K-NN SINIFLANDIRMA ALGORİTMASININ PARAMETRELERİNİN ZEKİ OPTİMİZASYON TEKNİKLERİYLE İYİLEŞTİRİLMESİ

Erdem Korhan AKÇAY*Eskişehir Teknik Üniversitesi

Özet

Makine öğrenmesi alanında verilerin sınıflandırılması algoritmalarının içinden sıkça kullanılan K-NN algoritmasının daha da güçlendirilmesi ve daha doğru sonuçları elde etmek için parametre optimizasyonun sağlanması önemli bir adımdır. Parametre optimizasyonunun en etkin şekilde kullanılabilmesi için zeki optimizasyon tekniklerinden olan Genetik Algoritması, Parçacık Sürü Algoritmaları ile K-NN algoritmasıyla birlikte çalıştırılmasıyla güçlü sonuçlar elde edilecektir. Genetik algoritma ve parçacık sürü optimizasyonu, K-NN algoritmasının kullanıcı tarafından belirlenen parametrelerini otomatik olarak ayarlayarak, daha iyi sınıflandırma sonuçları elde etmeyi hedefler.

1. Giriş

Sınıflandırma problemleri, makine öğrenimi alanında büyük bir öneme sahiptir. Bu problemlerde, veri noktaları farklı sınıflara atanırken, karar verme sürecinde kullanılan algoritmalara ihtiyaç duyulur. K-NN (k-en yakın komşu) sınıflandırma algoritması, basit ancak etkili bir yöntem olarak bilinir. Temel fikri, bir veri noktasını sınıflandırmak için çevresindeki k en yakın komşusuna bakmaktır. Ancak, K-NN algoritması birkaç parametreye sahiptir ve bu parametrelerin optimal değerlerini belirlemek her zaman kolay değildir.

Bu bağlamda, zeki optimizasyon teknikleri, K-NN sınıflandırma algoritmasının parametrelerinin iyileştirilmesinde önemli bir rol oynamaktadır. Zeki optimizasyon teknikleri olarak, genetik algoritmalar, parçacık sürü optimizasyonu, bulanık mantık gibi doğal veya yapay zeka temelli yöntemleri içerir. Bu teknikler, karmaşık ve çok boyutlu parametre arama uzaylarında optimal çözümleri bulmak için kullanılır.

K-NN sınıflandırma algoritmasının temel parametreleri arasında komşuluk sayısı (k), veri noktalarının benzerlik ölçüsü ve ağırlıklandırma faktörü yer alır. Komşuluk sayısı, sınıflandırma doğruluğunu etkileyen kritik bir faktördür. Veri noktaları arasındaki benzerlik ölçüsü, K-NN'nin sınıflandırma kararlarını doğrudan etkiler. Ağırlıklandırma faktörü ise her bir komşunun sınıflandırmaya olan katkısını belirler.

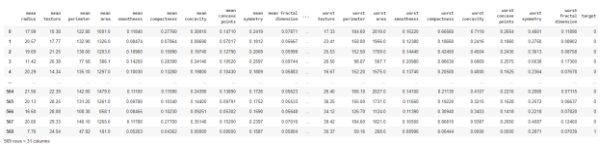
Bu makalenin amacı, K-NN sınıflandırma algoritmasının parametrelerini iyileştirmek için zeki optimizasyon tekniklerinin nasıl kullanılabileceğini incelemektir. Zeki optimizasyon teknikleri, parametre arama uzayında etkin bir şekilde gezinerek en iyi parametre değerlerini bulmayı hedefler. Bu sayede, K-NN algoritmasının sınıflandırma performansı artırılabilir ve daha doğru sonuçlar elde edilebilir.

