UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET

Milica Vukčević

Detekcija tipova zemljišta u Crnoj Gori upotrebom algoritama za klasterizaciju

MAGISTARSKI RAD

Podgorica, 2022.

Univerzitet Crne Gore
Elektrotehnički fakultet

Milica Vukčević

Detekcija tipova zemljišta u Crnoj Gori upotrebom algoritama za klasterizaciju

MAGISTARSKI RAD

Podgorica, 2022.

PODACI I INFORMACIJE O MAGISTRANDU

Ime i prezime: Milica Vukčević

Datum i mjesto rođenja: 09.09.1989. Podgorica, Crna Gora

Prethodno završene studije:

- Osnovne studije: Elektrotehnički fakultet Podgorica, Univerzitet Crne Gore, smjer: Elektronika, telekomunikacije i računari, 180 ECTS kredita, 2013.godine
- Specijalističke studije: Elektrotehnički fakultet Podgorica, Univerzitet Crne Gore, smjer: Računari, 60 ECTS kredita, 2013.godine

INFORMACIJE O MAGISTARSKOM RADU

Fakultet na kojem je rad odbranjen: Elektrotehnički fakultet

Studijski program: Elektronika, telekomunikacije i računari - Računari

Naslov rada: Detekcija tipova zemljišta u Crnoj Gori upotrebom algoritama za klasterizaciju

UDK, OCJENA I ODBRANA MAGISTARSKOG RADA

Datum prijave magistarskog rada:

Datum sjednice Vijeća na kojoj je prihvaćena tema: 23.07.2020. god.

Komisija za ocjenu teme i podobnosti magistranda:

Prof. dr Vesna Popović Bugarin

Prof. dr Miloš Daković

Prof. dr Slobodan Đukanović

Mentor: Prof. dr Vesna Popović Bugarin

Komisija za ocjenu rada:

Prof. dr Vesna Popović Bugarin

Prof. dr Miloš Daković

Prof. dr Slobodan Đukanović

Komisija za odbranu rada:

Prof. dr Vesna Popović Bugarin

Prof. dr Miloš Daković

Prof. dr Slobodan Đukanović

Datum odbrane: 05.10.2022.

Datum promocije:

Ime i prezime autora: Milica Vukčević, Spec. Sci.

ETIČKA IZJAVA

U skladu sa članom 22 Zakona o akademskom integritetu i članom 24 Pravila studiranja na postdiplomskim studijama, pod krivičnom i materijalnom odgovornošću, izjavljujem da je magistarski rad pod naslovom

„Detekcija tipova zemljišta u Crnoj Gori upotrebom algoritama za klasterizaciju"

moje originalno djelo.

Podnosilac izjave,

Milica Vukčević, Spec. Sci.

Podgorica, 17.05.2022. godine.

Sadržaj

Izvod rada

Velike količine podataka i njihov svakodnevni rast doveli su do potrebe za što bržom i jednostavnijom obradom podataka. Brojne tehnike data mining-a su se pokazale kao odlično rješenje u ekstrahovanju korisnog znanja iz velikih baza podataka. U ovom magistarskom radu su u fokusu tehnike klasterizacije. Tehnike klasterizacije formiraju klastere tako da su podaci unutar jednog klastera međusobno sličniji u odnosu na podatke koji se nalaze u drugim klasterima.

U ovom istraživanju se koristi dio podataka iz pedološke baze Crne Gore, ustupljene od Biotehničkog fakulteta Univerziteta Crne Gore i digitalizovane od strane BIO-ICT Centra izvrsnosti.

Implementacija algoritma zasnovanog na gustini raspodjele podataka (DBSCAN), kao i *k*-medoids (CLARA i CLARANS) i fuzzy *k*-medoids (RFCMdd, FCMRANS i FCLARANS) algoritama i njihova primjena na fizičko-hemijskim karakteristikama zemljišta u Crnoj Gori su predmet ovog istraživanja. Kroz primjere je data komparativna analiza algoritama kako međusobno tako i sa *k*-means i fuzzy *k*-means algoritmima, koji su ranije primijenjeni nad istim podacima [1]. Rezultati su prikazani tabelarno i u grafičkoj formi pogodnoj za detaljnu uporednu analizu. Pokazane su prednosti i nedostaci primjene analiziranih algoritama na pedološkim podacima.

Primjenom analiziranih algoritama klasterizacije na 5 hemijskih parametara automatizuje se detekcija zastupljenih tipova zemljišta, čijom će se vizuelizacijom dobiti tematska pedološka mapa Crne Gore. Bez obzira na prisustvo šuma i velikog broja nedostajućih podataka, dobijena tematska mapa biće uporediva sa postojećom ekspertskom. Veća robusnost analiziranih algoritama na šum u odnosu na *k*-means i fuzzy *k*-means algoritme, rezultiraće boljim mapama. Identifikacija tipova zemljišta je izvršena uz minimalno znanje o bazi koja se koristi.

Za implementaciju algoritama i njihovu primjenu na pedološkim podacima Crne Gore, kao i dobijanje pedoloških tematskih mapa korišćen je R programski jezik.

Ključne riječi: data mining, klasterizacija, algoritmi klasterizacije, DBSCAN, CLARA, CLARANS, RFCMdd, FCMRANS, FCLARANS, *k*-medoids, fuzzy *k*-medoids, pedološki podaci.

Abstract

Large amounts of data and their daily growth have led to the need for faster and simpler data processing. Numerous data mining techniques have proven to be an excellent solution in extracting useful knowledge from such databases. In this master's thesis, the focus is on clustering techniques. Clustering techniques form clusters so that the data within one cluster are more similar to each other than the data in other clusters.

This research uses part of the data from the pedological database of Montenegro, provided by the Biotechnical Faculty of the University of Montenegro and digitized by the BIO-ICT Center of Excellence.

The implementation of the algorithm based on density distribution (DBSCAN), as well as *k*-medoids (CLARA and CLARANS) and fuzzy k-medoids (RFCMdd, FCMRANS and FCLARANS) algorithms and their application to physical and chemical soil characteristics data in Montenegro are the subject of this research. Through the examples, a comparative analysis of the algorithms is given which were previously applied over the same data [1]. The results are presented in the tables and in a graphical form suitable for detailed comparative analysis. The advantages and disadvantages of applying the analyzed algorithms to pedological data are shown.

By applying the analyzed clustering algorithms on 5 chemical parameters, the detection of represented soil types is automated, the visualization of which will provide a thematic pedological map of Montenegro. Regardless of the presence of noise and a large number of missing data, the obtained thematic map will be comparable to the expert's one. Greater robustness of the analyzed algorithms to noise compared to k-means and fuzzy k-means algorithms will result in better maps. Identification of the soil types was performed with minimal knowledge of the database used.

The R programming language is used for the implementation of algorithms and their application on the pedological data of Montenegro, as well as obtaining pedological thematic maps.

Keywords: data mining, clustering, clustering algorithms, DBSCAN, CLARA, CLARANS, RFCMdd, FCMRANS, FCLARANS, *k*-medoids, fuzzy *k*-medoids, pedological data.

1 Uvod

Zemljište pokriva 1/3 površine planete Zemlje. Predstavlja glavno stanište čovjeka, mnogih biljaka, životnja, mikroorganizama, itd. Kao ogromno prirodno bogatstvo koje se teško obnavlja, potrebno ga je što racionalnije koristiti.

Na sastav i kvalitet zemljišta utiču brojni faktori, među kojima najviše: klima, biljni i životinjski svijet, mikroorganizmi, brojni biološki i fizičko-hemijski procesi koji se neprestano odvijaju, kao i čovjek svojim svakodnevnim djelovanjem. Otpadni produkti elektrana i fabrika, razgradivi i nerazgradivi materijali, razne hemikalije dospijevaju u zemlju i utiču na povećanje njene zagađenosti. Ovo je sve od velike važnosti za čovjekov opstanak, s obzirom da zemljište učestvuje u lancu ishrane i da najveći procenat hrane čovjek dobija njegovim obrađivanjem. U tom pogledu bi poznavanje zastupljenih tipova zemljišta, kao i brza dostupnost informacija o sadržaju materija i svojstava koji ga karakterišu, bilo od velike važnosti u smanjenju negativnog antropogenog uticaja (npr. primjenom adekvatnih tehnika i sredstava tretiranja zemljišta u zavisnosti od regije, samim tim i sastava zemljišta, kako bi se dobili što veći benefiti njegovom obradom) i sprečavanju remećenja sadržaja materija koje se u njemu nalaze. Iz tog razloga je potrebno da informacije o zemljištu budu pristupačne svima u najčitljivijem obliku i za kratak vremenski period.

U ovom istraživanju je korišćena pedološka baza Crne Gore, nastala prije više od 30 godina. Pedolozi su prikupljali podatke o zemljištu. Ručno su ih upisivali u sveske, a kasnije su prekucavani u *Excel* tabele od strane zaposlenika Biotehničkog fakulteta, Univerziteta Crne Gore. Prebacivanje iz jednog u drugi format dovelo je do grešaka u podacima. Tokom projekta BIO ICT Centra Izvrsnosti odrađena je digitalizacija baze koja je obuhvatila analizu i ispravljanje grešaka među podacima. Digitalizacijom je objedinjen veliki broj *Excel* fajlova u jedinstvenu normalizovanu *SQLite* baza podataka. Na taj način podaci su postali dostupniji i pogodniji za dalje korišćenje i obradu.

Prva pedološka, tematska mapa Crne Gore sa zastupljenim tipovima zemljišta je ručno kreirana od strane eksperata. Da bi se u budućnosti olakšalo i ubrzalo dobijanje željenih informacija o zemljištu, cilj ovog rada je automatizovati dobijanje korisnih informacija iz velikih baza podataka. Najbolje rješenje za automatizaciju tih procesa predstavlja primjena data mining tehnika, koje obuhvataju implementaciju sistema koji će za kratko vrijeme, na osnovu ulaznih parametara, naći šablone i izvući korisno znanje iz velikih baza podataka, kakva je pedološka baza Crne Gore.

U fokusu rada su tehnike klasterizacije, njihova implementacija i primjena na mehaničko-fizičke i hemijske parametre zemljišta. Klasterizacija predstavlja grupisanje međusobno najsličnih podataka u iste klastere, dok se podaci koji pripadaju različitim klasterima međusobno razlikuju. Predmet istraživanja je primjenljivost odabranih tehnika klasterizacije na pedološkim podacima uz minimalno posjedovanje znanja o bazi, te nesavršenostima iste. Poređenjem analiziranih algoritama pokazano je šta su prednosti a šta nedostaci njihove primjene nad ovakvim tipovima podataka. Svi rezultati primjene su predstavljeni tabelarno i u grafičkom obliku, radi jednostavnije validacije.

Cilj je automatizovano dobijanje pedološke tematske mape Crne Gore sa zastupljenim tipovima zemljišta, koja je uporediva sa pedološkom, tematskom mapom dobijenom primjenom *k*-means algoritma [1], a samim tim i ekspertskom pedološkom mapom. Vizuelizacija podataka je i glavni vid validacije tačnosti dobijenih rezultata.

U nastavku ovog poglavlja je objašnjen pojam data mining-a i tehnika data mining-a, klasterizacija podataka i predstavljena baza koja se koristi tokom istraživanja. U drugom poglavlju rada dat je pregled i analiza algoritama klasterizacije. U trećem su predstavljani korišćeni pedološki podaci, kao i rezultati primijene predstavljenih algoritama nad tim podacima. U četvrtom su date dobijene pedološke mape i uporedna analiza sa ekspertskom pedološkom mapom Crne Gore. Peto poglavlje predstavlja zaključak rada sa rezimeom magistarskog rada.

1.1 Data mining i tehnike data mining-a

Rudarenje podataka (engl. data mining) je, već sad dobro poznata, oblast koja obuhvata tehnike pomoću kojih se, pronalaženjem skrivenih obrazaca, izvlače korisne informacije iz velikih baza i prevode u oblik razumljiv za dalju analizu i tumačenje. Drugim riječima, suština data mining tehnika je ekstrahovanje korisnog znanja iz sirovih podataka i njihovo transformisanje u oblik pogodan za dalju upotrebu.

Postoji veći broj tehnika data mining-a, među kojima su:

- Klasterizacija
- Klasifikacija
- Regresija
- Stablo odlučivanja
- Asocijacijska pravila
- Predviđanje
- Vizuelizacija

Data mining tehnike mogu pripadati predikativnom ili desktiptivnom modelu učenja.

Kod predikativnog modela (tj. nadgledano učenje, engl. directed, supervised data mining) prave se predviđanja bazirana na zaključcima dobijenim iz dostupnih podataka. Analitičar već posjeduje neko znanje o problemu, koje je stekao u prošlosti, i primjenjuje ga u cilju dobijanja što boljih rezultata. Metode koje se koriste u predikativnoj analizi su: klasifikacija, regresija, stabla odlučivanja i predviđanje.

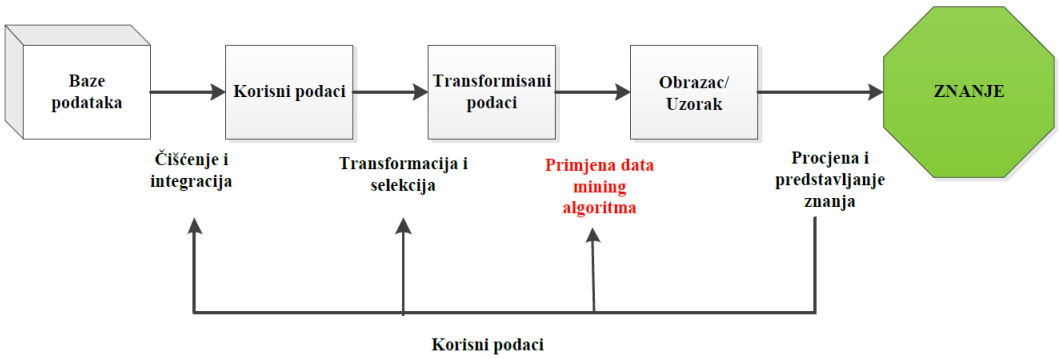
Nenadgledano učenje (eng. undirected, unsupervised data mining) se često naziva i deskriptivnim modeliranjem. U ovom slučaju ne posjedujemo nikakvo znanje, osim samih ulaznih podataka, tako da nemamo prisutan proces učenja iz podataka. Teži se pronalaženju obrazaca na osnovu kojih se može doći do korisnih saznanja. Metode koje se koriste u deskriptivnoj analizi su: asocijacijska pravila, klasterizacija i vizuelizacija.

Zaključuje se da je osnovna razlika između ova dva modela u tome što nenadgledane metode (npr. klasterizacija) ne mogu učiti iz podataka, jer podaci nisu označeni (nije poznato kojem klasteru pripadaju), dok kod nadgledanog učenja (npr. klasifikacije) već se posjeduje određeno znanje o podacima, tj. jedan dio podataka je već označen (poznato je kojoj klasi pripadaju), a algoritam treba da nauči da dobro klasifikuje neoznačene podatke.

Jedan od pojmova koji je potrebno uvesiti kada je u pitanju data mining je velika količina podataka (engl. big data). Odnosi se na kompleksne i heterogene baze podataka, kao što su relacione, prostorne, vremenske baze. U njima količina podataka raste konstantno prikupljanjem podataka iz različitih izvora.

Data mining se često upotrebljava kao sinonim za otkrivanje znanja u bazama podataka (engl. KDD – Knowledge Discovery in Databases), što su dva skroz različita pojma. Tehnike rudarenja podataka su samo dio procesa koji se odnosi na izvlačenje korisnog znanja iz velikih baza podataka. Otkrivanje znanja obuhvata sljedeće faze (slika 1):

- čišćenje podataka: eliminacija šuma, izuzetaka, duplikata, nedostajućih vrijednosti i nekonzistentnosti u podacima,
- integracija podataka: objedinjavanje podataka iz više izvora,
- izbor podataka: u ovom koraku podaci od važnosti za analizu se preuzimaju iz baza podataka,
- transformacija podataka: u ovom koraku podaci se transformišu i konsoliduju u oblik pogodan za primjenu data mining algoritma,
- selekcija: generisanje podataka koji su potrebni za analizu,
- data mining: izbor i primjena algoritama koji će otkriti skrivene šablone (obrazce) u podacima,
- procjena obrazaca: na osnovu utvrđenih mjera identifikuju se korisni šabloni,
- predstavljanje znanja: grafičko predstavljanje otkrivenog znanja.



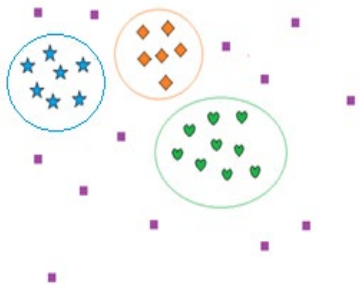
Slika 1. Proces otkrivanja znanja

U velikim količinama podataka sasvim je očekivano prisustvo šuma, izuzetaka i nepotpunih podataka. Šum predstavljaju podaci koji mnogo odstupaju od njihovih srednjih

vrijednosti, neodređeni podaci, greške među podacima koje treba ukloniti. Izuzeci su podaci koji se nalaze na ivicama klastera, najudaljeniji su od centra svog klastera i mnogo odstupaju od vrijednosti ostalih podataka. Mogu biti stvarni podaci, ali mnogo češće predstavljaju šum. Dakle, izuzeci su širi pojam od šuma, jer obuhvataju šum podatke i stvarne podatke koje mnogo odstupaju od vrijednosti ostalih podataka. Loš uticaj na rezultate data mining modela imaju i nepotpuni podaci koji se odnose na nedostajuće vrijednosti atributa podataka za neke uzorke. Sve ovo može unijeti nesigurnost i učiniti data mining model manje pouzdanim. Iz tog razloga se teži redukciji, pripisivanju vrijednosti koje nedostaju ili eliminisanju „loših“ podataka. Razvijaju se poboljšani data mining modeli koji prevazilaze pomenute vrste problema, dajući očekivane rezultate. Da bi se dobili što efikasniji modeli data mining-a važno je da je njegov dizajner upoznat sa bazom i problemima koje treba prevazići, kao i sa tim šta korisnici žele da otkriju i nauče iz podataka. Nakon toga se može pristupiti dizajniranju i implementaciji modela. Heterogeni podaci, svakodnevni porast količine podataka, oskudni i neodređeni podaci, iz više različitih autonomnih izvora se obrađuju i dobijaju se lokalni obrazci. Dobijeno globalno znanje nastalo razmjenom informacija, analizom i spajanjem lokalnih učenja se testira, prelazi se u fazu preprocesiranja i na osnovu povratnih informacija podešava model, kako bi se njegovom upotrebom postigli što precizniji rezultati. Zaključuje se da je data mining iterativni proces.

1.2 Algoritmi klasterizacije

Klasterizacija ili klaster analiza je veoma zastupljena tehnika data mining-a, koja je našla široku primjenu u mnogim oblastima. Klaster analiza služi za otkrivanje nepoznatih šablona u struktuiranim podacima. Cilj algoritama klasterizacije je da na osnovu atributa uzoraka formiraju klastere podataka, grupišući ih tako da su uzorci unutar jednog klastera međusobno sličniji u odnosu na uzorke dodijeljene drugim klasterima, slika 2. Grupisanje podataka u klastere olakšava njihovu analizu, pošto se umjesto analize cijelog skupa podataka mogu posmatrati podaci kroz predstavnike klastera i same klastere, jer svaki klaster nosi određene atribute.



Slika 2. Podjela uzoraka u 3 klastera, podaci van kružnica predstavljaju šum

Kod algoritama klasterizacije postoje razlike u zavisnosti od principa po kome vrše grupisanje. Zajedničko svima je da su uzorci unutar jednog klastera međusobno sličniji u odnosu na uzorke u ostalim klasterima. Tehnike klasterizacije se dijele na:

- **Hijerarhijske tehnike** – zasnivaju se na dendogramu, tj. grafičkom prikazu klastera u obliku stabla povezivanja. Dijele se na dvije osnovne vrste: na aglomerativne, tj. tehnike udruživanja (engl. bottom-up, svaki objekat se posmatra kao poseban klaster, u svakom koraku spajuju se najsličniji klasteri sve dok se ne spoje svi klasteri u jedan) i, suprotno njima, tehnike razdvajanja (engl. top-down, divisive, gdje svi uzorci pripadaju jednom klasteru, od kojeg u sljedećem koraku formiraju dva nova klastera, koji se ponovo dijele i tako sve dok se ne postigne da svaki uzorak pripada samo jednom klasteru). Dakle, klasteri se formiraju na osnovu tehnika spajanja ili razdvajanja. Prednost ove metode je što nije potrebno poznavanje broja klastera. S druge strane nedostatak je što jednom spojeni uzorci se više ne mogu razdvojiti, i obratno. Prisutna je velika osjetljivost hijerarhijskih tehnika na šum i izuzetke. Među ovim tehnikama spadaju: BIRCH (engl. Balanced Iterative Reducing and Clustering), CURE (engl. Clustering Using REpresentatives), ROCK (engl. RObust Clustering using linKs).
- **Tehnike particionisanja** (ili nehijerarhijske, tehnike rasčlanjivanja) – podaci koji imaju slične atribute se grupišu u klastere, dok klasteri međusobno nose različite atribute. Podaci se grupišu na osnovu zadatog broja klastera. Kroz iteracije se uzorci premještaju između klastera u cilju poboljšanja neke ciljne funkcije. Ovoj grupi pripadaju *k*-medoids i *k*-means algoritmi klasterizacije, kao i njihovi fuzzy oblici. Karakteriše ih veća robusnost na šum i izuzetke u odnosu na hijerarhijske tehnike.
- **Tehnike zasnovane na gustini raspodjele** – uzorci se grupišu u klastere na osnovu gustine raspodjele, formirajući klastere na mjestima gdje je gustina podataka veća. Pogodne su za otkrivanje klastera proizvoljnog oblika i detekciju šuma. Među njima su DBSCAN (engl. Density-Based Spatial Clustering of Applications with Noise) i OPTICS (engl. Ordering Points To Identify the Clustering Structure), itd.
- **Tehnike zasnovane na modelu** – postoje dva pristupa ove tehnike i to statistički pristup i pristup zasnovan na neuronskim mrežama.
- **Tehnike zasnovane na mreži** – prostor podataka se posmatra kao mreža, koju čini određeni broj ćelija. Na taj način su objekti dodijeljeni ćelijama. Nad svakom ćelijom se primjenjuje klasterizacija i određuje se gustina podataka u svakoj ćeliji. Klasteri se formiraju od susjednih grupa ćelija koje imaju traženu gustinu, minimizujući funkciju cijene. Sve ćelije sa manjom gustinom podataka od tražene se smatraju šumom i eliminišu se. Za ove

algoritme je karakteristično brzo izvršenje algoritma, koje isključivo zavisi od broja ćelija od kojih je mreža formirana.

Podjela algoritama klasterizacije vrši se i po kriterijumu preklapanja klastera, gdje postoji tvrdo (engl. hard, crisp) i meko (engl. soft) klasterovanje. Tvrda klaster analiza podrazumijeva da su klasteri međusobno isključivi, tj. da jedan uzorak može pripadati samo jednom klasteru. Suprotno tome, meka klasterizacija je klasterizacija koju karakterišu nejasne, zamućene granice između klastera, dolazi do njihovog miješanja jer svaki uzorak pripada svakom od klastera sa različitim stepenom pripadnosti.

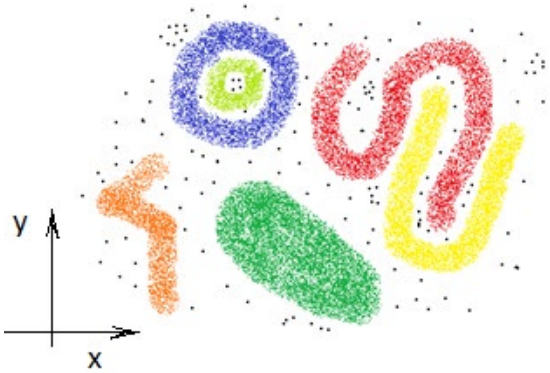
Klasterizacija je jedna od tehnika data mining-a koja se pokazala kao vrlo pouzdana za izvlačenje korisnog znanja iz relacionih, prostornih baza. Razvijeni su brojni algoritmi klasterizacije, sa različitim vremenom izvršavanja i primjenljivošću na različite veličine baza. Grupisanje podataka je našlo primjenu u mnogim oblastima: u pedologiji [1]-[2], segmentaciji slike [3], poljoprivredi [4], biomedicini za klasterizaciju EKG signala [5], genetici [6], meteorologiji [7] itd. Na osnovu ulaznih podataka algoritmi analiziraju bazu, prepoznaju šablone, donose zaključke i kao rezultat grupišu podatke.

1.3 Pedološki podaci i pedološka mapa Crne Gore

Pedološka baza Crne Gore nastala je u periodu 1958. do 1988. godine pedogenskim izučavanjem zemljišta od strane eksperata. Kao rezultat je nastalo 6 svezaka u kojima su ručno upisivani podaci. Sa njima je objedinjen dio podataka koji je prikupljen od strane hrvatskih pedologa Zavoda za agroekologiju u Zagrebu, a koji nijesu postojali zabilježeni u navedenim sveskama. Nakon toga je uslijedilo njihovo ručno prekucavanje u *Excel* tabele. Iz značaja pedoloških podataka za cijelu državu proizilazi potreba za objedinjavanjem svih *Excel* fajlova u jedinstvenu bazu. U toku trajanja BIO-ICT projekta, njihovi istraživači su odradili proces digitalizacije podataka. Digitalizacija je obuhvatila prije svega analizu i ispravljanje grešaka koji su nastali prebacivanje iz jednog u drugi format, kao i povezivanje karakteristika zemljišta istih profila. S obzirom na veliki broj ljudi koji su učestvovali u procesu prekucavanja podataka u *Excel* fajlove, kao i brojnih drugih faktora kao što su korišćenje različitih verzija excel-a, formatiranje podataka u tabelama, duplo kucanje stranica itd., došlo je do grešaka među podacima. Prilikom nailaženja na greške, iste su ispravljane u originalnim *Excel* fajlovima, u skladu sa podacima koji su postojali u sveskama. Tako ispravljeni fajlovi su ponovo učitavani u bazu. Na kraju je nastala objedinjena *SQLite* baza podataka, koja je pogodna za korišćenje i dobijanje potrebnih pedoloških informacija o zemljištu u Crnoj Gori. Ovo ujedno predstavlja i prvu pedološku bazu Crne Gore.

2 Pregled primijenjenih algoritama klasterizacije

Odabir odgovarajuće tehnike klasterizacije pri analizi podataka je od suštinske važnosti za kvalitet klasterizacije. Svi algoritmi klasterizacije imaju zajedničku osobinu, da formiraju klastere tako da podaci unutar jednog klastera imaju veći stepen sličnosti međusobno, u odnosu na podatke u ostalim klasterima. Razlike u klasterizaciji među algoritmima se ogledaju u tome da li otkrivaju klastere proizvoljnog oblika (klasteri koji nijesu sfernog ili konveksnog oblika, već mogu biti izduženi, linearni, izvučeni, „S“ oblika [8], slika 3), da li su pogodne za različite tipove baza podataka, kolika je njihova skalabilnost kod visoko dimenzionisanih podataka, koliko su uspješni u prevazilaženju šuma među podacima, kao i nedostajućih podataka, koliko je minimalno znanje o podacima i zahtjevima za ulaznim argumentima algoritma. Radi lakše ilustracije, na slici 3 su dati podaci koji se sastoje od dvije karakteristike, preslikane na *x* i *y* osu, na osnovu kojih su grupisani u klastere.



Slika 3. Klasteri proizvoljnog oblika predstavljeni različitim bojama, dok su crnom označene tačke šuma

U ovom radu su implementirani, primijenjeni i analizirani algoritmi:

- zasnovani na gustini raspodjele – klasteri se formiraju u oblasti gdje su podaci gušće skoncentrisani u odnosu na oblasti sa manjom koncentracijom podataka: DBSCAN [2], [8], [9],
- *k*-medoids – za razliku od algoritama zasnovanih na gustini raspodjele, kod *k*-medoids algoritama klastere formiraju uzorci koji se nalaze na najmanjoj udaljenosti od medoida, tj. uzorci su najsličniji medoidu klastera kome pripadaju. Medoidi su centri klastera, selektovani iz ulaznog skupa podataka. Kao i algoritmi zasnovani na gustini raspodjele, *k*-medoids algoritmi formiraju međusobno isključive klastere sa jasnim granicama između klastera: CLARA (engl. Clustering Large Applications) i CLARANS (engl. Clustering of Large Applications based on RANdomized Search) ([2], [10], [11], [12]),

- fuzzy *k*-medoids – svaki uzorak pripada svakom klasteru, sa određenim stepenom pripadnosti i blago zamućenim prelazima između klastera: RFCMdd (engl. Robust Fuzzy *C*-medoids Algorithm), FCMRANS (engl. Fuzzy C-Medoids based on RANdomized Search), FCLARANS (engl. Fuzzy Clustering of Large Applications based on RANdomized Search), [13], [14].

2.1 DBSCAN algoritam

DBSCAN algoritam klasterizacije je predložen od strane Martin Ester, Hans-Peter Kriegel, Jörg Sander i Xiaowei Xu, 1996. godine [8].

DBSCAN formira klastere na mjestima veće koncentracije uzoraka. Klaster se formira od tačaka koje u svom *eps* susjedstvu (definisanom *eps* radiusom) sadrže minimalan broj tačaka, odnosno susjeda (*MinPts*). Ovo je ujedno i glavni uslov koji treba da se zadovolji kod DBSCAN algoritma. Algoritam počinje slučajnim odabirom proizvoljne tačke i ispituje se da li je prethodni uslov ispunjen. Ukoliko jeste, formira se klaster od te tačke i njenih susjeda. Dalje se ispituje da li dobijeni susjedi u svom *eps* susjedstvu sadrže *MinPts* tačaka. Ukoliko sadrže taj klaster se proširuje novim susjedstvom. Postupak se ponavlja sve dok makar jedan susjed zadovoljava uslov. Ukoliko više nema susjeda koji zadovoljavaju uslov, klaster je formiran i prelazi se na traženje novog klastera. Slučajno se bira nova tačka koja prethodno nije dodijeljena nijednom od klastera i ispituje se njeno *eps* susjedstvo. Postupak formiranja klastera ispitivanjem susjedstva se ponavlja dok se ne ispitaju sve tačke skupa i ne otkriju svi klasteri. Tačke koje u svom *eps* susjedstvu ne sadrže *MinPts* tačaka i nijesu priključene nijednom od klastera se proglašavaju šumom. Rezultat klasterizacije DBSCAN algoritma su klasteri proizvoljnog oblika.

Postoje dva tipa tačaka u klasteru, tačke koje se nalaze unutar klastera ili tačke jezgra (engl. core points) i tačke na ivicama klastera ili granične tačke (engl. border points). *eps* susjedstvo granične tačke sadrži znatno manji broj tačaka u odnosu na *eps* susjedstvo tačke jezgra. U skladu sa tim, za *MinPts* se bira mala vrijednost da bi sve tačke koje pripadaju jednom klasteru bile uključene u taj klaster.

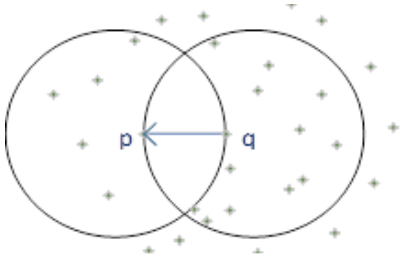
Klaster mora sadržati najmanje jednu jezgro tačku. Sve tačke koje su označene kao tačke jezgra se mogu dodijeliti isključivo jednom klasteru. Jedna granična tačka može biti granična tačka za više od jednog klastera. U tom slučaju tačka pripada klasteru kojem je prvo dodijeljena. Ovo dovodi do zaključka da jedan klaster može sadržati manje tačaka nego što je *MinPts* vrijednost.

Neka su dati skup podataka *D* kao dvodimenzioni, *eps* susjedstvo i *MinPts* [8]:

Definicija 1 - ***eps susjedstvo*** (oznaka $N_{eps}(p)$) posmatrane tačke *p* obuhvata sve tačke *q* koje se nalaze na rastojanju iz intervala [0, eps], tj. $N_{eps}(p) = \{q \in D, dist(p, q) \leq eps\}$.

Definicija 2 – Tačka *p* je ***direktno dostižna gustinom*** iz tačke *q*, ako *p* pripada *eps* susjedstvu tačke *q* ($p \in N_{eps}(q)$) i ako *q* zadovoljava uslov tačke jezgra ($|N_{eps}(q)| \geq MinPts$) (slika

4 - Radi lakše ilustracije algoritma posmatra se dvodimenzioni skup podataka, a sve može da se primijeni na skup većih dimenzija u slučaju da podaci imaju više atributa.)



Slika 4. *p* je direktno dostižno gustinom iz *q*, ali *q* nije direkno dostižna iz tačke *p*.

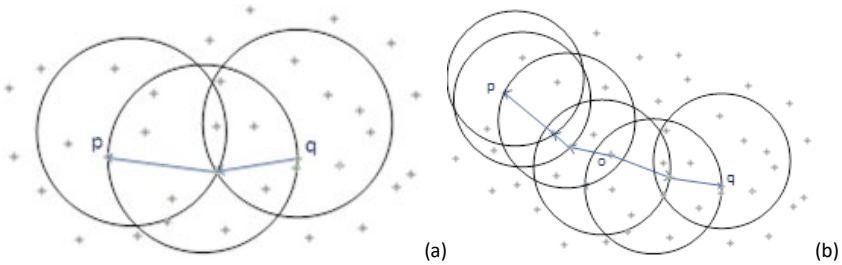
Definicija 3 – Tačka *p* je ***dostižna gustinom*** iz tačke *q* ako postoji lanac tačaka p_1, p_2, p_3, …p_n, $p_1=q$, $p_n=p$ tako da je p_{i+1}, i=1, 2, 3, … n, direktno dostižna iz p_i (slika 5).

Definicija 4 – Tačka p je ***povezana gustinom*** sa tačkom *q* ako postoji tačka *o* takva da su *p* i *q* dostižne gustinom iz tačke *o* (slika 5).

Definicija 5 – ***klaster***: Za date parametre *eps* i *MinPts* i *D* kao ulazni skup podataka, postoji neprazan klaster *C*, koji je podskup skupa *D*, a zadovoljava sljedeće uslove:

Maksimalnost: za svako *p* koje pripada klasteru *C*, postoji neka tačka $q \in C$ koja je dostižna gustinom iz *p*.

Konektivnost: za svako $p \in C$ postoji tačka $q \in C$ takvo da je *p* povezana gustinom sa *q*.



Slika 5. (a) *p* je dostižno gustinom iz *q*, ali *q* nije dostižno gustinom iz *p*; (b) *p* i *q* su međusobno povezane gustinom preko tačke *o*.

Definicija 6 - ***šum***: Neka je *D* analizirana baza podataka. Klasteri C_1, C_2, … C_k su podskupovi baze *D*, eps_i i $MinPts_i$ parametri gustine klastera za i=1, 2, … k. Tada tačke šuma *p* mogu biti definisane kao tačke koje ne pripadaju nijednom od pomenutih klastera C_i iz skupa *D*, ali su $p \in D$, i=1, 2, 3, … k.

Pored navedenih definicija za tačnost algoritma važne su i sljedeća pravila:

1. Ako je *p* tačka iz skupa *D* i ako je ispunjen uslov tačke jezgra, tj. $|N_{eps}(q)| \geq MinPts$, tada postoji skup $O = \{o \mid o \in D$ je dostižno gustinom iz *p* za dato *MinPts* i *eps*$\}$ koji se označava kao klaster za dato *eps* i *MinPts*.

2. Neka je *C* klaster i *p* bilo koja tačka iz *C* za koju je ispunjen uslov tačke jezgra $|N_{eps}(q)| \geq MinPts$, onda je *C* klaster koji je jednak $O = \{o \mid o$ je dostižno gustinom iz *p* za dato *MinPts* i *eps*$\}$.

