# Feature selection results for big\_dataset and Nmin = 15

1. result: **(signal\_strength, peak\_frequency, max\_frequency)**

**Estimated Epsilon: 0.0079026, Picked Epsilon: 0.037903, MinNumPoints: 15**

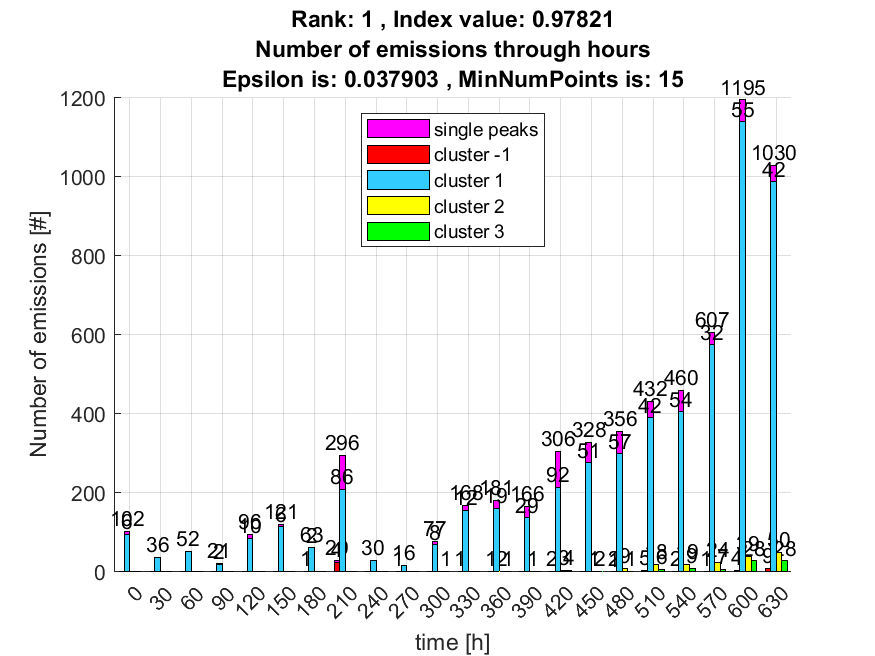
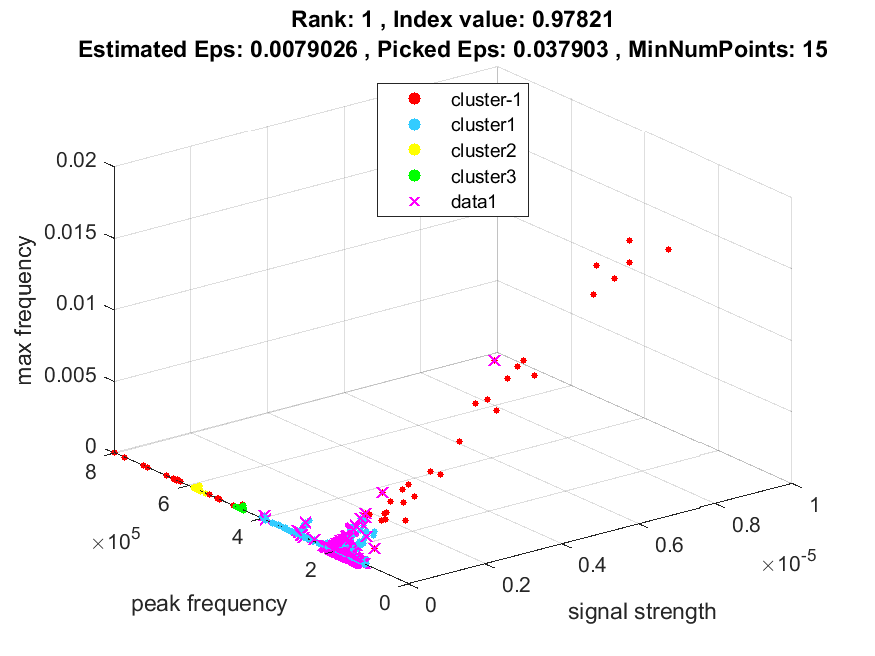
**DBCV index value: 0.97821**

**cluster -1 had 0.81967% single peaks**

**cluster 1 had 99.1803% single peaks**

**cluster 2 had 0% single peaks**

**cluster 3 had 0% single peaks**



**2. result: (peak\_amplitude, peak\_frequency, max\_frequency)**

**Estimated Epsilon: 0.010077, Picked Epsilon: 0.030077, MinNumPoints: 15**

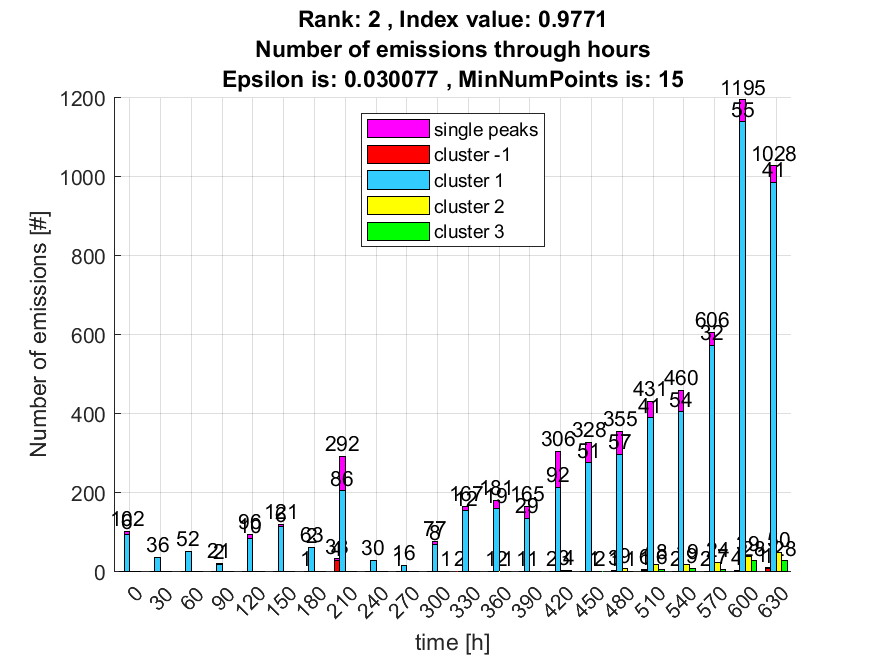
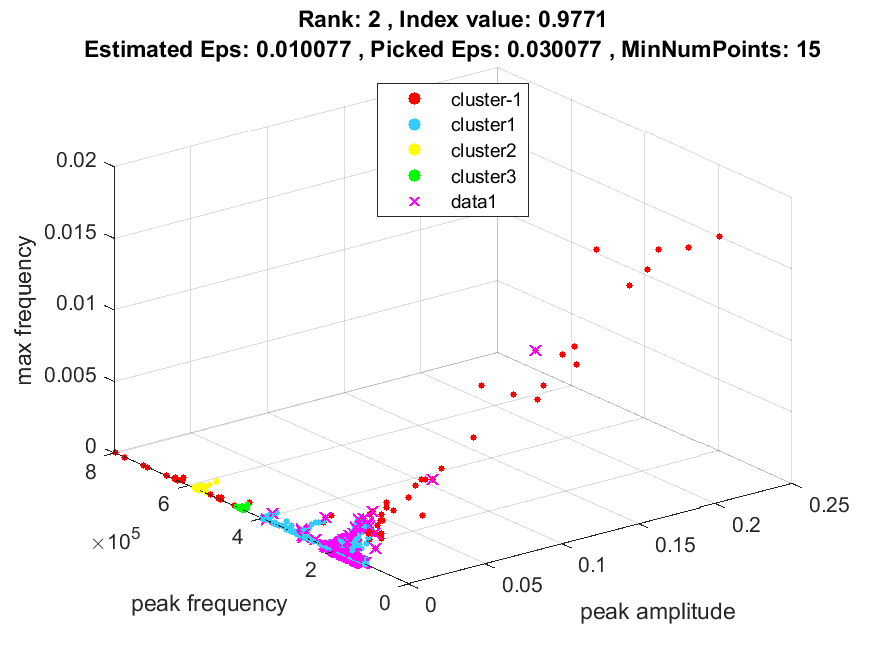
**DBCV index value: 0.9771**

