Hibrit FCM-BOA Veri Kümeleme Algoritması
Hybrid FCM-WOA Data Clustering Algorithm

Hatice ARSLAN ve Metin TOZ
Elektrik Elektronik ve Bilgisayar Mühendisliği, Bilgisayar Mühendisliği
Düzce Üniversitesi
Düzce, Türkiye
haticearslan8154@gmail.com, metintoz@duzce.edu.tr

***Özetçe*— Bu çalışmada, Chebyshev uzaklık fonksiyonu kullanılarak Fuzzy C-Means (FCM) ve Balina Optimizasyon Algoritmasını (BOA) bütünleştiren hibrit bir kümeleme algoritması önerilmiştir. FCM algoritması veriler arasındaki benzerliği ölçmek için Öklid uzaklığını kullanmaktadır. Öklid uzaklığının mevcut dezavantajlarını gidermek için FCM algoritmasındaki tüm uzaklıklar Chebyshev uzaklık fonksiyonu ile hesaplanmıştır. BOA algoritması başlangıç küme merkezlerini optimize etmek için kullanılmıştır. Önerilen algoritma UCI Machine Learning Repository veri tabanından seçilen 3 farklı veri seti ile test edilmiştir. Sonuç olarak önerilen algoritmanın kümeleme performansının FCM algoritmasından çok daha iyi olduğu görülmüştür.**

Anahtar Kelimeler —FCM; BOA; Chebsyhev; uzaklık fonksiyonu, veri kümeleme

***Abstract*— In this work, we propose a hybrid clustering algorithm that integrates Fuzzy C-Means (FCM) and Whale Optimization Algorithm (WOA) using the Chebshev distance function. The FCM algorithm uses Euclidean distance to measure the similarity between the data. To avoid the existing disadvantages of the Euclidean distance, all distances in the FCM algorithm is calculated with the Chebsyhev distance function. The BOA algorithm is used to optimize the initial cluster centers. The proposed hybrid algorithm is tested with three different sets of data selected from UCI Machine Learning Repository database. As a result, it is seen that the clustering performance of the proposed algorithm is much better than the FCM algorithm.**

Keywords —FCM; WOA; *Chebsyhev; distance function; data clustering*

I. GİRİŞ

Kümeleme, verileri sınıflandırmak için kullanılan denetimsiz bir öğrenme tekniğidir [1]. Veriler, aralarındaki uzaklık ve benzerlik kriterlerine göre gruplandırılırlar. Kümeleme sonucu gruplar içi benzerliğin çok, gruplar arası benzerliğin az olması hedeflenmektedir. Kümeleme algoritmaları görüntü tanıma, veri madenciliği, makine öğrenmesi ve sinyal işleme gibi pek çok alanda kullanılmaktadır [2]. Literatürde en sık kullanılan kümeleme algoritmalarından biri bulanık kümeleme prensibine dayanan Fuzzy C-Means (FCM) algoritmasıdır. FCM algoritması Dunn [3] tarafından önerilip Bezdek [4] tarafından geliştirilmiştir. FCM algoritmasında veriler kesin olarak bir kümeye ait olmak yerine her kümeye farklı üyelik dereceleri ile aittirler. Kümeleme işleminde veri noktaları ve küme merkezleri arasındaki benzerlik bir uzaklık fonksiyonu ile ölçülür. FCM algoritması küme merkezlerinin rastgele atanması ile başlar. Verilerin bu küme merkezlerine olan uzaklıkları tek tek hesaplanır ve her veri kendine en yakın kümeye atanır. Bu sebeple uygun uzaklık fonksiyonunu kullanmak önemlidir [5]. İki nokta arasındaki uzaklığı hesaplamak için literatürde pek çok farklı teknik vardır. En iyi bilinen ve en çok kullanılan Euclidean (Öklid) uzaklığının yanı sıra, Manhattan uzaklığı, Cosine uzaklığı, Pearson uzaklığı, Jaccard uzaklığı ve bu çalışmada kullanılan Chebyshev uzaklığı bunlardan bir kaçıdır [6,7].

