Univerza v Mariboru

Fakulteta za zdravstvene vede

AVTOMATSKO NAČRTOVANJE IN VREDNOTENJE KLASIFIKACIJSKIH CEVOVODOV V BIOINFORMATIKI

(Magistrsko delo)

Maribor, 2019 Iztok Fister ml.

Univerza v Mariboru

Fakulteta za zdravstvene vede

AVTOMATSKO NAČRTOVANJE IN VREDNOTENJE KLASIFIKACIJSKIH CEVOVODOV V BIOINFORMATIKI

(Magistrsko delo)

Maribor, 2019 Iztok Fister ml.

Univerza v Mariboru

Fakulteta za zdravstvene vede

Mentor: red. prof. dr. Milan Zorman

Zahvala

Zahvaljujem se mentorju prof. dr. Milanu Zormanu za vso pomoč pri nastanku tega zaključnega dela. Zahvala tudi vsem sodelavcem iz laboratorija Newton za podporo ob mojih odsotnostih zaradi predavanj in izpitov.

Navsezadnje velika hvala tudi moji družini za potrpljenje ob mojih odsotnostih tekom tega študija.

Priloga 6 – IZJAVA O AVTORSTVU IN ISTOVETNOSTI TISKANE IN ELEKTRONSKE OBLIKE ZAKLJUČNEGA DELA

UNIVERZA V MARIBORU
Fakulteta za zdravstvene vede
(ime članice UM)

IZJAVA O AVTORSTVU IN ISTOVETNOSTI TISKANE IN ELEKTRONSKE OBLIKE ZAKLJUČNEGA DELA

Ime in priimek študent-a/-ke: Iztok Fister
Študijski program: BIOINFORMATIKA
Naslov zaključnega dela: Avtomatsko načrtovanje in vrednotenje klasifikacijskih cevovodov v bioinformatiki

Mentor: Milan Zorman
Somentor: _____

Podpisan-i/-a študent/-ka Iztok Fister

- izjavljam, da je zaključno delo rezultat mojega samostojnega dela, ki sem ga izdelal/-a ob pomoči mentor-ja/-ice oz. somentor-ja/-ice;
- izjavljam, da sem pridobil/-a vsa potrebna soglasja za uporabo podatkov in avtorskih del v zaključnem delu in jih v zaključnem delu jasno in ustrezno označil/-a;
- na Univerzo v Mariboru neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve avtorskega dela v elektronski obliki, pravico reproduciranja ter pravico ponuditi zaključno delo javnosti na svetovnem spletu preko DKUM; sem seznanjen/-a, da bodo dela deponirana/objavljena v DKUM dostopna široki javnosti pod pogoji licence Creative Commons BY-NC-ND, kar vključuje tudi avtomatizirano indeksiranje preko spleta in obdelavo besedil za potrebe tekstovnega in podatkovnega rudarjenja in ekstrakcije znanja iz vsebin; uporabnikom se dovoli reproduciranje brez predelave avtorskega dela, distribuiranje, dajanje v najem in priobčitev javnosti samega izvirnega avtorskega dela, in sicer pod pogojem, da navedejo avtorja in da ne gre za komercialno uporabo;
- dovoljujem objavo svojih osebnih podatkov, ki so navedeni v zaključnem delu in tej izjavi, skupaj z objavo zaključnega dela;
- izjavljam, da je tiskana oblika zaključnega dela istovetna elektronski obliki zaključnega dela, ki sem jo oddal/-a za objavo v DKUM.

Uveljavljam permisivnejšo obliko licence Creative Commons: _____ (navedite obliko)

Začasna nedostopnost:

Zaključno delo zaradi zagotavljanja konkurenčne prednosti, zaščite poslovnih skrivnosti, varnosti ljudi in narave, varstva industrijske lastnine ali tajnosti podatkov naročnika:
_____(naziv in naslov naročnika/institucije) ne sme biti javno dostopno do _____(datum odloga javne objave ne sme biti daljši kot 3 leta od zagovora dela). To se nanaša na tiskano in elektronsko obliko zaključnega dela.

Temporary unavailability:

To ensure competition priority, protection of trade secrets, safety of people and nature, protection of industrial property or secrecy of customer's information, the thesis _____(institution/company name and address) must not be accessible to the public till_____(delay date of thesis availability to the public must not exceed the period of 3 years after thesis defense). This applies to printed and electronic thesis forms.

Datum in kraj: Maribor, 08.07.2019 Podpis študent-a/-ke:

_____

Podpis mentor-ja/-ice: _____
(samo v primeru, če delo ne sme biti javno dostopno)

Ime in priimek ter podpis odgovorne osebe naročnika in žig:

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

AVTOMATSKO NAČRTOVANJE IN VREDNOTENJE KLASIFIKACIJSKIH CEVOVODOV V BIOINFORMATIKI

Povzetek

Izhodišča in namen: Velikokrat na bioinformatskih podatkih izvajamo klasifikacijo, tj. razvrščanje elementov, predstavljenih z značilnicami, v enega od vnaprej določenih razredov. Sam postopek klasifikacije je zelo kompleksen, saj sestoji iz preprocesiranja podatkov, izbire klasifikatorske metode in optimizacije hiperparametrov. Zaradi kompleksnosti vse tri omenjene korake združujemo v t. i. klasifikacijske cevovode, katere morajo uporabniki, ki niso specialisti na področju strojnega učenja, načrtovati ročno. Ta postopek je časovno zelo zapleten, v določenih primerih pa se ne uspemo približati optimalni rešitvi.

Raziskovalna metodologija: Avtomatski razvoj in vrednotenje klasifikacijskih cevovodov smo donedavno reševali s pomočjo genetskega programiranja (angl. Genetic Programming, krajše GP), kjer posameznike predstavimo z drevesnimi strukturami. V tem magistrskem delu predlagamo novo rešitev za reševanje omenjenega problema s pomočjo stohastičnih populacijskih algoritmov po vzorih iz narave, kjer so posamezniki predstavljeni kot vektorji realnih števil.

Rezultati: Rezultati na bioinformatskih podatkovnih zbirkah dokazujejo, da so stohastični populacijski algoritmi po vzorih iz narave enostavni za uporabo in hkrati učinkoviti za avtomatski razvoj klasifikacijskih cevovodov.

Diskusija in zaključek: Ugotavljamo, da predlagana metoda omogoča uporabo poljubnega stohastičnega populacijskega algoritma po vzorih iz narave za avtomatsko načrtovanje klasifikacijskih cevovodov, kjer so posamezniki predstavljeni kot vektorji realnih števil.

Ključne besede: algoritmi po vzorih iz narave, AutoML, diferencialna evolucija, klasifikacija, optimizacija

AUTOMATIC DESIGN AND VALUATION OF CLASSIFICATION PIPELINES IN BIOINFORMATICS

Abstract

Purpose: Many times, we conduct classification on bioinformatics data, i.e. classifying elements represented by features into one of several predefined classes. The classification process is very complex because of performing many complex tasks, like preprocessing data, selecting the classifier method and hyperparameter optimization. Due to the complexity, all three steps are merged in so-called classification pipelines, where users who are not machine learning experts need to manage them manually. However, this process is very time-consuming, and does not ensure that the optimal solution for the particular pipeline is found.

Methodology: Until now, an automatic development and evaluation of classification pipelines was performed using Genetic Programming (GP). In this master thesis, we propose a new method for solving the problem using stochastic population-based nature-inspired algorithms, where individuals are represented as real valued vectors.

Results: The results on bioinformatic datasets demonstrate that stochastic population-based nature-inspired algorithms are user friendly, and effective for the automatic design of classification pipelines.

Discussion and conclusions: We conclude that the proposed method enables the use of any stochastic population-based nature-inspired algorithm for the automatic design of classification pipelines, where individuals are represented as real valued vectors.

Keywords: nature-inspired algorithms, AutoML, Differential Evolution, classification, optimization

Kazalo vsebine

Kazalo slik

Kazalo tabel

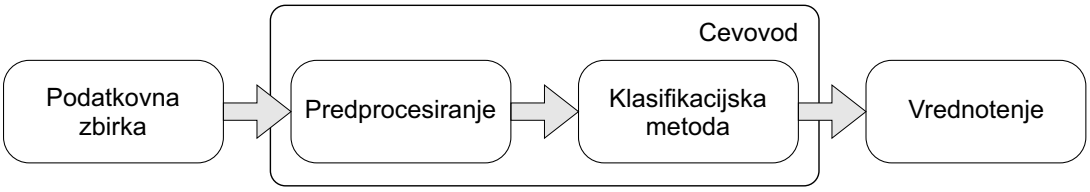
1 Uvod in opis problema

V magistrskem delu predstavljamo stohastične populacijske algoritme po vzorih in narave (angl. stochastic population-based nature-inspired algorithms) (Back, 1996) ter njihovo uporabo pri avtomatskem načrtovanju klasifikacijskih cevovodov (angl. classification pipelines) (Russell & Norvig, 2016) v bioinformatiki (angl. bioinformatics). Stohastični populacijski algoritmi po vzoru iz narave imajo že dolgo tradicijo in se uporabljajo za reševanje težkih optimizacijskih problemov na različnih področjih, kot npr. v ekonomiji (Chen & Kuo, 2002) ali medicini (Pena-Reyes & Sipper, 2000). Ti algoritmi poskušajo reševati težke probleme s posnemanjem principov naravnih, bioloških, fizikalnih in kemijskih sistemov. V družino stohastičnih populacijskih algoritmov po vzoru iz narave v grobem štejemo: evolucijske algoritme (angl. Evolutionary Algorithms, krajše EA) (Eiben & Smith, 2003) in algoritme na osnovi inteligence rojev (angl. Swarm Intelligence, krajše SI) (Beni, 2009).

Bioinformatika je področje, na katerem so raziskovalci dnevno soočeni z veliko količino podatkov, npr. pri sekvenciranju DNA (Glenn, 2011) ali eksperimentih s pomočjo mikromrež (Ding & Peng, 2005), kjer generirajo ogromne količine podatkov in jih kasneje tudi analizirajo. Velikokrat na teh podatkih izvajajo klasifikacijo, tj. razvrščanje elementov, predstavljenih z značilnicami, v enega od vnaprej določenih razredov. Klasifikacijske probleme navadno rešujemo s pomočjo strojnega učenja, ki ponuja veliko število raznovrstnih metod. Številne izmed njih so se skozi dolgotrajen razvoj izkazale kot zelo uspešne. A kljub temu uspešnosti strojnega učenja ne gre ocenjevati na osnovi rezultatov, dobljenih na posameznih klasifikacijskih problemih oz. posebnih družin problemov. Uporabniki klasifikacijskih metod morajo zato določen nabor metod predhodno preizkusiti, kar je sicer dolgotrajen proces. Po drugi strani pa je klasifikacijski postopek zelo kompleksen, saj sestoji iz naslednjih glavnih korakov:

- predprocesiranja podatkov,
- izbire klasifikacijske metode,
- izbire najboljše kombinacije specifičnih parametrov izbrane klasifikacijske

Slika 1: Tipični klasifikacijski cevovod.



metode, tj. optimizacija hiperparametrov.