**cluster -1 had 1.1475% single peaks**

**cluster 1 had 98.8525% single peaks**

**cluster 2 had 0% single peaks**

**cluster 3 had 0% single peaks**



**3. result: (peak\_amplitude, pp4, peak\_frequency)**

**Estimated Epsilon: 0.024553, Picked Epsilon: 0.044553, MinNumPoints: 15**

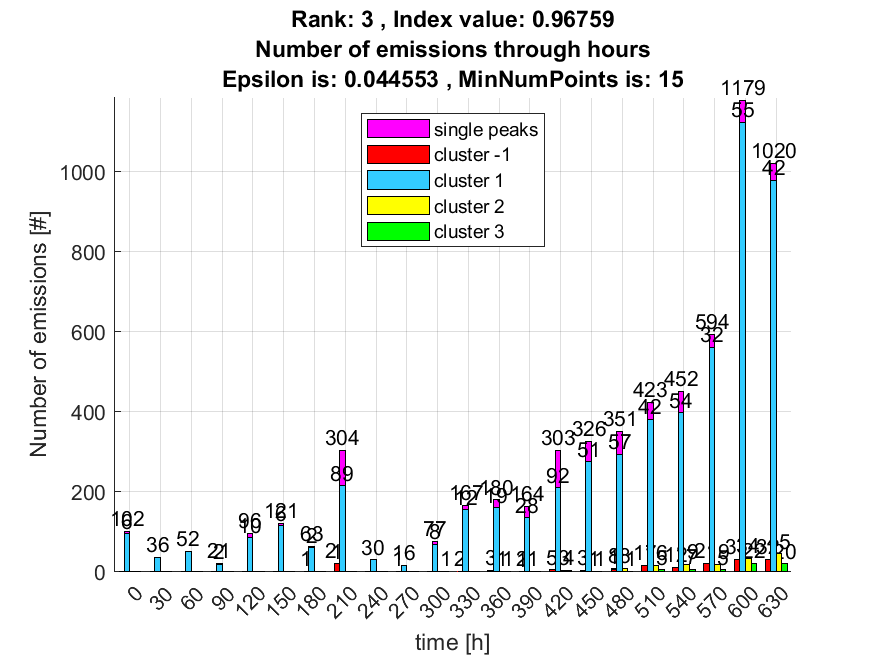
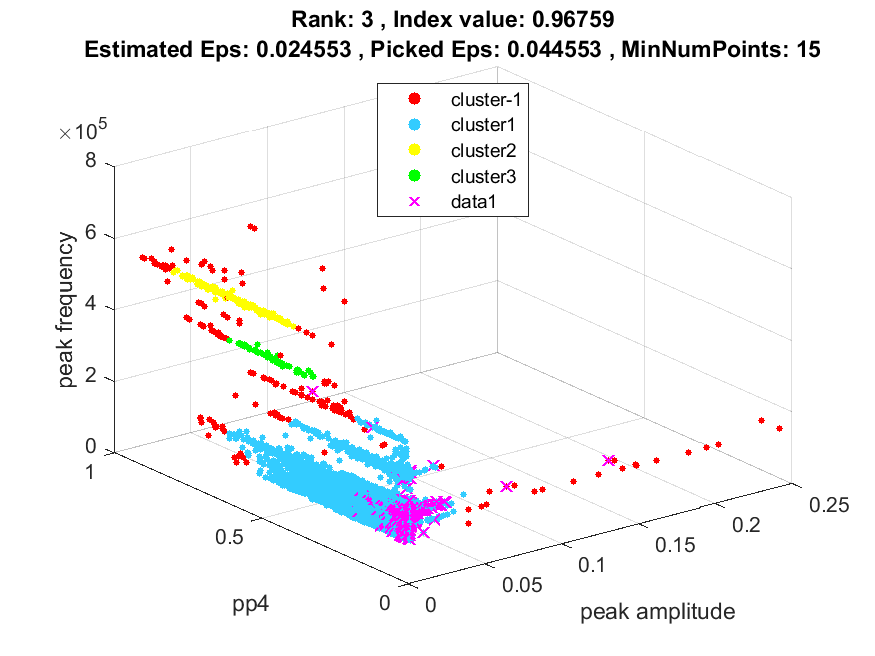
**DBCV index value: 0.96759**

**cluster -1 had 0.4918% single peaks**

**cluster 1 had 99.5082% single peaks**

**cluster 2 had 0% single peaks**

**cluster 3 had 0% single peaks**



**4. result: (signal\_strength, peak\_frequency, total\_counts)**

**Estimated Epsilon: 0.025843, Picked Epsilon: 0.045843, MinNumPoints: 15**

**DBCV index value: 0.96564**

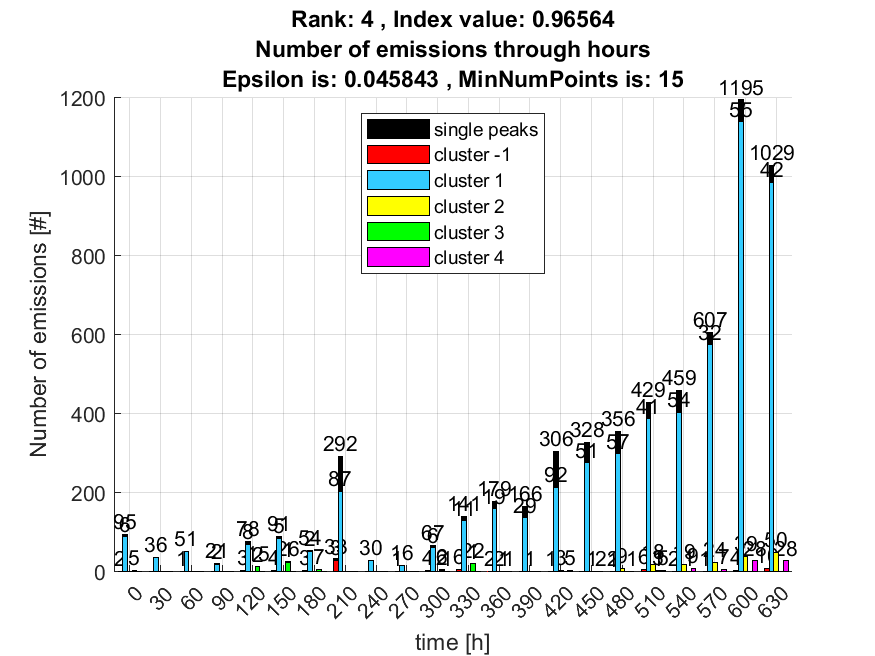
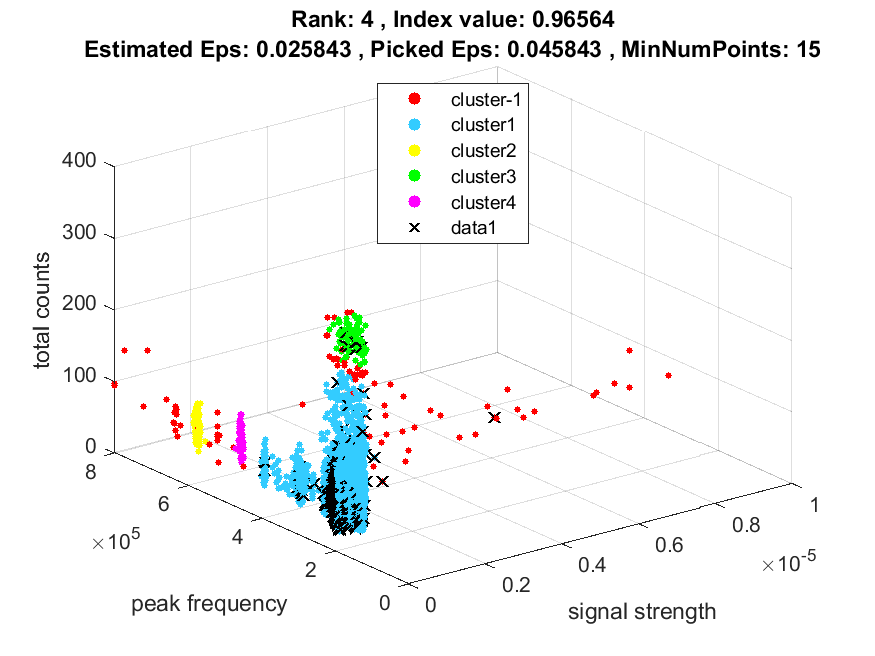
**cluster -1 had 0.65574% single peaks**