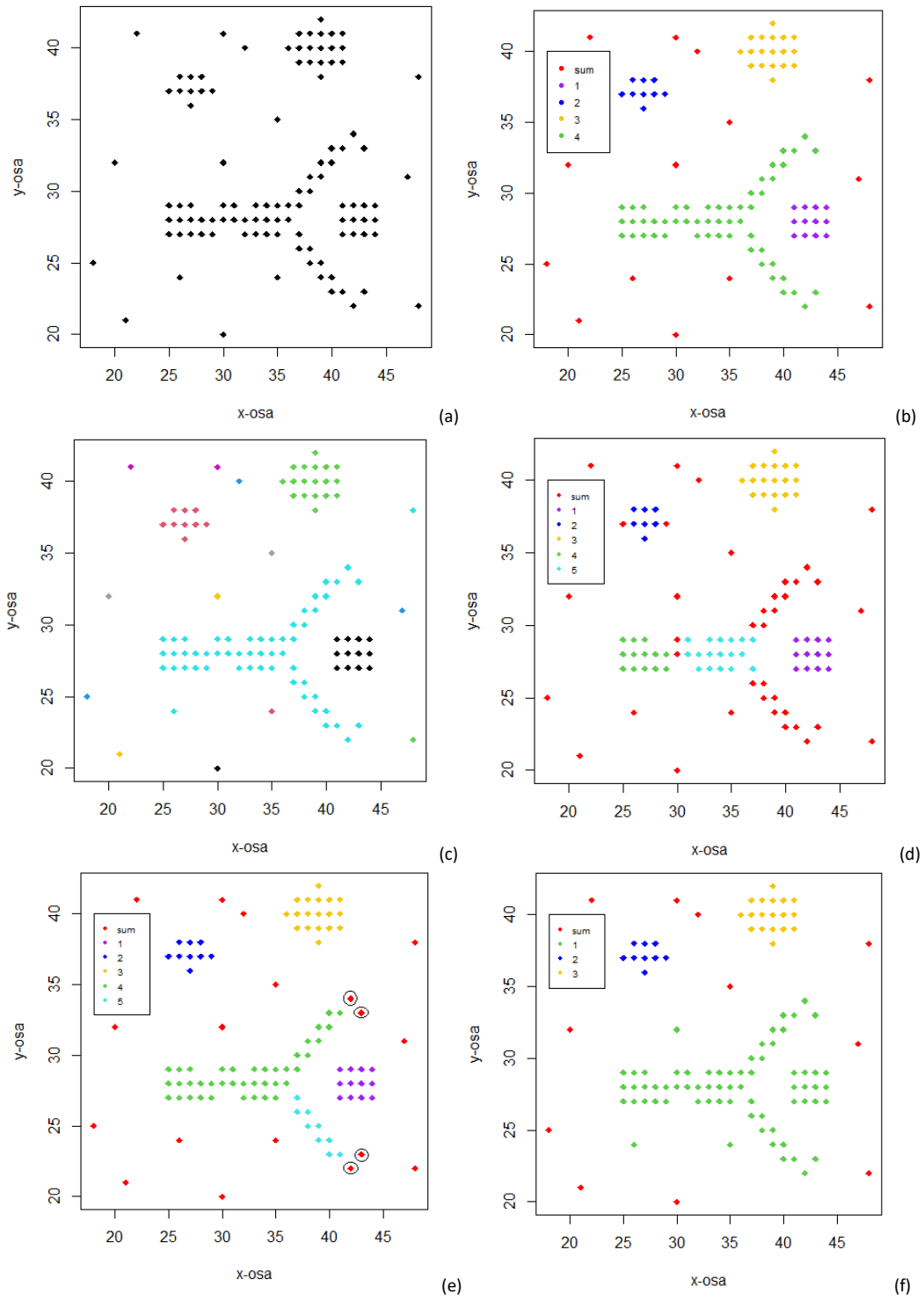
DBSCAN je implementiran tako što počinje slučajnim odabirom proizvoljne tačke *p*. Za tu tačku se ispituje da li u svom definisanom *eps* susjedstvu ima *MinPts* tačaka. Ukoliko ima, tj. ukoliko je zadovoljen uslov dostižnosti gustinom (Definicija 3) za globalno *eps* i *MinPts*, onda se od tačaka koje ispunjavaju uslov formira klaster (Definicija 5), u suprotnom je *p* okarakterisana kao šum (Definicija 6). Na taj način su sve tačke iz *eps* susjedstva tačke *p* označene da pripadaju tom klasteru. Postupak se ponavalja sa novom, neposjećenom tačkom, dok se ne ispitaju sve tačke i otkriju svi klasteri koji zadovoljavaju uslove za globalne vrijednosti *eps* susjedstva i *MinPts*. Globalna vrijednost parametara gustine je vrijednost parametara koja se koristi na nivou baze. Pored, *eps* i *MinPts*, kao ulazni parametar algoritma definiše se i funkcija rastojanja.

Kompleksnost DBSCAN algoritma je ($n \log n$), *n* je broj članova skupa koji se klasterizuje [8].

2.1.1 Određivanje globalnih vrijednosti parametara gustine, *eps* i *MinPts*

Kako gustina raspodjele podataka u velikim bazama nije uniformna, tako i kod DBSCAN algoritma postoji nekoliko mogućih vrijednosti parametara *eps* i *MinPts*. Iz tog razloga je potrebno odrediti njihove globalne vrijednosti, primjenljive na cjeli skup podataka. U cilju što boljeg prikazivanja principa po kome funkcioniše DBSCAN algoritam, kreirana je sintetička baza (slika 6 (a)) sa jasno definisana 4 klastera formirana na mjestima najveće gustine podataka i sa izolovanim tačkama koje predstavljaju šum. Rezultati primjene DBSCAN-a za optimalne vrijednosti ulaznih parametara, *eps* i *MinPts* je dat na slici 6 (b). Tu se jasno vide 4 formirana klastera, svaki označen različitom bojom i crvene tačke koje predstavljaju šum.

Globalna vrijednost *MinPts* parametra se uzima u skladu sa dimenzijom ulaznog skupa podataka. Ako je *D* analizirani skup podataka, gdje je *n* broj uzoraka u tom skupu za koje postoji tačno *br* atributa koji ih karakterišu, tada važi uslov $MinPts \geq br + 1$. Minimalan broj tačaka (*MinPts*) treba odabrati tako da ne bude previše mali, jer će onda svaki uzorak predstavljati poseban klaster. U tom slučaju algoritam formira klastere od tačaka koje su šum, što gubi smisao (slika 6 (c) – gdje su sve tačke šuma označene kao poseban klaster, ukupno je 18 klastera). Za podatke koji imaju veći procenat šuma potrebno je uzeti veće vrijednosti *MinPts*, ali ne previše velike da se ne bi tačke koje pripadaju jednom klasteru proglašavale kao šum, što takođe implicira lošu klasterizaciju (slika 6 (d) – gdje se vidi da su od svijetlo plavog klastera pod (b) formirana dva manja klastera, zeleni i svijetlo plavi, a jedan dio tačaka je identifikovan kao šum. Takođe, iz tamno plavog klastera su izvojene dvije tačke šuma, a razlog je jer one u svom susjedstvu ne sadrže definisani broj *MinPts* tačaka.



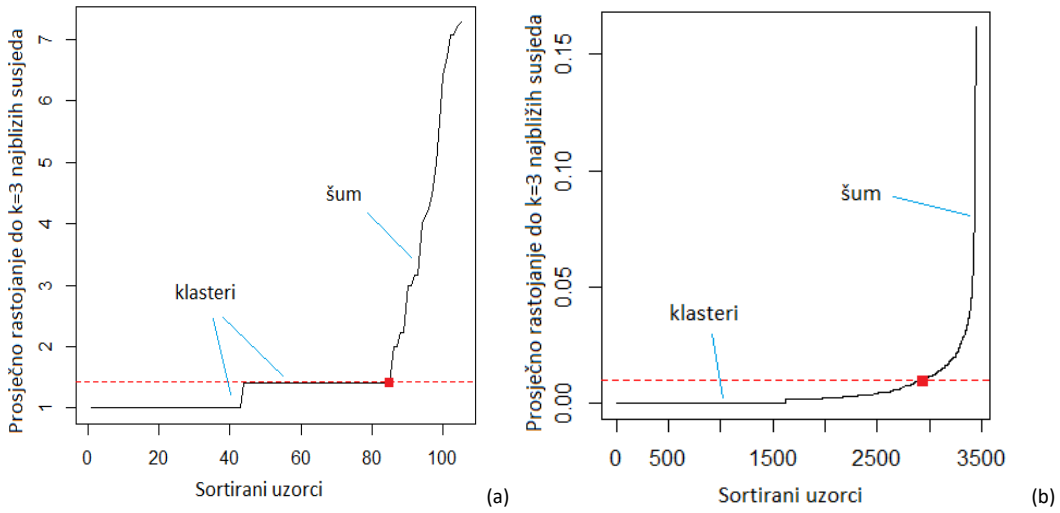
Slika 6. (a) Sintetička baza na koju se primjenjuje DBSCAN i rezultati primjene DBSCAN-a za: (b) optimalno *eps* i *MinPts*, (c) optimalno *eps* i *MinPts* manje od optimalnog, (d) optimalno *eps* i *MinPts* veće od optimalnog, (e) optimalno *MinPts* i *eps* manje od optimalnog i (f) optimalno *MinPts* i *eps* veće od optimalnog.

Za determinisanje vrijednosti *eps* parametra se koristi prosječna udaljenost do *k* najbližih susjeda.

Globalna vrijednost *eps* parametra gustine se dobija računanjem prosječne udaljenosti svakog uzorka do *k* = *MinPts* najbližih susjeda, prema formuli:

$$R_{kNN} = \sum_{j=1}^{k} dist(x_i\text{-}v_{ij})^2 / MinPts \quad (1)$$

Za svaku x_i tačku skupa, gdje je $i = 1, 2, \ldots n$, računa se prosječna suma rastojanja R_{kNN} do njenih $j = 1, 2, \ldots k$ najbližih susjeda. Dobijene vrijednosti za sve tačke iz skupa se sortiraju u rastući niz. Tačka, koja se nalazi u „laktu" grafika, u kojoj vrijednost rastojanja počinje naglo da raste predstavlja globalnu *eps* vrijednost na novou baze. Primjer grafika zavisnosti sortiranih uzoraka i prosječne sume rastojanja svakog uzorka do njihovih najbližih *k* susjeda je dat na slici 7 - (a) i (b). Za uzorke koji pripadaju nekom od klastera R_{kNN} nije veliko, dok za uzorke koji predstavljaju šum ima velike vrijednosti, jer se tačke šuma nalaze na velikoj udaljenosti u odnosu na ostale uzorke. Globalno *eps* je definisano u tački presjeka krive i crvene linije. Sve tačke sa rastojanjima jednakim ili manjim od te vrijednosti kod DBSCAN algoritma predstavljaju tačke jezgra i granične tačke. Ostalo je šum.

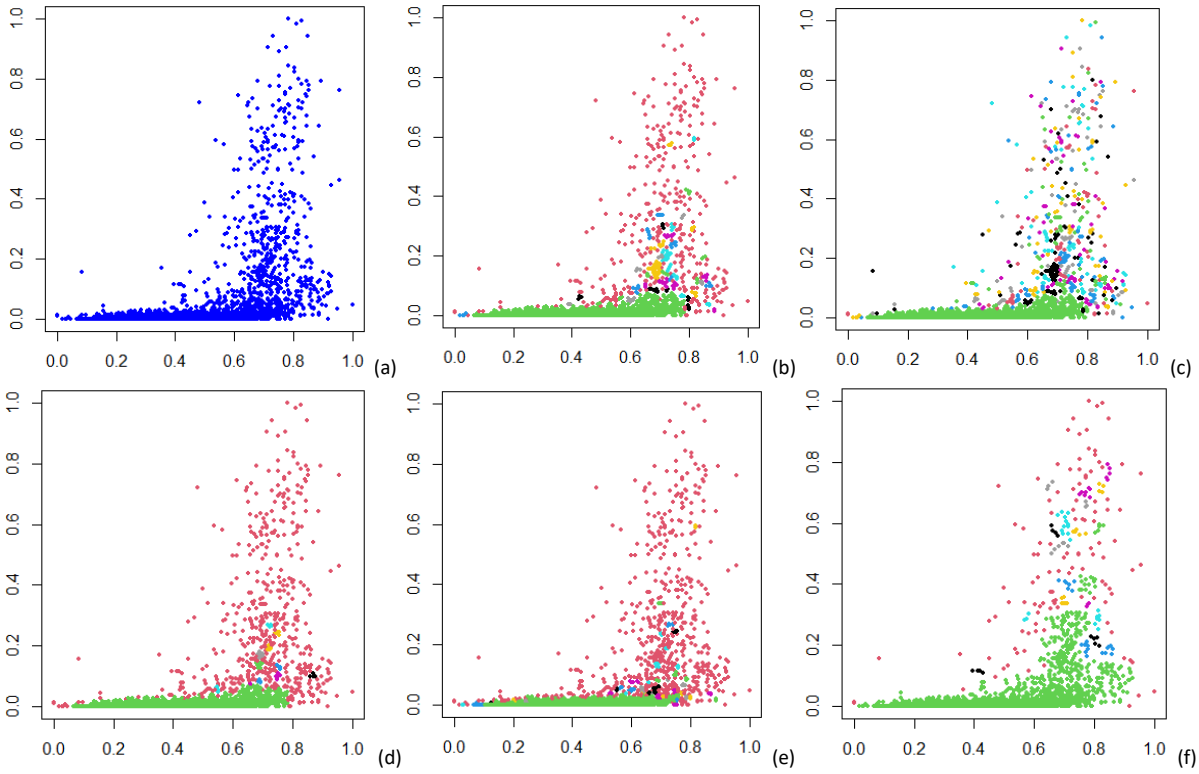


Slika 7. Određivanje optimalnog *eps*-a za (a) sintetičku bazu *eps* = 1.43 i (b) izdvojena 2 parametra pedološke baze *eps* = 0.01

Dobijena globalna vrijednost *eps*-a će predstavljati gustinu najmanjeg klastera (slika 6 (b)), jer ako se uzme *eps* (slika 6 (e)) manje od globalne vrijednosti formiraće se veći broj manjih klastera i veliki broj podataka će biti identifikovan kao šum. U odnosu na grafik (b) uočava se da je zeleni klaster razbijen na dva manja klastera i 4 tačke šuma koje su zaokružene. Za razliku od *MinPts* slučaja, kod različitih vrijednosti *eps* parametra veličina zelenog klastera se primjetno mijenja, a razlog je što biranjem većeg *eps*-a obuhvata se veći radius susjedstva, time i svi podaci koji spadaju unutar njega. *MinPts* je samo povećanje broja susjeda koji se mogu pridružiti klasteru, a moraju se nalaziti unutar *eps*-a. Suprotno tome, veća vrijednost *eps* parametra (slika 6 (f)) može doći do spajanja više klastera u jedan veći ukoliko se nalaze

na rastojanju manjem od *eps* vrijednosti (zeleni klaster) i dodjeljivanjem tačaka šuma nekom od klastera. Na slici 6 (f) uočava se zeleni dominantni klaster, koji je obuhvatio više podataka u odnosu na grafik (b), uključujući i šum.

Prethodno korišćeni podaci su kreirani za ilustrovanje načina klasterizacije DBSCAN-a, koji se po prirodi razlikuje od *k*-medoids i fuzzy *k*-medoids algoritama. Za predstavljanje ostalih analiziranih algoritama će se koristiti podaci dati na slici 8 (a). Iz tog razloga i nad njima je primijenjen DBSCAN, a rezultati prikazani na slici 8 (b), (c), (d), (e) i (f) potvrđujući prethodno donešene zaključke. Optimalano *eps* je dobijeno na slici 7 (b). Rezultat klasterizacije DBSCAN algoritma za optimalno *eps* i *MinPts* = 3 je na slici 8 (b). Svaki klaster označen je različitom bojom. Crveno su tačke šuma. Među ovim podacima uočava se dominantni klaster na svim slikama prikazan zelenom bojom i formiran na mjestu najveće koncentracije uzoraka.



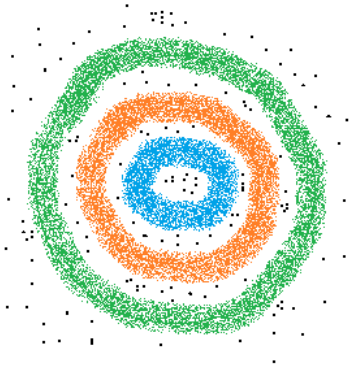
Slika 8. (a) Podaci o 2 pedološka parametra izdvojena iz baze i primjeri rezultata primjene DBSCAN-a za*:* (b) optimalno *eps* i *MinPts*, (c) optimalno *eps* i *MinPts* manje od optimalnog, (c) optimalno *eps* i *MinPts* veće od otpimalnog (e) optimalno *MinPts* i *eps* manje od optimalnog i (f) optimalno *MinPts* i *eps* veće od optimalnog.

Izbor *eps* je usko povezan sa izborom funkcije rastojanja od koje uveliko i zavise rezultati klasterizacije. Funkcija rastojanja predstavlja mjeru sličnosti između uzoraka koji pripadaju istom klasteru (uzorci se međusobno nalaze na manjoj udaljenosti, ako su u istom klasteru), odnosno mjeru različitosti uzoraka između svih klastera (veća udaljenost uzoraka koji pripadaju različitim klasterima). Uzorci koji pripadaju klasteru najbliži su medoidu tog klastera. Mjere sličnosti/različitosti su [14]: Manhattan rastojanje, Minkowski rastojanje, euklidsko rastojanje, kosinusna distanca, Pirsonova korelaciona distanca, distanca

Mahalanobis-a, itd. Funkcija udaljenosti se bira i u zavisnosti od tipa podataka nad kojima se primjenjuju data mining tehnike. U većini slučajeva, kao i u ovom radu, se koristi euklidsko rastojanje.

2.1.2 Prednosti i nedostaci DBSCAN algoritma

Jedna od prednosti DBSCAN algoritma je što ne zahtijeva definisanje broja klastera u koje treba grupisati podatke, pa nije neophodno dobro poznavanje baze koja se koristi. Formirajući klastere proizvoljnog oblika može se npr. pronaći klaster koji se nalazi unutar drugog klastera, u formi koncentričnih krugova [8], a da su nezavisni jedan od drugog (slika 9). Na mjestima gdje je koncentracija podataka mnogo manja primjećuju se tačke šuma (označene crnom bojom).



Slika 9. Jedan od načina klasterizacije podataka primjenom DBSCAN algoritma

Dovoljno znati *MinPts* vrijednost, koja je uvećana za 1 u odnosu na broj atributa podataka, da se *eps* vrijednost podesi (koristeći metodu lakta) da daje očekivanu klasterizaciju.

Nedostaci se ogledaju u tome što tačke koje se nalaze na granici klastera mogu pripadati većem broju klastera, ali se uvijek dodjeljuju onom klasteru koji je prvi "pronašao". U zavisnosti od redosljeda odabira proizvoljne tačke kao potencijalnog člana klastera i ispitivanja njenog susjedstva zavisi kojem će klasteru biti dodijeljena granična tačka. Ovo može dovesti do toga da formirani klaster sadrži manji broj tačaka od vrijednosti *MinPts*. Susjedstvo granične tačke se ne ispituje, jer ona nikad nema *MinPts* susjeda da bi se od nje formirao klaster, bez obzira kojem klasteru je prvo dodijeljena. Slučajnim odabirom granične tačke iz neodabranih tačaka i ispitivanjem njenog susjedstva bila bi proglašena šumom, sve dok se ne nađe u *eps* susjedstvu neke ispitivane tačke čijem će se klasteru dodijeliti.

2.2 *k*-medoids algoritmi klasterizacije

k-medoids algoritmi klasterizacije grupišu podatke dodjeljujući jedan uzorak samo jednom klasteru, pa pripadaju grupi tvrdih algoritama klasterizacije. Ime su dobili po

predstavniku klastera, medoidu, koji je najcentralniji uzorak posmatranog klastera. Kao i kod ostalih algoritama klasterizacije, cilj je vezati svaku tačku iz skupa podataka za najsličniji medoid.

2.2.1 CLARA

CLARA predstavlja proširenje PAM algoritma (engl. Partitioning Around Medoids) [9], [10]. PAM nasumično bira *k* medoida iz ulaznog skupa podataka, dodjeljujući im najsličnije uzorke. Za tako formirane klastere računa se funkcija cijene kao suma rastojanja svih uzoraka do njihovog najsličnijeg medoida. Cilj je naći skup medoida koji će smanjiti funkciju cijene. U sljedećem koraku se prolazi kroz skup medoida. Jedan po jedan medoid se redom mijenjaju sa slučajno odabranim neselektovanim uzorkom. Ako je poslije zamjene pojedinačnog medoida dobijena manja funkcija cijene onda taj skup postaje novi skup medoida i prelazi se na ažuriranje sljedećeg člana skupa medoida. U suprotnom, sljedeći član skupa medoida se ažurira. Postupak se ponavlja sve dok se ne postigne konvergencija funkcije cijene, odnosno dok nema promjena u članovima skupa medoida. Njegov nedostatak je primjenljivost samo na male skupove podataka. U cilju primjene na velike skupove podataka CLARA algoritam je predložen od strane Kaufman-a i Rousseeuw-a, 1986. godine u radu [11]. Broj medoida *k* oko kojih će se podaci grupisati je potrebno unaprijed odrediti.

CLARA je iterativna procedura. U svakoj od iteracija se slučajnim odabirom iz ulaznog skupa podataka *X* izdvaja manji skup podataka veličine 40+2*k*. Odabrani podskup opisuje ulazni skup podataka, omogućavajući primjenu CLARA algoritma nad velikim skupovima podataka. Iz tako selektovanog podskupa, primjenom PAM algoritma bira se *k* uzoraka, koji se proglašavaju trenutnim skupom medoida $V = \{v_1, v_2, \ldots v_k\}$. Preostali uzorci skupa *X* se klasterizuju tako što se svaki uzorak pridružuje klasteru koji je definisan sa njemu najbližim medoidom. Za svaki par uzorka i njemu odgovarajućeg medoida CLARA algoritam računa funkciju cijene u skladu sa formulom:

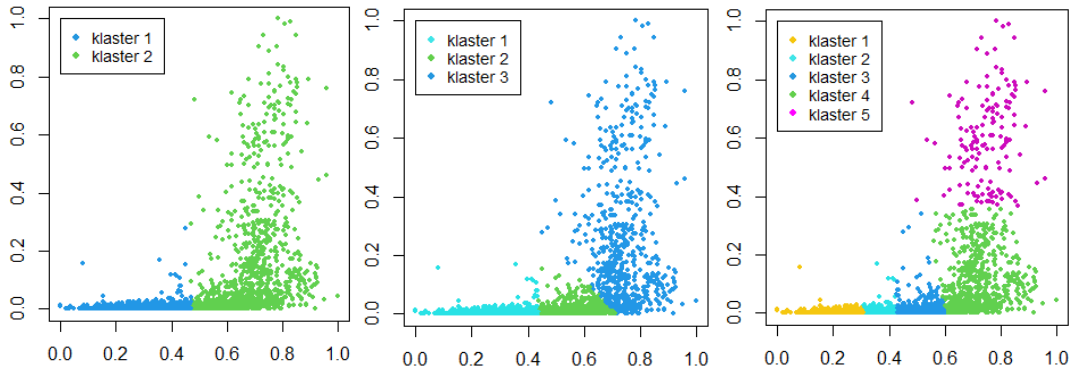
$$C = \sum_{i=1}^{n} \sum_{j=1}^{k} d(x_i, v_{ij})/n \qquad (2)$$

Gdje je *n* broj uzoraka u skupu *X*, a $d(x_i, v_{ij})$ rastojanje između posmatranog uzorka x_i i medoida v_{ij} kome pripada. Korišćena je euklidska funkcija rastojanja. Dobijena funkcija cijene je funkcija cijene za definisani skup medoida. Rezultat CLARA algoritma je skup medoida određen najmanjom funkcijom cijene za zadati broj iteracija.

Neka od prethodnih istraživanja pokazala su da je 5 iteracija dovoljno da se pronađe minimalna funkcija cijene i optimalni skup medoida [10]. Kompleksnost CLARA algoritma po svakoj iteraciji je $O(k^3+nk)$ [10].

Na slici 10 je dat primjer klasterizacije podataka predstavljenih sa dva pedološka parametra primjenom CLARA algoritma za različit broj klastera. Svaka boja označava zaseban klaster. Dodatno se može primijetiti podjela podataka prvenstveno duž *x* ose, što je posljedica neravnomjerne raspodjele podataka u prostoru, i toga što su podaci usko koncentrisani oko

nulte tačke *y* ose i rasprostranjeni cijelom dužinom *x* ose. Za podatke u desnom dijelu grafika je obrnuta situacija, zbog čega je za veći broj klastera prisutna podjela duž *y* ose.



Slika 10. Rezultati primjene CLARA algoritma u k = 2, 3 i 5 klastera

Klasterizacija dobijena primjenom CLARA algoritma se razlikuje od one dobijene primjenom DBSCAN za optimalne vrijednosti *eps*-a i *MinPts*-a (slika 8 (b)). DBSCAN je formirao klastere na mjestima gdje je veća gustina podataka i identifikovao tačke šuma na mjestima gdje su podaci rijetko raspoređeni. CLARA ne identifikuje tačke šuma, a klasteri se formiraju bez obzira na gustinu raspodjele podataka, sa jasnim prelazima između različitih vrijednosti parametara zbog čega se smatra efikasiniji u klasterizaciji prostornih, pedoloških podataka.

2.2.2 CLARANS

CLARANS su kao metod klasterizacije predložili Raymond T. Ng and Jiawei Han, 2002. godine [10]. Predstavlja modifikaciju CLARA. Primjenljiv je na male i velike skupove podataka.

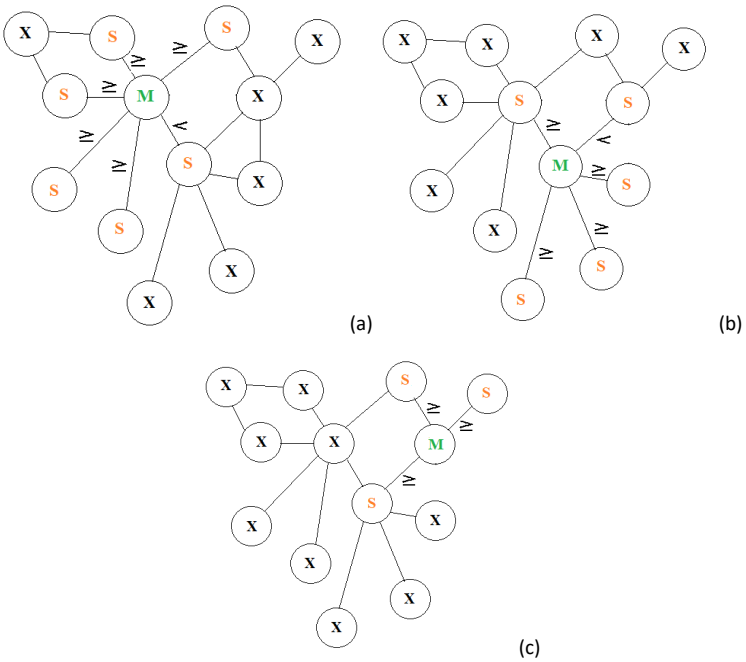
Neka je dat ulazni skup podataka $X = \{x_1, x_2, \ldots x_n\}$, veličine n i k broj klastera u koji se žele grupisati podaci. CLARANS algoritam u svakoj iteraciji nasumično bira k medoida iz skupa X (za razliku od CLARA koja odabir skupa medoida ograničava na $40+2k$ uzoraka). Nakon inicijalizacije trenutnog skupa medoida, u svakoj iteraciji postoji tačno *maxneighbor* = *max* (1.25%k(n-k), 250) susjeda [10] koji se trebaju ispitati kao alternativni skupovi medoidi. Susjede čine svi skupovi od k uzoraka koji se od trenutnog skupa medoida razlikuju za samo jedan element. Nasumično se bira jedan od tih susjeda trenutnog skupa medoida sve dok se ne ispitaju svi susjedi, a da ne dođe do promjene skupa medoida ili dok se ne naiđe na susjeda sa manjom funkcijom cijene. U drugom slučaju, ispitivani susjed postaje trenutni skup medoida i procedura se ponavlja za njegove susjede. U slučaju kada trenutni skup medoida ima manju funkciju cijene od svih njegovih susjeda, procedura se završava i trenutni skup medoida se proglašava najboljim rješenjem i lokalnim minimumom trenutne iteracije. Krajnji rezultat CLARANS algoritma je skup medoida sa najmanjom vrijednošću funkcije cijene za prethodno definisani broj iteracija. Na osnovu maksimalnog broja susjeda koji je potrebno ispitati u jednoj iteraciji, zaključuje se da izvršavanje ovog algoritma zahtijeva dosta vremena. Kompleksnost CLARANS-a po svakoj iteraciji iznosi O(n) [10]. Na osnovu ove vrijednosti može se zaključiti da je CLARA kompleksnija.

Funkcija cijene se definiše slično kao i kod CLARA algoritma:

$$C = \sum_{i=1}^{n} \sum_{j=1}^{k} d(x_i, v_{ij}) \quad (3)$$

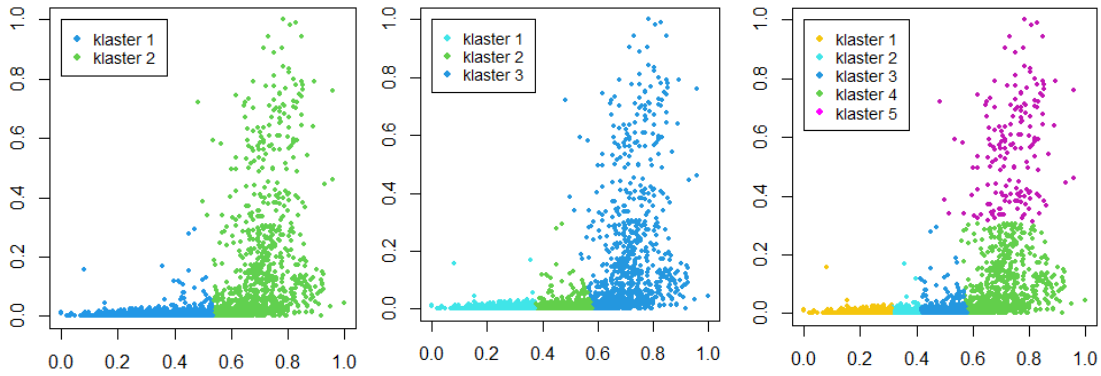
gdje $d(x_i, v_{ij})$ označava rastojanje između uzorka x_i i odgovarajućeg medoida v_{ij}.

Slika 11 ilustruje princip ažuriranja medoida kod CLARANS algoritma, gdje je sa *M* označen trenutni skup medoida, sa *S* susjedi kao potencijalni skupovi medoida, dok *X* predstavljaju ostali podskupovi ulaznog skupa podataka koji imaju više od jednog različitog uzorka u odnosu na trenutni skup medoida i zato ne predstavljaju susjede trenutnog skupa medoida *M*. Oznake navedene na linijama koje povezuju susjede sa trenutnim skupom medoida pokazuju koji od susjeda ima manju, a koji veću-jednaku funkciju cijene od trenutnog skupa medoida. *M* (slika 11 (a)) se ažurira sa susjedom koji ima manju funkciju cijene, slika 11 (b), nakon čega se vidi da se taj skup medoida *M* ažurira sa novim susjedom datim na slici 11 (c), potom se ispituju njegovi susjedi dok se ne ispita maksimalan broj susjeda ili dok se ne nađe novi susjed sa manjom funkcijom cijene. Na slici 11 (c) su predstavljeni takvi susjedi da svi imaju funkciju cijene veću od trenutnog skupa medoida *M*. Ako je ispitan maksimalan broj susjeda, onda trenutni skup medoida *M* postaje najbolje rješenje te iteracije, odnosno lokalni minimum. U suprotnom se traži susjed koji ima manju funkciju cijene od trenutne funkcije cijene. Procedura se ponavlja za definisani broj iteracija.



Slika 11. Proces ažuriranja medoida kod CLARANS algoritma

Primjer klasterizacije podataka CLARANS algoritmom nalazi se na slici 12.



Slika 12. Rezultati primjene CLARANS algoritma u *k*=2, 3 i 5 klastera

Klasterizacije dobijena primjenom CLARANS algoritma (slika 12) gotovo su identični sa rezultatima klasterizacije CLARA algoritma (slika 10). S obzirom da su algoritmi zasnovani na sličnom principu, dobijeni rezultati su očekivani. Iz tog razloga, pri poređenju rezultata klasterizacije CLARANS-a i DBSCAN algoritma (slika 8), važe isti zaključci.

2.2.3 Prednosti i nedostaci *k*-medoids algoritama klasterizacije i njihovo poređenje

Jedan od glavnih nedostataka kod *k*-medoids algoritama je to što je potrebno definisati broj klastera u koje se žele klasterizovati podaci. Za razliku od *k*-means algoritama, gdje je predstavnik klastera – centroid predstavljen kao srednja vrijednost sume svih uzoraka koji pripadaju posmatranom klasteru, medoidi su reprezenti ulaznog skupa podataka. Kako šum i izuzeci mogu ući u računanje centroida, kod medoida, kao uzrocima iz baze, se prevazilazi taj nedostatak. Iz toga proizilazi da su *k*-medoids robusniji na šum i izuzetke u odnosu na *k*-means algoritme.

Kao što je ranije navedeno, CLARANS predstavlja proširenje CLARA algoritma. Ova dva algoritma klasterizuju podatke po istom principu, tako da su rezultati klasterizacije slični. CLARANS algoritam postiže iste ili bolje rezultate za veće skupove podataka u odnosu na CLARA-u. Razlog je što započinje proizvoljnim skupom medoida, nasumično generišući i upoređujući susjede u svakom koraku algoritma. CLARA u svakoj iteraciji nasumično izvlači manji podskup, fiksne veličine 40+2*k* podataka, iz ulaznog skupa podataka i koristeći PAM algoritam pronalazi skup medoida iz posmatranog podskupa, ograničavajući pretragu na određeno područje. CLARANS je manje kompleksan od CLARA algoritma.

2.3 Fuzzy *k*-medoids algoritimi

Fuzzy *k*-medoids kao “meki” algoritmi klasterizacije, formiraju klastere sa nejasnim, blago zamućenim granicama. Odatle potiče sam naziv ovih algoritama (engl. fuzzy - zamućen, nejasan).

Ulazni parametar fuzzy algoritama **je stepen zamućenosti** (*m* – engl. fuzzifier**)** koji definiše nivo "zamućenosti" klastera. Njegova vrijednost se kreće u intervalu [1, ∞]. U nekim od ranijih istraživanja [13], [16], pokazano je da je za *m* najbolje uzeti vrijednost između 1.5 i 2. Što je ovaj broj bliži 1 to su klasteri bolje definisani (slika 13 (a)), granice između klastera jasnije pa algoritmi daju rezultate približno jednake *k*-medoids algoritmima. Dat je primjer za jedan klaster na slici 13 (a). Tamno crvenom su označeni uzorci koji imaju najveću pripadnost posmatranom klasteru, dok su tamno plavom označeni uzorci sa najmanjom pripadnošću, tj. uzorci sa većom pripadnošću drugim klasterima. Pripadnost klasteru (data na *color bar*-u) je zavisna od odabira stepena zamućenosti (formula (4)), što je objašnjeno u nastavku. Ostale boje u skladu sa *color bar*-om su uzorci koji se nalaze na granicama klastera i imaju približno jednake pripadnosti za više klastera, zbog čega su i granice zamućenije. Za srednju vrijednost *m* iz intervala [1.5, 2] prelazi između klastera su blago zamućeni, slika 13 (b). Za veće vrijednosti *m* klasteri se gotovo miješaju, što se vidi na slici 13 (c), jer vrijednosti stepena pripadnosti uzoraka klasterima su približno jednake za sve uzorke. Ovo rezultira loše rezultate klasterizacije.

Stepen pripadnosti igra ključnu ulogu u klasterizaciji podataka kod fuzzy algoritama. Svaki uzorak se djelimično dodjeljuje svakom klasteru. Djelimično dodjeljivanje je definisano stepenom pripadnosti klasteru i ima vrijednost u intervalu [0,1] – kao što je predstavljeno na *color bar*-u. Na konačno formiranje klastera utiču svi uzorci, u zavisnosti od stepena pripadnosti posmatranom klasteru.

Neka je $X = \{x_1, x_2, \ldots x_n\}$ dati skup od *n* uzoraka. $V = \{v_1, v_2, \ldots v_k\}$ je podskup skupa *X* sa *k* brojem uzoraka i predstavlja skup medoida. Neka je $d(x_i, v_{ij})$ rastojanje između uzorka x_i i medoida v_{ij}. U ovom radu stepen pripadnosti klasteru je definisan na sljedeći način [13]:

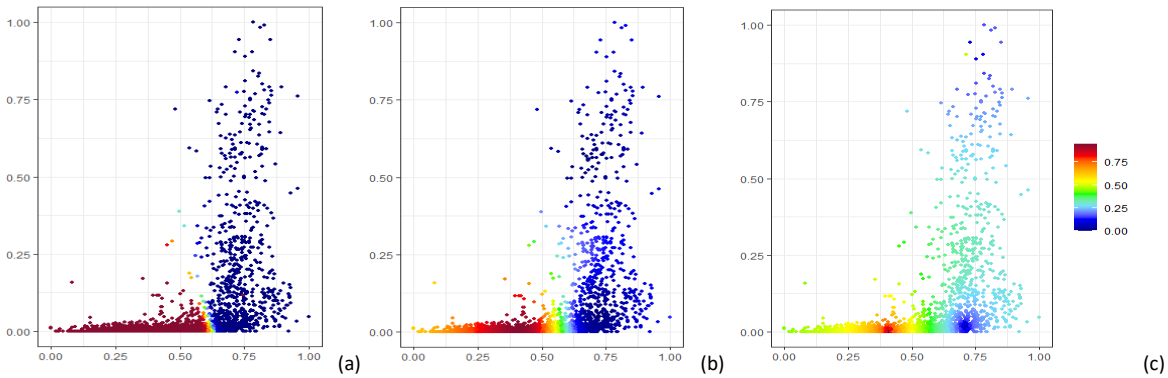
$$u_{ij} = ((1/d(x_i, v_{ij}))^{\frac{1}{m-1}}) / \left(\sum_{c=1}^{k} (1/d(x_i, v_{ic}))^{\frac{1}{m-1}}\right) \quad (4)$$

u_{ij} je stepen pripadnosti uzorka x_i klasteru v_{ij}, a $d(x_i, v_{ij})$ euklidsko rastojanje između uzorka x_i i medoida v_{ij}, odnosno medoida v_{ic}. U [17] se mogu naći ostali načini definisanja stepena pripadnosti.

