



Anlaşılabilir Sınıflandırma Kurallarının Ayçiçeği Optimizasyon Algoritması ile Otomatik Keşfi

Suna YILDIRIM^{*1}, Güngör YILDIRIM², Bilal ALATAŞ³

¹ Elazığ İl Özel İdaresi, Bilgi İşlem Birimi, Elazığ, Türkiye

² Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Elazığ, Türkiye

³ Fırat Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, Elazığ, Türkiye

Suna YILDIRIM ORCID No: 0000-0002-8246-0515

Güngör YILDIRIM ORCID No: 0000-0002-4096-4838

Bilal ALATAŞ ORCID No: 0000-0002-3513-0329

*Sorumlu yazar: sunayildirim23@gmail.com

(Alınış: 30.07.2021, Kabul: 30.09.2021, Online Yayınlanma: 31.12.2021)

Anahtar Kelimeler
 Optimizasyon,
 Kural Madenciliği,
 Ayçiçeği Optimizasyon Algoritması,
 Doğruluk

Öz: Kural madenciliği, veri madenciliğinin önemli alt dallarından biri olup günümüzde hala üzerinde çalışılan sıcak bir çalışma alanıdır. Nicel nitelik içeren veri setleri üzerinde çalışan standart sınıflandırma yöntemleri genellikle ön işlem aşamalarına ihtiyaç duyarlar. Bu yapılan ayrıklaştırmalar ise başarımlarına yol açabilmektedir. Buna ek olarak standart sınıflandırma algoritmalarının kara-kutu yapılarından dolayı kural açıklanabilirlikleri iyi değildir. Bu noktada, sürekli veriler ile çalışabilen optimizasyon algoritmaları, bu dezavantajların üstesinden gelebilir. Bu çalışmada, son yılların başarılı optimizasyon algoritmalarından olan Ayçiçeği Optimizasyon algoritmasını kullanarak verimli bir kural madenciliği gerçekleştirilmiştir. Bunun için, farklı bir temsil biçimi kullanan aday bitki yapısı, bu optimizasyon algoritmasına uyarlanmıştır. Arama uzayı olarak üç farklı disipline ait veri seti kullanılmış ve yöntemin başarımlarını gözlemlemek için iyi bilinen beş farklı sınıflandırma algoritmasına ait sonuçlar paylaşılmıştır. Elde edilen sonuçlar, optimizasyon temelli yaklaşım ile veri setleri üzerinde herhangi bir ön işlem yapmaya gerek kalmadan açıklanabilir kurallar üretilebileceğini ispatlamaktadır.

233

Automatic Discovery of Comprehensible Classification Rules with Sunflower Optimization Algorithm

Keywords
 Optimization,
 Rule Mining,
 Sunflower Optimization Algorithm,
 Accuracy

Abstract: Rule mining is one of the important sub-branches of data mining, and it is still a hot topic study area for researchers. Standard classification methods usually require pre-processing steps when working with datasets containing quantitative attributes. On the other hand, discretization at these stages may lead to a loss of performance and accuracy. In addition, due to the black-box nature of standard classification algorithms, rule explicability is not good. At this point, optimization algorithms that can work with continuous data can overcome these disadvantages. This study focuses on rule mining using the Sunflower Optimization algorithm, one of the successful optimization algorithms of recent years. For this, the candidate plant structure using a different representation format was adapted to this optimization algorithm. Data sets from three different disciplines were used as the search space, and the results of five different well-known classification algorithms were shared for performance observations. The results obtained proved that, with the optimization-based approach, explicable rules can be produced without any pre-processing on the data sets.

1. GİRİŞ

Kural madenciliği, kısaca, bir veri setinden anlamlı ve yorumlanabilir kurallar çıkarabilen veri madenciliği

dalıdır. Çıkarılan bu kurallar, kontrol sistemlerinde ve sınıflandırma temelli sistemlerde kullanılabilir. Bununla beraber, çıkarılan kuralların doğruluğu kullanılacak sistemin başarısını doğrudan etkileyeceğinden dolayı kural kapsamı metriği burada kritik öneme sahiptir. Bu

metrik özellikle verinin nümerik olduğu durumlarda hatalara karşı daha duyarlı hale gelebilmektedir. Standart sınıflandırma algoritmaları, nicel veriler üzerinde genellikle ayrıklaştırma mekanizması kullandıklarından yaptıkları sınıflandırmalarda kayıplar meydana gelebilmektedir. Sınıflandırma algoritmaları, birkaç küme halinde gruplandırılmıştır ve bu kümeler sıralı iletişim ile katmanlar halinde düzenlenmiştir. Bununla birlikte, topluluk yöntemleri tasarlamak karmaşıktır ve çok fazla hesaplama süresi gerektirir. Ayrıca, sınıflandırma yöntemlerinin çoğu yanlış tahminlerde bulunursa, bu da topluluk metotlarının iyi performans vermesini engeller [1,2]. Sınıflandırma algoritmalarının çoğu kara-kutu yaklaşımlardır. Diğer taraftan, veri kümeleri içerisinde doğru kurallar kadar, anlaşılır kuralların da çıkarılması açıklanabilir yapay zeka, makine öğrenmesi ve sınıflandırma kuralları madenciliğinde de önemlidir [3]. Ayrıca, sayısal özniteliklerden oluşan veri kümeleri içinde ilginç, anlaşılır ve doğru sınıflandırma kurallarının madenciliği daha karmaşıktır. Nicel veri kümeleri için sınıflandırma algoritmaları, bilgi kaybına neden olabilecek bir ön işlem olabilecek bir tür ayrıklaştırma gerçekleştirir. Bu durumda, veri seti değiştirilmiş olur. Böylece, keşfedilen sınıflandırma kuralları, değiştirilen veri kümesinin modeline aittir. Veri kümesini değiştirmek mantıksızdır ve ayrıca hesaplama yükü gerektirir [4]. Veri kümesini değiştirmeden sınıflandırma yöntemini uyarlamak daha mantıklıdır. Nicel öznitelikler için ilgili aralıkların bulunmasını ve yüksek kaliteli nicel sınıflandırma kurallarının çoklu amaçlarla madenciliğini yalnızca bir adımda entegre etmek, hız ve doğruluk açısından çok daha anlamlı görünmektedir. Veriler çoklu sınıflara, çoklu özelliklere sahipse ve oldukça dengesizse, sınıflandırma işlemi daha karmaşık hale gelir [5]. Olası öznitelik kombinasyonlarının sayısı çok büyük olduğunda, klasik yöntemlerin performansları düşüktür.

