Al-Azhar Engineering 16th International Conference

Vol. 19, No. 72, July 2024, 250 - 269

OPTIMIZING MULTIPLE-TARGET CFAR DETECTION EFFICACY THROUGH ADVANCED INTELLIGENT CLUSTERING ALGORITHMS WITHIN K-DISTRIBUTION SEA CLUTTER ENVIRONMENTS

Mansoor M. Al-dabaa[1*], Ahmed A. Emran[1], Ahmed Yahya[1], Mohamed B. El-Mashade[1], Ashraf Aboshosha[2]

[1]Electrical Engineering Department, Faculty of Engineering, Al-Azhar University, Nasr City, 11884, Cairo, Egypt,

[2]Rad. Eng. Dept, NCRRT, Egyptian Atomic Energy Authority, EAEA, Cairo, Egypt

***Correspondence:** Mansoor.Mohammed622@azhar.edu.eg

Citation:

M.M. Al-dabaa, A.A. Emran, A. Yahya, M.B. El-Mashade and A. Aboshosha. "Optimizing Multiple-Target CFAR Detection Efficacy through Advanced Intelligent Clustering Algorithms within K-Distribution Sea Clutter Environments ", Journal of Al-Azhar University Engineering Sector, vol. 19, pp. 250 - 269, 2024.

Received: 10 December 2023

Revised: 29 January 2024

Accepted: 08 February 2024

DoI:10.21608/auej.2024.255544.1574

ABSTRACT

In K-distribution sea clutter environments, maintaining a constant false alarm rate (CFAR) is essential due to the unpredictable and dynamic nature of the background. However, CFAR detectors often face reduced performance in scenarios with multiple targets due to a masking effect. To combat this issue, a technique known as "Space-Based Linear Density Clustering for Applications with Noise" (Lin-DBSCAN) is employed alongside CFAR. Lin-DBSCAN is adept at pinpointing both interference targets and sea spikes, typically appearing as outliers, in the designated areas before and after the cell under test (CUT). By integrating Lin-DBSCAN, these irregular signals are efficiently identified and segregated from the general sea clutter, significantly improving target detection accuracy. Extensive simulations under various conditions—varying false alarm rates, target numbers, and shape parameters—have shown that Lin-DBSCAN-CFAR outperforms traditional CFAR methods. Additionally, it reduces computational complexity compared to its counterpart, DBSCAN-CFAR. These enhancements significantly boost the practicality and efficiency of CFAR detection in K-distribution sea clutter scenarios, offering a robust solution to the challenges posed by multiple target environments.

KEYWORDS: Constant false alarm rate, Linear Density-Based Spatial Clustering, cell under test, Lin-DBSCAN-CFAR, SO-CFAR.

تحسين فعالية كشف CFAR لأهداف متعددة من خلال خوارزميات التجميع الذكية المتقدمة ضمن بيئات فوضى البحر بتوزيع K

منصور محمد الدبعاء[1]*، احمد عبدالرحمن عمران[1]، احمد يحيى[1]، محمد بكري المشد[1]، اشرف ابو شوشه[2]

[1]قسم الهندسة الكهربية، كلية الهندسة، جامعة الأزهر، مدينة نصر، 11884، القاهرة، مصر
[2]قسم الهندسة الإشعاعية، المركز القومي لبحوث وتكنولوجيا الإشعاع، الهيئة المصرية للطاقة الذرية، القاهرة، مصر
*البريد الاليكتروني للباحث الرئيسي: Mansoor.Mohammed622@azhar.edu.eg

الملخص

في بيئات الفوضى البحرية المعتمدة على توزيع K، يُعد ضمان استقرار معدل الإنذار الخاطئ (CFAR) أمرًا ضروريًا بالنظر إلى التغير المستمر في البيئة المحيطة. تعاني أنظمة كشف CFAR من انخفاض حاد في الأداء عند التعامل مع أهداف متعددة بسبب تأثيرات التمويه. لمواجهة هذه التحديات، نستخدم نهجًا يُعرف بـ "تجميع الكثافة الخطية المستندة إلى الفضاء للتطبيقات في ظروف الضوضاء" (Lin-DBSCAN) بالاقتران مع CFAR. يبرز Lin-DBSCAN في قدرته على تحديد الأهداف المتداخلة والتكوينات البحرية، التي تظهر كنقاط شاذة ضمن المناطق المجاورة للخلية قيد الفحص (CUT). من خلال دمج Lin-DBSCAN، تُعزل الإشارات الشاذة بكفاءة عن الفوضى البحرية الأساسية، مما يعزز بشكل كبير دقة تحديد الأهداف. المحاكاة المفصلة في ظروف متنوعة، تشمل تغييرات في معدلات الإنذار الخاطئ وعدد الأهداف ومعاملات الشكل، تُظهر أن Lin-DBSCAN-CFAR الذي استخدمناه يتفوق بشكل واضح على الطرق التقليدية لـ CFAR. بالإضافة إلى ذلك، يقلل نهجنا المستخدم من التعقيد الحسابي مقارنةً بنظام DBSCAN-CFAR. هذه التحسينات الجوهرية تُحسن من فعالية وجدوى نظام كشف CFAR في بيئات الفوضى البحرية بتوزيع K، مقدمة حلولًا فعالة للتحديات المتعلقة بالأهداف المتعددة.

الكلمات المفتاحية : - معدل الإنذار الخاطئ الثابت، التجميع الفضائي المبني على الكثافة الخطية، الخلية قيد الاختبار، Lin-DBSCAN-CFAR، SO-CFAR.

1-INTRODUCTION

The K distribution has emerged as a highly reliable method for high-resolution simulation and modeling of sea clutter, surpassing traditional statistical distributions like the Rayleigh, Weibull, and lognormal models in effectively capturing the statistical nature of sea clutter [1]. This model assumes that sea clutter amplitude conforms to the Rayleigh distribution (known as the speckle component) at every distance point, while its intensity is influenced by the gamma distribution. The K-distribution's capability to accurately reflect the scattering mechanism of clutter has been extensively verified through practical tests and real-world data analysis [2]. These thorough assessments have proven the model's effectiveness and dependability in representing the intricate features of clutter, thereby confirming its valuable application in a variety of real-world settings [27, 25].

Radar systems primarily function to identify targets, and Constant False Alarm Rate (CFAR) processors play a pivotal role in enhancing target detection efficacy [5]. These processors, known for their proficiency in maintaining a steady rate of false alarms, significantly boost detection probabilities.

Numerous studies have explored various CFAR processors that utilize the sliding reference window approach. These processors are instrumental in automating target detection through statistical analysis of the surrounding clutter. Leveraging insights from extensive research [9], [20], this method calculates dynamic thresholds compared against values from the Cell Under Test

(CUT). This process is critical for the efficient differentiation and recognition of targets in a variety of environments, as corroborated by multiple studies [10, 21].

To effectively address the challenge of detecting multiple targets amidst complex backgrounds such as K-distribution marine clutter, where the phenomenon of target masking is pronounced, a nuanced approach to background noise level assessment is essential. Traditional mean-level processors like Smallest-Of CFAR (SO-CFAR) [22], Average Cell CFAR (CA-CFAR) [12], and Greatest-Of CFAR (GO-CFAR) [13] may not deliver optimal detection performance in such scenarios. The presence of interfering targets within the reference window often leads to an increased likelihood of missing targets [11].

To mitigate this issue, classification-based processors, specifically the CFAR Ordered Statistics (OS-CFAR), have been developed. OS-CFAR operates by sorting the sampled values in the reference window and selecting a reference cell based on a predefined range [22]. This method excludes a fraction of high amplitude reference cells to better represent the average power clutter. In this process, sea spikes and interfering targets within the reference window are treated as outliers. Unlike the mean-level processors (SO, GO, and CA-CFAR), sorting-based processors like OS-CFAR demonstrate superior performance in multi-target situations. However, their effectiveness relies on prior knowledge regarding the distribution and number of interference targets. This refined approach to target detection ensures a more accurate assessment of the background noise level, crucial for environments with complex interference patterns [20].