Na slici 13 (c), na primjeru jednog klastera, pokazano je da za veće *m* većina uzoraka je označena zelenom i svijetlo plavom bojom, jer povećanjem vrijednosti *m* stepen pripadnosti uzoraka postaje približno jednak za sve klastere, u skladu sa *color bar*-om. Veoma mali broj uzoraka ima najveći (crveni uzorci) i najmanji (tamno plavi uzorci) stepen pripadnosti. Rezultat toga je miješanje klastera i manje pouzdani rezultati klasterizacije.

Cilj ovih algoritama je kao i kod prethodnih, naći skup medoida sa najmanjom funkcijom cijene. Kod *k*-medoids algoritama funcija cijene se računa kao suma rastojanja između medoida i njemu pridruženog uzorka. Kod fuzzy algoritama funkcija cijene je definisana kao suma proizvoda svih rastojanja između svakog od *n* uzorka do svakog *k* medoida i njihovih odgovarajućih stepena pripadnosti u_{ij}^m, što pokazuje formula (5) [13]:

$$C_m(V, X) = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij}^m \, d(x_i, v_{ij}) \quad (5)$$



Slika 13. Rezultati klasterizacije fuzzy *k*-medoids algoritama za jedan klaster i stepen zamućenosti: (a) $m < 1.5$, (b) $1.5 < m < 2$ i (c) $m > 2$. Dodjeljivanje uzorka klasteru je definisano stepenom pripadnosti klasteru i ima vrijednost [0,1] kao što je prikazano na *color bar*-u.

Fuzzy *k*-medoids algoritmi su implementirani u cilju dobijanja veće robusnosti na podatke šuma i izuzetke u odnosu na obične *k*-medoids. To je postignuto tako što šum i izuzeci, kao uzorci koji imaju male stepene pripadnosti svakom od klastera, nikada neće učestvovati u ažuriranju medoida, čime će se izbjeći i njihov loš uticaj na kvalitet klasterizacije. Drugim riječima, uzorak sa manjom vrijednosti stepena pripadnosti nikada neće biti izabran kao medoid, iz čega proizilazi da je mogućnost odabira šuma kao medoida umanjena.

2.3.1 RFCMdd

Ulazni argumenti RFCMdd algoritma su, pored podataka koji se klasterizuju, broj klastera u koji će se podaci grupisati, parametar zamućenosti fuzzifier *m*, broj *p* uzoraka koji imaju najveće članstvo u nekom klasteru, procenat podataka šuma i izuzetaka *h* (prag šuma i izuzetaka) i broj iteracija kroz koje se algoritam izvršava.

RFCMdd je iterativna procedura. Algoritam počinje inicijalizacijom trenutnog skupa medoida *V*. Na početku svake iteracije računa stepen pripadnosti svih uzoraka do svih medoida, nakon čega isključuje podatke koji prelaze definisani prag šuma i izuzetaka, kako bi bili izuzeti kao potencijalni medoidi. Medoidi se jedan po jedan ažuriraju sa jednim od *p* mogućih uzoraka koji imaju najveće članstvo u jednom klasteru, smanjujući tako funkciju cijene. Broj uzoraka *p* se uzima da je mnogo manja od veličine ulaznog skupa podataka i ne treba da bude veća od reda veličine klastera. Skup medoida dobijen na kraju jedne iteracije je ujedno inicijalni skup medoida sljedeće iteracije. Na taj način se kao novi medoidi biraju uzorci koji imaju manju funkcija cijene u odnosu na medoide odabrane u prethodnoj iteraciji, poboljšavajući tako kvalitet klasterizacije. Algoritam se završava za definisani broj iteracija ili kada je novo dobijeni skup medoida isti kao inicijalni skup medoida te iteracije.

Ranije je pomenuto da kod fuzzy *k*-medoids algoritama funkcija cijene predstavlja sumu proizvoda udaljenost između svakog uzorka i medoida i stepena pripadnosti posmatranog uzorka klasteru. Među uzorcima su prisutni podaci šuma i izuzeci, koji degradiraju kvalitet klasterizacije. Kako bi se prevazišao problem uticaja na klasterizaciju kod

RFCMdd algoritma je modifikovana funkcija cijene, tako da se "loši" podaci izuzmu pri ažuriranju medoida.

Zamjenom formule (4) u formulu (5) dobija se:

$$C_m(V, X) = \sum_{i=1}^{n} \left(\sum_{j=1}^{k} (d(x_i, v_{ij}))^{1/(1-m)} \right)^{(1-m)}$$

Gdje se:

$$\text{harm}(x_j) = \left(\sum_{j=1}^{k} (d(x_i, v_{ij}))^{1/(1-m)} \right)^{(1-m)} > h \qquad (6)$$

vrijednost uzima da bude veća od nekog zadatog praga h. Od odabira praga zavisi robusnost samog algoritma. Prag predstavlja procenat šuma i izuzetaka koji je očekivan među podacima i koji će se isključiti pri ažuriranju medoida.

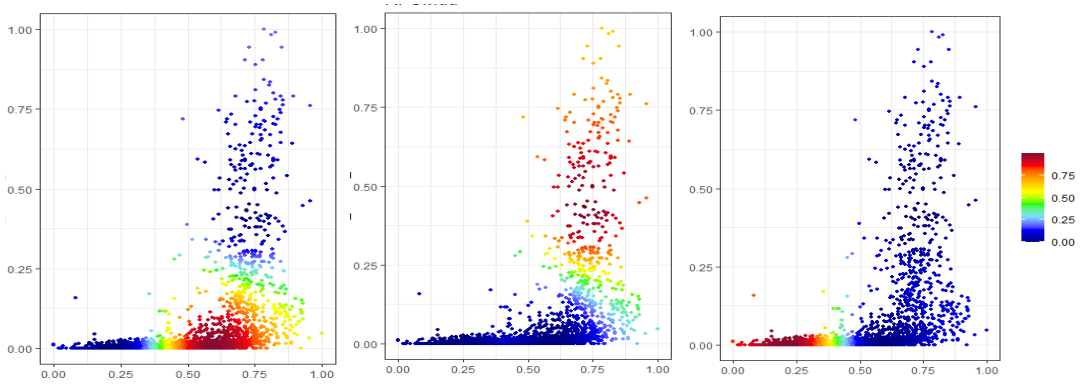
Redukcija ulaznog skupa podataka X, na ovaj način, doprinosi i smanjenju kompleksnosti RFCMdd algoritma. Kompleksnost algoritma iznosi (n log n) [13], [17]. Iz tako dobijenog redukovanog skupa kao potencijalni medoidi se uzimaju samo uzorci koji se nalaze u neposrednom okruženju trenutnog skupa medoida. Oni predstavljaju susjede. Za svaki član iz trenutnog skupa medoida izdvaja se p najbližih susjeda. Pri ažuriranju medoida, od p najbližih susjeda medoida (centra klastera j) pronalazi se pozicija njegovog susjeda x_k koji ima najmanju funkciju cijene (formula (7)), posmatrano za sve x_i uzorke redukovanog ulaznog skupa.

$$q = arg \min_{x_k \in X_{(p)_j}} \sum_{x_i \in h} u_{ij}^m \, d(x_k, x_i) \qquad (7)$$

gdje je m fuzzifier.

Postupak se ponavlja za zadati broj iteracija ili dok se ne postigne konvergencija skupa medoida, što podrazumijeva da se trenutni skup medoida nije promijenio u odnosu na prethodnu iteraciju.

Rezultati klasterizacije dobijeni primjenom RFCMdd algoritma na podatke predstavljene sa po dva pedološka parametra su data na slici 14.



Slika 14. Rezultati klasterizacije dobijeni primjenom RFCMdd algoritma za k = 3

Na svakom grafiku (slika 14) je prikazan po jedan klaster, označen crvenom bojom. Plavom bojom su dati uzorci sa najmanjom pripadnošću tom klasteru. Između njih se nalaze podaci koji imaju približne stepene pripadnosti u dva klastera. U odnosu na rezultate dobijene kod prethodno analiziranih *k*-medoids algoritama, vidi se da klasteri nijesu jasno definisani, dok se sličnost može uočiti u položaju klastera.

2.3.2 FCMRANS

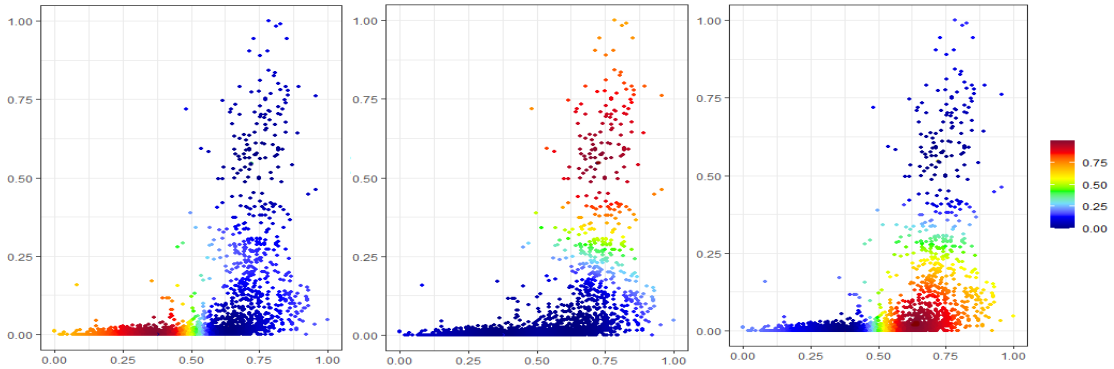
FCMRANS algoritam je, kao i RFCMdd, iterativna procedura sa istim ulaznim parametrima algoritma i razlikom u načinu ažuriranja medoida. Algoritam počinje inicijalizacijom skupa medoida i postavljanjem vrijednosti maksimalnog broja susjeda koji se ispituju kao potencijalni medoidi *maxneighbor = 1.25%k*(*n-k*) [13]. Susjede trenutnog skupa medoida predstavljaju potencijalni skupovi medoida koji se od trenutnog skupa razlikuju za samo jedan element. U svakoj iteraciji se prvo računa stepen pripadnosti svakog uzorka do svakog člana trenutnog skupa medoida, a zatim se isključuju podaci šuma iz ulaznog skupa *X* (u skladu sa formulom (6) i dato *h*). Dalje se nasumično bira jedan od *p* uzoraka (koji imaju najveće članstvo u klasteru čiji se medoid posmatra), koji će zamijeniti jedan, slučajno odabrani uzorak trenutnog skupa medoida. Na taj način se ispituju susjedi kao potencijalni skupovi medoida, sve dok se ne ispita *maxneighbor* susjeda, a da ne dođe do promjene skupa medoida, ili dok se ne naiđe na susjeda sa manjom funkcijom cijene, odnosno dok razlika funkcija cijene susjeda i funkcije cijene trenutnog skupa medoida nije manja od 0 (formula (8)). U slučaju pronalaska susjeda sa manjom funkcijom cijene, ispitivani susjed postaje trenutni skup medoida i procedura se ponavlja za njegove susjede. Ukoliko trentuni skup medoida ima najmanju funkciju cijene od svih njegovih susjeda, prelazi se na sljedeću iteraciju algoritma i procedura se ponavlja. Dobijeni skup medoida iz prethodne iteracije predstavlja inicijalni skup medoida nove iteracije, smanjujući na taj način funkciju cijene iz iteracije u iteraciju. Algoritam se izvršava za definisani broj iteracija ili dok se ne postigne konvergencija skupa medoida. Rezultat algoritma je skup medoida koji ima najmanju funkciju cijene kroz iteracije.

Razlika funkcije cijene za redukovani skup podataka veličine *S* računa se na sljedeći način:

$$E = \sum_{i=1}^{s} (d(x_i, x_q) - d(x_i, v_{ij}))\, u_{ij}^m \qquad (8)$$

gdje je $d(x_i, x_q)$ rastojanje *i*-tog uzorka x_i iz redukovanog skupa *X* do *q*-tog susjeda x_q, $d(x_i, v_{ij})$ rastojanje *i*-tog uzorka x_i iz redukovanog skupa *X* do *k*-tog medoida v_{ij} trenutnog skupa medoida, dok je u_{ij} stepen pripadnost *i*-tog uzorka *j*-tom medoidu, *m* je fuzzifier.

Na slici 15 je primjer klasterizacije dva pedološka parametra u tri klastera, primjenom FCMRANS algoritma. Klasteri su formirani na gotovo isti način kao kod RFCMdd algoritma, pa važe isti zaključci. U poglavlju sa rezultatima pokazano je da RFCMdd ima kraće vrijeme izvršavanja u odnosu na FCMRANS, bez obzira na broj iteracija.



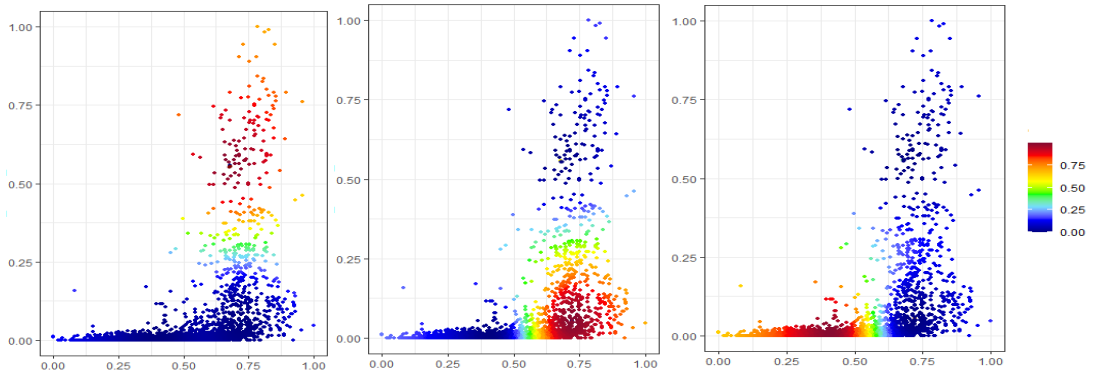
Slika 15. Rezultati primjene FCMRANS algoritma za $k = 3$

2.3.3 FCLARANS

FCLARANS je fuzzy oblik CLARANS algoritma, implementiran u cilju dobijanja veće robusnosti na podatke šuma. Što je fuzzifier m bliži 1, to je algoritam sličniji CLARANS-u.

Na početku algoritma se definiše maksimalan broj susjeda koji se računa formulom: $maxneighbor = 1.25\%k(n-k)$ [13]. Za dato X i k se inicijalizuje trenutni skup medoida V, na isti način kao kod RFCMdd i FCMRANS algoritama. Ostatak algoritma je sličan CLARANS-u, sa tom razlikom što FCLARANS samo kroz jednu iteraciju ažurira skup medoida, a funkcija cijene se računa u skladu sa formulom (5). Skup medoida će biti zamijenjen sa nasumično odabranim susjedom koji ima manju funkciju cijene od trenutnog skupa medoida. Dalje se ispituju njegovi susjedi. Dakle, FCLARANS algoritam se završava kada se ispita maksimalan broj susjeda, a da pri tome nije došlo do promjene trenutnog skupa medoida. Kao rezultat dobija se optimalan skup medoida.

Na slici 16 su predstavljeni klasteri na svakom grafiku odvojeno. Može se zaključiti da svi fuzzy *k*-medoids algoritmi klasterizaciju podatke na gotovo isti način i sa istim pozicijama klastera.



Slika 16. Rezultati klasterizacije FCLARANS algoritma za $k = 3$

2.3.4 Prednosti i nedostaci fuzzy *k*-medoids algoritama klasterizacije i njihovo poređenje

Nedostatak fuzzy *k*-medoids algoritama, kao i kod *k*-medoids, jeste potreba za unaprijed definisanim brojem klastera u koji se žele grupisati podaci. Osim broja klastera, kod RFCMdd i FCMRANS algoritama je potrebno odrediti procenat prisutnosti šuma među podacima.

Šum nema uticaj na ažuriranje medoida kod RFCMdd i FCMRANS algoritama. Medoidi se ažuriraju iz redukovanog ulaznog skupa podataka iz koga je eliminisan šum, pa je njegov uticaj na rezultat klasterizacije minimalan. Kod FCLARANS-a skup medoida se ažurira iz ulaznog skupa podataka, bez eliminacije „loših" podataka, što ga čini manje robustnim na šum u odnosu na RFCMdd i FCMRANS algoritme.

2.4 Određivanje optimalnog broja klastera kod *k*-medoids i fuzzy *k*-medoids algoritama

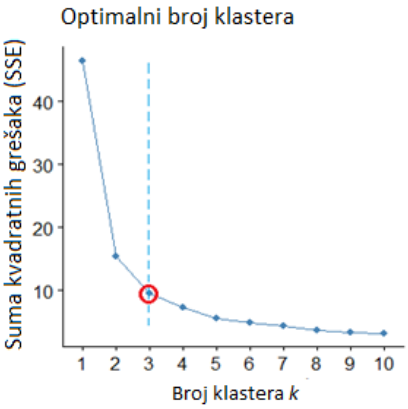
Svi analizirani algoritmi klasterizacije su osjetljivi na ulazne parametre algoritama. Da bi se obezbijedila uspješna klasterizacije potrebno je odrediti njihove optimalne vrijednosti koje će se primijeniti na cijeli skup podataka pri jednom pozivu algoritma. Takođe, za određivanje nekih od ulaznih parametara potrebno je poznavanje baze podataka nad kojom se klasterizacija primjenjuje.

Neke od metoda koje se kod *k*-medoids algoritama koriste za određivanje optimalnog broja klastera su metod lakta (engl. Elbow method) i metod srednje siluete (engl. Average Silhouette, tj. AS metod). Osim dvije navedene metode, postoji još mjera kvaliteta klasterizacije, među kojima su neke opisane u radu [19].

Elbow metod funkcioniše po principu sume kvadratnih grešaka. Za niz vrijednosti broja klastera *k*, primjenom *k*-medoids algoritma, sa kojim se žele klasterizovati podaci, nalaze se prototipovi skupova medoida. Najčešće se računa i analizira rastojanje za niz od 10 klastera. Za tako dobijene skupove medoida za svako *k* se formiraju klasteri od najsličnijih uzoraka i pronalazi se suma kvadratnih rastojanja. Suma kvadratnih rastojanja ili suma kvadratnih grešaka (engl. Sum of Squared Errors – SSE) je kvadratno rastojanje svih tačaka iz ulaznog skupa podataka do klastera kojem pripadaju i data je formulom (9) :

$$\text{SSE} = \sum_{j=1}^{k} \sum_{i=1}^{n} dist(x_i - v_{ij})^2 \qquad (9)$$

Drugi naziv za SSE je i funkcija cijene, pa se kaže da je SSE mjera kvaliteta rezultata klasterizacije [18]. Vizuelnim predstavljanjem se utvrđuje optimalan broj klastera. Na slici 17 je data zavisnost broja klastera u odnosu na sume kvadratnih grešaka. Klaster nakon koga SSE vrijednost više nema velikih, naglih promjena nego počinje da konvergira, proglašava se optimalnim brojem klastera za posmatrane podatke. Na slici 17 optimalan broj klastera je k = 3.



Slika 17. Određivanje optimalnog broja klastera primjenom metode lakta

Metod srednje siluete podataka je još jedna mjera kvaliteta grupisanja podataka. Računa se koeficijent srednje siluete (engl. Average Silhouette – AS) za različit broj klastera. Najčešće se za analizu uzima *k* od 1 do 10. Prvo se, kao i kod Elbow metode, primjenom *k*-medoids algoritama pronađu skupovi medoida za sve vrijednosti *k*. Za sve vrijednosti *k* uzorci se dodjeljuju najsličnijem medoidu i formiraju se prototipovi klastera. Dalje se posmatra jedan po jedan uzorak skupa podataka. Neka je x_i srednja vrijednost sume rastojanja *i*-tog uzorka do svih uzoraka njegovog klastera *x*, y_i srednja vrijednost sume rastojanja *i*-tog uzorka do svih uzoraka njemu najbližeg klastera *y*. Najbliži klaster predstavlja klaster koji ima najmanju srednju vrijednost sume svih udaljenosti *i*-tog uzorka do uzoraka koji pripadaju ostalim klasterima (ne uzimajući u obzir klaster *x* kome uzorak *i* pripada). Koeficijent siluete (S_i) se računa u skladu sa formulom (10) [20]:

$$S_i = (y_i - x_i)/max\,(x_i, y_i), \tag{10}$$

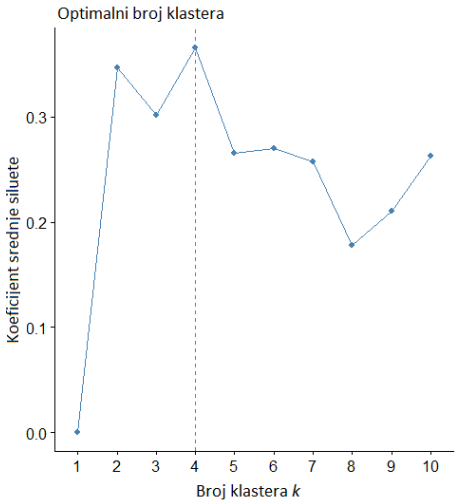
Koeficijent srednje siluete (AS) predstavlja prosječnu sumu koeficijenata siluete za svako *i*, *i*= 1, 2, ... *n* (11):

$$AS = \sum_{i=1}^{n} S_i/n \tag{11}$$

Optimalan broj klastera je onaj koji ima najveći koeficijent srednje siluete za niz mogućih *k* vrijednosti. Na slici 18 je prikazan grafik zavisnosti koeficijenta srednje siluete od broja klastera. Vrijednost koeficijenta srednje siluete nalazi se u intervalu između -1 i 1. Što je vrijednost AS za optimalni broj klastera veća i bliža 1, to će klasterizacija podataka biti bolja. Što su podaci više skoncentrisani oko medoida klastera, to je vrijednost ove funkcije veća, samim tim optimalni broj klastera će dati bolje rezultate. AS metod se može koristiti sa euklidskim rastojanjem ili Menhetn rastojanjem.

Determinisanje broja klastera kod fuzzy algoritama se razlikuje, zbog postojanja nejasnih granica između različitih klastera usljed dodjeljivanja uzoraka svakom od klastera sa različitim stepenom pripadnosti. Iz tog razloga nije moguće izračunati prosječne udaljenosti klastera po kome metoda srednje siluete i funkcioniše. U radovima [21] i [22] je predložena

modifikacija ove metode, tzv. fuzzy silueta, u cilju dobijanja optimalnog broja klastera kod fuzzy *k*-medoids algoritama.



Slika 18. Određivanje optimalnog broja klastera primjenom metode srednje siluete

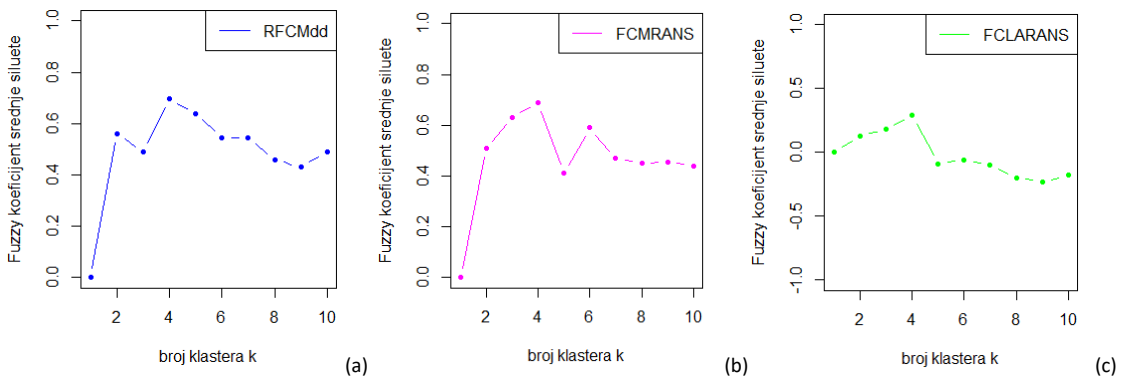
Fuzzy silueta (FS) je dizajnirana da razdvoji podatke između klastera koji se preklapaju na način što otkriva regije sa većom gustinom podataka. Smatra se da su ti podaci, inače smješteni u neposrednoj blizini prototipa klastera, najznačajniji. Manje značajni podaci su oni koji se nalaze u dijelu preklapanja klastera. *FS* je data sljedećom jednačinom:

$$FS = \sum_{i=1}^{n} ((u_{pi} - u_{qi})^{\alpha} s_i) \; / \; \sum_{i=1}^{n} (u_{pi} - u_{qi})^{\alpha}) \qquad (12)$$

gdje je S_i koeficijent siluete *i*-tog uzorka iz formule (10), u_{pi} i u_{qi} su prvi *p* i drugi *q* klaster u kojima *i*-ti uzorak ima najveću pripadnost. α je težinski koeficijent, $\alpha > 0$, a u slučaju da nije definisan podrazumijeva se da je $\alpha = 1$. Kada je vrijednost težinskog koeficijenta 0, tada vrijednost fuzzy siluete teži metodi srednje siluete (formula (11)). Povećanjem težinskog koeficijenta kod fuzzy siluete se smanjuje važnost uzoraka koji se nalaze u dijelu preklapanja klastera, a uzorci koncentrisani oko prototipova klastera dobijaju na značaju. Ovo je od važnosti kada je prisutna veća količina šuma među podacima.

Formula (12) predstavlja razliku stepena pripadnosti *i*-tog uzorka njegovom prvom i drugom klasteru u kojima ima najveću pripadnost. Na ovaj način uzorak koji je najbliži prototipu klastera dobija na važnosti u odnosu na uzorak koji se nalazi u regiji preklapanja klastera (gdje je stepen pripadnosti za minimum dva klastera sličan).

Vrijednost fuzzy siluete se kreće u intervalu od [-1, 1]. Kao i kod AS metode, pri analizi se uzima različit broj klastera *k*, npr. od 1 do 10, i za svaki do njih se nalazi vrijednost FS. Optimalan broj klastera je onaj koji ima najveću vrijednost fuzzy siluete za niz od *k* klastera. Na slici 19 su dati primjeri fuzzy siluete za RFCMdd, FCMRANS i FCLARANS algoritme. Optimalan broj klastera za sva tri algoritma je 4. Vrijednosti FS su uzete kao prosjek 3 poziva odgovarajućeg algortima.



Slika 19. Određivanje optimalnog broja klastera primjenom fuzzy siluete za: (a) RFCMdd, (b) FCMRANS i (c) FCLARANS.

Osim metode fuzzy siluete jedan od načina određivanja optimalnog broja klastera je računanje **koeficijenta podjele** (engl. Partition coefficient – PC), predložen u radu [23]. Neka je dat skup podataka *X*, koji se sastoji od *n* uzoraka. Slično kao kod prethodne dvije metode, za niz od *k* klastera (npr. od 1 do 10) primjenjuje se fuzzy *k*-medoids algoritam i na osnovu rezultata primjene algoritma za različito *k* izračunaće se stepeni pripadnosti svakog uzorka svakom od klastera (formula (4)). Iz dobijene vrijednosti u_{ij} pronaći će se koeficijent podjele prema formuli:

$$PC(k) = \sum_{j=1}^{k} \sum_{i=1}^{n} u_{ij}^2 / n \tag{12}$$

U radu [23] je predloženo poboljšanje koeficijenta podjele (engl. Improved Partition Coefficient - IPC). Poznato je da funkcija cijene opada sa povećanjem broja klastera. Kako se povećava broj klastera stepen pripadnosti uzoraka klasterima opada, tako da će i vrijednost koeficijenta podjele biti manji. Na osnovu toga, smatra se da koeficijent podjele *X* skupa u (*k*-1) klaster se dosta razlikuje od koeficijenta podjele *X* u *k* klastera, dok podjelom *X* u *k* klastera nema velike razlike u njegovoj vrijednosti u odnosu na podjelu *X* u (*k*+1) klastera. Razlika između dva susjedna klastera je data jednačinom:

$$r(k, k+1) = 100(PC(k) - PC(k+1))/ PC(k) \tag{13}$$

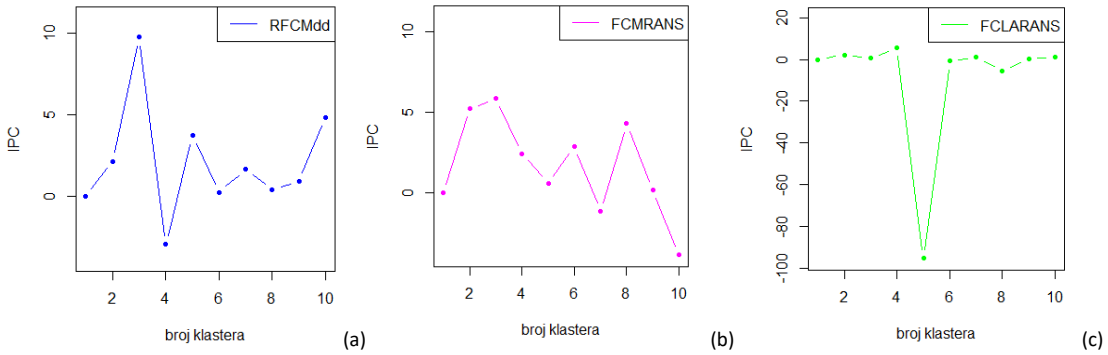
Poboljšani koeficijent podjele je:

$$IPC = r(k-1, k) - r(k, k+1) \tag{14}$$

Optimalni broj klastera za *k*= 1, 2, ... k_{max}, se dobija kao maksimalna vrijednost poboljšanog koeficijenta podjele.

$$k = arg \max_{1 < k < k_{max}} \{ IPC \} \tag{15}$$

Na slici 20 je grafik sa vrijednostima poboljšanog koeficijenta podjele za *k* = 1, ... 10, na osnovu koga se zaključuje da je optimalan broj klastera za RFCMdd i FCMRANS *k* = 3, dok je za FCLARANS optimalno *k* = 4 posmatrano za istu bazu. Za svaku vrijednost *k* klastera *PC* je dobijen kao prosjek za tri poziva algoritma, nakon čega je izračuto *IPC*. Razlika u broju klastera postoji zbog razlike u implementaciji algoritama i načinu na koji se ažuriraruju medoidi.



Slika 20. Određivanje optimalnog broja klastera primjenom koeficijenta podjele za: (a) RFCMdd, (b) FCMRANS i (c) FCLARANS.

2.5 Odabir broja iteracije – lokalnih minimuma

U ovom radu za broj lokalnih minimuma uzeto je *numlocal*=2. U radu [10] pokazano je da za *numlocal* = 2 postoji poboljšanje u klasterizaciji u odnosu kada bi se lokalni minimum tražio samo jednom, jer se nalaženjem drugog lokalnog minimuma dobija bolje rješenje ukoliko je prvi lokalni minimum odabran loše. Za *numlocal* > 2 nema velike razlike u kvalitetu klasterizacije podataka, a vrijeme izvršavanja algoritma se povećava. Iz tog razloga se smatra da je dovoljno naći dva lokalna minimuma da bi se postigla dobra klasterizacija.

2.6 Poređenje primijenjenih algoritama

k-medoids i fuzzy *k*-medoids algoritmi su bazirani na različitim principima u odnosu na DBSCAN algoritam, pa se podaci grupišu na drugačiji način. DBSCAN formira klastere proizvoljnog oblika na mjestima sa većom koncentracijom podataka. DBSCAN prepoznaje i tačke šuma, koje ne dodjeljuje nijednom klasteru. Nije u stanju da odvoji različite tipove zemljišta sa graničnim tačkama koje imaju slične karakteristike, jer nema jasnih granica između različitih parametara kod sličnih tipova zemljišta. Može se zaključiti da je DBSCAN pogodan za detekciju šuma među pedološkim podacima, ali ne i za klasterizaciju podataka.

Osnovna razlika *k*-medoids i fuzzy *k*-medoids algoritama je što *k*-medoids algoritmi imaju međusobno isključive klastere. Jedna od prednosti fuzzy *k*-medoids u odnosu na *k*-medoids algoritme jeste njihova veća robusnost na šum i izuzetke, jer odabir medoida zavisi od vrijednosti pripadnosti uzorka, kao potencijalnog medoida, klasteru. Za svaki od klastera, šum ima malu vrijednost funkcije pripadanja klasteru za fuzzifier izabran ≤ 2, tako da oni nikada neće biti odabran kao optimalni skup medoida. Fuzzy *k*-medoids algoritmi, zbog blagih prelaza koje prave između klastera, su pogodniji za klasterizaciju pedoloških podataka gdje nema naglih prelaza između različitih tipova zemljišta, kao i za klasterizaciju podataka kada su prisutni dominantni klasteri, što je slučaj sa korišćenim podacima.

Obje grupe algoritama su osjetljive na odabir inicijalnog skupa medoida, od koga zavise dobijeni optimalni skupovi medoida za svaki lokalnim minimumim. Na ažuriranje medoida kod fuzzy *k*-medoids algoritama utiču svi uzorci, ne samo oni koji pripadaju posmatranom klasteru

kao što je slučaj kod *k*-medoids klasterizacije. Za razliku od "običnih" *k*-medoids algoritama, računanje funkcije cijene kod fuzzy *k*-medoids oblika zavisi od pripadnosti uzoraka klasterima, kao i od stepena zamućenosti granica između klastera.

U tabeli 1 su objedinjene prethodno pomenute karakteristike za svaki analizirani algoritam pojedinačno.

	Karakteristike algoritama					
Algoritam	**oblik klastera**	**veličina baze**	**uticaj šuma i nedostajućih podataka**	**poznavanje baze**	**ulazni arugmenti algoritma**	**kompleksnost**
DBSCAN	proizvoljnog oblika	male i velike	identifikuje šum	ne	eps, MinPts	O(n log(n))
CLARA	sferni, konveksni	male i velike	nema	da	k	$O(k^3+nk)$ (*po iteraciji*)
CLARANS	sferni, konveksni	male i velike	nema	da	k	O(n) (*po iteraciji*)
RFCMdd	sferni, konveksni	male i velike	nema	da	k, fuzzifier, noise, Pobjects, itermax	O(n log(n)) (*po iteraciji*)
FCMRANS	sferni, konveksni	male i velike	nema	da	k, fuzzifier, noise, Pobjects, itermax	-
FCLARANS	sferni, konveksni	male i velike	nema	da	k, fuzzifier	-

Tabela 1. Karakteristike algoritama

3 Rezultati dobijeni primjenom algoritama klasterizacije na pedološkim podacima Crne Gore

Zemljište je jedna od osnovnih komponenti ekostistema. Čine ga organske i neorganske materije, voda i gasovi. Samo zemljište je površinski dio Zemljine kore, čiji kvalitet određuje uticaj brojnih faktora, biljnog pokrivača, mikroorganizma i životinja koje ga nastanjuju, brojnih materija koje su uključene u sastav zemljišta, klimatskih zona i vremenskih uslova, antorpogenog faktora, itd. Kako zemljište igra ključnu ulogu u poljoprivredi, samim tim u čovjekovoj ishrani, od velike važnosti je poznavati i tipove zemljišta određene teritorije. Poznavanje tipova zemljišta omogućava njegovo pravilno tretiranje i adekvatnu obradu, u cilju postizanja većih benefita, samim tim i većih prinosa kada je u pitanju proizvodnja hrane.

Svako zemljište se sastoji od više karakterističnih slojeva ili profila, koji predstavljaju hemijska i fizičko-mehanička svojstva zemljišta. Fizička svojstva zemljišta se odnose na njegov prirodni sastav: strukturu, poroznost, zapreminu, sastav, vlažnost zemljišta, vazduha, temperatura zemljišta, minerali. Mehaničko-fizička svojstva su materije koje ga čine: šljunak, pijesak, prah, glina, kamen, itd. Hemijska svojstva obuhvataju prisutnost hemijskih jedinjenja u zemljištu.

Crna Gora je zemlja u kojoj je zastupljeno nekoliko klimatskih regija, raznolika flora i fauna, što rezultira i heterogenost zemljišta. Postoji 7 zastupljenih tipova zemljišta, od kojih su 2 dominantna.

Digitalizovanu pedološku bazu Crne Gore, koja se koristi u ovom radu, čini preko 20.000 redova uzoraka i preko 200 kolona koji nose informacije o uzorcima, kao što su mehaničko-fizičke, hemijske karakteristike zemljišta, geografska širina, geografska dužina. Za svaki par koordinata vezuje se maksimalno po jedna vrijednost svakog od parametara iz baze.

Od svih karakteristika zemljišta dostupnih u bazi, u ovom istraživanju su se za klasterizaciju koristili sljedeći mehaničko-fizički:

- MP_clay_value – sadržaj gline,
- MP_sand_value – pijesak,
- MP_humidity_value – vlažnost zemljišta

i hemijski parametri:

- C_humus_value – prisustvo humusa,
- C_H2O_value – aktivna kisjelost, tj. pH vrijednost izvedena u vodi,
- C_CaCO3_value – prisustvo kalcijum karbonata,
- C_P2O5_value – prisustvo lakopristupnog fosfora,
- C_K2O_value – prisustvo lakopristupnog kalijum-oksida,

- C_KCl_value – pH vrijednost izvedena u kalijum-hloridu.

Za potrebe analize klasterizacije algoritama izdvojena su 2: *MP_clay_value* i *C_humus_value*, odnosno 3 parametra: *MP_clay_value*, *C_humus_value*, *C_K2O_value* iz baze.