FCM algoritması kolay uygulanabilir ve büyük veri kümelerinde hızlı çalışır olmasına rağmen yerel optimuma kolayca takılabilmektedir. Ayrıca başlangıç küme merkezlerinin rastgele seçilmesi de bu durumu tetikleyerek bir dezavantaj oluşturmaktadır. Son yıllarda bu problemlerin üstesinden gelmek için meta sezgisel algoritmalar kullanılmaya başlanmıştır [8]. Kümeleme problemi de bir çeşit optimizasyon problemi olarak düşünüldüğünde bu tip algoritmaların optimum çözümü bulup yerel optimuma takılma riskini azalttığı söylenebilir [1]. Literatür incelendiğinde FCM algoritmasının pek çok meta sezgisel optimizasyon algoritması ile birleştirildiği görülmektedir [9-12].

Bu çalışmada FCM algoritmasında kullanılan Öklid uzaklık fonksiyonu yerine Chebyshev uzaklık fonksiyonu kullanılmıştır. Ardından Mirjalili ve Lewis [13] tarafından geliştirilen popülasyon tabanlı meta sezgisel bir optimizasyon algoritması olan Balina Optimizasyon Algoritması (BOA) ile FCM algoritması birleştirilerek hibrit bir kümeleme algoritması önerilmiştir. Önerilen algoritma başlangıç küme merkezlerinin BOA algoritması ile optimize edilmesi esasına dayanmaktadır. Her bir başlangıç küme merkezi için FCM algoritması küme merkezlerini güncellemekte ve FCM-BOA algoritması da amaç fonksiyonu minimize etmeye çalışmaktadır. FCM-BOA algoritması ile UCI Machine Learning Repository [14] veri tabanından seçilen üç farklı veri kümesi kullanılarak kümeleme yapılmıştır. Elde edilen sonuçlarının gerçek sonuçlarla ilişkisi Rand İndeks ve Adjust Rand İndeks değerleriyle karşılaştırılmıştır. Sonuçlar incelendiğinde FCM-BOA

978-1-5386-1501-0/18/$31.00 ©2018 IEEE

algoritmasının FCM algoritmasından daha iyi sonuçlar verdiği görülmektedir.

Çalışmanın ikinci bölümünde FCM, BOA, Öklid uzaklığı ve Chebyshev uzaklığı ile ilgili bilgi verilip önerilen FCM-BOA algoritması anlatılmıştır. Üçüncü bölümde deneysel çalışmalara ve son olarak dördüncü bölümde çalışmanın kısa bir özetine yer verilmiştir.

II. FCM VE BOA İLE VERİ KÜMELEME

A. FCM Algoritması

FCM algoritması bulanık kümeleme tekniği ile çalışır. Verilen bir n elemanlı X veri kümesini, $X = \{x_1, x_2, \dots, x_n\}$, c adet bulanık kümeye ayırır [11]. Bir v_i vektörü, $v_i = [v_1, v_2, \dots, v_c]$, i. nci küme merkezini temsil eder. Her veri örneği U üyelik matrisi ile temsil edilen bir üyelik derecesine sahiptir. Bir verinin tüm kümelere ait üyelik dereceleri toplamı 1 olmalıdır. Veri hangi kümeye daha yakınsa o kümeye ait üyelik derecesi daha büyük olacaktır. Üyelik matrisi aşağıdaki şekilde temsil edilmektedir [11].

$$\sum_{i=1}^{c} U_{ij} = 1 \qquad j = 1,2, \dots n \tag{1}$$

FCM algoritması amaç fonksiyon tabanlı bir algoritma olup, en küçük kareler yönteminin genellemesi olan aşağıdaki amaç fonksiyonu minimize etmeye çalışır [11].

$$J_m(U,V) = \sum_{i=1}^{n}\sum_{j=1}^{c} U_{ij}^{m} \left\| x_i - v_j \right\|^2 , \quad 1 \le m < \infty \tag{2}$$

Algoritma U üyelik matrisinin rastgele atanması ile başlatılır. Ardından (3) teki formüle göre küme merkezleri hesaplanır [11].

$$v_j = \frac{\sum_{i=1}^{n} U_{ij}^{m} x_i}{\sum_{i=1}^{n} U_{ij}^{m}} \tag{3}$$

Hesaplanan küme merkezlerine göre aşağıdaki formül kullanılarak U matrisi güncellenir [11].

$$U_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - v_i\|}{\|x_i - v_k\|} \right)^{2/(m-1)}} \tag{4}$$

Eski U matrisi ile yeni U matrisi arasındaki fark ε' dan küçük olana kadar yukarıdaki işlemler tekrar eder.