Environments with complex interference patterns [20]. Machine learning, a field that intertwines various disciplines, has seen remarkable growth in recent years. This surge in interest is largely due to its wide application across different academic areas, particularly through technologies like clustering algorithms, Artificial Neural Networks (ANN), and deep learning [23]. Key among these technologies is the clustering algorithm, which comprises methods like Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Local Outlier Factor (LOF), k-means, and isolation forests. These algorithms play a pivotal role in machine learning, as indicated in several studies [11,14, 24, 7].

One significant application of clustering algorithms is in radar target recognition, especially in environments with non-uniform jamming. They are especially effective in multi-target situations. For instance, a modified version of the CA-CFAR (Constant False Alarm Rate) approach, which integrates the Grubbs criterion, has shown promise in these scenarios, as documented in [28]. However, it’s crucial to recognize the limitations of Grubbs’ criterion. While effective in identifying a single outlier in datasets with a near-normal distribution, it falls short of achieving optimal detection in K-distribution disorders, which is a common challenge in marine environments.

This limitation underscores why only a few studies have successfully applied clustering techniques to address multi-target CFAR detection in marine environments with K-distributed interference [19]. The need for more research in this area is evident, as these techniques hold significant potential in improving detection and analysis in complex and noise-rich environments.

Recent advancements in Synthetic Aperture Radar (SAR) imaging have revolutionized applications ranging from structural monitoring to automotive radar systems, with Ground-Based SAR (GBSAR) and its Circular Scanning (CS) modality leading the way in high-resolution imaging and micro deformation sensing [29, 30]. Innovations such as a two-step CFAR-based 3D point cloud extraction method, integrating maximum projection with a modified Density-Based Spatial Clustering of Applications with Noise (DBSCAN), have significantly improved sidelobe suppression and noise reduction, enhancing point cloud accuracy against LiDAR benchmarks. Concurrently, the development of the F2T-CFAR algorithm, leveraging Fast Fourier Transform (FFT) for ship target detection, and adaptations in DBSCAN for high-resolution data, demonstrate significant strides in computational efficiency and accuracy in target detection and clustering [31, 32].

These technological advancements have not only streamlined the processing of complex radar data but also bolstered the reliability of radar-equipped systems in automotive and urban applications. By refining CFAR and DBSCAN algorithms for high-resolution data, these innovations enable more precise detection, tracking, and clutter management. This evolution underscores the critical role of sophisticated clustering and detection methodologies in overcoming the challenges associated with high-resolution SAR imaging and target detection.

The proposed integration of DBSCAN clustering into a sophisticated CFAR processor signifies a substantial advancement in terms of effectiveness and resilience, contrasting starkly with traditional models [11]. However, this innovation brings with it a degree of complexity. The computational demands of this approach are proportional to the size of the dataset, and efforts to optimize its performance have been confined to an O (n log n) complexity framework. To address these challenges, we are shifting our focus toward the creation of an innovative CFAR processor powered by Lin-DBSCAN clustering. Lin-DBSCAN is specifically designed to overcome the computational challenges of traditional DBSCAN, especially with smaller datasets, while still adhering to the principles of sequential programming. It combines density-based properties with grid-based clustering methods, representing a significant shift from examining individual data points to a more efficient evaluation of grid cells, thus facilitating more streamlined data processing.

In the concluding section of our introduction, we summarize our paper's key contributions, highlighting the innovative aspects and advancements that set our research apart. Our work is distinguished by several notable achievements:

- We introduce a groundbreaking development with the Lin-DBSCAN-CFAR processor, a novel approach that utilizes the Lin-DBSCAN clustering algorithm. This innovation is particularly adept at filtering anomalies around the Cell Under Test (CUT) without the need for prior knowledge of interference targets, showcasing an exceptional ability to adapt to clutter background levels in environments with multiple targets.
- Our research presents Lin-DBSCAN, an algorithm specifically designed for clustering geospatial data, marking a significant advancement in this field. Lin-DBSCAN is engineered for linear time complexity, which drastically reduces computational costs and significantly improves the efficiency of real-time geospatial data applications.
- We demonstrate our dedication to improving clustering techniques with the development of the Lin-DBSCAN-CFAR processor. This processor not only exemplifies our innovative clustering approach but also offers an effective solution tailored to meet the challenges of real-world scenarios, thereby promising enhanced efficiency and broader applicability across diverse practical contexts.

In this paper, we first explore the K-distribution model of sea clutter and its detection using CFAR, followed by a detailed examination of the Lin-DBSCAN-CFAR implementation. Subsequently, we evaluate the Lin- DBSCAN-CFAR processor's performance and results, culminating in a final section that presents key conclusions from the study and suggests avenues for future research.

2. The conceptual framework of the K-distribution Sea Clutter Model and CFRA Detection

2.1 K-distribution Sea Clutter Model

The K-distribution is a sophisticated model known for its exceptional accuracy in representing echoes from various sea surface states. It is particularly effective in analyzing scattering radiation at low backscatter angles [27, 25]. The probability density function (PDF) of this distribution can be mathematically expressed as:

$$f(x) = \frac{-4v}{\Gamma(v)} (cv)^{v} k(v-1)(2cx) \tag{1}$$

In this equation, c is the scaling parameter, representing the average power of sea clutter. The shape parameter is denoted as v, and the range of v indicates the "spikiness" of sea clutter. The term $\Gamma(v)$ is the gamma function, while $k(v-1)$ is the modified Bessel function. This model's precision makes it invaluable for specific maritime applications, especially in the context of sea clutter analysis [25].

2.2-CFAR Detection

The primary objective of CFAR processing is to devise an adaptable threshold. This threshold adeptly differentiates between genuine radar targets and jamming signals. Its adaptability hinges on the predetermined probability of false alarms and prevailing noise levels, aiming for precise target detection while curtailing false alarms due to external disruptions, as outlined in [17].

Following a series of filtering steps, the signal's quadrature and in-phase components are further analyzed using square law processing. The resulting data then undergoes analysis within a sliding window setup. This configuration includes reference cells, the CUT, and guard cells, detailed in [26]. Assume the reference cell samples in the main window are {x1, x2, ..., xn} and in the sliding window are {xn + 1, xn + 2, ..., x2n}, with the reference window length denoted by N = 2n. In this context, the CUT has identical leading and trailing windows. The inherent spatial and temporal correlation within radar echoes enables effective noise level estimation using the reference cells adjacent to the CUT. This method is validated by the research presented in [13]. To counteract the effects of extended targets on detection accuracy, various CFAR techniques like GO, CA, and SO-CFAR employ a small number of guard cells flanking the CUT for additional protection.

3- Lin-DBSCAN-CFAR Processor

The Lin-DBSCAN-CFAR process is an advanced method used for determining the background level in sea surface clutter, utilizing an ANN model. This model is specifically designed to calculate the shape parameter of sea surface clutter with high accuracy. Lin-DBSCAN, a variant of machine learning models, is based on a density-based clustering algorithm.

This method hinges on two key parameters: the minimum number of points within a neighborhood, denoted as MinPts, and the neighborhood radius, represented as Eps. These parameters are integral to the discretization step, which is a critical aspect of the Lin-DBSCAN methodology for network modeling. The parameters Eps, the neighborhood radius, and MinPts, the minimum number of points within this radius, are closely associated with the discretization step in Lin-DBSCAN [15].

Lin-DBSCAN, unlike the standard DBSCAN which focuses on each point and its neighborhood, identifies connected areas based on density by analyzing grid cells. The method involves uniformly subdividing a hyper-rectangle by overlaying a multidimensional grid. The size of this grid is synchronized with the chosen discretization step. In Lin-DBSCAN, this discretization step is directly proportional to the Eps parameter used in DBSCAN.