**cluster 1 had 98.1967% single peaks**

**cluster 2 had 0% single peaks**

**cluster 3 had 1.1475% single peaks**

**cluster 4 had 0% single peaks**



**5. result: (peak\_amplitude, pp1, peak\_frequency)**

**Estimated Epsilon: 0.016395, Picked Epsilon: 0.046395, MinNumPoints: 15**

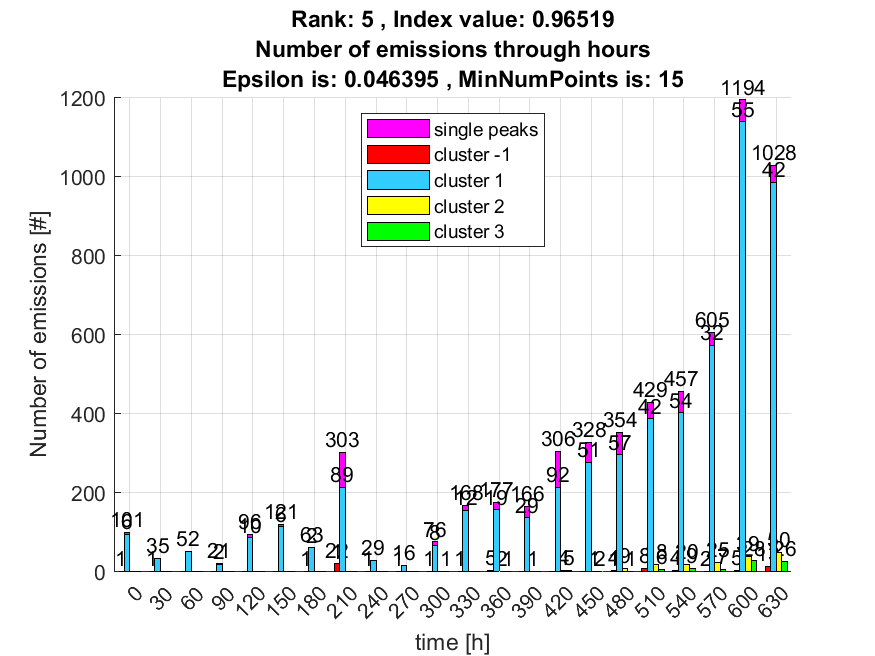
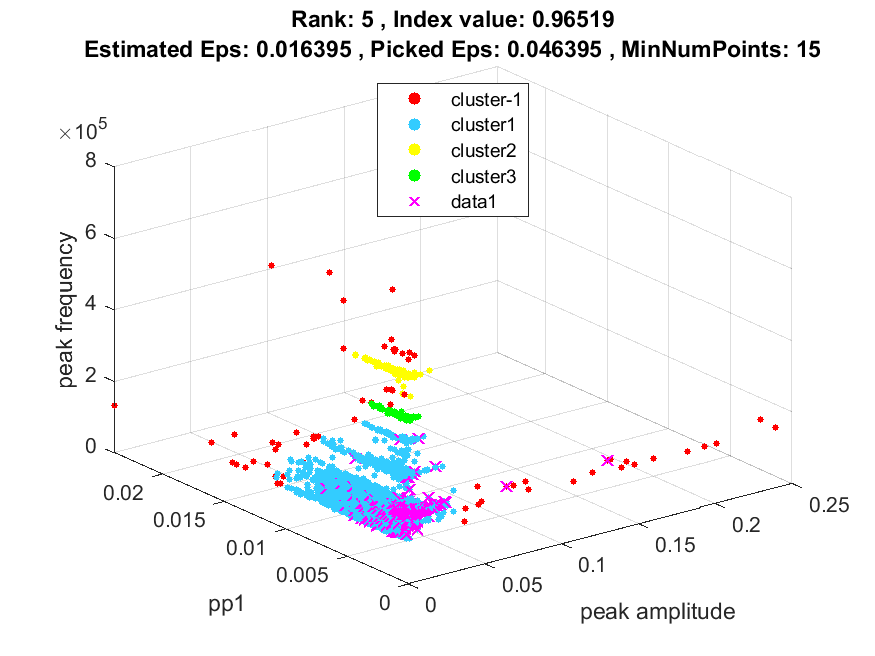
**DBCV index value: 0.96519**