A.1 FCM Algoritmasında Kullanılan Uzaklık Fonksiyonları

FCM algoritmasında verilerin küme merkezlerine olan uzaklığı Öklid uzaklık fonksiyonu ile ölçülmektedir. Öklid uzaklığı iki nokta arasındaki en kısa mesafedir. $A(x_1, y_1)$ ve $B(x_2, y_2)$ bir düzlemde iki farklı nokta olmak üzere A noktası ile B noktası arasındaki uzaklık aşağıdaki formül ile hesaplanır [7].

$$d_{\text{öklid}} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{5}$$

İki nokta arasındaki uzaklığı hesaplamak için pek çok farklı teknik vardır. Seçilen tekniğin uygunluğu verinin niteliğine ve veri kümesinin boyutuna göre farklılık gösterebilir [6]. Kümeleme problemlerinde genellikle Öklid uzaklığı kullanılmasına rağmen bu uzaklık fonksiyonu kompleks şekillerde her zaman verimli olamamaktadır [5]. Bu çalışmada uzaklık fonksiyonu olarak Chebyshev uzaklığı kullanılmıştır. $A(x_1, y_1)$ ve $B(x_2, y_2)$ noktaları arasındaki uzaklık Chebsyhev uzaklık fonksiyonu ile aşağıdaki şekilde hesaplanır [7].

$$d_{chebyshev} = max(|x_1 - x_2|, |y_1 - y_2|) \tag{6}$$

Chebyshev uzaklığı, şahın satranç tahtasında başka bir kareye geçmek için yapması gereken hamle sayısıdır. Bu sebeple Chessboard (satranç tahtası) uzaklığı olarak da bilinmektedir.

B. BOA Algoritması

Balina optimizasyon algoritması (BOA) kambur balinaların avlanma stratejilerinden esinlenerek Mirjalili ve Lewis [13] tarafından geliştirilen global bir optimizasyon algoritmasıdır. Kambur balinalar kabarcık ağ beslenme metodu olarak adlandırılan eşsiz bir avlanma davranışına sahiptir. Su içerisinde oluşturdukları hava kabarcıkları avlarını sürü halinde bir araya toplayıp beslenmelerini sağlar.

BOA algoritmasında optimizasyon probleminin çözümü, araştırma ajanlarının lokasyonu ile temsil edilmektedir. Algoritmanın verimliliği ise amaç fonksiyonla ölçülür. İlk olarak, algoritma bir dizi rastgele çözüm ile başlar. Her bir iterasyonda araştırma ajanları konumlarını %50 ihtimalle ya rastgele seçilen araştırma ajanına ya da şu ana kadar elde edilen en iyi çözüme göre güncellerler. Keşif ve çalıştırma fazlarının sağlanması için a parametresi 2'den 0'a azaltılır. P'nin değerine bağlı olarak BOA, spiral ya da dairesel hareket arasında seçim yapabilir. Son olarak BOA, durdurma kriterinin sağlanmasıyla sonlandırılır. BOA algoritmasının matematiksel modelinin detaylı tanımı için [13]'e bakınız. BOA algoritmasının sözde kodu Şekil 2. de verilmiştir.

Balina popülasyonunu X_i $(i = 1, 2, \dots, n)$ başlatın.
Her araştırma ajanı için amaç fonksiyonu hesapla.
***While** (t<maksimum iterasyon sayısı)*
  ***for** her araştırma ajanı*
  a, A, C, l ve P'yi güncelle.
    ***İf1** $(p < 0.5)$*
      ***İf2** $(|A| < 1)$*
        Mevcut araştırma ajanının konumunu güncelle.
        $$\vec{D} = \left| \vec{C}.\overrightarrow{X^*}(t) - \vec{X}(t) \right|$$
      ***else if2** $(|A| \ge 1)$*
        Mevcut araştırma ajanının konumunu güncelle.
        $$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A}.\vec{D}$$
      end if2
    ***else if1** $(p \ge 0.5)$*
      Mevcut araştırma ajanının konumunu güncelle.
      $$\vec{X}(t+1) = \overrightarrow{D'}.e^{bl}\cos(2\pi l) + \overrightarrow{X^*}(t)$$
    end if1
  end for
  Herhangi bir araştırma ajanının araştırma uzayını aşıp aşmadığını kontrol et ve düzelt.

