	50,00
	5	85,71	**92,86**	50,00	50,00
	$\bar{x}$	**94,48**	**94,48**	52,60	52,76
	σ	7,57	3,10	**2,96**	**2,96**

Tabela 8.8: Točnost klasifikacije C4.5, Naive Bayes, SVM in BPSO+SVM (nadaljevanje)

Podatkovna množica	**Del**	**C4.5** [%]	**Naive Bayes** [%]	**SVM** [%]	**BPSO+SVM** [%]
DLBCL	1	56,25	75,00	75,00	**75,00**
	2	**87,50**	75,00	75,00	75,00
	3	73,33	**100**	80,00	80,00
	4	66,67	**73,33**	**73,33**	**73,33**
	5	80,00	**86,67**	73,33	73,33
	$\bar{x}$	72,75	**82,00**	75,33	75,33
	σ	12,04	11,39	**2,74**	**2,74**
CNS	1	33,33	**66,67**	58,33	58,33
	2	41,67	41,67	**66,67**	**66,67**
	3	33,33	**66,67**	**66,67**	**66,67**
	4	66,67	**91,67**	66,67	66,67
	5	50,00	58,33	**66,67**	**66,67**
	$\bar{x}$	45,00	**65,00**	**65,00**	**65,00**
	σ	13,95	18,07	**3,73**	**3,73**
Brain_Tumor2	1	40,00	80,00	30,00	30,00
	2	40,00	**70,00**	30,00	30,00
	3	60,00	**70,00**	30,00	30,00
	4	**70,00**	60,00	30,00	30,00
	5	70,00	**80,00**	30,00	30,00
	$\bar{x}$	56,00	**72,00**	30,00	30,00
	σ	15,17	8,37	**0**	**0**
Prostate_Tumor	1	**85,71**	66,67	52,38	52,38
	2	**85,71**	66,67	52,38	52,38
	3	**85,00**	60,00	50,00	50,00
	4	**70,00**	**70,00**	50,00	50,00
	5	**75,00**	45,00	50,00	50,00
	$\bar{x}$	**80,28**	61,67	50,95	50,95
	σ	7,33	10,00	**1,30**	**1,30**
Leukemia2	1	**100**	**100**	40,00	40,00
	2	86,67	**93,33**	40,00	40,00
	3	**92,86**	**92,86**	35,71	35,71
	4	85,71	**92,86**	35,71	35,71
	5	78,57	**92,86**	42,86	42,86
	$\bar{x}$	88,76	**94,38**	38,86	38,86
	σ	8,07	3,15	**3,10**	**3,10**

Povprečna točnost klasifikacijskih metod C4.5, Naive Bayes, SVM in algoritma BPSO+SVM nad vsemi enajstimi množicami je prikazana na sliki 8.11.



Slika 8.11: Povprečje klasifikacijske točnosti algoritmov BPSO+SVM, C4.5, NB in SVM na intervalu zaupanja 95 %

Povprečne vrednosti mere F in vrednosti AUC pri vseh enajstih podatkovnih množicah so prikazane v tabeli 8.9.

Tabela 8.9: Povprečne vrednosti mere F in vrednosti AUC pri algoritmu BPSO+SVM

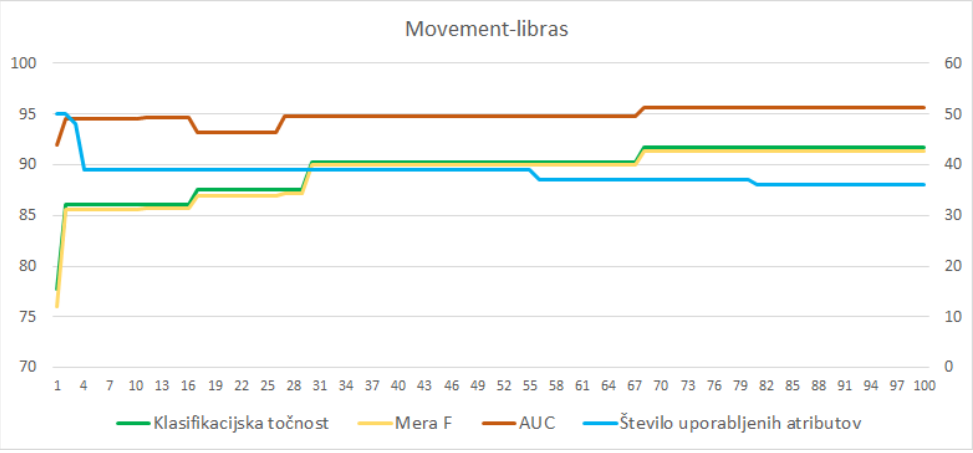
	Podatkovna množica	**mera F**	**AUC**
1	Primary_Tumor	0,384	0,696
2	Ionosphere	0,974	0,964
3	Soybean	0,931	0,964
4	Movement-libras	0,388	0,699
5	SRBCT	1	1
6	Leukemia1	0,365	0,500
7	DLBCL	0,648	0,500
8	CNS	0,513	0,500
9	Brain_Tumor2	0,938	0,969
10	Prostate_Tumor	0,344	0,500
11	Leukemia2	0,973	0,981

Kljub temu, da se je algoritem BPSO+SVM v primerjavi z algoritmoma BPSO+C4.5 in BPSO+NB odrezal najslabše, je nad množico Ionosphere dosegel boljšo povprečno klasifikacijsko točnost (97,54 %) in izbral manj atributov (povprečno 8,76) kot so jih objavili raziskovalci v članku [56] (97,33 % točnost klasifikacije in izbranih 15 atributov).

V tabeli 8.10 so zbrani vsi rezultati povprečnih točnosti klasifikacije in povprečnega števila izbranih atributov. Najvišjo povprečno točnost je dosegel algoritem BPSO+C4.5, ki ga je v zgolj dveh primerih prehitel algoritem BPSO+NB z višjo povprečno točnostjo. Največ izločenih atributov je bilo pri kombinaciji algoritma BPSO+SVM, vendar je imel algoritem tudi najmanjšo izboljšavo točnosti klasifikacije.

Tabela 8.10: Primerjava povprečnih vrednosti točnosti klasifikacije, števila uporabljenih atributov in odstotka izločenih atributov

	Podatkovna množica	**Točnost klasifikacije [%]**			**# uporabljenih atributov**			**izločeni atributi [%]**		
		BPSO +C4.5	**BPSO +NB**	**BPSO +SVM**	**BPSO +C4.5**	**BPSO +NB**	**BPSO +SVM**	**BPSO +C4.5**	**BPSO +NB**	**BPSO +SVM**
1	Primary_Tumor	52,45	**56,58**	49,33	9,24	10,12	**8,20**	45,65	40,47	**51,76**
2	Ionosphere	**97,71**	96,17	97,54	9,96	**7,88**	8,76	70,71	**76,82**	74,24
3	Soybean	**96,69**	95,70	94,06	16,00	15,12	**14,16**	54,29	56,80	**59,54**
4	Movement-libras	**82,56**	73,17	43,28	37,20	37,70	**34,52**	58,67	58,11	**61,64**
5	SRBCT	**100**	**100**	**100**	1037,76	**1025,56**	1028,40	55,04	**55,56**	55,44
6	Leukemia1	**100**	99,18	52,76	2470,16	2496,20	**2450,20**	53,63	53,14	**54,00**
7	DLBCL	**99,23**	92,25	75,33	2569,48	2550,52	**2517,32**	53,02	53,36	**53,97**
8	CNS	**96,00**	76,67	65,00	3408,68	3339,80	**3312,64**	52,19	53,15	**53,53**
9	Brain_Tumor2	**94,00**	82,00	30,00	4958,60	4930,08	**4897,40**	52,17	52,44	**52,76**
10	Prostate_Tumor	**98,21**	64,62	50,95	5012,92	**4975,72**	4984,40	52,30	**52,65**	52,57
11	Leukemia2	97,50	**98,93**	38,86	5359,92	5369,16	**5323,40**	52,25	52,17	**52,58**



Slika 8.12: Prikaz vrednosti mere F, AUC, točnosti klasifikacije in izbranih atributov v odvisnosti od razvoja generacij podatkovne množice Movement-libras

Na sliki 8.12 so prikazane vrednosti mere F, AUC, točnosti klasifikacije in števila izbranih atributov v odvisnosti od razvoja generacij za podatkovno množico Movement-libras. Leva, primarna, os ima definicijsko območje $[0, 100]$ in na njej so prikazane vrednosti klasifikacijske

Tabela 8.11: Primerjava izboljšanja točnosti klasifikacije hibridnega algoritma v primerjavi z uporabljeno klasifikacijsko metodo v cenitveni funkciji

	Podatkovna množica	**C4.5** [%]	**BPSO+C4.5** izboljšanje [%]	**NB** [%]	**BPSO+NB** izboljšanje [%]	**SVM** [%]	**BPSO+SVM** izboljšanje [%]
1	Primary_Tumor	42,79	9,66	49,56	7,02	41,00	8,33
2	Ionosphere	90,87	6,84	82,60	13,57	92,29	5,25
3	Soybean	91,22	5,47	92,68	3,02	88,15	5,91
4	Movement-libras	65,83	16,73	61,39	11,78	29,16	14,12
5	SRBCT	79,70	20,30[1]	98,75	1,25[1]	97,57	2,43[1]
6	Leukemia1	94,48	5,52[1]	94,48	4,70	52,60	0,16
7	DLBCL	72,75	26,48	82,00	10,25	75,33	0
8	CNS	45,00	51,00	65,00	11,67	65,00	0
9	Brain_Tumor2	56,00	38,00	72,00	10,00	30,00	0
10	Prostate_Tumor	80,28	17,93	61,67	2,95	50,95	0
11	Leukemia2	88,76	8,74	94,38	4,55	38,86	0

[1] dosežena 100 % točnost

točnosti, mere F in AUC. Sekundarna, desna, os je namenjena predstavitvi števila uporabljenih atributov.

