Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 26, S. 234-241, Temmuz 2021

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 26, pp. 234-241, July 2021

Research Article

İkili Gri Kurt ve İkili Harris Şahin Optimizasyonları ile Web Haber Sayfalarının Sınıflandırılması

Muhammet Aktaş[1*], Fatih Kılıç[2]

[1*] Adana Alparslan Türkeş Bilim ve Teknoloji Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Adana, Türkiye (ORCID: 0000-0002-2598-3387), maktas@atu.edu.tr

[2] Adana Alparslan Türkeş Bilim ve Teknoloji Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Adana, Türkiye (ORCID: 0000-0002-8550-1562), fkilic@atu.edu.tr

(3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications June 11-13, 2021)

(**DOI:** 10.31590/ejosat.950497)

ATIF/REFERENCE: Aktaş, M. & Kılıç, M. (2021). İkili Gri Kurt ve İkili Harris Şahin Optimizasyonları ile Web Haber Sayfalarının Sınıflandırılması. *Avrupa Bilim ve Teknoloji Dergisi*, (26), 234-241.

Öz

İnternetin hızlı gelişmesi ile başta haber kaynakları, e-ticaret ve sosyal ağ uygulamaları olmak üzere çok sayıda web hizmeti ve sayfaları kullanıma sunuldu. Bu uygulamaların kullanımı ile inanılmaz büyüklükte video, ses ve metin gibi içerikler oluştu. Oluşan bu verilerin doğru olarak sınıflandırılması, web uygulamasından faydalanan kullanıcıların istedikleri verilere daha hızlı ve kolay erişmesini sağlar. Çok sayıda öznitelikten oluşan bu veriler metin sınıflandırması için yüksek hesaplama sürelerine neden olur. Yüksek boyutlara sahip veriler için daha az öznitelik ve düşük hesaplama süresi ile yüksek doğrulukta metin sınıflandırma başarısını öznitelik seçimi metotları kullanımı ile sağlamak mümkündür. Literatürde metin sınıflandırmasında kullanılan öznitelik seçim metotları filtreleme, sarma, gömülü ve hibrit yöntemler olarak sınıflandırılmaktadır. Bu çalışmada, metin sınıflandırılmasında öznitelik seçimi için İkili Gri Kurt Optimizasyonu (IGKO) ve İkili Harris Şahin Optimizasyonu (IHSO) algoritmaları ReliefF ile beraber kullanılmıştır. Çalışmada algoritmaların sonuçlarını değerlendirmek için 2 farklı özelliğe sahip veri kümesi kullanılmıştır. Birincisi,100 web belgesinden oluşan 2 kategoriye sahip bir veri kümesi, ikincisi ise 9 kategoriden oluşan (fizik, biyoloji, genetik vs) bilim haberleriyle ilgili web sayfalarından çıkarılan 450 web belgesini içeren veri kümesidir. Sonuçlara göre, IHSO amaç fonksiyonu ve öznitelik sayısına göre karşılaştırma yapılan diğer öznitelik seçim metotlarından daha performanslı olduğu görülmüştür.

Anahtar Kelimeler: Metin sınıflandırma, Öznitelik seçimi, İkili Gri kurt algoritması, İkili Harris Şahin algoritması, Metin madenciliği.

Classification of News Web Pages using Binary Grey Wolf and Binary Harris Hawk Optimizations

Abstract

With the rapid development of the internet, many web services and pages, especially news sources, e-commerce, and social network applications, have been released to use. Using these applications creates an incredible amount of content such as video, audio, and text. The classification of these data with high accuracy provides faster and easier access to the data which the users search for using the web applications. These datasets, consisting of high dimension features, give rise to high computation times for text classification. It is possible to achieve high accuracy with fewer features and less computation time for classification using feature selection methods on these datasets having high dimensions. In the literature, feature selection methods used in text classification can be classified as filtering, wrapping, embedded, and hybrid methods. In this study, Binary Grey Wolf Optimization (BGWO) and Binary Harris Hawk Optimization (BHHO) algorithms are used with ReliefF for feature selection in text classification. To evaluate the results of the proposed algorithms, two datasets having two different characteristics are used. The first dataset has 2 categories and 100 web documents. The second dataset has 9 categories (physics, biology, genetics, etc.) and 450 web documents extracted from science news web pages. The results show that BHHO has better performance than the compared feature selection methods according to fitness and the number of selected features.
Keywords: Text Classification, Feature Selection, Binary Grey Wolf algorithm, Binary Harris Hawk algorithm, Text Mining.

[*] Sorumlu Yazar: maktas@atu.edu.tr

1. Giriş

İnternetin hayatımıza girmesi ile başta haber kaynakları, e-ticaret ve sosyal ağ uygulamaları olmak üzere çok sayıda web hizmetleri ortaya çıktı. Bu hizmetlerin artması ile farklı yapıda ve yaş guruplarında kullanıcı sayılarında büyük ölçüde artış oldu. Böylece inanılmaz büyüklükte video, ses ve metin gibi içerikler oluştu. Bu oluşan içerikler içerisinde arama yapmak veya oluşan verilerden anlamlı çıkarımlar yapmak aranan bir içeriğe ulaşmada hız ve anlam bakımından kolaylıklar sağlamasının yanında otomatik mesaj sistemleri ile ilgili kişilere istedikleri içerikler anlık olarak ulaşmaktadır. Boyut ve tür açısından farklı özelliklere sahip bu veriler 3 gruba ayrılır: yapılandırılmış, yarı yapılandırılmış, yapılandırılmamış veriler. Ham veri kümeleri yüksek boyutlara sahiptir. Bu nedenle, yüksek boyutlarda olan verilerin işlenmesi doğru sınıflama, hızlı erişim gibi birçok sorunu beraberinde getirir (Shang vd., 2007). Büyük veri kümelerinin anlam kaybetmeden azaltılma yöntemleri, doğru analitik ve tahmin işlemleri için önemlidir. İndirgeme yöntemlerinden biri olan öznitelik seçimi, büyük hacimli verilerin boyutunu azaltarak yüksek doğruluk ve anlamlı veri elde etmek için gerekli bir işlemdir.