**cluster -1 had 0.32787% single peaks**

**cluster 1 had 99.6721% single peaks**

**cluster 2 had 0% single peaks**

**cluster 3 had 0% single peaks**



**6. result: (peak\_frequency, max\_frequency, total\_counts)**

**Estimated Epsilon: 0.024849, Picked Epsilon: 0.044849, MinNumPoints: 15**

**DBCV index value: 0.96518**

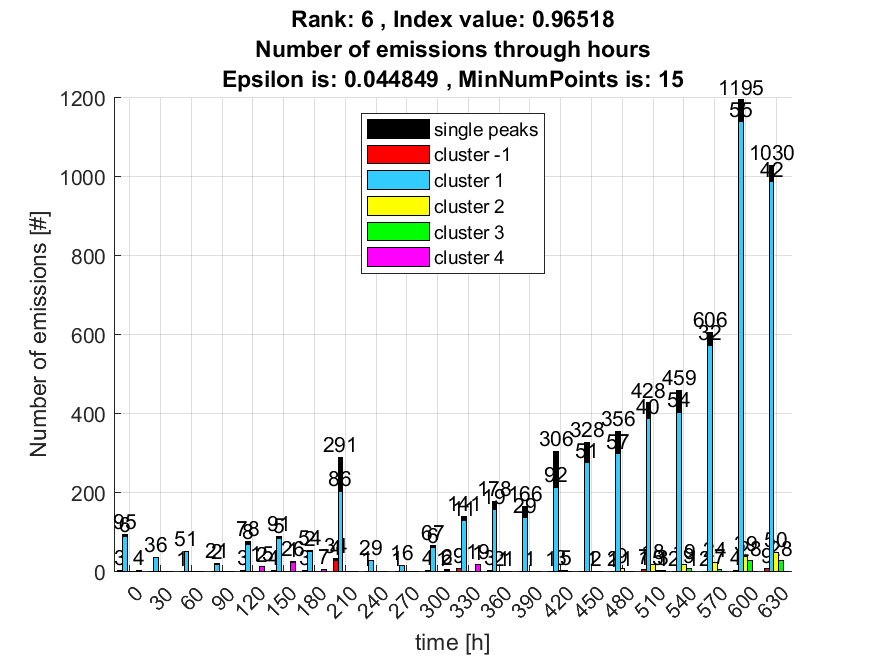
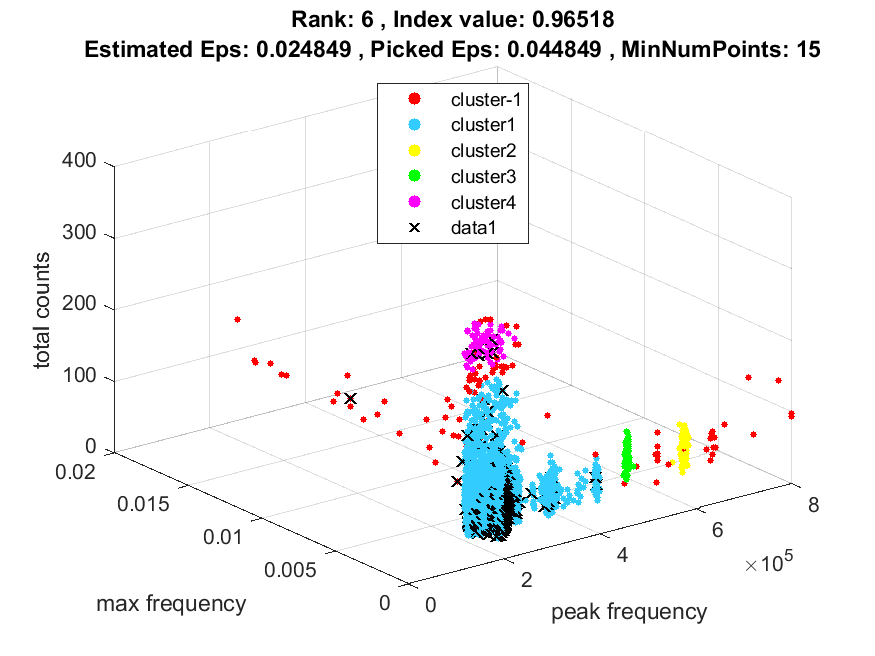
**cluster -1 had 0.98361% single peaks**

**cluster 1 had 97.8689% single peaks**

**cluster 2 had 0% single peaks**

**cluster 3 had 0% single peaks**

**cluster 4 had 1.1475% single peaks**



**7. result: (peak\_amplitude, peak\_frequency, fall\_time)**

**Estimated Epsilon: 0.019445, Picked Epsilon: 0.049445, MinNumPoints: 15**

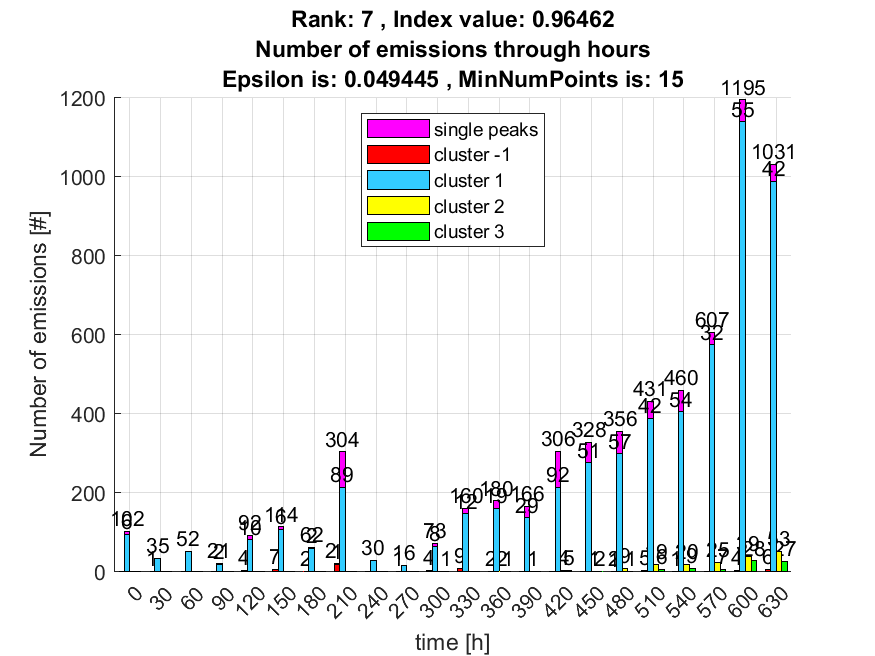
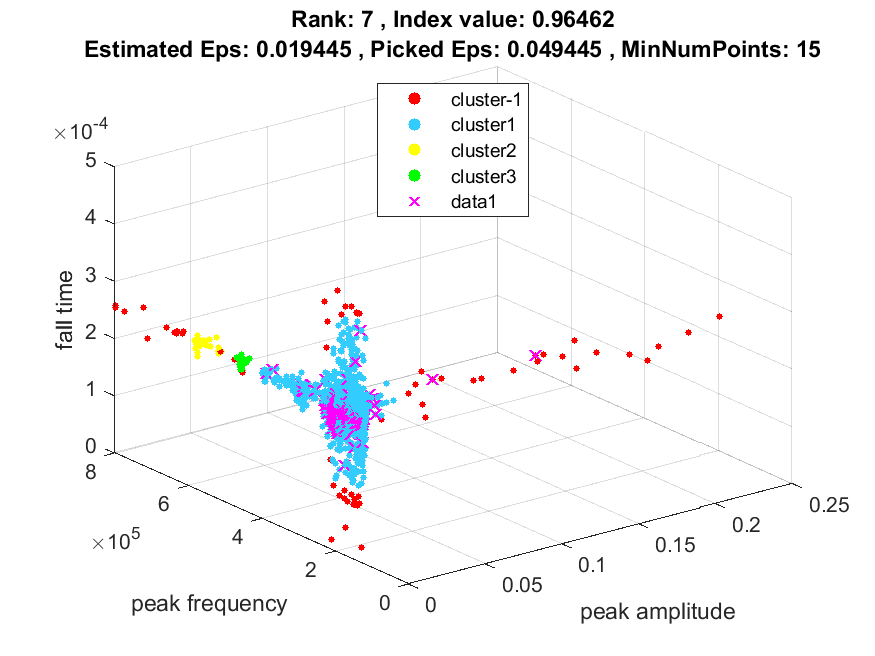
**DBCV index value: 0.96462**