Za dobijanje konačne pedološke tematske mape sa zastupljenim tipovima zemljišta korišćeno je 5 parametara (pH vrijednost izvedena u vodi (H_2O), kalcijum-karbonat – $CaCO_3$, humus, lakopristupni fosfor – P_2O_5 i lakopristupni kalijum-osid – K_2O) iz baze. Navedeni podaci nisu visoko korelisani, što je pokazano u [1].

U tabeli 2 su date minimalna, maksimalna vrijednost, prvi i treći kvartil, median i srednja vrijednost korišćenih parametara. *C_KCl_value* je jedini od izdvijenih parametara koji je visoko korelisan (u ovom slučaju sa *C_H2O_value*), što je dokazano u ranije sprovedenom istraživanju [1]. To je razlog njegovog isključenja pri primjeni analiziranih algoritama i dobijanju konačnih rezultata.

MP_sand_value	MP_clay_value	MP_humidity_value	C_H2O_value
Min. : 3.57	Min. :0.905	Min. : 0.020	Min. :0.460
1st Qu.: 39.97	1st Qu.:39.675	1st Qu.: 2.910	1st Qu.:5.520
Median : 50.70	Median :49.300	Median : 4.620	Median :6.120
Mean : 51.62	Mean :49.434	Mean : 4.923	Mean :6.213
3rd Qu.: 60.30	3rd Qu.:60.010	3rd Qu.: 6.692	3rd Qu.:6.900
Max. :4742.00	Max. :433.95	Max. :35.000	Max. :9.180

C_P2O5_value	C_KCl_value	C_humus_value	C_K2O_value
Min. : 0.000	Min. :0.406	Min. : 0.000	Min. : 0.00
1st Qu.: 1.100	1st Qu.:4.490	1st Qu.: 2.800	1st Qu.: 8.80
Median : 2.000	Median :5.150	Median : 5.120	Median :14.20
Mean : 4.002	Mean :5.236	Mean : 6.387	Mean :17.64
3rd Qu.: 3.500	3rd Qu.:5.980	3rd Qu.: 8.280	3rd Qu.:22.50
Max. :50.000	Max. :8.750	Max. :43.570	Max. :50.10

C_CaCO3_value
Min. : 0.000
1st Qu.: 0.000
Median : 0.000
Mean : 3.817
3rd Qu.: 1.260
Max. :88.410

Tabela 2. Parametri izdvojeni iz baze i korišćeni u ovom magistarskom radu

Maksimalne vrijednosti pojedinih parametara mnogo odstupaju od ostalih odgovarajućih vrijednosti usljed prisutnosti podataka šuma. Podaci šuma, koji svojim vrijednostima mnogo odstupaju od ostalih podataka su eliminisani. Uzorci kojima je vrijednost nekog od korišćenih parametara NA su eliminisani. Prije primjene algoritama analiziranih u ovom radu, urađena je još normalizacija podataka. Normalizacijom su vrijednosti parametara

dovedene u interval od 0 do 1. U nastavku se nalazi funkcija korišćena za normalizaciju (formula (16)).

$$Npodaci_i = (x_i - min(x))/(max(x) - min(x)) \quad (16)$$

Podaci korišćeni za dobijanje pedoloških mapa su dati u tabeli 3. Nakon eliminacije šuma i podataka sa NA vrijednošću parametara dobijeno je 6188 uzoraka za 5 izdvojenih hemijskih parametara. Izvršena je njihova normalizacija, a zatim primijenjeni analizirani algoritmi.

C_H2O_value	C_CaCO3_value	C_P2O5_value
Min. :0.0000	Min. :0.00000	Min. :0.00000
1st Qu.:0.5668	1st Qu.:0.00000	1st Qu.:0.01300
Median :0.6319	Median :0.00000	Median :0.02200
Mean :0.6411	Mean :0.03273	Mean :0.04131
3rd Qu.:0.7110	3rd Qu.:0.01188	3rd Qu.:0.03600
Max. :1.0000	Max. :1.00000	Max. :1.00000

C_humus_value	C_K2O_value
Min. :0.0000	Min. :0.0000
1st Qu.:0.0879	1st Qu.:0.1111
Median :0.1418	Median :0.1756
Mean :0.1768	Mean :0.2119
3rd Qu.:0.2439	3rd Qu.:0.2778
Max. :1.0000	Max. :1.0000

Tabela 3. Statističke veličine podataka koji su korišćeni za dobijanje pedoloških mapa

S obzirom da je za dobijanje pedološke mape Crne Gore korišćen samo dio podataka cjelokupne baze i da cijela teritorija Crne Gore nije pokrivena podacima, od velike važnosti je odabrati adekvatne tehnike klasterizacije, koje će uspješno klasterizovati podatke, prevazilazeći nedostatke među podacima. Jedino takvi algoritmi mogu omogućiti dobijanje rezultata uporedivih sa stvarnim stanjem. Što je više podataka dostupno to će i kvalitet klasterizacije biti bolji.

Primjenljivost analiziranih algoritama je predstavljena kroz primjere, rezultati su dati na graficima.

Implementacija algoritama, generisanje grafika i pedoloških dinamičkih mapa je rađeno u R programskom jeziku.

Prije primjene na pedološkim podacima Crne Gore, za predstavljanje rezultata analiziranih algoritama korišćena je Iris baza podataka, za koju su poznati broj i pozicija klastera. Čini je skup podataka o cvijetu šarenice koju je predstavio statističar i biolog Ronald Fisher 1936. godine. Baza obuhvata tri vrste šarenica (setosa, versicolor i virginica) sa po 50 uzoraka, a svaki je opisan sa četiri karakteristike: dužinom i širinom čašica (engl. Sepal length/width), i dužinom i širinom latica (engl. Petal length/width). Na osnovu ovih karakteristika, Fisher je razvio linearni diskriminativni model za razlikovanje ove tri vrste šarenica. Kako je poznato da postoje tri vrste šarenica, očekuje se da će se primjenom

analiziranih algoritama, na četiri karakteristike, identifikovati tri klastera tako da svaki predstavlja vrstu šarenica. Grupa setosa je linearno odvojiva od druge dvije vrste koje se preklapaju [24].

Rezultati klasterizacije analiziranih algoritama, uključujući *k*-means i fuzzy *k*-means, na normalizovane podatke iz Iris baze su dati u tabeli 4 i tabeli 5. Prva kolona je oznaka klastera: 1 (setosa), 2 (versicolor) i 3 (virginica) u zavisnosti od vrste šarenice kojoj uzorak pripada. U svakoj koloni tabele 4 je broj uzoraka po klasteru dobijenih primjenom navedenih algoritama. Zaključuje da su *k*-means, *k*-medoids i njihovi fuzzy oblici svi pravilno identifikovali prvi klaster. Kod preostala 2 klastera došlo je do greške prilikom pridruživanja uzoraka odgovarajućoj vrsti šarenica (klasteru). Razlog dobre identifikacije setosa šarenica je što je ova vrsta izlovana cjelina u odnosu na versicolor i virginica vrste koje se preklapaju u jednom dijelu.

U tabeli 5 je pregled broja uzoraka koji pri klasterizaciji nijesu pravilno dodijeljeni odgovarajućoj grupi šarenica. Prednost u tačnosti klasterizacije se može dati CLARA i FCLARANS algoritmima. Tabela 5 potvrđuje i bolju podjelu uzoraka primjenom CLARA i CLARANS algoritama u odnosu na *k*-means. Analizirani fuzzy *k*-medoids algoritmi su, takođe, uspješnije klasterizovali podatke u odnosu na fuzzy *k*-means. Slični rezultati za *k*-means i *k*-medoids algoritme su dobijeni u radu [24], gdje je isto prvi klaster identifikovan tačno, dok kod druga dva postoje podaci koji nijesu pravilno dodijeljeni. U radu [24] je potvrđena veća uspješnost klasterizacije Iris podataka sa *k*-medoids algoritmima (92%) u odnosu na *k*-means (88.7%), što se poklapa sa podacima u tabeli 5.

klaster	k-means	CLARA	CLARANS	RFCMdd	FCMRANS	FCLARANS	fuzzy k-means
1	50	50	50	50	50	50	50
2	61	47	55	52	44	44	60
3	39	53	45	48	56	56	40

Tabela 4. Klasterizacija IRIS baze primjenom *k*-means, *k*-medoids algoritama i njihovih fuzzy oblika

klaster	k-means	CLARA	CLARANS	RFCMdd	FCMRANS	FCLARANS	fuzzy k-means
1	0	0	0	0	0	0	0
2	3	7	4	6	10	8	4
3	13	4	9	9	4	2	14
uk.	16	11	13	15	14	10	18

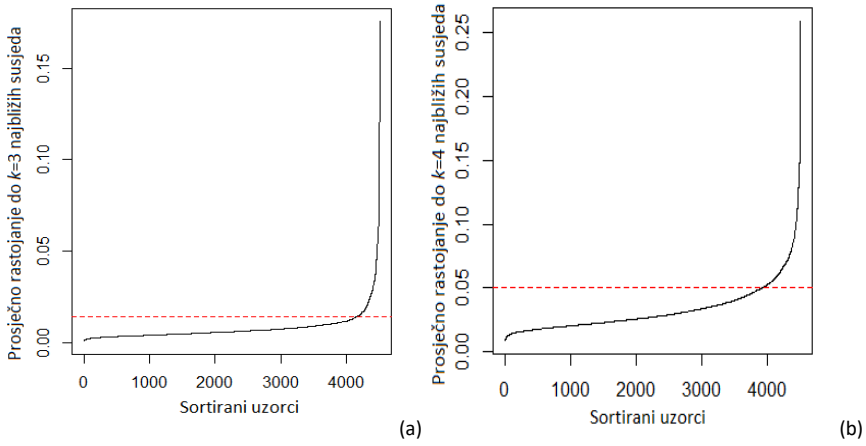
Tabela 5. Broj uzoraka po klasteru koji nije pravilno dodijeljen za svaki od algoritama

Primjenom DBSCAN algoritma na Iris bazu, za optimalne vrijednosti ulaznih parametara, dobijena su 2 klastera i podaci šuma. Prvi klaster, od 46 uzoraka sa 4 uzorka koja

su označena kao šum predstavlja setosa vrstu šarenice. Od preostalih 100 uzoraka, 73 je izdvojio kao drugi klaster i 27 uzoraka kao šum, što je i očekivano za DBSCAN s obzirom na prirodu algoritma i to da je setosa linearno odvojena cjelina u odnosu na versicolor i virginica vrste.

Prvi korak prije primjene svih algoritama, na dio pedološke baze Crne Gore, je pronalaženje optimalnih vrijednosti ulaznih parametara. Za analizu kroz primjere odabrana su dva i tri nekorelisana parametra iz pedološke baze kako bi se lakše ilustrovala primjenljivost analiziranih algoritama na pedološkim podacima i uporedili rezultati klasterizacije primijenjenih algoritama sa rezultatima dobijenim primjenom *k*-means i fuzzy *k*-means [1].

Kod DBSCAN algoritma za dva odabrana pedološka parametra uzeto je *MinPts* = 3. Korišćenjem ugrađene R funkcije *kNNdistplot*, dobija se grafik gdje je optimalna vrijednost *eps* susjedstva u tački presjeka crvene prave i krivulje i iznosi 0.014 (slika 21 (a)).



Slika 21. Određivanje optimalne *eps* vrijednosti primjenom *kNNdistplot* funkcije za (a) dva i (b) tri pedološka parametra

U tabeli 6 su dati rezultati primjene DBSCAN algoritma sa brojem uzoraka po klasterima i za različite vrijednosti *eps* susjedstva (optimalno *eps*, veće od optimalnog *eps* i manje od optimalnog *eps*). U sva tri slučaja se uočava dominantni klaster kome pripada neuporedivo veći broj uzoraka u odnosu na ostale manje klastere, što potvrđuju grafici na slici 22. Svaki klaster predstavljen je različitom bojom. Crvenom bojom su identifikovane tačke šuma. Za optimalno *eps* = 0.014 dominantni klaster je označen zelenom bojom (slika 22 (a)) i čini ga 3928 uzoraka (tabela 6). Ostali uzorci su podijeljeni u 47 manjih klastera i šum koji obuhvata 253 tačke koje ne pripadaju nijednom od klastera (crveni uzorci). Uzorci koji su sadržani u manjim klasterima nijesu označeni kao šum iz razloga što zadovoljavaju uslove optimalnog minimalnog broja tačaka *MinPts* = 3 koje treba da sadrži *eps* susjedstvo posmatrane tačke, da bi ona bila dodijeljena klasteru. Još jedan razlog tome je što pri kreiranju ovog klastera za optimalno *eps* i *MinPts*, posmatrane tačke u njenom susjedstvu imaju određeni broj graničnih tačaka koje su prethodno dodijeljene nekom drugom klasteru, pa nisu povezane sa posmatranim, novoformiranim klasterom.

***eps* = 0.014, *MinPts* = 3**	
Broj klastera	Broj uzoraka po klasteru
šum	253
klaster 1	**3928**
klaster 2	5
klaster 3	17
klaster 4	7
klaster 5	5
klaster 6	4
klaster 7	7
klaster 8	8
klasteri….	…..
klaster 48	3

***eps* = 0.035, *MinPts* = 3**	
Broj klastera	Broj uzoraka po klasteru
šum	34
klaster 1	**4428**
klaster 2	5
klaster 3	5
klaster 4	3
klaster 5	13
klaster 6	3
klaster 7	3

***eps* = 0.008, *MinPts* = 3**	
Broj klastera	Broj uzoraka po klasteru
šum	862
klaster 1	**2160**
klaster 2	6
klaster 3	7
klaster 4	92
klaster 5	8
klaster 6	3
klaster 7	20
klaster 8	3
klaster 9	4
klaster 10	3
klaster 11	11
klaster 12	5
klaster 13	11
klaster 14	9
klaster 15	7
klaster 16	28
klaster 17	19
klaster 18	17
klaster 19	8
klasteri....	.....
klaster 230	3

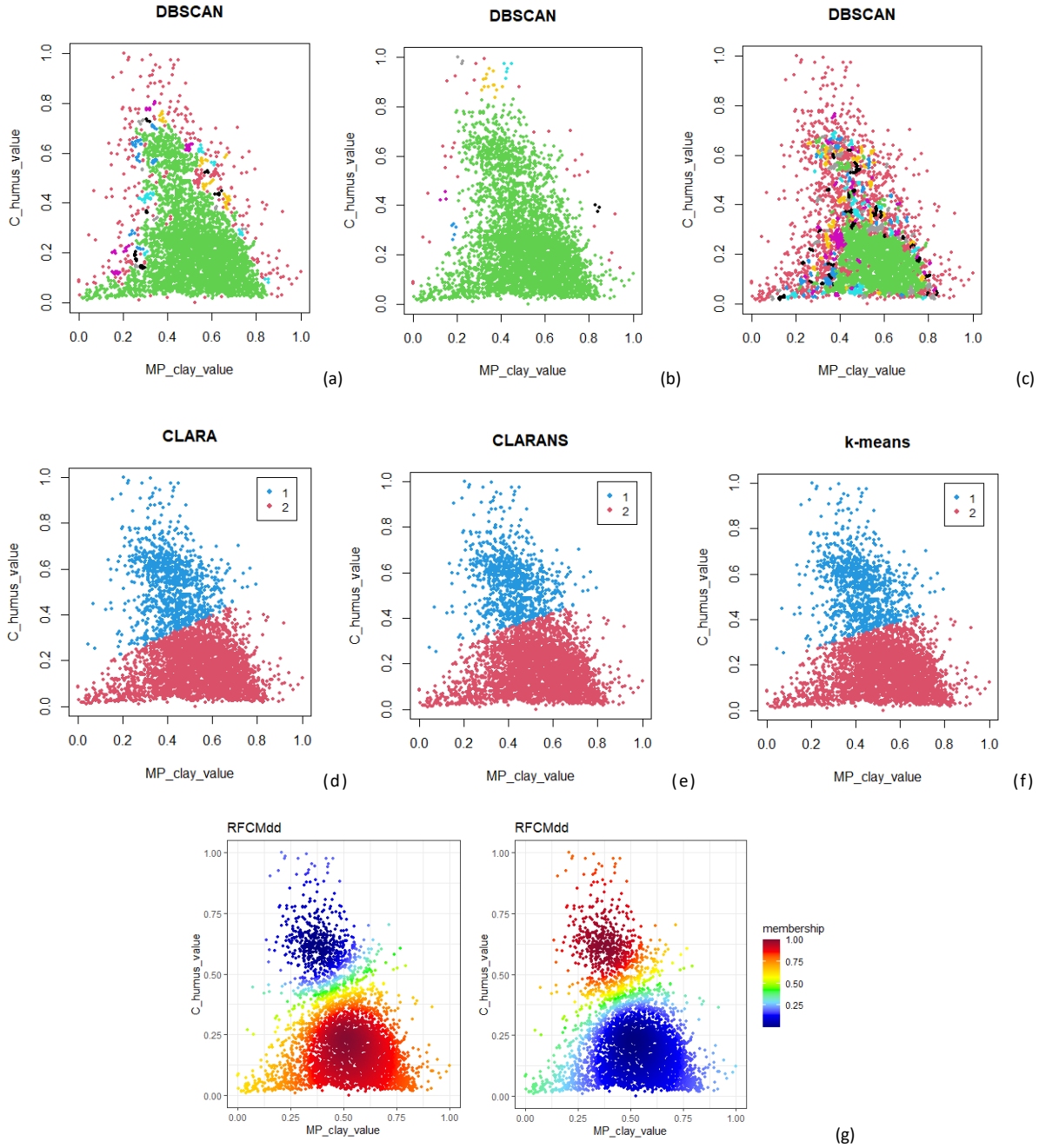
Tabela 6. Broj uzoraka za svaki od klastera, za različite vrijednosti ulaznih parametara

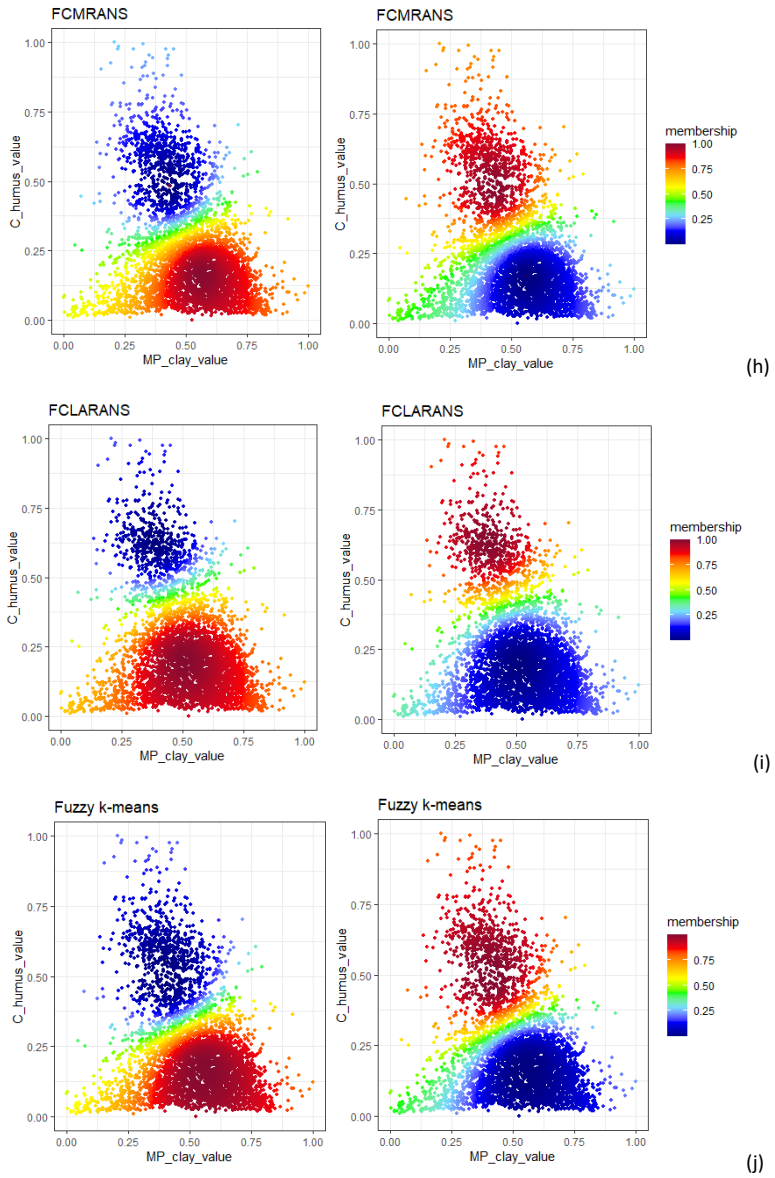
Na slici 22 (b) i (c) su dati rezultati klasterizacije dobijeni za *eps* = 0.035 i *eps* = 0.008. Njihovom analizom utvrđeno je da uzimajući *eps* vrijednost veću od optimalne broj dobijenih klastera opada, tj. konvergira ka jednom klasteru, dok smanjenjem *eps* u odnosu na optimalnu vrijednost broj klastera raste. Ovo potvrđuje broj uzoraka u svakom od klastera, dat u tabeli 6.

Korišćenjem globalnih vrijednosti ulaznih argumenata kod DBSCAN algoritma postoji mogućnost spajanja dva klastera različitih gustina raspodjele uzoraka u jedan veći klaster, ako su lokalizovani na rastojanju manjem od *eps* vrijednosti. Slika 22 (b) potvrđuje da je veći broj manjih klastera sa slike 22 (a) pridružen u jedan dominatan klaster (zelene boje).

Optimalan broj klastera CLARA algoritma je dobijen korišćenjem ugrađene R funkcije *fviz_nbclust* (baziranoj na AS metodi) i iznosi *k* = 2, za dva korišćena pedološka parametra (slika 23 (a)). Rezultat klasterizacije je na slici 22 (d). Vizuelno se zaključuje da je rezultat CLARA klasterizacije grupisanje međusobno najsličnijih uzoraka u iste klastere, uporedivih veličina, bez prisustva dominantnog klastera i šuma što je bio slučaj kod DBSCAN-a.

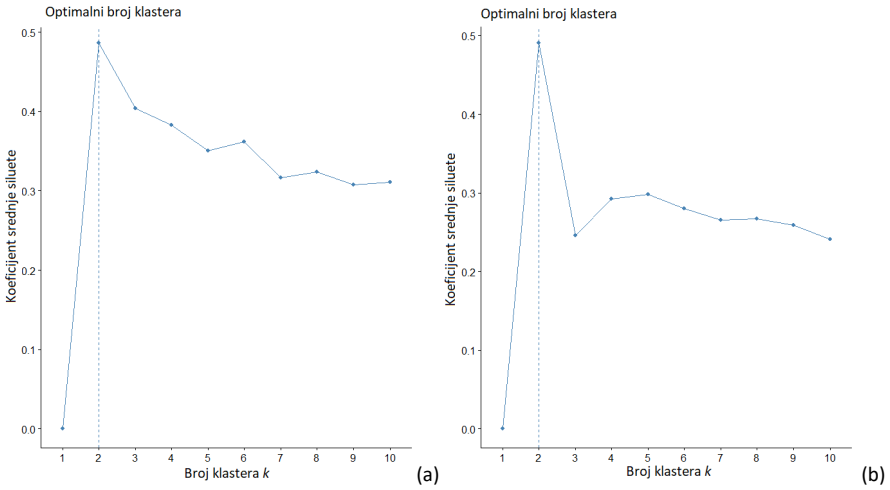
Za determinisanje optimalnog broja klastera kod CLARANS algoritma koristi se AS metoda. Radi lakšeg poređenja sa rezultatima primjene CLARA i ostalih analiziranih algoritama, uzorci će se grupisati u k = 2. Primjer klasterizacije CLARANS algoritma za dva pedološka parametra je na slici 22 (e). Uporednom analizom rezultata CLARA i CLARANS, potvrđeno je da je njihov princip klasterizacije isti. Kako CLARANS predstavlja samo optimizovanu verziju CLARA algoritma (u pogledu primjene na velikim skupovima podataka) ovakvi rezultati su bili i za očekivati.





Slika 22. Rezultati klasterizacije dobijeni primjenom algoritama: (a) DBSCAN-a za optimalno *eps*, (b) DBSCAN-a za *eps* veće od optimalnog, (c) DBSCAN-a za *eps* manje od optimalnog i (d) CLARA, (e) CLARANS, (f) *k*-means, (g) RFCMdd, (h) FCMRANS, (i) FCLARANS i (j) fuzzy *k*-means za *k* = 2

Kao što je ranije pomenuto, za broj iteracija kod CLARANS algoritma dovoljno je uzeti numlocal = 2 da bi klasterizacija bila uspješna. Veće poboljšanje u vrijednosti funkcije cijene se primjećuje za numlocal = 2 u odnosu na numlocal = 1. Za svako numlocal > 2 vrijednost funkcije cijene neznatno varira, zbog čega se smatra da je dovoljno naći dva lokalna minimuma da bi se dobila dobra klasterizacija - tabela 7.



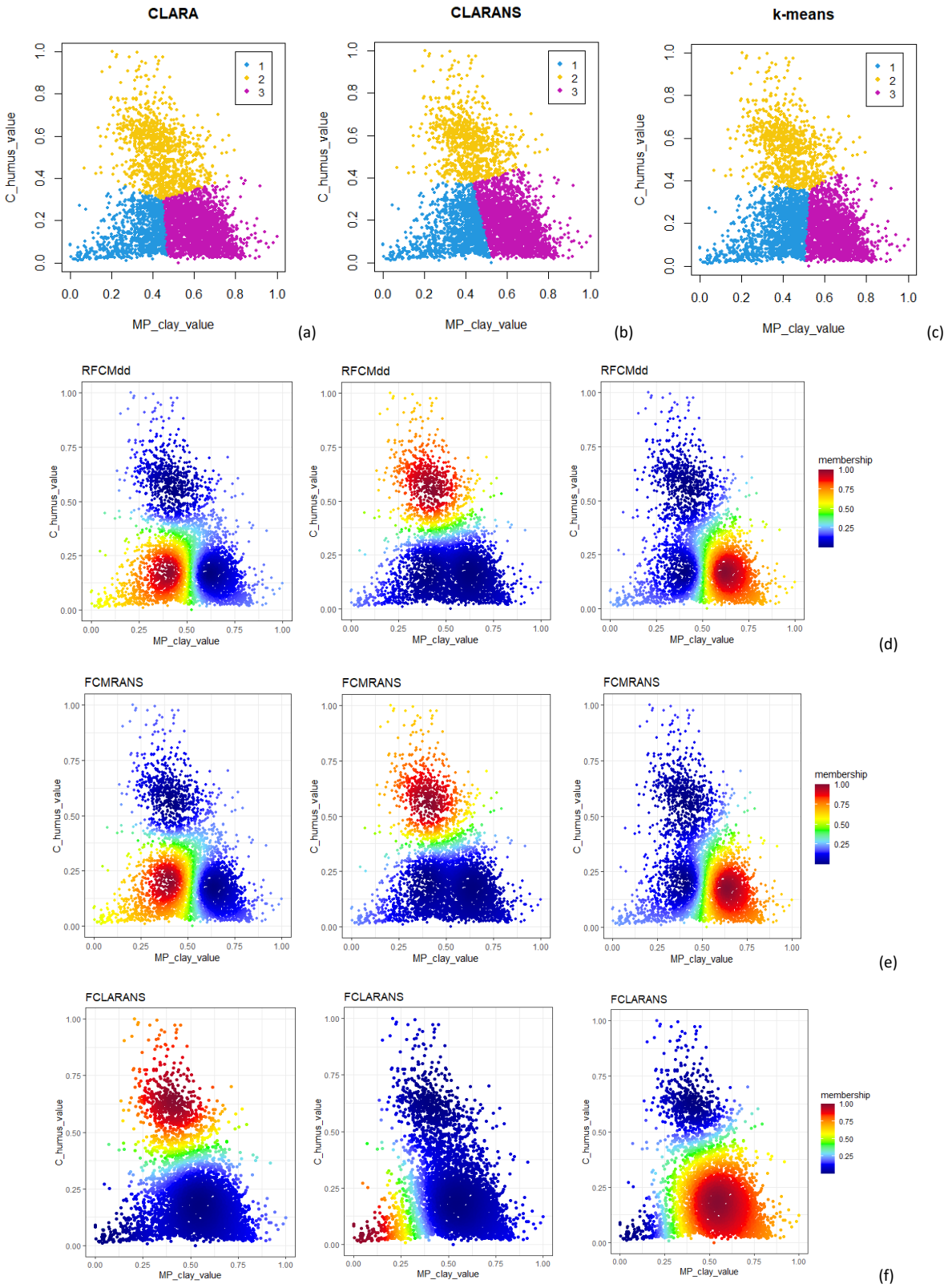
Slika 23. Određivanje optimalnog broja klastera za CLARA algoritam (a) dva i (b) tri pedološka parametra

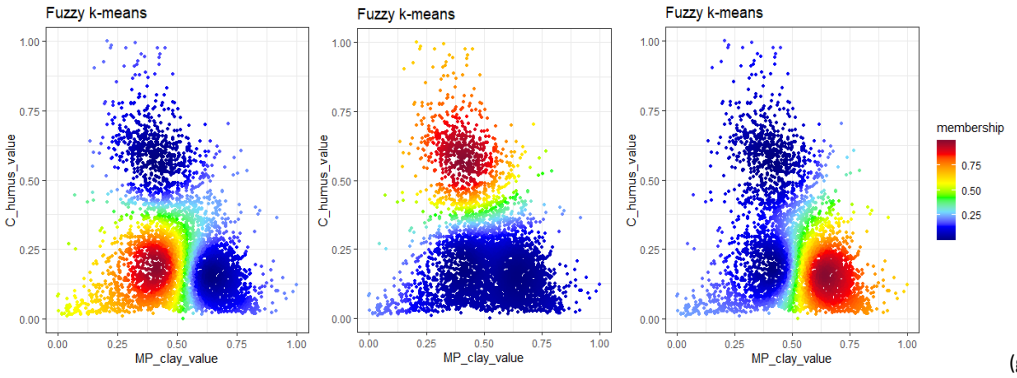
broj iteracija	**funkcija cijene za klasterizaciju**	
	3 klastera	**5 klastera**
1	0.55261	0.71821
2	0.45812	0.63773
3	0.45985	0.62998
4	0.44983	0.63211

Tabela 7. Zavisnost funkcije cijene u odnosu na broj lokalnih minimuma

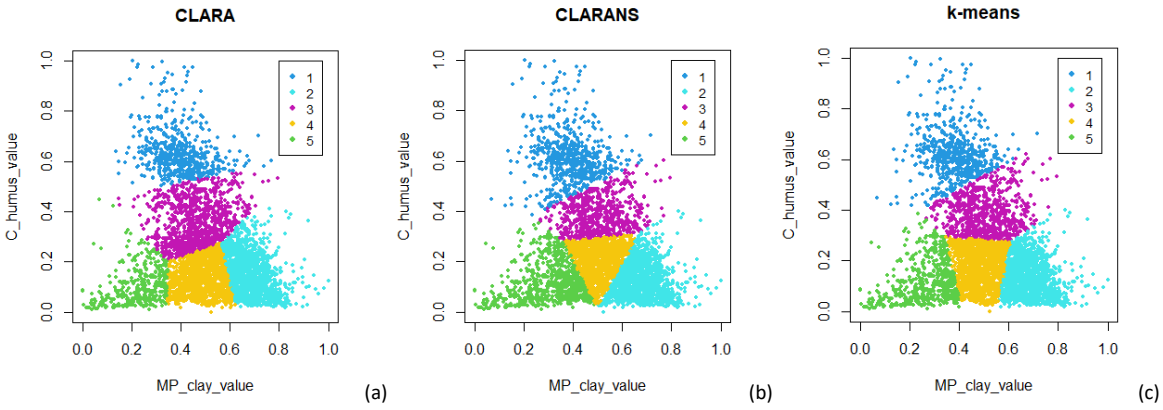
U ranijem istraživanju [1] predstavljena je primjena *k*-means algoritma klasterizacije nad istom pedološkom bazom Crne Gore. *k*-means klasterizacija ili klasterizacije *k*-srednjih vrijednosti predstavlja grupisanje sličnih podataka u iste klastere kao i kod *k*-medoids algoritama. Razlika je što kod *k*-means algoritama grupisanje se vrši oko centroida (centralne tačke klastera), koji se dobija kao srednja vrijednost rastojanja između svih uzoraka unutar tog klastera. Dakle, za razliku od medoida, centroidi nijesu članovi ulaznog skupa podataka. U njihovo računanje mogu ući i podaci šuma, što ih čini manje robusnim u odnosu na *k*-medoids. Obje vrste algoritama pripadaju algoritmima kod kojih su klasteri međusobno isključivi. Podaci se klasterizuju na sličan način, što se može potvrditi poređenjem dobijenih grafika (slika 22 (d), (e) i (f)). Ovo je za očekivati s obzirom da su *k*-medoids modifikacija *k*-means algoritama klasterizacije. Zbog veće robusnosti medoida na podatke šuma u odnosu na centroide, prednost se daje *k*-medoids algoritmima. Prednost *k*-medoids u odnosu na *k*-means algoritme potvrđeni su i u radovima [25], [26], [27].

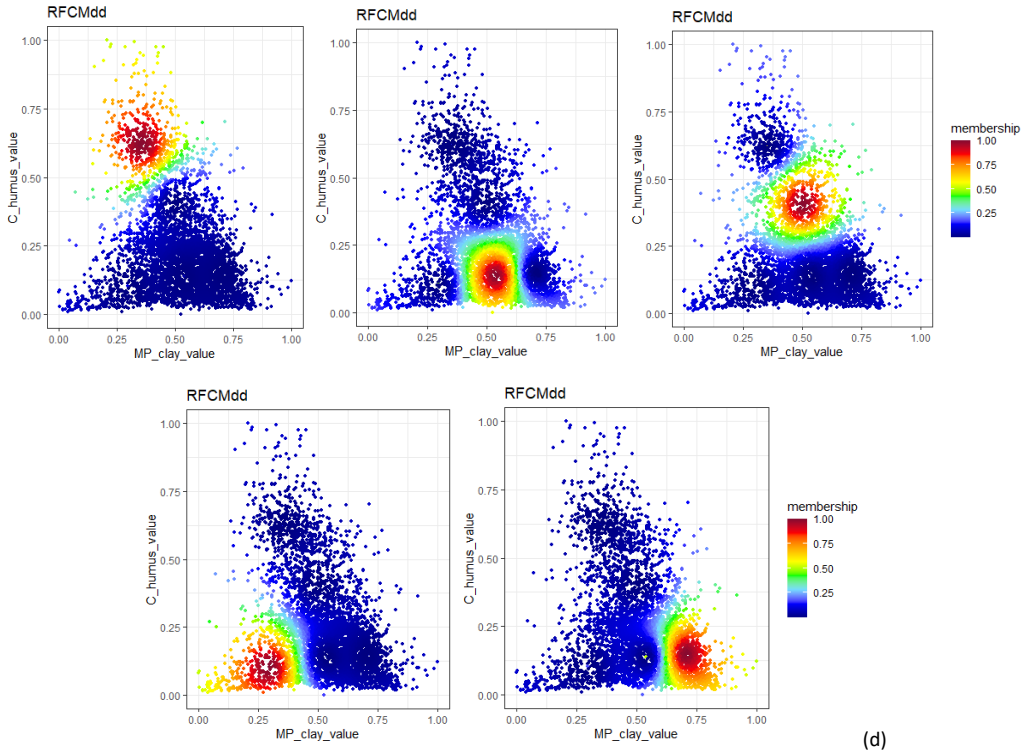
Na slikama 24, 25 i 26 ((a), (b) i (c)) su dati primjeri rezultata klasterizacije analiziranih *k*-medoids i *k*-means algoritama za $k = 3$, $k = 5$ i $k = 7$ klastera. Vidi se da promjena broja klastera dovodi do razbijanja većih klastera (npr. za $k = 3$) na veći broj manjih klastera (koji imamo u slučaju kada je $k = 5$ i $k = 7$), i dalje grupišući najsličnije podatke u isti klaster.