Temel olarak, bir belgenin metin sınıflandırması için bir kaç aşamalı görevler vardır. Ana görevlerden biri, metin belgesini bir terim-frekans vektörüne dönüştürmektir (Labani vd., 2018). Bu dönüşüm, her belgedeki her benzersiz terim, özellik uzayındaki bir boyuta karşılık geldiğinden, özellik uzayının yüksek boyutlu olmasına yol açar. İkinci süreç, terim-frekans vektörünü kullanan sınıflandırma sürecidir. Bununla birlikte, yüksek boyutlu veriler nedeniyle, yüksek hesaplama maliyetleri ve sınıflandırma doğruluğu önemli ölçüde azalır (Deng vd., 2019). Bu nedenle CPU ve bellek maliyetlerini düşürmek için sınıflandırma doğruluğunu artırarak öznitelik sayısını azaltmak gerekir.

Genel olarak, öznitelik seçim yöntemleri şu şekilde sınıflandırılır: filtre, sarıcı, gömülü ve hibrit yaklaşımlar (Wah vd., 2018). Filtre yöntemi her bir öznitelik için bir ağırlık hesaplamasını istatistiksel bir ölçü eşitliği kullanarak yapmaktadır (Das, 2001). Bu öznitelikler, özel sıralama yöntemi kullanılarak ağırlıklarına göre listelenir. Sınıflandırma için belirli bir eşik değerden daha büyük ağırlığa sahip öznitelikler kullanılırken, diğer öznitelikler arama özelliği uzayından kaldırılır. Sarmalayıcı yaklaşımlar (Liu ve Setiono, 1997), bir arama problemi gibi tüm özniteliklerin en alakalı alt öznitelik kümesini araştırır. Bu yaklaşımlar, aday çözümleri değerlendirmek için bir veya daha fazla uygunluk işlevi kullanır. Bu uygunluk işlevi genellikle sınıflandırma doğruluğundan ve/veya seçilen özelliklerin sayısından oluşur. Gömülü yaklaşım (Xing, Jordan ve Karp, 2001), sınıflandırıcının eğitim aşamasında öznitelik seçimini bütünleştirir; bu nedenle, bu yöntemler, sarmalayıcı yöntemi olarak kullanılan öğrenme modeline özgüdür. Hibrit yaklaşımlar, filtre, sarıcı ve gömülü yaklaşımların birleştirerek kullanılmasıdır.

Son yıllarda, optimizasyon problemlerini çözmek için doğal arama mekanizmaları ve evrimsel ilkeleri taklit ederek önerilen birçok optimizasyon algoritması vardır. Bunlar, amaç sayılarına, arama mekanizmalarına ve bir çözümün veya popülasyonun kullanımı gibi çeşitli bakış açılarına göre sınıflandırılabilir. Bu algoritmalar kabul edilebilir hesaplama süreleri ile optimum çözüme yakın uygun çözümlere ulaşabilmektedir. Evrim temelli genetik algoritmanın ve kaos optimizasyonunun hibrit kullanımı, özellik seçimi için metin sınıflandırmasında etkili olmuştur (Chen vd., 2013). Diğer bir hibrit model, filtre ve sarma yönteminin kombinasyonu Günal tarafından yapılmıştır (Günal, 2012). Sürü zekâsına dayalı algoritmalar basit ve kolay uygulanması nedeniyle araştırmacılar tarafından büyük ilgi görmüştür. Lee, Park, Kim ve Kim (2019) 'daki metin sınıflandırma problemi için kuşların yiyecek arama davranışını taklit ederek parçacık sürüsü optimizasyonu (PSO), Manoj, Praveena ve Vijayakumar (2019)' da ise karınca kolonilerinin en kısa yolunu feromonlarla bulmasını modelleyen Karınca Koloni optimizasyon (KKO) algoritması tercih edilmiştir. Kurt sürüsünü modelleyen BGWO algoritması, Arapça metin sınıflandırması için elit tabanlı çaprazlama kullanılır (Chantar vd., 2020), Sinir Ağı (NN) sınıflandırıcısı, metin sınıflandırması için çok amaçlı Gri Kurt (MOGW) algoritması ile birlikte uygulanır (Asgarnezhad vd., 2020), Ateş böceklerinin sosyal davranışlarından esinlenen Ateşböceği algoritması (AA) Arapça metin sınıflandırması için kullanılmıştır (Marie-Sainte ve Alalyani, 2020). Metin sınıflandırmasına ilişkin literatürün kapsamlı bir incelemesi (Aggarwal ve Zhai, 2012; Jindal, Malhotra ve Jain, 2015) 'da bulunabilir. Literatür incelediğinde, farklı metin madenciliği problemleri üzerine çok sayıda çalışma olduğu görülmektedir.

Yapılan çalışmalar incelendiğinde sürü tabanlı algoritmalar ön plana çıkmaktadır. Aktaş ve Kılıç (2021)'deki çalışmada 2 sınıftan oluşan haberlerin sınıflandırılması için ikili Gri Kurt Optimizasyonu (IGKO) algoritması ile başarılı sonuçlara ulaşıldığı sunulmuştur. Bu sebeple bu çalışmada, en yeni optimizasyon tekniklerinden olan ikili Gri Kurt Optimizasyonu (IGKO) ve ikili Harris Şahin Optimizasyonu (IHSO) algoritmaları ReliefF algoritması ile birleştirilerek haber sayfalarının metin sınıflandırma çalışması yapılmıştır. Ayrıca önerilen modellerin performanslarını değerlendirmek için K-Nearest-Neighbor (KNN) sınıflandırıcı kullanılırken, (Aktaş ve Kılıç, 2021)'de oluşturulan 2 sınıflı veri kümesi ile 9 farklı sınıf sayılarına sahip bir veri kümesi bir haber sitesi kullanılarak oluşturulmuştur. Böylelikle algoritmaların performansları farklı sınıf ve sayıdaki veri kümeleri üzerine etkisi incelenmiştir.

Makalenin geri kalanı aşağıdaki şekilde düzenlenmiştir: Bölüm 2'de, ikili Gri Kurt Optimizasyonu ve ikili Harris Şahin Optimizasyonu açıklanmaktadır. Ayrıca uygunluk fonksiyonu ve web sayfasından alınan metinlerin ön işleme aşamaları, TF-IDF değerlerinin hesaplanması ve ReliefF algoritması Bölüm 2'de verilmiştir. Bölüm 3'te deneysel sonuçlar ve tartışmalar sunulmuştur. Son olarak, Bölüm 4'de makale sonuçları değerlendirilmesine yer verilmiştir.