Bu çalışmanın bulguları, literatürdeki bazı önemli çalışmaların sonuçlarına da atıfta bulunarak sunulmuştur. Örneğin, "Classification of Heart Disease Using K-Nearest Neighbor and Genetic Algorithm" adlı çalışmada, jenerasyon sayısı 20 olarak alındığında, göğüs kanseri veri setinde %94 doğruluk oranı elde edilmiştir. Ayrıca, "Particle Swarm Optimization Feature Selection for Breast Cancer Prediction" adlı başka bir çalışmada ise göğüs kanseri veri setinde %97.54 ve %96.66 doğruluk oranlarına ulaşılmıştır.

Bu bulgular, zeki optimizasyon tekniklerinin K-NN algoritmasının parametrelerini iyileştirme potansiyelini göstermektedir. Bu makalenin devamında, detaylı analizler ve sonuçların daha geniş bir değerlendirmesi sunulacaktır.

2. Veri Seti

Uygulamanın yapıldığı veri seti Python programlama dilinin makine öğrenmesi uygulamaları amacıyla oluşturulmuş paketi "**sklearn**"in içerisinde barındıran "**Göğüs kanseri**" veri seti kullanılmıştır. Veri setini incelediğimizde **Şekil 1**'de genel bir bakışını görmekteyiz. İncelediğimizde 31 değişkenimiz ve her bir değişkenin 568 değeri bulunmaktadır.



Şekil 1

Değişkenleri incelediğinde:

radius: Tümörün ortalama yarıçapı,

texture: Tümörün ortalama dokusu.

perimeter: Tümörün ortalama çevresi,

area: Tümörün ortalama alanı,

smoothness: Tümörün ortalama düzgünlüğü

compactness: Tümörün ortalama sıkışıklığı.

concavity: Tümörün ortalama çöküklüğü.

concave points: Tümördeki ortalama çukur nokta sayısı.

symmetry: Tümörün ortalama simetrisi.

fractal dimension: Tümörün ortalama fraktal boyutu.

Göğüs kanseri veri setinde hedef değer, iyi huylu veya kötü huylu olup olmadığını belirten 0 veya 1'dir

3. Yöntem

3.1. K – En Yakın Komşu Algoritması

K-NN algoritması, en temel örnek tabanlı öğrenme algoritmaları arasındadır. Örnek tabanlı öğrenme algoritmalarında, öğrenme işlemi eğitim setinde tutulan verilere dayalı olarak gerçekleştirilmektedir. Yeni karşılaşılan bir örnek, eğitim setinde yer alan örnekler ile arasındaki benzerliğe göre sınıflandırılmaktadır. K-NN algoritmasında, eğitim setinde yer alan örnekler n boyutlu sayısal nitelikler ile belirtilir. Her örnek n boyutlu uzayda bir noktayı temsil edecek biçimde tüm eğitim örnekleri n boyutlu bir örnek uzayında tutulur. Bilinmeyen bir örnek ile karşılaşıldığında, eğitim setinden ilgili örneğe en yakın k tane örnek belirlenerek yeni örneğin sınıf etiketi, k en yakın komşusunun sınıf etiketlerinin çoğunluk oylamasına göre atanır.

3.1.2. K-NN Algoritmasının Genel İşleyişi

K-En Yakın Komşular algoritmasının performansı için kritik öneme sahip noktalardan birisi örnekler arası yakınlığın nasıl ölçümleneceğidir. Yakınlık, Manhattan, Minkowski, Öklid uzaklığı gibi uzaklık ölçütleri kullanılarak hesaplanır. K-NN algoritması az sayıda parametre gerektirmesi ve basit bir yapıya sahip olması önemli noktalarıdır. K-NN algoritmasında, ortaklık yapısının çoğunluk oylamasına dayalı olarak belirlenmesi, simetrik olmayan dağılıma sahip veri setlerinde sıklıkla görülen sınıfların, yeni örneklerin sınıf etiketlerinin belirlenmesinde daha baskın bir role sahip olmalarına neden olmaktadır. Bu nedenle, temel K-NN algoritmasının uzaklık

ölçütünün etki değerine farklı şekillerde ağırlık değeri atayan yöntemler bulunmaktadır.