Density-based clustering, as illustrated by the DBSCAN algorithm, clusters data points based on their spatial density, enabling the identification of natural clusters within the data without pre-specifying the number of clusters. In DBSCAN, core points are identified as those having a specified number of neighboring points within a given radius, forming the nucleus of clusters. These clusters are then expanded by incorporating additional points that meet the density criteria. Points failing to meet these density requirements are classified as noise, making this method particularly effective for dealing with data that has non-uniform dense distributions and identifying outlier points [21].

Grid-based clustering methods partition the data space into a finite number of cells that form a grid structure and treat each cell as a unit for density analysis. Cells with high density are grouped to form clusters. This approach significantly speeds up the clustering process by reducing the need for intensive distance calculations between every pair of points, making it suitable for very large datasets. The grid-based approach simplifies the data space into a manageable number of cells, allowing for efficient processing and scalability [15].

The size of the grid cell in Lin-DBSCAN, denoted as ϵ, is defined by the equation $\gamma = \epsilon/2 * 2$, where γ represents a specific constant. This equation illustrates the direct relationship between ϵ and γ in a two-dimensional dataset, demonstrating how the uniform discretization step in Lin-DBSCAN is related to the DBSCAN parameters. **Fig.1** shows the relationship between ϵ and γ for a two-dimensional data set

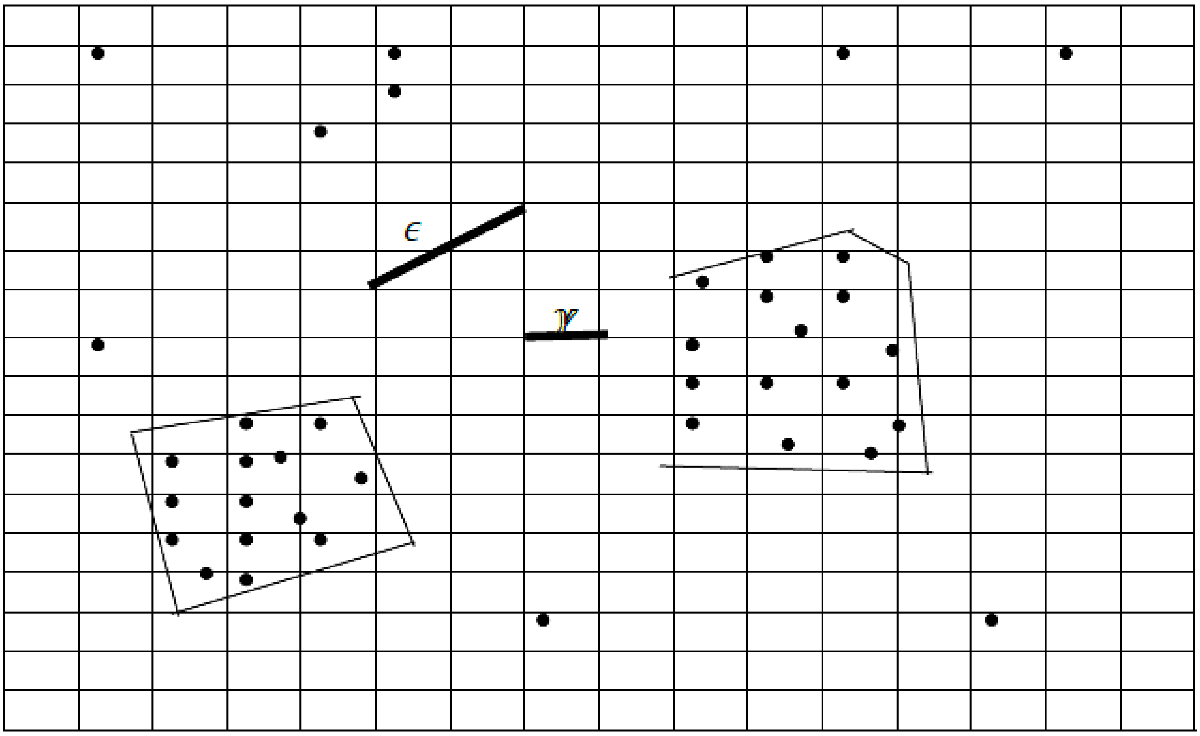

Fig.1: Relationship between ϵ and γ for a two-dimensional dataset

The study introduces a novel CFAR processor named Lin-DBSCAN-CFAR, designed for detecting multiple targets in K-distributed sea clutter. This innovative processor integrates the Lin-DBSCAN algorithm to effectively distinguish potential targets and sea spikes, typically statistical outliers, from the background clutter. Lin-DBSCAN-CFAR retains the foundational structure of traditional CFAR processors but adds an outlier rejection feature to the detection process. A notable innovation in this processor is the use of in-phase and quadrature components of radar returns for identifying outliers with Lin-DBSCAN. This ensures that signals classified as outliers are consistently recognized as such even after undergoing square-law detection. The study introduces a novel CFAR processor named Lin-DBSCAN-CFAR, designed for detecting multiple targets in K-distributed sea clutter. This innovative processor integrates the Lin-DBSCAN algorithm to effectively distinguish.

potential targets and sea spikes, typically statistical outliers, from the background clutter. Lin-DBSCAN-CFAR retains the foundational structure of traditional CFAR processors but adds an outlier rejection feature to the detection process. A notable innovation in this processor is the use of in-phase and quadrature components of radar returns for identifying outliers with Lin-DBSCAN. This ensures that signals classified as outliers are consistently recognized as such even after undergoing square-law detection.

Another key advancement in Lin-DBSCAN-CFAR is the calculation of the threshold factor a from an estimated shape parameter v, using a trained Artificial Neural Network (ANN) model, set against a predetermined probability of false alarm. This method contrasts with the closed-form solution typically employed in traditional CFAR processors for Rayleigh-distributed clutter. By filtering out outliers in the reference windows, Lin-DBSCAN-CFAR substantially improves the accuracy in estimating the clutter background level, which in turn enhances detection performance.

In radar signal processing, the complex echo signal received within a reference window is described by the equation:

$$X = [x1I + jx1Q, x2I + jx2Q, ..., xNI + jxNQ] \quad (2)$$

where N = 2n represents the length of the reference window. Furthermore, when each range unit sample undergoes processing through a square law detector, the resultant signal is given by:

$$x_i = x_{iI}^2 + x_{iQ}^2 \quad (3)$$

where i ranges from 1 to N. This formulation captures the essential components of the radar received signal and its subsequent processing as shown in the **Fig.2**.

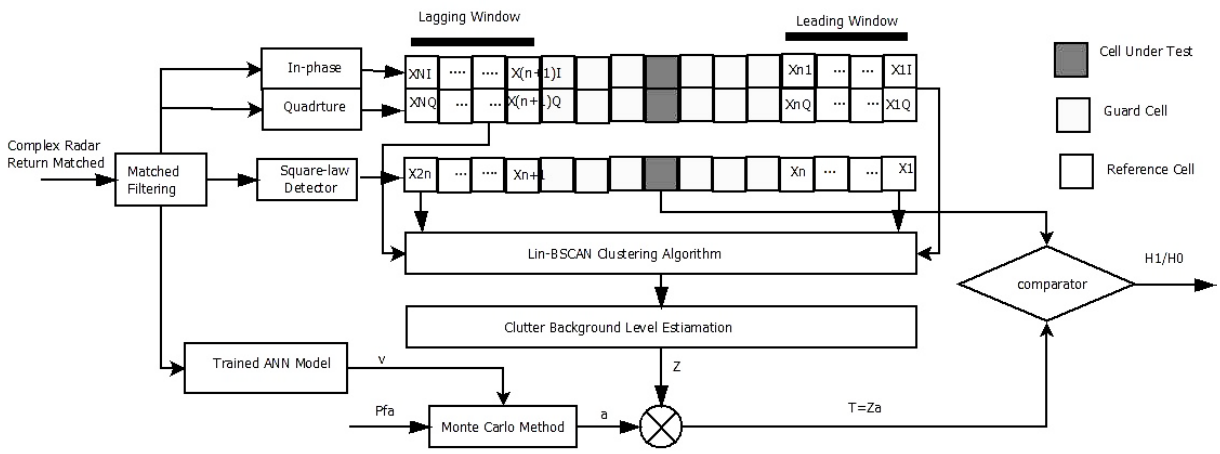

Fig.2: Block diagram of the proposed Lin-DBSCAN-CFAR processor

The Lin-DBSCAN-CFAR processor, used in signal processing, employs a sophisticated clustering technique to differentiate between normal data points and outliers in a set of signal samples. This method is vital in determining the background level of clutter in the CUT by identifying and excluding outliers.