2. Materyal ve Metot

2.1. İkili Gri Kurt Optimizasyonu

Gri Kurt Optimizasyonu (GKO), Mirjalili ve diğerleri tarafından sunulan kurt sürüsü davranışını taklit eden popülasyon tabanlı bir sürü algoritmasıdır (Mirjalili vd.,2014). Gri kurt sürüsü arasındaki hiyerarşi, bir besin piramidine benzer şekilde katı bir sosyal ilişkiye sahiptir. Kurt sürülerinde alfa (α), beta (β), delta (δ) ve omega (ω) olmak üzere 4 farklı tür vardır. Alpha, kurt sürüsüne avlanma ve keşifte liderlik eder. Beta ve delta sürüleri, av keşfi için alfa'ya karar desteği sağlar. Sürüdeki her kurt, aday çözümü temsil eder ve av, mümkün olan en iyi çözümdür. α, β ve δ en iyi üç değeri temsil eder. Kalan ω çözümleri konumlarını Alfa (α), Beta (β) ve Delta (δ) çözümlerine göre günceller. Her kurt (i), (1-3) denklemlerini

kullanarak mümkün olan en iyi üç çözüme olan mesafesini hesaplar.

$$\vec{A} = 2\vec{a} \,.\, \overrightarrow{r_1} - \vec{a} \quad (1)$$

$$\vec{B} = 2\vec{a} \,.\, \overrightarrow{r_2} \quad (2)$$

$$\vec{D}_\alpha = \left|\vec{B}.\ \vec{X}_\alpha - \vec{X}_i\right| \ , \vec{D}_\beta = \left|\vec{B}.\ \vec{X}_\beta - \vec{X}_i\right|, \ \vec{D}_\delta = \left|\vec{B}.\ \vec{X}_\delta - \vec{X}_i\right| \quad (3)$$

$\vec{X}_i , \vec{X}_\alpha, \vec{X}_\beta , ve\ \vec{X}_\delta$: $sırasıyla\ \ i.$çözümün , $\alpha, \beta\ ve\ \delta$ pozisyon vektörü.

$\vec{D}_\alpha, \vec{D}_\beta, ve\ \vec{D}_\delta$ $sırasıyla\ \alpha, \beta, ve\ \delta\ kurtları\ uzaklık\ vektör$ü.

$\overrightarrow{r_1}\ ve\ \overrightarrow{r_2}$ [0,1] $arasında$ üretiken $rastgele\ vektör.$

$\vec{A}\ ve\ \vec{B}$ $katsayı\ vektör$ü.

vektör $\vec{a}$, denklem (4) kullanılarak güncellenir.

$$\vec{a} = 2 - 2 * iter/MAX\_ITER \quad (4)$$

$iter$, geçerli yineleme sayısını gösterir, MAX_ITER maksimum yineleme sayısını temsil eder.

Standart GKO'da konum, sürekli uzaydaki herhangi bir noktayı temsil eder. Bu nedenle güncelleme prosedürü kolaylıkla uygulanabilir. Ancak İkili GKO (IGKO), çözüm uzayında bir ikili vektör (ikili çözüm) arar. IGKO'nun güncelleme prosedürü, sürekli çözüm alanından ikili çözüm alanına uyarlanmıştır. Kurtların $\vec{A}$ ve $\vec{B}$ konum vektörleri ve $\vec{D}_\alpha$, $\vec{D}_\beta$, $\vec{D}_\delta$ vektör değerleri aynı denklemler (1-3) kullanılarak hesaplanır. s_1^d, s_2^d ve s_3^d sürekli değerli adım boyutlarıdır ve sigmoid fonksiyonları denklem (5-7) kullanılarak hesaplanır.

$$s_1^d = 1/(1 + e^{-10(A^d.D_\alpha^d - 0.5)}) \quad (5)$$

$$s_2^d = 1/(1 + e^{-10(A^d.D_\beta^d - 0.5)}) \quad (6)$$

$$s_3^d = 1/(1 + e^{-10(A^d.D_\delta^d - 0.5)}) \quad (7)$$

burada d, bir çözümün d'inci boyutudur.

Aktarım işlevi, devamlı değerleri ikili değerlere dönüştürmek için kullanılır. $nstep_1^d, nstep_2^d$ ve $nstep_3^d$, d boyutundaki ikili adımlardır ve denklem (8-10) ile hesaplanır.

$$nstep_1^d = \begin{cases} 1 & if\ (s_1^d \geq rand) \\ 0 & else \end{cases} \quad (8)$$

$$nstep_2^d = \begin{cases} 1 & if\ (s_2^d \geq rand) \\ 0 & else \end{cases} \quad (9)$$

$$nstep_3^d = \begin{cases} 1 & if\ (s_3^d \geq rand) \\ 0 & else \end{cases} \quad (10)$$

$rand$ [0,1] arasında üretilen rastgele bir sayıdır.

$X_1^d, X_2^d, ve\ X_3^d$ ifadeleri α, β ve δ kurtlarının pozisyon vektörleridir. Değerler denklem (11-13) ile hesaplanır:

$$X_1^d = \begin{cases} 1 & if\ (X_\alpha^d + nstep_1^d) \geq 1) \\ 0 & else \end{cases} \quad (11)$$

$$X_2^d = \begin{cases} 1 & if\ (X_\beta^d + nstep_2^d) \geq 1) \\ 0 & else \end{cases} \quad (12)$$

$$X_3^d = \begin{cases} 1 & if\ (X_\delta^d + nstep_3^d) \geq 1) \\ 0 & else \end{cases} \quad (13)$$

Sonuç olarak, sonraki iterasyonda kullanılacak olan pozisyon bilgisi çaprazlama metoduna benzer şekilde tüm boyutta denklem (14) ile hesaplanır.

$$\vec{X}_i(nit) = \begin{cases} X_1 & if\ rand < 1/3 \\ X_2 & elseif\ 1/3 \leq rand \leq 2/3 \\ X_3 & else \end{cases} \quad (14)$$

İkili Gri Kurt Optimizasyon algoritmasının kaba kodu Algoritma 1 de verilmiştir.