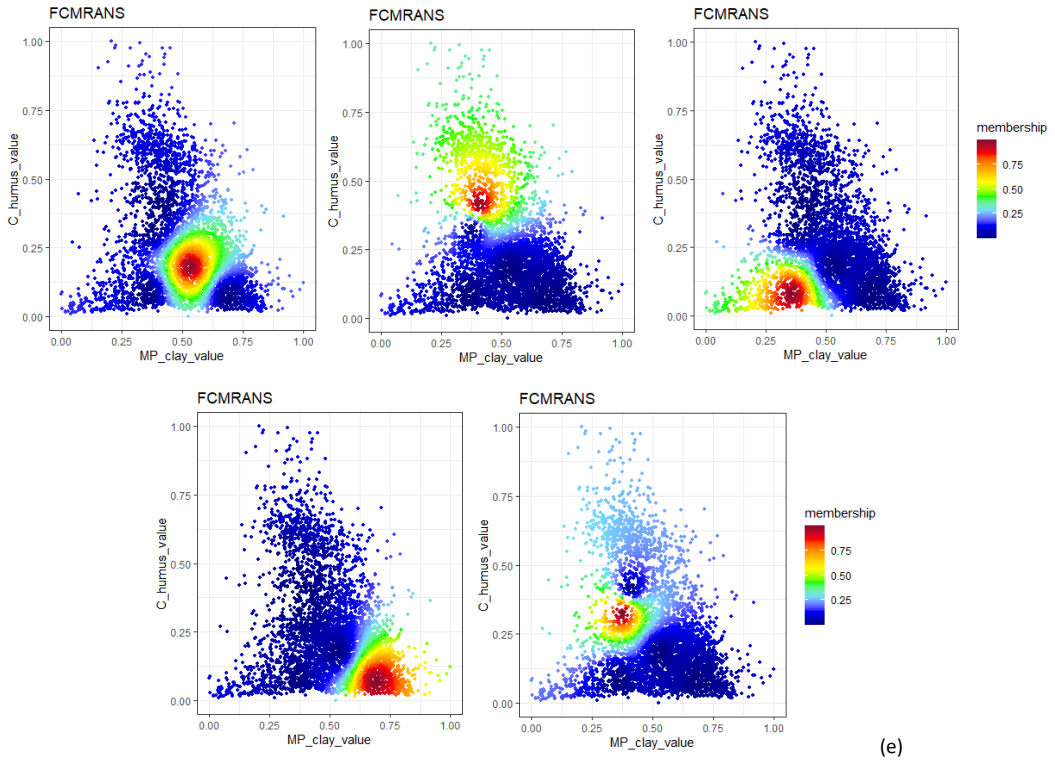

CLARA
CLARANS
k-means
C_humus_value
MP_clay_value
1
2
3
(a)
(b)
(c)
RFCMdd
membership
(d)
FCMRANS
(e)
FCLARANS
(f)



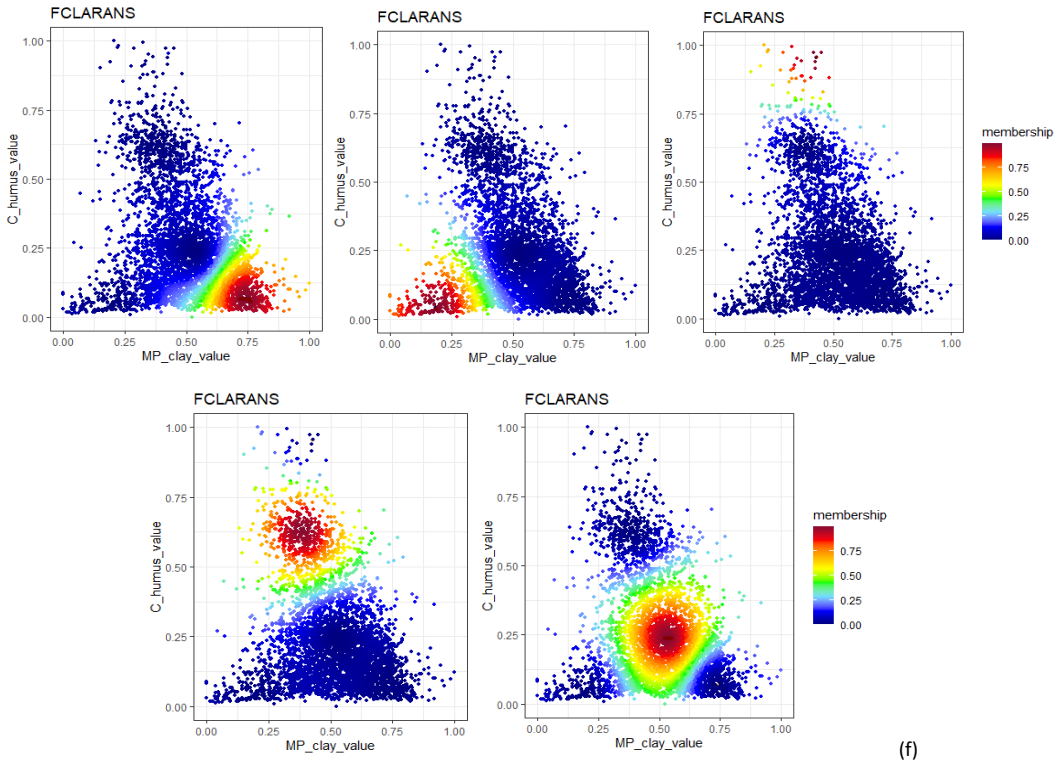
Slika 24. Rezultati klasterizacije dobijeni primjenom algoritama: (a) CLARA, (b) CLARANS, (c) *k*-means, (d) RFCMdd, (e) FCMRANS, (f) FCLARANS i (g) fuzzy *k*-means za *k* = 3



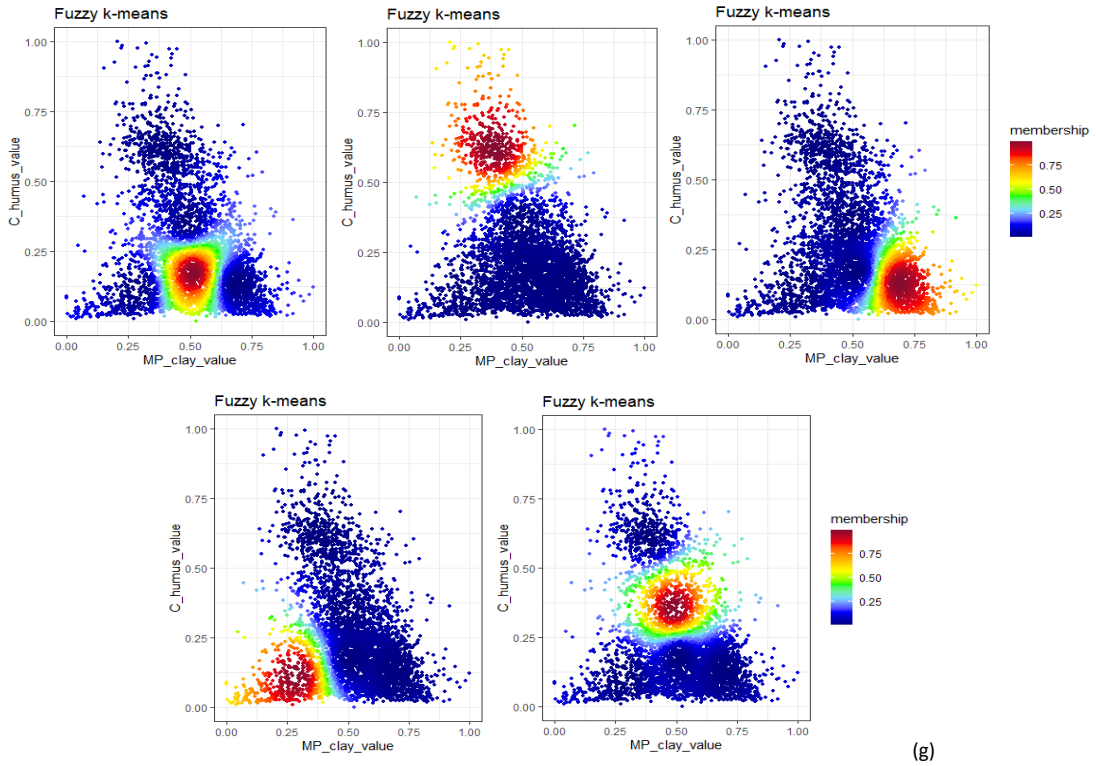



FCMRANS
FCMRANS
FCMRANS
FCMRANS
FCMRANS
C_humus_value
MP_clay_value
membership
0.75
0.50
0.25

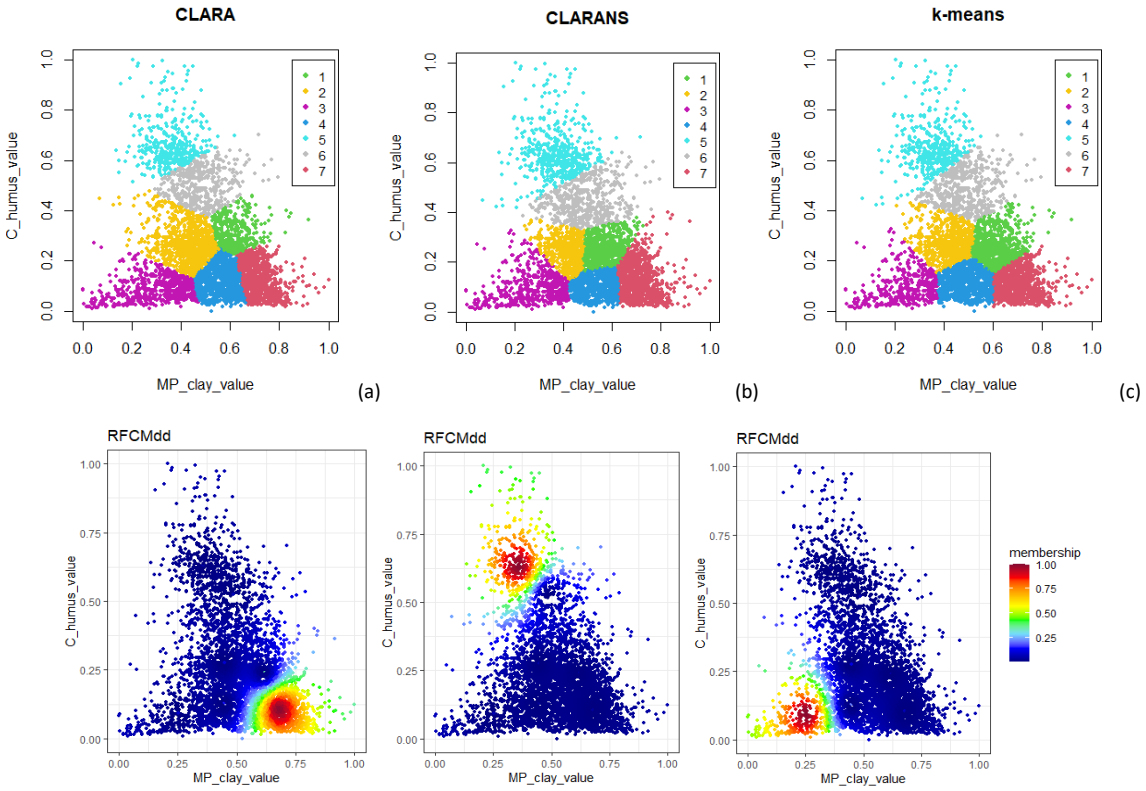
(e)

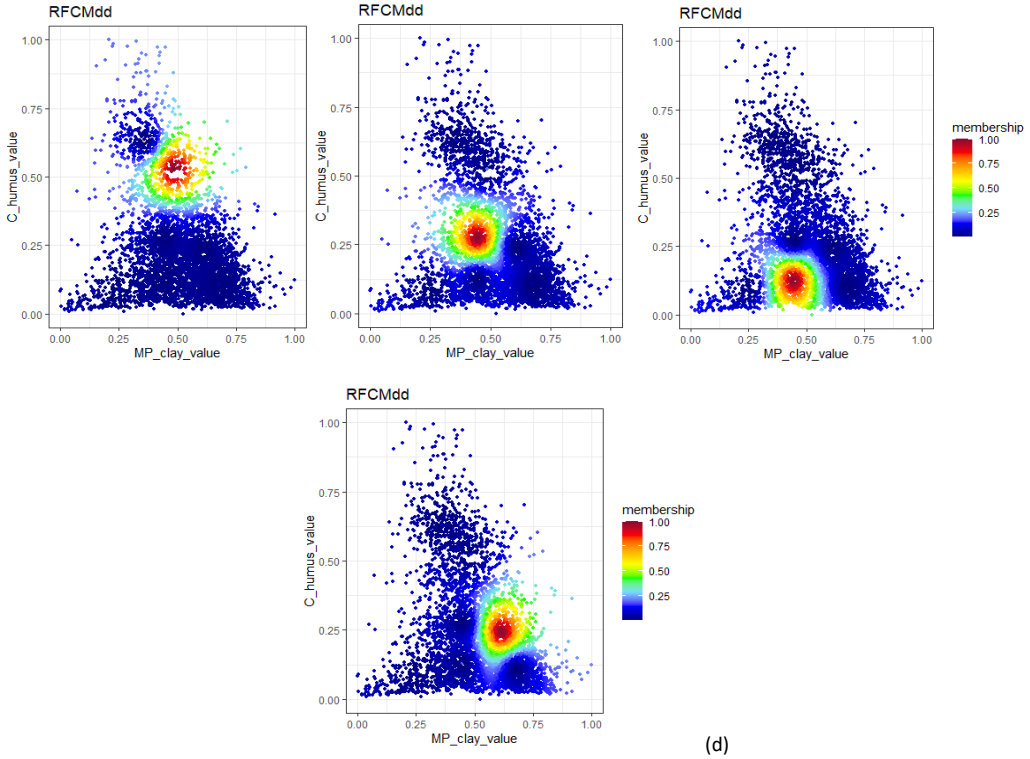

FCLARANS
FCLARANS
FCLARANS
FCLARANS
FCLARANS
C_humus_value
MP_clay_value
membership
0.75
0.50
0.25
0.00

(f)

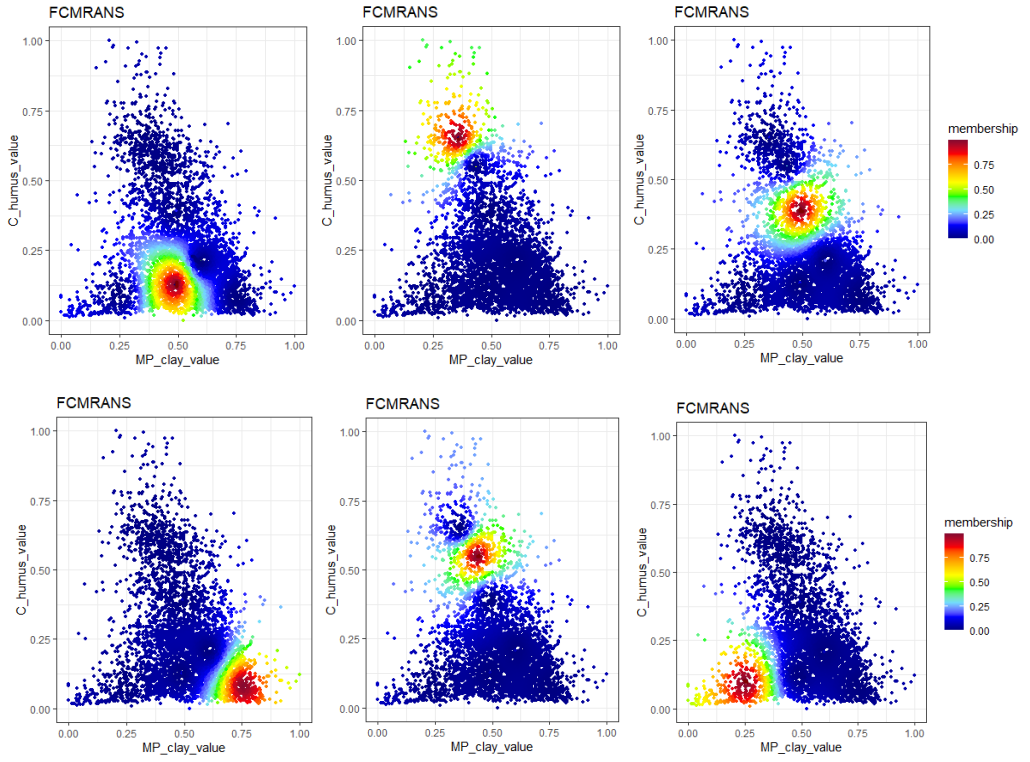


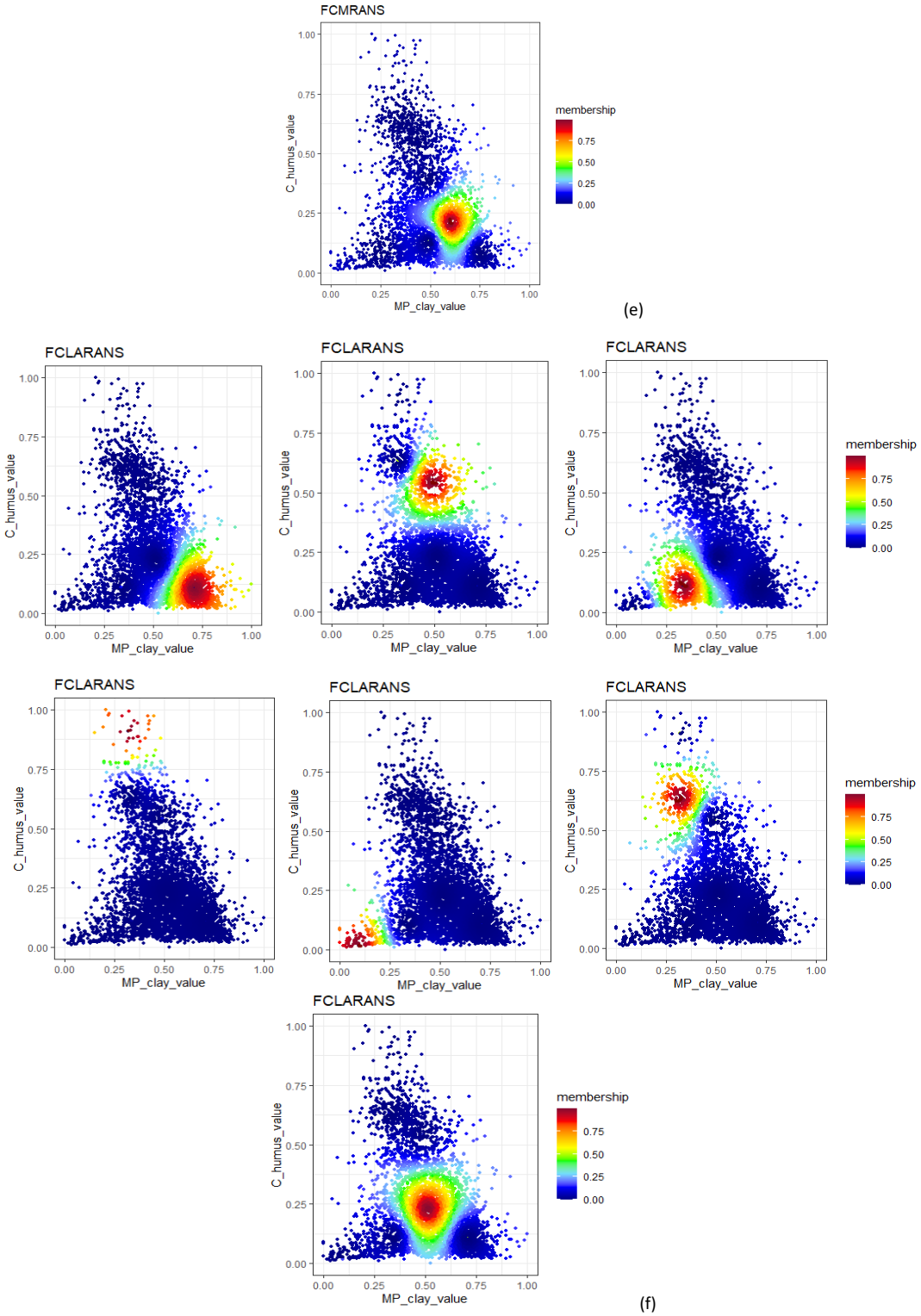
Slika 25. Rezultati klasterizacije dobijeni primjenom algoritama: (a) CLARA, (b) CLARANS, (c) *k*-means, (d) RFCMdd, (e) FCMRANS, (f) FCLARANS i (g) fuzzy *k*-means za *k* = 5

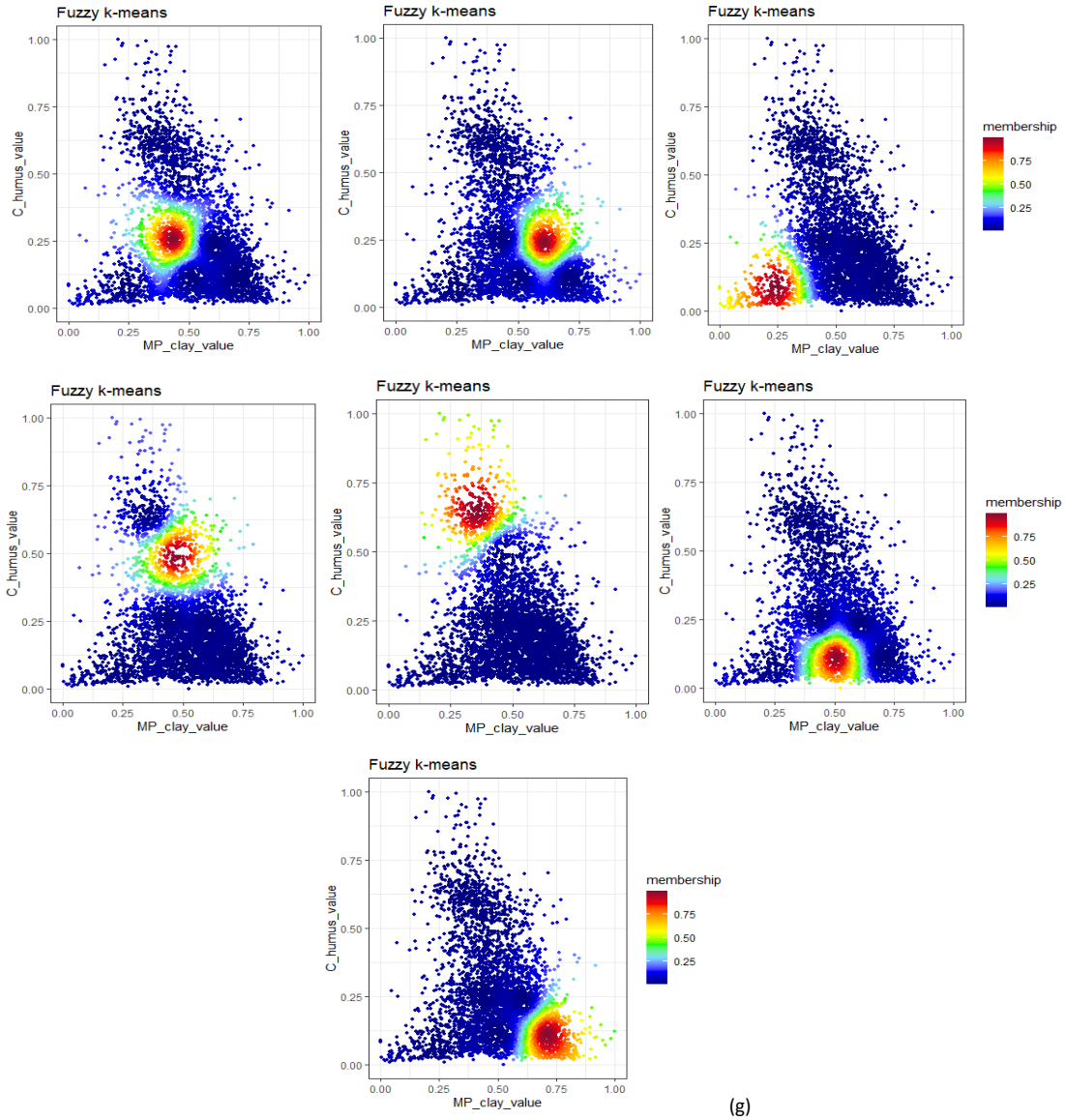



RFCMdd
RFCMdd
RFCMdd
RFCMdd
C_humus_value
MP_clay_value
membership
1.00
0.75
0.50
0.25
0.00

(d)


FCMRANS
FCMRANS
FCMRANS
FCMRANS
FCMRANS
FCMRANS
C_humus_value
MP_clay_value
membership
1.00
0.75
0.50
0.25
0.00


FCMRANS
C_humus_value
MP_clay_value
membership
0.75
0.50
0.25
0.00
(e)
FCLARANS
FCLARANS
FCLARANS
FCLARANS
FCLARANS
FCLARANS
FCLARANS
(f)



Slika 26. Rezultati klasterizacije dobijeni primjenom algoritama: (a) CLARA, (b) CLARANS, (c) *k*-means, (d) RFCMdd, (e) FCMRANS, (f) FCLARANS i (g) fuzzy *k*-means za *k* = 7

Sljedeća grupa algoritama primijenjena na parametre iz pedološke baze Crne Gore su fuzzy oblici *k*-medoids algoritama: RFCMdd, FCMRANS i FCLARANS. U sva tri slučaja će se dobiti klasteri koji nijesu međusobno isključivi i čije su granice blago zamućene u zavisnosti od odabira stepena zamućenosti *m*. Ulazni parametri algoritama su određeni na osnovu objašnjenja datih u opisu implementacije algoritama. Kod navedena 3 algoritma za inicijalizaciju medoida je realizovana funkcija koja nasumično odabere početni medoid, a svaki sljedeći (od ukupno *k* medoida) bira isto nasumično tako da nijesu ni najbliži ni najsličniji prethodno odabranim medoidima [13].

Rezultati dobijeni primjenom RFCMdd algortima na osnovu dva pedološka parametra za prethodno definisane ulazne parametre su data na slikama 22 (g), 24 (d), 25 (d) i 26 (d).

Broj klastera u koji su podaci grupisani je k = 2, k = 3, k = 5 i k = 7, redom. Svaki klaster predstavljen je na odvojenom grafiku. Uzorci koji definišu klaster imaju najveći stepen pripadnosti (uzorci označeni tamno crvenom bojom, u skladu sa *color bar*-om), odnosno članstvo u tom klasteru, dok je u ostalim klasterima ta vrijednost znatno manja. Što su uzorci udaljeniji od centra klastera (tamno crvenih uzoraka) tako se njihova boja preliva prema *color bar*-u, preko žute, zelene, plave do tamnoplave kojom su označeni uzorci sa najmanjim članstvom u posmatranom klasteru. Drugim riječima, kako se udaljavaju od centra klastera tako uzorci imaju manji stepen pripadnosti posmatranom klasteru. Posmatrajući i upoređujući sva tri grafika vidi se da su uzorci koji su označeni žutom, zelenom i svijetlo plavom bojom u stvari uzorci koji imaju približne pripadnosti u dva ili više klastera. Na tim mjestima klasteri blago zalaze jedan u drugi i dolazi do njihovog miješanja. Ne postoje jasne granice između klastera.

Na slikama 22 (h), 24 (e), 25 (e) i 26 (e) je dat primjer klasterizacije FCMRANS algoritma za dva pedološka parametra iz baze u k = 2, k = 3, k = 5 i k = 7 klastera, respektivno. Kao i u prethodnim primjerima, klasteri su predstavljeni na odvojenim graficima. Na mjestima na kojima su uzorci označeni tamno crvenom bojom, tu je njihova pripadnost klasteru najveća, a kako se boje mijenjaju u skladu sa *color bar*-om, tako i pripadnost uzoraka posmatranom klasteru opada, do tamno plave gdje vrijednost teži nuli. Može se uočiti da se podaci klasterizuju gotovo na isti način kao kod RFCMdd algoritma sa blago zamućenim granicama između klastera.

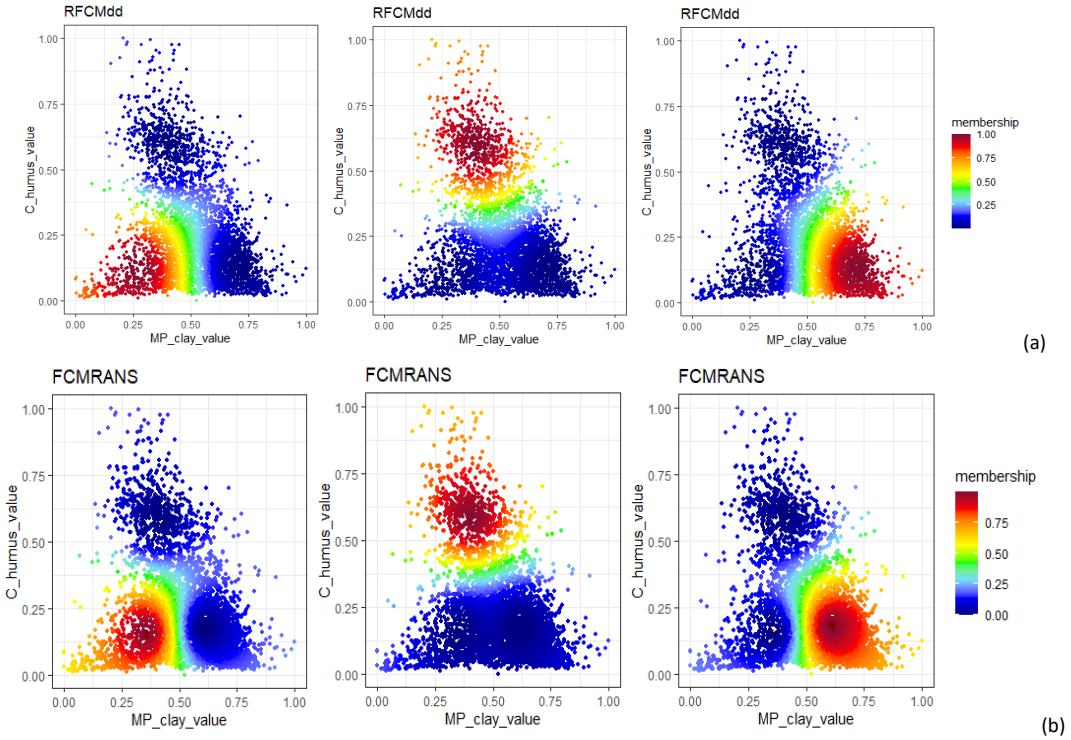
Slike 22 (i), 24 (f), 25 (f) i 26 (f) prikazuju primjere klasterizacije FCLARANS algoritma na dva parametra i to za k = 2, k = 3, k = 5 i k = 7, respektivno. Upoređujući grafike dobijene primjenom RFCMdd i FCMRANS algoritama za dva odabrana pedološka parametra, vizuelno se može potvrditi da, u odnosu primjere u drugom poglavlju, FCLARANS formira klastere na mjestima gdje je manja koncentracija podataka, tj. na mjestima gdje su podaci izolovaniji u odnosu na ostale podatke, što potvrđuje klasterizacija u 3 klastera gdje imamo jedan dominantni klaster i 2 manja na mjestima gdje su podaci više izolovani u odnosu na ostale. Povećanjem broja klastera u tom dijelu su formirani klasteri sa manjim brojem uzoraka u odnosu na klastere dobijene primjenom RFCMdd i FCMRANS algoritama. U ovom slučaju potvrđuje se manja robusnost FCLARANS-a na šum među podacima. Kako su tri analizirana fuzzy k-medoids algoritma zasnovana na istom principu, očekivano je da daju približno jednake rezultate. Razlika koja postoji među njima se ogleda u uticaju šuma na klasterizaciju kod FCLARANS-a, koja je očigledno uzrokovana nasumičnim odabirom susjeda, kao potencijalnog medoida, iz cijelog skupa, bez ograničavanja potencijalnih medoida na one koji imaju najveći stepen pripadnosti klasteru (kod RFCMdd i FCMRANS algoritama medoidi se biraju među p uzoraka sa najvećim stepenom pripadnosti klasteru).

Dobijeni rezultati potvrđuju da će se primjenom RFCMdd-a i FCMRANS-a postići bolja klasterizacija u odnosu na onu dobijenu primjenom FCLARANS-a. Što je podatak o količini šuma preciznije određen kao ulazni argument, to će i prva dva algoritma dati bolje rezultate.

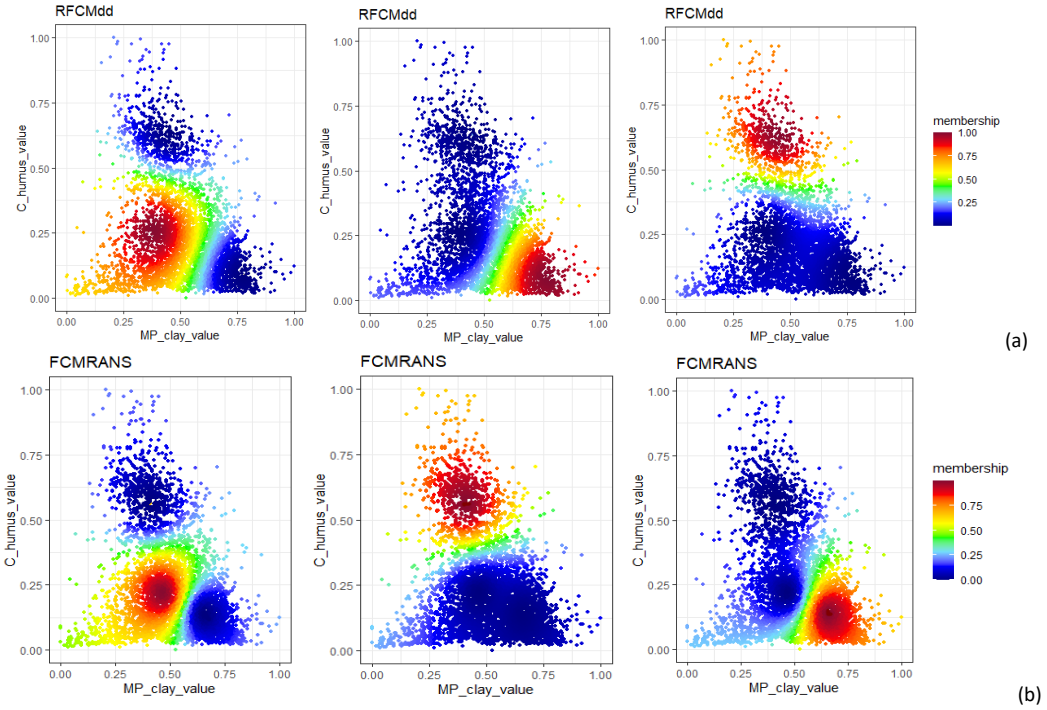
Osim broja klastera i stepena zamućenosti, kod RFCMdd i FCMRANS algoritama je potrebno definisati prag koji označava količinu šuma među podacima. U ovom radu je

korišćen DBSCAN algoritam u određivanju količine šuma. Za dva pedološka parametra iz baze 4.23% čine podaci šuma, a za tri pedološka parametra kao šum je označeno 6.15% podataka. Obje vrijednosti su dobijene na osnovu tabele 6.

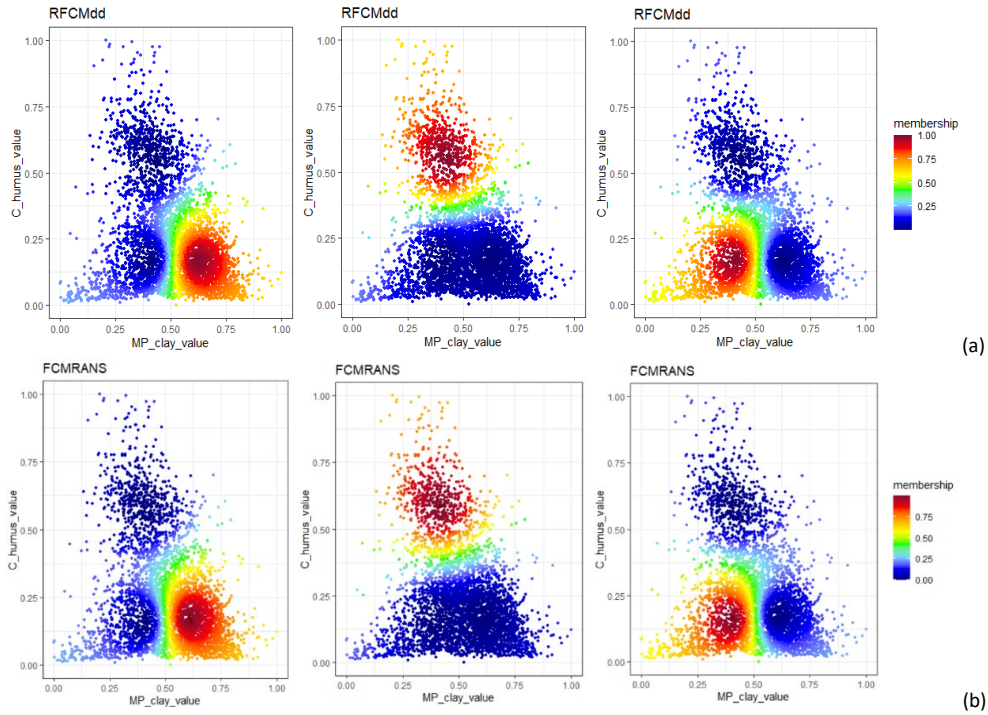
Radi smanjenja kompleksnosti i povećanja robusnosti na šum kod algoritama RFCMdd i FCMRANS potrebno je odrediti *p* ulazni parametar. Ranije je pojašnjeno da broj *p* predstavlja broj uzoraka koji imaju najveće članstvo u svakom od klastera posebno. Za svaki od klastera, *p* predstavlja uzorke koji mogu biti potencijalni medoidi. Za *p* se uvijek uzima manja vrijednost nego što je to prosječna veličina klastera, koja je s druge strane dovoljno velika da postoji mobilnost medoida pri njihovom ažuriranju. U ovom istraživanju pokazano je da ukoliko je *p* = 80 (slika 27) onda se medoid posmatranog klastera zamijeniti medoidom koji je među prvih 80 najbližih uzoraka. Vrijednosti koje su manje smanjuju i broj potencijalnih medoida i mogu uticati na klasterizaciju. Na slici 28 (a) za *p* = 30 vidi se da je drugi izabrani klaster najmanji (drugi grafik), jer je mobilnost medoida ograničena na 30 najbližih uzoraka. Za vrijednosti *p* > 80 (slika 29) nema velikog uticaja na rezultate klasterizacije u odnosu na *p* = 80, ali se povećava vrijeme izvršavanja algoritama, jer se ispituje veći broj uzoraka pri ažuriranju medoida. Iz tog razloga je kao optimalna vrijednost uzeto *p* = 80.