K-NN algoritması, büyük eğitim setlerinin varlığında son derece etkili sonuçlar üretebilir. K-NN algoritması, ilgisiz değerlerin varlığında dahil sınıflandırma modeli oluşturabilir. Bu durumlarda eğitim süresi önemli ölçüde artmaktadır. K-NN algoritmasının basit yapısına rağmen, hesaplaması yüksek maliyetlere sahip olmaktadır. Özellikle büyük eğitim veri setleri için, sınıf etiketini belirlemek istenen örnek ile veri setindeki örnekler arasındaki farklılıkları belirlemek oldukça maliyetli olabilmektedir. K-NN algoritması, temel bileşenler analizi gibi boyut azaltma teknikleri veya arama ağaçları gibi daha güçlü veri yapıları kullanılarak bu maliyeti azaltabilir. Bunun yanı sıra, K-NN algoritması, komşu sayısı ve uzaklık ölçütü gibi parametrelere karşı hassas bir yapıya sahiptir ve çok boyutlu veri setlerinde çalışması etkili olmaz.

3.1.3. K-NN Parametreleri

Uzaklık ölçütü, komşu sayısı (k) ve ağırlıklandırma yöntemi, K-NN algoritmasının performansında önemli parametrelerdir. Bu parametreler alt bölümlerde açıklanmaktadır.

3.1.3.1 Uzaklık Ölçütleri

K-En Yakın Komşular algoritmasının kullandığı uzaklık ölçütleri olarak başlıca, Manhattan, Minkowski, Öklid uzaklıkları kullanılmaktadır.

3.1.3.1.1 Manhattan Uzaklığı

Manhattan uzaklığı, boyutu n olan iki nokta arasındaki uzaklığın mutlak değerlerince toplamıdır. Diğer bir deyişle, A ve B noktaları arasındaki Manhattan uzaklığı'nı hesaplamak istersek, $A = (x_1, \dots, x_n)$ ve $B = (y_1, \dots, y_n)$ olmak üzere,

$$\left[\sum_{i=1} |x_i - y_i| \right]$$

Formülüyle hesaplanır.

3.1.3.1.2 Minkowski Uzaklığı

Öklid uzayında tanımlı bir dizi olan Minkowski uzaklığı. Sınıflandırma, makine öğrenmesi, veri madenciliği alanlarında yoğun bir şekilde kullanılan Öklid uzaklığı, Uzaklık ölçütlerinin genelleştirilmiş halidir. Diğer bir deyişle, A ve B noktaları arasındaki Minkowski uzaklığı'nı hesaplamak istersek, $A = (x_1, x_2, \dots, x_n)$ ve $B = (y_1, y_2, \dots, y_n)$ olmak üzere,

$$\left[\sum_{i=1} |x_i - y_i|^p \right]^{1/p}$$

Minkowski uzaklığı, genel bir formül olarak tanımlanmakta olup, p'nin çeşitli değerleri için farklı uzaklık ölçütlerini tanımlar. Minkowski ölçütünde p=2 durumunda Öklid uzaklığı elde edilir, p=1 durumunda Manhattan uzaklığı, aynı zamanda $n \to \infty$ durumunda Chebyschev uzaklığı elde edilir.

3.1.3.1.3 Öklid Uzaklığı

Öklid uzaklığı, sınıflandırma algoritmalarında yaygın olarak kullanılan uzaklık ölçütüdür. Öklid uzaklığı, iki nokta arasındaki doğrusal uzaklığı belirtir. Diğer bir deyişle A ve B noktaları arasındaki Öklid uzaklığı'nı hesaplamak istersek,

$A = (x_1, x_2, \dots, x_n)$ ve $B = (y_1, y_2, \dots, y_n)$ olmak üzere,

$$\left[\sum_{i=1} (x_i - y_i)^2 \right]^{1/2}$$

Öklid uzaklığı, K-ortalama kümeleme algoritması, temel K-NN algoritması gibi sınıflandırma ve kümeleme algoritmalarında yakınlığın ölçümlenmesi için kullanılan temel uzaklık ölçütüdür.