Kural madenciliği farklı disiplinlerde ortaya çıkan problemlerin çözümünde kullanılmaktadır. Gündoğan ve ark. [6] sınıflandırma kuralı madenciliği için bir genetik algoritma (GA) kullanmışlardır. Yöntemleri, genelleştirilmiş tek tip popülasyon yöntemi ve bundan ilham alan tek tip bir operatör kullanarak anlaşılır EĞER-İSE kurallarını keşfetmektedir. Uygulama için meme kanseri ve dermatoloji veri setlerini kullanmışlardır. Pourpanaha ve ark. [7] veri sınıflandırması ve kural çıkarımı için hibrit model önermiştir. Çalışma iki aşamadan oluşmaktadır. İlk aşama, sınıflandırma için Q-öğrenmeyi, ikinci aşama ise Q-öğrenme'den genetik algoritma ile kural çıkarımını başarmaktır. Tasarlanan yöntemin etkinliği Iris, PID, Dermatology, Glass, Sonar, Wine, Statlog (Heart) veri setleri üzerinde ispatlanmıştır. Tripathy ve ark. [8] sınıflandırma kuralı madenciliği için MTACO-Miner (Modified Threshold Ant Colony Optimization-Miner) olarak adlandırılan gelişmiş bir karınca kolonisi optimizasyon algoritması önermişlerdir. Uygulama, Wisconsin Diagnostic Meme Kanseri veri setini kullanılmış ve sonuçlar MTACO-Miner ve Ant-Miner algoritmaları ile karşılaştırılmıştır. Taboada ve ark. [9] yaptıkları çalışmada, genetik ağ programlama kullanan bulanık sınıflandırma kurallarına dayalı bir sınıflandırma

yöntemi sunmuşlardır. Genetik ağ programlamanın karakteristiğinin, önerilen modelin formüle edilmesini ve kullanılmasını kolaylaştırdığını göstermişlerdir. Önerilen algoritmanın performansları diğer ilgili algoritmalar ile karşılaştırılmış ve deneysel sonuçlar, önerilen modelin avantajlarını ve etkinliğini gösterdiğini belirtmişlerdir. Dehuri ve ark. [10], çok amaçlı sınıflandırma kuralı madenciliği için bir Genetic Expression Programming-GEP yaklaşımı önermişlerdir. Amaç olarak, sınıflandırma doğruluğu ve anlaşılabilirlik kriterleri dikkate alınmıştır. GEP yaklaşımının diğer algoritmalara göre daha verimli olduğunu ve kural ağaç tabanlı genetik programlama sınıflandırıcılarına kıyasla daha kısa çözümler üretme eğiliminde olduğunu belirtmişlerdir. Önerilen yöntemin kuralın anlaşılabilirliğini arttırdığını savunmuşlardır. GEP sınıflandırıcısı birkaç kıyaslama veri seti ile test edilmiş ve tatmin edici sonuçlar elde edildiği makalede gösterilmiştir. Zhong-Yang ve ark. [11] yaptıkları çalışmada, genetik algoritma ve tabu arama ile birleştirilmiş gelişmiş bir hibrit genetik algoritma kullanan bir sınıflandırma yöntemi kullanmışlardır. Sınıflandırma doğruluğunu artırmak ve ayrıca sınıflandırma kural kümesini yoğunlaştırmak için bir kural çıkarma yaklaşımını da vermişlerdir. Alınan sonuçlar dört kıyaslama veri seti üzerinde doğrulanmış ve diğer algoritmalarla karşılaştırılmıştır. Bu deneylerin, önerilen yöntemin iyi bir performansla sahip olduğunu ve bir dizi özlü, verimli ve anlaşılır sınıflandırma kurallarını keşfetme yeteneğine sahip olduğunu gösterdiğini belirtmişlerdir.

Optimizasyon yöntemleri, kısıtlamaları karşılayan bir dizi mevcut parametreden en iyi değerleri belirleyerek problemleri çözmek için kullanılır. Akıllı metasezgisel algoritmalar, çok farklı alanlarda uygulandığında, üretilen sonuçların sağlamlığı ve basitliği nedeniyle arama ve optimizasyon problemlerinin çözümünde kesin klasik algoritmalara göre daha fazla popülerlik kazanmıştır [12, 13]. Pek çok problemin çözümünde başarı gösteren akıllı optimizasyon algoritmaları kural madenciliğinde de etkili sonuçlar üretebilmektedir [6, 14, 15]. Bu çalışma, bitki zekası temelli optimizasyonu ile kural madenciliğine odaklanmıştır. Son yılların başarılı metasezgisel yaklaşımlarından olan Ayçiçeği Optimizasyon Algoritması (AOA) kural madenciliği için yeniden uyarlanmıştır. Bu çalışma, AOA'nın kural madenciliğine uygulanabilirliğine odaklanmaktadır. Bu algoritma, yazarların bildiği kadarıyla, daha önce kural madenciliğinde kullanılmamıştır. Bununla beraber, literatürdeki diğer uygulamalara benzer olarak, standart AOA da doğrudan kural çıkarım sistemlerine uygulanmaz. Sürekli optimizasyon problemlerine yönelik olan AOA'nın bu amaç için uyarlanmış versiyonuna ihtiyaç vardır. Bu uyarlama için kural çıkarım adaylarının temsil biçimlerinin doğru belirlenmesi gerekmektedir. Uyarlanmış AOA'da bitki adaylar için üç nitelikli bir temsil formatı kullanılmıştır. İlgili veri seti ise arama uzayı olarak kullanılmış ve aday bitkiler bu uzayda kural keşfi için modellenmiş bir amaçla uygunluk fonksiyonu önerilmiştir. Bu amaçla makalede ilk olarak standart AOA açıklanacak ve daha sonra farklı bir temsil biçimi kullanan uyarlanmış AOA versiyonu detaylandırılacaktır.

Çalışmanın bundan sonraki organizasyonu şu şekildedir; klasik AOA'nın literatür özeti ve mantıksal detayları ikinci bölümde sunulacaktır. AOA temelli yeni sınıflandırma kural keşif yaklaşımı üçüncü bölümde sunulacaktır. Performans testleri ve elde edilen sonuçlar dördüncü bölümde, sonuç değerlendirmeleri ise beşinci bölümde paylaşılacaktır.

2. AYÇİÇEĞİ OPTİMİZASYON ALGORİTMASI (AOA)

Yapılan çalışmanın detaylarına geçmeden önce başarılı bir bitki zekası algoritması olan AOA hakkında detay bilgilerin verilmesi faydalı olacaktır. Ayçiçeği algoritmasının tercih edilmesinin iki önemli sebebi bulunmaktadır. Birincisi, kural çıkarım problemlerinde bitki zekası temelli yaklaşımlar bilindiği kadarıyla önceden çalışılmamıştır. Bu amaçla bu tip algoritmaların performansları ölçebilecek bir çalışmaya bu alanda ihtiyaç vardır. İkincisi dahili döngü prensibi kullanan AOA'nın yüksek performansı diğer bitki zekası temelli yaklaşımlara göre avantaj sağlamaktır. Bu nedenlerden dolayı AOA bu problemin çözümü için seçilmiştir. Bunun için bu bölümde öncelikle bu algoritmaya ait literatür özeti verilerek çalışma mekanizmasına ait detay bilgiler paylaşılacaktır. Daha sonra, kural çıkarım için uyarlanmış AOA versiyonu detaylandırılacaktır.

