

# Documentație Tehnică: Strategii de Detectie CFAR și Clustering

Analiză comparativă: CA, OS, SOCA, Agglomerative și HDBSCAN

2026

## 1 Algoritmi de detectie CFAR

### 1.1 Principii fundamentale și concepte comune

Deși arhitecturile specifice variază, toți algoritmii de tip CFAR (Constant False Alarm Rate) îurmărează principii comune, menite să mențină rata alarmelor false la un nivel constant în condițiile unui zgomot de fond (clutter) variabil:

- **Procesarea prin Fereastră Glisantă:** Algoritmii analizează spectrograma sau semnalul temporal utilizând o fereastră mobilă compusă din trei zone distințe:
  1. **CUT (Cell Under Test):** Celula centrală evaluată pentru prezența unei eventuale ținte.
  2. **Zona de gardă (Guard Cells):** Celule adiacente CUT-ului, lăsate neprocesate pentru a evita "scurgerea" energiei țintei în statistica zgomotului, fenomen ce ar putea duce la auto-mascare.
  3. **Celulele de antrenament (Training Cells):** Celule periferice utilizate pentru eșantionarea și estimarea statistică a puterii zgomotului local ( $Z$ ).
- **Pragul adaptiv:** Spre deosebire de detectoarele cu prag fix, pragul CFAR ( $T$ ) se recalculează dinamic pentru fiecare poziție a ferestrei:

$$T = \alpha \cdot Z$$

unde  $Z$  reprezintă puterea estimată a zgomotului, iar  $\alpha$  (sau  $R$ ) este un factor de scalare dependent de probabilitatea de alarmă falsă ( $P_{fa}$ ) dorită și de numărul de celule de antrenament.

- **Condiția de detectie:** Dacă puterea semnalului în CUT depășește pragul calculat ( $CUT \geq T$ ), celula este marcată ca detectie (țintă), în caz contrar fiind considerată zgomot.

### 1.2 Descrierea algoritmilor

- **CA-CFAR (Cell Averaging):** Estează nivelul de zgomot prin media aritmetică a celulelor de antrenament. Este cel mai eficient în zgomot alb omogen, dar suferă de fenomenul de "mascare" în prezență țintelor multiple (de exemplu, două avioane foarte apropiate sunt procesate simultan, având loc o singură detectie).
- **OS-CFAR (Ordered Statistic):** Utilizează o percentilă în locul mediei. Este robust la interferențe și țintă apropiate, ignorând valorile extreme.
- **SOCA-CFAR (Smallest Of Cell Averaging):** Selectează minimul dintre mediile subregiunilor fereștri. Este optim pentru separarea țintelor de lângă marginile zonelor cu clutter dens.

### 1.3 CFAR: Formule și pseudocod

## 2 Analiză detaliată a algoritmilor de clustering

### 2.1 Observații despre algoritmii de clustering

- **Agglomerative:** Lucrează prin proximitate directă. Nu elimină zgomotul, ceea ce poate duce la gruparea eronată a detectiilor false CFAR în clustere valide.
- **HDBSCAN:** Utilizează *Mutual Reachability Distance* pentru a penaliza punctele din zonele cu densitate mică. Acest lucru elimină fenomenul de *chaining* (unirea eronată a două ținte prin puncte de zgomot intermediare).

Metodă	Model matematic	Pseudocod simplificat
CA-CFAR	$Z = \frac{1}{N_T} \sum_{i=1}^{N_T} x_i, T = R \cdot Z$	<ol style="list-style-type: none"> <li>Extrage celulele de antrenament <math>S_{train}</math> (fără zona de gardă)</li> <li><math>Z = \text{mean}(S_{train})</math></li> <li>IF <math>CUT \geq T</math> THEN detectie</li> </ol>
OS-CFAR	$Z = x_{(k)}, T = \alpha \cdot Z$	<ol style="list-style-type: none"> <li>Sortează <math>S_{train} \rightarrow S_{sorted}</math></li> <li><math>Z = S_{sorted}[k]</math> (valoarea de rang k)</li> <li>IF <math>CUT \geq T</math> THEN detectie</li> </ol>
SOCA-CFAR	$Z = \min(\mu_1, \dots, \mu_4), T = R \cdot Z$	<ol style="list-style-type: none"> <li>Împarte fereastra în 4 subregiuni</li> <li>Calculează media <math>\mu_i</math> pentru fiecare subregiune</li> <li><math>Z = \min(\mu_1, \dots, \mu_4)</math></li> <li>IF <math>CUT \geq T</math> THEN detectie</li> </ol>