Običajno združujemo omenjene korake v t. i. klasifikacijski cevovod, kjer izhod enega koraka predstavlja vhod v drugega. Tipični klasifikacijski cevovod je predstavljen na Sliki 1, pri čemer velja, da je predprocesiranje podatkov (Schmieder & Edwards, 2011) najbolj zapleten korak v celotnem ekosistemu strojnega učenja, v katerem se soočamo s podatki, ki bodo odigrali osnovno vlogo tudi v vseh prihajajočih korakih (García, Luengo & Herrera, 2015). Surovi podatki so lahko zelo grdi (angl. ugly data), saj lahko vsebujejo manjkajoče ali nestandardizirane podatke, lahko pa so že v dobri obliki in pripravljeni za uporabo. V skladu s tem velja odstraniti nekatere značilnice, ki ne vplivajo na rezultate klasifikacije.

Pri izbiri ustreznega klasifikatorja navadno predhodno preverimo metode sorodnih študij, a trenutno takih, ki neposredno prikazujejo uspešnost klasifikacijskih metod na določenih problemih, obstaja le malo. Ta metoda zato ni učinkovita, saj nimamo dovolj informacij o pripadajočih podatkih. Po drugi strani nas včasih omejuje tudi strojna oprema, ker v tem primeru ne gre izkoristiti celotnega procesa strojnega učenja. Še posebej se ta problem odraža na novodobnem globokem učenju (LeCun, Bengio & Hinton, 2015), kjer je učinkovitost učnega procesa izdatno povezana s procesorsko močjo.

Nenazadnje velik problem uporabnikom predstavljajo tudi specifični nadzorni parametri klasifikacijskih metod, ki so po navadi cela oz. realna števila ali celo nizi znakov (angl. strings). Pri metodi naključnih gozdov (Breiman, 2001) npr. obstaja parameter, ki ima pomemben vpliv na klasifikacijo in se imenuje število ocenjevalcev (angl. Number of estimators). Vsi koraki načrtovanja klasifikacijskih cevovodov so zaradi kompleksnosti posameznih korakov zelo zahtevni.

Avtomatizirano strojno učenje (angl. automated machine learning, krajše Au-

toML) (Feurer et al., 2015) predstavlja ključno rešitev tega problema. AutoML je skupek metod za avtomatizacijo nekaterih korakov strojnega učenja, še posebej preprocesiranja, izbire klasifikacijske metode in izbire optimalnih hiperparametrov. Velik doprinos AutoML je tudi posplošitev in poenostavitev celotnega klasifikacijskega ekosistema za uporabnike, ki niso programerji.

Trenutno obstaja že kar nekaj prosto dostopnih rešitev za avtomatizirano strojno učenje, kot npr. Auto-WEKA (Thornton et al., 2013), auto-sklearn (Feurer et al., 2015) in Auto-Keras (Jin, Song & Hu, 2018). AutoML lahko modeliramo kot optimizacijski problem. Obstaja že kar nekaj rešitev, ki temeljijo na stohastičnih optimizacijskih algoritmih po vzorih iz narave. TPOT (Olson & Moore, 2016) je evolucijski algoritem, predstavljen z drevesi za avtomatizacijo strojnega učenja. Temelji na genetskem programiranju, ki ga je razvil Koza (1992). Nekaj izboljšav tega algoritma je predstavljeno v magistrski nalogi Gijsbers (2018). RECIPE avtorjev de Sá et al. (2017) je še en primer genetskega programiranja za načrtovanje klasifikacijskih cevovodov, medtem ko je članek (Xavier-Júnior et al., 2018) predstavil evolucijski algoritem za avtomatsko izbiro najboljšega ansambla klasifikatorjev, kot tudi njegovih pripadajočih parametrov.

Kot je razvidno iz prejšnjega odstavka, večina obstoječih metod temelji na genetskem programiranju. Različne algoritme po vzorih iz narave je zelo težko uporabiti na problemu avtomatiziranega strojnega učenja, saj je potrebno vsak osnovni algoritem po vzoru iz narave prilagoditi temu problemu. Prav tako se velikokrat zgodi, da rešitve, ki temeljijo na genetskem programiranju, generirajo tudi t. i. nedopustne (angl. infeasible) klasifikacijske cevovode. V naši raziskavi gremo zato korak naprej. Razviti želimo rešitev, ki bo temeljila na algoritmih po vzorih iz narave, znotraj katerih so rešitve (tj. posamezniki v populaciji) predstavljeni kot vektorji realnih števil. V tem primeru ni potrebno definirati nobenih posebnih operatorjev, kot tudi ne prilagajati nobenega osnovnega algoritma (angl. canonical algorithm) za problem avtomatiziranega strojnega učenja.

1.1 Struktura magistrske naloge

Magistrska naloga je razdeljena na teoretični in eksperimentalni del. V teoretičnem delu na kratko opišemo algoritme po vzorih iz narave za optimizacijo ter principe avtomatiziranega strojnega učenja. Zatem opišemo metodo NiaAML, ki je nastala v sklopu tega magistrskega dela. Eksperimentalni del zajema praktične eksperimente z razvito metodo na treh bioinformatskih podatkovnih zbirkah.

Struktura magistrske naloge je naslednja: v Poglavju 2 so opisani glavni cilji tega magistrskega dela, medtem ko so v Poglavju 3 zajeta glavna raziskovalna vprašanja. Raziskovalna metodologija je opisana v Poglavju 4. Teoretično poglavje o stohastičnih populacijskih algoritmih po vzorih iz narave se nahaja v Poglavju 5. Eksperimentalni del je zajet v Poglavju 7. Magistrsko nalogo zaključuje sklep v Poglavju 8.

2 Namen in cilji zaključnega dela

Cilj magistrskega dela je proučevanje različnih algoritmov po vzorih iz narave in njihova uporaba za avtomatsko načrtovanje klasifikacijskih cevovodov na področju bioinformatike.

Cilji teoretičnega dela magistrskega dela so naslednji:

1. pregled literature o algoritmih po vzorih iz narave,
2. pregled literature o obstoječih metodah za avtomatski razvoj in vrednotenje klasifikacijskih cevovodov in
3. načrtovanje metode za avtomatski razvoj in vrednotenje klasifikacijskih cevovodov s pomočjo algoritmov po vzorih iz narave.

Cilji eksperimentalnega dela magistrskega dela so naslednji:

1. implementacija metode (iz prejšnje točke) v programskem jeziku Python,
2. ovrednotenje metode na bioinformatskih podatkovnih zbirkah in
3. prikazovanje praktične vrednosti predlagane metode.

3 Raziskovalna vprašanja

V tem delu smo si zastavili naslednja raziskovalna vprašanja, na katera želimo dobiti odgovore:

- RV1: Katere so obstoječe metode za avtomatsko načrtovanje klasifikacijskih cevovodov?
- RV2: Ali lahko s pomočjo stohastičnih populacijskih algoritmov po vzorih iz narave, kjer so posamezniki predstavljeni z realnimi števili, razvijemo avtomatsko metodo za razvoj in vrednotenje klasifikacijskih cevovodov v bioinformatiki?
- RV3: Kateri stohastični populacijski algoritmi po vzorih iz narave so najbolj primerni za obravnavane probleme?

Na RV1 bomo odgovorili s pregledom obstoječe literature, medtem ko bomo odgovore na RV2 in RV3 dobili z izvedbo eksperimentov.

4 Raziskovalna metodologija

Glavni del raziskav tekom magistrskega dela bo usmerjen v razvoj nove rešitve za avtomatsko načrtovanje in vrednotenje klasifikacijskih cevovodov. Najprej bomo pregledali sorodna dela in natančno analizirali izbrane algoritme po vzorih iz narave. Za tem bomo implementirali izbrane algoritme po vzorih iz narave in jih prilagodili problemu avtomatskega načrtovanja klasifikacijskih cevovodov. Na koncu bomo izvedli še ustrezno verifikacijo na realnih bioinformatskih podatkovnih zbirkah. Opravili bomo tudi primerjalno analizo vsaj dveh različnih algoritmov po vzorih iz narave za izbrani problem.

4.1 Raziskovalne metode

Koraki raziskav bodo naslednji:

1. pregled in analiza različnih algoritmov po vzorih iz narave,
2. pregled in analiza znanih rešitev avtomatskega načrtovanja klasifikacijskih cevovodov,
3. razvoj nove metode NiaAML,
4. primerjava različnih algoritmov po vzorih iz narave za avtomatsko načrtovanje klasifikacijskih cevovodov.

4.2 Raziskovalno okolje

Za izvedbo eksperimenta ne potrebujemo posebnega laboratorijskega okolja. Uporabili bomo prosto dostopne podatkovne zbirke, ki jih uporabljajo tudi ostali raziskovalci na tem področju. Prav tako ne potrebujemo posebne strojne opreme. Dovolj je le računalnik, ki ima vsaj 16 GB pomnilnika. Dodatno je zaželjena tudi grafična kartica, ki omogoča hitrejše izvajanje algoritmov.

4.3 Raziskovalni vzorec

V tem delu uporabimo javno dostopne podatkovne zbirke iz UCI Machine Learning Repository, ki spadajo pod okrilje bioinformatike. Tabela 1 predstavlja glavne karakteristike izbranih zbirk.

Tabela 1: Uporabljene podatkovne zbirke.

Ime zbirke	Število primerkov	Število atributov	Manjkajoči podatki	Referenca
Yeast	1484	8	Ne	UCI (2019*c*)
Ecoli	336	8	Ne	UCI (2019*b*)
Abalone	4177	8	Ne	UCI (2019*a*)

Pri podatkovni zbirki Yeast (angl. kvasovke) in Ecoli napovedujemo lokalizacijsko mesto beljakovin, medtem ko pri zbirki Abalone napovedujemo starost morskega ušesa (angl. abalone) iz fizičnih meritev.

4.4 Etični vidik

Eksperimente bomo izvedli na že obstoječih podatkovnih zbirkah, ki jih uporabljajo tudi ostali raziskovalci. Avtorji podatkovnih zbirk ne omenjajo nobenih potencialnih problemov z etičnim vidikom.

4.5 Predpostavke in omejitve raziskave

Trenutno obstaja že več kot 100 različnih populacijskih algoritmov po vzorih iz narave. V naši raziskavi se bomo na podlagi obstoječe literature osredotočili le na najbolj razširjene in učinkovite. Eksperimente bomo izvedli s pomočjo algoritma roja delcev (Kennedy & Eberhart, 1995) in diferencialne evolucije (Storn & Price, 1997).