AOA'nda, ayçiçeği bitkilerinin güneşe yönelim hareketlerinden esinlenilmiştir. Algoritmada temel prensip, güneş olarak adlandırılan en iyi çözüme ulaşmaktır. Literatürde yeni olmasına karşın bazı problemlerin çözümünde başarılı sonuçlar üretmeyi başarmıştır. Qais ve ark. [16] yaptıkları çalışmada bazı parametre seçimleri için AOA'nı kullanarak fotovoltaiik modüllerin modellenmesi ve simülasyonunu gerçekleştirmişlerdir. Standart AO parametrelerini kullanarak bile başarılı sonuçlar aldıklarını kaydetmişlerdir. Gomes ve ark. [17], plaka benzeri yapıların hasar tespiti için bir optimizasyon yaklaşımı önermiş ve önerilen yöntemin verimliliğini; genetik algoritma, AOA ve geliştirilmiş bir AOA'nı kullanarak karşılaştırmalı olarak sunmuşlardır. Bu sonuçlara bağlı olarak önerilen yaklaşımın genetik algoritmalara göre daha iyi sonuçlar verdiği iddia edilmiştir. Yuan ve ark. [18], proton değişim membranlı yakıt hücresi (PEMFC - Proton Exchange Membrane Fuel Cell) modellerinde parametrelerin optimum seçimi için yeni bir metodoloji önererek, PEMFC yığınının tahmini ve gerçek çıkış voltajı arasındaki hata karesi değerinin toplamını en aza indirmek için geliştirilmiş bir AOA sunmuşlardır. Geliştirilen AOA'nda, kendi kendine uyarlanabilir ağırlıklandırma kullanılarak en iyi çözüme ulaşma hedeflenmiştir. Hussein ve ark. [19] ise yaptıkları çalışmada, AOA'nı Proportional Integral (PI) kontrolör parametrelerini seçmek için kullanmışlardır. Elde ettikleri sonuçları Parçacık Sürü Optimizasyon algoritması sonuçları ile kıyaslamışlardır. Shaheen ve diğ. [20], güç sistemleri alanında optimum güç akışı problemini çözmek için AOA'nı kullanan yeni bir girişim sunmuşlardır. Parçacık sürü optimizasyonu ve genetik algoritmaları ile karşılaştırmalı olarak sunulan

AO Algoritması sonuçlarının daha başarılı olduğu bu çalışmada gösterilmiştir. Alshammari ve ark. [21] ise Logistic kaotik haritalama metodunu kullanan bir AOA geliştirerek, bunu güç sistemi stabilizatörlerinin optimal ayarını yapmak için kullanmışlardır. Farklı test fonksiyonlarını kullanarak, kaotik AOA'nın, kullanılan diğer algoritmalarından daha iyi performans gösterdiğini belirtmişlerdir.

Ayçiçeği Optimizasyon algoritmasının temelinde, ayçiçeklerinin güneş ışınlarını takip etme hareketleri yatar [22]. Ayçiçeklerin (adayların) hedefi, güneşin bulunduğu yöne doğru yönelip en iyi pozisyonu bulmaktır. AOA mekanizmasında, adaylar, eğer güneşe yakın iseler buldukları yeri korumak ve eğer uzak iseler daha büyük yönelimler gösterip güneşe yaklaşmak isterler. Lokasyonlarını güncellemenin yanında, komşu olan ayçiçekleri ile tozlaşarak yeni bireyler de üretebilirler. Gerçek dünyada milyonlarca tozlaşma oluşurken, algoritmada kolaylık olması için her ayçiçeğinin bir tozlaşma gerçekleştirdiği varsayılmıştır. AOA'da, ilk aşama, başlangıç popülasyonu oluşturmaktır. Başlangıç popülasyonu oluşturulduktan sonra çözüme en yakın birey, güneş olarak adlandırılır. Güneş belirlendikten sonra, güneşin pozisyonuna göre diğer bireylerden komşu olanlar, kendi aralarında tozlaşarak yeni bireyler oluştururlar. Yeni bireyler üretildikçe arama uzayının boyutu artar ve bunun engellenmesi için belli bir oranda ayçiçeği bireyleri elenir. Gerekli güncelleme ve yeni birey üretiminden sonra bu süreçler, sonlandırma kriteri sağlanana kadar iteratif şekilde devam ettirilir.

Algoritmada önemli prensiplerden biri ayçiçeklerinin güneşe yöneliminin temsildir. Popülasyondaki anlık en iyi çözüm olan güneş (X^*), diğer ayçiçeği bireylerine (X_i) referans olur. n_p sayıda birey içeren popülasyonun güneşe yönelimi s_i ile gösterilir ve genel ifadesi Denklem 1'deki gibidir.

$$s_i = \frac{X^* - X_i}{\|X^* - X_i\|}, \quad i = 1, 2, \dots, n_p \quad (1)$$

Algoritmanın en iyiye ilerleyişi yönelim olarak adlandırılır ve Denklem 2 ile ifade edilir. Yönelim, iki temel parametreye dayalıdır. Bunlar λ eylemsizlik katsayısı ve i . birey ve $(i-1)$. birey arasındaki tozlaşma olasılığını ifade eden P_i 'dir. Genel ifadeden de görüleceği üzere güneşe yakın bireyler küçük adımlarla güneşe yönelirken, daha uzak bireyler daha büyük adımlarla yönelim sağlamaya çalışacaklardır.

$$d_i = \lambda P_i (\|X_i + X_{i-1}\|) \|X_i + X_{i-1}\| \quad (2)$$

Klasik AOA'nda her iterasyonda güneşe uzak olan bazı bireyler, popülasyondan çıkarılma oranına (O) göre arama sürecinden çıkarılmakta ve yerine yeni bireyler üretilmektedir. Bu yeni bireyler, her iterasyonda popülasyona katılarak farklı keşif noktalarının bulunmasına yardımcı olmaktadır. Bunun dışında diğer bireylerin yeni konumlarının belirlenmesinde, global minimum adaylarını kaçırmamak için maksimum yönelim miktarı değerine de dikkat etmek gerekir. Bu

temel prensip, Denklem 3 ile sağlanmaktadır. Bu denklemde, X_{max} ve X_{min} üst ve alt sınır değerlerini, N_{pop} ise popülasyondaki toplam birey sayısını belirtmektedir. Yeni bireye ait değer hesaplanmasında, Denklem 4 kullanılmaktadır. AOA'nın bu temel adımları ve mantıksal fonksiyonları, Algoritma 1'de paylaşılan sözde kodda görülebilir.