Algoritma 1. İkili Gri Kurt Optimizasyon algoritması

$NWolf$ ve MAX_ITER parametrelerini belirle

Kurtlara rastgele başlangıç pozisyon bilgisi ata

Her kurt için Fitness değerlerini hesapla

$\vec{X}_\alpha, \vec{X}_\beta\ ve\ \vec{X}_\delta$ vektörlerini bul

for i=1: MAX_ITER **do**

$\vec{A}, \vec{B}\ ve\ \vec{a}$ yı denklem (1,2 ve 4) ile kullanarak güncelle

Kurtların pozisyon bilgilerini denklem (5-14) ile güncelle

Her kurt için Fitness değerlerini hesapla

$\vec{X}_\alpha, \vec{X}_\beta\ ve\ \vec{X}_\delta$ değerlerini güncelle

end for

En iyi çözüm $\vec{X}_\alpha$ döndür

2.2. İkili Harris Şahin Optimizasyonu

Harris şahin optimizasyonu (HSO), Heidari ve arkadaşları tarafından 2019'da önerilen yeni bir meta-sezgisel algoritmadır (Heidari vd., 2019). HSO, doğadaki Harris şahinlerinin avını, sürpriz saldırısını ve farklı saldırı stratejilerini keşfetmek için Harris şahinlerini taklit eder. HSO'da, aday çözümler şahinler tarafından temsil edilirken, en iyi çözüm av olarak bilinir. Harris şahinleri, güçlü gözlerini kullanarak avın izini sürmeye çalışır ve tespit edilen avı yakalamak için sürpriz saldırılar gerçekleştirir.

Genel olarak HSO, sömürü ve keşif aşamalarına göre modellenmiştir. HSO algoritması keşiften sömürüye aktarılabilir ve ardından keşif davranışı, avın kaçan enerjisine bağlı olarak değiştirilir. Bu değerler denklem (15) ve (16) ile hesaplanır.

$$E = 2E_0\left(1 - \frac{t}{T}\right) \quad (15)$$

$$E_0 = 2r - 1 \quad (16)$$

Burada t mevcut yinelemedir, T maksimum yineleme sayısıdır, E_0 ilk enerjidir ve [−1, 1] 'de rastgele oluşturulmuştur. r [0, 1] arasında rastgele bir sayıdır. Avın kaçan enerjisi | E | ≥ 1 olduğunda HSO şahinlerin farklı bölgelerde global olarak arama yapmasına izin verilir. Yırtıcı kaçan enerji | E | <1 iken HHO iyi çözümlerden çevrede yerel arama teşvik etmek eğilimindedir.

Keşif aşamasında, şahinin konumu rastgele konum ve diğer şahinler aracılığıyla denklem (17) kullanılarak güncellenir.

$$X(t+1)\begin{cases} X_k(t) - r_1|X_k(t) - 2r_2X(t)| & q \geq 0.5 \\ (X_r(t) - X_m(t)) - r_3(lb + r_4(ub - lb)) & q < 0.5 \end{cases} \quad (17)$$

X şahinin pozisyonu, X_k rastgele seçilen şahinin pozisyonudur, X_r avın pozisyonudur (tüm popülasyonda küresel en iyi çözüm), t mevcut yinelemedir, ub ve lb üst ve arama uzayının alt sınırları, r_1, r_2, r_3, r_4 ve q [0, 1] 'deki beş bağımsız rasgele sayıdır. X_m, mevcut şahin popülasyonunun ortalama pozisyonudur ve denklem (18) kullanılarak hesaplanabilir.

$$X_m(t) = \frac{1}{N}\sum_{n=1}^{N} X_n(t) \quad (18)$$

burada X_n, popülasyondaki n. şahin ve N şahinlerin sayısıdır.

Kullanım aşamasında şahinin konumu dört farklı duruma göre güncellenir. Bu davranış, avın kaçan enerjisine (E) ve sürpriz sekmeden önce avın başarılı bir şekilde kaçma (r <0.5) veya başarılı bir şekilde kaçamama (r ≥ 0.5) şansına göre manipüle edilir.

1.Yumuşak kuşatma, r ≥ 0.5 ve | E | ≥ 0.5. Bu durumda şahin, denklem (19) ile konumunu günceller.

$$X(t+1) = \Delta X(t) - E|JX_r(t) - X(t)| \quad (19)$$

burada E, avın kaçan enerjisi, X şahinin pozisyonu, t mevcut yineleme, ΔX, avın pozisyonu ile mevcut şahinin arasındaki fark ve J sıçrama gücüdür. ΔX ve J sırasıyla denklem (20) ve (21) ile hesaplanır.

$$\Delta X(t) = X_r(t) - X(t) \quad (20)$$

$$J = 2(1 - r_5) \quad (21)$$

burada r_5, [0, 1] 'de her yinelemede rastgele değişen rastgele bir sayıdır.

2.Sert kuşatmada HHO, r ≥ 0.5 ve | E | <0.5. Bu durumda şahin pozisyonu aşağıdaki şekilde güncellenir.

$$X(t+1) = X_r(t) - E|\Delta X(t)| \quad (22)$$

burada X şahinin pozisyonu, X_r avın pozisyonudur, E avın kaçan enerjisidir ve ΔX, avın konumu ile mevcut şahinin arasındaki farktır.

