

POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)
URL: http://dergipark.org.tr/politeknik



Zeki optimizasyon tabanlı destek vektör makineleri ile diyabet teşhisi

Diabetes diagnosis with intelligent optimization based support vector machines

***Yazar(lar) (Author(s))**: Utku KÖSE*

ORCID: 0000-0002-9652-6415

Bu makaleye şu şekilde atıfta bulunabilirsiniz(To cite to this article): Köse U., "Zeki optimizasyon tabanlı destek vektör makineleri ile diyabet teşhisi", ***Politeknik Dergisi***, 22(3): 557-566, (2019).

Erişim linki (To link to this article): http://dergipark.org.tr/politeknik/archive

DOI: **10.2339/politeknik.418851**

Zeki Optimizasyon Tabanlı Destek Vektör Makineleri ile Diyabet Teşhisi

Araştırma Makalesi / Research Article

Utku KÖSE*

Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Süleyman Demirel Üniversitesi, Türkiye

(Geliş/Received : 28.02.2018 ; Kabul/Accepted : 30.03.2018)

ÖZ

Yapay Zekâ, farklı gerçek dünya problemlerine etkin bir şekilde uygulanabilen ve geleceğimizi uzun bir süredir şekillendiren, önemli bilim alanlarından birisidir. Uygulandığı problem türleri çok çeşitli olmakla birlikte, bunlardan en dikkat çekenlerinden birisi de medikal teşhistir. Açıklamalardan hareketle bu çalışmanın amacı, zeki optimizasyon tabanlı Destek Vektör Makineleri'ni (DVM) kullanarak diyabet teşhisi gerçekleştirmektir. Bu bağlamda, günümüz güncel zeki optimizasyon algoritmalarından beş tanesi Gaussian-RBF kernel fonksiyonu kullanan bir non-lineer DVM'yi optimize etmek amacıyla kullanılmıştır. Elde edilen bulgular, farklı algoritmalar ile kurulmuş hibrit sistemlerin, farklı düzeyde başarımlar gösterdiğini ancak genel anlamda zeki optimizasyon-DVM yaklaşımıyla diyabet teşhisinde yüksek oranda tutarlı sonuçlar elde edilebildiğini ortaya koymuştur. Çalışma bu yönüyle izlenen yaklaşımın Yapay Zekâ tabanlı teşhis açısından önemli bir potansiyele sahip olduğunu da teyit etmektedir.

Anahtar Kelimeler: Zeki optimizasyon, destek vektör makineleri, diyabet teşhisi, medikal teşhis, yapay zekâ.

Diabetes Diagnosis with Intelligent Optimization Based Support Vector Machines

ABSTRACT

Artificial Intelligence is one of the most important scientific fields that can be applied effectively in different real world problems and has been shaping our future for a long time. While there are various types of problems in which it is applied, medical diagnosis is one of the most remarkable one among them. Moving from the explanations, objective of this study is to realize diabetes diagnosis by using intelligent optimization based Support Vector Machines (SVM). In this context, five ones of today's recent intelligent optimization algorithms were used for optimizing a non-linear SVM, which is using a Gaussian-RBF kernel function. Obtained findings showed that hybrid systems formed with different algorithms show different-level success but in general, good level accurate results can be achieved via intelligent optimization-SVM approach. Hereby, the study also confirms that this followed approach has a significant potential for Artificial Intelligence based diagnosis.

Keywords: Intelligent optimization, support vector machines, diabetes diagnosis, medical diagnosis, artificial intelligence

1. GİRİŞ (INTRODUCTION)

Bilimsel ve teknolojik gelişmeler, beraberinde birçok farklı sonucu ortaya çıkarmaktadır. Söz konusu sonuçlar, gelişmelerle ilişkili bağlı teknolojilerin ivme kazanması ya da bilimsel literatüre yeni alanların eklenmesi yönündeki durumları içerebilmekte ve özellikle birbirini destekleyici teknolojilerin karşılıklı değişimiyle beraber süreç ilerleyebilmektedir. Bu bağlamda, hızlı yükselişlerini yirminci yüzyılın ortalarından başlayarak, günümüze kadar sürdüren ve etkilerini insan hayatında yoğun bir şekilde hissettiren; bilgisayar, elektronik ve iletişim teknolojileri gibi teknolojiler, geleceğe ışık tutan bilim alanlarının ortaya çıkmasına da sebep olmuştur. Yapay Zekâ, söz konusu bu alanlardan birisi olarak dikkat çekmektedir.