$$d_{max} = \frac{\|X_{max} - X_{min}\|}{2N_{pop}} \quad (3)$$

$$\vec{X}_{i+1} = \vec{X}_i + d_i \vec{s}_i \quad (4)$$

Algoritma 1. Ayçiçeği Optimizasyon Algoritması sözde kodu

1. $X_{max}, X_{min} \leftarrow$ problem tanımlı üst sınır ve alt sınırlar
2. $Itr_{max} \leftarrow$ maksimum iterasyon sayısı
3. $N_{pop} \leftarrow$ popülasyon boyutu
4. $O \leftarrow$ Popülasyondan çıkarma oranı (%)
5. for $i=1$ 'den N_{pop}
6. $X_i =$ BireyÜreteçFonksiyonu(X_{max}, X_{min})
7. PopulasyonaEkle(X_i)
8. end for
9. AmaçDeğerHesapla ()
10. $X^* \leftarrow$ GüneşBul()
11. GüneşeYonlendir()
12. while($i < Itr_{max}$)
13. for $j=1$ 'den N_{pop}
14. BireyVektörHesapla()
15. UzakBireyleriÇıkar(% O , i)
16. YeniBireyleriDeğerlendir(X^*)
17. end for
18. AmaçDeğerHesapla()
19. $X^* \leftarrow$ GüneşBul()
20. End while
21. EniyiyiGöster()

3. AOA TABANLI OTOMATİK SINIFLANDIRMA KURAL MADENCİLİĞİ SİSTEMİ

Bu çalışma, yazarların bildiği kadarıyla, daha önce kural madenciliğinde kullanılmayan AOA'nın kural madenciliğine uygulanabilirliğine odaklanmaktadır. Bununla beraber, literatürdeki diğer uygulamalara benzer olarak, standart AOA da doğrudan kural çıkarım sistemlerine uygulanmaz ve sürekli optimizasyon problemlerine yönelik olan AOA'nın bu amaç için uyarlanmış versiyonuna ihtiyaç olduğu düşünülmektedir. Bu uyarlama için kural çıkarım adaylarının temsil biçimlerinin doğru belirlenmesi gerekmektedir.

Sınıflandırma kural madenciliği için kullanılan birçok yöntem ve algoritma vardır [23, 24]. Bu sınıflandırma algoritmalarının ve modellerinin çoğu kara-kutu tabanlı yaklaşımlardır. Ancak, anlaşılabilir kurallar, veri setlerinde olduğu kadar, doğru kurallara ulaşılmasını hedefleyen açıklanabilir yapay zeka veri madenciliğinde de oldukça önemlidir [3]. Ayrıca, sayısal özelliklerden oluşan veri setinde ilginç, anlaşılabilir ve doğru sınıflandırma kurallarının madenciliği daha karmaşıktır. Nicel veriler için sınıflandırma algoritmaları, bilgi kaybına neden olabilecek bir ön işlem yaparak bir tür ayrıklaştırma gerçekleştirir. Bu durumda, veri madenciliğinde veri seti problemi değiştirilir. Bu yüzden; keşfedilen sınıflandırma kuralları değiştirilmiş veri setinin modeline ait olur. Veri setini değiştirmek

hataya açıktır ve ayrıca hesaplama yükü gerektirir [4]. Veri seti değiştirildiğinde asıl çözümü istenilen sınıflandırma probleminin doğruluğu da azalacaktır. Bu sebeple, veri seti değiştirmeden sınıflandırma yöntemini uyarlamak daha mantıklıdır [14]. Nicel özellikler için ilgili aralıkları bulmanın ve yüksek kaliteli nicel sınıflandırma kurallarının madenciliğinin birden fazla hedefle tek bir adımda birleştirilmesi hız ve doğruluk açısından çok anlamlı görünmektedir.

Bu çalışma, nicel kural çıkarım problemlerinde, n bireylik popülasyondaki her çözüm için farklı bir temsil biçimini esas almıştır [14]. Bu temsil biçiminde, d (nitelik sayısı) boyutlu bir X_i çözümü, X_i^b, X_i^a ve X_i^u ($i=1, 2, \dots, n$) olmak üzere üç adet alt veri içermektedir. $X_i^b = (x_1^b, x_2^b, \dots, x_j^b, \dots, x_d^b)$, bir çözümün her bir niteliği için ikili bir veri tutar. Bir x_j niteliğinin, çözümün kuralına eklenip eklenmeyeceği x_j^b değeri ile belirlenir. Başlangıç popülasyonunda x_j^b , $t(\cdot)$ rastgele üreteç fonksiyonu ve daha önceden tanımlanmış bir " δ " eşik değeri ile belirlenir. Bu işlem Denklem 5 ile hesaplanmıştır.

$$x_j = \begin{cases} 1 & (\text{eğer } x_j \text{ kuralda kullanılmışsa}) \\ 0 & (\text{eğer } x_j \text{ kuralda kullanılmamışsa}) \end{cases} \quad (5)$$

$X_i^a = (x_1^a, x_2^a, \dots, x_j^a, \dots, x_d^a)$, bir çözüme ait kuralın her niteliği için hesaplanan alt limitleri gösterirken, $X_i^u = (x_1^u, x_2^u, \dots, x_j^u, \dots, x_d^u)$ ise ilgili çözüm kuralındaki niteliklerin üst limitlerini gösterir ($x_d^a, x_d^u \in R$). Arama uzayında alt değeri N^a ve üst değeri N^u olan " j " niteliği için, " $N^a \leq x_j^a < x_j^u \leq N^u$ " koşulu her iterasyonda her zaman geçerlidir. İterasyonlar süresince her bir aday kendine ait X_i^b, X_i^a ve X_i^u değerlerini yeniden hesaplar. İterasyonların sonunda $x_j^b > \delta$ şartını sağlayan her nitelik " X_i " çözümünün kuralına eklenir. Veri setindeki her sınıf için kural madenciliği optimizasyonu ayrı ayrı gerçekleştirilmiştir. Bu, ana veri setinin (arama uzayının) her bir sınıf için alt uzaylara bölündüğü ve her optimizasyon işleminin ilgili alt uzayda gerçekleştiği anlamına gelmektedir. Bir çözümün X_i^b, X_i^a , ve X_i^u değerleri, ilgili arama uzayındaki her veri için kontrol edilir ve amaç fonksiyonunun uygunluk değerini hesaplamak için kullanılır. Örneğin, "S" sınıfı için gerçekleştirilen bir kural madenciliği optimizasyon işleminde, bir çözümün kendi kuralında 1., 3. ve 5. nitelikleri kullandığını varsayalım (x_1^b, x_3^b , ve $x_5^b > \delta, i$. iterasyonda). Bu durumda, aday çözümün " k " verisi ($X_{k_1}, X_{k_2}, \dots, X_{k_m}$), için kontrol edilmesi Denklem 6'ya göre gerçekleştirilecektir.