Tabela 8.11 prikazuje primerjavo izboljšanja točnosti klasifikacije predlaganega algoritma v primerjavi z uporabljeno klasifikacijsko metodo C4.5 ali NB ali SVM v cenitveni funkciji. Do izboljšanja točnosti klasifikacije je prišlo pri vseh uporabljenih množicah, kjer smo algoritem BPSO kombinirali s klasifikacijskima metodama C4.5 in NB v cenitveni funkciji (tabela 8.11). Pri petih množicah ob uporabi algoritma BPSO+SVM do izboljšanja točnosti klasifikacije ni prišlo, vendar je bilo pri tem vseeno izločenih več kot polovica atributov (tabela 8.10). V štirih primerih je algoritem BPSO izboljšal točnost klasifikacije na 100 %.

Rezultate, dobljene z hibridnim algoritmom BPSO+C4.5, smo primerjali z rezultati, objavljenimi v članku [11], kjer so kombinirali algoritem BPSO in tabu iskanje (v nadaljevanju BPSO+TS). Članek navaja zgolj najboljšo točnost klasifikacije in delec z izbranim najmanjšim številom atributov. V primeru, da podatkovna množica, ki smo jo uporabili mi, v članku ni bila uporabljena, je le-to v tabeli 8.12 označeno z znakom /.

Tabela 8.12 prikazuje povprečno vrednost točnosti klasifikacije in najboljšo vrednost klasifikacije hibridnega algoritma BPSO+C4.5 v primerjavi z algoritmom BPSO+TS in povprečno ter najmanjše število izbranih atributov hibridnega algoritma BPSO+C4.5 v primerjavi z BPSO+TS.

Tabela 8.12: Primerjava rezultatov BPSO+C4.5 in BPSO+TS

	Podatkovna množica	**Točnost klasifikacije**			**Število izbranih atributov**		
		BPSO+C4.5		**BPSO+TS**	**BPSO+C4.5**		**BPSO+TS**
		[povprečje]	[najboljši]	[najboljši]	[povprečje]	[najboljši]	[najboljši]
1	Primary_Tumor	/	/	/	/	/	/
2	Ionosphere	/	/	/	/	/	/
3	Soybean	/	/	/	/	/	/
4	Movement-libras	/	/	/	/	/	/
5	SRBCT	**100**	**100**	**100**	1037,76	**1007**	1084
6	Leukemia1	**100**	**100**	**100**	2470,16	**2434**	2577
7	DLBCL	99,23	**100**	**100**	2569,48	**2508**	2671
8	CNS	/	/	/	/	/	/
9	Brain_Tumor2	94,00	**100**	92,65	4958,60	**4847**	5086
10	Prostate_Tumor	98,21	**100**	95,45	5012,92	**4923**	5320
11	Leukemia2	92,86	**100**	**100**	5359,92	**5305**	5609

Najboljše vrednosti točnosti klasifikacije hibridnega algoritma BPSO+C4.5 so enake ali boljše od najboljših vrednosti točnosti klasifikacije pri BPSO+TS. Tudi ob upoštevanju povprečnih vrednosti rezultatov točnosti klasifikacije (npr. najboljši rezultat točnosti klasifikacije BPSO+TS nad podatkovno množico Prostate_Tumor je 95,45 %, že zgolj povprečna točnost klasifikacije algoritma BPSO+C4.5 pa znaša 98,21 %), se je hibridni algoritem BPSO+C4.5 samo pri dveh podatkovnih množicah odrezal slabše kot najboljša klasifikacijska točnost algoritma BPSO+TS. V številu izbranih atributov pa hibridni algoritem BPSO+C4.5 prekaša BPSO+TS tako po najboljši kot tudi povprečni vrednosti izbranih atributov.

9 Statistična obdelava rezultatov

Vse dobljene rezultate smo statistično obdelali. Pri tem smo uporabili opisno oziroma deskriptivno statistiko. To smo storili z namenom odkritja statistično značilnih razlik med predlaganim algoritmom in znanimi klasifikacijskimi algoritmi.

Za pravilno izbiro statističnih testov smo najprej preverili, ali imajo rezultati normalno porazdelitev ali ne, kar je vplivalo na izbiro statističnega testa. Če imajo rezultati normalno porazdelitev, izbiramo med parametričnimi testi, sicer pa med neparametričnimi.

Stopnja ali meja statistične značilnosti, označena z α, je določena subjektivno. Navadno izbiramo med vrednostmi 0,001, 0,05, 0,01 in 0,10. Pri statistični obdelavi smo uporabili stopnjo značilnosti $\alpha = 0,05$, kot jo je predlagal eden izmed največjih sodobnih statistikov R. A. Fisher [55].

Teste smo izvedli s pomočjo programske opreme SPSS [26] za prediktivno analitiko.

9.1 Rezultati algoritma BPSO+C4.5

Začeli smo s preverjanjem uspešnosti algoritma BPSO+C4.5, ki ima v cenitveni funkciji uporabljeno klasifikacijsko metodo C4.5, z uspešnostjo preostalih klasifikacijskih metod C4.5, Naive Bayes in SVM. Statistično smo tako primerjali rezultate točnosti klasifikacije omenjenih algoritmov in iskali statistično značilno razliko med algoritmom BPSO+C4.5 in klasifikacijskimi metodami C4.5, Naive Bayes in SVM.

Najprej smo izvedli Shapiro-Wilkov test, katerega rezultati so predstavljeni v tabeli 9.1. Z njim smo preverili, če rezultati ustrezajo normalni porazdelitvi.

Shapiro-Wilkov test sta leta 1965 uvedla Samuel Sanford Shapiro in Martin Wilk [21]. Uporablja se za preverjanje ničelne hipoteze, da je proučevana porazdelitev enaka normalni

porazdelitvi. Hipoteze ne zavrnemo v primeru, če je ničelna hipoteza pravilna, tj. večja od 0,05 ($p > 0,05$). S p označujemo vrednosti statistične značilnosti naših ugotovitev.

Ker rezultati ne ustrezajo normalni porazdelitvi (tabela 9.1), smo v nadaljevanju uporabili neparametrične teste.

Tabela 9.1: Shapiro-Wilkov test (BPSO+C4.5)

		Shapiro-Wilk		
	algoritem	**statistika**	**df**	**Sig.**
točnost	C4.5	0,909	55	,001
	NB	0,915	55	,001
	SVM	0,926	55	,002
	BPSO+C4.5	0,601	55	,000

Neparametrični izboljšani Friedmanov test smo uporabili za preverjanje prisotnosti statistično značilnih razlik med algoritmi. Če primerjamo več algoritmov na več domenah med seboj, število njihovih primerjav hitro narašča in s tem se zvišuje tudi verjetnost, da so dobljene razlike zgolj naključne. Zato se primerjave navadno lotimo tako, da razviti algoritem primerjamo z referenčnimi.

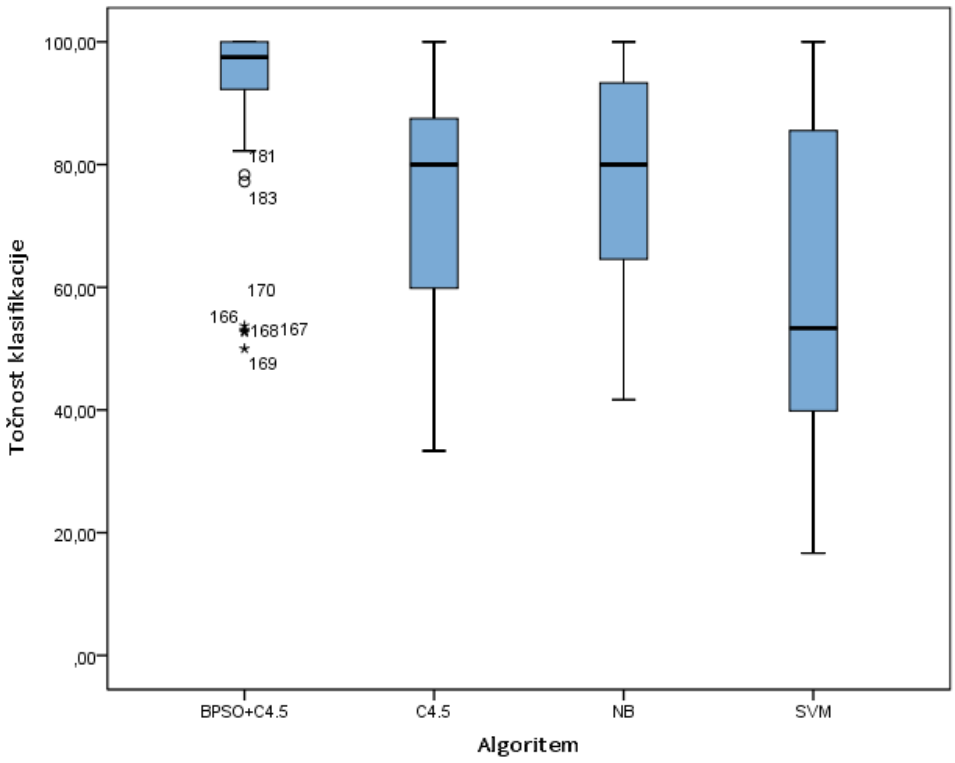
Friedmanov test torej rangira uspešnost vseh testiranih algoritmov na vsaki domeni posebej. Če imamo več algoritmov, ki so enako uspešni, se vsakemu dodeli povprečni rang [35]. V tabeli 9.2 je prikazan rezultat Friedmanovega testa.