Slika 27. Klasterizacija (a) RFCMdd i (b) FCMRANS algoritma u 3 klastera za *p*=80



Slika 28. Rezultati klasterizacije primjenom (a) RFCMdd i (b) FCMRANS algortima u 3 kalstera za *p*=30

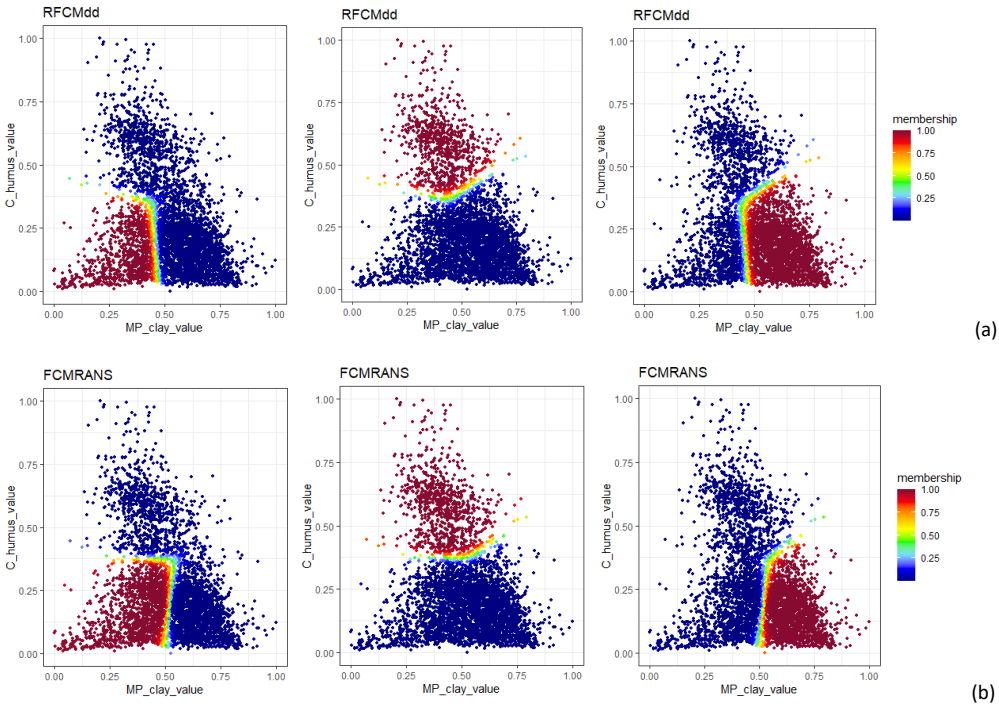


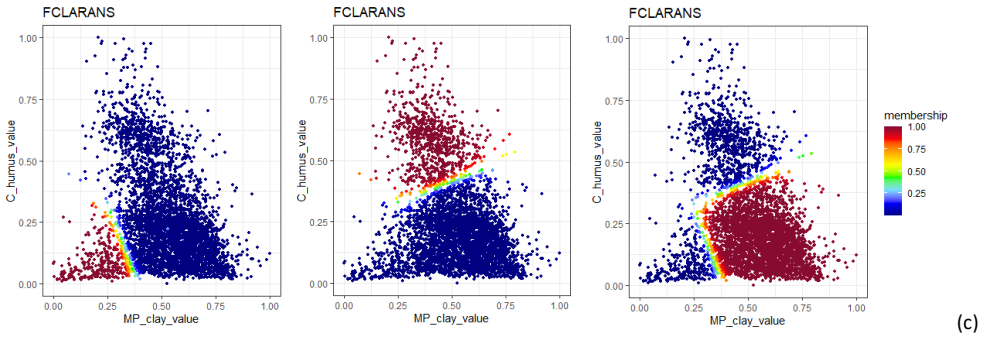
Slika 29. Rezultati klasterizacije primjenom (a) RFCMdd i (b) algortima u 3 klastera za *p*=300

3.1 Određivanje optimalne vrijednosti fuzzifier-a

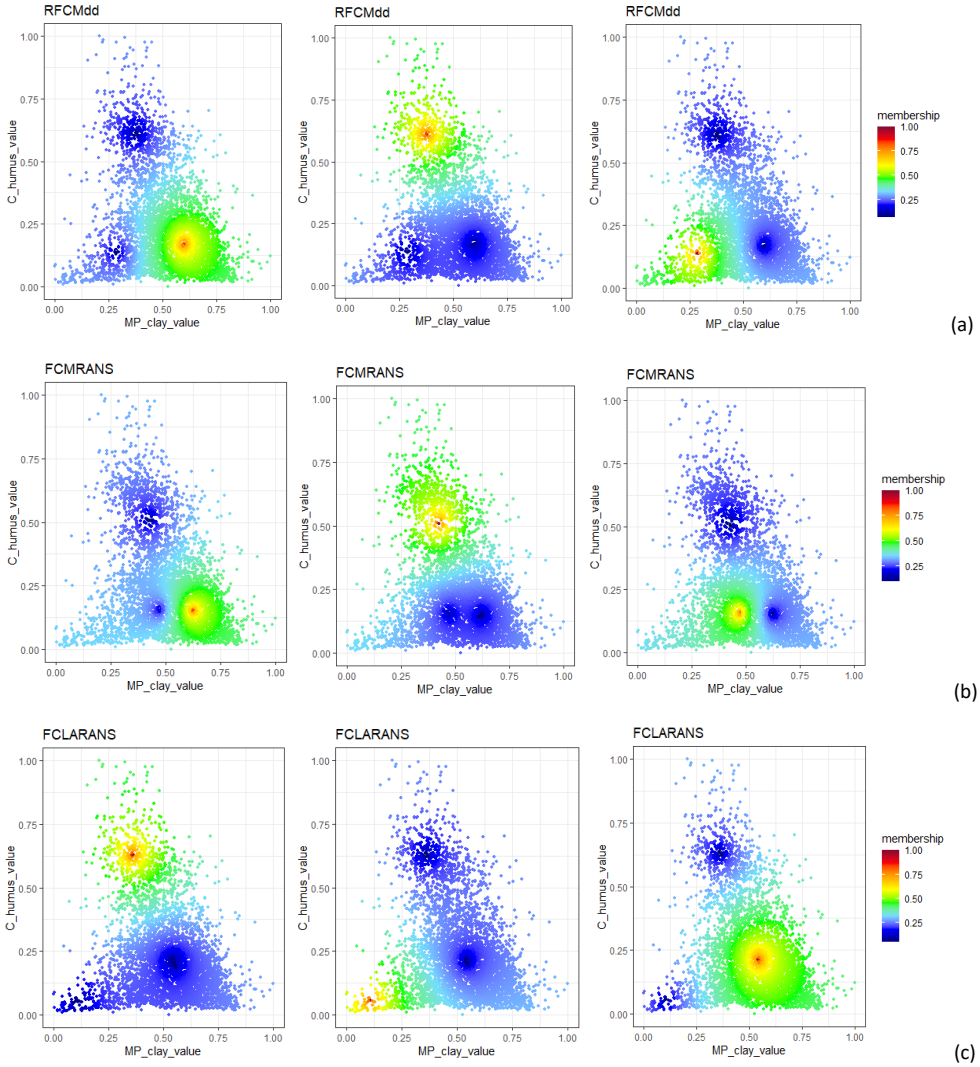
Vrijednost parametra *m* koji određuje stepen zamućenosti klastera je za sve primjere i algoritme postavljen na *m* = 1.5. Što je *m* bliže jedinici granice između klastera postaju jasnije. Na slici 30 je predstavljena klasterizacija za *m* = 1.1. Vidi se da na graficima preovladavaju tamno crveni i tamnoplavi uzorci. Tamno crvenom su predstavljeni uzorci koji pripadaju trenutno posmatranom klasteru (gdje je stepen članstva uzoraka u klasteru jednak ili približno jednak 1), dok su tamnoplavi uzorci koji pripadaju ostalim klasterima, oni imaju najmanje članstvo u posmatranom klasteru (članstvo koje ima vrijednost jednako ili približno jednako 0, u skladu sa *color bar*-om). Mali je broj uzoraka koji se nalaze na prelazu između klastera (žute i zelene nijanse u skladu sa *color bar*-om), a predstavljeni gotovo tankom linijom kao granica između klastera.

Nasuprot tome, kada je *m* >> 1 članstva uzoraka u klasterima opadaju i uzorci imaju približno jednake vrijednosti članstva u svim klasterima. Na slici 31 predstavljeni su grafici za *m* = 3. Na njima se vidi da je za sve algoritme mali broj uzoraka koji imaju najveću pripadnost klasterima, čak bi se mogli reći da je jedino medoid taj koji je označen tamno crvenom bojom u skladu sa *color bar*-om i vrijednostima koje definišu stepen pripadnosti. Tamno plava boja opisuje ostale klastere i uzorke koji imaju najveću pripadnost u njima. Većina uzoraka je označena žutim i zelenim nijansama, što vodi zaključku da ti uzorci imaju približno jednake pripadnosti klasterima. Ovo rezultira lošu klasterizaciju. Dakle, kada je *m* >> 1 mobilnost klastera opada kroz iteracije, jer stepeni pripadnosti u klasterima imaju sve manje i manje vrijednosti, osim samog medoida posmatranog klastera. Grafici potvrđuju da promjena vrijednosti *m* parametra utiče jednako na sve algoritme.





Slika 30. Rezultati klasterizacije za *k*=3 i *m*=1.1 za svaki od analiziranih *k*-medoids algoritama: (a) RFCMdd, (b) FCMRANS i (c) FCLARANS



Slika 31. Rezultati klasterizacije za *k*=3 i *m*=3 za svaki od analiziranih *k*-medoids algoritama: (a) RFCMdd, (b) FCMRANS i (c) FCLARANS

3.2 Određivanje broja iteracija

Kod RFCMdd i FCMRANS algoritama determinisanje broja iteracija je od velikog značaja za kvalitet klasterizacije. U tabeli 8 i tabeli 9 date su odvojeno vrijednosti srednjeg kvadratnog rastojanja između uzoraka i medoida za definisani broj klastera i broj iteracija kroz koje algoritam pronalazi optimalni skup medoida. Isti rezultati su predstavljeni grafički na slici 32. Vrijednosti su dobijene za 2 pedološka parametra iz baze i predstavljaju prosjek za 5 poziva algoritma za odgovarajuće ulazne agrumente. Na osnovu dobijenih brojčanih vrijednosti podataka zaključuje se da je kvalitet klasterizacije bolji (srednje kvadratno rastojanje između uzoraka i medoida opada) sa povećanjem broja iteracija, što je očekivano. Potvrđeno je da su dovoljne dvije iteracije, odnosno da je dovoljno naći drugi lokalni minimum da bi se postigla optimalna klasterizacija (slika 32). Za broj iteracija veći od 2 nema velikih varijacija u kvalitetu klasterizacije. Odabirom manjeg broja iteracija će se zasigurno skratiti vrijeme izvršavanja algoritma. Za razliku od RFCMdd-a i FCMRANS-a, FCLARANS je implementiran da se izvršava kroz jednu iteraciju. Razlog je što kod njega promjene broja iteracije ne utiču mnogo na rezultat klasterizacije, nema većih varijacija u vrijednostima srednjeg kvadratnog rastojanja između uzoraka i medoida, što potvrđuju podaci u tabeli 10 i na slici 32. U ovom pogledu se prednost može dati FCLARANS algoritmu u odnosu na RFCMdd i FCMRANS. Primjećuje se da je srednje kvadratno rastojanje kod FCLARANS-a znatno manje nego kod RFCMdd-a i FCMRANS-a. Razlog je što se kod FCLARANS-a novi medoidi biraju u odnosu na sve uzorke. Kod RFCMdd i FCMRANS algoritama pri ažuriranju medoida se isključuju tačke šuma i sa tim podacima se i računa funkcija cijene koja se kroz iteracije poredi u cilju poboljšanja klasterizacije. Dakle, ova dva algoritma u potpunosti izuzimaju šum prilikom svog izvršavanja i pronalaze medoide unutar „očišćenog“ skupa podataka. Njihove pozicije će se u tom dijelu malo razlikovati od pozicija medoida rezultiranih primjenom FCLARANS-a. Kako u srednje kvadratno rastojanje ulaze udaljenosti između svih medoida i uzoraka, oni uzorci koji su kod RFCMdd-a i FCMRAN-a bili izdvojeni kao šum biće na većoj udaljenosti od medoida nego što je to slučaj kod FCLARANS-a koji ih je cijelo vrijeme izvršavanja uzimao u obzir.

U tabeli 11 i tabeli 12 su vremena izvršavanja algoritama u zavisnosti od broja klastera i broja iteracija. Na grafičkom prikazu tabelarnih vrijednosti (slika 33 (a)) vidi se da je vrijeme izvršavanja RFCMdd algoritma mnogo manje i sa blagim promjenama u zavisnosti od broja iteracija, nego kod FCMRANS-a, gdje su skokovi veći. Ovo je za očekivati s obzirom da FCMRANS obuhvata dvije *repeat* petlje, jednu koja provjerava maksimalan broj susjeda, i drugu za definisani broj iteracije, za razliku od RFCMdd algoritma gdje imamo samo jednu za izvršavanje definisanog broja iteracije. Na slici 33 (b) je predstavljena zavisnost promjene vremena izvršavanja algoritama u odnosu na broj klastera.

RFCMdd **Broj iteracija**	**Srednje kvadratno rastojanje između uzoraka i medoida**		
	3 klastera	**5 klastera**	**7 klastera**
1	1.130	1.740	2.401
2	0.906	1.510	2.250
3	0.868	1.480	2.204
4	0.814	1.462	2.184
5	0.798	1.388	2.164

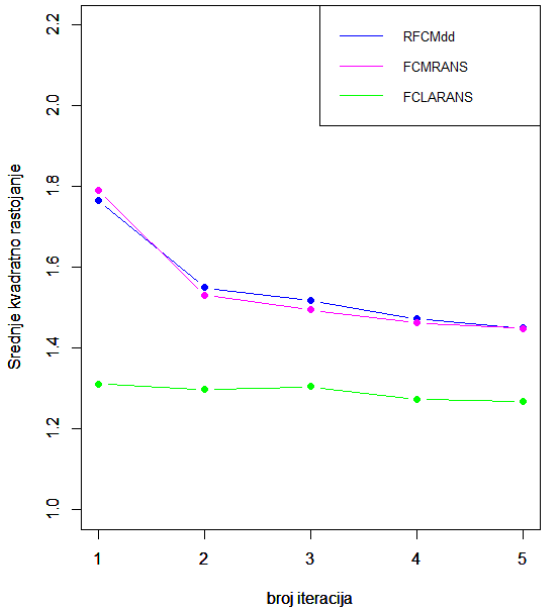
Tabela 8. Zavisnost srednjeg kvadratnog rastojanja u odnosu na broj iteracija i broj klastera kod RFCMdd algoritma

FCMRANS **Broj iteracija**	**Srednje kvadratno rastojanje između uzoraka i medoida**		
	3 klastera	**5 klastera**	**7 klastera**
1	1.19	1.677	2.510
2	0.854	1.508	2.231
3	0.832	1.460	2.193
4	0.801	1.416	2.173
5	0.779	1.401	2.166

Tabela 9. Zavisnost srednjeg kvadratnog rastojanja u odnosu na broj iteracija i broj klastera kod FCMRANS algoritma

FCLARANS **Broj iteracija**	**Srednje kvadratno rastojanje između uzoraka i medoida**		
	3 klastera	**5 klastera**	**7 klastera**
1	0.771	1.342	1.823
2	0.724	1.340	1.830
3	0.782	1.287	1.845
4	0.791	1.231	1.785
5	0.743	1.265	1.795

Tabela 10. Zavisnost vrijednosti funkcije cijene u donosu na broj iteracija i broj klastera kod FCLARANS algoritma



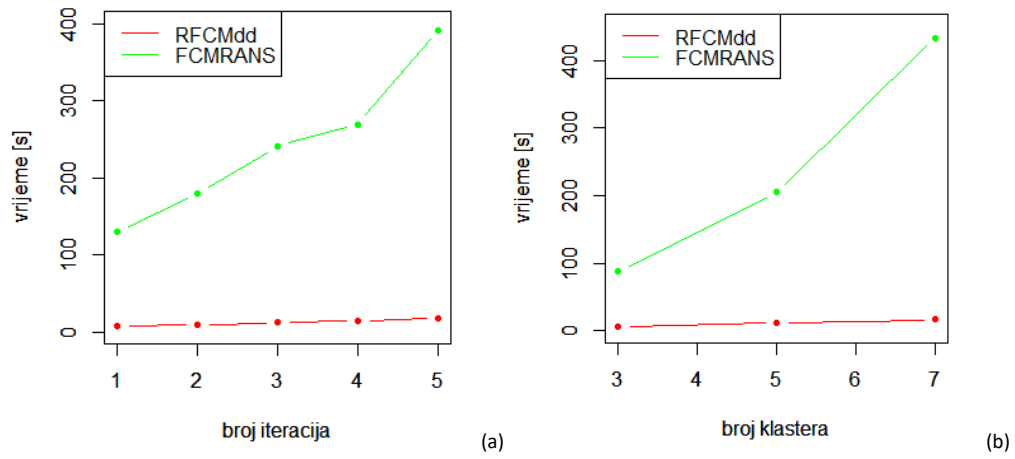
Slika 32. Grafik zavisnosti srednjeg kvadratnog rastojanja u odnosu na broj iteracija

RFCMdd	**Vrijeme izvršavanja algoritma**		
Broj iteracija	**3 klastera**	**5 klastera**	**7 klastera**
1	3.43	6.37	10.61
2	5.47	8.88	12.27
3	6.93	12.06	16.73
4	8.71	14.64	20.10
5	10.38	17.06	24.83

Tabela 11. Vrijeme [s] izvršavanja algoritma u odnosu na broj iteracija i broj klastera kod RFCMdd algoritma

FCMRANS	**Vrijeme izvršavanja algoritma**		
Broj iteracija	**3 klastera**	**5 klastera**	**7 klastera**
1	60.00	102.98	227.65
2	69.33	172.14	297.59
3	98.17	188.79	436.85
4	89.80	242.62	476.65
5	125.52	320.76	728.73

Tabela 12. Vrijeme [s] izvršavanja algoritma u odnosu na broj iteracija i broj klastera kod FCMRANS algoritma



Slika 33. Grafik zavisnosti vremena izvršavanja u odnosu na (a) broj iteracija i (b) broj klastera

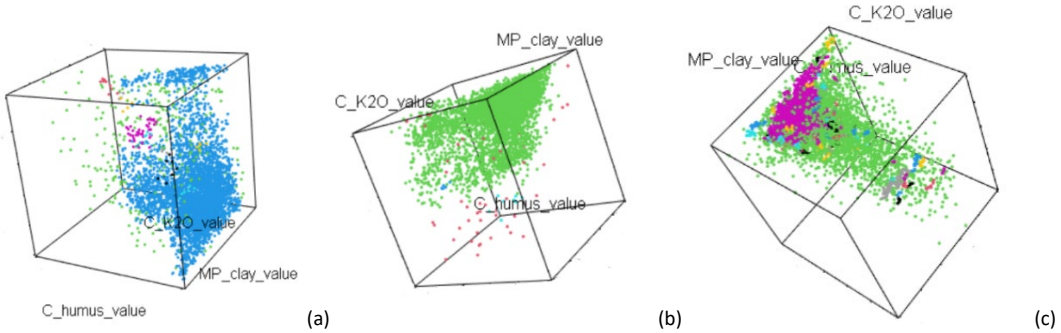
U radu [1] je predstavljen fuzzy oblik *k*-means algoritam i njegova primjena na istim pedološkim podacima. Za razliku od običnog *k*-means algoritma, njegov fuzzy oblik nema isključive klastere, već uzorci pripadaju svakom klasteru sa različitim stepenom pripadnosti. Fuzzy *k*-medoids algoritmi se smatraju modifikacijama fuzzy *k*-means algoritma, sa većom robusnosti na šum zbog veće robusnosti medoida kao centara klastera u odnosu na centroide, što je ranije objašnjeno. Na slikama 22 (j), 24 (g), 25 (g) i 26 (g) su primjeri klasterizacije dva pedološka parametra koristeći fuzzy *k*-means algoritam. Uočava se sličnost u klasterizaciji fuzzy *k*-means i fuzzy *k*-medoids algoritmima klasterizacije.

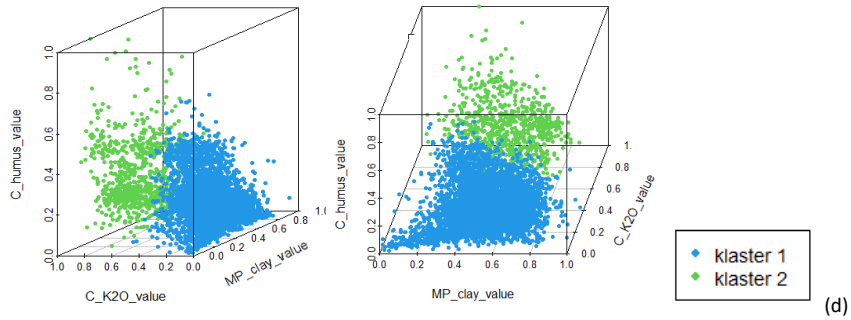
U nastavku su dati rezultati primjene prethodnih algoritama na tri pedološka parametra, kako bi potvrdili zaključke dobijene za dva parametra.

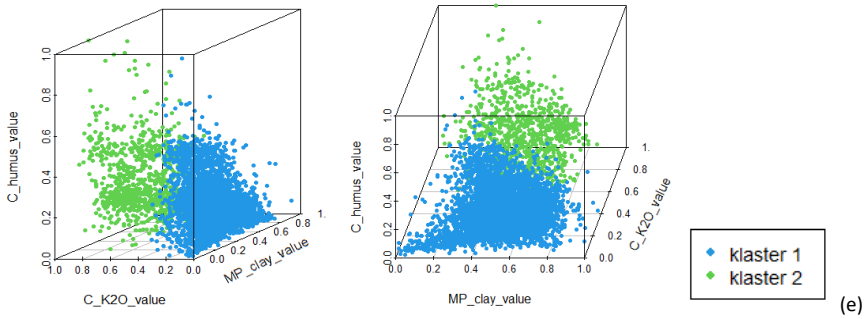
Gotovo ista klasterizacija je rezultirana primjenom DBSCAN na tri mehaničko-fizička pedološka parametra. Na slici 21 (b) je grafik pomoću koga je određena optimalna *eps* vrijednost, *eps* = 0.05. Minimalan broj uzoraka koji jedna tačka treba imati u svom susjedstvu da bi se dodijelila klasteru je *MinPts* = 4. Slika 34 (a) prikazuje rezultate klasterizacije primjenom DBSCAN algoritma na 3 pedološka parametra za prethodno definisane optimalne vrijednosti ulaznih parametara. Proizilaze isti zaključci kao i kod primjera klasterizacije DBSCAN-om za dva pedološka parametra. Za optimalne vrijednosti ulaznih parametara algoritma, za tri pedološka parametra, je prisutan jedan dominanti klaster (plavi klaster), veći broj manjih klastera i tačke koje predstavljaju šum (zeleni uzorci). Kako se povećava *eps* vrijednost (slika 34 (b)), tako podaci konvergiraju ka jednom klasteru (zeleni klaster) i tačkama šuma (crveni uzorci), a kako *eps* vrijednost opada (slika 34 (c)), tako se povećava broj manjih klastera i tačaka šuma (zeleni uzorci). Zaključuje se da DBSCAN algoritam nije pogodan za klasterizaciju pedoloških podataka pošto formira klastere na mjestima gdje je koncentracija uzoraka veća od predefinisanog praga, pa je za razdvajanje klastera neohodno da budu jasno razdvojeni jedni od drugih. U analiziranoj bazi klasteri su proizvoljnog oblika sa različitim brojem uzoraka, neuporedivih veličina (dominantni i veći broj manjih klastera). DBSCAN nije pogodan za razdvajanje klastera sa graničnim tačkama sličnih karakteristika, što je slučaj sa pedološkim podacima o tipovima zemljišta. Pozitivna strana primjene ove tehnike

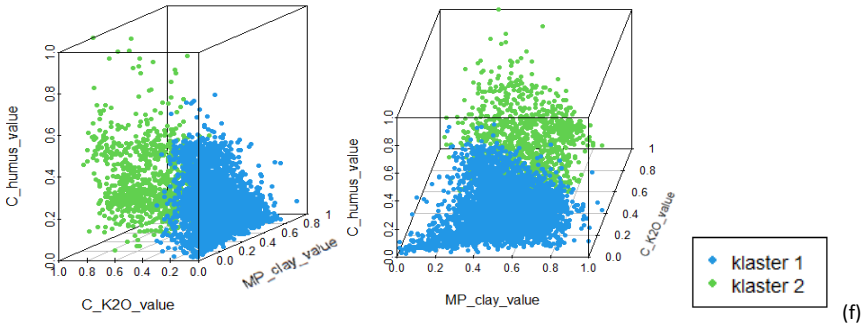
klasterizacije jeste identifikacija šuma među podacima. Ovo se može iskoristiti za određivanje procenta šuma, koji se kod pojedinih analiziranih algoritama zahtijeva kao ulazni parametar algoritma, poboljšavajući tako performanse algoritma.

Optimalan broj klastera CLARA algoritma u primjeni nad tri pedološka parametra je *k* = 2 (slika 23 (b)). Rezultat klasterizacije dat je na slici 34 (d). Primjeri klasterizacije tri pedološka parametra u 2 klastera dati su i za CLARANS (slika 34 (e)), RFCMdd (slika 34 (g)), FCMRANS (slika 34 (h)), FCLARANS (slika 34 (i)). Na slici 34 - (f) i (j) su grafici klasterizacije *k*-means i fuzzy *k*-means algoritama, radi poređenja rezultata. Na slikama 35, 36 i 37 je klasterizacija istih algoritama za *k* = 3, *k* = 5 i *k* = 7. Zbog boljeg uočavanja klastera kod *k*-medoids i *k*-means algoritama rezultati su prikazani iz dva ugla. Kod fuzzy oblika svaki klaster predstavljen je na posebnom grafiku. Nema razlike u načinu klasterizacije u odnosu na primjere za dva pedološka parametra, što je i očekivano za iste algoritme, tako da sva pravila i zaključci donešeni za dva parametra važiće i u ovom slučaju.

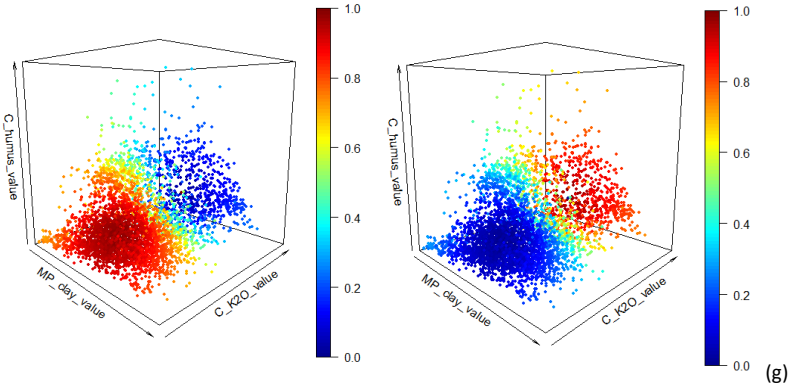




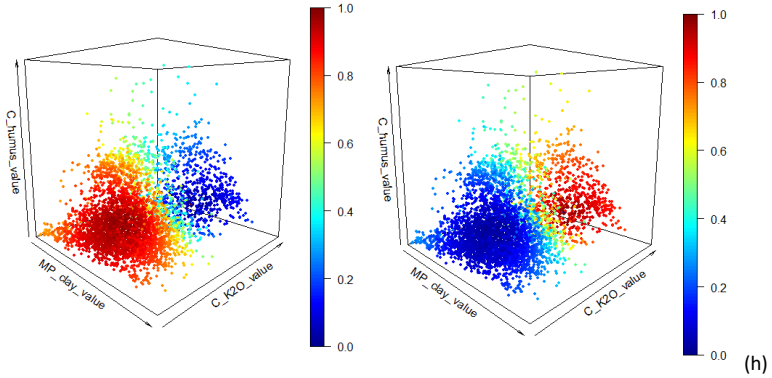



C_humus_value
C_K2O_value
MP_clay_value
C_humus_value
MP_clay_value
C_K2O_value
klaster 1
klaster 2

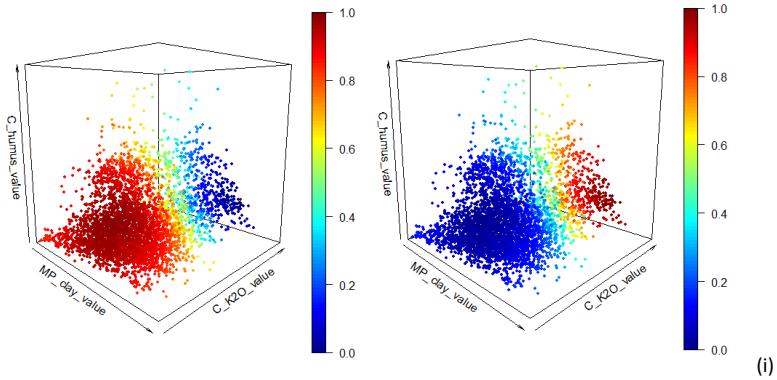
(f)


C_humus_value
MP_clay_value
C_K2O_value
C_humus_value
MP_clay_value
C_K2O_value

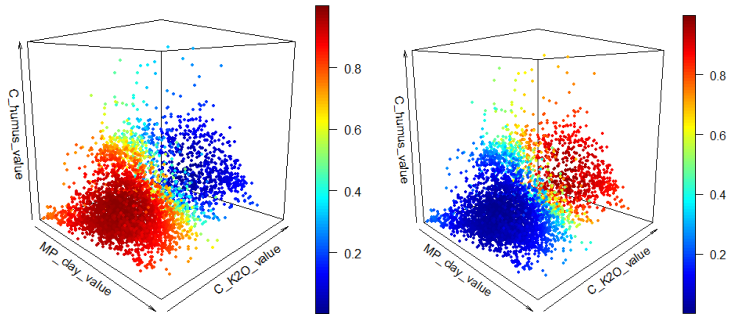
(g)


C_humus_value
MP_clay_value
C_K2O_value
C_humus_value
MP_clay_value
C_K2O_value

(h)

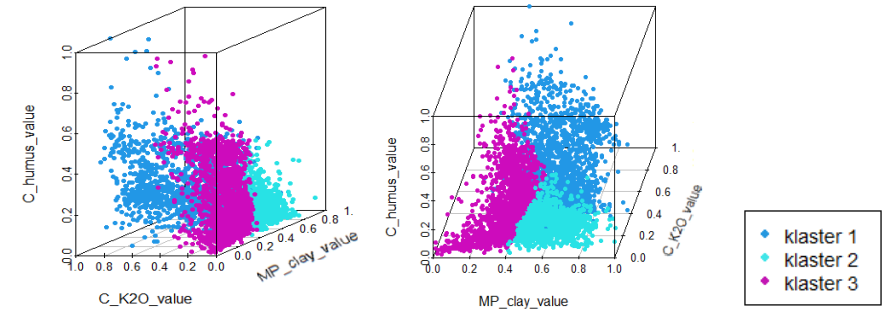

C_humus_value
MP_clay_value
C_K2O_value
C_humus_value
MP_clay_value
C_K2O_value

(i)

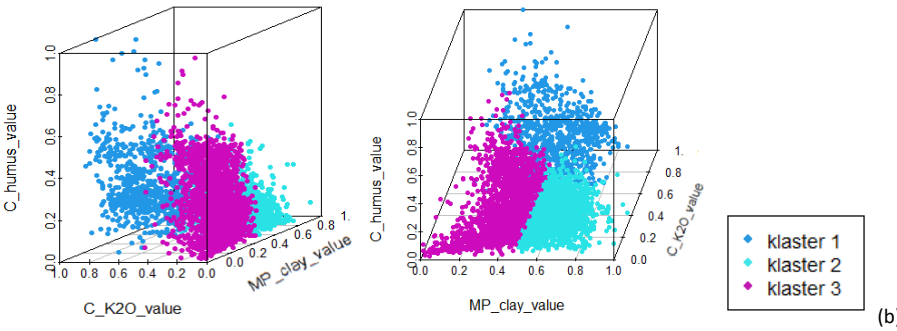


(j)

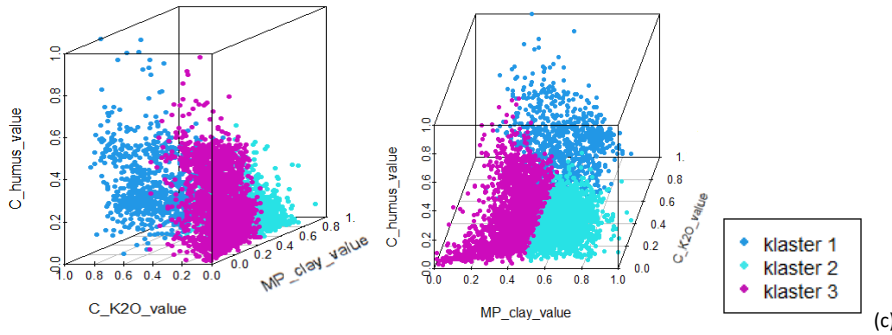
Slika 34. Rezultati klasterizacije dobijeni primjenom algoritama: (a) DBSCAN-a za optimalno *eps*, (b) DBSCAN-a za *eps* veće od optimalnog, (c) DBSCAN-a za *eps* manje od optimalnog i (d) CLARA, (e) CLARANS, (f) *k*-means, (g) RFCMdd, (h) FCMRANS, (i) FCLARANS i (j) fuzzy *k*-means za *k* = 2, na tri pedološka parametra



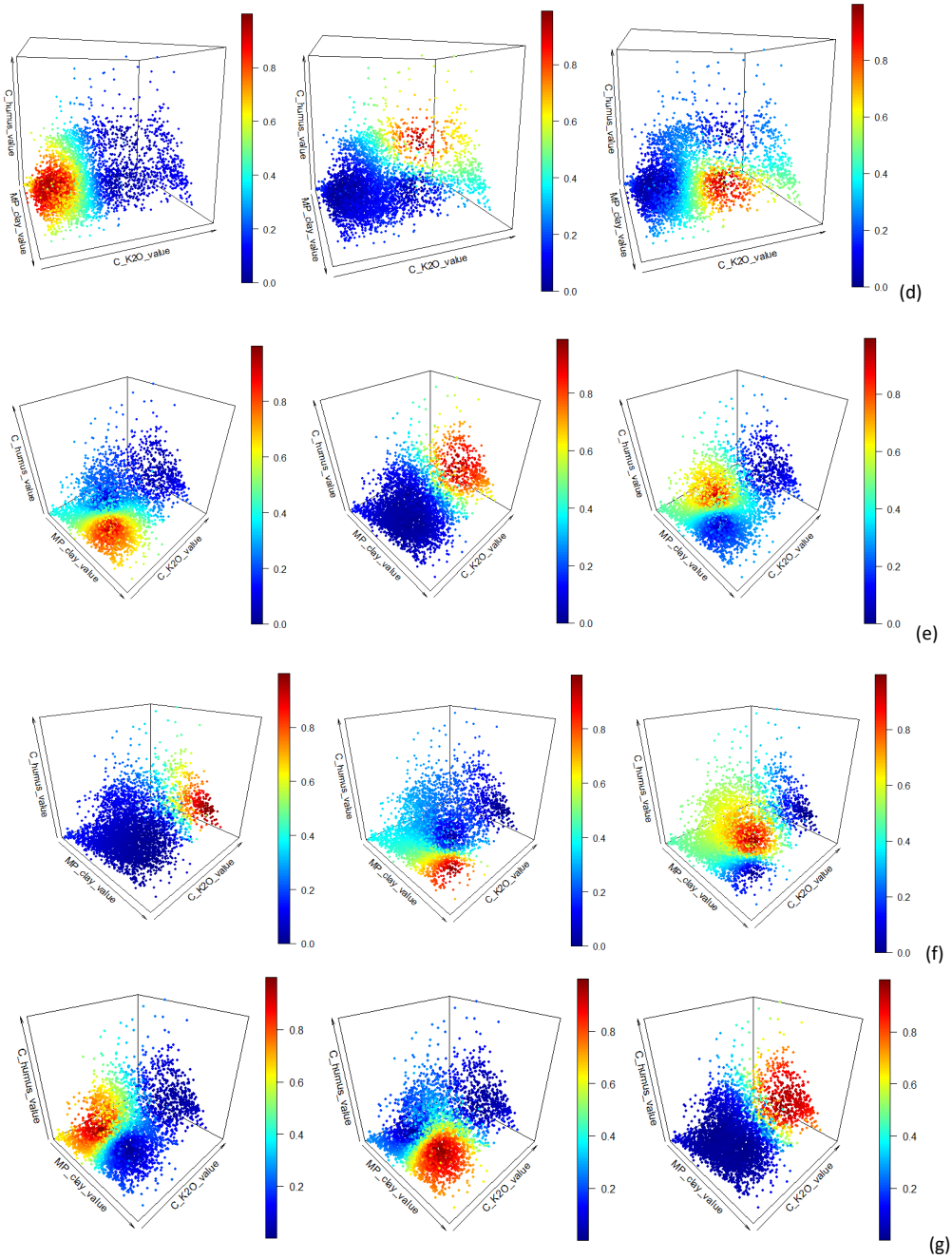
(a)