$$\text{eğer } x_1^a \leq X_{k_1} \leq x_1^u \text{ ve } x_3^a \leq X_{k_3} \leq x_3^u \text{ ve } x_5^a \leq X_{k_5} \leq x_5^u \quad (6)$$

Bu hesaplama işlemi, amacın uygunluk değerini hesaplamak için kullanılmaktadır. Çalışmada, amaç fonksiyonu olarak sınıflandırmanın "doğruluk" değeri alınmıştır. X_i çözümünün "doğruluk" değerini bulmak için ilgili "S" sınıfına ait her veri için Denklem 6'daki koşul tek tek kontrol edilir. Bu işlemlerin sonucunda

doğru pozitif (DP), doğru negatif (DN), yanlış pozitif (YP) ve yanlış negatif (YN) değerleri elde edilir. Bunun için, Tablo 1’de gösterildiği gibi, ilgili kuralın “eğer” kısmı (İSE öncesi) ve “ise” kısmı (İSE sonrası) ayrı ayrı değerlendirilir. Daha sonra, amaç fonksiyonu olarak kullanılan “doğruluk” değeri, ilgili kuralın DP, DN, YP ve YN değerleri baz alınarak Denklem 7’deki gibi hesaplanır.

Tablo 1. DP, DN, YP ve YN değerlerinin hesaplanması

“eğer” kısmı	“ise” kısmı	İşlem
Doğru	Doğru	DP’yi 1 artır
Yanlış	Yanlış	DN’yi 1 artır
Doğru	Yanlış	YP’yi 1 artır
Yanlış	Doğru	YN’yi 1 artır

$$\text{Doğruluk} = \frac{DP + DN}{DP + DN + YP + YN} \quad (7)$$

DP; kuralın hem sol hem de sağ tarafının doğru olduğu durum sayısı, DN; kuralın her iki tarafının yanlış olduğu durum sayısı, YP; kuralın sol tarafının doğru, sağ tarafının yanlış olduğu durum sayısı, YN ise kuralın sol tarafının yanlış, sağ tarafının doğru olduğu durum sayısıdır. Algoritma 2’de verilen sözde kod, kural çıkarım tabanlı Ayçiçeği Optimizasyon Algoritmasında yukarıda açıklanan süreçlerin işleyişini göstermektedir. Algoritmanın işleyişi iki aşamada gerçekleşmektedir. Birincisi eğitim aşamasıdır. Bu aşamada eğitim veri setleri için uygunluk değerine göre uyarlanmış AOA çalıştırılır. Algoritma iterasyonları sonunda elde edilen kurallar test verileri için tekrar denenmektedir. Aynı değerlendirme süreçleri bu kez test verileri için yapılır. Elde edilen sonuçlar test verileri ile yapılan deneylere aittir.

Algoritma 2. Uyarlanmış AOA sözde kodu

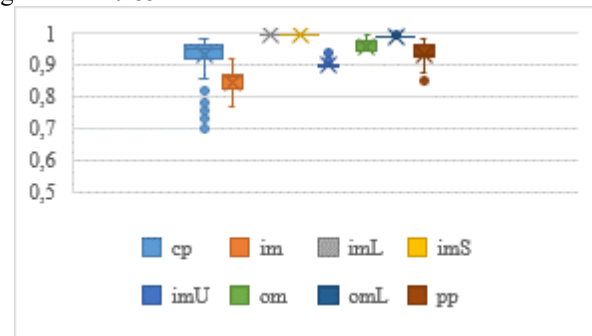
1. Veri setini al
2. *Aday sınıflandırma kural setini gösteren kodlanmış popülasyonu başlat (X_i için X_i^b, X_i^a ve X_i^u ’yu başlat)*
3. $Itr_{max} \leftarrow$ maksimum iterasyon sayısı
4. $N_{pop} \leftarrow$ popülasyon boyutu
5. $O \leftarrow$ Popülasyondan çıkarma oranı (%)
6. for $i=1$ ’den N_{pop}
7. $X_i =$ BireyÜreteçFonksiyonu(X_i^b, X_i^a, X_i^u)
8. PopülasyonaEkle(X_i)
9. end for
10. VarsayılanDeğerlereAyarla(DP, DN, YP, YN)
11. AmaçDeğerHesapla()
12. $X^* \leftarrow$ GüneşKuralıBul()
13. while($i < Itr_{max}$)
14. GüneşYonlendir()
15. for $j=1$ ’den N_{pop}
16. BireyVektörHesapla()
17. UzakBireyleriÇıkar(% O, i)
18. YeniBireyleriDeğerlendir(X^*)
19. end for
20. VarsayılanDeğerlereAyarla(DP, DN, YP, YN)
21. AmaçDeğerHesapla()
22. $X^* \leftarrow$ GüneşKuralıBul()
23. End while
24. *testVerileriİçinKurallarıDene()*
25. EniyiKuralıGöster()

4. DENEYLER

Çalışmada, literatürde iyi bilinen üç farklı veri tabanı ile deneyler gerçekleştirilmiştir. Bunlar Ecoli, Iris ve Glass veri setleridir. Amaç fonksiyonu olarak doğruluk

kullanılmış ve uygulama platformu olarak Python tercih edilmiştir. Bununla birlikte, elde edilen sonuçların daha sağlıklı değerlendirilmesi için literatürde iyi bilinen sınıflandırma algoritmalarına ait sonuçlar kullanılmıştır. Kullanılan sınıflandırma algoritmaları karar ağacı, k-NN, JRIP, Naive Bayes ve Destek Vektör Makinesi (DVM)’dir. Bu veri setlerinin %66’sı eğitim, %34’ü ise test işlemleri için ayrılmıştır. Ayrırma işlemi, her bir sınıfın ağırlık oranı dikkate alınarak yapılmıştır. Böylece ana veri setindeki örnek sayılarına göre sınıf ağırlık oranları hem eğitim hem de test veri setlerinde korunmuştur. Kullanılan veri setlerine ait özellikler Tablo 2’deki gibidir. Deneylerde ilk olarak her bir sınıf için eğitim iterasyonları gerçekleştirilmiş daha sonra test süreçleri tamamlanmıştır. İlgili veri seti için algoritmanın doğruluk değeri, içerdiği tüm sınıfların ağırlıklı oranları ile elde edilmiştir. Her bir veri setine ait sınıf ağırlıkları bir sınıfın ilgili veri setindeki örnek sayısına göre belirlenmiştir. Karşılaştırmalı sonuçlar için ilgili veri setleri aynı eğitim ve test veri seti ayırma oranları korunarak standart sınıflandırma algoritmaları için denenmiştir.