**cluster -1 had 0.32787% single peaks**

**cluster 1 had 99.6721% single peaks**

**cluster 2 had 0% single peaks**

**cluster 3 had 0% single peaks**



**8. result: (peak\_amplitude, peak\_frequency, total\_counts)**

**Estimated Epsilon: 0.026485, Picked Epsilon: 0.056485, MinNumPoints: 15**

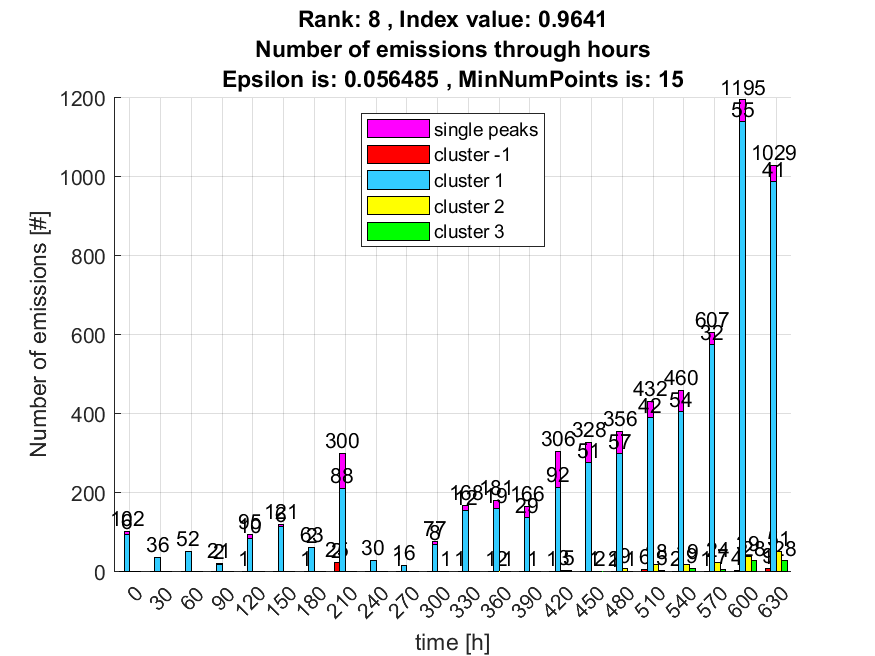
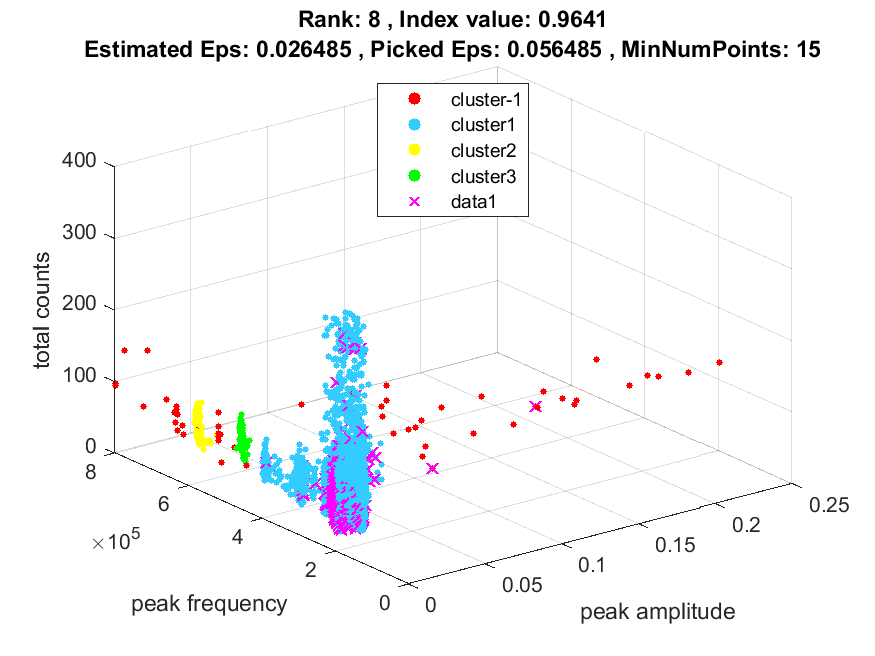
**DBCV index value: 0.9641**

**cluster -1 had 0.65574% single peaks**

**cluster 1 had 99.3443% single peaks**

**cluster 2 had 0% single peaks**

**cluster 3 had 0% single peaks**



**9. result: (peak\_frequency, max\_frequency, fall\_time)**

**Estimated Epsilon: 0.018014, Picked Epsilon: 0.048014, MinNumPoints: 15**

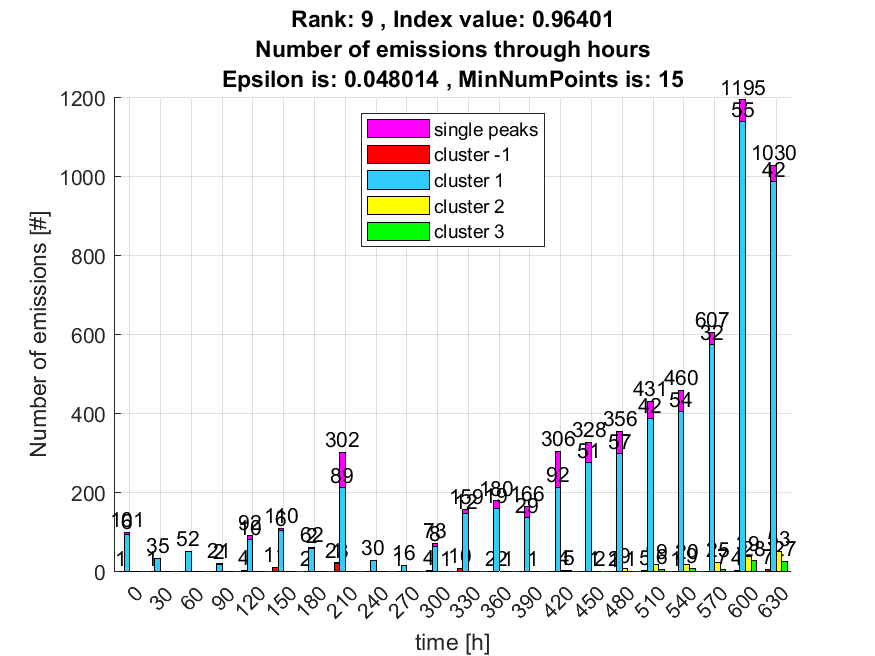
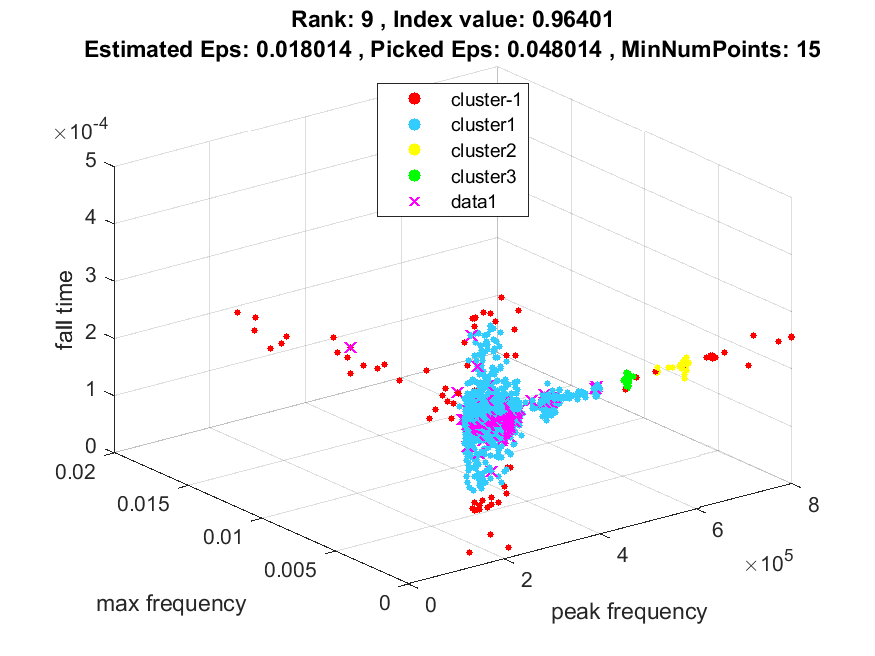
**DBCV index value: 0.96401**