Tabela 1: Formule și pseudocod pentru metodele CFAR

Metodă	Robustețe clutter	Ținte slabe	Complexitate	Ideal pentru
CA	Scăzută	Ridicată	Foarte mică	Spectrogramă omogenă
OS	Foarte bună	Medie	Mare	Medii multi-target
SOCA	Bună	Medie	Medie	Margini de clutter

Tabela 2: Comparație între metodele CFAR

Metodă	Descriere conceptuala	Pseudocod
Agglomerative	Ierarhic <i>bottom-up</i> . Fuzionează succesiv perchiile cele mai apropiate.	<ol style="list-style-type: none"> <li>Normalizează punctele <math>(f, t)</math></li> <li>Fiecare punct = cluster individual</li> <li>WHILE nr_clustere <math>&gt; K</math>: Găsește clusterele cu <math>d_{min}</math> Unește-le sub aceeași etichetă</li> <li>Reindexează etichetele <math>(0, 1, 2\dots)</math></li> </ol>
HDBSCAN	Bazat pe densitate ierarhică. Folosește <i>core distance</i> și <i>MST</i> pentru stabilitate.	<ol style="list-style-type: none"> <li>Normalizează punctele <math>(f, t)</math></li> <li>Calculează <b>core distance</b> (<math>c</math>) pentru fiecare punct</li> <li>Determină <b>MRD</b>: <math>d_{mrnd}(u, v) = \max(c(u), c(v), d(u, v))</math></li> <li>Construiește <b>MST</b> folosind MRD</li> <li>Extrage componente conexe cu prag <math>eps</math></li> <li>Elimină clusterele sub <math>min\_samples \rightarrow</math> zgromot (-1)</li> </ol>

Tabela 3: Sinteză implementării metodelor de clustering

### 3 Separare prin triangulare Delaunay

### 4 Extinderea experimentului pe semnalul sintetic din lucrarea de referință

#### 4.1 Scopul experimentului

Acest experiment urmărește evaluarea performanței metodelor de detecție și separare într-un scenariu **controlat**, similar celui utilizat în lucrarea de referință.

Metodă	Idee	Pseudocod simplificat
Detectare Vârfuri	Maxime locale în spectrograma STFT, filtrate prin prag percentilă: $\text{peak}(i, j) = \begin{cases} 1 & \text{dacă } S_{i,j} = \max(\text{vecini}) \text{ și } S_{i,j} > T \\ 0 & \text{altfel} \end{cases}$	<ol style="list-style-type: none"> <li>1. Calcul STFT și magnitudine</li> <li>2. Threshold = magnitudinea percentilei</li> <li>3. Identifică maxime locale în fereastra dată</li> <li>4. Returnează coordonatele vârfurilor</li> </ol>
Triangulare Delaunay	Construiește triunghiuri pentru punctele detectate astfel încât niciun punct să nu fie în cercul circumscris al unui triunghi existent	<ol style="list-style-type: none"> <li>1. Creează un triunghi care înconjoară toate punctele</li> <li>2. Pentru fiecare punct nou introdus, se identifică triunghiurile care încalcă criteriul Delaunay (punctul se află în interiorul cercului circumscris acestora)</li> <li>3. Elimină triunghiurile găsite și formează poligonul gol</li> <li>4. Triunghiuri noi = combină punctul curent cu fiecare latură a poligonului</li> <li>5. Elimină triunghiurile care conțin vârfuri supertriunghi</li> </ol>
Gruparea triunghiurilor	Triunghiurile vecine sunt conectate în funcție de energia lor medie, dacă au o muchie comună	<ol style="list-style-type: none"> <li>1. Construiește lista muchiilor pentru toate triunghiurile</li> <li>2. Adaugă muchiile comune ca legături dacă diferența relativă de magnitudine <math>&lt; \varepsilon</math></li> <li>3. Parcuregere DFS pentru a identifica componentele conexe</li> <li>4. Calculează puncte, energie, centroid pentru fiecare componentă</li> </ol>

Tabela 4: Sinteză a pașilor de separare prin triangulare Delaunay și pseudocod asociat

Semnalul de test este sintetic, cu structură bine definită în timp și frecvență, iar zgomotul este adăugat (AWGN) pentru valori SNR cuprinse între 5 și 30 dB.