Tabela 9.2: Friedmanov test (BPSO+C4.5)

Friedmanov test	
N	55
Chi-Square	102,379
df	3
p	,000

Vidimo, da lahko ničelno hipotezo – vsi algoritmi so enako uspešni – zavržemo, saj je Friedmanov test pokazal obstoj statistično značilnih razlik med različnimi klasifikacijskimi metodami (algoritmi), saj je $p < 0,001$.

Škatla z brki ali kvartilni diagram (angl. Box plot) za povprečje klasifikacijske točnosti algoritmov C4.5, NB in SVM je prikazan na sliki 9.1.



Slika 9.1: Mediana klasifikacijske točnosti algoritmov BPSO+C4.5, C4.5, NB in SVM

Če algoritme razvrstimo po vrednosti mediane (srednje vrednosti), si sledijo tako: BPSO+C4.5 (mediana = 97,50), Naive Bayes (mediana = 80,00), C4.5 (mediana = 80,00) in SVM (mediana = 53,33). Algoritem BPSO+C4.5 ima najmanjši kvartilni razmik (razpon od prvega do tretjega kvartila) in tudi najmanjši variacijski razmik (razpon od najmanjše do največje vrednosti statističnega znaka oziroma rezultatov) med vsemi analiziranimi algoritmi.

Nadaljevali smo s testi Post Hoc. Wilcoxonov test predznačenih rangov je neparametrični ekvivalent t-testu za dva odvisna vzorca [21], ki ga je predstavil Frank Wilcoxon leta 1945. Ne predpostavlja normalne porazdelitve v podatkih in rangira absolutne razlike uspešnosti med dvema algoritmoma na domenah [35].

Ker smo izvedli več primerjav različnih algoritmov nad podatkovnimi množicami, smo uporabili Holm-Bonferronijev popravek [21, 35]. Z njo lahko zavržemo ničelno hipotezo zgolj v primeru, če je razlika med algoritmi relativno velika. Popravek smo izvedli tako, da smo rezultate signifikance (p) razvrstili po naraščajočem vrstnem redu. Najmanjšega smo pomnožili z N, kjer N predstavlja število primerjav, ki v našem primeru znaša 3. Drugega najmanjšega smo pomnožili z N-1 in to ponavljali vse do zadnjega, ki smo ga pomnožili z 1.

Predstavimo postopek na primeru iz tabele 9.3. Po naraščajočem vrstnem redu si vrednosti dejanskega p sledijo tako: 2,3821E-9, 5,1174E-10 in 3,4841E-10. Največjega pomnožimo s 3 in dobimo vrednost 7,15E-9, drugega z 2 in dobimo vrednost 1,02E-9, zadnjega pa prepišemo in tako dobimo 3,48E-10.

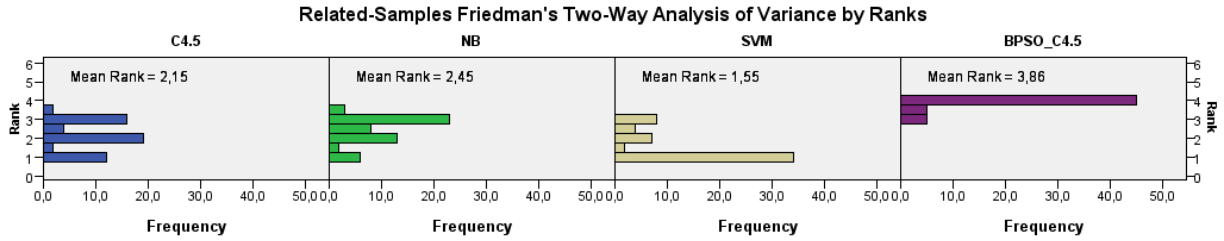
Rezultati Wilcoxonovega testa predznačenih rangov, vključujoč Holm-Bonferronijev popravek za algoritem BPSO+C4.5, so prikazani v tabeli 9.3.

Tabela 9.3: Wilcoxonov test predznačenih rangov s popravkom za BPSO+C4.5

	C4.5 - BPSO+C4.5	**NB - BPSO+C4.5**	**SVM - BPSO+C4.5**
Z	-6,215	-5,969	-6,276
p zaokrožen	,000	,000	,000
dejanski p	5,1174E-10	2,3821E-9	3,4841E-10
p s Holm-Bonferroni popravkom	,000***	,000***	,000***

Wilcoxonov test predznačenih rangov s Holm-Bonferronijevim popravkom razkriva, da obstajajo statistično značilne razlike med analiziranimi algoritmi. Algoritem BPSO+C4.5 je statistično značilno boljši od klasifikacijske metode C4.5 ($p < 0,001$), klasifikacijske metode Naive Bayes ($p < 0,001$) in klasifikacijske metode SVM ($p < 0,001$).

Ker med algoritmi obstajajo statistično značilne razlike, jih lahko razvrstimo po njihovi uspešnosti. Na podlagi rezultatov testa dvosmerne analize variance za ponavljajoče se meritve na rangih, prikazanega na sliki 9.2, vidimo, kakšne range so dosegli analizirani algoritmi. Najboljši, tj. najvišji rang je dosegel algoritem BPSO+C4.5 (povprečni rang = 3,86), sledi mu klasifikacijska metoda Naive Bayes (povprečni rang = 2,45), nato klasifikacijska metoda C4.5 (povprečni rang = 2,15) in klasifikacijska metoda SVM (povprečni rang = 1,55).



Slika 9.2: Rangi za C4.5, NB, SVM in BPSO+C4.5

9.2 Rezultati algoritma BPSO+Naive Bayes

Postopek smo ponovili še za algoritem BPSO, kjer smo v cenitveni funkciji uporabili klasifikacijsko metodo Naive Bayes. Najprej smo preverili, če rezultati ustrezajo normalni porazdelitvi s Shapiro-Wilkovim testom, katerega rezultat je prikazan v tabeli 9.4.

Tabela 9.4: Shapiro-Wilkov test (BPSO+NB)

		Shapiro-Wilk		
	algoritem	**statistika**	**df**	**Sig.**
točnost	C4.5	,909	55	,001
	NB	,915	55	,001
	SVM	,926	55	,002
	BPSO+NB	,846	55	,000

Ker tudi tukaj rezultati ne ustrezajo normalni porazdelitvi, smo nadaljevali z neparametričnimi testi. Friedmanov test je pokazal, da tudi pri algoritmu BPSO+NB obstajajo statistično značilne razlike med algoritmi (tabela 9.5), saj je $p < 0,001$.

Tabela 9.5: Friedmanov test (BPSO+NB)

Friedmanov test	
N	55
Chi-Square	81,042
df	3
p	,000

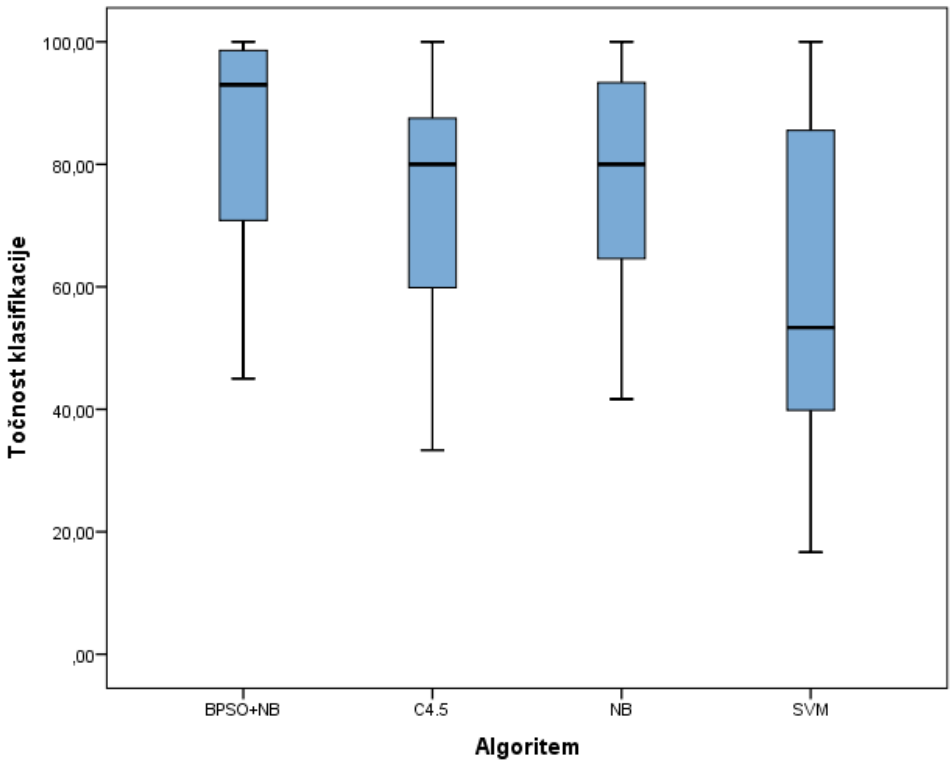
Nadaljevali smo s primerjavo in algoritme razvrstili po vrednosti mediane. Njihov vrstni red je naslednji: BPSO+NB (mediana = 92,99), klasifikacijska metoda Naive Bayes (mediana = 80,00), klasifikacijska metoda C4.5 (mediana = 80,00) in klasifikacijska metoda SVM (mediana = 53,33), kar je razvidno iz slike 9.3.