Kural çıkarım tabanlı AOA ile yapılan deneylerde veri setlerindeki her bir sınıf için yirmi ayrı deney gerçekleştirilmiştir. Böylece toplamda 340 deney yapılmıştır. İlk deney Ecoli veri seti için gerçekleştirilmiştir. Ecoli veri seti için karşılaştırmalı doğruluk değeri sonuçları Tablo 3’te görülmektedir. Tablo 4’te bu veri seti ile yapılan deneylerden birine ait elde edilen doğruluk değerleri ve çıkarılan kurallar paylaşılmıştır. Bu veri seti için yapılan tüm deneylere ait istatistiksel sonuçlar ise Tablo 5’te verilmiştir. Ecoli veri setinde elde edilen sonuçlar ve sınıf ağırlık değerlerine göre ortalama doğruluk değeri 0,945 olmuştur. Buna göre standart sınıflandırma algoritmaları sonuçları ile karşılaştırıldığında kural çıkarım temelli AOA en başarılı sonucu vermeyi başarmıştır. Standart sınıflandırma algoritmaları içinde en başarılı sonuç Naive Bayes en düşük sonuç ise Karar Ağacı tarafından elde edilmiştir. Tablo 4 ve 5’ten görüleceği gibi AOA başarımı düşük yoğunluk değerli sınıflarda daha yüksektir. Yüksek yoğunluk sınıflarda ise ortalama başarımların 0,850’nin altına inmemiştir. Standart sınıflar içerisinde en yüksek doğruluk değeri sırasıyla Naive Bayes ve DVM tarafından elde edilmiştir. En düşük başarımlar ise karar ağacına aittir. Bu veri seti ile yapılan deneylere ait sonuçların dağılımı Şekil 1’de verilmiştir. Şekilden de görüleceği üzere neredeyse tüm sınıflarda başarımlar genellikle 0.85’in üzerindedir.



Şekil 1. Ecoli veri seti deneyleri sonuç dağılımları için çeyrekler açıklığı grafikleri

Tablo 2. Veri Setleri

Data seti	Alanı	Örnek Sayısı	Sınıflar	Sınıflardaki Örnek Sayısı	Nitelikler	Min. Değer	Maks. Değer
Ecoli	Biyoinformatik	336	cp	143	mcg	0	0,89
			im	77	gvh	0,16	1
			pp	52	lip	0,48	1
			imU	35	chg	0,5	1
			om	20	aac	0	0,88
			omL	5	alm1	0,03	1
			imL	2	alm2	0	0,99
			imS	2			
Glass	Fizik	214			Ri	1,51115	1,53393
					Na	10,73	17,38
			1	70	Mg	0,0	4,49
			2	76	Al	0,29	3,5
			3	17	Si	69,81	75,41
			5	13	K	0,0	6,21
			6	9	Ca	5,43	16,19
			7	29	Ba	0,0	3,15
Iris	Yaşam	150	Setosa	50	sepal_uzunluk	4,3	7,9
			Versicolor	50	sepal_genişlik	2,0	4,4
			Virginica	50	petal_uzunluk	1,0	6,9
					petal_genişlik	0,1	2,5

Tablo 3. Ecoli veri seti için karşılaştırmalı “doğruluk” değeri sonuçları

Data Seti	Karar Ağacı	k-NN	JRIP	Naive Bayes	DVM	AoA
Ecoli	0,767	0,803	0,806	0,854	0,842	0,945

Tablo4.Ecoli veri seti kuralları

Sınıf	Doğruluk	Kural
cp	0,936	EĞER 0,160 < gvh <0,581 VE 0,030 < alm1 <0,521 İSE cp
im	0,919	EĞER 0,000 < mcg <0,633 VE 0,596 < alm1 <0,940 İSE im
imS	0,995	EĞER 0,600 < mcg <0,879 VE 0,467 < gvh <0,892 VE 0,744 < lip <0,974 VE 0,303 < aac <0,541 İSE imS
imL	1	EĞER 0,327 < mcg <0,880 VE 0,169 < gvh <0,752 VE 0,861 < lip <1,000 VE 0,426 < aac <0,527 VE 0,711 < alm2 <0,938 İSE imL
imU	0,946	EĞER 0,743 < mcg <0,889 VE 0,001 < aac <0,876 VE 0,442 < alm2 <0,939 İSE imU
om	0,995	EĞER 0,544 < mcg <0,802 VE 0,637 < aac <0,880 VE 0,142 < alm2 <0,604 İSE om
omL	1	EĞER 0,879 < lip <1,000 VE 0,165 < alm2 <0,502 İSE omL
pp	0,982	EĞER 0,567 < mcg <0,831 VE 0,561 < gvh <0,866 VE 0,077 < aac <0,634 İSE pp

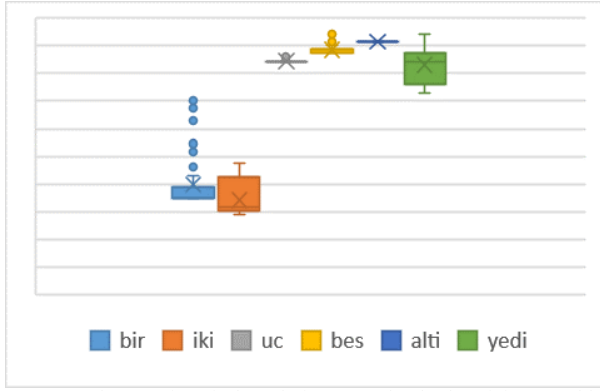
Tablo5. Ecoli veri seti için istatistiksel sonuçlar

	cp	im	imS	imL	imU	om	omL	pp
Min	0,701	0,769	0,995	0,995	0,895	0,941	0,986	0,846
Maks	0,936	0,919	0,995	1,000	0,946	0,995	1,000	0,982
Ortalama	0,901	0,850	0,995	0,995	0,901	0,958	0,990	0,935
Ortanca	0,920	0,860	0,995	0,995	0,896	0,950	0,986	0,932
Std.Sapma	0,052	0,042	2,66454E-14	0,0003	0,009	0,017	0,005	0,031