3.Aşamalı hızlı dalışlarla yumuşak kuşatma r <0.5 ve | E | ≥ 0.5. Şahin, avı rekabetçi bir şekilde yakalamak için kademeli olarak mümkün olan en iyi dalışı seçer. Bu durumda, şahinin yeni pozisyonu denklem (23) ve (24) ile hesaplanır.

$$Y = X_r(t) - E|JX_r(t) - X(t)| \quad (23)$$

$$Z = Y + \alpha * Levy(D) \quad (24)$$

burada Y ve Z yeni oluşturulan iki şahin, E kaçan enerjidir, J sıçrama gücüdür, X şahinin pozisyonudur, t mevcut yinelemedir, X_r avın pozisyonudur, D toplam sayıdır. α, D boyutuna sahip rastgele bir vektördür ve denklem (25)' de sunulan $Levy$ fonksiyonu ile hesaplanır.

$$Levy(x) = 0.01x\frac{\mu x\sigma}{|v|^{1/\beta}} \quad (25)$$

burada μ, v normal dağılımdan üretilen iki bağımsız rastgele sayıdır ve σ şu şekilde tanımlanır:

$$\sigma = \left(\frac{\tau(1+\beta)x\sin(\frac{\pi\beta}{2})}{\tau(\frac{(1+\beta)}{2})x\beta x2^{(\frac{\beta-1}{2})}}\right)^{1/\beta} \quad (26)$$

burada β 1,5'e ayarlanmış varsayılan bir sabittir. Bu aşamada şahin pozisyonu denklem (27) 'te olduğu gibi güncellenir.

$$X(t+1) = \begin{cases} Y \ if\ F(Y) < F(X(t)) \\ X \ if\ F(Z) < (F(X(t)) \end{cases} \quad (27)$$

F (.) uygunluk fonksiyonudur, Y ve Z denklemler (23) ve (24) 'den elde edilen iki yeni çözümdür.

4. Aşamalı hızlı dalışlarla sert kuşatma ise, r <0.5 ve | E | <0.5. Bu durumda, aşağıdaki gibi iki yeni çözüm üretilir.

$$Y = X_r(t) - E|JX_r(t) - X_m(t)| \quad (28)$$

$$Z = Y + \alpha * Levy(D) \quad (29)$$

burada E kaçan enerji, J sıçrama gücü, X_m şahinlerin mevcut popülasyondaki ortalama konumu, t mevcut yineleme, X_r avın konumu, D toplam boyut sayısı, α , D boyutuna sahip rastgele vektör ve $Levy$, uçuş işlevidir. Daha sonra şahinin konumu denklem (30) ile güncellenir:

$$X(t+1) = \begin{cases} Y \ if\ F(Y) < F(X(t)) \\ X \ if\ F(Z) < (F(X(t)) \end{cases} \quad (30)$$

Transfer fonksiyonunu ikili HSO (IHSO) entegre ederek, algoritma aramayı ikili arama alanında gerçekleştirebilir (Too vd., 2019). IHSO 'da şahinin konumu iki aşamada güncellenir. İlk aşamada, IHSO şahinin konumunu günceller $X_i^d(t)$ IHSO'ya benzer yeni bir konum $\Delta X_i^d(t)$. Yeni konumun $\Delta X_i^d(t)$ sürekli biçimde sunulduğuna dikkat edilir. İçinde ikinci aşamada, yeni konumu olasılık değerine dönüştürmek için S-şekilli transfer fonksiyonu kullanılır. Şahinin yeni konumu daha sonra denklem (31) kullanılarak güncellenir. Bu şekilde şahin pozisyonu ikili formda ifade edilebilir. IHSO algoritması S-şekilli transfer fonksiyonu ile şahinlerin yeni pozisyonlarını aşağıda verilen denklem ile günceller:

$$X_i^d(t+1) = \begin{cases} 1 & if\ rand(0,1) < T(\Delta X_i^d(t+1))) \\ 0 & diğer\ durumda \end{cases} \quad (31)$$

burada $T(x)$ S-şekilli transfer fonksiyonu, $rand$ [0,1] 'de rastgele bir sayıdır, X şahinin pozisyonudur, i popülasyondaki şahinin sırasıdır, d boyut ve t mevcut yinelemedir. İkili Harris Şahin Optimizasyon (IHSO) algoritmasının kaba kodu Algoritma 2 de verilmiştir.

Algoritma 2. İkili Harris Şahin Optimizasyon algoritması

Inputs: N ve T değerlerini belirle
N şahin için X_i değerlerini oluştur.
for (t = 1 to T)
  Şahinlerin fitness değerlerini hesapla
  En iyi çözümü X_r olarak tanımla
  for (i = 1 to N)
    E_0 ve J değerlerini Denklem (16) ve (21) ile hesapla
    E değerini Denklem (15) ile güncelle
    // Keşif aşaması //
      if (|E| ≥ 1)
      Şahinlerin pozisyonlarını Denklem (17) ile güncelle
     Transfer fonksiyonu ile olasılıkları hesapla
     Şahinlerin yeni pozisyonlarını Denklem (31) ile güncelle
   // Sömürü aşaması //
    elseif (|E| < 1)
      // Yumuşak kuşatma //
      if (r ≥ 0.5) and (|E| ≥ 0.5)
         Şahin pozisyonlarını Denklem (19) ile güncelle
         Transfer fonksiyonu ile olasılıkları hesapla
         Şahinlerin yeni pozisyonlarını Denklem (31) ile güncelle
        // Sert kuşatma //
       elseif (r ≥ 0.5) and (|E| < 0.5)
           Şahin pozisyonlarını Denklem (22) ile güncelle
           Transfer fonksiyonu ile olasılıkları hesapla
           Şahinlerin yeni pozisyonlarını Denklem (31) ile güncelle
       // Aşamalı hızlı dalışlarla yumuşak kuşatma //
      elseif (r < 0.5) and (|E| ≥ 0.5)
         Şahin pozisyonlarını Denklem (27) ile güncelle
         Transfer fonksiyonu ile olasılıkları hesapla
         Şahinlerin yeni pozisyonlarını Denklem (31) ile güncelle
       // Aşamalı hızlı dalışlarla sert kuşatma //
      elseif (r < 0.5) and (|E| < 0.5)
         Şahin pozisyonlarını Denklem (30) ile güncelle
         Transfer fonksiyonu ile olasılıkları hesapla
         Şahinlerin yeni pozisyonlarını Denklem (31) ile güncelle
      end if
  end if
 next i
 X_r Değerini daha iyi değer varsa güncelle
next t
Output: Küresel en iyi çözüm

2.3. Fitness Fonksiyon Tasarımı

Sınıflandırma tahmini yapılırken, sınıflandırma hatası genellikle uygunluk değeri olarak kabul edilir. FS için sadece sınıflandırma hatasını kullanılması her zaman kaliteli bir çözüm olmayabilir. Sınıflandırma hatasına ek olarak, uygunluk fonksiyonunda seçilen alt özelliklerin toplam özellik sayısına oranı da kullanılır. Böylece sınıflandırma hatası olan iki çözümün daha az özelliğine sahip olan çözüm seçilir. Bu çalışmada, denklem (32) uygunluk işlevi olarak kullanılır.

$$fitness = \lambda * Hata + (1 - \lambda) * |Sel\_F|/|All\_F| \quad (32)$$

Hata oranı, 10-Fold çapraz doğrulamanın ortalama sınıflandırma hatasıdır, $|Sel\_F|$ seçili özelliklerin sayısıdır ve $|All\_F|$ veri kümesindeki toplam özelliklerdir. λ katsayıları 0,90'dır. Bu çalışmada, sınıflandırma hatasını hesaplamak için KNN sınıflayıcı kullanılmıştır.