Wilcoxonov test predznačnih rangov, vključujoč Holm-Bonferronijev popravek za algoritem BPSO+Naive Bayes, je prikazan v tabeli 9.6. Iz nje je razvidno, da je algoritem BPSO+Naive Bayes statistično značilno boljši od klasifikacijske metode C4.5 ($p < 0,001$), klasifikacijske metode Naive Bayes ($p < 0,001$) in klasifikacijske metode SVM ($p < 0,001$).

Zaradi dokaza, da je algoritem BPSO+NB statistično značilno boljši, lahko algoritme rangi-

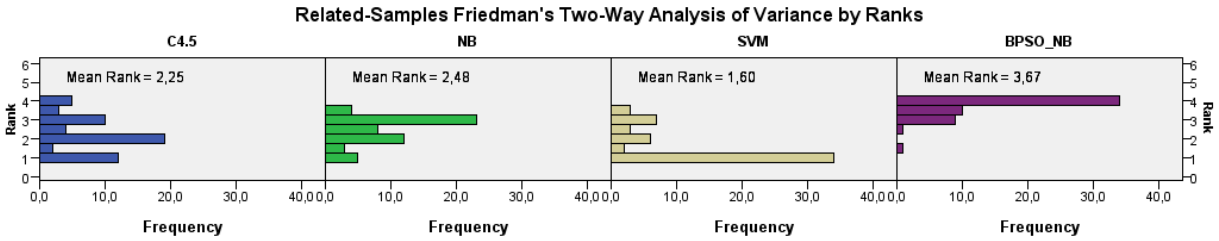
Tabela 9.6: Wilcoxonov test predznačenih rangov s popravkom za BPSO+NB

	C4.5 - BPSO+NB	**NB - BPSO+NB**	**SVM - BPSO+NB**
Z	-4,750	-5,713	-5,990
p zaokrožen	,000	,000	,000
p s Holm-Bonferroni popravkom	,000***	,000***	,000***



Slika 9.3: Mediana klasifikacijske točnosti algoritmov BPSO+NB, C4.5, NB in SVM

ramo. Tudi tukaj si želimo čim višje vrednosti ranga, saj želimo imeti čim višjo klasifikacijsko točnost. Rangirani algoritmi na podlagi testa dvosmerne analize variance za ponavljajoče se meritve na rangih si sledijo v naslednjem vrstnem redu (slika 9.4): BPSO+NB (povprečni rang = 3,67), klasifikacijska metoda Naive Bayes (povprečni rang = 2,48), klasifikacijska metoda C4.5 (povprečni rang = 2,25) in klasifikacijska metoda SVM (povprečni rang = 1,60).



Slika 9.4: Rangi za C4.5, NB, SVM in BPSO+NB

9.3 Rezultati algoritma BPSO+SVM

Preverjanje smo izvedli še z algoritmom BPSO+SVM. Test normalne porazdelitve je prikazan v tabeli 9.7 in razkriva, da podatki ne ustrezajo normalni porazdelitvi.

Tabela 9.7: Shapiro-Wilkov test (BPSO+SVM)

		Shapiro-Wilk		
	algoritem	**statistika**	**df**	**Sig.**
točnost	C4.5	,909	55	,001
	NB	,915	55	,001
	SVM	,926	55	,002
	BPSO+SVM	,886	55	,000

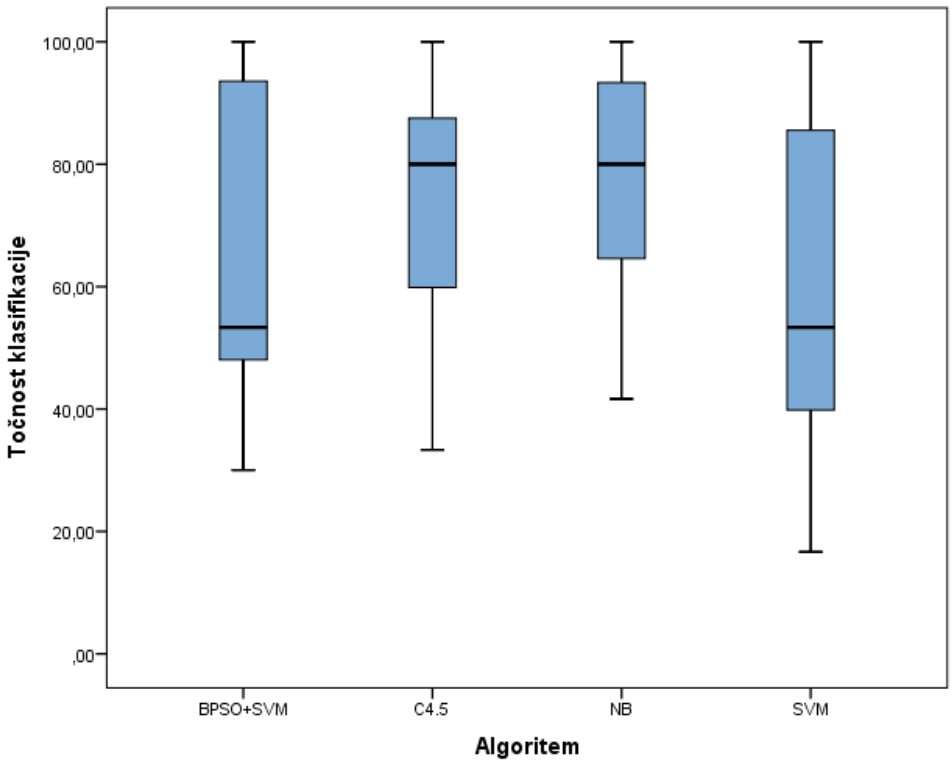
Tudi pri algoritmu BPSO+SVM je Friedmanov test pokazal, da obstajajo med algoritmi statistično značilne razlike (tabela 9.8), saj je $p < 0,001$.

Tabela 9.8: Friedmanov test (BPSO+SVM)

Friedmanov test	
N	55
Chi-Square	28,915
df	3
p	,000

Nadajevali smo s primerjavo median analiziranih algoritmov, kar prikazuje slika 9.5. Algoritmi si po vrstnem redu sledijo tako: Naive Bayes (mediana = 80,00), C4.5 (mediana = 80,00), BPSO+SVM (mediana = 53,33) in SVM (mediana = 53,33).

Wilcoxonov test predznačnih rangov, vključujoč Holm-Bonferronijev popravek za algoritem BPSO+SVM, je prikazan v tabeli 9.9.

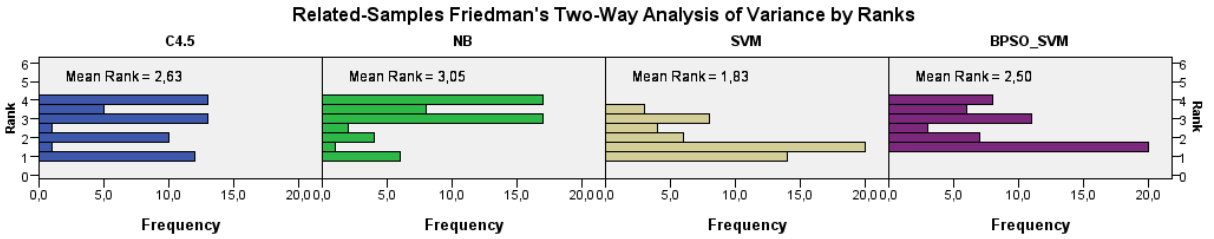


Slika 9.5: Mediana klasifikacijske točnosti algoritmov BPSO+SVM, C4.5, NB in SVM

Iz tabele 9.9 je razvidno, da med algoritmom BPSO+SVM in klasifikacijskima metodama NB in SVM statistično značilne razlike obstajajo. Med algoritmom BPSO+SVM in klasifikacijsko metodo C4.5 statistično značilne razlike ni ($p = 0{,}057$), zato ne moremo sklepati o tem, kateri algoritem izmed njiju je boljši.