Kriminal soruşturma çalışmalarından elde edilen bir veri seti olan Glass, çalışılan diğer bir arama uzayıdır. Toplamda altı sınıflı bu veri seti için yapılan deneylerden birinde elde edilen doğruluk değerleri ve bu değerleri sağlayan kurallar Tablo 6 ve 7’de paylaşılmıştır. Tablo 8 ise bu veri seti ile yapılan, toplamda 120 farklı deneye ait istatistiksel sonuçları göstermektedir. Tablo 7’deki deneye ait sonuçlar dikkate alındığında, sınıf ağırlık değerlerine göre ortalama doğruluk değerinin 0,676 olduğu görülmektedir. Bu değer, diğer standart sınıflandırma algoritma sonuçlarına göre en iyi dördüncü değer olmuştur. Bu veri seti için en iyi sonucu k-NN algoritması elde etmiştir. En kötü sonuç ise Naive Bayes’e aittir. Genel başarımlar olarak uyarlanmış AOA yöntemi her ne kadar vasat bir görüntü verse de kendisinden daha iyi sonuç veren algoritmalara yakın değerler üretebilmiştir. Sınıf bazında incelendiğinde bu veri setinde başarımları düşüren sınıf aslında “üç” sınıfıdır. Bu sınıfa ait başarılı bir kural çıkarımı tam olarak elde edilememiştir. Bununla beraber son üç sınıfta başarımlar

değerleri yüksektir. Öte yandan bu sınıflarda yoğunluk katsayılarının düşük olması nedeniyle başarımları çok fazla etki etmemiştir. İstatistiksel sonuçları oluşturan deney sonuç dağılımları Şekil 2’de verilmiştir. Sadece bir sınıfı için ayrıntı değerlerin olduğu deneylerde üç ve üstü sınıflarda açıklık düşük ve kararlılık yüksektir.

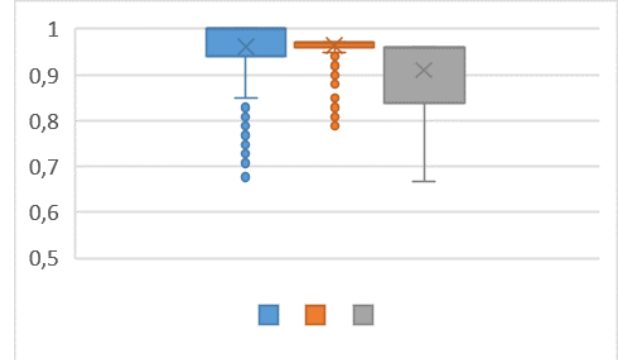
Her biri elli örnek içeren üç sınıflı Iris, literatürde iyi bilinen bir veri seti olup son deneyler bu veri seti ile gerçekleştirilmiştir. Bu veri seti ile toplamda 60 farklı deney yapılmıştır. Tablo 9 ve 10 bu deneylerden birine ait sonuçları ve diğer algoritmalara ait sonuçları karşılaştırmalı göstermektedir. Tüm deneyler için elde edilen istatistiksel sonuçlar ise Tablo 11’de paylaşılmıştır.



Şekil 2. Glass veri seti deneyleri sonuç dağılımları için çeyrekler açıklığı grafikleri

Kural çıkarım temelli AOA en iyi sonuçlardan birini bu veri setinde elde etmiştir. Standart sınıflandırma algoritmalarında en iyi sonuç 0,966 ile JRIP tarafından elde edilmişken, uyarlanmış AOA bu veri setinde tüm sınıflar için 0,977 ağırlıklı ortalama değerine ulaşmayı başarmıştır. Sınıf bazında değerlendirildiğinde tüm sınıfların yoğunluk katsayıları aynı olduğu için ağırlıklı ortalama etkileri aynı olmuştur. Setosa sınıfında başarımlar %100 oranına ulaşmayı başarmıştır. Diğer

iki sınıfta da ortalama başarımlar 0,965 olmuştur. Karar ağacı algoritması, Ecoli veri setinde olduğu gibi en düşük başarımlar değerine sahip algoritma olmuştur. Şekil 3'te verilen sonuç grafikleri incelendiğinde her ne kadar Setosa ve Versicolor'da ayrı değerler olsa da başarımlar genellikle yüksek olduğu görülmektedir.



Şekil 3. Iris veri seti deneyleri sonuç dağılımları için çeyrekler açıklığı grafikleri

Tablo 6. Glass veri seti için karşılaştırmalı "doğruluk" değeri sonuçları

Data Seti	Karar Ağacı	k-NN	JRIP	Naive Bayes	DVM	AoA
Glass	0,700	0,705	0,696	0,495	0,574	0,676

Tablo7. Glass veri seti kuralları

Sınıf	Doğruluk değeri	Kural
bir	0,651	EĞER 1,517 < RI < 1,530 VE 2,628 < Mg < 3,758 VE 7,550 < Ca < 12,281 İSE bir
iki	0,580	EĞER 11,484 < Na < 15,111 VE 1,408 < Al < 2,680 VE 6,717 < Ca < 8,360 İSE iki
üç	0,322	EĞER 70,315 < Si < 73,056 VE 3,654 < K < 4,765 VE 6,607 < Ca < 6,923 İSE üç
beş	0,972	EĞER 1,021 < Al < 2,546 VE 71,616 < Si < 74,376 VE 11,317 < Ca < 12,920 İSE beş
altı	0,957	EĞER 12,693 < Na < 15,404 VE 2,374 < Mg < 2,580 VE 71,459 < Si < 72,593 İSE altı
yedi	0,972	EĞER 0,274 < Ba < 2,058 İSE yedi

Tablo8. Glass veri seti için istatistiksel sonuçlar

	bir	iki	üç	beş	altı	yedi
Min	0,574	0,505	0,122	0,936	0,957	0,865
Maks	0,651	0,580	0,322	0,972	0,957	0,972
Ortalama	0,605	0,554	0,221	0,942	0,957	0,916
Ortanca	0,604	0,559	0,198	0,943	0,957	0,922
Std.Sapma	0,045	0,030	0,004	0,007	5,88418E-15	0,032

Tablo 9. Iris veri seti için karşılaştırmalı "doğruluk" değeri sonuçları

Data Seti	Karar Ağacı	k-NN	JRIP	Naive Bayes	DVM	AoA
Iris	0,926	0,953	0,966	0,96	0,960	0,977

Tablo 10. Iris veri seti kuralları

Sınıf	Doğruluk değeri	Kural
Setosa	1	EĞER 1,000 < petal_length < 2,126 İSE setosa
Versicolor	0,970	EĞER 2,856 < petal_length < 5,109 VE 0,811 < petal_width < 1,786 İSE versicolor
Virginica	0,960	EĞER 1,649 < petal_width < 2,500 İSE virginica

Tablo 11. Iris veri seti için istatistiksel sonuçlar

	setosa	versicolor	virginica
Min	0,677	0,788	0,667
Maks	1	0,970	0,960
Ortalama	0,961	0,963	0,914
Ortanca	1	0,970	0,960
Std.Sapma	0,060	0,014	0,070