Genel olarak, insa n ve doğa esinli, matematiksel ve mantıksal yaklaşımlarla gerçek dünya problemlerini başarılı bir şekilde çözümleyen Yapay Zekâ [1, 2], esnek yapısı ve çok disiplinli uygulanabilirliği neticesinde yükselişini hızlı bir şekilde sürdürmektedir [1-3]. Burada özellikle geleneksel yaklaşım, yöntem ve tekniklerle çözümlenemeyen ya da istendik düzeylerde çözümlenemeyen problemler, Yapay Zekâ ile oluşturulan zeki sistemler aracılığıyla başarılı bir şekilde çözüme kavuşturulabilmektedir. Özellikle tahmin, teşhis, örüntü tanıma, optimizasyon, adaptif kontrol, yorumlama ve bilgi işleme – modelleme tabanlı gerçek dünya problemlerine yönelik başarısı gözler önünde olan Yapay Zekâ [2, 4-9], artık günlük hayatta da kendini yakından hissettirir hale gelmiştir. Mühendislik tabanlı alanlar, eğitim, ekonomi, askeriye gibi alanların yanında kuşkusuz ki sağlık da, Yapay Zekâ'nın ön planda kullanıldığı alanlardan birisi olmuştur. Sağlık alanında birçok farklı çözüm ve uygulama kapsamı söz konusu olmakla birlikte, ilgili literatür ve mevcut gelişmeler dikkate alındığında, Yapay Zekâ tabanlı yaklaşımların yaygın bir biçimde kullanıldığı uygulamalardan birisinin de medikal teşhisler olduğunu ifade edebiliriz [10-16].

**Sorumlu Yazar (Corresponding Author)*
e-posta : utkukose@sdu.edu.tr

Bu çalışmanın amacı, zeki optimizasyon tabanlı Destek Vektör Makineleri'ni (DVM) kullanarak diyabet teşhisi gerçekleştirmektir. Bu bağlamda, günümüz yaygın zeki optimizasyon algoritmalarından beş tanesi (ikisi temel, üçü güncel algoritmalardan olmak üzere): Parçacık Sürü Optimizasyonu (PSO), Genetik Algoritma (GA), Guguk Kuşu Araması (GKA), Bakteriyel Yiyecek Arama Optimizasyonu (BYAO) ve Çiçek Tozlaşma Optimizasyonu (ÇTO), Gaussian-RBF kernel fonksiyonu kullanan bir non-lineer DVM'yi optimize etmek amacıyla kullanılmıştır. Literatürde Gaussian-RBF kernel fonksiyonunun farklı parametreleri yönünden optimize edildiği farklı çalışmalar mevcuttur [17-21]. Bu çalışmada ise, Gaussian-RBF kernel fonksiyonu için önem arz eden sigma (σ) parametresi değerinin belirlenmesi adına ilgili zeki optimizasyon algoritmalarının devreye sokulduğu bir hibrit sistem düzeni oluşturulmuştur. Sistem kapsamındaki amaç, algoritmaların optimizasyon yönelimli mekanizmalarının optimum parametre belirlemek için iteratif bir şekilde çalıştırılmasıdır. Çalışmadaki diyabet teşhisi yaklaşımı, literatürde kullanılagelen Pima Yerlileri veri seti dikkate alınarak değerlendirilmiştir.