Tabela 9.9: Wilcoxonov test predznačenih rangov s popravkom za BPSO+SVM

	C4.5 - BPSO+SVM	**NB - BPSO+SVM**	**SVM - BPSO+SVM**
Z	-2,346	-3,799	-4,015
p zaokrožen	,019	,000	,000
p s Holm-Bonferroni popravkom	,057	,000***	,000***



Slika 9.6: Rangi za C4.5, NB, SVM in BPSO+SVM

Glede na primerjavo rangov algoritmov na podlagi testa dvosmerne analize variance za ponavljajoče se meritve na rangih (slika 9.6), si algoritmi sledijo v naslednjem vrstnem redu: klasifikacijska metoda Naive Bayes (povprečni rang = 3,05), klasifikacijska metoda C4.5 (povprečni rang = 2,63), BPSO+SVM (povprečni rang = 2,50) in klasifikacijska metoda SVM (povprečni rang = 1,83).

9.4 Rezultati primerjave algoritmov BPSO+C4.5, BPSO+NB in BPSO+SVM

Nazadnje smo izvedli še primerjave med algoritmi BPSO+C4.5, BPSO+NB in BPSO+SVM. Test normalne porazdelitve je prikazan v tabeli 9.10 in razkriva, da podatki ne ustrezajo normalni porazdelitvi.

Tabela 9.10: Shapiro-Wilkov test (BPSO+C4.5, BPSO+NB in BPSO+SVM)

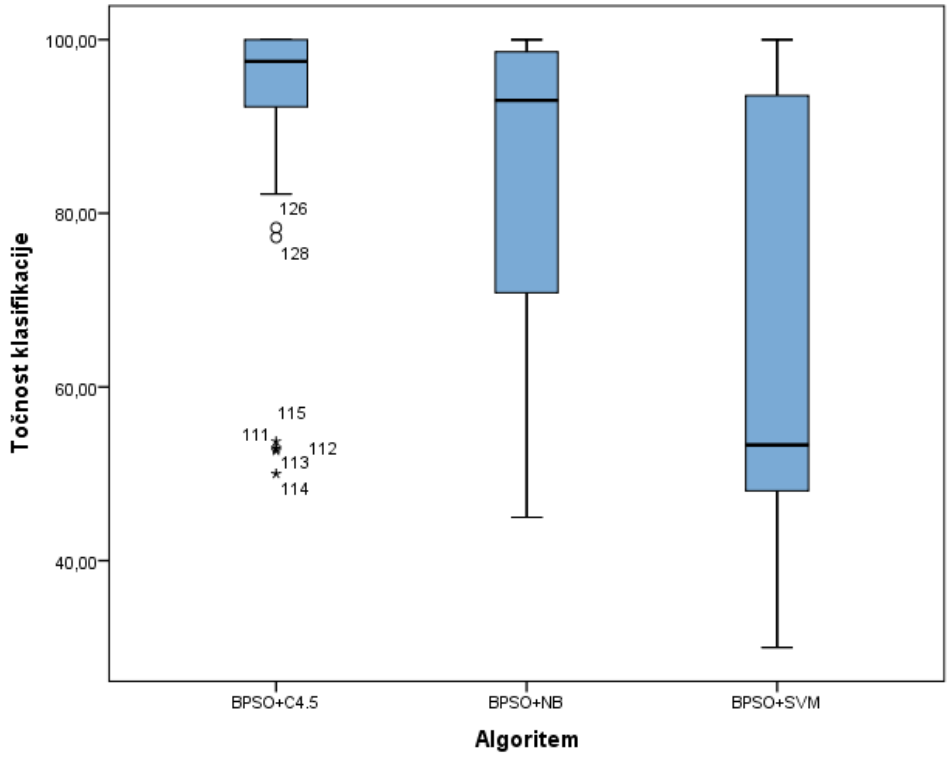
		Shapiro-Wilk		
	algoritem	**statistika**	**df**	**Sig.**
točnost	BPSO+C4.5	,601	55	,000
	BPSO+NB	,846	55	,000
	BPSO+SVM	,886	55	,000

Friedmanov test je pokazal, da med algoritmi BPSO+C4.5, BPSO+NB in BPSO+SVM statistično značilne razlike obstajajo (tabela 9.11), saj je $p < 0,001$. Nadajevali smo s primerjavo median analiziranih algoritmov, kar prikazuje slika 9.7. Algoritmi si po vrstnem redu sledijo tako: BPSO+C4.5 (mediana = 97,50), BPSO+NB (mediana = 92,99) in BPSO+SVM (mediana = 53,33).

Tabela 9.11: Friedmanov test (BPSO+C4.5, BPSO+NB, BPSO+SVM)

Friedmanov test	
N	55
Chi-Square	64,043
df	2
p	,000

Wilcoxonov test predznačnih rangov, vključujoč Holm-Bonferronijev popravek za algoriteme BPSO+C4.5, BPSO+NB in BPSO+SVM, je prikazan v tabeli 9.12.

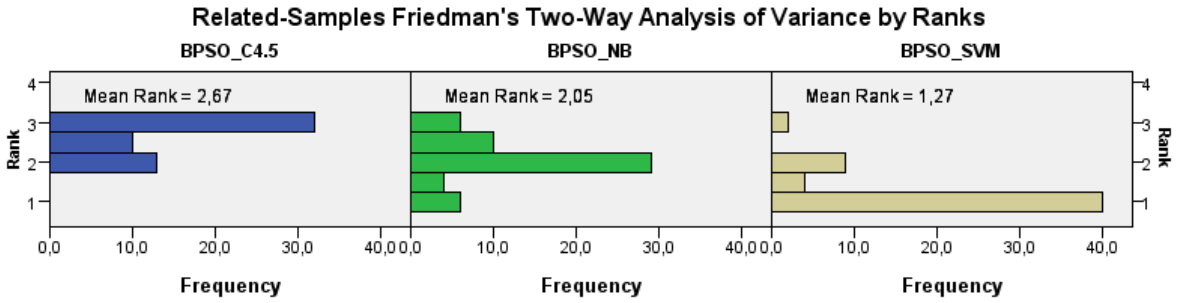


Slika 9.7: Mediana klasifikacijske točnosti hibridnih algoritmov BPSO+C4.5, BPSO+NB in BPSO+SVM

Tabela 9.12: Wilcoxonov test predznačenih rangov s popravkom

	BPSO+C4.5 - BPSO+NB	**BPSO+C4.5 - BPSO+SVM**	**BPSO+NB - BPSO+SVM**
Z	-4,289	-6,102	-5,578
p zaokrožen	,000	,000	,000
p s Holm-Bonferroni popravkom	,000***	,000***	,000***

Iz tabele 9.12 je razvidno, da med vsemi algoritmi statistično značilne razlike obstajajo. Zato lahko nadaljujemo s sklepanjem o tem, kateri izmed njih je boljši.



Slika 9.8: Rangi za BPSO+C4.5, BPSO+NB in BPSO+SVM

Glede na primerjavo rangov algoritmov na podlagi testa dvosmerne analize variance za ponavljajoče se meritve na rangih (slika 9.6), si algoritmi sledijo v naslednjem vrstnem redu: BPSO+C4.5 (povprečni rang = 2,67), BPSO+NB (povprečni rang = 2,05) in BPSO+SVM (povprečni rang = 1,27).

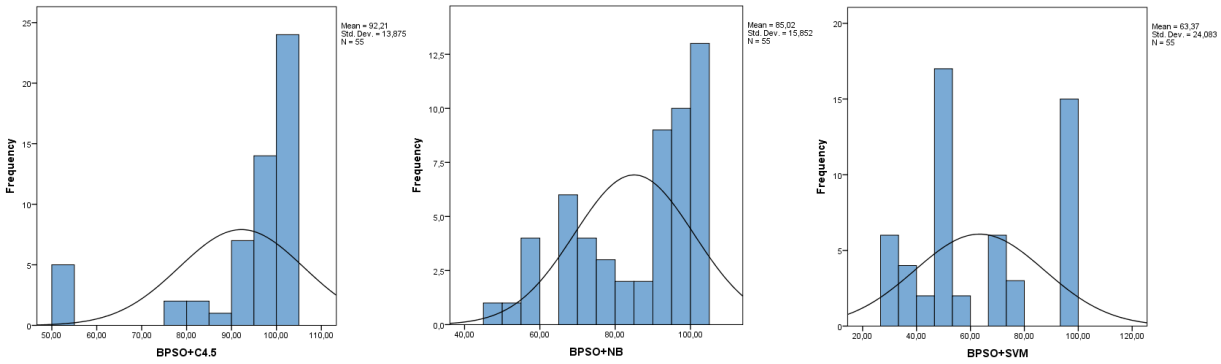
Glede na statistično značilno razliko med algoritmi BPSO+C4.5, BPSO+NB in BPSO+SVM ter na podlagi testa dvosmerne analize variance za ponavljajoče se meritve na rangih smo dokazali, da je izmed primerjanih algoritmov najboljši BPSO+C4.5.