2.4. Metin Sınıflama için Veri Kümesi Oluşturma

2.4.1. Veri Küme Oluşturma Ön-işleme Süreci

Web sitelerinden toplanan veri kümesinde gürültülü veriler ve HTML / CSS etiketleri bulunur. Bu ham verilerin kullanılması için uygun bir biçimde düzenlenmesi gerekmektedir. Web sayfasından çekilen haber içeriklerine aşağıdaki adımlara göre ön işlem uygulanır:

- Noktalama işaretleri, kısa çizgiler, sayılar ve rakamlar kaldırılır.
- 2 veya daha az karakter içeren kelimelerle, 15 veya daha fazla karakter içeren kelimeler kaldırılır.
- Zamirler ve edatlar (a, an, the ...) gibi sözcükler temizlenir.
- Kelimenin köklerini oluşturma süreci (-ed, -ing…) yapılır.

Genel olarak bilinen metin sınıflandırması alanında ağırlıklandırma terimlerinde terim-frekansı ve ters belge sıklığı (TF.IDF) formülü kullanılır. Bu terim ağırlıklandırmayı hesaplamak için kullanılan etkili ve basit bir istatistiksel yaklaşımdır. TF, bir belgede görünen özellik kelimelerinin ağırlığını t_i belirli bir d_j metninde özelliğin bulunma sayısını temsil eder. Bir özelliğin (t_i) belge sıklığı (DF), t_i'nin en az bir kez görüldüğü belgedeki metin sayısıdır. Belirtilen özelliğin veya t_i teriminin ters belge frekansı (IDF), denklem (33) ile hesaplanır.

$$. IDF(t_i) = \log \frac{D}{DF(t_i)} \quad (33)$$

burada d_j, veri kümesindeki metinlerin sayısını ifade eder. TF.IDF kullanılarak, belirli bir d_j metnindeki t_i teriminin ağırlığı denklem (34) ile hesaplanır.

$$TF.IDF(t_i, d_j) = TF(t_i, d_j)\ X\ IDF(t_i) \quad (34)$$

2.4.2. Veri Kümesi Bilgileri

AstArc veri kümesi (Aktaş ve Kılıç, 2021)'deki çalışma performansını ölçmek için oluşturulmuştur. SScience veri kümesi önerilen algoritmaların 2'den daha fazla sınıfa sahip bir veri kümesinde performans analizi yapmak için oluşturulmuştur. Her iki veri kümesi de SCI-NEWS web sitesindeki (SCI News, 2021) haberler kullanılarak oluşturulmuştur. AstArc veri kümesi Astronomi ve Arkeoloji haberlerinden oluşturulurken, SScience veri kümesi 9 farklı (Fizik, Biyoloji, Astronomi ..vs) sınıftaki haberlerden oluşmaktadır. Her iki veri kümesi ReliefF algoritması kullanılarak 500 öznitelik sayısına indirgenerek sırası ile AstArc-ReliefF ve SScience-ReliefF olarak adlandırılmıştır. Relief, Kira ve Rendell tarafından 1992 yılında geliştirilen ve özellik etkileşimlerine duyarlı olan özellik seçimine filtre yöntemi yaklaşımını benimseyen bir algoritmadır (Kira vd., 1992; Kira vd., 1992). Relief, her özellik için bir özellik puanı hesaplar ve

bu puan, özellik seçimi için en yüksek puanı alan özellikleri sıralamak ve seçmek için uygulanabilir. Kononenko ve ark. Relief için öklit uzaklığı yerine Manhattan uzaklığını kullanıp güncellemeler yaparak ReliefF'i önermişlerdir (Kononenko vd.,1997). Veri kümesi detayları Tablo 1'de gösterilmektedir

Tablo 1. Veri Kümesi Bilgileri

Veriseti	Kayıt Sayısı	Öznitelik Sayısı	Sınıf Sayısı	Öznitelik Tipi	Kayıp Değer
AstArc	100	6224	2	Tamsayı, Gerçek	Yok
AstArc-ReliefF	100	500	2	Tamsayı, Gerçek	Yok
SScience	450	15969	9	Tamsayı, Gerçek	Yok
SScience-ReliefF	450	500	9	Tamsayı, Gerçek	Yok

3. Araştırma Sonuçları ve Tartışma

Bu bölümde IGKO ve IHSO algoritmalarının performansı bölüm 2.4'de tanıtılan veri kümeleri üzerinde KNN sınıflandırıcısı ile analiz edilmektedir. Deneysel sonuçların analizi K-fold cross validation kullanılarak fitness, accuracy, f-score değerleri baz alınarak yapılmıştır. IHSO ve IGKO veri kümeleri üzerindeki istatistiksel sonuçlarını analiz etmesi için bağımsız olarak 10 kez çalıştırılır. Çalışmada kullanılan algoritma ve problem parametreleri Tablo 2'de verilmiştir.

Tablo 2. Algoritma ve problem Parametreleri

Parametre	Değeri
KNN sınıflayıcı için k	5
Fitness fonksiyonu için λ	0.90
Kurt ve Şahin popülasyonu sayısı	10
IGKO ve IHSO için maksimum iterasyon sayısı	100
K-Fold Cross validation için k değeri	10

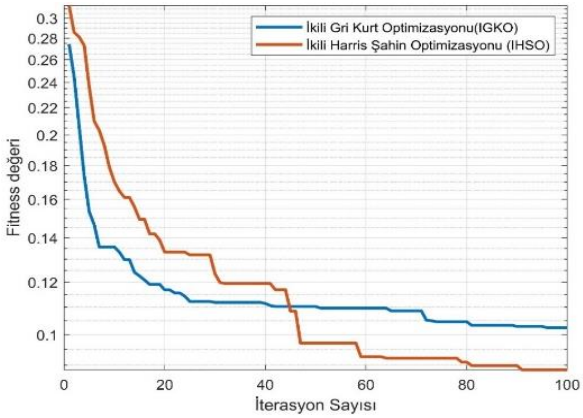
Tablo 3, veri kümesinde 10 bağımsız çalışma için KNN sınıflandırıcısının performansını ve tüm özellikleri (FS yöntemleri olmadan) göstermektedir.