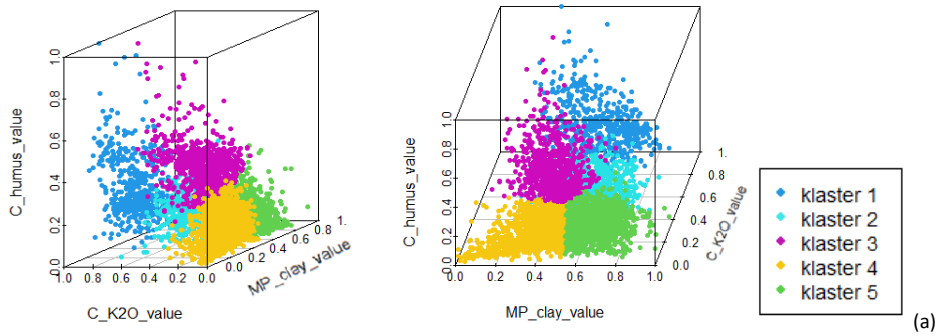
(b)



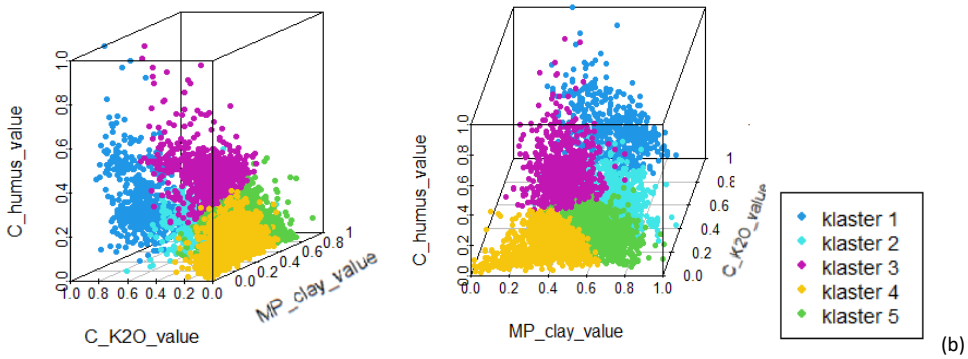
(c)



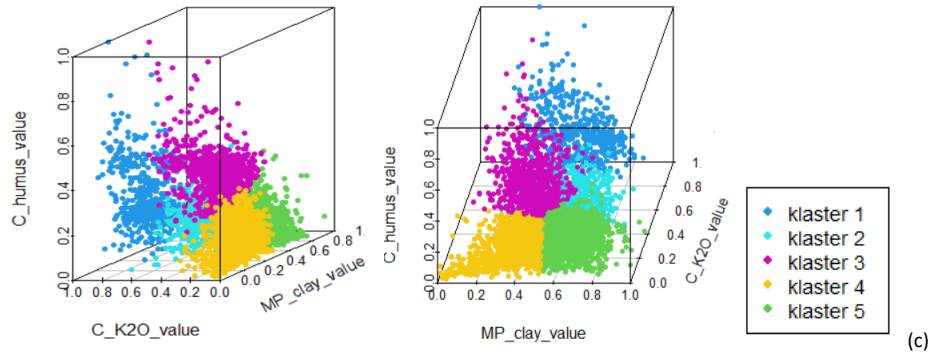
Slika 35. Rezultati klasterizacije dobijeni primjenom algoritama: (a) CLARA, (b) CLARANS, (c) *k*-means, (d) RFCMdd, (e) FCMRANS, (f) FCLARANS i (g) fuzzy *k*-means za *k* = 3, na tri pedološka parametra


C_humus_value
C_K2O_value
MP_clay_value
C_K2O_value
MP_clay_value
klaster 1
klaster 2
klaster 3
klaster 4
klaster 5

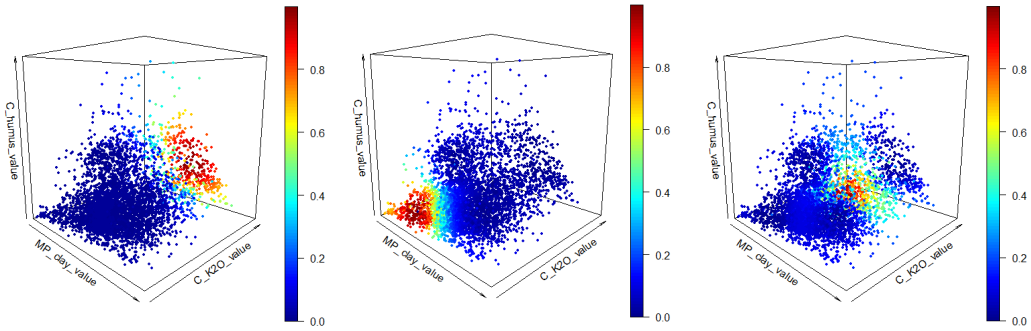
(a)

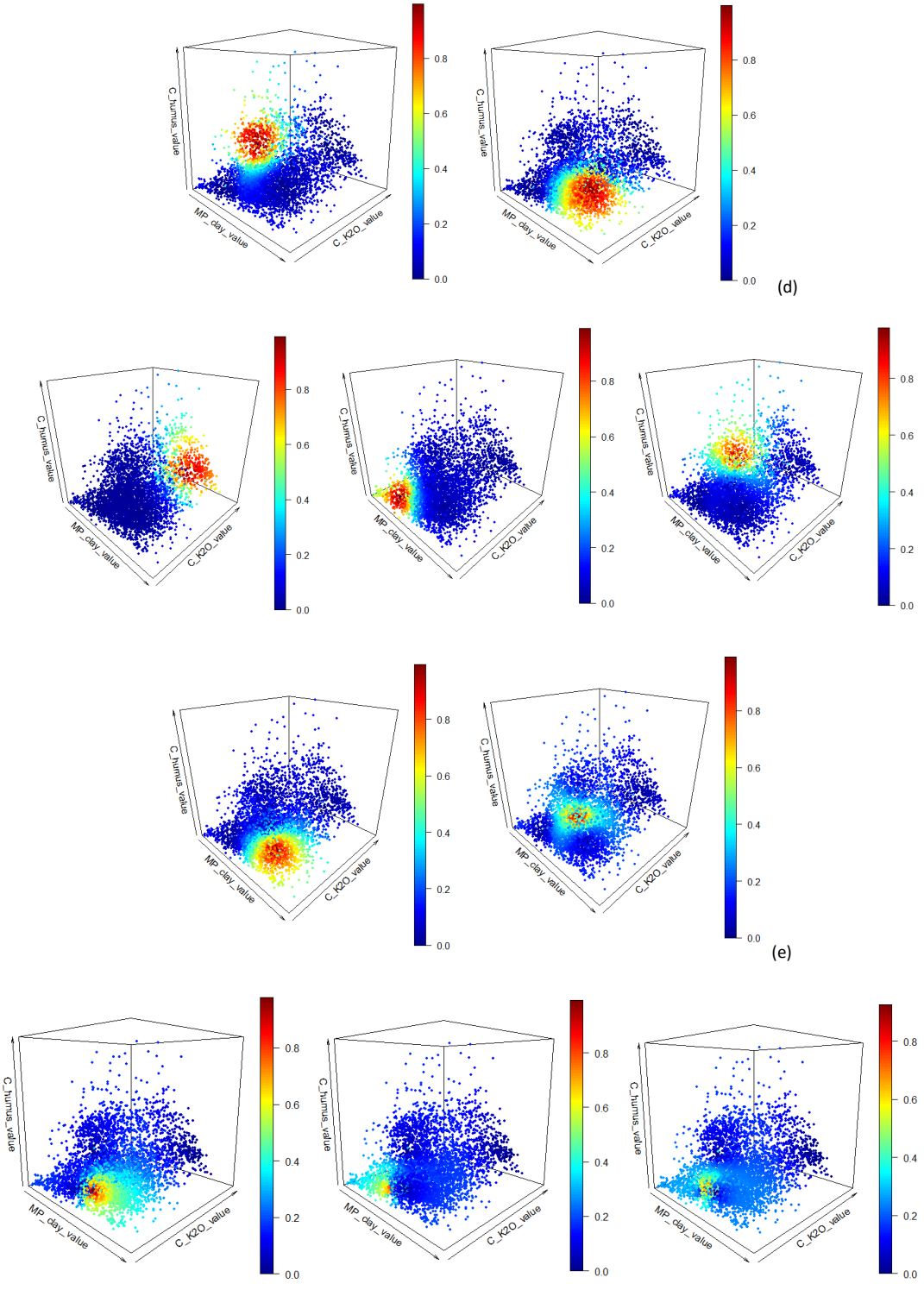

C_humus_value
C_K2O_value
MP_clay_value
C_K2O_value
MP_clay_value
klaster 1
klaster 2
klaster 3
klaster 4
klaster 5

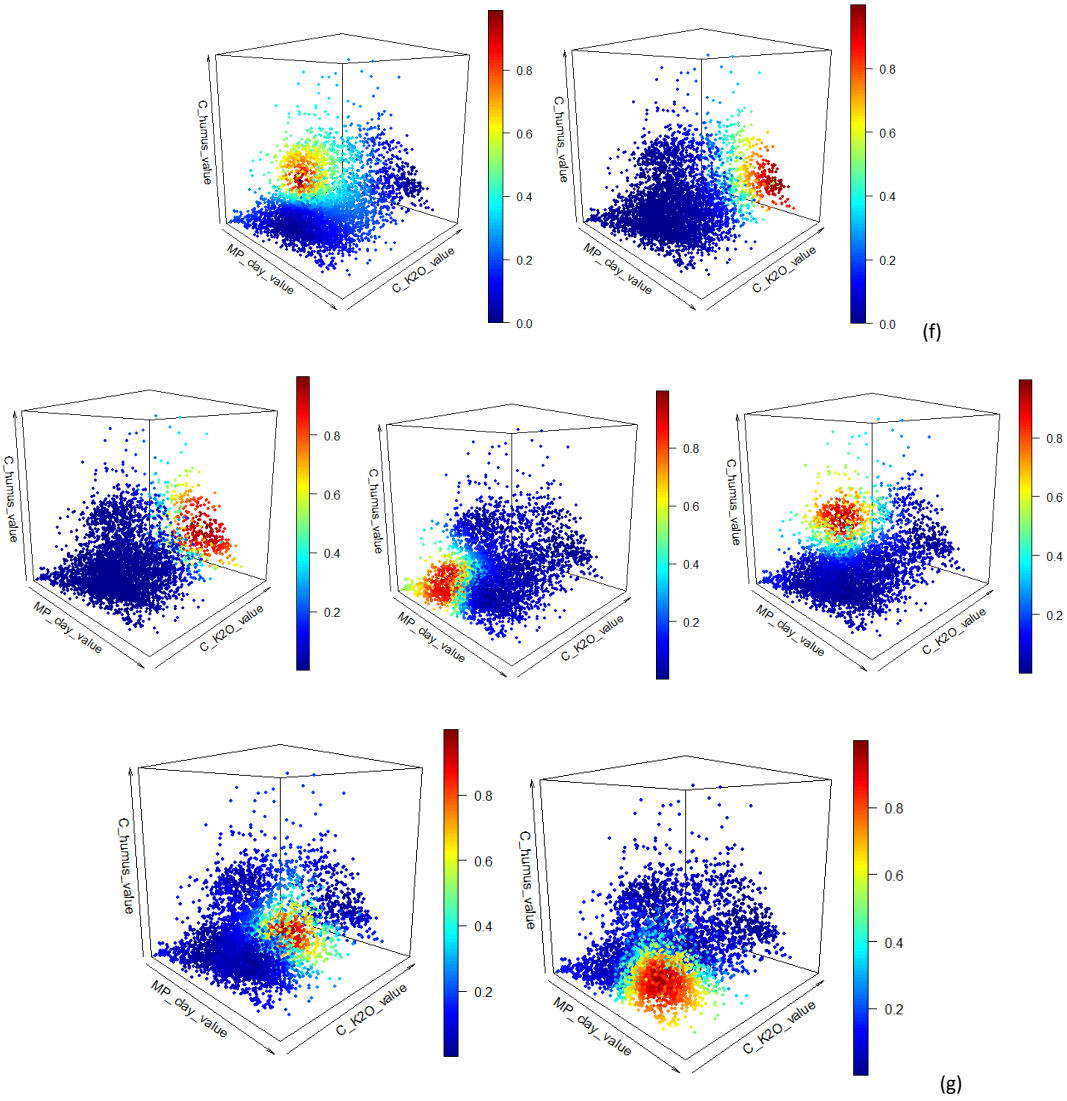
(b)


C_humus_value
C_K2O_value
MP_clay_value
C_K2O_value
MP_clay_value
klaster 1
klaster 2
klaster 3
klaster 4
klaster 5

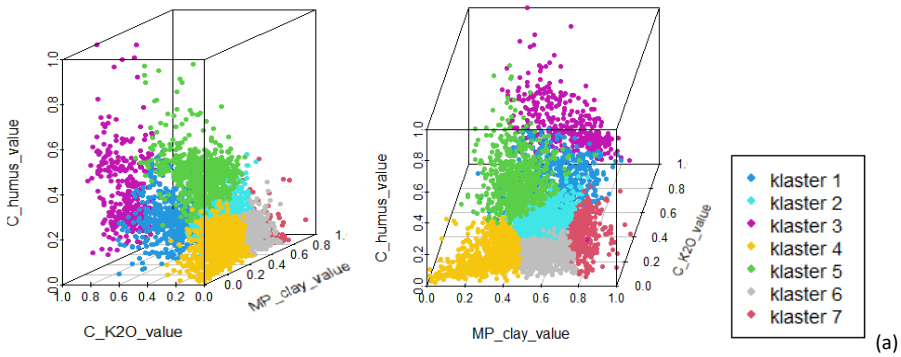
(c)

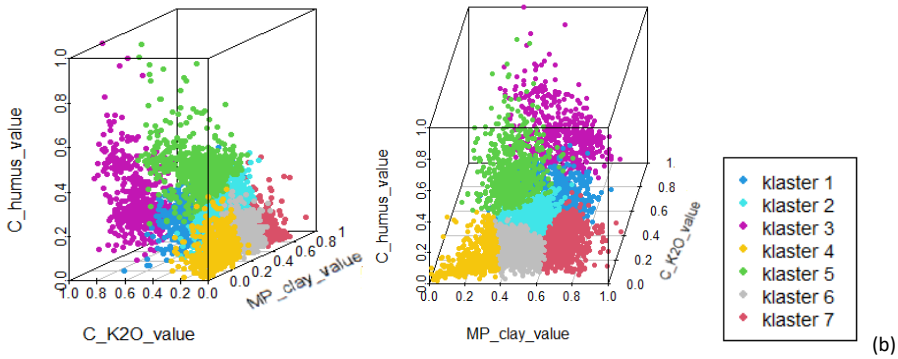

C_humus_value
MP_clay_value
C_K2O_value


C_humus_value
MP_clay_value
C_K2O_value
0.0
0.2
0.4
0.6
0.8
(d)
(e)

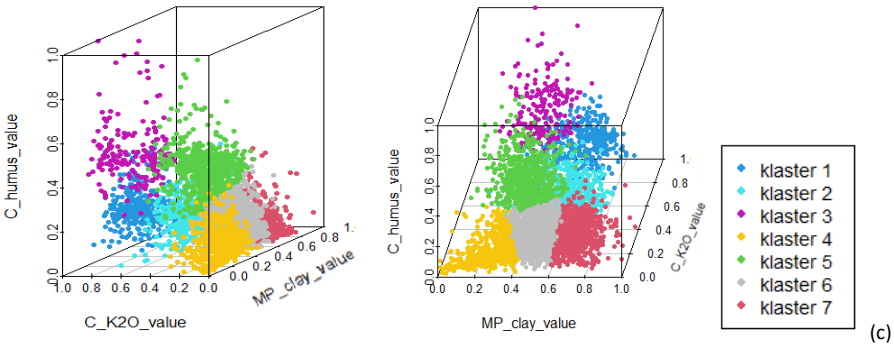


Slika 36. Rezultati klasterizacije dobijeni primjenom algoritama: (a) CLARA, (b) CLARANS, (c) *k*-means, (d) RFCMdd, (e) FCMRANS, (f) FCLARANS i (g) fuzzy *k*-means za *k* = 5, na tri pedološka parametra

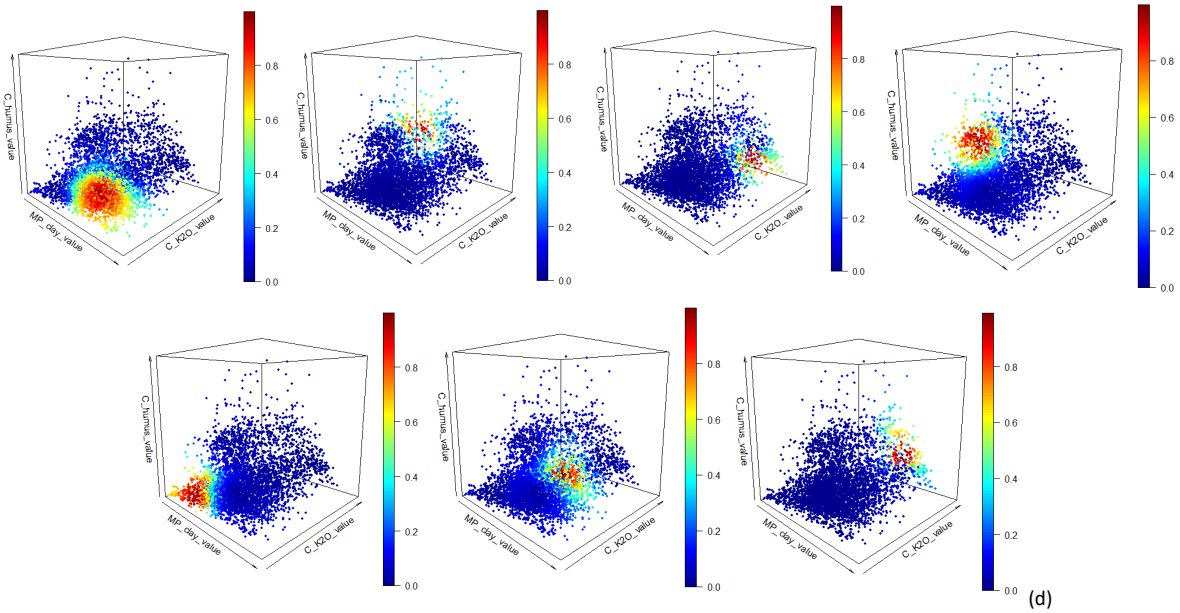



C_humus_value
C_K2O_value
MP_clay_value
klaster 1
klaster 2
klaster 3
klaster 4
klaster 5
klaster 6
klaster 7

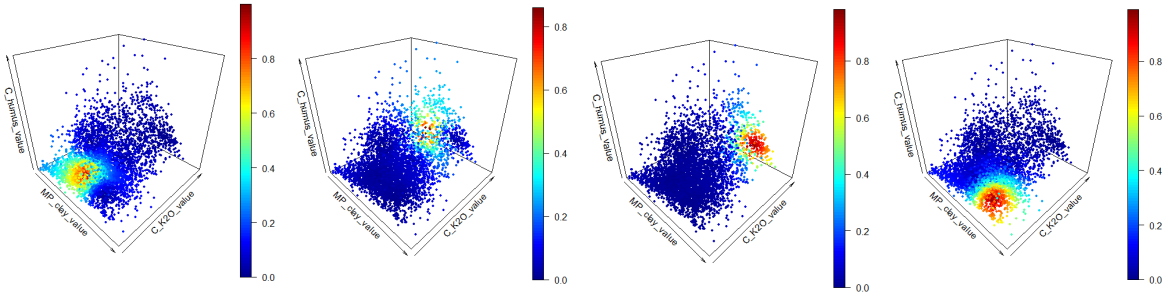
(b)

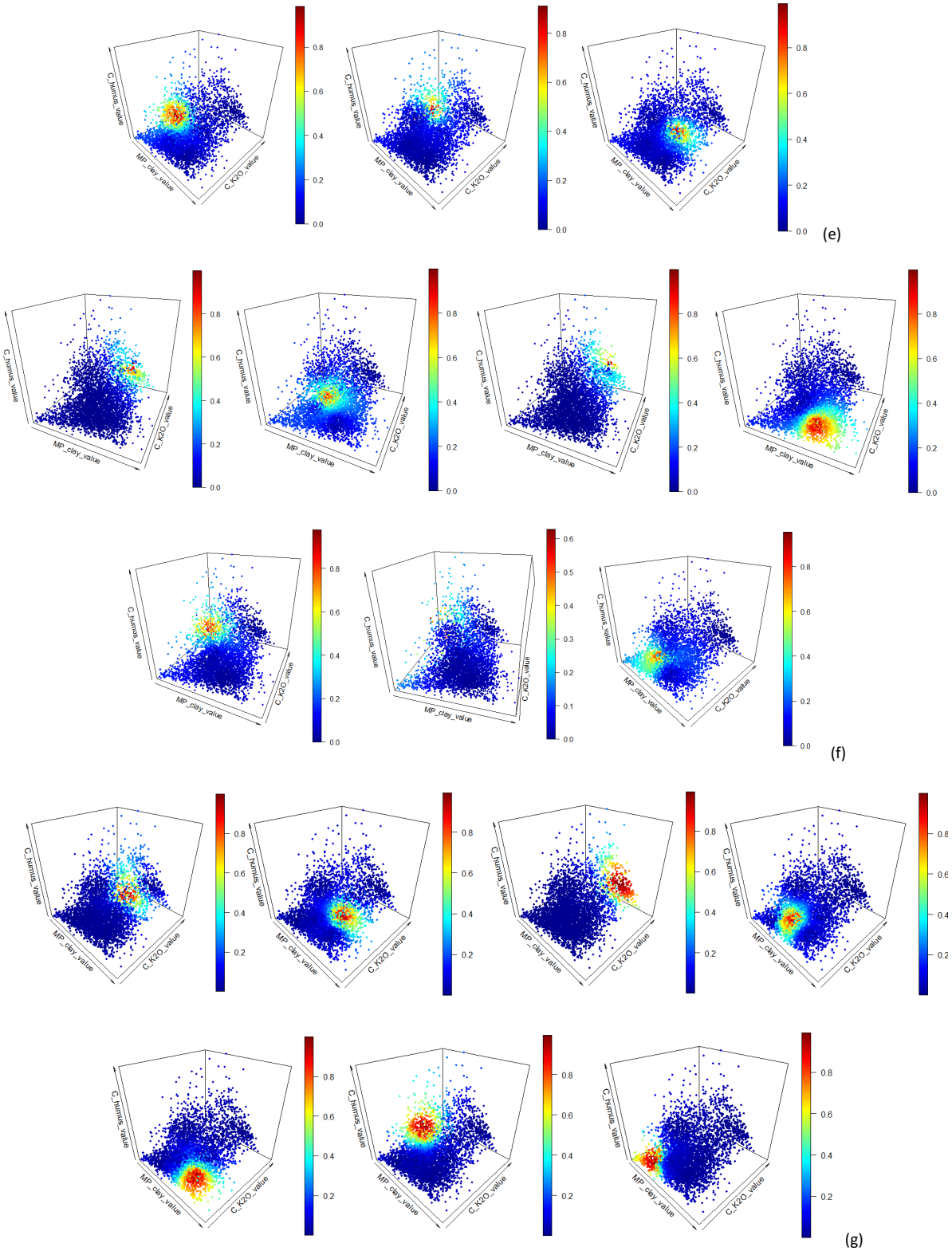

C_humus_value
C_K2O_value
MP_clay_value
klaster 1
klaster 2
klaster 3
klaster 4
klaster 5
klaster 6
klaster 7

(c)



(d)





Slika 37. Rezultati klasterizacije dobijeni primjenom algoritama: (a) CLARA, (b) CLARANS, (c) *k*-means, (d) RFCMdd, (e) FCMRANS, (f) FCLARANS i (g) fuzzy *k*-means za *k* = 7, na tri pedološka parametra

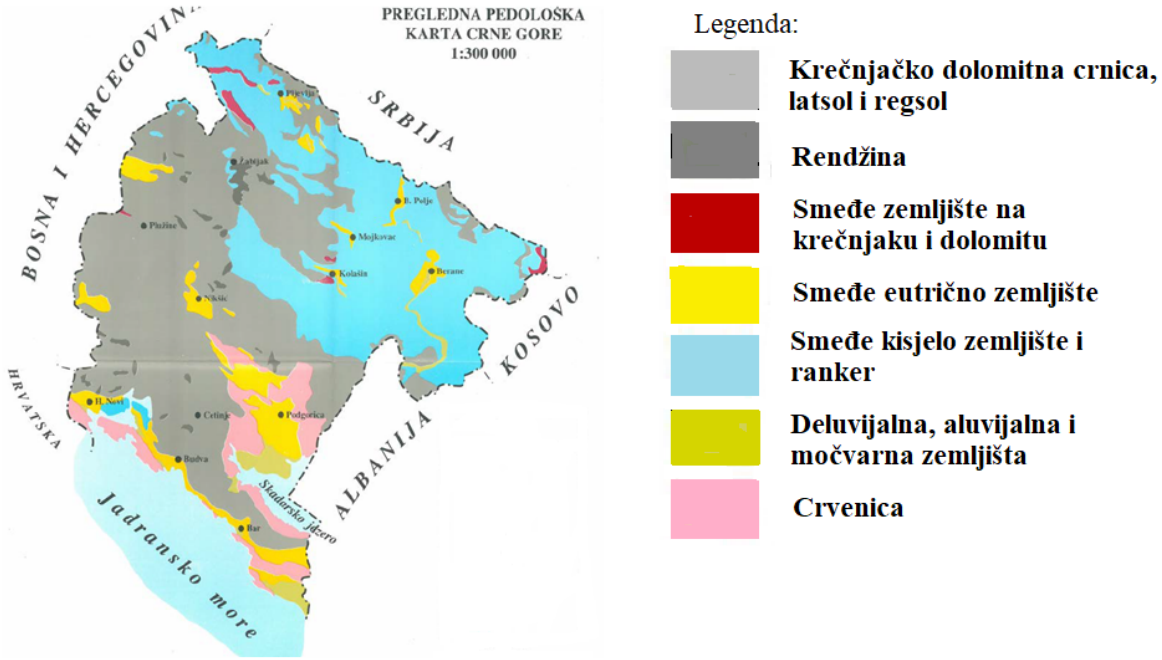
DBSCAN algoritam baziran na različitom principu u odnosu na ostale analizirane algoritme pa je i njegovom primjenom nad istim podacima dobijeno različito grupisanje podataka u odnosu na ostale analizirane algoritme. DBSCAN algoritam sa optimalnim vrijednostima ulaznih parametara daje klasterizaciju u jedan dominantni i veći broj manjih klastera uz prisutvo tačaka šuma. Dobijeni klasteri nisu uporedivih veličina, za razliku od klastera koji su rezultat primjene *k*-medoids i fuzzy *k*-medoids algoritama. Pored činjenice da se karakteristike zemljišta ne mijenjanju drastično kada se prelazi iz jednog tipa zemljišta u drugi, zaključuje se da su *k*-medoids i fuzzy *k*-medoids pogodniji za klasterizaciju pedoloških podataka. DBSCAN ne razdvaja različite tipove zemljišta koji imaju slične karakteristike graničnih tačaka, što je slučaj sa pedološkim podacima. Iz ovog razloga, DBSCAN će se izuzeti u daljoj analizi kao tehnika koja nije adekvatna za ovu vrstu baze podataka.

U odnosu na *k*-means i fuzzy *k*-means algoritme analizirane u radu [1], vizuelnim poređenjem dobijenih grafika na prethodnim primjerima, može se potvrditi sličnost sa graficima na kojima su predstavljeni dobijeni rezultati za *k*-medoids i fuzzy *k*-meodids. S obzirom na prethodno data objašnjenja vezana za veću robusnost na šum medoida u odnosu na centroide, prednost će se dati *k*-medoids i njegovim fuzzy oblicima. Iz tog razloga *k*-medoids i fuzzy *k*-medoids algoritmi su primjenljivi na istim pedološkim podacima Crne Gore kao u [1] i [28], u cilju identifikacije tipova zemljišta.

Svi zaključci u ovom poglavlju, donešeni primjenom analiziranih algoritama na dva, odnosno tri posmatrana parametra izdvojena iz pedološke baze, primjenljivi su i nad drugim parametrima posmatrane baze, pod uslovom da podaci nijesu međusobno visoko korelisani.

4 Pedološka mapa Crne Gore dobijena primijenom algoritama klasterizacije

Mape su najbolji način za predstavljanje pedoloških podataka i omogućavaju njihovu razumljivost široj javnosti. Na slici 38 je prikazana ekspertska pedološka tematska mapa Crne Gore sa 7 zastupljenih tipova zemljišta [1]. Mapa je ručno kreirana 2000. godine od strane dr Budimira Fuštića i dipl. ing Grujice Đuretića. Vizuelno poređenje ekspertske mape sa pedološkim mapama dobijenim primjenom odgovarajućih algoritama klasterizacije omogućilo je vizuelnu validaciju dobijenih rezultata.



Slika 38. Osnovna ekspertska, pedološka mapa Crne Gore sa zastupljenim tipovima zemljišta

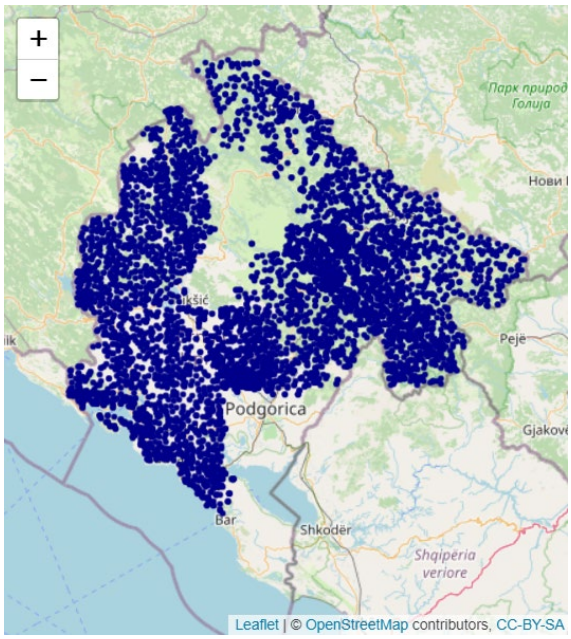
Na teritoriji Crne Gore postoji 7 tipova zemljišta i to:

- krečnjačko dolomitna crnica, latosol i regsol (47%),
- smeđe kisjelo zemljište i ranker (28%),
- smeđe eutrično zemljište (8%),
- crvenica (6%),
- deluvijalna, aluvijalna i močvarna zemljišta,
- rendžina,
- smeđe zemljište na krečnjaku i dolomitu.

Dva tipa zemljišta su najdominantnija, dok su ostala zastupljena u manjem procentu.

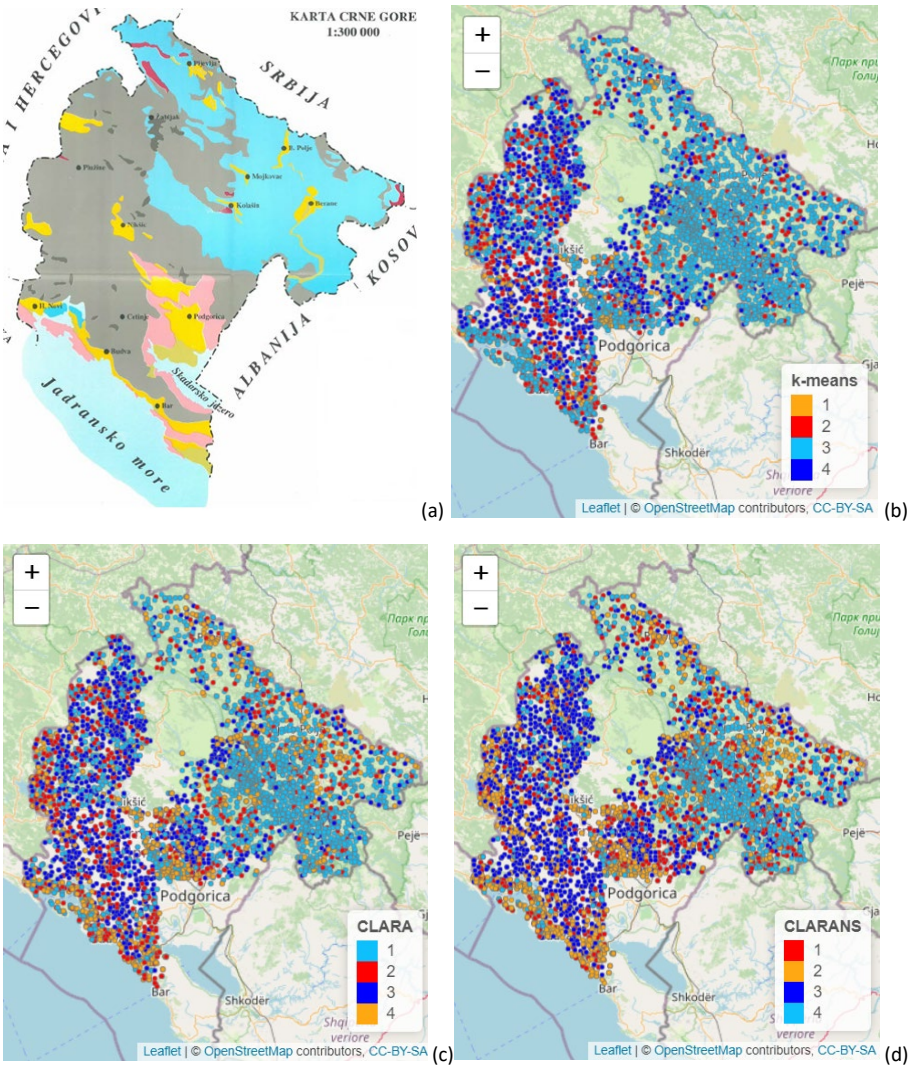
Validaciju rezultata je gotovo nemoguće odraditi bez vizuelizacije istih. Iz tog razloga, rezultati klasterizacije su prikazani na digitalizovanoj diskretnoj pedološkoj mapi. Za razvoj mapa korišćena je R biblioteka *leaflet*, koja ima mogućnost prikaza *OpenStreet* mape. Poređenjem sa ekspertskom tematskom mapom potvrdiće se uspješnost primjene analiziranih algoritama klasterizacije u identifikaciji 4 dominantna tipa zemljišta.

Pedološki podaci dostupni za klasterizuju su označeni plavim markerima na mapi datoj na slici 39. Jasno se uočava veliki broj nedostajućih podataka na cijeloj teritoriji Crne Gore. Predjeli prekriveni vodenim površinama takođe su bez definisanih vrijednosti uzoraka.



Slika 39. Podaci dostupni za klasterizaciju predstavljeni na diskretnoj dinamičkoj mapi

U radu [1], gdje je primijenjena *k*-means klasterizacija, za dobijanje pedološke tematske mape sa tipovima zemljišta teritorije Crne Gore je izdvojeno 6188 uzoraka opisanih sa 5 hemijskih parametara i klasterizovanih u 4 klastera. Na osnovu rezultata dobijenih na slici 18 (za CLARA i CLARANS), gdje su primijenjene metode srednje siluete, u ovom radu je uzeto optimalno $k = 4$. Za određivanje optimalnog k kod metode srednje siluete je korišćena ugrađena R funkcija *fviz_nbclust* iz paketa *factoextra*. Slika 40 (b) prikazuje rezultate klasterizacije primjenom *k*-means algoritma. Na slici 40 su diskretne mape kao rezultat klasterizacije istih podataka sa CLARA i CLARANS algoritmima. Razvijene su diskretne dinamičke mape sa markerima, gdje svaka boja predstavlja poseban klaster, u skladu sa *color bar*-om. Na sve tri mape se uočavaju 2 dominantna tipa zemljišta koja su zastupljena. *k*-means je bolje identifikovao svijetlo plavi klaster, dok su CLARA i CLARANS regiju oko Podgorica i duž Primorja izdvojili kao poseban narandžasti klaster.



Slika 40. (a) Osnovna pedološka mapa Crne Gore i dinamičke diskretne mape sa markerima dobijene primjenom: (b) *k*-means, (b) CLARA i (c) CLARANS algoritama za $k = 4$. Svaka boja predstavlja poseban klaster.

Poznato je da na teritorije Crne Gore postoji 7 tipova zemljišta, ali da korišćeni podaci (slika 39) kada se uporede sa ekspertskom mapom, ne sadrže uzorke zelenog klastera. Osim toga, jedan dio žutog i rozog tipa zemljišta nije obuhvaćen podacima. Dolazi se do zaključka da dostupni podaci opisani sa 5 hemijskih parametara sigurno nijesu dovoljni za preciznu identifikaciju zastupljenih tipova zemljišta. U ovom radu je korišćeno samo 5 parametara iz razloga što ostali parametri imaju dosta NA vrijednosti uzoraka, čijom bi se eliminacijom ukupan broj uzoraka za klasterizaciju smanjio skoro duplo, što ponovo rezultira lošiju klasterizaciju.