2.1.3.2 Komşu Sayısı (k)

K-NN algoritmasında, komşu sayısı (k) parametresinin değeri sınıflandırmayı belirler. Sınıflandırma sürecinde, k=1 için, sadece en yakın komşunun bulunduğu sınıfa seçilir. Örneğin, k sayısı örnek sayısına (N) yaklaştıkça veri setindeki tüm veriler gözden geçirilir ve oylama yoluyla seçilir.

3.1.3.3 Ağırlıklandırma

Ağırlık değerleri sınıflandırılmak istenen noktaya daha yakın olan komşu noktaların, çoğunluk oylamasına daha fazla katkı koyması amaçlanır. Kullanılan ağırlık değeri atama yöntemleri, her bir komşunun ağırlığının, *d,* komşuların arasındaki uzaklık olmak üzere*, 1/d* ya da $1/d^2$ şeklinde alınmasıdır.

3.2 Genetik Algoritma

Genetik algoritma (GA), çevrelerindeki değişikliklere uyum sağlayabilen türlerin hayatta kalmayı ve üremeyi başarabildiği doğal seçilim sürecini simüle eden bir algoritmadır. Basitçe söylemek gerekirse, bir problemin çözümü için ardışık nesillerin bireyleri arasındaki "en uygun olanın hayatta kalması" durumunu simüle ederler. Her nesil, bireylerin bir popülasyonunu içerir. Her birey arama uzayında bir noktayı ve olası bir çözümü temsil eder. Her birey, bir dize olarak temsil edilir. Bu dize, Kromozom ile benzerlik gösterir.

GA, popülasyon adı verilen bir çözüm setiyle başlatılmaktadır. Her bir jenerasyon boyunca popülasyon büyüklüğü korunmaktadır. Her bir jenerasyonda, her bir kromozomun uygunluk değeri değerlendirilmekte ve sonra uygunluk değerlerine göre bir sonraki jenerasyon için kromozomlar olasılıksal biçimde seçilmektedir. Seçilen bazı kromozomlar rassal olarak eşleşmekte ve çaprazlama ve mutasyon rassal olarak gerçekleştirilerek genç bireyler üretilmektedir. Yüksek uygunluk değerine sahip kromozomlar yüksek olasılıkla seçildiğinden, yeni jenerasyona ait kromozomlar eski jenerasyona ait kromozomlardan daha iyi uygunluk değerine sahip olabilmektedir. Bu evrim süreci sonlandırma şartı sağlanana kadar tekrar etmektedir. GA'daki çözümler diziler veya kromozomlar olarak adlandırılmaktadır. Çoğu durumda, kromozomlar listeler veya diziler olarak gösterilmektedir. Dolayısıyla, GA ile yapılan pek çok işlem liste veya dizilerle yapılan işlemlerdir (Kumar vd., 2010).

GA, ilerleyiş olarak sırasıyla başlangıç popülasyonunun oluşturulması, uygunluk değerlerinin hesaplanması, seçim, çaprazlama, mutasyon, yeni nesil üretimi ve en iyi değerin döndürülmesi biçiminde yedi aşamada incelenebilmektedir.

i. Başlangıç Popülasyonu Oluşturulması:

Çözümler gösterimi oluşturulup, amaç tanımlandıktan sonra, değerlendirme sürecine ilerlenmektedir. İlk olarak, rassal olarak veya önceden sahip olunan bilgiyle kodlanmış çözümlerin bir başlangıç popülasyonu veya kromozomu yaratılmaktadır (Zhou, 2006)

ii. Uygunluk Değerlerinin Hesaplanması:

Başlangıç çözümlerinin oluşturulmasının ardından popülasyonun iteratif bir değerlendirme ve yeniden üretim sürecine girmesi sağlanmaktadır. Değerlendirme aşamasında amaç fonksiyonu doğrultusunda her bir kromozoma bir uygunluk değeri atanmaktadır. (Zhou, 2006)

iii. Seçim Operatörü:

Uygunluk değerleri hesaplandıktan sonra seçim aşaması ile devam edilmektedir. Her

bir jenerasyon sırasında var olan popülasyonun bir grubu yeni jenerasyonlar oluşturmak için korunmaktadır. Kromozomlar, uygunluk değeri daha iyi olan bireylerin daha büyük olasılıkla seçildiği uygunluk tabanlı bir süreçle seçilmektedir. Belirli seçim metotları ile her bir çözümün uygunluğu değerlendirilmekte ve tercihe bağlı olarak en iyi çözümler seçilmektedir. Diğer metotlarla sadece popülasyonun rassal bir örneklemi değerlendirilmekte, ancak bu süreç epey zaman alabilmektedir. Seçim metotlarına rulet çemberi seçimi, stokastik evrensel seçim, sıralama seçimi ve turnuva seçimi örnek gösterilebilmektedir. (Sastry, Goldberg ve Kendall, 2014; Zhou, 2006)

iv. Çaprazlama Operatörü:

Seçim aşamasından sonra belirlenen bir çaprazlama operatörü ve bir çaprazlama oranıyla ebeveynler üzerinde yeni bireylerin çaprazlaması gerçekleşmektedir. Çaprazlama operatörü, her bir ebeveynden seçilen genleri kopyalayarak iki yeni genç birey üretmektedir. Her bir genç bireydeki belirli bir pozisyonda bulunun gen, iki ebeveynden birindeki aynı pozisyondaki genden kopyalanmaktadır. Belirlenen pozisyonlardaki genler için hangi ebeveynden genlerin kopyalanacağına çaprazlama maskesi adı verilen ek bir dizi ile karar verilmektedir. Tek noktalı çaprazlama, iki noktalı çaprazlama ve uniform çaprazlama çaprazlama operatörlerine örnek olarak gösterilmiştir. Çaprazlama ile yeni kromozomlar oluşturulmaktadır. Böylece uygunluk değeri daha yüksek kromozomların ortaya çıkma olasılığı sağlanmaktadır. (Haldurai, Madhubala ve Rajalakshmi, 2016)

v. Mutasyon Operatörü:

Mutasyon operatörü genellikle çaprazlama operatöründen sonra devreye girmektedir. Mutasyon operatörüyle, tek bir rassal gen seçilerek birey alelleri rassal kromozomların lokuslarında çok küçük bir olasılıkla değiştirilmektedir. Bir sonraki seçimde gelişecek veya yok olacak uygunluk değeri daha yüksek veya daha düşük bir kromozom yaratılabilmektedir. Amaç en iyiyi bulmak olduğundan eğer popülasyonda bir tek kötü çözüm varsa olumsuzluk yaratabilmektedir. Diğer bir taraftan mutasyon sürecinde iyi bir çözüm yaratılırsa en iyi çözüme ulaşmada çok yardımcı olmaktadır. Mutasyon operatörü için nokta mutasyonu, güç mutasyonu, küçültme mutasyonu, denetleme mutasyonu, eşşizlik mutasyonu, değişken-olasılık mutasyonu, iki noktalı mutasyon, gauss mutasyonu, değiş tokuş mutasyonu ve uniform mutasyon, non-uniform mutasyon örnek verilmiştir. Pek çok evrimsel algoritma yerel optimum bir noktada sıkışma eğilimi gösterebildiği için nihai sonuç evrensel optimum noktadan çok daha uzakta sonlandırılabilmektedir. Bunun önüne geçilmesi ve algoritmanın yerel optimumlardan kurtarılabilmesi için mutasyon operatörü ayrıca önem kazanmaktadır. (Chauhan, Singh ve Aggarwal, 2021)

vi. Sonlandırma:

Uygunluk değerinin hesaplanması, seçim, çaprazlama, mutasyon adımları ve bir GA stratejisi döngüsü sonlandırma şartı sağlanana kadar uygulanmaktadır. onlandırma şartları, en küçük uygunluk değeri şartını sağlayan bir çözümün bulunması, belirli sayıda jenerasyona ulaşılması, ayırılan bütçeye ulaşılması (hesaplama zamanı/para), en yüksek sıralamadaki çözümün uygunluk değerine ulaşılması veya daha fazla iyileşme olmayacak kadar başarılı bir çözüme ulaşılması olarak düşünebilmektedir. (Kumar vd., 2010)

vi.i. Yer Değiştirme (Replacement):

Her bir jenerasyonda genetik operatörleri gerçekleştirmek için bireyler seçilmektedir. Yaratılan her bir yeni birey popülasyondaki uygunluk değeri en kötü bireyle karşılaştırılmaktadır. Eğer yeni birey popülasyondaki uygunluk değeri en düşük bireyden daha iyi bir uygunluk değerine sahipse o bireyin yerine seçilmektedir.(Wu ve Ji, 2007)

vi.ii. Elitizm:

Bu süreçte, sonraki jenerasyonda yeniden üretim için en yüksek uygunluğa sahip bireyler korunmaktadır. Elitizm ile mevcut popülasyondan elde edilen en iyi çözümün sonraki popülasyon için değişmemiş biçimde kopyalanması ve GA tarafından elde edilen çözüm kalitesinin bir jenerasyondan diğerine geçerken her bir iterasyondan sonra artması garanti edilmektedir (Markandeswar vd., 2016)

3.3 Parçacık Sürü Optimizasyonu

Parçacık sürü optimizasyonu, balık ve kuş gibi canlıların doğadaki sosyal davranışlarından esinlenerek geliştirilen meta-sezgisel bir optimizasyon algoritmasıdır (Eberhart ve Kennedy, 1995). Parçacık sürü optimizasyonunun köklerini, evrimsel hesaplama ve yapay yaşam örüntüleri olmak üzere iki ana bileşene dayanır. Bu nedenle, parçacık sürü optimizasyonu genetik algoritmalarla vev evrimsel programlama ile doğrudan bir ilişkisi vardır.

Bir kuş sürüsü birbirine çarpmadan birlikte uçar, yön değiştirir ve bunları eşzamanlı olarak gerçekleştirir. Sürüdeki bir kuş yiyecek bulduğunda, diğer kuşlarla iletişim kurarak onların da yiyeceğe ulaşmasını sağlar. Aynı şekilde, balık sürülerinde de bir balık bir tehlike sezdiğinde sürüdeki diğer balıklarla iletişime geçerek onların tehlikeden haberdar olmasını sağlar. Özellikle kuş ve balık sürülerinin iş birlikçi davranışlarından ilham alan parçacık sürü optimizasyonu, yapay parçacıkların arama uzayında kuş ve balık sürülerinin hız ve konum gibi fiziksel özelliklerini taklit etmesiyle optimal çözümü bulmayı amaçlar. Algoritmada, her parçacık muhtemel bir çözümü temsil etmektedir (Eberhart ve Kennedy, 1995; Kaewkamnerdpong ve Bentley, 2005; Krause vd., 2013; Vasuki, 2020).

Bir sürü modeli geliştirmek için belirli prensiplerin sağlanması gerekmektedir. Bunlar,

• Çoklu birimler arasında bilgi paylaşımı,

• Birimlerin kendi kendini organize etmesi ve evrim geçirmesi,

• Birimlerin eş öğrenmesi açısından etkili olması,

• Uygulamalı ve gerçek zamanlı problemlere kolayca uyarlanabilmesi

Parçacıkların arama uzayındaki konumlarının değişimi bireylerin sosyo-bilişsel eğilimine bağlıdır ve parçacıkların mevcut konumlarının güncellenmesi için hız vektörü kullanılır. Bir parçacığın en iyi konumu "pbest" olarak adlandırılır ve bu değer hız vektörünün güncellenmesi için hafızada tutulur. Bununla birlikte, hız vektörünün hesaplanmasında popülasyondaki herhangi bir parçacık tarafından elde edilen en iyi konum da kullanılmaktadır. Bu konum ise "gbest" olarak adlandırılmaktadır. "pbest" ve "gbest" değerleri bulunduktan sonra parçacıkların konumu ve hızı güncellenir (Eberhart ve Kennedy, 1995). Bir parçacığın hızı,