Her araştırma ajanı için amaç fonksiyonu hesapla.
Daha iyi bir çözüm varsa X^*' ı güncelle.
t=t+1
end while
X^*' a geri dön.

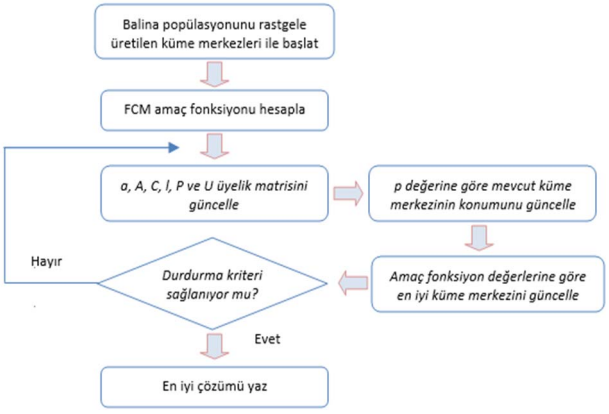
Şekil 1. *BOA algoritmasının sözde kodu* [13]

C. BOA ve FCM ile kümeleme

BOA algoritmasında araştırma ajanlarının aday konumlarını temsil eden X matrisi, hibrit algoritmada aday küme merkezlerini temsil etmektedir. n, veri kümesindeki eleman sayısı, k veri kümesindeki özellik sayısı ve c, küme sayısı olmak üzere X matrisi $n \times ck$ boyutunda aşağıdaki gibi tanımlanır.

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1,cxk} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{n,cxk} \end{bmatrix} \quad (7)$$

BOA algoritması ile oluşturan X matrisinin her bir satırı FCM algoritması için aday küme merkezleri içermektedir. X matrisinin her satırı için, yani her aday küme merkezi için, BOA-FCM algoritması FCM amaç fonksiyonunu minimize etmeye çalışır. FCM-BOA algoritmasının akış diyagramı şekil 2. de gösterilmektedir.



Şekil 2. *FCM-BOA Algoritmasının akış diyagramı*

D. Rand ve Adjust Rand İndeks

Rand indeks iki küme arasındaki benzerlik oranını ölçer. Aynı ve farklı kümelere atanmış veri sayısının, veri kümesinin eleman sayısına oranı ile hesaplanır. Birebir uyuşma sağlandığı durumda 1, aksi halde 0 değerini alır [15].

Adjust Rand indeks ise Rand indeksin düzeltilmiş halidir. Tahmine dayalı olarak benzerlik hesaplar. En kötü tahmini uyumda -1, tam tahmini uyumda 1 değerini alır [16].

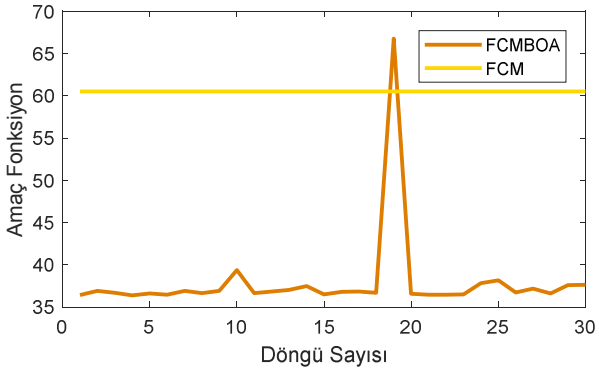
III. DENEYSEL ÇALIŞMALAR

Bu çalışmada önerilen FCM-BOA algoritmasının kümeleme performansını değerlendirmek için UCI Machine Learning Repository veri tabanından seçilen İris, Balance Scale ve Fertility veri kümeleri kullanılmıştır [14]. FCM ve FCM-BOA algoritmalarının kümeleme sonuçlarının gerçek sonuçlara yakınlığı literatürde sıkça kullanılan iki indeks olan Rand Indeks ve Adjust Rand Indeks değerleriyle test edilmiştir. Yapılan tüm çalışmalar aynı şartlar altında Intel Core i7-7700HQ CPU 2.80GHz işlemci ve 16 GB Ram özelliklerine sahip bir bilgisayarda Matlab R2017b programında gerçekleştirilmiştir. FCM ve FCM-BOA algoritmaları 30 kez art arda çalıştırılmış ve her ikisinde de m değeri 2 alınmıştır. FCM-BOA algoritmasında maksimum iterasyon sayısı 1000 ve popülasyon büyüklüğü 50' dir.