Tablo 3. Özellik seçim metodu kullanılmadan KNN ile veriseti performans metrik değerleri

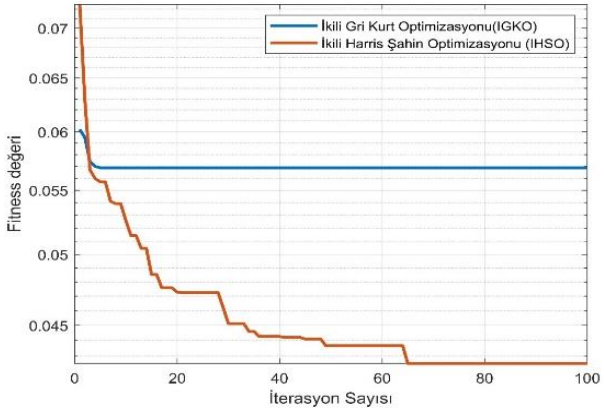
Veri Seti	Metrik	Accuracy	F-score	Özellik sayısı
AstArc	Ortalama	0.5100	0.3515	
	Standart Sapma	0.0094	0.0164	6224
AstArc-ReliefF	Ortalama	0.9640	0.9623	
	Standart Sapma	0.0143	0.0160	500
SScience	Ortalama	0.8309	0.1979	
	Standart Sapma	0.0019	0.0092	15969
SScience-ReliefF	Ortalama	0.9329	0.6949	
	Standart Sapma	0.0027	0.0096	500

Tablo 4'de IHSO ve IGKO algoritmalarının veri kümeleri üzerinde ortalama en iyi uygunluk değeri, accuracy, f-score ve seçilen öznitelik sayısını göstermektedir. Tablo 4 incelendiğinde uygunluk değeri bakımından IHSO algoritmasının en iyi değerlere ulaştığı görülmektedir. Bir diğer taraftan, IGKO algoritması bütün veri kümeleri için en iyi accuracy değerine ulaşmıştır IHSO algoritmasının accuracy değerleri IGKO'nun accuracy değerine oldukça yakınken öznitelik sayısı oldukça düşüktür

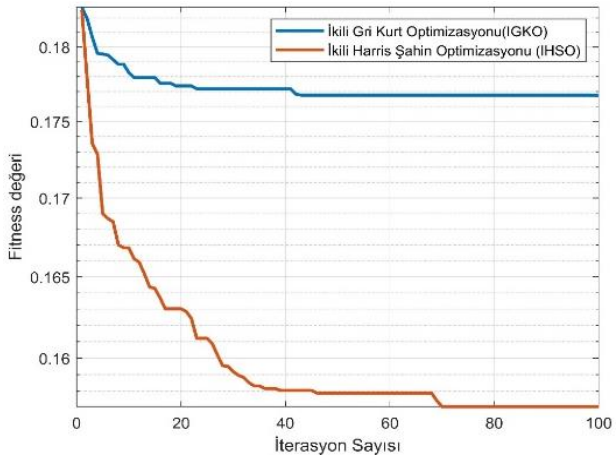
Şekil 1, 2, 3 ve 4'de sırasıyla *AstArc, AstArc- ReliefF, SScience* ve *SScience- ReliefF* veri kümelerinin IHSO ve IGKO algoritmalarının uygunluk yakınsama grafiği verilmiştir. Şekiller incelendiğinde, IHSO algoritması daha hızlı yakınsama ve düşük ortalama uygunluk değerlerine sahip olmuştur.



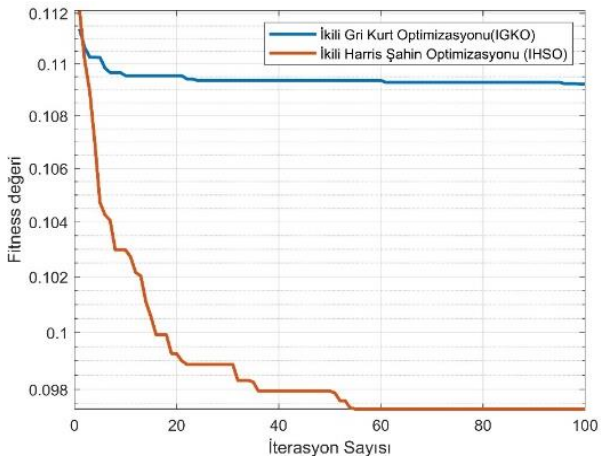
Şekil 1. AstArc veri kümesi için IHSO ve IGKO uygunluk değerleri



Şekil 2. AstArc- RelifF veri kümesi için IHSO ve IGKO uygunluk değerleri



Şekil 3. SScience veri kümesi için IHSO ve IGKO uygunluk değerleri



Şekil 4. SScience-RelifF veri kümesi için IHSO ve IGKO uygunluk değerleri

Tablo 4. IHSO ve IGKO ile Özellik seçimi performans metrik değerleri

Algoritma	**Veri Seti**	**Metrik**	**Uygunluk Değeri**	**Accuracy**	**F-score**	**Özellik Sayısı**
IGKO	*AstArc (Aktaş ve Kılıç, 2021)*	Ortalama.	0.1024	**0.9700**	0.9679	4696.70
		Standart Sapma	0.0343	0.0343	0.0394	471.75
	AstArc-ReliefF	Ortalama	0.0568	**1**	1	284.4
		Standart Sapma	0.0041	0.00	0.00	20.54
	SScience	Ortalama	0.1767	**0.8760**	0.3930	10417.20
		Standart Sapma	0.0023	0.0074	0.0431	950.18
	SScience-ReliefF	Ortalama	0.1092	**0.9464**	0.7544	305.2
		Standart Sapma	0.0038	0.0034	0.0154	15.26
IHSO	*AstArc*	Ortalama	**0.0885**	0.9580	0.9570	**3157.1**
		Standart Sapma	0.0280	0.0252	0.0260	662.94
	AstArc-ReliefF	Ortalama	**0.0425**	0.9950	0.9949	**190.1**
		Standart Sapma	0.0050	0.0070	0.0070	29.86
	SScience	Ortalama	**0.1570**	0.8636	0.3510	**5489.2**
		Standart Sapma	0.0058	0.0039	0.0241	699.52
	SScience-ReliefF	Ortalama	**0.0973**	0.9288	0.6760	**166.4**
		Standart Sapma	0.0026	0.0054	0.0281	23.75