Pored toga što je baza nepotpuna i ima veliki broj podataka sa šumom, rezultati dobijeni na slici 40 pokazuju da postoji potencijal u upotrebi algoritama za klasterizaciju u obradi pedoloških podataka i automatizovanju dobijanja pedoloških mapa.

5 Zaključak

U ovom radu su predstavljene neke od tehnika klasterizacije i analizirana njihova primjenljivost na prostorne, pedološke podatke. Algoritmi klasterizacije formiraju klastere kao smislene grupe podataka, gdje su podaci unutar jednog klastera sličniji međusobno, u odnosu na one u drugim klasterima.

Analizirane su tri grupe algoritama klasterizacije. Principi po kojima svaka od njih klasterizuje podatke ilustrovani su njihovom primjenom na 2 i 3 izdvojena pedološka parametra. U prvu grupu algoritama su algoritmi koji grupišu podatke na osnovu njihove gustine raspodjele, kao što je DBSCAN. Kroz primjere je vizuelno pokazano da on ne predstavlja najbolje rješenje pri klasterizaciji pedoloških podataka. Drugu grupu čine *k*-medoids, a treću fuzzy *k*-medoids algoritmi. Njihovom primjenom nad istom grupom parametara ilustrovani su principi po kojima su ovi algoritmi implementirani. Takođe su se pokazali bolji u prevazilaženju šuma u odnosu *k*-means i fuzzy *k*-means koji su ranije primijenjeni nad istom bazom [1]. Dobijeni rezultati, za 2 i 3 pedološka parametra, i uporedivost sa rezultatima dobijenim *k*-means i fuzzy *k*-means klasterizacijom [1] nad istom bazom motivisali su njihovu primjenu nad većim brojem nekorelisanih parametara u cilju identifikacije tipova zemljišta u Crnoj Gori.

U ovom radu 5 digitalizovanih hemijskih parametara iz pedološke baze Crne Gore su iskorišćeni za dobijanje pedološke tematske mape Crne Gore. Podaci su klasterizovani za optimalni broj klastera $k = 4$. Razvijene pedološke diskretne mape sa markerima i njihova uporedivost sa diskretnim mapama dobijenim primjenom *k*-means algoritma [1], kao i ekspertskom mapom potvrdile su primjenljivost algoritama klasterizacije na pedološkim podacima i automatizovanju identifikacije tipova zemljišta.

Poznato je da veća količina podataka doprinosi boljim rezultatima data mining tehnika, samim tim i analiziranih algoritama klasterizacije. Iz tog razloga zaključuje se da nije dovoljno 6188 uzoraka opisanih samo sa 5 hemijskih parametara da bi se izvršila pravilna identifikacija 7 tipova zemljišta. U ovom istraživanju proširenje na ostale hemijske i mehaničko-fizičke karakteristike nije bilo moguće iz razloga što uključivanjem ostalih parametara i eliminacijom cijelih uzoraka kojim je makar jedan parametar označen sa NA, broj uzoraka za klasterizaciju bi se gotovo duplo smanjio, što ne bi dovelo do poboljšanja u rezultatima. Poboljšanjem baze i dostupnosti informacija o stvarnim vrijednostima NA polja bi mogli biti obuhvaćeni i ostali fizičko-hemijski parametri koji nijesu visoko korelisani, a nad njima biti primijenjeni analizirani algoritmi. Primjenljivost algoritama i zaključaka donešenih u ovom radu biće moguća i na pedološkim bazama drugih teritorija za identifikaciju zastupljenih tipova zemljišta, kao i za njihovo predstavljanje kroz dinamičke diskretne i raster pedološke mape.

6 Dodatak

Normalizacija podataka:

```
normalizacija <- function(x){(x-min(x))/(max(x)-min(x))}
pedoloski_podaci_normal <- apply(pedoloski_podaci,2,normalnizacija)
```

Zbog obimnosti u nastavku je dat kod samo za 2 parametra iz baze.

DBSCAN algoritam:

```
DBSCAN <- function(niz,eps,MinPts) {
        ClID=1
        br=0
        niz1=niz
        for (i in 1:nrow(niz1)) {
          tacka=niz1[i,]
          niz1=sirenjeN(niz1,i,ClID,eps,MinPts)
          for(k in 1:nrow(niz1)) {
            if(niz1[k,3]==ClID) {
              br=br+1
            }
            br
          }
          if(br>0) {
            ClID=ClID+1
          }
        }
        return (niz1)
      }

sirenjeN <- function(niz,p,ClID,eps,MinPts) {
        seeds=regionQuery(niz[p,1:2],niz,eps)
        point=niz[p,1:2]
        tacka=niz[p,]
        if (nrow(seeds)<MinPts) {
         for (m in 1:nrow(niz)) {
           if    (identical(as.numeric(tacka[1]),as.numeric(niz[m,1]))    &&
identical(as.numeric(tacka[2]),as.numeric(niz[m,2]))   &&   niz[m,3]==0   &&
(tacka[3]==0 || tacka[3]==-1)){
              niz[m,3]=-1 #NOISE
              }
           }
         }
         else {
           stariClID=0
           stariClID=stariKlasterID(seeds,niz)+150
           for(q1 in 1:nrow(seeds)) {
             for(m1 in 1:nrow(niz)) {
               if (identical(as.numeric(seeds[q1,1]),as.numeric(niz[m1,1]))
&& identical(as.numeric(seeds[q1,2]),as.numeric(niz[m1,2])))
                 if((niz[m1,3]==0 || niz[m1,3]==-1) && stariClID==0) {
                    niz[m1,3]=ClID

                 }
```

```
                else if(!stariClID==0) {
                  for (i1 in 1:nrow(seeds)) {
                    if(!seeds[i1,3]==0      &&      !seeds[i1,3]==-1      &&
!seeds[i1,3]==stariClID){
                      kl2=seeds[i1,3]
                      for (i2 in 1:nrow(niz)) {
                        if(niz[i2,3]==kl2){
                          niz[i2,3]=stariClID
                        }
                      }
                    }
                  }
                }
              }
            }
            seeds=poredjenje(niz,seeds)
          }
          seeds=seeds[-1,]

          if(nrow(seeds)>0){
              curP=gdata::first(seeds[])
              curNeigh=regionQuery(curP[1:2],niz,eps)
              curNeigh1=poredjenje(niz,curNeigh)
              if(curNeigh1[,3]==0 || curNeigh1[,3]==-1 && !curP[,3]==0) {
                curNeigh1[,3]=curP[,3]
                for(j in 1:nrow(niz)){
                  if(niz[j,3]==0 || niz[j,3]==-1){
                    m=poredjenje(curNeigh1,niz)
                  }
                }
                curN=curNeigh1
                if(nrow(curN)>=MinPts) {
                  for(p1 in 1:nrow(curN)) {
                    curN1=curN[p1,]
                    if(curN1[3]==0 || curN1[3]==-1){
                      if(curN1[3]==0){
                        seeds=rbind(seeds,curN1[])
                      }
                      for(m in 1:nrow(niz)){

if(identical(as.numeric(niz[m,1]),as.numeric(curN1[1]))                    &&
identical(as.numeric(niz[m,2]),as.numeric(curN1[2]))   &&   (niz[m,3]==0   ||
niz[m,3]==-1)){
                        niz[m,3]=curN1[3]
                        }
                      }
                    }
                  }
                }
              seeds=seeds[-1,]
              }
          }
        }
        for(n1 in 1:nrow(niz)){
          if(n1==niz[n1,3]){
            niz[n1,3]=-1 #NOISE
```

```
          }
        }
        return(niz)
      }

regionQuery <- function(point=c(),P2,eps) {
        neighborhood=rbind(point)
        cc=0
        for (i in 1:nrow(P2)) {
          temp = sqrt((point[1]-P2[i,1])^2+(point[2]-P2[i,2])^2)
          t=c(0,0)
          if (temp<=eps) {
            cc=cc+1
            t=P2[i,1:2]
            t1=P2[i,]
            if (!identical(t,point)) {
                neighborhood=rbind(neighborhood,t)
            }
          }
        }
        return(cbind(neighborhood,0))
      }

stariKlasterID <- function(seeds,niz){
        stariClID=0
        for(q1 in 1:nrow(seeds)) {
          for(m1 in 1:nrow(niz)) {
            if(identical(as.numeric(seeds[q1,1]),as.numeric(niz[m1,1]))&&
               identical(as.numeric(seeds[q1,2]),as.numeric(niz[m1,2]))   &&
!niz[m1,3]==0 && !niz[m1,3]==-1){
              stariClID=niz[m1,3]
              break
            }
          }
        }
        return(stariClID)
      }

poredjenje <- function(niz,curN){
      for(q2 in 1:nrow(curN)) {
        for(q3 in 1:nrow(niz)){
          if(identical(as.numeric(niz[q3,1]),as.numeric(curN[q2,1]))&&
            identical(as.numeric(curN[q2,2]),as.numeric(niz[q3,2]))      &&
!niz[q3,3]==0 && (curN[q2,3]==0 ||
            curN[q2,3]==-1)){
            curN[q2,3]=niz[q3,3]
            }
          }
        }
        return(curN)
        }
```

CLARA algoritam:

```
CLARA <- function(sub,brK){ # sub ulazni skup podataka, brK broj klastera
        sub=data.frame(sub)
        subSet=as.matrix(dplyr::sample_n((sub),40+2*brK))
        Vmam=0
        iter=0
        sumR=0
        b=cluster::pam(subSet[,1:2],brK)
        med=cbind(b$medoids,as.matrix(b$id.med,nrow=k,ncol=1))
        med=as.matrix(med)
        sub1=as.matrix(cbind(sub,0))
        for (k in 1:brK){
          for (n in 1:max(dim(sub1))){
            if(identical((as.numeric(sub1[n,1])),as.numeric(med[k,1]))    &&
identical(as.numeric(sub1[n,2]),as.numeric(med[k,2]))){
              sub1[n,3]=k
            }
          }
        }
        print(sub1)
        for (m in 1:nrow(sub1)){
          diss=0
          dissN=0
          for (j in 1:brK){
            if(!identical(sub1[m,1],med[j,1])||
!identical(sub1[m,2],med[j,2])){
              dissN=as.numeric(sqrt((as.numeric(sub1[m,1])-
as.numeric(med[j,1]))^2+(as.numeric(sub1[m,2])-as.numeric(med[j,2]))^2))
              if (diss!=0 && as.numeric(dissN)<as.numeric(diss)){
                diss=dissN
                pripada=j
                }
              else if (diss==0){
                diss=dissN
                pripada=j
                }
              }
            }
            pripadaM=pripada
            if (as.numeric(sub1[m,3])==0){
              sub1[m,3]=pripadaM
            }
            sumR=sumR+diss
        }
        suma=sumR/max(dim(sub))
        print(suma)
        print(med)
        return(sub1[,3])
      }
```

CLARANS algoritam:

```
CLARANS <- function(X,k) { # X ulazni skup podataka, k broj klastera
  mincost=100000
  n=nrow(X)
  p=round(1.25*k*(n-k)/100)-1
  maxneighbor=max(p,250)
  V=matrix(nrow=k,ncol=3)
  numlocal=2
  for(i in 1:numlocal) {
    b=cluster::pam(X[,1:2],k)
    V=cbind(b$medoids,as.matrix(b$id.med,nrow=k,ncol=1))
    V=as.matrix(V)
    j=1
    currCost=costClarans(X,V)
    num=as.matrix(V[,3])
    noSelectedObj=X[-as.numeric(num),]
      k1=k
      brojac=k1
      niz=matrix(1:k1,nrow=1,ncol=k1)
      if(brojac>=1 ){
        m1=sample(k1,1)
        m=niz[m1]
        niz=niz[-m1]
        k1=k1-1
        repeat{
          M1=sample(1:nrow(noSelectedObj),1)
          neighbor=matrix(noSelectedObj[M1,],nrow=1,ncol=3)
          pozX=as.numeric(neighbor[1,3])
          Vneighbor=V
          Vneighbor[m,]=neighbor
          Vneighbor=matrix(Vneighbor,nrow=k,ncol=3)
          newCost=costClarans(X,Vneighbor)
          if(currCost>newCost) {
            V=Vneighbor
            currCost=newCost
            j=1
            num=as.matrix(V[,3])
            noSelectedObj=X[-as.numeric(num),]
          } else {
            j=j+1
            num=rbind(num,pozX)
            noSelectedObj=X[-as.numeric(num),]
          }
          if(j>maxneighbor) {
            break
          }
        }
        brojac=brojac-1
        if (mincost>currCost){
          minCost=currCost
          bestNode=V
          break
        }
        brojac=brojac-1
      }
```

```
      i=i+1
  }
  return(bestNode)
}

costClarans <- function(X,V) { # X ulazni skup podataka, V skup medoida
  V=as.matrix(V)
  sum=0
  for(n1 in 1:nrow(V)){
    diss=0
    for(n2 in 1:nrow(X)){
      diss1=0
      if(!identical(as.numeric(V[n1,1]),as.numeric(X[n2,1]))              ||
!identical(as.numeric(V[n1,2]),as.numeric(X[n2,2]))) {
        diss1=as.numeric(sqrt((as.numeric(V[n1,1])-
as.numeric(X[n2,1]))^2+(as.numeric(V[n1,2])-as.numeric(X[n2,2]))^2))
      }
      diss=diss+diss1
    }
    sum=sum+diss
  }
  currCost=sum/nrow(X)
  return(currCost)
}
```

RFCMdd algoritam:

```
RFCMdd <- function(x,fuzzifier,outliers,Pobjects,itermax,k) {
# x ulazni skup podataka, k broj klastera
        V=InitRand(x,k)
        vm=V
        k=nrow(V)
        m=fuzzifier
        X=as.matrix(x)
        h=outliers
        duzina=nrow(X)
        u=matrix(nrow=duzina,ncol=k)
        vm_old=matrix(0,nrow=k,ncol=3)
        iter=0
        red.br=as.matrix(vm[,3])
        q=matrix(nrow=1,ncol=k)
        repeat {
          u=calculate_membership(X,vm,m)
          novoX=noviNiz(X,vm,h,m)
          vm_old=vm
          nX=nrow(novoX)
          rast2=matrix(nrow=Pobjects,ncol=k)
          q=matrix(nrow=1,ncol=k)
          XpIndex=calculateXp(X,vm,m,Pobjects
          for(i in 1:k){
            Xp=X[as.numeric(XpIndex[,i]),]
            for(p2 in 1:Pobjects){
              rast1=0
              rast=0
              for(j2 in 1:nrow(novoX)) {
```

```
                if
(!identical(as.numeric(Xp[p2,1]),as.numeric(novoX[j2,1]))                ||
!identical(as.numeric(Xp[p2,2]),as.numeric(novoX[j2,2]))) {
                  rast=sqrt((as.numeric(Xp[p2,1])-
as.numeric(novoX[j2,1]))^2+(as.numeric(Xp[p2,2])-
as.numeric(novoX[j2,2]))^2)
                  rastN=(u[novoX[j2,3],i]^m)*rast
                  rast1=rast1+rastN
                }
              }
              rast2[p2,i]=rast1
            }
            q[i]=NNTbiomarker::argmin(rast2[,i])
            for(c in 1:nrow(vm)){
              if(identical(as.numeric(q[i]),as.numeric(q[c]))            ||
identical(as.numeric(XpIndex[q[i],i]),as.numeric(vm[c,3]))) {
                rast2[q[i],i]=rast2[q[i],i]*10000
                q[i]=NNTbiomarker::argmin(rast2[,i])
              } else {
                vm[i,]=X[XpIndex[q[i],i],]
                red.br=as.matrix(vm[,3])
              }
            }
          }
          status=0
          for(i1 in 1:k){
            for(i2 in 1:k){
              if(identical(as.numeric(vm[i1,1]),as.numeric(vm_old[i2,1]))
&&     identical(as.numeric(vm[i1,2]),as.numeric(vm_old[i2,2]))          &&
identical(as.numeric(vm[i1,3]),as.numeric(vm_old[i2,3]))){
                status=status+1
              }
            }
          }
          iter=iter+1

          if (iter>=itermax || status==nrow(vm)){
            break
          }
        }
        return (vm)
      }
```

FCMRANS algoritam:

```
FCMRANS <- function(X,k,outliers,Pobjects,fuzzifier,itermax) {
# x ulazni skup podataka, k broj klastera
        vm=InitRand(X,k)
        h=outliers
        n=nrow(X)
        maxneighbor=1.25*k*(n-k)/100
        m=fuzzifier
        iter=0
        vm_old=as.matrix(0,nrow=k,ncol=3)
        status=1
```

```
        repeat {
          u=calculate_membership(X,vm,m)
          novoX=noviNiz(X,vm,h,m)
          novoX=as.matrix(novoX)
          vm_old<-vm
          nX=nrow(novoX)
          jj=0
          red.br=as.matrix(vm[,3])
          number=red.br
          noSelectedObj=X[-as.numeric(number),]
          num=sample(k,1)
          repeat {
            Ecost=0
            XpIndex=calculateXp(X,vm,m,Pobjects)
            Xp=X[as.numeric(XpIndex[,num]),]
            newV=change1medoid(Xp,vm,num)
            poz=as.numeric(newV[num,3])
            pozX=as.numeric(Xp[poz,3])
            newV[num,]=X[pozX,]
            jj=jj+1
            r1=matrix(nrow=nX,ncol=1)
            for(j in 1:nrow(novoX)){
             if((!identical(as.numeric(novoX[j,1]),as.numeric(newV[num,1]))
||      !identical(as.numeric(novoX[j,2]),as.numeric(newV[num,2])))      ||
(!identical(as.numeric(novoX[j,1]),as.numeric(vm[num,1]))                ||
!identical(as.numeric(novoX[j,2]),as.numeric(vm[num,2])))) {
              d1=sqrt((as.numeric(novoX[j,1])-
as.numeric(newV[num,1]))^2+(as.numeric(novoX[j,2])-
as.numeric(newV[num,2]))^2)
              d2=sqrt((as.numeric(novoX[j,1])-
as.numeric(vm[num,1]))^2+(as.numeric(novoX[j,2])-as.numeric(vm[num,2]))^2)
              if(u[novoX[j,3],num]!=0){
                r1[j]=as.numeric((d1-d2)*u[novoX[j,3],num]^m)
              } else {
                r1[j]=0
              }
            }
          }
          E=sum(r1)
          Ecost=E
          if(as.numeric(Ecost)<0){
            vm=newV
            jj=1
            number=as.matrix(vm[,3])
            noSelectedObj=X[-as.numeric(number),]
            num=sample(k,1)
          } else {
            number=rbind(number,pozX)
            noSelectedObj=X[-as.numeric(number),]
          }
          if(jj>maxneighbor) {
            break
          }

        }
        iter=iter+1
```

```
          status=0
          for(i3 in 1:k) {
            for(i4 in 1:k){
              if(identical(as.numeric(vm[i3,1]),as.numeric(vm_old[i4,1]))
&&     identical(as.numeric(vm[i3,2]),as.numeric(vm_old[i4,2]))          &&
identical(as.numeric(vm[i3,3]),as.numeric(vm_old[i4,3]))) {
                status=status+1
              }
            }
          }

      if(status==nrow(vm) || iter>=itermax) {
            break
          }
        }
        return (vm)
      }
```

FCLARANS algoritam:

```
FCLARANS <- function(X,fuzzier,k) { # x ulazni skup podataka, k broj klastera
        V=InitRand(X,k)
        n=nrow(X)
        maxneighbor=1.25*k*(n-k)/100
        m=fuzzier
        iter=0
        V_old=matrix(0,nrow=k,ncol=3)
        jj=1
        u=calculate_membership(X,V,m)
        red.br=as.matrix(V[,3])
        number=(red.br)
        noSelectedObj=X[-as.numeric(red.br),]
        repeat {
          num=sample(k,1)
          newV=change1medoid(noSelectedObj,V,num)
          poz=as.numeric(X[as.numeric(newV[num,3]),3])
          newV[num,]=X[poz,]
          uN=calculate_membership(X,newV,m)
          u1=calculate_membership(X,V,m)
          jj=jj+1
          E=costF(X,newV,uN,V,u1,m)
          if(E<0){
            V[num,]=X[poz,]
            u=uN
            red.br=rbind(red.br,poz)
            noSelectedObj=X[-as.numeric(newV[,3]),]
            jj=1
          }  else {
            number=rbind(number,poz)
            noSelectedObj=X[-as.numeric(number),]
          }
          if (jj>maxneighbor){
            break
          }
        }
```

```
        return(V)
      }
```

Funkcija za inicijalizaciju skupa medoida:

```
InitRand <- function (X,k) { # x ulazni skup podataka, k broj klastera
      duzina=nrow(X)
      V=NULL
      M1=sample(1:nrow(X),1)
      medoid1=matrix(X[M1,],nrow=1,ncol=3)
      sprintf('random medoid je: %i. element skupa', M1)
      maxiter=k
      V=as.matrix(medoid1)
      iter=1
      noSelectedObj=X[-M1,]
      dist1=matrix(nrow=duzina,ncol=nrow(V))
      pozX=0
      poz=0
      repeat {
        maxdist=0
        for(i in 1:(duzina-nrow(V))) {
          rast=0
          distN=0
          novi_med=NULL
          for(j in 1:nrow(V)) {
              distN=sqrt((as.numeric(noSelectedObj[i,1])-
as.numeric(V[j,1]))^2+(as.numeric(noSelectedObj[i,2])-
as.numeric(V[j,2]))^2)
              if(rast>distN || rast==0) {
                rast=distN
                pozX=noSelectedObj[i,3]
                poz=i
              }
            }
          mdist=rast
          pozicija1=pozX
          pozicija=poz
          if(mdist>maxdist){
            r=runif(1)
            if(r<((mdist-maxdist)/mdist)){
              q=pozX
              maxdist=mdist
              }
            }
          }
        novi_med=as.numeric(X[q,])
        br=0
        for(i in 1:nrow(V)){
          if(!identical(as.numeric(novi_med[1]),as.numeric(V[i,1]))      &&
!identical(as.numeric(novi_med[2]),as.numeric(V[i,2]))){
            br=br+1
          }
        }
        if(br==nrow(V)){
          V=rbind(V,novi_med)
          noSelectedObj=noSelectedObj[-pozicija,]
```

```
          iter=iter+1
        }
        if(iter==k) {
          break
          }
        }
      return(as.matrix(V))
      }
```

Funkcija za računanje stepena pripadnosti klasteru:

```
calculate_membership <- function(x,currMedoid,m) {
# x ulazni skup podataka, currMedoid trenutni skup medoida, m fuzzifier
        X=x
        k=as.numeric(nrow(currMedoid))
        V=currMedoid
        duzina=as.numeric(nrow(x))
        rastojanje=matrix(nrow=duzina,ncol=k)
        kvadratRast=matrix(nrow=duzina,ncol=k)
        u=matrix(nrow=duzina,ncol=k)
        sumRast=0
        kvadrat_sumrast=0
        sumaRast=0
        for(j in 1:duzina) {
          for(kk in 1:k) {
            if(!identical(as.numeric(X[j,1]),as.numeric(V[kk,1]))        ||
!identical(as.numeric(X[j,2]),as.numeric(V[kk,2]))) {
              sumRast=1/sqrt((as.numeric(X[j,1])-
as.numeric(V[kk,1]))^2+(as.numeric(X[j,2])-as.numeric(V[kk,2]))^2)
              kvadrat_sumrast=as.numeric(sumRast)^(1/(m-1))
              sumaRast=sumaRast+kvadrat_sumrast
            }
          }
          for(i in 1:k){
            if(!identical(as.numeric(X[j,1]),as.numeric(V[i,1]))         ||
!identical(as.numeric(X[j,2]),as.numeric(V[i,2]))) {
              rastojanje[j,i]=1/sqrt((as.numeric(X[j,1])-
as.numeric(V[i,1]))^2+(as.numeric(X[j,2])-as.numeric(V[i,2]))^2)
              kvadratRast[j,i]=as.numeric(rastojanje[j,i])^(1/(m-1))
              u[j,i]=as.numeric(kvadratRast[j,i]/sumaRast)
            } else {
              u[j,i]=0
            }
          }
          sumRast=0
          sumaRast=0
        }
        return(u)
      }
```

Sljedeće funkcije nalaze novi niz iz koga se isključuju tačke šuma:

```
noviNiz <- function(X,V,threshold,m) {
# X ulazni skup podataka, V trenutni skup medoida, m fuzzifier
        n=nrow(X)
        k=nrow(V)
```

```
        harm=round(n*(1-threshold))
        nn1=matrix(0,nrow(X),ncol=2)
        harmN=matrix(0,nrow=harm,ncol=2)
        nn=calculate_harm(X,V,m)
        nn1=cbind(nn,1:nrow(nn))
        sortharm=(sort(nn1[,1],decreasing = TRUE, index.return = TRUE))
        sortM=as.matrix(sortharm$x)
        sortIndex=as.matrix(sortharm$ix)
        harmN=cbind(sortM,sortIndex)
        i3=0
        bezV=matrix(0,nrow=(n-k),ncol=3)
        for(i1 in 1:nrow(harmN)){
          br=0
          for(i2 in 1:k){
            if(!identical(as.numeric(harmN[i1,2]),as.numeric(V[i2,3]))){
              br=br+1
            }
          }
          if(br==k){
            i3=i3+1
            bezV[i3,1]=X[as.numeric(harmN[i1,2]),1]
            bezV[i3,2]=X[as.numeric(harmN[i1,2]),2]
            bezV[i3,3]=X[as.numeric(harmN[i1,2]),3]
          }
        }
        novoX=matrix(0,nrow=harm,ncol=3)
        for(j1 in 1:harm){
          novoX[j1,1]=bezV[j1,1]
          novoX[j1,2]=bezV[j1,2]
          novoX[j1,3]=bezV[j1,3]
        }
        return(novoX)
      }

calculate_harm <- function(X,vm,m) {
        k=nrow(vm)
        n=nrow(X)
        outL=NULL
        novoX=NULL
        harm2=matrix(nrow=nrow(X),ncol=1)
        harm1=0
        for(i in 1:nrow(X)) {
          br=1
          harm=0
          harm1=0
          for (j in 1:k) {
            if(!identical(as.numeric(X[i,1]),as.numeric(vm[j,1]))        ||
!identical(as.numeric(X[i,2]),as.numeric(vm[j,2]))){
              harm=as.numeric(sqrt((as.numeric(X[i,1])-
as.numeric(vm[j,1]))^2+(as.numeric(X[i,2])-as.numeric(vm[j,2]))^2))^(1/(1-
m))
            }
            harm1=harm1+harm
          }
          harm2[i]=harm1^(1-m)
```

```
        }
        return(harm2)
}
```

Funkcija koja pronalazi skup sa P objekata koji su najbliži trenutnom skupu medoida:

```
calculateXp <-function(X,vm,m,Pobjects) {
        n=nrow(X)
        k=nrow(vm)
        Xp=matrix(nrow=Pobjects,ncol=k)
        XpIndex=matrix(nrow=Pobjects,ncol=k)
        sortM=matrix(nrow=(n-k), ncol=k)
        sortIndex=matrix(nrow=(n-k), ncol=k)
        sortIndex1=matrix(nrow=(n-k), ncol=k)
        sortmembr=matrix(nrow=(n-k),ncol=1)
        X1=matrix(nrow=(n-k),ncol=3)
        vmIndex=as.matrix(vm[,3])
        X1=X[-as.numeric(vmIndex),]
        red.br=as.matrix(X1[,3])
        u=calculate_membership(X1,vm,m)
        u1=matrix(nrow=(n-k),ncol=2)
        u1[,2]=red.br
        for(j1 in 1:k) {
          u1[,1]=u[,j1]
          sortmembr=(sort(u1[,1],decreasing = TRUE, index.return = TRUE))
          sortM[,j1]=as.matrix(sortmembr$x)
          sortIndex[,j1]=as.matrix(sortmembr$ix)
          sortIndex1[,j1]=as.matrix(u1[sortmembr$ix,2])
        }
        for(j1 in 1:k) {
          for(p1 in 1:Pobjects) {
              XpIndex[p1,j1]=as.numeric(X[as.numeric(sortIndex1[p1,j1]),3])
          }
        }
        return(XpIndex)
        }
```

Funkcija koja ažurira jedan od medoida:

```
change1medoid <- function(x,currMedoid,num) {
        k=nrow(currMedoid)
        n=nrow(x)
        randM=sample(n,1)
        newM=NULL
        newM=cbind(x[randM,1],x[randM,2],randM)
        brojac=0
        xNew=cbind(x,0)
        Cnew=currMedoid
        for(b in 1:k){
          if(identical(as.numeric(newM[1]),as.numeric(Cnew[b,1]))         &&
identical(as.numeric(newM[2]),as.numeric(Cnew[b,2]))){
            brojac=brojac+1
          }
        }
        if(brojac!=0) {
          b=0
```

```
          while(b<k){
            if(identical(as.numeric(newM[1]),as.numeric(Cnew[b,1]))       &&
identical(as.numeric(newM[2]),as.numeric(Cnew[b,2]))){
              randM=sample(n,1)
              newM=cbind(x[randM,1],x[randM,2],randM)
            }
            b=0
          }
        Cnew[num,]=newM
        }
        if(brojac==0) {
          Cnew[num,]=cbind(x[randM,1],x[randM,2],randM)
        }
        return(Cnew)
      }
```

Funkcija cijene za FCLARANS:

```
costF <- function(X,newV,uN,V,u,m) {
        E1=0
        E=0
        r2=0
        n=nrow(X)
        k=nrow(V)
        for(j in 1:n){
          for(i in 1:k){
            if((!identical(as.numeric(X[j,1]),as.numeric(newV[k,1]))      ||
!identical(as.numeric(X[j,2]),as.numeric(newV[k,2])))                     ||
(!identical(as.numeric(X[j,1]),as.numeric(V[k,1]))                        ||
!identical(as.numeric(X[j,2]),as.numeric(V[k,2])))){
              d1=sqrt((as.numeric(X[j,1])-
as.numeric(newV[k,1]))^2+(as.numeric(X[j,2])-as.numeric(newV[k,2]))^2)
              d2=sqrt((as.numeric(X[j,1])-
as.numeric(V[k,1]))^2+(as.numeric(X[j,2])-as.numeric(V[k,2]))^2)
              d11=d1*u[j,i]^m
              d22=d2*uN[j,i]^m
              r1=d11-d22
            }
              r2=r2+r1
          }
          E1=E1+r2
          r2=0
        }
        E=E1
        return(E)
      }
```

Literatura

[1] E.Hot, "Klasterizacija i vizuelizacija pedoloških podataka korišćenjem data-mining tehnika i prostorne interpolacije," *magistarski rad, Podgorica 2017*.

[2] M. Vukčević, V. Popović – Bugarin, E. Dervić, "DBSCAN and CLARA clustering algorithms and their usage for the soil data clustering," *IEEE 8th Mediterranean Conference on Embedded Computing (MECO)*, Budva, Montenegro, pp. 456-461, 10-14 June 2019.

[3] Nameirakpam Dhanachandra, Khumanthem Manglem and Yambem Jina Chanu, "Image Segmentation using K-means Clustering Algorithm and Subtractive Clustering Algorithm," *Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)*

[4] Jharna Majumdar, Sneha Naraseeyappa and Shilpa Ankalaki, " Analysis of agriculture data using data mining techniques: application of big data," *Springer Open, Published online*, 5 July 2017

[5] Yongli Liu, Jingli Chen, Shuai Wu, Zhizhong Liu, and Hao Chao, "Incremental fuzzy C medoids clustering of time series data using dynamic time warping distance," *Published online* 2018 May 24. doi: 10.1371/journal.pone.0197499

[6] Sampreeti Ghosh, Sushmita Mitra, Rana Dattagupta, " Fuzzy clustering with biological knowledge for gene selection ", *Applied Soft Computing*, vol. 16, March 2014.

[7] Upa Gupta, Kulsawasd Jitkajornwanich, Ramez Elmasri, Leonidas Fegaras, "Adapting K-Means Clustering to Identify Spatial Patterns in Storms," *Conference: 2016 IEEE International Conference on Big Data (Big Data)*, December 2016.

[8] M. Ester, H. P. Kriegel, J. Sander, X. Xu: "A Density-Databased Algorithm for Discovery Clusters in Large Spartial Databases with Noise," *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining* (KDD-96), 1996.

[9] Jelili Oyelade, Itunuoluwa Isewon, Olufunke Oladipupo, Onyeka Emebo, Zacchaeus Omogbadegun, Olufemi Aromolaran, Efosa Uwoghiren, Damilare Olaniyan, Obembe Olawole: „Data Clustering: Algorithms and Its Applications," *19th International Conference on Computational Science and Its Applications* (ICCSA), 2019.

[10] Raymond T. Ng and Jiawei Han, "CLARANS: A Method for Clustering Objects for Spatial Data Mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5, September/October 2002

[11] L. Kaufman-a and P. Rousseeuw, „Clustering Large Data Sets,“ *In: Pattern Recognition in Practice, Elsevier*, pp 425–437, DOI 10.1016/b978-0-444-87877-9.50039-x, 1986.

[12] Vijaya Sagvekar, Vidya Sagvekar, & Kalpana Deorukhkar, “Performance assessment of CLARANS: A Method for Clustering Objects for Spatial Data Mining,” *Global Journal of Engineering, Design & Technology, Pubished by: Global Institute for Research / Education*, vol. 2 (6):1-8, November/December 2013.

[13] Mohamed A. Mahfouz, and M. A. Ismail, “Fuzzy Relatives of the CLARANS Algorithm With Application to Text Clustering,” *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 3, no. 1, 2009

[14] Raghu Krishnapuram, Anupam Joshi, Olfa Nasraoui and Liyu Yi, “Low-Complexity Fuzzy Relational Clustering Algorithms for Web Mining”, *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, August 2001.

[15] C. Cassisi, P. Montalto, M. Aliotta, A. Cannata and A. Pulvirenti: “Similarity Measure and Dimensionality Reducstion Techniques For Time Series Data Mining,” Pp 71-96, September 12th, 2012.

[16] Pradipta Maji, Sankar K. Pal, „Rough-Fuzzy C-Medoids Algorithm and Selection of Bio-Basis for Amino Acid Sequence Analysis,“ *IEEE Transactions on Knowledge and Data Engineering*, 19(6):859-872, July 2007

[17] R.Krishnapuram, R.Joshi, A.Nasraoui and O.Yi, “Low-complexity fuzzy relational clustering algorithms for Web mining,” Fuzzy Systems, IEEE Transactions, vol.9, pp. 595--607, Aug 2001.

[18] Marcio Trindade Guerreiro, Eliana Maria Andriani Guerreiro, Tathiana Mikamura Barchi, Juliana Biluca, Thiago Antonini Alves, Yara de Souza Tadano, Flávio Trojan, Hugo Valadares Siqueira, „Anomaly Detection in Automotive Industry Using Clustering Methods—A Case Study, “ *Appl. Sci. 2021*, 11, 9868. https://doi.org/10.3390/app11219868, 22 October 2021

[19] Danial Hooshyar, Margus Pedaste, Yuen-Min Huang, „Clustering Algorithms in an Educational Context: An Automatic Comparative Approach, “ *IEEE Access*, Volume: 8, published: 7 Avgust 2020, doi: 10.1109/ACCESS.2020.3014948

[20] Aziz Mahboub, Mounir Arioua, Hatim Anas, „Performance Evaluation of Cluster Validity Methods an Energy Optimization in Wireless Sensor Networks Using Hybrid *K*-Medoids Algorithm, “ *BDCA'17: Proceedings of the*

2nd international Conference on Big Data, Cloud and Applications, March 2017 Article No.: 68, Pages 1–7, https://doi.org/10.1145/3090354.3090424

[21] R. J. G. B. Campello, E. R. Hruschka, „A fuzzy extension of the silhouette width criterion for cluster analysis," *Fuzzy Sets and System 157 (2006) 2858-2875*

[22] M. Rawashdeh, A. Ralescu, „Fuzzy Cluster Validity with Generalized Silhouettes," *Published in MAICS 2012 – Computer Science*, 2012

[23] Chun sheng Li, „The improved Partition Coefficient," *2011 International Conference on Advances in Engineering, Procedia* Engineering 24 (2011) 534-583

[24] Kalpit G. Soni, Dr. Atul Patel, „Comparative Analysis of K-means and K-medoids Algorithm on IRIS Data," *International Journal of Computational Intelligence Research*, ISSN 0973-1872 Volume 13, Number 5 (2017), pp. 899-906

[25] Dr. Aishwarya Batra, "Analysis and Approach: K-Means and K-Medoids Data Mining Algorithms," *5th IEEE International Conference on Advanced Computing & Communication Technologies* [ICACCT-2011] ISBN 81-87885-03-3, 2011.

[26] Nurhayati, Nadika Sigit Sinatrya, Luh Kesuma Wardhani, Busman, "Analysis of K-Means and K-Medoids's Performance Using Big Data Technology," *The 6th International Conference on Cyber and IT Service Management (CITSM 2018)*, August 7-9, 2018

[27] Santosh Nirmal, „Comparative Study between K-Means and K-Medoids Clustering Algorithms," *International Research Journal of Engineering and Tehnology (IRJET)*, Volume: 06 Issue: 03, Mart 2019

[28] Elma Hot, Vesna Popović – Bugarin, "Soil data clustering by using K-means and fuzzy K-means algorithm," *Telfor Journal,* Vol. 8, No. 1, 2016.