5 Stohastični populacijski algoritmi po vzorih iz narave

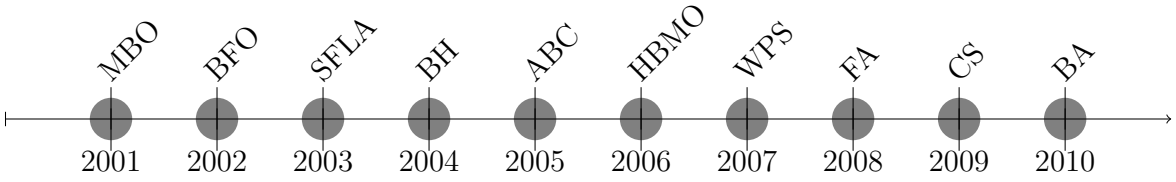
V tem poglavju bralca seznanimo z osnovnimi koncepti stohastičnih populacijskih algoritmov po vzorih iz narave za reševanje optimizacijskih problemov. Podpoglavje 5.1 osvetli navdihe, ki so služili raziskovalcem pri razvoju teh algoritmov, medtem ko podpoglavji 5.1.1 in 5.1.2 podrobneje opišeta izbrana algoritma, ki ju uporabimo v eksperimentalnem delu te magistrske naloge.

5.1 Različni navdihi za razvoj algoritmov po vzorih iz narave

Kot pravi že sam termin *algoritmi po vzorih iz narave* (Fister Jr. et al., 2013), snovalci teh algoritmov iščejo navdih za njihovo delovanje v opazovanju narave. Prvi predstavniki algoritmov po vzorih iz narave so evolucijski algoritmi. Navdih za te algoritme leži v Darwinovi teoriji o boju za obstanek. Darwin (1872) je trdil, da imajo tisti posamezniki, ki so se sposobni prilagoditi pogojem v okolju, več možnosti, da preživijo in se reproducirajo. Slabe posameznike bo proces naravne selekcije izločil iz populacije. Evolucijski algoritmi sestojijo iz treh operatorjev: mutacija, križanje in selekcija (de Castro, 2007). Prvi zametki evolucijskih algoritmov so se že pojavili v 60. letih prejšnjega stoletja. Dandanes celotna družina evolucijskih algoritmov zajema: genetske algoritme, genetsko programiranje, evolucijske strategije, diferencialno evolucijo.

Nekatere biološke vrste žuželk in ptičev so razvile posebno vedenje, ki jim pomaga bodisi pri iskanju hrane ali zaščiti pred plenilci in je na nek način inteligentno. To vedenje imenujemo tudi inteligenca roja. Definicija, ki jo podaja članek avtorjev Fister Jr. et al. (2013), pravi, da se inteligenca rojev nanaša na kolektivno, nastajajoče vedenje več agentov, ki pri sobivanju upoštevajo nekaj preprostih pravil. Medtem ko lahko agenta kot posameznika obravnavamo kot omejenega, saj je sposoben izvajanja zgolj preprostih akcij, pa večagentni sistem, kjer ti delujejo kot celota, lahko kaže samoorganizacijsko vedenje in izkazujejo določeno vrsto kolektivne inteligence (Yang, 2014). Čebele se, na primer, med seboj usklajujejo pri iskanju hrane. Pri signaliziranju, kje so najboljši viri hrane (Karaboga, 2005), uporabljajo poseben ples s kriljenjem (angl. waggle dance). Drugi primer

Slika 2: Kronološki pregled popularnih algoritmov, ki so nastali med 2001 do 2010



predstavljajo mravlje, ki med nošenjem hrane v mravljišče odlagajo feromon, ki privlači druge mravlje, da sledijo potem poti z največ feromona (Dorigo, Bonabeau & Theraulaz, 2000).

Tabela 2 povzema nekatere trenutno najbolj popularne algoritme po vzorih iz narave s pripadajočimi navdihi, ki so služili za njihov razvoj.

Tabela 2: Najbolj popularni algoritmi in njihovi navdihi

Algoritem	Navdih	Referenca
ABC	Poseben ples s kriljenjem	Karaboga & Basturk (2007)
ACO	Odlaganje feromona	Dorigo & Di Caro (1999)
BA	Eholokacija netopirjev	Yang (2010*b*)
CS	Parazitizem pri kukavicah	Yang & Deb (2009)
EA	Darwinova naravna selekcija	Eiben & Smith (2003)
FA	Parjenje in odganjanje vsiljivcev	Yang (2010*a*)

Kot lahko povzamemo iz Tabele 2, večina algoritmov po vzorih iz narave posnema naravne ali biološke sisteme (Fister Jr. et al., 2013). Ugotovimo lahko, da so nekateri algoritmi navdahnjeni tudi s posnemanjem fizikalnih (Biswas et al., 2013) in družbenih pojavov (Ahmadi-Javid, 2011) ali celo pojavi, ki nastopajo v športu (Osaba, Diaz & Onieva, 2014). Po letu 2000 se je število novih algoritmov drastično povečalo. Slika 2 predstavlja kronološki pregled nekaterih bolj znanih algoritmov, ki so nastali v obdobju od leta 2001 do leta 2010. Za podrobnejši opis teh algoritmov usmerjamo bralca na pregledni članek Brezočnik, Fister Jr. & Podgorelec (2018).

Za konec je potrebno pripomniti, da ima celotno raziskovalno področje algoritmov po vzorih iz narave tudi nekaj temnih trenutkov. Številni raziskovalci namreč dvomijo v edinstvenost in resnično znanstveno vrednost novo nastalih algoritmov, npr. Sörensen (2015) oz. Fister Jr. et al. (2016). Nekateri članki so tako že pokazali, da je veliko avtorjev sintetično definiralo t. i. "nove algoritme", temelječe

na vzorih iz narave in so jih uspeli objaviti v različnih strokovnih publikacijah. Večina teh avtorjev, ki želi s tem pritegniti pozornost strokovne javnosti, se v teh publikacijah posveča vzoru iz narave bolj kakor samemu opisu notranjih komponent algoritma in njegovemu delovanju. Rezultat tega je, da ti novi algoritmi ne prinašajo nič bistveno novega, ampak so samo boljša oz. slabša kopija že znanih obstoječih (Fong et al., 2016). V splošnem lahko rečemo, da je ta proces objavljanja novih algoritmov po vzorih iz narave skoraj vsak dan podoben poskusom ponovnega odkrivanja kolesa (angl. reinventing the wheel).

5.1.1 Optimizacija z roji delcev

Optimizacija z roji delcev (angl. Particle Swarm Optimization, krajše PSO) je eden prvih predstavnikov algoritmov inteligence roja delcev, ki sta ga razvila Kennedy & Eberhart (1995). Navdih za algoritem predstavlja obnašanje jat ptic ali rib. Populacija v omenjenem algoritmu je sestavljena iz NP delcev, kjer vsak delec predstavlja rešitev zadanega problema. Vsako rešitev lahko predstavimo kot vektor $\mathbf{x}_i = \{x_{i,j}\}$ za $j = 1, \ldots, D$, katerega elementi so realna števila $x_{i,j} \in \mathcal{R}$, kjer D določa dimenzijo problema (angl. dimension of the problem).

Algoritem PSO vzdržuje tudi populacijo najboljših lokalnih rešitev $\mathbf{p}_i^{(t)}$ za $i = 1, \ldots, NP$. Najboljša rešitev v populaciji $\mathbf{g}^{(t)}$ se določi po vsaki generaciji, medtem ko premikanje delcev v prostoru preiskovanja izvedemo po naslednji formuli:

$$\begin{aligned} \mathbf{v}_i^{(t+1)} &= \mathbf{v}_i^{(t)} + C_1 U(0,1)(\mathbf{p}_i^{(t)} - \mathbf{x}_i^{(t)}) + C_2 U(0,1)(\mathbf{g}^{(t)} - \mathbf{x}_i^{(t)}), \\ \mathbf{x}_i^{(t+1)} &= \mathbf{x}_i^{(t)} + \mathbf{v}_i^{(t+1)}, \end{aligned} \tag{1}$$

kjer faktorja C_1 in C_2 označujeta kognitivno in socialno komponento in $U(0,1)$ označuje naključno število izbrano iz uniformne distribucije v intervalu $[0,1]$ (angl. uniform distribution).

Celotni psevdokod Fister Jr. (2017) algoritma PSO je predstavljen v algoritmu 1.

Več informacij o algoritmu PSO lahko bralec najde v naslednjih publikacijah: Zhang, Wang & Ji (2015), in Kulkarni & Venayagamoorthy (2010).

Algorithm 1 Osnovni algoritem PSO

Vhod: Populacije delcev $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,n})^T$ for $i = 1, \ldots, NP$, MAX_FE.
Izhod: Najboljša rešitev $\mathbf{g}$ in njena vrednost $f_{min} = \min_{i=1,\ldots,NP}(f(\mathbf{x_i}))$.

```
1: init_particles;
2: evals = 0;    // števec števila ocenitev
3: while ustavitveni_pogoj_ni_dosežen do
4:     for i = 1 to NP do
5:         f_i = oceni_novo_rešitev(x_i^(t));
6:         evals = evals + 1;
7:         // shrani lokalno najboljšo rešitev
8:         if f_i ≤ pBest_i then
9:             p_i^(t) = x_i^(t); pBest_i^(t) = f_i;
10:        end if
11:        // shrani globalno najboljšo rešitev
12:        if f_i ≤ f_min then
13:            g^(t) = x_i^(t); f_min = f_i;
14:        end if
15:        x_i^(t) = generaj_novo_rešitev(x_i^(t));
16:    end for
17: end while
```

5.1.2 Diferencialna evolucija

Diferencialna evolucija (angl. Differential Evolution, krajše DE) je evolucijski algoritem, ki sta ga razvila Storn & Price (1997). Algoritem sestoji iz posameznikov, ki so predstavljeni kot vektorji realnih števil:

$$\mathbf{x}_i^{(t)} = (x_{i,1}^{(t)}, \ldots, x_{i,D}^{(t)}), \quad \text{za } i = 1, \ldots, NP, \tag{2}$$

kjer D označuje dimenzijo problema, NP število posameznikov v populaciji in t je števec generacij.

V diferencialni evoluciji poznamo tri glavne operatorje (Fister Jr., 2017):

- mutacija,
- križanje in
- selekcija.

Mutacijo lahko matematično predstavimo z naslednjo enačbo:

$$\mathbf{u}_i^{(t)} = \mathbf{x}_{r1}^{(t)} + F \cdot (\mathbf{x}_{r2}^{(t)} - \mathbf{x}_{r3}^{(t)}), \quad \text{za } i = 1, \ldots, NP, \tag{3}$$

kjer $\mathbf{u}_i^{(t)}$ označuje mutirani vektor, $F \in [0.1, 1.0]$ je skalirni faktor, in so $r1$, $r2$ in $r3$ naključno generirane vrednosti izbrane iz uniformne porazdelitve v intervalu $1, \ldots, NP$. Pri prikazani mutaciji izberemo dve rešitvi naključno, njuna skalirana

razlika pa se doda poskusni rešitvi.