9.5 Interpretacija rezultatov

Izbrane statistične teste in njihove rezultate smo prikazali v podpoglavjih 9.1, 9.2, 9.3 in 9.4. V tem podpoglavju se bomo osredotočili na interpretacijo rezultatov in podali glavne ugotovitve.

Najprej smo preverili, če rezultati algoritmov ustrezajo normalni porazdelitvi. Ugotovili smo, da rezultati algoritmov BPSO+C4.5, BPSO+NB in BPSO+SVM (slika 9.9) ter klasifikacijskih metod C4.5, Naive Bayes in SVM nimajo normalne porazdelitve. Takšen rezultat je bil pričakovan, še posebno pri uporabi algoritma BPSO. Slednjega smo namreč uporabili z namenom izboljšanja klasifikacije (gravitiranje proti 100 % točnosti klasifikacije), kar pa ni v skladu z normalno porazdelitvijo.

Zaradi nenormalne porazdelitve rezultatov smo v nadaljevanju statističnega preverjanja uporabili neparametrične teste.



Slika 9.9: Prikaz nenormalne porazdelitve rezultatov BPSO+C4.5, BPSO+NB in BPSO+SVM

Wilcoxonov test predznačnih rangov, vključujoč Holm-Bonferronijev popravek, je dokazal, da obstajajo statistično značilne razlike med algoritmom BPSO v kombinaciji s klasifikacijskimi metodami C4.5, Naive Bayes in SVM v cenitveni funkciji ter klasifikacijskimi metodami C4.5, Naive Bayes in SVM. Edina primerjava, kjer statistično značilna razlika ni bila dokazana, je bila primerjava algoritma BPSO+SVM in klasifikacijske metode C4.5. Algoritem BPSO+SVM je statistično značilno slabši od klasifikacijskih metod Naive Bayes in C4.5 (slika 9.6), v vseh preostalih primerih pa je hibridni algoritem BPSO statistično značilno boljši.

Wilcoxonov test predznačenih rangov s Holm-Bonferronijevim popravkom (tabela 9.12) je

razkril, da med vsemi hibridnimi algoritmi (BPSO+C4.5, BPSO+NB in BPSO+SVM) statistično značilne razlike obstajajo. Zaradi tega lahko sklepamo o tem, kateri izmed njih je najboljši. Najbolje se je odrezal algoritem BPSO v kombinaciji s klasifikacijsko metodo C4.5 v cenitveni funkciji. Ta sklep potrjuje tudi prikaz rezultatov rangov testa, prikazanega na sliki 9.8.

Algoritem BPSO+C4.5 v zgolj treh primerih ni presegel klasifikacijske točnosti 96 % (tabela 8.10). V omenjenih treh primerih je dosegel točnost 52,45 %, 82,56 % in 94 %, kar pomeni, da je vseeno izboljšal točnost same klasifikacijske metode C4.5 nad istimi množicami, ki znašajo 42,79 %, 65,83 % in 56,00 %. Pri izboljšani točnosti klasifikacije pa je v povprečju izločil več kot polovico atributov iz izvorne podatkovne množice (tabela 8.10).

Na podlagi rezultatov, predstavljenih v poglavju 8, in statistične analize rezultatov v tem poglavju lahko hipotezo, da izbira reprezentativnih atributov v kombinaciji z algoritmom binarne optimizacije z rojem delcev izboljša točnost klasifikacije, potrdimo.

10 Sklep

Število znanstvenih in strokovnih objav na področju raziskav inteligence rojev, natančneje optimizacije z rojem delcev, se v zadnjih letih močno povečuje. Ravno ta dinamičnost in vsestranska možnost uporabe inteligence rojev na problemih različnih vrst je pretehtala k odločitvi, da smo v sklopu magistrskega dela izbrali področje inteligence rojev, natančneje algoritem binarne optimizacije z rojem delcev.

Metodo FS-BPSO, znotraj katere smo preverjali hibridne algoritme BPSO+C4.5, BPSO+NB in BPSO+SVM, smo uspešno implementirali. Hibridni algoritem binarne optimizacije z rojem delcev je z uporabo metode izbire atributov nad skoraj vsemi podatkovnimi množicami izboljšal točnost klasifikacije in pri tem uporabil manj kot polovico atributov iz podatkovnih množic.

Za primerjavo rezultatov točnosti klasifikacije z izbranimi klasifikacijskimi metodami in točnosti klasifikacije hibridnega algoritma smo izvedli eksperiment. Eksperiment tako predstavlja primarno raziskovalno metodo, s pomočjo katere smo preverili zastavljeno hipotezo. Odločitev za omenjeno metodo je bila jasna, saj omogoča preverjanje vplivanja neodvisnih spremenljivk na odvisne, pri čemer raziskovalec sodeluje v aktivni vlogi in operira s kvantitativnimi podatki. Subjekte oziroma primerke v uporabljenih množicah podatkov smo testirali večkrat in primerjali rezultate pred in po uporabi postopka, tj. hibridnega algoritma binarne optimizacije z rojem delcev.

Pri izvedbi magistrskega dela je imela bistveni pomen tudi natančna izvedba sistematičnega pregleda literature. Našli smo tako članke, ki se ukvarjajo s sorodno tematiko našega magistrskega dela, kot tudi izvedene pregledne članke, ki obravnavajo splošno področje optimizacije z rojem delcev. Iz vsakega članka s sorodno tematiko smo skušali pridobili čim več informacij, ki bi lahko izboljšale kakovost izvedbe eksperimenta.

Eksperimentov primarnih raziskav, predstavljenih v člankih, seveda ni možno neposredno primerjati med seboj, saj algoritmi uporabljajo različne nastavitve in nabore podatkov. Vsi eksperimenti tudi nimajo navedenih podatkov glede izvedbe, zato je njihova notranja veljavnost eksperimenta vprašljiva, predvsem pa niso ponovljivi.

S pomočjo statistične analize rezultatov smo dokazali, da je razvit hibridni algoritem BPSO v kombinaciji s klasifikacijskimi metodami C4.5, Naive Bayes in SVM v cenitveni funkciji statistično značilno boljši od nekaterih izmed najbolj uporabljanih klasifikacijskih metod, tj. C4.5, Naive Bayes in SVM. Tudi pri agoritmu BPSO+SVM, kjer je bila opažena najnižja stopnja točnosti klasifikacije, smo dokazali, da je izboljšal točnost klasifikacije v primerjavi s klasifikacijsko metodo SVM, ali pa pri isti stopnji točnosti klasifikacije uporabil več kot polovico manj atributov.

10.1 Omejitve eksperimenta

Podatkovnih množic v sklopu magistrskega dela nismo predprocesirali, kot so to na primer storili različni drugi avtorji [9, 10]. Želeli smo izpostaviti delovanje algoritma BPSO v kombinaciji z izbranimi klasifikacijskimi metodami in ne optimizirati rezultata na zgolj določeni podatkovni množici.

Nastavitve metode FS-BPSO so za vse analizirane podatkovne množice enake. Zaradi tega nismo mogli optimizirati rezultatov nad določeno množico s spreminjanjem nastavitev algoritma.

Za cenitveno funkcijo smo uporabili točnost klasifikacije. V primeru spremembe cenitvene funkcije bi lahko optimizirali kakšne druge vidike, kot so mera F, AUC in podobno. Lahko bi predlagali tudi hibridno cenitveno funkcijo, kjer bi z utežmi določili pomembnost posameznega dela cenitvene funkcije.

10.2 Priložnosti za nadaljnje delo

Priložnosti za nadaljnje delo je ogromno, še posebno zaradi dokazane statistične značilne razlike med točnostjo klasifikacije algoritma BPSO in klasifikacijskimi metodami v obliki cenitvene funkcije ter točnostjo klasifikacije samih klasifikacijskih metod. Glede na dobljene rezultate bomo večjo pozonost namenili optimizaciji algoritma BPSO+C4.5, ki je dal najboljše rezultate.

Narava metode FS-BPSO in njena implementacija dopuščata uporabo poljubne podatkovne množice. S tem je povečana uporabnost metode in zagotovljena splošnost ter neodvisnost od zastavljenega problema.

Smiselna bi lahko bila raziskava o tem, kako spreminjanje nastavitev algoritma BPSO vpliva na rezultate točnosti klasifikacije. Pri tem bi spreminjali vrednosti parametrov c_1, c_2, v_{min}, v_{max} in ω. Lahko bi uvedli spreminjajočo se vztrajnostno utež, ki bi svojo vrednost spreminjala skozi generacije.

Zanimiva je tudi že omenjena ideja o hibridni cenitveni funkciji ali kombinaciji algoritma BPSO s kakšno drugo metodo (genetski algoritmi, tabu iskanje ipd.), kjer bi primerjali vpliv takšne hibridnosti na klasifikacijsko točnost.

Navsezadnje si želimo izvesti interdisciplinarno raziskavo na področju medicine pri odkrivanju informativnih genov določenih bolezni. Primarna izbira za takšno raziskavo bi bila uporaba hibridnega algoritma BPSO+C4.5, saj klasifikacijska metoda C4.5 ne le da omogoča visoko točnost klasifikacije, pač pa je njen rezultat v obliki odločitvenega drevesa lahko preverjen s strani eksperta [48].