In this process, a complex signal X with N samples is analyzed. If an index in signal X is identified as an outlier, the corresponding sample XM in signal xi is also classified as an outlier. By removing these outliers from Xi and averaging the remaining samples, the Lin-DBSCAN-CFAR processor can estimate the background clutter level in the CUT. The equation for this estimation is as explained in the reference [27].

$$Z_{Lin-DBSCAN} = \sum_{i=1}^{N-M} \frac{X_i}{x_m} \;, i = 1,2,\ldots,N \; where \; M \in [\mathbf{1}\,,\mathbf{N}] \quad (\mathbf{4})$$

Here, M represents the number of outliers isolated using this clustering technique. The decision-making process of the Lin-DBSCAN-CFAR processor is encapsulated in a binary hypothesis test, concisely described by the following equation:

$$X_0 \lessgtr_{H_0}^{H_1} \frac{a}{N-M} \sum_{i=1}^{N-M} \frac{X_i}{x_m} \;, i = 1,2,\ldots,N \; where \; M \in [\mathbf{1}\,,\mathbf{N}] \quad (\mathbf{5})$$

In tis formula, H_1 signifies the presence of a target within the CUT, while H_0 indicates the absence of a target. X_0 and a represent the actual values of the CUT and the threshold factor, respectively. This approach allows for a precise and reliable identification of significant signal components, enhancing the effectiveness of the processor in signal detection tasks.

Algorithm.1 presents the detailed pseudo-code for the Lin-DBSCAN-CFAR processor, utilizing the specified ingredients

---

Algorithm.1 Proposed Lin-DBSCAN-CFAR Detection Method

1: Input: Reference cell count: N , Guard cell count: M , Complex radar return samples in reference window: (x1I + jx1Q), . . . , (xNI + jxNQ), DBSCAN clustering specifics: Eps and MinPts

2: Output: Determination of target presence: Either H1 (target detected) or H0 (no target detected).

3: procedure

4: Initialize empty cluster set: W = ∅.

5: Transform each complex radar sample into a 2D point to create dataset D = {(x1I , x1Q), . . . , (xNI , xNQ)}.

6: for each point p in dataset D do

7: if p is already processed then

8: Skip to the next point.

9: else

10: Evaluate the number of points within Eps-distance of p, denote as |NEps(p)|.

11: if |NEps(p)| < MinPts then

12: Label p as a border point or outlier.

13: else

14: Classify p as a core point and assign all points in its Eps-neighborhood to cluster W .

15: for each unprocessed point q in the Eps-neighborhood of p do

16: if |NEps(q)| ≥ MinPts then

17: Include its neighborhood points in cluster W .

18: end if

19: end for

20: end if

21: end if

22: end for

23: Lin-DBSCAN clustering outcome.

24: Compute the clutter level ZLin−DBSCAN based on Equation (4).

25: Determine target detection (H1 or H0) according to Equation (5).

26: end procedure

Previous studies indicate that to create smaller clusters and reduce the impact of outliers, the value of MinPts should not be excessively large or small. In general, for two-dimensional data, MinPts = 4 is appropriate. Several methods have been proposed to determine the value of the second parameter Eps, including equalization histograms and normalized density lists [15]. The value of Eps is estimated based on the characteristics of the sea clutter data [9, 3]. In general, the optimal value is considered to be Eps = 2, as suggested in previous works [7].

4- Evaluation and Results

In this section of our research paper, we delve into a comprehensive analysis and comparative study of a range of signal processing algorithms: SO, GO, OS, CA, DBSCAN, and Lin-DBSCAN-CFAR. Our focus is centered on unraveling the complexities and evaluating the impacts of interference in scenarios involving multiple targets.

To provide a thorough understanding, we have conducted extensive simulations. These are designed to showcase the detection capabilities of the Lin-DBSCAN-CFAR processor, setting it in comparison with its counterparts such as SO, GO, CA, OS, and the traditional DBSCAN-CFAR. Our simulations encompass a wide spectrum of shape parameters, probabilities of false alarms, and varied multi-target situations. The methodology behind these simulations is rooted in the use of MATLAB 2019a on a Windows 8 64-bit system, powered by a 2.40 GHz Intel Core i7 processor and equipped with 6 GB of RAM. This section aims to present a detailed and nuanced understanding of the effectiveness and superiority of these processors in complex and realistic operational environments.

4.1- Impact of Interference in Multi-Target Scenarios

In our study, we focused on analyzing the impact of interference targets within reference cells, specifically centering a primary target in the CUT for in-depth examination. We assumed these interference targets were of equal strength to the primary target. Following the methodology outlined in [27], we set the shape parameter 'v' to 2.02.

Our analysis compared Lin-DBSCAN-CFAR and DBSCAN-CFAR, adhering to parameters from [27]: using 64 reference cells (N) and 4 guard cells (M), Eps = 2, MinPts = 4 for both algorithms, a k value of 60 for OS- CFAR, and setting a Pfa of 10^{-4}.

We conducted extensive simulations over 10^{5} Monte Carlo iterations, testing various SNR for each CFAR processor.

Our results, illustrated in **Fig.3**, show how the Probability of Detection varies with different SNR levels under varying counts of interference targets, while maintaining constant shape parameters and false alarm probabilities. In scenarios using Lin-DBSCAN-CFAR and DBSCAN-CFAR, we found that an increase in interference targets minimally impacted detection probability. This resilience is attributed to their efficient filtering of interference targets as extraneous factors, without needing prior knowledge of their presence. Both Lin-DBSCAN-CFAR and DBSCAN-CFAR exhibited remarkably similar detection performances.

Fig.4 demonstrates that CA, GO, SO, and OS-CFAR processors generally experienced a decrease in detection probability with an increased number of interference targets. However, OS-CFAR maintained its effectiveness, likely due to its ability to eliminate interference targets using prior information. These findings suggest that Lin-DBSCAN-CFAR is effective against a high count of interference targets, offering comparable performance to DBSCAN-CFAR, but with less complexity.

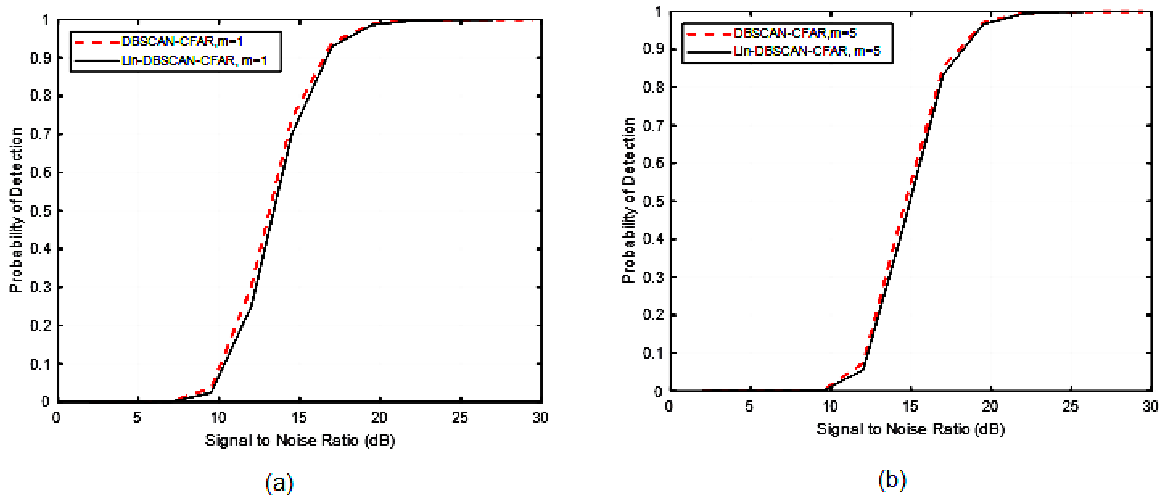

Fig.3: A comparison of the probabilities of detection with different numbers of interference targets for Lin-DBSCAN-CFAR and DBSCAN-CFAR (a) m=1, (b) m=5.