Križanje v diferencialni evoluciji lahko predstavimo z naslednjo enačbo:

$$w_{i,j} = \begin{cases} u_{i,j}^{(t)} & \text{rand}_j(0,1) \leq CR \vee j = j_{rand}, \\ x_{i,j}^{(t)} & \text{drugače}, \end{cases} \tag{4}$$

kjer $CR \in [0.0, 1.0]$ označuje verjetnost križanja. Relacija $j = j_{rand}$ zagotavlja, da se poskusni vektor razlikuje od originalnega $\mathbf{x}_i^{(t)}$ v vsaj enem elementu.

Operator selekcije matematično predstavimo kot:

$$\mathbf{x}_i^{(t+1)} = \begin{cases} \mathbf{w}_i^{(t)} & \text{če } f(\mathbf{w}_i^{(t)}) \leq f(\mathbf{x}_i^{(t)}), \\ \mathbf{x}_i^{(t)} & \text{drugače}, \end{cases} \tag{5}$$

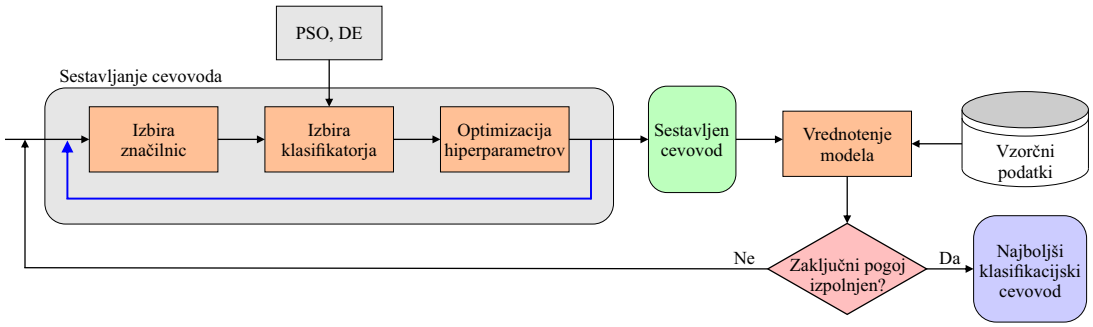
kjer poskusni vektor zamenja originalnega samo v primeru, da je vrednost njegove funkcije uspešnosti večja od vrednosti funkcije uspešnosti originalnega vektorja (tj. $f(\mathbf{w}_i^{(t)}) \leq f(\mathbf{x}_i^{(t)})$).

Za več informacij o algoritmu diferencialne evolucije vabimo bralca, da pogleda naslednji publikaciji: Das & Suganthan (2011) in Das, Mullick & Suganthan (2016).

6 Predlagana rešitev NiaAML

NiaAML je nova metoda za avtomatsko načrtovanje (sestavljanje) in vrednotenje klasifikacijskih cevovodov. Pri metodi NiaAML načrtujemo klasifikacijske cevovode s pomočjo različnih algoritmov po vzorih iz narave. Glavna značilnost NiaAML je, da so posamezniki pri algoritmih po vzoru iz narave predstavljeni z vektorji realnih števil, medtem ko so v sorodnih rešitvah ponavadi zapisani v strukturah dreves. Sama metoda (Slika 3) je zelo kompleksna in sestoji iz dveh

Slika 3: Celotni ekosistem metode NiaAML.



glavnih faz, ki se delita na več podkomponent. Dve glavni fazi sta sledeči:

1. sestavljanje (kompozicija) klasifikacijskega cevovoda, in
2. vrednotenje klasifikacijskega cevovoda.

Rezultat prve faze je klasifikacijski cevovod, ki ga moramo ovrednotiti v drugi fazi vrednotenja modela. Cilj druge faze je torej ocenitev kakovosti sestavljenega modela na izbranih testnih podatkih.

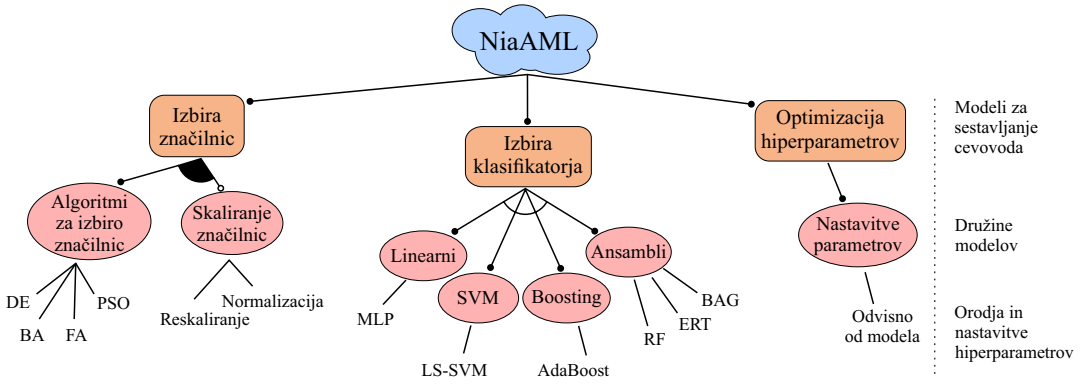
Sestavljanje klasifikacijskega cevovoda se deli na sledeče tri korake:

- izbiro algoritma za izbiro značilnic,
- izbiro klasifikatorja,
- optimizacijo hiperparametrov.

Prvi korak je namenjen izbiranju algoritma, s katerim se bo izvedel proces izbire značilnic podatkovne baze (angl. feature selection) (Guyon & Elisseeff, 2003). Drugi korak se osredotoča na izbiranje klasifikatorja, medtem ko služi tretji kot orodje za optimizacijo hiperparametrov izbranega klasifikatorja.

Arhitektura metode NiaAML je predstavljena na Sliki 4. Ta je shematsko prikazana s pomočjo diagramov značilnosti (angl. feature diagrams, krajše FD), kjer so zajete vse možne konfiguracije klasifikacijskega cevovoda. Pravzaprav je diagram značilnosti drevo, ki je sestavljeno iz vozlišč in povezav. Vozlišča predstavljajo modele pri sestavljanju klasifikacijskih cevovodov, medtem ko povezave določajo relacije med njimi. Vozlišča so bodisi obvezna ali neobvezna, pri čemer so prva označena s polnimi točkami, slednja pa z odprtimi točkami.

Slika 4: Diagram značilnosti metode NiaAML.



Kot je razvidno iz Slike 4, celotni klasifikacijski cevovod sestoji iz treh konceptualnih nivojev:

- modelov za sestavljanje cevovoda (značilnosti v diagramu značilnosti),
- družin modelov (pod-značilnosti v diagramu značilnosti),
- orodij in nastavitev hiperparametrov (atributi v diagramu značilnosti).

Zanimivo je, da je vsaka značilnost ali pod-značilnost določena z lastno množico atributov, ki so med seboj povezani z različnimi relacijami. V diagramu značilnosti obstajajo tri različne relacije: „in“, „eden od“ in „več“. Relacija „eden od“ je označena z odprtim polkrogom, ki s povezavami povezuje značilnosti z njihovimi pod-značilnostmi, „več“ z zaprtim polkrogom, medtem ko je razmerje „in“ brez polkroga. Pomen teh relacij pojasnjujemo hkrati s podrobnim opisom diagrama značilnosti v nadaljevanju.

Sestavljanje klasifikacijskega cevovoda z metodo NiaAML sestoji iz treh obveznih nalog, ki vključujejo modeliranje: izbire značilnic, izbire klasifikatorja in optimizacije hiperparametrov. Vse te naloge so povezane z relacijo „in“. To pomeni,

da morajo biti vse te naloge vključene v generiranem klasifikacijskem cevovodu. Modeliranje izbire značilnic sestoji iz modeliranja:

- algoritma za izbiro značilnic ter
- skaliranja značilnic.

Obe podfunkciji v diagramu značilnosti sta povezani z relacijo „več“. Ker je modeliranje skaliranja značilnic neobvezno, ta relacija pomeni, da se lahko modeliranje izbire značilnic izvede s skaliranjem ali brez. Algoritem za izbiro značilnic mora biti obvezno izbran. Klasifikacijsko metodo lahko modeliramo iz štirih družin, kot sledi: linear, SVM, boosting in decision tree. Te podznačilnosti so v FD med seboj povezane z relacijo „eden od“, kar pomeni, da mora biti izbrana ena družina v kompoziciji cevovoda. Najbolj specifična pa je optimizacija hiperparametrov, ki je odvisna od izbranega algoritma za izbiro značilnic, kot tudi od izbranega klasifikatorja.

Najnižji nivo v diagramu značilnosti predstavlja potencialne množice orodij in nastavitev hiperparametrov. Sestavljena je iz algoritmov za izbiro skaliranja značilnic, metod klasifikacije in optimizacije nastavitev hiperparametrov. Vsi okvirni elementi so izbrani glede na ustrezne družine modelov.

V nadaljevanju natančno predstavimo dve glavni fazi metode NiaAML, tj. sestavljanje klasifikacijskega cevovoda in njegovo vrednotenje.

6.1 Sestavljanje cevovoda

Kot smo že omenili v prejšnjih poglavjih, lahko za sestavljanje cevovoda v NiaAML uporabimo kateregakoli izmed algoritmov po vzorih iz narave, kjer so posamezniki predstavljeni kot vektorji realnih števil. Pri večini algoritmov ni potrebno prilagoditi nobenih operatorjev, temveč je potrebno definirati le preslikavo genotip-fenotip in definirati funkcijo uspešnosti. Obe komponenti sta opisani v nadaljevanju.

Tabela 3: Preslikava genotip-fenotip.

Značilnice	Elementi vektorja	Nabor funkcij
Izbira značilnic	$x_{i,1}^{(t)}$	$\{DE, PSO, GWO, BA\}$
Skaliranje značilnic	$x_{i,2}^{(t)}$	$\{No, Rescaling, Normalizacija\}$
Klasifikacija	$x_{i,3}^{(t)}$	$\{MLP, LS\_SVM, AdaBoost, RF, ERT, BAG\}$
Optimizacija hiperparametrov	$x_{i,4}^{(t)}, \ldots, x_{i,D}^{(t)}$	Odvisno od modela

6.1.1 Predstavitev posameznikov

V NiaAML so posamezniki algoritma za sestavljanje klasifikacijskega cevovoda predstavljeni kot vektorji realnih števil:

$$\mathbf{x}_i^{(t)} = \{ \underbrace{x_{i,1}^{(t)}}_{\text{Izbira značilnic}}, \underbrace{x_{i,2}^{(t)}}_{\text{Skaliranje značilnic}}, \underbrace{x_{i,3}^{(t)}}_{\text{Klasifikacija}}, \underbrace{x_{i,4}^{(t)}, \ldots, x_{i,k}^{(t)}, x_{i,k+1}^{(t)}, \ldots, x_{i,D}^{(t)}}_{\text{Optimizacija hiperparametrov}} \}, \quad (6)$$

kjer vsak element posameznika $x_{i,j}^{(t)}$ preslikamo v atribut iz pripadajočega nabora virov glede na ustrezno preslikavo genotip-fenotip, prikazano v Tabeli 3.