Besedilo magistrskega dela je oblikovano z urejevalnikom LaTeX

Literatura

[1] Adam. *Jata ptic*. Dostopno na: `https://www.sciencenews.org/sites/default/files/2015/12/main/articles/121615_ti_cellmovement_free.jpg` [2. 8. 2016].

[2] H. Ahmed and J. Glasgow. Swarm Intelligence: Concepts, Models and Applications. Technical report, 2012.

[3] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

[4] M. Belal, J. Gaber, H. El-Sayed, and A. Almojel. Swarm Intelligence. In *Handbook of Bioinspired Algorithms and Applications*, 2006.

[5] J. C. Bezdek. What is Computational Intelligence? *Computational Intelligence Imitating Life*, pages 1–12, 1994.

[6] A. Bhattacherjee. *Social science research: principles, methods, and practices*. CreateSpace Independent Publishing Platform, 2nd edition, 2012.

[7] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.

[8] C.-C. Chang and C.-J. Lin. LIBSVM: A Library for Support Vector Machines. Technical report, Department of Computer Science National Taiwan University, Taipei, Taiwan, 2013.

[9] K.-H. Chen, K.-J. Wang, M.-L. Tsai, K.-M. Wang, M.-A. Angelia, W.-C. Cheng, T.-S. Yang, N.-C. Teng, K.-P. Tan, and K.-S. Chang. Gene selection for cancer identification: a decision tree model empowered by particle swarm optimization algorithm. *BMC Bioinformatics*, 15(49):1–10, 2014.

[10] K.-H. Chen, K.-J. Wang, K.-M. Wang, and M.-A. Angelia. Applying particle swarm optimization-based decision tree classifier for cancer classification on gene expression data. *Applied Soft Computing*, 24:773–780, 2014.

[11] L.-Y. Chuang, C.-H. Yang, and C.-H. Yang. Tabu Search and Binary Particle Swarm Optimization for Feature Selection Using Microarray Data. *http://dx.doi.org/10.1089/cmb.2007.0211*, 2010.

[12] Cody. *Franchises: The First Batman Film Series. Batman Returns (1992)*. Dostopno na: `http://codysfilmandtvblog.blogspot.si/2012/07/franchises-first-batman-film-series.html` [2. 8. 2016].

[13] D. W. Corne, A. P. Reynolds, and E. Bonabeau. Swarm Intelligence. In *Handbook of Natural Computing*. 2012.

[14] E. S. Correa, A. A. Freitas, and C. G. Johnson. Particle Swarm for Attribute Selection in Bayesian Classification: An Application to Protein Function Prediction. *Journal of Artificial Evolution and Applications*, 2008:1–12, 2008.

[15] A. Cuthbertson. Artificial Intelligence Turns $20 Into $11,000 In Kentucky Derby Bet. *Newsweek*, May 2016.

[16] S. Dara and H. Banka. A Binary PSO Feature Selection Algorithm for Gene Expression Data. In *2014 International Conference on Advances in Communication and Computing Technologies*, pages 1–6. IEEE, 2014.

[17] M. Dorigo. The Editorial Special Issue: Swarm Intelligence. *Swarm Intelligence Journal*, 1(1):1–2, 2007.

[18] F. Ducatelle, G. A. Di Caro, and L. M. Gambardella. Principles and applications of swarm intelligence for adaptive routing in telecommunications networks. *Swarm Intelligence*, 4(3):173–198, September 2010.

[19] M. Elbedwehy, H. Zawbaa, N. Ghali, and A. Hassanien. Detection of heart disease using binary particle swarm optimization. In *Computer Science and Information Systems (FedCSIS), 2012 Federated Conference on*, pages 177–182, Wroclaw, 2012. IEEE.

[20] A. P. Engelbrecht. *Computational intelligence an introduction*. John Wiley & Sons, 2007.

[21] A. Field. *Discovering Statistics Using IBM SPSS Statistics*. SAGE Publications Ltd, 4th edition, 2013.

[22] A. C. Godinez, L. E. M. Espinosa, and E. M. Montes. An Experimental Comparison of Multiobjective Algorithms: NSGA-II and OMOPSO. In *2010 IEEE Electronics, Robotics and Automotive Mechanics Conference*, pages 28–33, Morelos, September 2010. IEEE.

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and W. I. H. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1), 2009.

[24] H. M. Harb and A. S. Desuky. Feature Selection on Classification of Medical Datasets based on Particle Swarm Optimization. *International Journal of Computer Applications*, 104(5):975–8887, 2014.

[25] F. Heppner and U. Grenander. A Stochastic Nonlinear Model for Coordinate Bird Flocks. In *The Ubiquity of Chaos*, pages 233–238, Washington, DC, 1990. AAAS Publications.

[26] IBM. *IBM SPSS Software*. Dostopno na: `http://www.ibm.com/analytics/us/en/technology/spss/` [25. 7. 2016].

[27] R. Jensen and Q. Shen. *Computational intelligence and feature selection: rough and fuzzy approaches*. IEEE Press, 2008.

[28] G. H. John, R. Kohavi, and K. Ppeger. Irrelevant Features and the Subset Selection Problem. In *Machine Learning: Proceedings of the Eleventh International Conference*, pages 121–129, San Francisco, CA, 1994. Morgan Kaufmann Publishers.

[29] S. Kar, K. Das Sharma, and M. Maitra. Gene selection from microarray gene expression data for classification of cancer subgroups employing PSO and adaptive K-nearest neighborhood technique. *Expert Systems with Applications*, 42(1):612–627, 2015.

[30] S. Kaur and R. Jyoti. Increasing Efficiency of Crowd Simulation Using Particle Swarm Optimization. *ISSN International Journal of Innovative Research in Computer and Communication Engineering*, 1(4):2320–9798, 2013.

[31] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, Perth, WA, 1995. IEEE.

[32] J. Kennedy and R. Eberhart. A discrete binary version of the particle swarm algorithm. In *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, volume 5, pages 4104–4108, Orlando, FL, 1997. IEEE.

[33] R. Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. 1995.

[34] P. Kokol, Š. Hleb Babič, V. Podgorelec, and M. Zorman. *Inteligentni sistemi*. Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru, 2001.

[35] I. Kononenko and M. Robnik Šikonja. *Inteligentni sistemi*. Založba FE in FRI, Ljubljana, 1st edition, 2010.

[36] J. Krause and G. D. Ruxton. *Living in groups*. Oxford University Press, 2002.

[37] J. Li, L. Ding, and B. Li. A Novel Naive Bayes Classification Algorithm Based on Particle Swarm Optimization. *The Open Automation and Control Systems Journal*, 6:747–753, 2014.

[38] M. Lichman. *UCI Machine Learning Repository*. Dostopno na: `http://archive.ics.uci.edu/ml` [20. 7. 2016].

[39] E. Limer. *Watch the Navy's LOCUST Launcher Fire Off a Swarm of Autonomous Drones*, 24. 5. 2016. Dostopno na: `http://www.popularmechanics.com/military/weapons/a21008/navy-locust-launcher-test-2016/` [21. 7. 2016].

[40] H. Liu and H. Motoda. *Computational methods of feature selection*. Chapman & Hall/CRC, 2008.

[41] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, and S. Wang. An Improved Particle Swarm Optimization for Feature Selection. *Journal of Bionic Engineering*, 8(2):191–200, July 2011.

[42] Massive. *Massive Software*. Dostopno na: `http://www.massivesoftware.com/index.html` [21. 7. 2016].

[43] U. Maulik, S. Bandyopadhyay, J. T. L. Wang, and Wiley InterScience (Online service). *Computational intelligence and pattern analysis in biology informatics*. John Wiley &

Sons, 2010.

[44] M. Meissner, M. Schmuker, and G. Schneider. Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training. *BMC Bioinformatics*, 7(125):1–11, 2006.

[45] M. S. Mohamad, S. Omatu, S. Deris, M. Yoshioka, A. Abdullah, and Z. Ibrahim. An enhancement of binary particle swarm optimization for gene selection in classifying cancer classes. *Algorithms for Molecular Biology*, 8(1):15, 2013.

[46] H. B. Nguyen, B. Xue, I. Liu, and M. Zhang. Filter based backward elimination in wrapper based PSO for feature selection in classification. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 3111–3118, Beijing, July 2014. IEEE.

[47] C. Perry. *The 1,000-robot swarm*. Dostopno na: `http://news.harvard.edu/gazette/story/2014/08/the-1000-robot-swarm/` [2. 8. 2016].

[48] V. Podgorelec, P. Kokol, B. Stiglic, and I. Rozman. Decision trees: an overview and their use in medicine. *Journal of Medical Systems*, 26(5):445–463, 2002.

[49] J. Pogačnik and S. Tamše. *Svet žuželk*. Dostopno na: `http://www.veselasola.net/veselasola.net/portal/ucne_poti/plus-zuzelke-20130202/index#state=1` [2. 8. 2016].

[50] D. Ramyachitra, M. Sofia, and P. Manikandan. Interval-value Based Particle Swarm Optimization algorithm for cancer-type specific gene selection and sample classification. *Genomics Data*, 5:46–50, sep 2015.

[51] C. W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*, 21(4):25–34, 1987.