**cluster -1 had 0.32787% single peaks**

**cluster 1 had 99.6721% single peaks**

**cluster 2 had 0% single peaks**

**cluster 3 had 0% single peaks**



**10. result: (pp1, peak\_frequency, max\_frequency)**

**Estimated Epsilon: 0.015981, Picked Epsilon: 0.045981, MinNumPoints: 15**

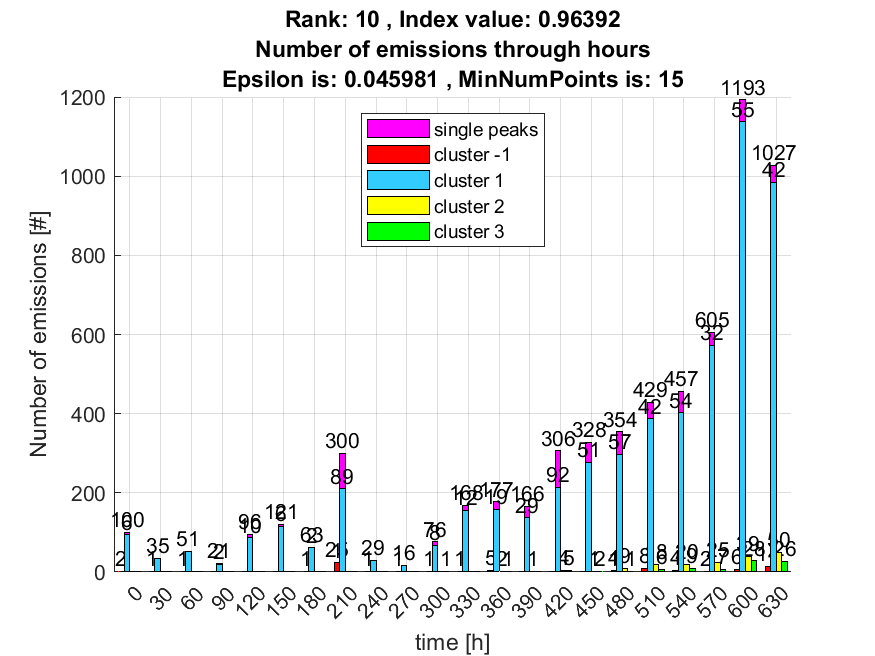
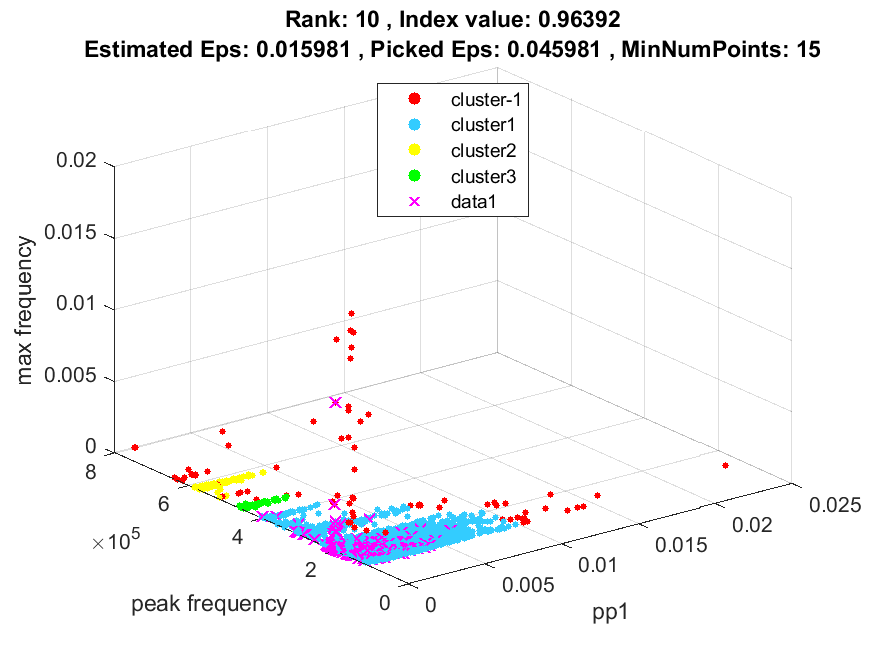
**DBCV index value: 0.96392**

**cluster -1 had 0.32787% single peaks**

**cluster 1 had 99.6721% single peaks**

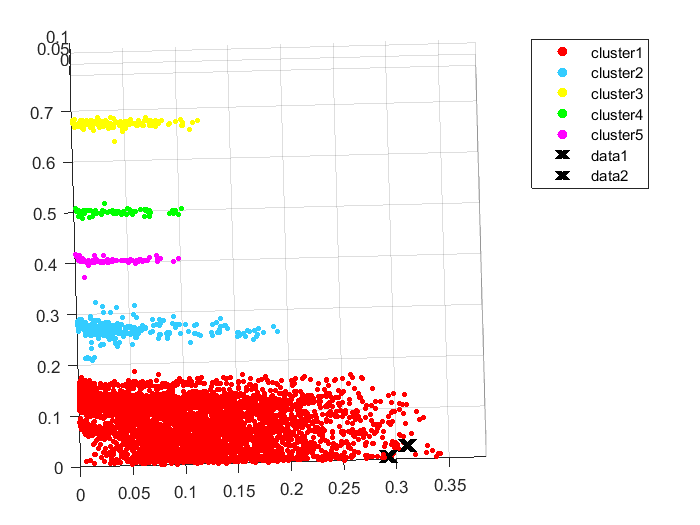
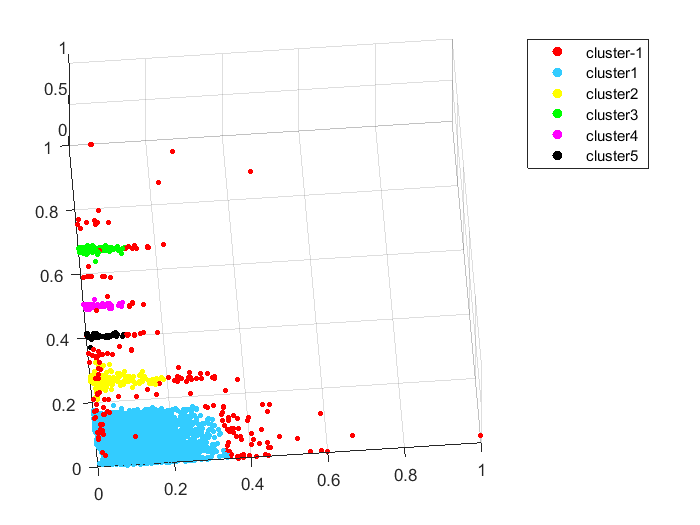
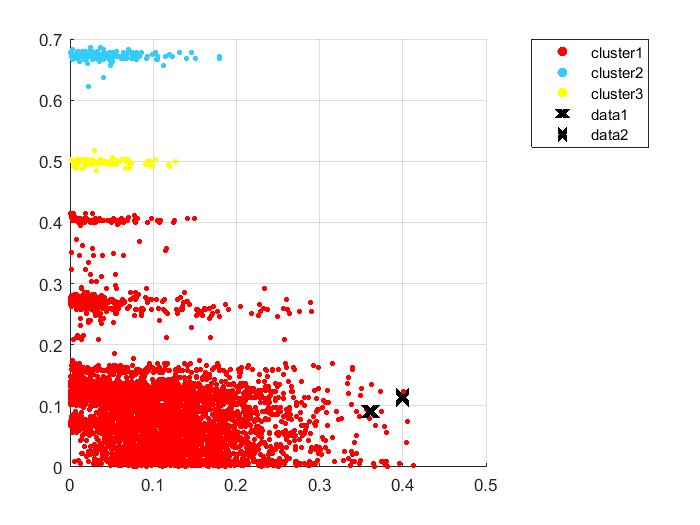
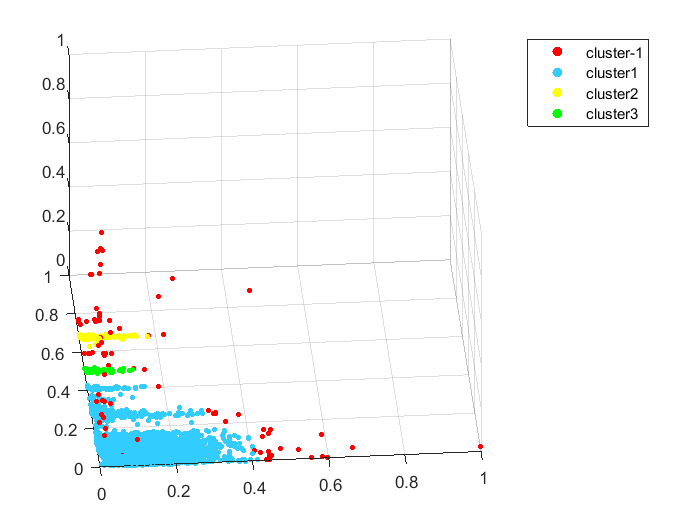
**cluster 2 had 0% single peaks**