Tabela prikazuje preslikavo elementov vektorja v značilnice in atribute ustreznega nabora virov. Prve tri elemente vektorja preslikamo v atribute po naslednji enačbi:

$$attr_{feat}^{(t)} = \left\lfloor \frac{x_{i,j}^{(t)}}{|attr_{feat}|} \right\rfloor, \quad \text{for } j = 1, \ldots, 3, \quad (7)$$

kjer $attr_{feat}^{(t)}$ označuje specifične atribute značilnic in je $|attr_{feat}|$ velikost množice značilnic $feat \in \{$Izbira značilnic, Skaliranje značilnic, Klasifikacija$\}$.

Preostali elementi vektorja predstavljajo absolutne vrednosti pripadajočih hiperparametrov. V trenutni inačici NiaAML je maksimalno število elementov fiksno, čeprav je število uporabljenih elementov odvisno od izbranega klasifikatorja in je zato variabilno. Domene določenih hiperparametrov smo v tej inačici NiaAML določili glede na izkušnje pridobljene prek eksperimentalnega dela.

Vrednosti hiperparametrov

Iskanje optimalnih nastavitev hiperparametrov je v ogrodju NiaAML del optimizacijskega procesa. Optimizacija hiperparametrov se v NiaAML zgodi v fazi sestavljanja cevovoda. Zaradi tega se cel postopek načrtovanje cevovoda zelo

Tabela 4: Zaloge vrednosti hiperparametrov.

Alg.	Hiperparameter	Zaloge vrednosti
DE	F, CR	$F \in [0.5, 0.9], CR \in [0.0, 1.0]$
PSO	C_1, C_2	$C_1 \in [1.5, 2.5], C_2 \in [1.5, 2.5]$
GWO	a	$a \in [0.0, 2.0]$
BA	$A, r, Q_{\min}, Q_{\max}$	$A \in [0.5, 1.0], r \in [0.0, 0.5], Q_{\min} \in [0.0, 1.0], Q_{\max} \in [1.0, 2.0],$
MLP	act, sol, lr	$act \in \{identity, logistic, tanh, relu\}$, $sol \in \{lbfgs, sgd, adam\}$ $lr \in \{constant, invscaling, adaptive\}$
SVM	$gamma, c$	$gamma \in [0.1, 100], c \in [0.1, 100]$
ADA	n_estim, alg	$n\_estim = [10, 110]$, $alg \in \{samme, samme.r\}$
RF	n_estim	$n\_estim \in [10, 110]$
ERT	n_estim	$n\_estim \in [10, 110]$
BAG	n_estim	$n\_estim \in [10, 110]$

pohitri glede na tradicionalne rešitve za načrtovanje klasifikacijskih cevovodov.

V sklopu naše študije v fazi skaliranja značilnic operiramo s ponovnim skaliranjem (angl. feature rescaling) in normalizacijo, podpiramo štiri algoritme za izbiro značilnic in šest različnih klasifikacijskih metod. Skaliranje značilnic in normalizacija nimata posebnih parametrov. V vlogi algoritmov za izbiro značilnic nastopajo naslednji algoritmi po vzorih iz narave[1]: DE, PSO, Grey Wolf Optimizer (GWO) Mirjalili, Mirjalili & Lewis (2014) in Bat Algorithm (BA) Yang (2010*b*). Klasifikatorji so pa lahko naslednji: MultiLayer Perceptron (MLP) Rosenblatt (1961), Least Square SVM Suykens & Vandewalle (1999), AdaBoost Schapire (1999), Random Forest (RF) Breiman (2001), Extremely Randomized Trees (ERT) Geurts, Ernst & Wehenkel (2006), and Bagging Breiman (1996). Tabela 4 prikazuje zaloge vrednosti za določene algoritme.

Maksimalno število parametrov izbranih algoritmov in klasifikacijskih metod je sedem. Zatorej je velikost vektorja, ki je predstavljen z realnimi števili za sestavljanje klasifikacijskih cevovodov, 10.

6.1.2 Definicija funkcije uspešnosti

Za izračun funkcije uspešnosti uporabimo 10-kratno prečno preverjanje (angl. 10-fold cross validation). V tem primeru so podatki razdeljeni na $k = 10$ enakih delov s pomočjo stratificiranega vzorčenja (angl. stratified sampling), kjer je vsak od k delov klasificiran v klasifikacijskem cevovodu. Glavna prednost tega pristopa je, da imajo vse testne množice 80 odstotkov skupnih podatkov, kadar je $k = 10$.

[1]naslednjih metod in algoritmov na tem delu ne prevajamo v slovenščino

Posledično je kompromis med pristranskostjo in deli variance napovedane napake minimiziran zaradi zmanjšanja obeh, kolikor je to mogoče.

Uspešnost klasifikacije ocenimo z merilom točnosti (*Accuracy*), ki ga matematično zapišemo kot:

$$Accuracy(M(\mathbf{x}_i)) = \frac{TP + TN}{TP + TN + FP + FN}, \tag{8}$$

kjer $M(\mathbf{x}_i)$ označuje model, zgrajen na podlagi sestavljenega klasifikacijskega cevovoda $\mathbf{x}_i$, TP = True Positive, TN = True Negative, FP = False Positive, in FN = False Negative [2].

Funkcijo uspešnosti definiramo na naslednji način:

$$f(\mathbf{x}_i) = 1 - \frac{1}{k}\sum_{i=1}^{k} Accuracy(M(\mathbf{x}_i)), \tag{9}$$

kjer $Accuracy(M(\mathbf{x}_i))$ izračunamo glede na enačbo (8). Poudarimo, da funkcija uspešnosti ovrednoti povprečno uspešnost klasifikacijskih metod, pridobljenih skozi 10-kratno prečno preverjanje. Naloga optimizacije je minimizacija funkcije uspešnosti.

6.2 Vrednotenje modela/cevovoda

V tem koraku ocenimo kakovost sestavljenega klasifikacijskega cevovoda, ki ga sestavljajo izbrane značilnice, izbran klasifikator in pripadajoči hiperparametri. Običajno se učinkovitost klasifikacijske metode ocenjuje z uporabo modela za nevidene testne podatke (angl. unseen test data). Pri tem smo uporabili standardno validacijo 80 %–20 %, kjer je 80 % primerkov uporabljenih za trening, medtem ko je 20 % primerkov za testiranje. Učinkovitost klasifikacije nato ocenimo glede na izraženo z merilom točnosti, kot ga prikazuje enačba (8).

[2] TP = resnično pozitivni, FP = lažno pozitivni, TN = resnično negativni, FN = lažno negativni

7 Eksperimenti in rezultati

Glavni cilji našega raziskovalnega dela so zaobjemali implementacijo metode NiaAML v programskem jeziku Python, ter ovrednotenje metode na realnih podatkovnih zbirkah in s tem tudi prikaz praktične vrednosti metode NiaAML.

V nadaljevanju tega poglavja prikažemo vrednotenje metode NiaAML na realnih bioinformatskih podatkih omenjenih v podpoglavju 4.3.

7.1 Implementacija metode NiaAML

Metoda NiaAML je v celoti implementirana v programskem jeziku Python. NiaAML uporablja le dve zunanji knjižnici:

- NiaPy (Vrbančič et al., 2018), kjer so implementirani vsi algoritmi po vzorih iz narave, ki jih uporabljamo v tem delu in
- scikit-learn (Pedregosa et al., 2011), kjer so implementirane vse klasifikacijske metode, ki jih uporabljamo v tem delu.

7.2 Nastavitve parametrov

Pri sestavljanju klasifikacijskega cevovoda smo se osredotočili na izbiranje algoritmov za izbiro in skaliranje značilnic, izbiranje klasifikacijske metode in optimizacijo hiperparametrov izbrane metode. Medtem ko pri skaliranju značilnic uporabljamo tradicionalne brezparametrske algoritme, optimalne nastavitve hiperparametrov klasifikacijskih metod pa iščemo z algoritmom za načrtovanje klasifikacijskih cevovodov, za izbiranje značilnic uporabljamo enega izmed stohastičnih populacijskih algoritmov po vzorih iz narave, omenjenega v poglavju 5.

Pri tem smo nastavitve dveh parametrov stohastičnih populacijskih algoritmov po vzorih iz narave za izbiro značilnic, ki sta skupna vsem, fiksirali na naslednje vrednosti: velikost populacije $NP = 15$ in število ovrednotenj funkcije uspešnosti $nFES = 400$. Nastavitve preostalih parametrov za omenjene algoritme so prikazani v Tabeli 5.

Pri algoritmih za sestavljanje klasifikacijskega cevovoda (PSO in DE) smo število

Tabela 5: Nastavitve nadzornih parametrov.

Algoritem	Okrajšava	Parameter 1	Parameter 2	Parameter 3	Parameter 4
Differential Evolution	DE	$F = 0.5$	$CR = 0.9$		
Grey Wolf Optimizer	GWO				
Particle Swarm Algorithm	PSO	$C_1 = 2.0$	$C_2 = 2.0$	$w = 0.7$	$v \in [-4, 4]$
Bat Algorithm	BA	$A = 0.5$	$r = 0.5$	$Q \in [0.0, 2.0]$	

ovrednotenj nastavili na $nFES = 1000$, velikost populacije pa na $Np = 20$. Nastavitve preostalih parametrov so enake kot pri algoritmih za izbiranje značilnic (Tabela 5).

7.3 Metrike

Kakovost sestavljenih cevovodov smo ocenjevali glede na tri statistične metrike:

- *Accuracy* (enačba 8),
- Cohenova kappa κ, in
- F_1-score.

Cohenova kappa κ predstavlja metriko za delo z multivariantnimi problemi in jo definiramo na naslednji način (Cohen, 1960):

$$\kappa(M(\mathbf{x}_i)) = \frac{n \sum_{i=1}^{k} CM_{ii} - \sum_{i=1}^{k} CM_{i.} CM_{.i}}{n^2 - \sum_{i=1}^{k} CM_{i.} CM_{.i}}, \tag{10}$$

kjer je $CM_{i.}$ seštevek elementov i-te vrstice matrike zmede CM in $CM_{.i}$ seštevek elementov i-tega stolpca CM Stephen (1997).