[52] S. Sathiyamoorthy, B. E. Caroline, and J. G. Jayanthi, editors. *Emerging Trends in Science, Engineering and Technology*. Lecture Notes in Mechanical Engineering. Springer India, India, 2012.

[53] A. Shukla, R. Tiwari, and R. Kala. *Real life applications of soft computing*. CRC Press, 2010.

[54] N. Siddique and H. Adeli. *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing*. Wiley-Blackwell, 2013.

[55] L. D. Torbeck. Statistical Solutions: On the Verge of Significance: Why 5%. *Pharmaceutical Technology*, 34(7), 2010.

[56] C.-J. Tu, L.-Y. Chuang, J.-Y. Chang, and C.-H. Yang. Feature Selection using PSO-SVM. *IAENG International Journal of Computer Science*, 1(33), 2007.

[57] P. Tucker. *Inside the Navy's Secret Swarm Robot Experiment*, 5. 10. 2014. Dostopno na: `http://www.defenseone.com/technology/2014/10/inside-navys-secret-swarm-robot-experiment/95813/` [21. 7. 2016].

[58] Unanimous. *UNU platforma*. Dostopno na: `http://unu.ai/` [21. 7. 2016].

[59] Univerza Plymouth. *Microarray Cancers*. Dostopno na: `http://www.tech.plym.ac.uk/spmc/links/bioinformatics/microarray/microarray_cancers.html` [20. 7. 2016].

[60] E. Vassev, R. Sterritt, C. Rouff, L. Martin, and M. Hinchey. Swarm Technology at NASA: Building Resilient Systems. *Computing now*, 2012.

[61] I. H. I. H. Witten and E. Frank. *Data mining: practical machine learning tools and techniques*. Morgan Kaufman, 2005.

[62] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. Mclachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowl Inf Syst*, 14:1–37, 2008.

[63] H. Zhang. The Optimality of Naive Bayes. In *American Association for Artificial Intelligence*, 2004.

[64] Y. Zhang, D.-w. Gong, and J. Cheng. Multi-objective Particle Swarm Optimization Approach for Cost-based Feature Selection in Classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, PP:1–13, 2015.

[65] Y. Zhang, S. Wang, and G. Ji. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. *Mathematical Problems in Engineering*, 2015:38, 2015.

Priloge

A Razredi podatkovnih množic

Za vsako uporabljeno podatkovno množico je prikazan seznam razredov množice. V primeru, da kodiranje nad množicami ni bilo uporabljeno, ima opis razredov le en stolpec.

Primary_Tumor
Uporabljeni so naslednji razredi:
==========
lung
head & neck
esophasus
thyroid
stomach
duoden & sm.int
colon
rectum
anus
salivary glands
pancreas
gallblader
liver
kidney
bladder
testis
prostate
ovary
corpus uteri
cervix uteri
vagina
breast

ionosphere Uporabljeno je naslednje kodiranje:
==========
good g
bad b

soybean

Uporabljeno je naslednje kodiranje:

diaporthe-stem-canker	0
charcoal-rot	1
rhizoctonia-root-rot	2
phytophthora-rot	3
brown-stem-rot	4
powdery-mildew	5
downy-mildew	6
brown-spot	7
bacterial-blight	8
bacterial-pustule	9
purple-seed-stain	10
anthracnose	11
phyllosticta-leaf-spot	12
alternarialeaf-spot	13
frog-eye-leaf-spot	14
diaporthe-pod-&-stem-blight	15
cyst-nematode	16
2-4-d-injury	17
herbicide-injury	18

movement-libras

Uporabljeni so naslednji razredi:

curved swing
horizontal swing
vertical swing
anti-clockwise arc
clockwise arc
circle
horizontal straight-line
vertical straight-line
horizontal zigzag
vertical zigzag
horizontal wavy
vertical wavy
face-up curve
face-down curve
tremble

SRBCT

Uporabljeno je naslednje kodiranje:

EWS	0
RMS	1
BL	2
NB	3

Leukemia1

Uporabljeno je naslednje kodiranje:

ALL B-cell	0
ALL T-cell	1
AML	2

DLBCL

Uporabljeno je naslednje kodiranje:

DLBCL	0
FL	1

CNS

Uporabljeno je naslednje kodiranje:

umrli	Class0
preživeli	Class1

Brain_Tumor2

Uporabljeno je naslednje kodiranje:

Classic Glioblastomas	0
Classic Anaplastic Oligodendrogliomas	1
Non-classic Glioblastomas	2
Non-classic Anaplastic Oligodendrogliomas	3

Prostate_Tumor

Uporabljeno je naslednje kodiranje:

Tumor	0
Normal	1

Leukemia2

Uporabljeno je naslednje kodiranje:

AML	0
ALL	1
MLL	2

Univerza v Mariboru

Fakulteta za elektrotehniko, računalništvo in informatiko

FERI

IZJAVA O AVTORSTVU

Spodaj podpisani/-a LUCIJA BREZOČNIK

z vpisno številko E5022845

sem avtor/-ica magistrskega dela z naslovom:

OPTIMIZACIJA Z ROJEM DELCEV ZA IZBIRO ATRIBUTOV PRI KLASIFIKACIJI

(naslov magistrskega dela)

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)

  red. prof. dr. VILI PODGORELEC

  in somentorstvom (naziv, ime in priimek)

- so elektronska oblika magistrskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko magistrskega dela.
- soglašam z javno objavo elektronske oblike magistrskega dela v DKUM.

V Mariboru, dne 12. 8. 2016

Podpis avtorja/-ice:

Lucija Brezočnik

Univerza v Mariboru

Fakulteta za elektrotehniko, računalništvo in informatiko

FERI

IZJAVA O USTREZNOSTI ZAKLJUČNEGA DELA

Podpisani mentor :

red. prof. dr. VILI PODGORELEC

(ime in priimek mentorja)

in somentor (eden ali več, če obstajata):

(ime in priimek somentorja)

Izjavljam (-va), da je študent

Ime in priimek: LUCIJA BREZOČNIK

Vpisna številka: E5022845

Na programu: INFORMATIKA IN TEHNOLOGIJE KOMUNICIRANJA MAG

izdelal zaključno delo z naslovom:

OPTIMIZACIJA Z ROJEM DELCEV ZA IZBIRO ATRIBUTOV PRI KLASIFIKACIJI

(naslov zaključnega dela v slovenskem in angleškem jeziku)

PARTICLE SWARM OPTIMIZATION IN FEATURE SELECTION FOR CLASSIFICATION

v skladu z odobreno temo zaključnega dela, Navodilih o pripravi zaključnih del in mojimi (najinimi oziroma našimi) navodili.

Preveril (-a, -i) in pregledal (-a, -i) sem (sva, smo) poročilo o plagiatorstvu.

Datum in kraj:
MARIBOR, 12. 8. 2016

Podpis mentorja:

Datum in kraj:

Podpis somentorja (če obstaja):

Priloga:

- Poročilo o preverjanju podobnosti z drugimi deli.

Univerza v Mariboru

Fakulteta za elektrotehniko, računalništvo in informatiko

FERI

IZJAVA O ISTOVETNOSTI TISKANE IN ELEKTRONSKE VERZIJE ZAKLJUČNEGA DELA IN OBJAVI OSEBNIH PODATKOV DIPLOMANTOV

Ime in priimek avtorja-ice: LUCIJA BREZOČNIK

Vpisna številka: E5022845

Študijski program: INFORMATIKA IN TEHNOLOGIJE KOMUNICIRANJA MAG

Naslov zaključnega dela: OPTIMIZACIJA Z ROJEM DELCEV ZA IZBIRO ATRIBUTOV PRI KLASIFIKACIJI

Mentor: red. prof. dr. VILI PODGORELEC

Somentor:

Podpisani-a LUCIJA BREZOČNIK izjavljam, da sem za potrebe arhiviranja oddal elektronsko verzijo zaključnega dela v Digitalno knjižnico Univerze v Mariboru. Zaključno delo sem izdelal-a sam-a ob pomoči mentorja. V skladu s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovoljujem, da se zgoraj navedeno zaključno delo objavi na portalu Digitalne knjižnice Univerze v Mariboru.

Tiskana verzija zaključnega dela je istovetna z elektronsko verzijo elektronski verziji, ki sem jo oddal za objavo v Digitalno knjižnico Univerze v Mariboru.

Zaključno delo zaradi zagotavljanja konkurenčne prednosti, varstva industrijske lastnine ali tajnosti podatkov naročnika: _____ ne sme biti javno dostopno do _____(datum odloga javne objave ne sme biti daljši kot 3 leta od zagovora dela).

Podpisani izjavljam, da dovoljujem objavo osebnih podatkov, vezanih na zaključek študija (ime, priimek, leto in kraj rojstva, datum zaključka študija, naslov zaključnega dela), na spletnih straneh in v publikacijah UM.

Datum in kraj: MARIBOR, 12. 8. 2016 Podpis avtorja-ice: Lucija Brezočnik

Podpis mentorja: _____
(samo v primeru, če delo ne sme biti javno dostopno)

Podpis odgovorne osebe naročnika in žig:
(samo v primeru, če delo ne sme biti javno dostopno) _____