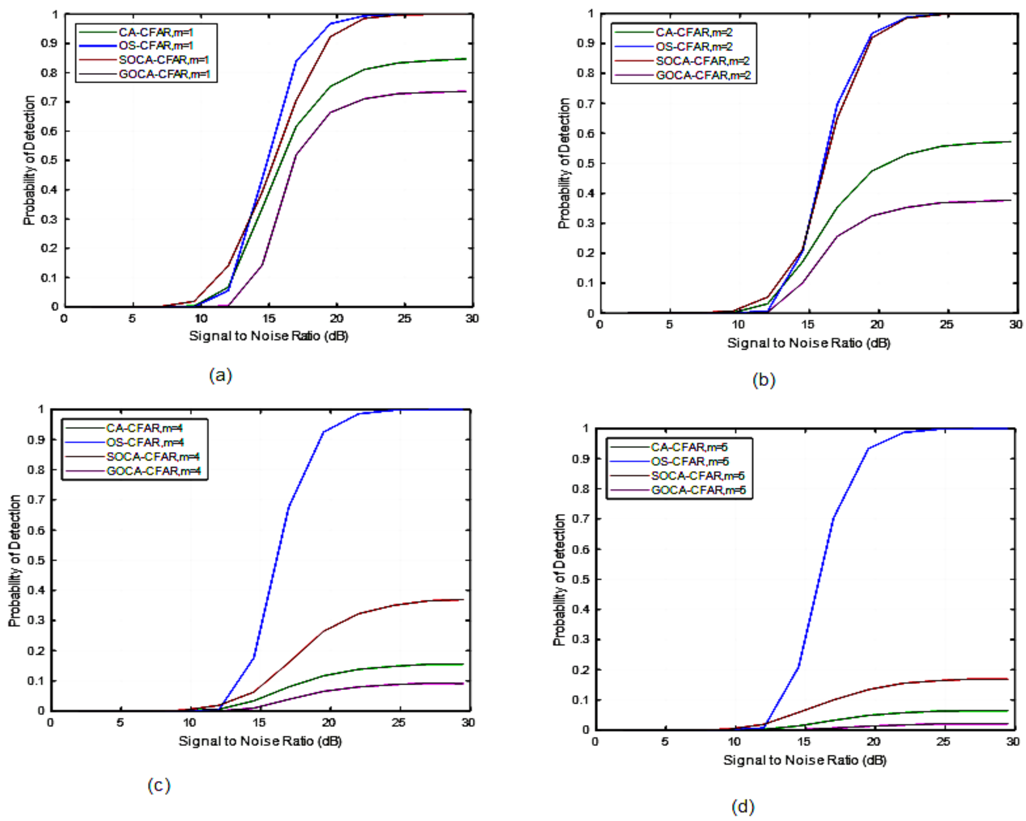

Fig.4: A comparison of the probabilities of detection with different numbers of interference targets for CA, SO, GO, OS-CFAR (a) m = 1. (b) m = 2. (c) m = 4. (d)m =5.

4.2- Effects of Different Shape Parameters

In our study, we concentrate on evaluating the performance of CFAR processors through extensive Monte Carlo simulations. These simulations are designed to assess the detection capabilities of these processors, with a special emphasis on the influence of different shape parameters, while maintaining a constant probability of false alarms. The specific settings for these simulations include overlaps (m) set to 5, a Pfa = $10^{-4}$and utilizing N = 64 and M = 4.

The primary goal of our research is to thoroughly understand how varying shape parameters affect CFAR detection in diverse sea clutter environments. We meticulously examine shape parameters at values of 0.201, 0.502, 1.023, 6.027, 10.109, and 20.121. This analysis aims to determine the impact of these parameters on the efficiency of CFAR detection processors and their adaptability in various sea clutter scenarios

In one part of our study, depicted as **Fig.5**, we illustrate the correlation between detection probability and Signal-to-Noise Ratio under the specified shape parameters, keeping overlaps and the probability of false alarms constant. Our observations indicate that the detection performance generally improves with higher shape parameters. Specifically, we notice a significant decline in detection probability with a shape parameter below one, attributed to increased sea peaks, as observed in both Lin-DBSCAN-CFAR and DBSCAN-CFAR algorithms Another aspect of our analysis, shown as **Fig.6**, offers a more comprehensive examination. It reveals that for CA, SO, and GO-CFAR algorithms, the detection probability tends to approach zero when the shape parameter is below one. However, their performance markedly improves when the parameter exceeds this threshold. Interestingly, the OS-CFAR method consistently exhibits superior detection probability across all tested shape parameter settings, highlighting its effectiveness in such conditions.