F_1-score je izražena na naslednji način:

$$F_1(M(\mathbf{x})) = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \tag{11}$$

kjer je *recall* definiran kot:

$$recall(M(\mathbf{x}_i)) = \frac{TP}{TP + FN} \tag{12}$$

in *precision* kot:

$$precision(M(\mathbf{x}_i)) = \frac{TP}{TP + FP}. \tag{13}$$

Pomembno je omeniti, da lahko *precision*, *recall* in F_1 določimo samo za binarno

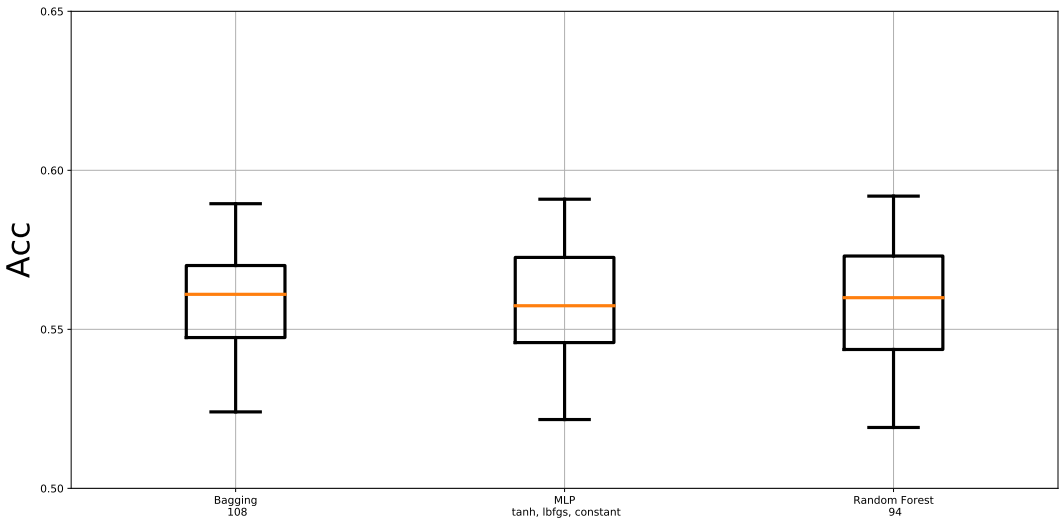
klasifikacijo. Z uporabo knjižnice *sklearn* pa lahko izračunamo uteženo povprečje med številnimi binarnimi klasifikacijskimi nalogami in s tem uporabimo te meritve tudi pri večrazredni klasifikaciji.

7.4 Rezultati klasificiranja zbirke Abalone

Rezultate 10-kratnega prečnega preverjanja pri klasifikaciji zbirke Abalone predstavljamo vizualno s t. i. grafom škatla z brki (angl. boxplot), kjer s svetlo-oranžno linijo označimo mediano, s črno pa porazdelitev standardnega odklona. Višja kot je vrednost mediane ali srednje vrednosti, tem boljši je klasifikacijski cevovod. Črne pike, ki se pojavijo na določenih slikah, predstavljajo osamelce (angl. outlier).

Slika 5 prikazuje rezultate 10-kratnega prečnega preverjanja za tri najboljše klasifikacijske cevovode, ki smo ga sestavili s pomočjo algoritma DE (Tabela 6). Najboljši rezultat smo dosegli s klasifikacijskim cevovodom, ki je uporabljal klasifikacijsko metodo "bagging" s številom ocenjevalcev 108. Drugi najboljši rezultat smo dosegli s klasifikacijskim cevovodom, ki je uporabljal metodo večnivojske nevronske mreže, medtem ko tretjega z metodo naključnih gozdov.

Slika 5: Primerjava rezultatov treh najboljših klasifikacijskih cevovodov, sestavljenih z algoritmom DE na zbirki Abalone grafično.



Pri uporabi algoritma za sestavljanje cevovodov PSO so rezultati nekoliko različni

Tabela 6: Primerjava rezultatov treh najboljših klasifikacijskih cevovodov, sestavljenih z algoritmom DE na zbirki Abalone analitično.

Mesto	Izbira značilnic	Skaliranje	Klasifikacijska metoda	Točnost
1	[8/8]	n/a	Bagging [108]	0.5592
2	[6/8]	n/a	MLP ['tanh', 'lbfgs', 'constant']	0.5583
3	[8/8]	n/a	Random forest [94]	0.5581

(Slika 6), saj smo tukaj najboljši klasifikacijski cevovod dobili z uporabo klasifikacijske metode ERT, medtem ko z uporabo klasifikacijske metode "bagging" dosežemo drugo mesto. Tretji najboljši cevovod dobimo pri uporabi klasifikacijskega cevovoda, ki uporablja klasifikacijsko metodo ERT (Tabela 7).

Slika 6: Primerjava rezultatov treh najboljših klasifikacijskih cevovodov, sestavljenih z algoritmom PSO na zbirki Abalone grafično.

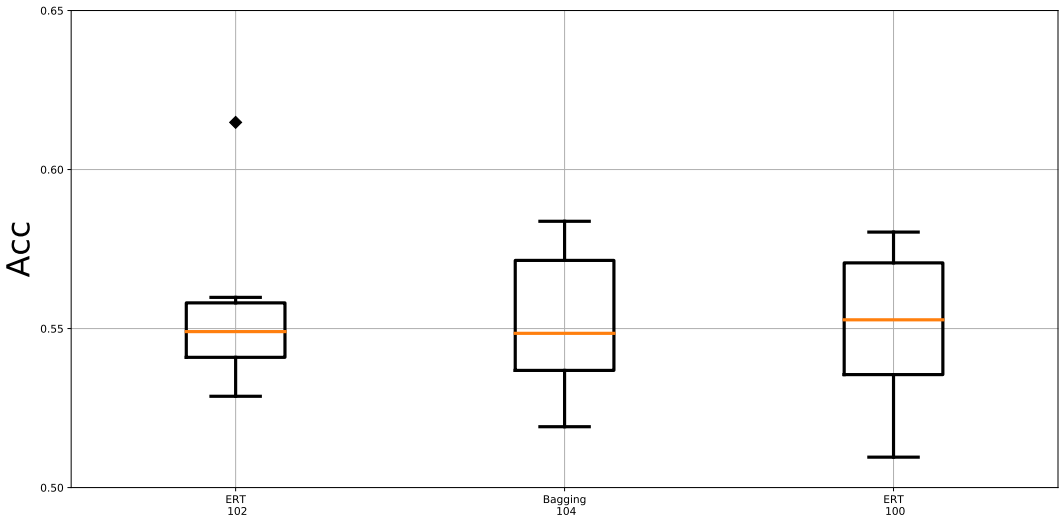


Tabela 7: Primerjava rezultatov treh najboljših klasifikacijskih cevovodov, sestavljenih z algoritmom PSO na zbirki Abalone analitično.

Mesto	Izbira značilnic	Skaliranje	Klasifikacijska metoda	Točnost
1	[8/8]	n/a	ERT [102]	0.5533
2	[8/8]	n/a	Bagging [104]	0.5520
3	[8/8]	n/a	ERT [100]	0.5516

7.5 Rezultati klasificiranja zbirke Ecoli

Ecoli je med vsemi tremi podatkovnimi zbirkami najenostavnejša za klasifikacijo. Razlog za to leži v številu primerkov, ki jih je v tej podatkovni zbirki zelo malo.

Zaradi tega ni presenetljivo, da je kot pri sestavljanju klasifikacijskih cevovodov z algoritmov DE (Slika 7) kot tudi z algoritmom PSO (Slika 8) dosegel najboljše rezultate z uporabo klasifikacijske metode naključnih gozdov (Tabeli 8 in 9).

Slika 7: Primerjava rezultatov treh najboljših klasifikacijskih cevovodov, sestavljenih z algoritmom DE na zbirki Ecoli grafično.

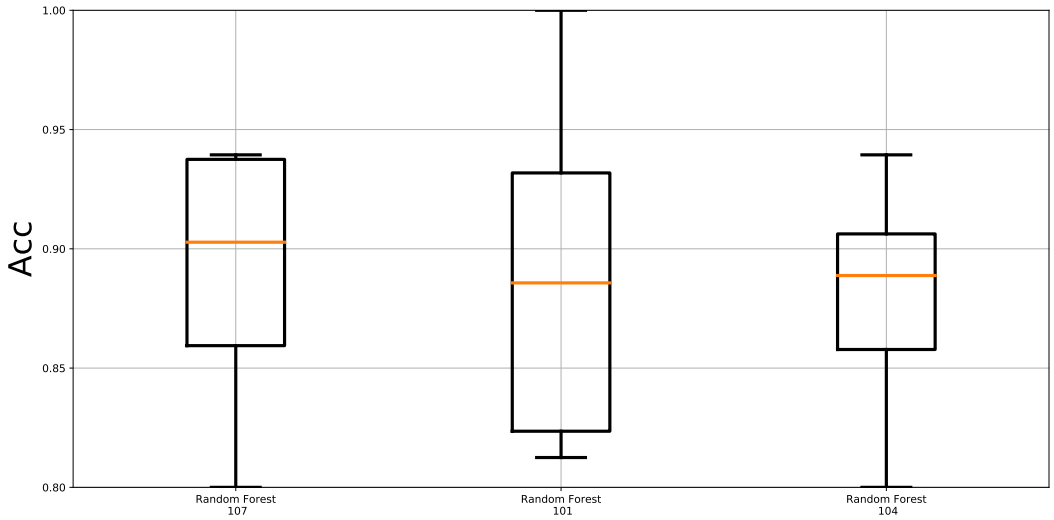


Tabela 8: Primerjava rezultatov treh najboljših klasifikacijskih cevovodov, sestavljenih z algoritmom DE na zbirki Ecoli analitično.

Mesto	Izbira značilnic	Skaliranje	Klasifikacijska metoda	Točnost
1	[6/8]	n/a	Random forest [107]	0.8934
2	[7/8]	n/a	Random forest [101]	0.8873
3	[7/8]	n/a	Random forest [104]	0.8845

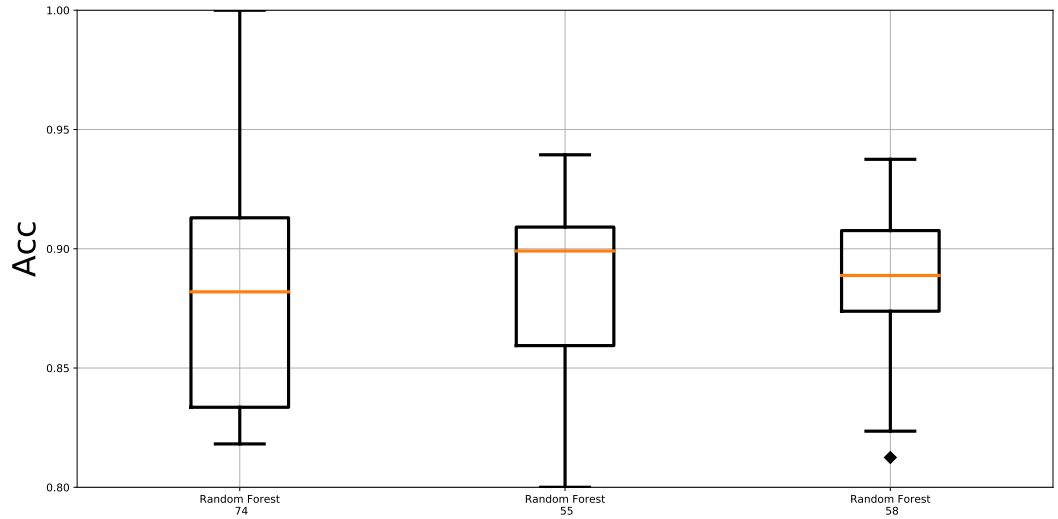
Tabela 9: Primerjava rezultatov treh najboljših klasifikacijskih cevovodov, sestavljenih z algoritmom PSO na zbirki Ecoli analitično.