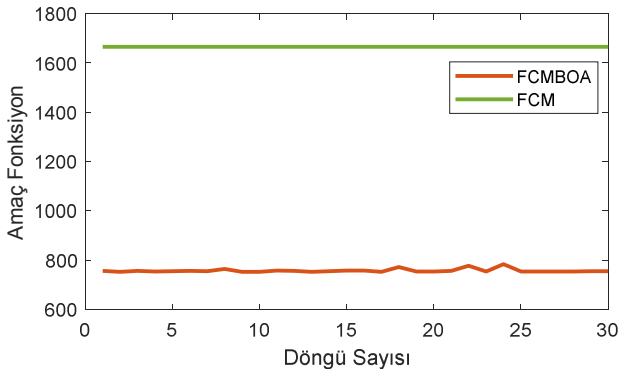
TABLO 1. Veri Kümeleme Sonuçları

Kümeleme Sonuçları	Kullanılan Algoritma	Amaç Fonksiyon (Ortalama)	Rand İndeks (Maksimum)	Rand İndeks (Ortalama)	Adjust Rand İndeks (Maksimum)	Adjust Rand İndeks (Ortalama)
İris Veri kümesi	FCM	60,5057	0,8797	0,8797	0,7294	0,7294
	FCM-BOA	**37,9383**	**0.9124**	**0.8891**	**0.8019**	**0.7518**
Balance Scale Veri Kümesi	FCM	1666,6600	0,7078	0,5758	0,3870	0,1169
	FCM-BOA	**757,7287**	**0.7208**	**0.5975**	**0.4187**	**0.1644**
Fertility	FCM	114,6610	0,5000	0,5000	0,0037	0,0037
	FCM-BOA	**48,0235**	**0.6768**	**0.5252**	**0.1589**	**0.0154**

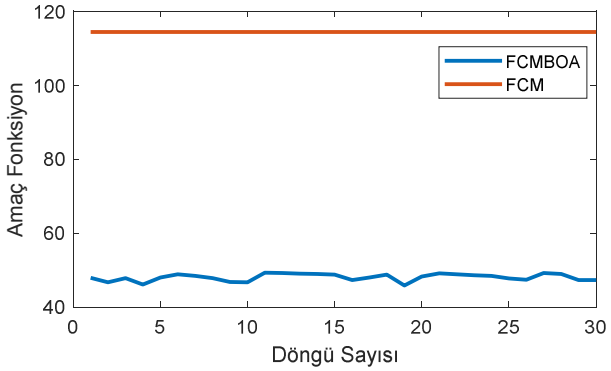
Tablo 1.' de 3 veri kümesinin her biri için ortalama amaç fonksiyon, Rand indeks ve Adjust Rand indeks değerleri verilmiştir. FCM-BOA algoritması tüm veri kümelerinde amaç fonksiyon değerlerini FCM algoritmasına göre çok daha iyi minimize etmiştir. Amaç fonksiyon değerlerine paralel olarak Rand indeks ve Adjust Rand indeks değerleri de tüm veri kümeleri için daha yüksektir. FCM-BOA algoritması için ortalama Rand indeks değerleri FCM algoritmasından daha iyi olmasının yanında maksimum Rand indeks değerleriyle ciddi farklar oluşturmuştur. Örneğin Rand indeks değerlerinin 30 çalıştırılma sonucu aldığı maksimum değerlere bakıldığında İris veri kümesi için FCM 0,8797 değerindeyken, FCM-BOA 0,9124, Balance Scale veri kümesi için FCM 0,7078 değerindeyken FCM-BOA 0,7208, Fertility veri kümesi için FCM 0,5 değerindeyken, FCM-BOA 0,6768 değerine kadar çıkabilmektedir. Ayrıca amaç fonksiyon değerleri Şekil 3-5 de grafiksel olarak gösterilmiştir. Grafikler incelendiğinde amaç fonksiyonlar arasındaki fark daha net bir şekilde görülmektedir.