**cluster 3 had 0% single peaks**



**Problem odabira clustering rezultata koji uključuje ogromni cluster koji sadrži više vizualno uočljivih manjih clustera. Problem nastaje jer DBCV indeks se bazira na gustoći pa kako su granice između linija clustera dovoljno gusto tako DBCV procjenjuje da je izabrani cluster kvalitetan. Na ovaj problem nisam naišao prije kako kod manjih clustera točke između clustera nisu toliko gusto raspoređene. Iako navedeni podskupovi značajki jesu zadovoljavajući jer izabiru najgušće rezultate, njihovo clusteriranje nije te zbog toga je potrebno probati izmijeniti DBCV indeks da uočava rupe unutar clustera. Naravno zadane izmjene će promijeniti raspon DBCV indeksa.**

**Indeks određuje gustoću unutar clustera kao maksimalnu udaljenost između čvorova minimalno razapinjajućeg stabla (minimum spanning tree-a) ili MST clustera (teoretski najmanja gustoća unutar clustera). To nije zapravo gustoća na kojoj se temelji OPTICS međutim je približno dobro. Kod dobrog OPTICS clusteriranja (dobri ulazni parametri) našeg skupa podataka točke s najmanjom gustoćom su one na rubovima clustera, dok u sredini su gušće raspoređene točke. Međutim kod lošeg clusteriranja vidimo rupe između grupa točaka unutar istog clustera koje su manje ili slične gustoće kao rubovi clustera. Navedeno znači da će algoritam za cluster s rupama s većim brojem točaka te cluster bez rupa s manjim brojem točaka dati istu vrijednost unutarnje gustoće ili DSC-a jer za nju uzima maksimalnu udaljenost između MST-a. Na slici ispod su za naveden clustere označene točke za koje se uzima udaljenost.**



**Na slici ispod možemo vidjeti MST clustera 1 te vidimo usporedbu udaljenosti čvorova MST-a kod rupe između clustera te na rubu clustera. Vidimo da na rubu clustera je procjenjena veća udaljenost pa zaključujemo da uzimanjem maksimalne udaljenosti između čvorova nije dobar način za određivanje rupa ako postoje. Čak iako su rubne točke veće gustoće potrebno je pronaći rupe te kazniti indeks ako postoje.**

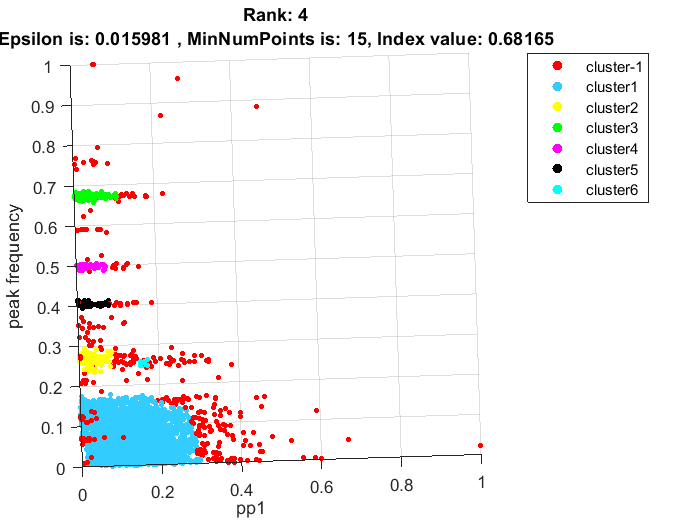
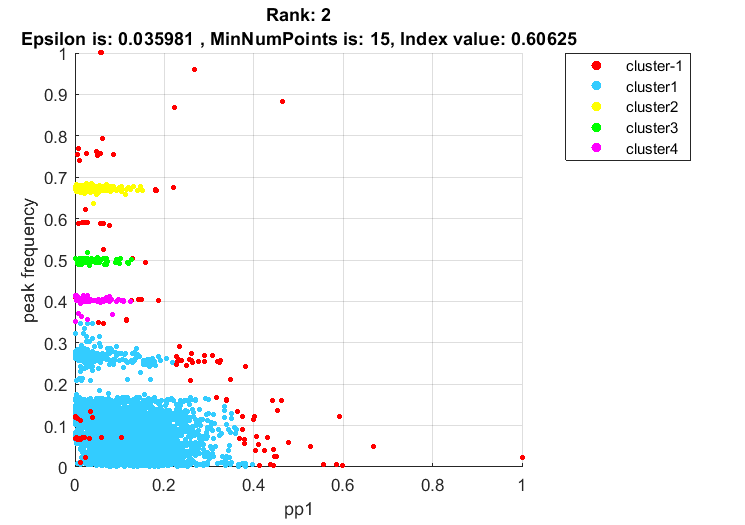
**Zbog toga sam promjenio određivanje DSC-a (gustoće unutar clustera) da se temelji na najmanjoj gustoći čvora našeg MST-a koja se računa kao srednja udaljenost čvora do 15 najbližih točaka. Uzeo sam 15 kako clusteri nikada neće biti manji od 15 jer mi je to minimalna vrijednost MinNPoints parametra. Isto tako kako bi algoritam mogao pronaći rupe iako nisu najmanje gustoće potrebno je pretražiti prvih 10 čvorova MST-a s najmanjom gustoćom te odrediti jesu li rupe ili točke na granici clustera. Točka će biti definirana kao rupa ako dijele stablo na više stabla koje imaju svaki više od 15 čvorova (15 kao minimalna veličina clustera). Na slici ispod lijevo možemo vidjeti kako čvor u rupi dijeli MST-a na više većih stabla pa je rupa, dok čvor izvan na granici dijeli MST-a na više manjih stabla od 15 čvorova i jednog veliko (slika ispod desno).**