## 4.2 Arhitecturile testate

### Metoda 1: CA-CFAR + HDBSCAN

- STFT cu fereastră Hamming
- Deteceție CA-CFAR pe spectrogramă
- Clusterizare HDBSCAN a punctelor detectate
- Reconstituire a componentei dominante
- Evaluare prin RQF (Reconstruction Quality Factor)

Această configurație corespunde îndeaproape metodologiei din articolul original.

### Metoda 2: Separare prin Triangulare Delaunay

- STFT cu fereastră Gaussiană
- Detectarea maximelor locale peste prag percentilă
- Triangulare Delaunay în plan timp-frecvență
- Gruparea componentelor după energie
- Estimare Doppler din componenta cu energie maximă

## 4.3 Observații experimentale

- În scenariul sintetic, metoda **CA-CFAR + HDBSCAN** prezintă:

- rate de detecție ridicate
- valori RQF apropriate de cele ale metodei din lucrare
- comportament stabil relativ la SNR
- Acest rezultat confirmă validitatea CFAR atunci când ipoteza de zgomot AWGN este respectată.
- Metoda de **triangulare Delaunay** obține rezultate bune, în special la SNR mediu și ridicat, dar prezintă:
  - o variabilitate mai mare a RQF
  - o sensibilitate crescută la pragul de detecție al vârfurilor

## 5 Observații pe Date IPIX

### 5.1 Setul de Date

Experimentele au fost realizate pe date radar reale din setul **IPIX**, caracterizate prin **clutter marin sever** și variații puternice de energie în timp și frecvență. Au fost analizate două scenarii:

- **Low Sea State** – clutter moderat
- **High Sea State** – clutter intens, neomogen

Semnalele sunt procesate segmentat (1 s), PRF = 1000 Hz.

### 5.2 Metodele comparate

Configurațiile sunt cele descrise în secțiunea anterioară

### 5.3 Rezultate

Metodă	State	Componente	Viteză [m/s]
CA-CFAR + HDBSCAN	HIGH	$1.0 \pm 0.0$	-0.054
	LOW	$1.0 \pm 0.0$	-0.008
Triangulare	HIGH	4.2	1.30
	LOW	13.6	-0.30

Tabela 5: Performanță comparativă pe IPIX (30 segmente  $\times$  1s)

### 5.4 Observații Experimentale

#### CA-CFAR + HDBSCAN:

- Rata de detecție: 100% (detectează în toate segmentele)
- *Problema*: Detectează **clutter**, nu ținte
- Variabilitate zero ( $\sigma = 0$ )  $\rightarrow$  HDBSCAN grupează tot clutter-ul într-un cluster unic
- **Concluzie**: CA nu poate separa ținta de clutter în mediu neomogen

#### Triangulare Delaunay:

- Rate de detecție ridicate, dar *instabile*
- HIGH sea: 4.2 componente, LOW sea: 13.6 componente (ar trebui invers!)
- Viteza Doppler variază de 4× între scenarii (+1.30 vs. -0.30 m/s)
- **Concluzie**: Metoda geometrică fragmentează clutter-ul în componente multe componente, reducându-se consistența detecțiilor

## 5.5 Concluzie validată experimental

Pentru sea clutter real neomogen:

- **CA-CFAR** eșuează la separarea țintă-clutter (detectează doar clutter)
- **Triangularea** oferă detectie geometrică, dar instabilă (fragmentare excesivă)
- **GOCA-CFAR** cu DBSCAN tradițional (metoda paper) oferă performanță optimă