$$v_i^{t+1} = wv_i^t + c_1r_1(pbest_i - x_i^t) + c_2r_2(gbest - x_i^t)$$

biçiminde hesaplanır. Burada, v_i^{t+1}, i. parçacığın $(t+1)$. iterasyondaki hızını; v_i^t i. parçacığın t. iterasyondaki hızını ve x_i^t, i. parçacığın t. iterasyondaki konumunu ifade etmektedir. Ayrıca, w, atalet ağırlığıdır ve genellikle [0,1.2] aralığından seçilir. Öğrenme katsayıları olarak adlandırılan c_1 ve c_2 genellikle [0, 2] aralığında belirlenir.

c_1, parçacığın kendi deneyimlerine, c_2 ise parçacığın sürüdeki diğer parçacıkların deneyimlerine göre hareket etmesini sağlar. r_1 ve r_2 ise [0, 1] aralığında düzgün dağılımdan rastgele üretilen değerlerdir.

Parçacıkların konumu ise,

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

biçiminde güncellenir. Burada, x_i^{t+1} , i. parçacığın $(t + 1)$. iterasyondaki konumunu ifade etmektedir (Eberhart ve Kennedy, 1995; Shi ve Eberhart, 1998; Kashani vd., 2021).

4. Bulgular

Bu çalışmanın amacı, K-NN sınıflandırma algoritmasının parametrelerini zeki optimizasyon teknikleriyle iyileştirerek, göğüs kanseri teşhisinde daha yüksek doğruluk oranları elde etmektir. Bu amaç doğrultusunda, GridSearch, genetik algoritma ve parçacık sürü optimizasyonu gibi zeki optimizasyon teknikleri uygulanmış ve elde edilen sonuçlar analiz edilmiştir.

PSO'nun ilerle şekli farklı olduğu için eklenmemiştir.

	Uzaklık	Komşu Sayısı	Ağırlık
G.A.	Manhattan	13	Uniform
K-NN	Manhattan	7	Uniform

Tablo 1

Analizler sonucu K-NN uygulamasını geliştirmek adına makine öğrenmesinde mutlak doğruluğu yakalamak adına belirli parametreleri bulmak adına uygulanan GridSearch ile birlikte uyguladığımızda en iyi metrikler olarak bizlere **{'distance': 'manhattan', 'n_neighbors': 13, 'weights': 'uniform'}** Metriklerini önerdiğini görüyoruz bu metriklerde accuracy(doğruluk) değerimize baktığımızda 0.9340 olarak yakalamış oluyoruz.

Genetik Algoritma'da ise parametre optimizasyonunu GridSearch mantığıyla ilerleyerek en optimal sonucu bizlere veriyor. Parametre optimizasyonunda olası parametre değerlerinin verilmesinin yanı sıra genetik algoritmanın parametrelerini "jenerasyon = 20, popülasyon boyutu = 30, yavruların boyutu = 12" olarak girildi. "**Tpot**" kütüphanesini kullanarak uyguladığımızda 20. Jenerasyonda 0.95614035 Doğruluk oranını yakalamış olduk ve en iyi metrikler olarak **{'distance': 'manhattan', 'n_neighbors': 7, 'weights': 'uniform'}** belirlenmiştir.

Parçacık Sürü Optimizasyonu uygulamasında "**pyswarm**" kütüphanesi yardımıyla uygulandığında model doğruluk oranımız 0.9385964 olmuştur. Diğer modellerin aksine PSO'da en iyi metriğin hesaplanması yerine, sistemden çıkarılması gereken belirli özellikleri(feature) saptar ve o saptanan özelliklerin çıkartılması üzerine kurulmuş bir sistem olarak dizayn edilmiştir. Bu sisteme göre PSO'nun uygulama için belirlediği featurelar **"mean texture, radius error, area error, smoothness error, compactness error, worst radius, worst texture, worst perimeter"** olarak saptanmıştır.