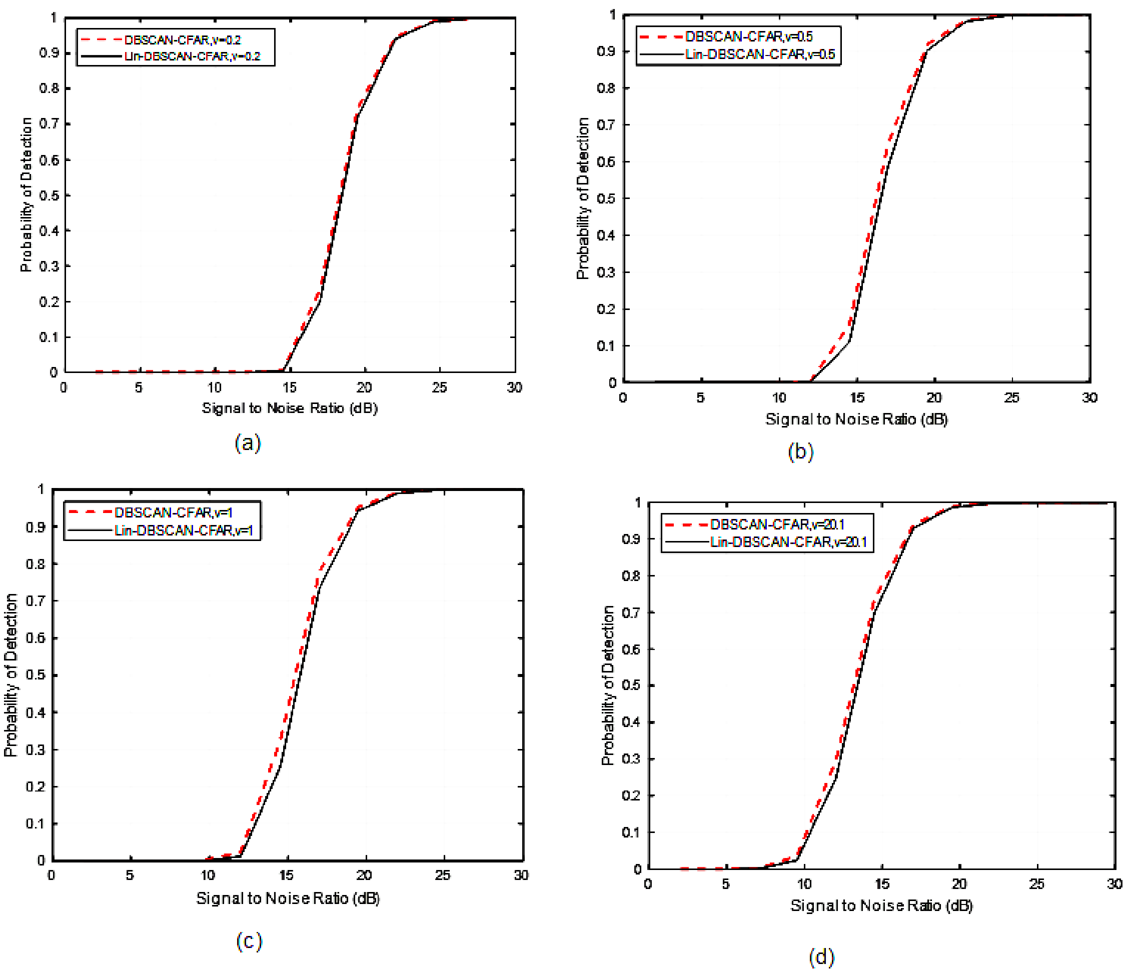

Fig.5: Probability of detection with a different number of ship parameter for for Lin-DBSCAN-CFAR and DBSCAN-CFAR (a) v=0.2, (b), v=0.5, (c) v=1,(d) v=20.1

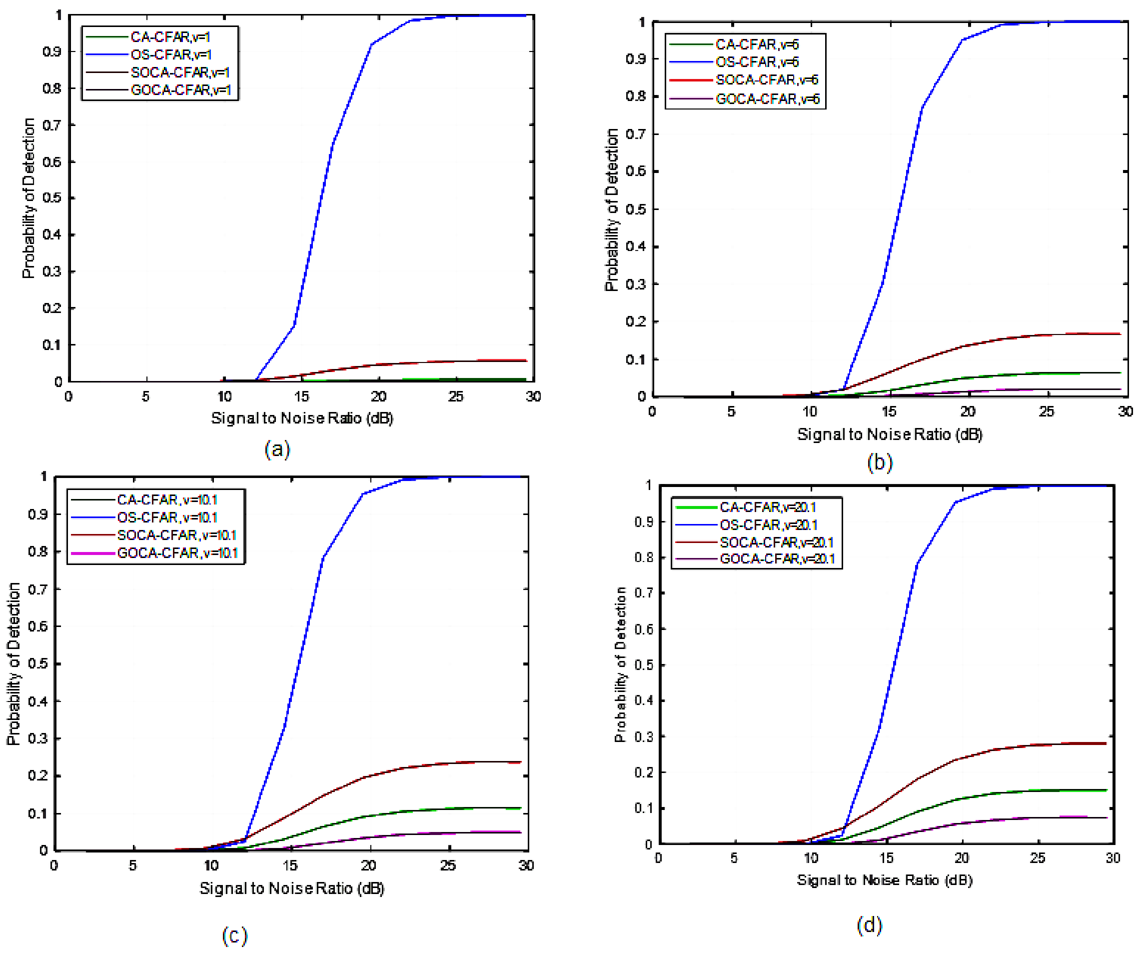

Fig.6: Probability of detection with a different number of ship parameter for CA, SO, GO, OS-CFAR (a) v=0.2, (b), v=0.5, (c)v=1, (d) v=20.1.

4.3- The effect of the probability of a false alarm

In this detailed examination, we evaluate the performance of different CFAR processors, emphasizing their dependability and resilience across a range of false alarm probabilities. We analyze four specific false alarm probabilities: Pfa = 10^{-2}, Pfa = 10^{-3}, Pfa = 10^{-4} and Pfa =10^{-5}. Our study maintains constant variables such as the number of interfering targets (m = 5) and system parameters (N = 64, M = 4), while also considering specific values for the shape parameter.

The data, illustrated in **Fig.7**, highlights the relationship between the detection probability and the SNR for these varying Pfa values. This graph is crucial for understanding how CFAR processors perform under different false alarm conditions. It is observed that the detection probability for both the DBSCAN-CFAR and Lin-DBSCAN-CFAR processors consistently increases with the SNR, which ranges from 5 to 30 dB in each simulation.

Fig.8 demonstrates an interesting trend: as the probability of false alarms decreases, there is a noticeable drop in detection performance. Despite this, both Lin-DBSCAN-CFAR and DBSCAN-CFAR processors outperform their counterparts across all tested false alarm probabilities. These results confirm the exceptional reliability and robustness of the Lin-DBSCAN-CFAR and DBSCANCFAR processors. They demonstrate superior performance compared to other

CFAR processors, making them highly effective and dependable in various detection scenarios, particularly those with variable false alarm probabilities. This study reaffirms the efficiency of these proposed CFAR processors as resilient and reliable options for detection applications.