**Ovakvo određivanje rupa je primjenjivo samo kada skup podataka sadrži clustere gušće u sredini nego na rubovima što je slučaj kod našeg skupa podataka. Kod manjih clustera 10 pretraživanja rupa jer će se pronaći „rupa“ na temelju manjeg broja čvora za izbor. Pa se u algoritam dodaje da se pretraživanje rupa obustavlja kada su pronađene dvije uzastopne granične točke (ne „rupe“) za clustere čijih 3% točaka je manje od 15 (imaju manje od 500 točaka).**

**Kažnjavanje postojanja rupa unutar clustera odraditi će se množenjem gustoće čvora MST-a s najmanjom gustoćom s faktorom kažnjavanja. Faktor kažnjavanja određen je odnosima veličina podstabla na koje se razdvaja MST izbacivanjem čvora rupe iz njega. Želimo da što je najveće podstablo veće ili približno ukupnom broju čvorova da je kazna nepostojeća ili minimalna, dok što je manje najveće podstablo , što znači da su ostala podstabla veća, da je kazna eksponencijalno veća. To postižemo množenjem DSC-a clustera s inverzom postotka čvorova najvećeg podstabla od ukupnog broja čvorova. Eksponencijalnost kazne postiže se tako da još za svakih 5% što je manji postotak veličine najvećeg podstabla naši faktor kažnjavanja se množi s 2. Testiranjem navedene kazne dobivamo dobre rezultate prikazane slikama ispod.**

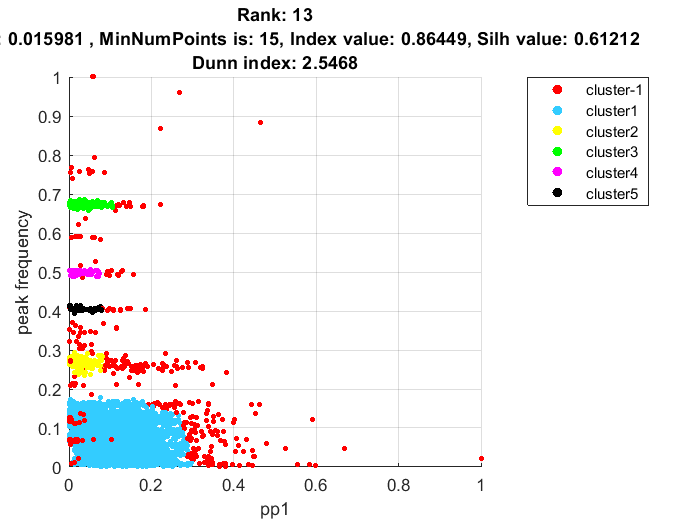
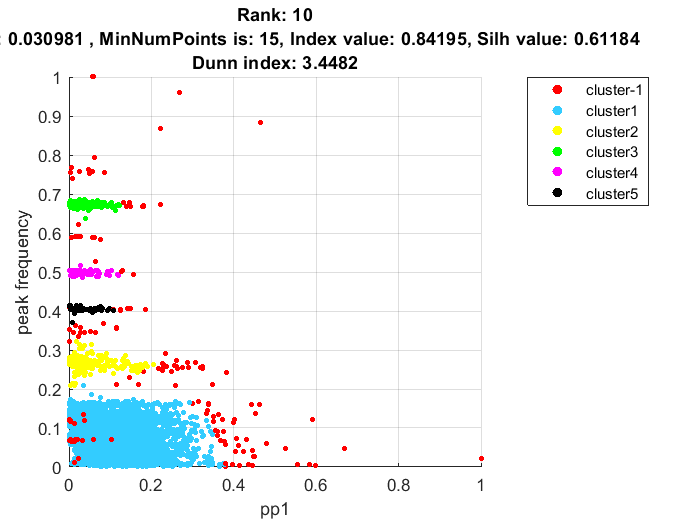
**Još je potrebno promijeniti za clustere bez rupe da prava vrijednost gutoće ne ovisi o nekolicini outliera ako su oni čvorove najmanje gustoće kao što vidimo na slici ispod. Zbog toga se za clustere bez rupe gustoća uzima kao srednja vrijednost 15 najmanje gušćih točaka clustera.**

**Istu stvar možemo napraviti za mjeru gustoće između clustera gdje će se umjesto minimalne udaljenosti čvorava MST-a dvaju clustera uzimati srednja vrijednost prvih 15 najbližih čvorova dvaju clustera kao oporavak u slučaju skupine outliera. Vidimo na slici ispod da bih uzimanje udaljenost najbližih točaka bilo krivo jer su outlieri.**

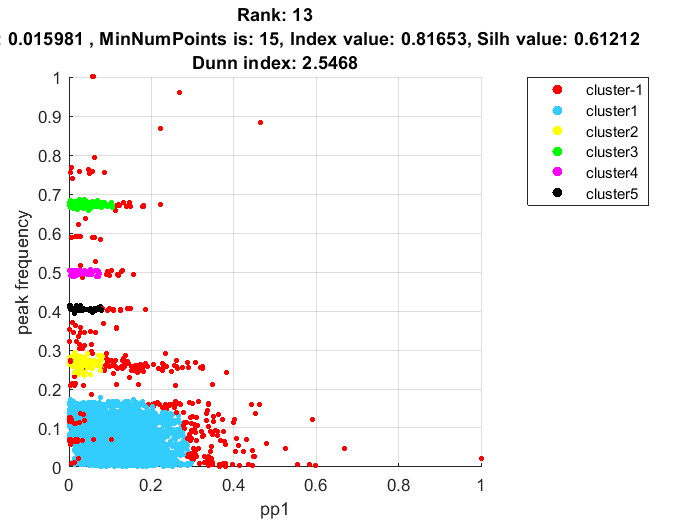
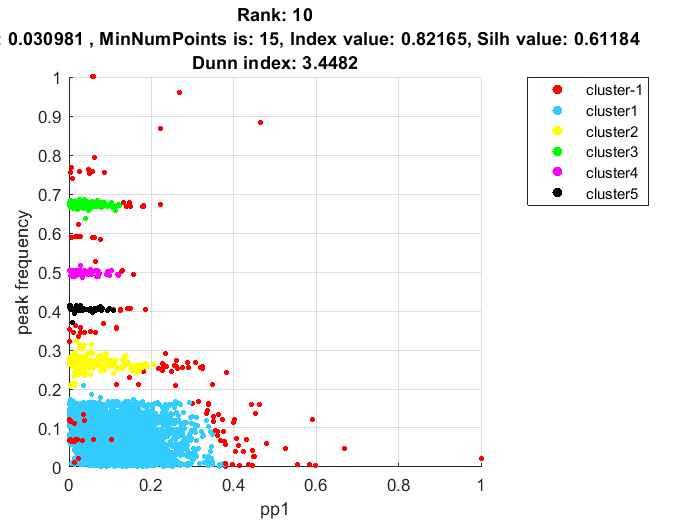


**Problem manjeg clustera usred većeg , ali kako je mali ima jako mali utjecaj.**

**Problem izbora manjih no gušćih clustera, dok mi želimo da bude što više kvalitetnih točaka unutar clustera rješavamo tako da pomnožimo DBCV indeks s postotkom validnih podataka (ne outliera) unutar skupa podataka te ako je više validnih podataka i manje outliera onda će indeks biti veći. Na slici ispod vidimo da indeks preferira manje gušće clustera.**



**Na slici ispod vidimo ispravljeni indeks koji preferira veće clustere s manje outliera u ukupnom skupu podataka.**



**Primjer dobrog kažnjavanja clustera:**