4. Sonuç

Bu çalışmada İkili Harris Şahin Optimizasyon (IHSO) ve İkili Gri Kurt Optimizasyon (IGKO) algoritmaları kullanılarak web sayfasından elde edilen veri kümelerinin sınıflandırma doğruluğu öznitelik seçme problemi için değerlendirilmiştir. IHSO ve IGKO algoritmaları ile tüm öznitelikler arasında daha alakalı özellikler aramak için KNN sınıflandırıcı kullanılmıştır. Ayrıca bu çalışmada, IHSO ve IGKO algoritmalarının performansını doğrulamak için çeşitli bilim dallarından haberlerle ilgili web sayfalarından çıkarılan web belgeleri içeren 9 sınıflı yeni kıyaslama veri kümesi tanıtıldı. Veri kümesi hazırlama aşamalarında gereksiz özelliklerin sayısını azaltmak için kötü karakterler, rakamlar, durdurma kelimeleri, karakter sayısı 3'ten az ve 14'ten fazla olan kelimeler kaldırıldı. Bu işlem ile hesaplama süresi azaltılıp sınıflandırma performansı artırıldı. Veri kümesinde deneysel sonuçlara göre, IHSO algoritması IGKO algoritmasına göre daha iyi uygunluk değerine sahip olup daha az öznitelik seçerek hızlı yakınsama performansı göstermiştir. İleride yapılacak çalışmalarda, farklı sınıflandırıcı ve derin öğrenme yöntemleri araştırılacak olup meta-öğrenme yöntemleri ile çalışma genişletilecektir.

Kaynakça

Aggarwal, C. C. ve Zhai, C. (2012). A survey of text classification algorithms. In *Mining text data* (pp. 163-222). Springer, Boston, MA.

Aktaş, M. ve Kılıç, F. (2021) Binary Grey Wolf Optimizer using Archeology and Astronomy News for Text Classification, II. International Conference on Innovative Engineering Applications (CIEA' 2021).

Asgarnezhad, R., Monadjemi, S. A. ve Soltanaghaei, M. (2020) An application of MOGW optimization for feature selection in text classification. *The Journal of Supercomputing*, 1-34.

Chantar, H., Mafarja, M., Alsawalqah, H., Heidari, A. A., Aljarah, I. ve Faris, H. (2020) Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. *Neural Computing and Applications*, 32(16), 12201-12220.

Chen, H., Jiang, W., Li, C. ve Li, R. (2013) A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm. *Mathematical problems in Engineering*.

Das, S. (2001)Filters, wrappers and a boosting-based hybrid for feature selection. In *Icml*, 1, 74-81.

Deng, X., Li, Y., Weng, J. ve Zhang, J. (2019) Feature selection for text classification: A review. *Multimedia Tools and Applications*, *78*(3), 3797-3816.

Günal, S. (2012) Hybrid feature selection for text classification. *Turkish Journal of Electrical Engineering and Computer Science*, 20(Sup. 2); 1296-1311.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. ve Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, *97*, 849-872.

Jindal, R., Malhotra, R. ve Jain, A. (2015) Techniques for text classification: Literature review and current trends. *Webology*, 12(2).

Kononenko, I., Šimec, E. ve Robnik-Šikonja, M. (1997). Overcoming the myopia of inductive learning algorithms with RELIEFF. *Applied Intelligence*, *7*(1), 39-55.

Kira, K. ve Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. In *Aaai* (Vol. 2, pp. 129-134).

Kira, K. ve Rendell, L. A. (1992). A practical approach to feature selection. In *Machine learning proceedings 1992* (pp. 249-256). Morgan Kaufmann.

Labani, M., Moradi, P., Ahmadizar, F. ve Jalili, M. (2018). A novel multivariate filter method for feature selection in text classification problems. *Engineering Applications of Artificial Intelligence*, *70*, 25-37.

Lee, J., Park, J., Kim, H. C. ve Kim, D. W. (2019). Competitive particle swarm optimization for multi-category text feature selection. *Entropy*, 21(6), 602.

Liu, H. ve Setiono, R. (1997). Feature selection and classification-a probabilistic wrapper approach. In *Proceedings of 9th International Conference on Industrial and Engineering Applications of AI and ES*, January, p.419-424.

Manoj, R. J., Praveena, M. A. ve Vijayakumar, K. (2019) An ACO–ANN based feature selection algorithm for big data. *Cluster Computing*, 22(2), 3953-3960.

Marie-Sainte, S. L. ve Alalyani, N. (2020). Firefly algorithm based feature selection for Arabic text classification. *Journal of King Saud University-Computer and Information Sciences*, 32(3), 320-328.

Mirjalili, S., Mirjalili, S. M. ve Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, *69*, 46-61.

Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y. ve Wang, Z. (2007). A novel feature selection algorithm for text categorization. *Expert Systems with Applications*, 33(1), 1-5.

The SCI-NEWS website. (2021). (online), Available: http://www.sci-news.com/

Too, J., Abdullah, A. R. ve Mohd Saad, N. (2019). A new quadratic binary harris hawk optimization for feature selection. *Electronics*, *8*(10), 1130.

Wah, Y. B., Ibrahim, N., Hamid, H. A., Abdul-Rahman, S. ve Fong, S. (2018). Feature Selection Methods: Case of Filter and Wrapper Approaches for Maximising Classification Accuracy. *Pertanika Journal of Science & Technology*, 26(1), 329-340.

Xing, E. P., Jordan, M. I. ve Karp, R. M. (2001). Feature selection for high-dimensional genomic microarray data. In *Icml*, 1, 601- 608.