Mesto	Izbira značilnic	Skaliranje	Klasifikacijska metoda	Točnost
1	[7/8]	n/a	Random forest [74]	0.8845
2	[6/8]	n/a	Random forest [55]	0.8840
3	[7/8]	n/a	Random forest [58]	0.8836

7.6 Rezultati klasificiranja zbirke Yeast

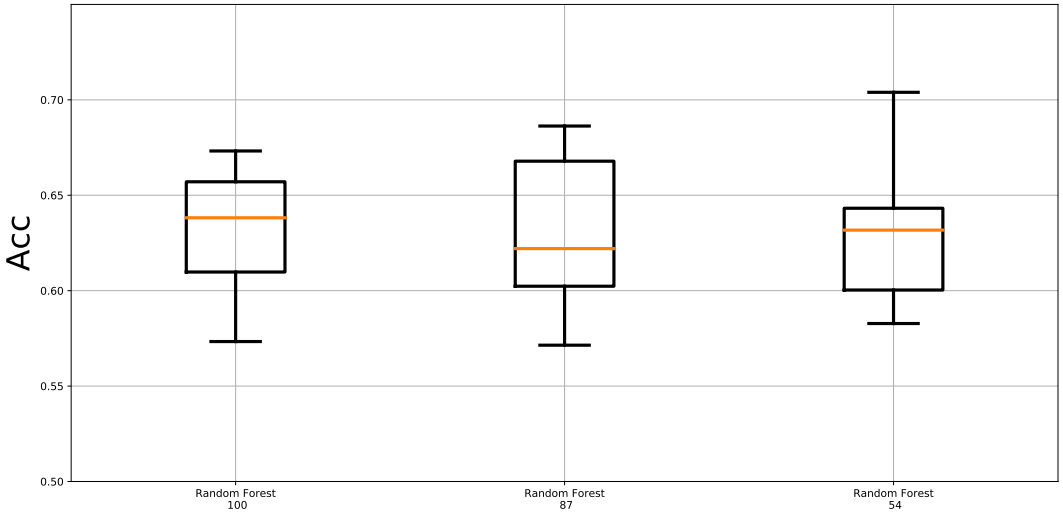
Tudi Yeast spada med enostavnejše podatkovne zbirke, saj vsebuje majhno število primerkov. Zaradi tega je bilo pričakovano, da bo tudi na tej zbirki dominiral RF (Slika 9 in Slika 10). Podrobnejše rezultate klasifikacije prikazujeta Tabeli 10 in 11. Kot vidimo, so vse točnosti prvih treh cevovodov zelo podobne. Kot

Slika 8: Primerjava rezultatov treh najboljših klasifikacijskih cevovodov, sestavljenih z algoritmom PSO na zbirki Ecoli grafično.



zanimivost je vredno omeniti, da algoritmi za izbiro značilnic pri sestavljanju cevovoda niso izločili nobene značilnice.

Slika 9: Primerjava rezultatov treh najboljših klasifikacijskih cevovodov, sestavljenih z algoritmom DE na zbirki Yeast grafično.



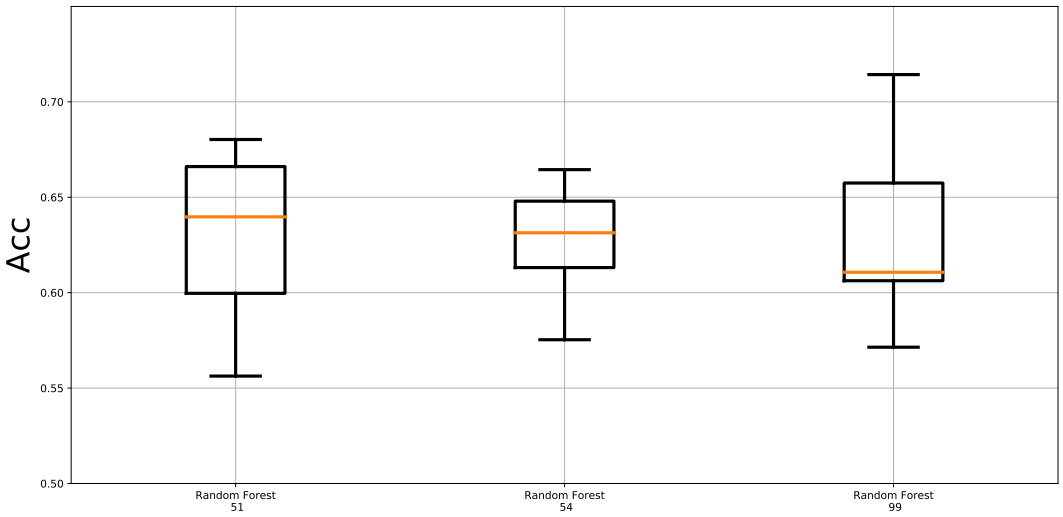
7.7 Validacija

Rezultate validacije opravimo z vzorčenjem po načelu 80 %–20 %, kjer 80 % vzorcev uporabimo za trening, preostalih 20 % pa za validacijo. Za validacijo upora-

Tabela 10: Primerjava rezultatov treh najboljših klasifikacijskih cevovodov, sestavljenih z algoritmom DE na zbirki Yeast analitično.

Mesto	Izbira značilnic	Skaliranje	Klasifikacijska metoda	Točnost
1	[8/8]	n/a	Random forest [100]	0.6314
2	[8/8]	n/a	Random forest [87]	0.6313
3	[8/8]	n/a	Random forest [54]	0.6285

Slika 10: Primerjava rezultatov treh najboljših klasifikacijskih cevovodov, sestavljenih z algoritmom PSO na zbirki Yeast grafično.



bimo konfiguracijo najboljšega klasifikacijskega cevovoda dobljenega po prečnem preverjanju.

Rezultate validacije prikazujemo tabelarično. Rezultati algoritma za sestavljanje klasifikacijskih cevovodov DE so prikazani v Tabeli 12, medtem ko rezultati algoritma PSO v Tabeli 13. Opazimo, da je končna točnost klasificiranja podatkovne zbirke Abalone najnižja, saj je slednja najkompleksnejša izmed treh uporabljenih podatkovnih zbirk. Tako je tudi merilo Cohenova kappa κ, ki priča o dostojnem ujemanju napovedi z dejanskimi razredi. Merilo F_1 dosega najmanjšo izmed vrednosti treh podatkovnih zbirk. Opazimo, da je pri vseh merilih algoritem DE nekoliko uspešnejši od algoritma PSO.

Podatkovna zbirka Ecoli je sodeč po rezultatih najpredvidljivejša. To dokazujeta visoka točnost klasificiranja in merilo F_1, kakor tudi zelo ugodno merilo Cohenova kappa κ. Sodeč po rezultatih, v tem primeru ni opaznih neposrednih razlik med

Tabela 11: Primerjava rezultatov treh najboljših klasifikacijskih cevovodov, sestavljenih z algoritmom PSO na zbirki Yeast analitično.

Mesto	Izbira značilnic	Skaliranje	Klasifikacijska metoda	Točnost
1	[8/8]	n/a	Random forest [51]	0.6313
2	[8/8]	n/a	Random forest [54]	0.6286
3	[8/8]	n/a	Random forest [99]	0.6274

Tabela 12: Validacija rezultatov klasifikacijskih cevovodov, sestavljenih z algoritmom DE na treh podatkovnih zbirkah.

	Abalone	Ecoli	Yeast
Točnost	0.5777	0.9117	0.6387
Cohen κ	0.3651	0.8718	0.5254
F1	0.5752	0.9137	0.6344

algoritmoma DE in PSO.

Rezultati podatkovne zbirke Yeast nakazujejo na zmerno ujemanje predikcije in dejanske klasifikacije. Merilo F_1 je tudi tokrat podobno vrednosti merila točnosti *Accuracy*. Opazimo, da sta obe merili nekoliko boljši pri algoritmu PSO, medtem ko je po merilu Cohenova kappa κ uspešnejši algoritem DE.

Tabela 13: Validacija rezultatov klasifikacijskih cevovodov, sestavljenih z algoritmom PSO na treh podatkovnih zbirkah.

	Abalone	Ecoli	Yeast
Točnost	0.5765	0.9117	0.6397
Cohen	0.3632	0.8718	0.5240
F1	0.5706	0.9137	0.6344

V splošnem lahko zaključimo, da ni večjih razlik med rezultati klasifikacijskih cevovodov, ki sta jih sestavila algoritma DE in PSO. Zatorej bi bilo potrebno v prihodnosti izvesti več poskusov, da bi lahko pokazali večje razlike med različnimi algoritmi po vzorih iz narave. Prav tako bi bilo potrebno preizkusiti tudi ostale algoritme, ki jih v tej študiji nismo zajeli, kot npr. BA, CS, FA.

7.8 Interpretacija in razprava

V prejšnjih poglavjih smo predstavili praktične rezultate uporabe metode Nia-AML. Ugotovili smo, da je podatkovna zbirka Abalone najzahtevnejša izmed treh uporabljenih v naši študiji, saj sestoji iz največ primerkov. Preostali podatkovni zbirki Ecoli in Yeast spadata glede na področje klasifikacije med enostavnejše. Naš glavni namen eksperimentalnega dela ni bil izboljševati rezultate obstoječih

metod, temveč predlagati novo metodo za avtomatizirano strojno učenje. Kljub temu pa lahko potrdimo, da so rezultati, ki smo jih dobili tekom tega eksperimentalnega dela, enaki oziroma celo boljši kot rezultati, predstavljeni v obstoječi literaturi (Soda & Iannello, 2010; Mohamed, Salleh & Omar, 2012; Zhu, Chen & Xing, 2011).

8 Sklep

V tem magistrskem delu smo predlagali novo metodo za avtomatizirano strojno učenje, poimenovano NiaAML, ki temelji na stohastičnih populacijskih algoritmih po vzorih iz narave. V teoretičnem delu tega magistrskega dela smo predstavili arhitekturo te metode, medtem ko smo v praktičnem delu to metodo implementirali in jo preizkusili na treh različnih podatkovnih zbirkah s področja bioinformatike. Rezultati eksperimentalnega dela potrjujejo:

- Učinkovitost metode NiaAML, saj so rezultati klasifikacije enaki ali celo boljši glede na literaturo.
- NiaAML nudi enostavno dodajanje novih značilnic v procesu sestavljanja cevovoda.
- NiaAML lahko uporabljamo na vseh platformah, ki podpirajo razvoj programske opreme v programskem jeziku Python.
- NiaAML omogoča rokovanje s klasifikacijskimi problemi tudi ljudem, ki po izobrazbi niso programerji, oz. se ne spoznajo na njihovo reševanje.