	K-NN(G.S.)	Genetik A.	Parçacık Sürü O.
Doğruluk	**0.9362**	**0.9561**	**0.9385**

Tablo 2

Bulgularımız, K-NN sınıflandırma algoritmasının parametrelerini zeki optimizasyon teknikleriyle iyileştirmenin, sınıflandırma performansını artırılmasında önemli bir potansiyele sahip olduğunu göstermektedir. GridSearch, Genetik Algoritma ve Parçacık Sürü Optimizasyonu yöntemleriyle elde edilen sonuçlar, doğruluk oranlarında belirgin bir iyileşme sağladığını göstermektedir. Bu çalışma, K-NN sınıflandırma algoritmasının kullanıldığı diğer problemler için de benzer iyileştirmelerin elde edilebilir. Gelecekteki çalışmalarda, daha geniş veri setleri ve

farklı zeki optimizasyon tekniklerinin kullanılmasıyla elde edilen sonuçların daha da geliştirilmesi hedeflenebilir.

5. Sonuç

Bu çalışmada, K-NN sınıflandırma algoritmasının parametrelerini zeki optimizasyon teknikleriyle iyileştirme potansiyeli incelenmiştir. GridSearch, genetik algoritma ve parçacık sürü optimizasyonu yöntemleri kullanılarak algoritmanın performansı artırılmış ve doğruluk oranları elde edilmiştir.

Bulunan parametrelerle elde edilen K-NN sınıflandırma modelinde, GridSearch parametre optimizasyonunda göğüs kanseri veri setinde %93.62 doğruluk oranı elde edilmiştir. Genetik algoritma ile yapılan optimizasyonda ise jenerasyon sayısı 20 olarak belirlenmiş ve %95.61 doğruluk oranı elde edilmiştir. Parçacık sürü optimizasyonu kullanılarak elde edilen sonuçlarda ise %93.85 doğruluk oranı elde edilmiştir.

Sonuçlarımız, genetik algoritmanın K-NN sınıflandırma algoritmasının parametrelerini iyileştirmek için etkili bir yöntem olduğunu göstermektedir. Ayrıca, GridSearch ve parçacık sürü optimizasyonu yöntemlerinin de sınıflandırma alternatif olarak kullabileceği görülmüştür.

Bu bulgular, zeki optimizasyon tekniklerinin K-NN sınıflandırma algoritmasının performansını artırmada önemli bir rol oynayabileceğini göstermektedir. Bu çalışma, gelecekteki araştırmalara ve uygulamalara yönelik bir temel sağlamaktadır. Daha fazla veri seti ve parametre kombinasyonunun incelenmesi, algoritmanın performansının daha da iyileştirilmesine katkı sağlayabilir.

6. Referanslar

- Erdal Taşcı1, Aytuğ Onan, "K-En Yakın Komşu Algoritması Parametrelerinin Sınıflandırma Performansı Üzerine Etkisinin İncelenmesi" *Akademik Bilişim Konferansı (AB 16),30.01.2016 05.02.2016*
- Anıl Yalçın, "Doktor Nöbet Çizelgeleme Problemi İçin Ağırlıklı Hedef Programlama Tabanlıgenetik Algoritma" *Kütahya Dumlupınar Üniversitesi Lisansüstü Eğitim Enstitüsü, Endüstri Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi*
- Eda Özkul, "Yapay Çekirge Sürü Optimizasyonu" *Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Trabzon, İstatistik ve Bilgisayar Bilimleri Anabilim Dalı, Doktora Tezi*
- Nurhayati, Fajar Agustian, Muhammad Dzil Ikram Lubis, "Particle Swarm Optimization Feature Selection for Breast Cancer Prediction" *2020 8th International Conference on Cyber and IT Service Management (CITSM)*
- M.Akhil jabbar, B.L Deekshatulu, Priti Chandra, "Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm" *Procedia Technology 10 (2013) 85 – 94*