5. SONUÇLAR

Ayrıklaştırma safhasından dolayı standart kural çıkarım tekniklerinde başarımlar olumsuz etkilenebilmektedir. Optimizasyon temelli yaklaşımların sürekli verileri üzerinde ayrıklaştırma yapmadan çalışabilmesi onlara bu konuda önemli avantajlar sağlayabilir. Bu amaçla bu çalışmada, optimizasyon temelli kural çıkarımına odaklanılmıştır. Çalışma, son yılların başarılı bitki zekası temelli meta sezgisel yaklaşımlarından olan Ayçiçeği Optimizasyon Algoritmasını (AOA) kullanmıştır. Bu amaçla bu problem için uygun temsil biçimi ve uygunluk fonksiyonu belirlenmiştir. Önerilen yaklaşım üç farklı disiplinde toplanmış olan veri seti için test edilmiş ve sonuçları literatürde iyi bilinen standart sınıflandırma algoritmaları ile karşılaştırılmıştır. Toplamda 340 ayrı deney gerçekleştirilmiştir. Başarım metriği olarak doğruluk kullanılmıştır. Her bir sınıf için gerçekleştirilen deneyler sonucunda, ilgili veri seti için sonuç doğruluk değeri sınıf yoğunluk katsayılarına göre hesaplanmıştır. Yapılan deneylerde, iki veri setinde en iyi sonuç, önerilen AOA temelli yaklaşımla elde edilmiştir. Elde edilen kurallar ve yapılan tüm deneylere ait istatistiksel sonuçlar önerilen uyarlanmış AOA yaklaşımının başarısını ispatlamıştır. Öte yandan kullanılan standart sınıflandırma algoritmaları içinde Naive Bayes ve DVM daha başarılı olurken Karar Ağacı iki veri setinde en düşük sonuçlara sahip olmuştur. Yazarlar bundan sonraki çalışmalarında çok amaçlı ve hibrid kural çıkarım çalışmalarına odaklanacaktır.

KAYNAKLAR

- [1] Savargiv M, Masoumi B, Keyvanpour MR. A new ensemble learning method based on learning automata. *Journal of Ambient Intelligence and Humanized Computing*.2020; 1-16.
- [2] Liu J, Chi Y, Liu Z, He S. Ensemble multi-objective evolutionary algorithm for gene regulatory network reconstruction based on fuzzy cognitive maps. *CAAI Transactions on Intelligence Technology*. 2019; 4(1): 24–12.
- [3] He C, Ma M, Wang P. Extract Interpretability-Accuracy balanced Rules from Artificial Neural Networks: A Review. *Neurocomputing*. 2020; 387(C):346-12.
- [4] Kiziloluk S, Alatas B. Automatic mining of numerical classification rules with parliamentary optimization algorithm. *Advances in Electrical and Computer Engineering*. 2015; 15(4): 17-8.
- [5] Phoungphol P, Zhang Y, Zhao Y. Robust multiclass classification for learning from imbalanced biomedical data. *Tsinghua Science and Technology*. 2012; 17(6): 619-9.
- [6] Gündoğan KK, Alataş B, Karci A. Mining Classification Rules by Using Genetic Algorithms with Nonrandom Initial Population and Uniform Operator. *Turk J Elec Engin*. 2004;12(1): 43-9.
- [7] Pourpanaha F, Limb CP, Saleha JM. A hybrid model of fuzzy ARTMAP and genetic algorithm for data classification and rule extraction. *Expert Systems with Applications*. 2016;49:74-11.
- [8] Tripathy S, Hota S, Satapathy P. MTACO-Miner: Modified Threshold Ant Colony Optimization Miner for Classification Rule Mining. *Emerging Research in Computing, Information, Communication and Applications*. Elsevier; 2013.p.1-5.
- [9] Taboada K, Mabu S, Gonzales E, Shimada K, Hirasawa K. Fuzzy Classification Rule Mining Based on Genetic Network Programming Algorithm. *IEEE Conference on Systems, Man, and Cybernetics*. USA: 2009. p. 3960-6.
- [10] Dehuri S, Cho S. Multi-objective Classification Rule Mining Using Gene Expression Programming. *Third International Conference on Convergence and Hybrid Information Technology*. Korea:2008. p. 755-7.
- [11] Zhong-Yang X, Lei Z, Yu-Fang Z. A Classification Rule Mining Method Using Hybrid Genetic Algorithms. *IEEE Region 10 Conference Tencon*. Thailand: 2004. p.207-4.
- [12] Ghobaei-Arani M, Sourı A, Safara F, Norouzi M. An efficient task scheduling approach using moth-flame optimization algorithm for cyber-physical system applications in fog computing. *Transactions on Emerging Telecommunications Technologies*. 2019; 31(1):1-14.
- [13] Safara F, Mohammed AS, Potrus MY, Ali S, Tho QT, Sourı A, et al. An Author Gender Detection Method Using Whale Optimization Algorithm and Artificial Neural Network. *IEEE Access*.2020; 8:48428-10.
- [14] Yildirim G, Alatas B. (2021), New Adaptive Intelligent Grey Wolf Optimizer based Multi-Objective Quantitative Classification Rules Mining Approaches. *Journal of Ambient Intelligence and Humanized Computing*. 2021; <https://doi.org/10.1007/s12652-020-02701-9>.
- [15] Akyol S, Alataş B. Plant intelligence based metaheuristic optimization algorithms. *Artificial Intelligence Review*. 2017;47:417–45.
- [16] Qais MH, Hasanien HM, Alghuwainem S. Identification of electrical parameters for three-diode photovoltaic model using analytical and sunflower optimization algorithm. *Applied Energy*. 2019;250:109-8.
- [17] Gomes GF, Almeida FA. Tuning metaheuristic algorithms using mixture design:Application of sunflower optimization for structural damage identification. *Advances in Engineering Software*. 2020;149:102877
- [18] Yuan Z, Wang W, Wang H, Razmjoooy N. A new technique for optimal estimation of the circuit-based PEMFCs using developed Sunflower Optimization Algorithm. *Energy Reports*. 2020; 6: 662-9.
- [19] Hussien AM, Hasanien HM, Mekhamer SF. Sunflower optimization algorithm-based optimal PI control for enhancing the performance of an autonomous operation of a microgrid. *Ain Shams Engineering Journal*. 2021; 12(2):1883-10.
- [20] Shaheen MAM, Hasanien HM, Mekhamer SF, Talaat HEA. Optimal Power of Power Systems Including Distributed Generation Units Using

- Sunflower Optimization Algorithm. IEEE Access. 2019; 7: 109289-11.
- [21] Alshammari BM, Guesmi T. New Chaotic Sunflower Optimization Algorithm for Optimal Tuning of Power System Stabilizers. Journal of Electrical Engineering & Technology. 2020; 15: 1985-12.
- [22] Gomes GF, Cunha Jr SS, Ancelotti Jr AC. A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates. Engineering with Computers. 2019; 35: 619-7.
- [23] Proença HM, Leeuwen M. Interpretable multiclass classification by MDL-based rule lists. Information Sciences. 2020; 512: 1372-21.
- [24] Miranda TZ, Sardinha DB, Cerri R. (2019). Preventing the Generation of Inconsistent Sets of Classification Rules. Expert Systems with Applications. 2019;165.