V tem delu smo hkrati uspešno odgovorili na raziskovalna vprašanja, ki smo si jih zastavili v uvodnem poglavju. Čeprav je to šele začetek širše raziskave tega novega raziskovalnega področja, smo prepričani, da bo magistrsko delo naredilo velik korak k popularizaciji algoritmov po vzoru iz narave za razvoj bioinformatskih cevovodov. Doslej namreč v namene avtomatiziranega strojnega učenja ni bilo uporabljenih veliko stohastičnih algoritmov po vzoru iz narave, kjer so posamezniki predstavljeni kot vektorji realnih števil.

V prihodnosti želimo za metodo NiaAML razviti tudi uporabniški grafični vmesnik (angl. Graphical User Interface, krajše GUI), ki bo potencialnim uporabnikom uporabo klasifikacijskih opravil še dodatno poenostavilo. Prav tako želimo to metodo preizkusiti tudi na podatkovnih zbirkah iz ostalih področij.

Literatura

Ahmadi-Javid, A. 2011. Anarchic Society Optimization: a human-inspired method. In *2011 IEEE Congress of Evolutionary Computation (CEC)*. New Orleans: IEEE pp. 2586–2592.

Back, T. 1996. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. New York: Oxford university press.

Beni, G. 2009. Swarm intelligence. In *Encyclopedia of complexity and systems science*. Berlin: Springer pp. 1–32.

Biswas, A., Mishra K. K., Tiwari S. & Misra A. K. 2013. "Physics-inspired optimization algorithms: a survey." *Journal of Optimization* 2013(438152):1–16.

Breiman, L. 1996. "Bagging predictors." *Machine Learning* 24(2):123–140.

Breiman, L. 2001. "Random forests." *Machine learning* 45(1):5–32.

Brezočnik, L., I. Fister Jr. & V. Podgorelec. 2018. "Swarm intelligence algorithms for feature selection: a review." *Applied Sciences* 8(9):1521.

Chen, S.-H. & T.-W. Kuo. 2002. *Evolutionary computation in economics and finance: a bibliography*. Heidelberg: Physica-Verlag HD pp. 419–455.

Cohen, J. 1960. "A coefficient of agreement for nominal scales." *Educational and Psychological Measurement* 20(1):37–46.

Darwin, C. 1872. *The origin of species: by means of natural selection Or the preservation of favored races in the Struggle for life*. Vol. 1 London: Modern library.

Das, S. & P. N. Suganthan. 2011. "Differential evolution: a survey of the state-of-the-art." *IEEE Transactions on Evolutionary Computation* 15(1):4–31.

Das, S., S. S. Mullick & P. N. Suganthan. 2016. "Recent advances in differential evolution–an updated survey." *Swarm and Evolutionary Computation* 27:1–30.

de Castro, L. N. 2007. "Fundamentals of natural computing: an overview." *Physics of Life Reviews* 4(1):1–36.

de Sá, Alex GC, Walter José GS Pinto, Luiz Otavio VB Oliveira & Gisele L Pappa. 2017. RECIPE: a grammar-based framework for automatically evolving classification pipelines. In *20th European Conference on Genetic Programming*. Springer Amsterdam: pp. 246–261.

Ding, C. & H. Peng. 2005. "Minimum redundancy feature selection from microarray gene expression data." *Journal of Bioinformatics and Computational Biology* 3(2):185–205.

Dorigo, M., E. Bonabeau & G. Theraulaz. 2000. "Ant algorithms and stigmergy." *Future Generation Computer Systems* 16(8):851–871.

Dorigo, M. & G. Di Caro. 1999. Ant colony optimization: a new meta-heuristic. In *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*. Vol. 2 IEEE Washington: pp. 1470–1477.

Eiben, A. E. & J. E. Smith. 2003. *Introduction to evolutionary computing*. Vol. 53 Berlin: Springer.

Feurer, M., A. Klein, K. Eggensperger, J. Springenberg, M. Blum & F. Hutter. 2015. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems 28*, ed. C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama & R. Garnett. Montreal: Curran Associates pp. 2962–2970.

Fister Jr., I. 2017. Algoritmi računske inteligence za razvoj umetnega športnega trenerja PhD thesis University of Maribor, Slovenia.

Fister Jr., I., U. Mlakar, J. Brest & I. Fister. 2016. A new population-based nature-inspired algorithm every month: is the current era coming to the end? In *StuCoSReC: proceedings of the 2016 3rd Student Computer Science Research Conference*. Koper: University of Primorska pp. 33–37.

Fister Jr., I., X.-S. Yang, I. Fister, J. Brest & D. Fister. 2013. "A brief review of Nature-inspired algorithms for Optimization." *Elektrotehniški Vestnik* 80(3):116–122.

Fong, S., X. Wang, Q. Xu, R. Wong, J. Fiaidhi & S. Mohammed. 2016. "Recent advances in metaheuristic algorithms: does the Makara dragon exist?" *The Journal of Supercomputing* 72(10):3764–3786.

García, S., J. Luengo & F. Herrera. 2015. *Data preprocessing in data mining*. Cham: Springer.

Geurts, P., D. Ernst & L. Wehenkel. 2006. "Extremely randomized trees." *Machine Learning* 63(1):3–42.

Gijsbers, P. 2018. Automatic construction of machine learning pipelines. Master's thesis Eindhoven University of Technology.

Glenn, T. C. 2011. "Field guide to next-generation DNA sequencers." *Molecular Ecology Resources* 11(5):759–769.

Guyon, I. & A. Elisseeff. 2003. "An introduction to variable and feature selection." *Journal of Machine Learning Research* 3(Mar):1157–1182.

Jin, H., Q. Song & X. Hu. 2018. "Efficient neural architecture search with network morphism." *arXiv preprint arXiv:1806.10282* .

Karaboga, D. 2005. *An idea based on honey bee swarm for numerical optimization.* Kayseri: Erciyes University, Engineering Faculty.

Karaboga, D. & B. Basturk. 2007. "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm." *Journal of Global Optimization* 39(3):459–471.

Kennedy, J. & R. Eberhart. 1995. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks.* Vol. 4 Perth: IEEE pp. 1942–1948.

Koza, J. R. 1992. *Genetic programming: on the programming of computers by means of natural selection.* Cambridge (MA): MIT Press.

Kulkarni, R. V. & G. K. Venayagamoorthy. 2010. "Particle swarm optimization in wireless-sensor networks: a brief survey." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 41(2):262–267.

LeCun, Y., Y. Bengio & G. Hinton. 2015. "Deep learning." *Nature* 521(7553):436.

Mirjalili, S., S. M. Mirjalili & A. Lewis. 2014. "Grey wolf optimizer." *Advances in Engineering Software* 69:46–61.

Mohamed, W., M. N. M. Salleh & A. H. Omar. 2012. A comparative study of reduced error pruning method in decision tree algorithms. In *2012 IEEE International Conference on Control System, Computing and Engineering.* Penang: IEEE pp. 392–397.

Olson, R. S. & J. H. Moore. 2016. TPOT: a tree-based pipeline optimization tool for automating machine learning. In *Workshop on automatic machine learning.* New York: JMLR pp. 66–74.

Osaba, E., F. Diaz & E. Onieva. 2014. "Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts." *Applied Intelligence* 41(1):145–166.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss & V. Dubourg. 2011. "Scikit-learn: machine learning in Python." *Journal of Machine Learning Research* 12(Oct):2825–2830.

Pena-Reyes, C. A. & M. Sipper. 2000. "Evolutionary computation in medicine: an overview." *Artificial Intelligence in Medicine* 19(1):1–23.

Rosenblatt, F. 1961. *Principles of neurodynamics. Perceptrons and the theory of brain mechanisms.* Buffalo: Cornell Aeronautical Lab.

Russell, S. J. & P. Norvig. 2016. *Artificial intelligence: a modern approach.* 3rd ed. Harlow: Pearson.

Schapire, R. E. 1999. A brief introduction to boosting. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI'99 San Francisco: Morgan Kaufmann Publishers pp. 1401–1406.

Schmieder, R. & R. Edwards. 2011. "Quality control and preprocessing of metagenomic datasets." *Bioinformatics* 27(6):863–864.

Soda, P. & G. Iannello. 2010. Decomposition methods and learning approaches for imbalanced dataset: an experimental integration. In *2010 20th International Conference on Pattern Recognition*. New York: IEEE pp. 3117–3120.

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

Stephen, V. S. 1997. "Selecting and interpreting measures of thematic classification accuracy." *Remote Sensing of Environment* 62(1):77–89.

Storn, R. & K. Price. 1997. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces." *Journal of Global Optimization* 11(4):341–359.

Suykens, J. A. K. & J. Vandewalle. 1999. "Least squares support vector machine classifiers." *Neural Processing Letters* 9(3):293–300.

Thornton, C., F. Hutter, H. Hoos & K. Leyton-Brown. 2013. Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York: ACM pp. 847–855.

UCI. 2019*a*. "Abalone dataset [podatkovna zbirka].".
URL: *https://archive.ics.uci.edu/ml/datasets/abalone*

UCI. 2019*b*. "Ecoli dataset [podatkovna zbirka].".
URL: *https://archive.ics.uci.edu/ml/datasets/ecoli*

UCI. 2019*c*. "Yeast dataset [podatkovna zbirka].".
URL: *https://archive.ics.uci.edu/ml/datasets/Yeast*

Vrbančič, G., L. Brezočnik, U. Mlakar, D. Fister & I. Fister Jr. 2018. "NiaPy: Python microframework for building nature-inspired algorithms." *Journal of Open Source Software* 3(23):613.

Xavier-Júnior, J. C., A. Freitas, A. Feitosa-Neto & T. B. Ludermir. 2018. A novel evolutionary algorithm for automated machine learning focusing on classifier ensembles. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*. Sao Pablo: IEEE pp. 462–467.

Yang, X.-S. 2010*a*. "Firefly algorithm, stochastic test functions and design optimisation." *International Journal of Bio-Inspired Computation* 2(2):78–84.

Yang, X.-S. 2010*b*. *A new metaheuristic bat-inspired algorithm*. Berlin: Springer pp. 65–74.

Yang, X.-S. 2014. "Swarm intelligence based algorithms: a critical analysis." *Evolutionary Intelligence* 7(1):17–28.

Yang, X.-S. & S. Deb. 2009. Cuckoo search via Lévy flights. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. Coimbatore: IEEE pp. 210–214.

Zhang, Y., S. Wang & G. Ji. 2015. "A comprehensive survey on particle swarm optimization algorithm and its applications." *Mathematical Problems in Engineering* 2015(931256):1–38.

Zhu, J., N. Chen & E. P. Xing. 2011. Infinite latent SVM for classification and multi-task learning. In *Advances in Neural Information Processing Systems 24*. Granada: Curran Associates pp. 1620–1628.