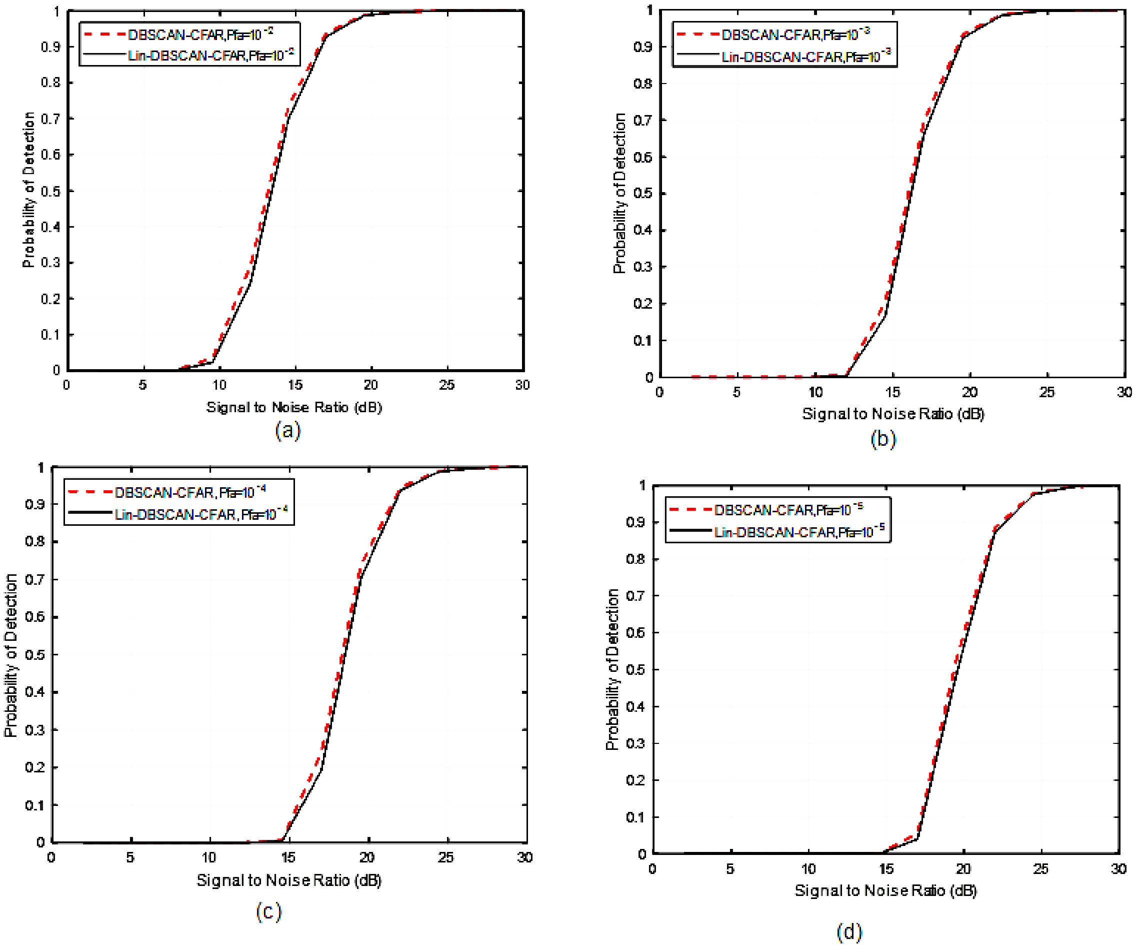

Fig.8: Probability of detection with a different number of probability of false alarm for Lin-DBSCAN-CFAR and DBSCAN-CFAR.(a)P fa = 10^{-2} (b) Pfa = 110^{-3} (c) Pfa = 110^{-4},(d) Pfa =10^{-5}

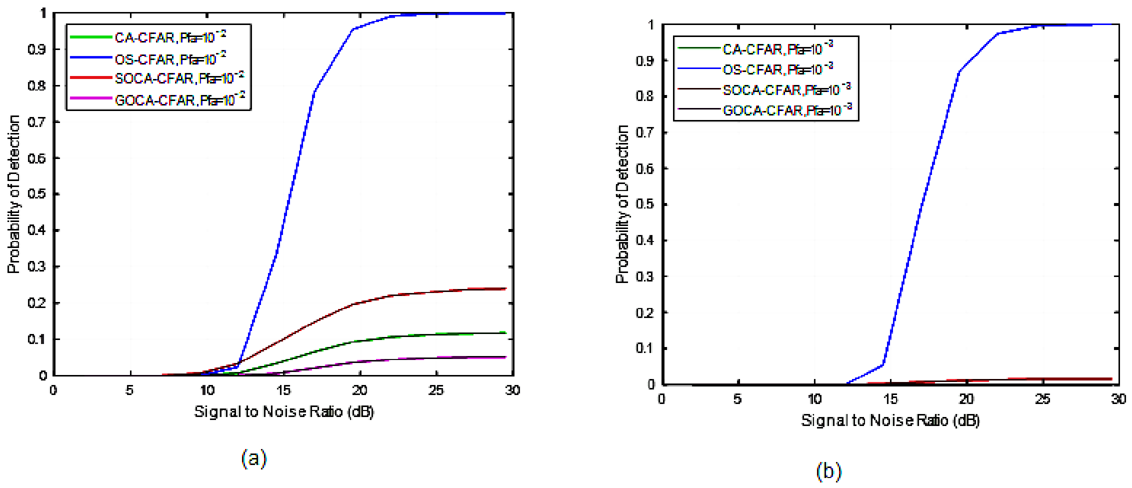

Fig.7: Probability of detection with a different number of probability of false alarm for CA, SO, GO, OS-CFAR. (a)P fa = 10^{-2} (b) Pfa = 10^{-3}

5-Aalysis of Computational Complexity

The complexity of the process in DBSCAN is O(n2)., which has led to the emergence of numerous algorithms aimed at enhancing its execution time. These include FDBSCAN [4], IDBSCAN [6], HDBSCAN [8], TI- DBSCAN [16], and Grid-DBSCAN [18]. Although these proposed algorithms do enhance execution speed in most cases, the best achievable process complexity is still O (n log n). As a response to this challenge, Lin-DBSCAN is proposed to further improve process complexity.

The Lin-DBSCAN algorithm checks each data point in the input set once to construct and fill the grid. It computes indexes for the cell to which each data point belongs in each iteration, checking if a non-empty cell already exists in the hash map of the grid. The total cost of this operation is O(n), where n is the total number of data points in the set. This cost efficiency is achieved because the hash map keeps the average access cost of a cell constant. In the collection phase, the Lin-DBSCAN algorithm accesses each cell only once. Therefore, the computational cost of the filling procedure is related to the number of non-empty cells in the grid, denoted by C. This quantity is normally proportional to the size and distribution of cells in the graph. Since the segmentation map ignores the empty cells, they are ignored. As a result, the overall computational complexity is for the algorithm O(n) + O(3dC). The optimal scenario will lead to a total cost of O(n) + O (1), since all the input points are contained within a single cell. Conversely, in the worst-case scenario, each entry point is isolated to a cell odd, resulting in a total number of cells equal to the number of points [21]. Here, the total cost becomes O(n) + O(3dn), where d represents the dimensionality of the data set and 3d signifies the number of cells within a neighborhood. From this analysis, it is evident that Lin-DBSCAN enhances the execution complexity more effectively than DBSCAN and other proposed algorithms.

The Lin-DBSCAN algorithm represents a significant advancement in data clustering techniques, particularly when compared to the standard DBSCAN algorithm and its various derivatives. The original DBSCAN algorithm, with a computational complexity of O(n2), was improved upon by several iterations like FDBSCAN [4], IDBSCAN [6], HDBSCAN [8], TI-

DBSCAN [16], and Grid-DBSCAN. These adaptations primarily aimed at accelerating the execution time, achieving a process complexity of at best O (n log n).

Lin-DBSCAN, however, introduces a more efficient approach. It operates by initially iterating through each data point in the set once, for the purpose of constructing and populating a grid. The algorithm assigns each data point to a specific cell in the grid, creating an index for each cell and checking against a hash map to identify if the cell is already populated. This operation bears a computational cost of O(n), with 'n' representing the total number of data points. The efficiency here is attributed to the hash map, which ensures a constant average access cost for each cell.

The collection phase of Lin-DBSCAN is even more efficient. Each cell in the grid is accessed only once, with the computational effort correlating to the count of non-empty cells, denoted as 'C'. This count generally depends on the grid's size and the distribution of cells. Importantly, empty cells are disregarded in this process. Consequently, the overall computational complexity of Lin-DBSCAN is O(n) + O (3d × C), where 'd' is the dimensionality of the dataset and 3d indicates the number of neighboring cells.

The best-case scenario for Lin-DBSCAN is when all input points fall within a single cell, yielding a total cost of O(n) + O (1). In contrast, the worst-case scenario is when every data point is isolated in its own cell, leading to a complexity of O(n) + O (3d ×n). This analysis clearly demonstrates that Lin-DBSCAN significantly improves upon the execution complexity, outperforming DBSCAN and other related algorithms in efficiency.