Şekil 3. *İris veri kümesi için elde edilen amaç fonksiyon grafiği*



Şekil 4.*Balance Scale veri kümesi için elde edilen amaç fonksiyon grafiği*



Şekil 5. *Fertility veri kümesi için elde edilen amaç fonksiyon grafiği*

IV. SONUÇ

Bu çalışmada, balinaların avlanma davranışından esinlenerek geliştirilen popülasyon tabanlı bir optimizasyon algoritması olan Balina Optimizasyon Algoritması ile kümeleme problemlerinde yaygın olarak kullanılan Fuzzy C-Means algoritması birleştirilerek hibrit bir kümeleme algoritması önerilmiştir. Ayrıca FCM algoritmasındaki uzaklık fonksiyonu Chebyshev uzaklık fonksiyonu ile değiştirilip kümeleme performansının artması sağlanmıştır. Önerilen algoritmanın performansı UCI Machine Learning Repository [14] veri tabanından seçilen 3 farklı veri kümesi kullanılarak test edilmiş ve Rand-Adjust Rand indeks değerleriyle ölçülmüştür. Nihayetinde FCM algoritmasından çok daha iyi sonuçlar elde edilmiştir.

KAYNAKLAR

[1] Maheshwar, K. Kaushik, and V. Arora, "A Hybrid Data Clustering Using Firefly Algorithm Based Improved Genetic Algorithm," in *Procedia Computer Science*, 2015, vol. 58, pp. 249–256.

[2] X. H. Han, L. Quan, X. Y. Xiong, M. Almeter, J. Xiang, and Y. Lan, "A novel data clustering algorithm based on modified gravitational search algorithm," *Eng. Appl. Artif. Intell.*, vol. 61, no. September 2016, pp. 1–7, 2017.

[3] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1973.

[4] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, no. 2–3, pp. 191–203, 1984.

[5] J. Arora, "Hybrid FCM PSO Algorithm with CityBlock Distance," pp. 2609–2614, 2016.

[6] D. J. Bora and A. K. Gupta, "Effect of Different Distance Measures on the Performance of K-Means Algorithm : An Experimental Study in Matlab," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 2, pp. 2501–2506, 2014.

[7] M. Kabak, F. Sağlam, and A. Aktaş, "Farkli Uzaklik Hesaplama Yaklaşimlarinin Topsis Üzerinde Kullanilabilirliğinin İncelenmesi," *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Derg.*, vol. 32, no. 1, pp. 35–43, 2017.

[8] S. J. Nanda and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering," *Swarm Evol. Comput.*, vol. 16, pp. 1–18, 2014.

[9] D. Wang, B. Han, and M. Huang, "Application of Fuzzy C-Means Clustering Algorithm Based on Particle Swarm Optimization in Computer Forensics," *Phys. Procedia*, vol. 24, pp. 1186–1191, 2012.

[10] W. Zhu, J. Jiang, C. Song, and L. Bao, "Clustering algorithm based on fuzzy c-means and artificial fish swarm," *Procedia Eng.*, vol. 29, pp. 3307–3311, 2012.

[11] H. Izakian and A. Abraham, "Fuzzy C-means and fuzzy swarm for fuzzy clustering problem," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 1835–1838, 2011.

[12] E. Esme and B. Karlik, "Fuzzy c-means based support vector machines classifier for perfume recognition," *Appl. Soft Comput. J.*, vol. 46, pp. 452–458, 2016.

[13] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016.

[14] M. Bramer, *Principles of Data Mining*. 2013.

[15] "Objective Criteria for the Evaluation of Clustering Methods Author (s): William M . Rand Source : Journal of the American Statistical Association , Vol . 66 , No . 336 (Dec ., 1971), pp . 846- Published by : American Statistical Association Stable URL," vol. 66, no. 336, pp. 846–850, 2009.

[16] L. Hubert and P. Arabie, "Comparing partitions," *J. Classif.*, vol. 2, no. 1, pp. 193–218, 1985.