Conclusion

The paper discussed the implementation of Lin-DBSCAN-CFAR processor as a method to reduce variance in background level estimation in K-distributed sea clutter, particularly in multi-target environments. Lin- DBSCAN clustering technology was employed to eliminate outliers, such as sea spikes and interference targets, from the reference window. The background level of the clutter was then calculated using the remaining reference cells. The detection threshold was set by comparing it to the CUT, determined by the product of the threshold factor and the background clutter level. Simulations showed that the Lin-DBSCAN-CFAR processor surpassed other CFAR processors in various metrics, including false alarm probabilities, target counts, and shape parameters.

Additionally, it was found to have lower computational complexity and cost compared to DBSCAN-CFAR. These findings highlight the Lin-DBSCAN-CFAR's efficiency and effectiveness, making it a valuable tool for addressing background level estimation and target discrimination in multi-target scenarios. The results also provide a solid base for future research in optimizing and expanding the Lin-DBSCAN-CFAR approach for various applications requiring robust data analysis and classification.

References

[1] Sébastien Angelliaume, Luke Rosenberg, and Matthew Ritchie. Modeling the amplitude distribution of radar sea clutter. Remote Sensing, 11(3):319, 2019.

[2] Barkahem Zattouta and Lotfi Messikh. An adaptive cfar processor based on automatic censoring technique for target detection in heterogeneous environments. International Journal of Applied Engineering Research, 13(15):11927–11936, 2018.

[3] Ramiz M Aliguliyev. Performance evaluation of density-based clustering methods. Information Sciences, 179(20):3583–3602, 2009.

[4] Amineh Amini, Hadi Saboohi, Teh Ying Wah, and Tutut Herawan. A fast density-based clustering algo- rithm for real-time internet of things stream. The Scientific World Journal, 2014.

[5] BC Armstrong and HD Griffiths. Cfar detection of fluctuating targets in spatially correlated k-distributed clutter. In IEE Proceedings F-Radar and Signal Processing, volume 138, pages 139–152. IET, 1991.

[6] Bhogeswar Borah and Dhruba K Bhattacharyya. An improved sampling-based dbscan for large spatial databases. In International conference on intelligent sensing and information processing, 2004. proceedings of, pages 92–96. IEEE, 2004.

[7] Avory Bryant and Krzysztof Cios. Rnn-dbscan: A density-based clustering algorithm using reverse nearest neighbor density estimates. IEEE Transactions on Knowledge and Data Engineering, 30(6):1109–1121, 2017.

[8] Ricardo JGB Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. ACM Transactions on Knowledge Discovery from Data (TKDD), 10(1):1–51, 2015.

[9] Xiaoming Chen, Wanquan Liu, Huining Qiu, and Jianhuang Lai. Apscan: A parameter free algorithm for clustering. Pattern Recognition Letters, 32(7):973–986, 2011.

[10] Mohamed Bakry El-Mashade. Inhomogeneous analysis of novel model of cfar approaches to detect two- degrees of freedom partially-correlated $\chi 2$-targets. WSEAS Transactions on Communications, 20:28–43, 2021.

[11] NB Gálvez, JE Cousseau, JL Pasciaroni, and OE Agamennoni. Improved neural network based cfar detection for non homogeneous background and multiple target situations. Latin American applied research, 42(4):343–350, 2012.

[12] Prashant P Gandhi and Saleem A Kassam. Analysis of cfar processors in nonhomogeneous background.
IEEE Transactions on Aerospace and Electronic systems, 24(4):427–445, 1988.

[13] V Gregers Hansen and James H Sawyers. Detectability loss due to" greatest of" selection in a cell-averaging cfar. IEEE Transactions on Aerospace and Electronic Systems, (1):115–118, 1980.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[15] Jian Hou, Huijun Gao, and Xuelong Li. Dsets-dbscan: A parameter-free clustering algorithm. IEEE Transactions on Image Processing, 25(7):3182–3193, 2016.

[16] Marzena Kryszkiewicz and Piotr Lasek. Ti-dbscan: Clustering with dbscan by means of the triangle inequality. In International Conference on Rough Sets and Current Trends in Computing, pages 60–69. Springer, 2010.

[17] Y Li, J Zhu, J Zhang, W Wang, and C Duan. Two-step detection algorithm for fluctuating weak target based on dynamic programming. J. Eng. Technol.(JET), 10(1), 2019.

[18] Shaaban Mahran and Khaled Mahar. Using grid for accelerating density-based clustering. In 2008 8th IEEE International Conference on Computer and Information Technology, pages 35–40. IEEE, 2008.

[19] Xiaoqian Mou, Xiaolong Chen, Jian Guan, Baoxin Chen, and Yunlona Dong. Marine target detection based on improved faster r-cnn for navigation radar ppi images. In 2019 International Conference on Control, Automation and Information Sciences (ICCAIS), pages 1–5. IEEE, 2019.

[20] Jun Pan, Shengbo Ye, Cheng Shi, Kun Yan, Xin Liu, Zhikang Ni, Guangyao Yang, and Guangyou Fang. 3d imaging of moving targets for ultra-wideband mimo through-wall radar system. IET Radar, Sonar & Navigation, 15(3):261–273, 2021.

[21] Roberto Pirrone, Vincenzo Cannella, Sergio Monteleone, and Gabriella Giordano. Linear density-based clustering with a discrete density model. arXiv preprint arXiv:1807.08158, 2018.

[22] Hermann Rohling. Radar cfar thresholding in clutter and multiple target situations. IEEE transactions on aerospace and electronic systems, (4):608–621, 1983.

[23] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A review of clustering techniques and developments. Neurocomput- ing, 267:664–681, 2017.

[24] Jianbing Shen, Xiaopeng Hao, Zhiyuan Liang, Yu Liu, Wenguan Wang, and Ling Shao. Real-time superpixel segmentation by dbscan clustering algorithm. IEEE transactions on image processing, 25(12):5933–5942, 2016.

[25] Sai-Nan Shi and Peng-Lang Shui. Optimum coherent detection in homogenous k-distributed clutter. IET Radar, Sonar & Navigation, 10(8):1477–1

[26] Ruifu Wang, Jie Li, Yaping Duan, Hongjun Cao, and Yingjie Zhao. Study on the combined application of cfar and deep learning in ship detection. Journal of the Indian Society of Remote Sensing, 46(9):1413–1421, 2018.

[27] Jiafei Zhao, Rongkun Jiang, Xuetian Wang, and Hongmin Gao. Robust cfar detection for mul- tiple targets in k-distributed sea clutter based on machine learning. Symmetry, 11(12):1482, 2019 https://www.mdpi.com/2073-8994/11/12/1482.

[28] Wei Zhou, Junhao Xie, Kun Xi, and Yuhan Du. Modified cell averaging cfar detector based on grubbs criterion in non-homogeneous background. IET Radar, Sonar & Navigation, 13(1):104–112, 2018.

[29] Sohee Lim, Seongwook Lee, and Seong-Cheol Kim. Clustering of detected targets using DBSCAN in automotive radar systems. In Proceedings of the 2018 19th International Radar Symposium (IRS), pages 1–7. IEEE, 2018.

[30] Liang Zhang, Zhijun Zhang, Shengtao Lu, Deliang Xiang, and Yi Su. Fast superpixel-based non-window CFAR ship detector for SAR imagery. Remote Sensing, 14(9):2092, 2022.

[31] Canming Yao, Chao Li, Xue Jin, and Lei Zhang. A Fast Two-Parameter CFAR Algorithm Based on FFT for Ship Detection in Large-Scale SAR Images. In Proceedings of the 2022 5th

International Conference on Information Communication and Signal Processing (ICICSP), pages 244–248. IEEE, 2022.
[32] Wenjie Shen, Jie Zhi, Yanping Wang, Jinping Sun, Yun Lin, Yang Li, and Wen Jiang. Two-Step CFAR-Based 3D Point Cloud Extraction Method for Circular Scanning Ground-Based Synthetic Aperture Radar. Applied Sciences, 13(12):7164, 2023.