Çalışmanın konusu ve amaçları doğrultusunda metnin ilerleyen bölümleri şu şekilde organize edilmiştir: Bir sonraki bölüm altında diyabet teşhisine yönelik uygulanan yöntem ve işe koşulan tekniklere dair detaylar vurgulanmıştır. Bu bağlamda öncelikli olarak diyabet veri setine ilişkin temel bilgilere, işe koşulan tekniklere ve yine tasalanan diyabet teşhisi yaklaşımına dair temel işlem adımlarına değinilmiştir. İkinci bölümü takiben, üçüncü bölüm altında diyabet teşhisi süreciyle elde edilen bulgular sunulmuş ve metin, çalışmayla ulaşılan sonuçlar, bu bağlamda genel tartışmalar ve düşünülen gelecek çalışmalardan bahsedilerek soma erdirilmiştir.

2. MATERYAL VE METOD (MATERIAL and METHOD)

Çalışmada gerçekleştirilen teşhis yaklaşımı ile elde edilen bulgulara değinmeden önce, zeki optimizasyon algoritmaları ve Destek Vektör Makineleri ile kurulan hibrit sistemlere dair detayları açıklamak gerekmektedir. Bu bağlamda bölüm içerisinde araştırmaya konu olan veri setine, Destek Vektör Makineleri tekniğine ve zeki optimizasyon kavramı ile işe koşulan algoritmalar (bu çalışmada kullanılan Sürü Zekâsı teknikleri – algoritmaları) açıklanmıştır.

2.1. Teşhis Amaçlı Diyabet Veri Seti (Diagnosis Purpose Diabetes Data Set)

Bu çalışmada amaçlanan diyabet teşhisi için, literatürde sıklıkla kullanılan Pima Yerlileri veri seti kullanılmıştır. Söz konusu veri seti, UCI Machine Learning Repository adlı veri havuzu Web sitesinden indirilebilmektedir [22]. Veri setinin alındığı hedef kişilere ve veri seti geneline dair bazı bilgiler şu şekildedir [23]:

- Kişiler, Arizona civarında yaşayan, Pima Yerlileri'dir.
- Veriler, en küçüğü 21 yaşındaki kişilerden alınmıştır.
- Veri setinde toplamda 768 örnek ve 8 nitelik vardır.
- Her bir örnekte ilgili kişinin diyabet pozitif ya da diyabet negatif (hasta ya da değil) olmasına göre 1 adet çıkış değeri vardır. Bu bağlamda, veri setindeki niteliklerin neler olduğuna dair açıklamalar Çizelge 1'de sunulmuştur [23].
- Veri setindeki 768 örnekten 500 tanesi diyabet negatif kişilere, geriye kalan 268 tanesi ise diyabet pozitif kişilere aittir.

Çizelge 1. Hedef diyabet veri setindeki nitelikler (Features in the target diabetes dataset) [23].

Nitelik No.	Niteliğin Açıklaması
1	Hamilelik Sayısı
2	Plazma Glukoz Konsantrasyonu
3	Diastolik Kan Basıncı
4	Kol Kası Cilt Kıvrım Kalınlığı
5	2-h serum insülin
6	Vücut Kütle İndeksi
7	Diyabet Soyağacı Fonksiyonu
8	Yaş

İlerleyen alt-bölümler çalışmada kullanılan tekniklerden ve uygulanan yaklaşımdan kısaca bahsedilmiştir.

2.2. Destek Vektör Makineleri (Support Vector Machines)

Bir Yapay Zekâ (Makine Öğrenmesi) ve Veri Madenciliği tekniği olarak Destek Vektör Makineleri (DVM), Vapnik tarafından literatüre kazandırılmış ve bu bağlamda, regresyon ve sınıflandırma tabanlı problemler için önerilmiştir [24, 25]. Söz konusu problem yaklaşımlarıyla farklı gerçek dünya problemlerine sunduğu etkin çözümler neticesinde, çok disiplinli uygulamalar kapsamında oldukça popüler bir hale gelmiştir [26-31]. Kısaca ifade etmek gerekirse DVM, Lagrange çarpanlarından yola çıkarak, farklı iki sınıfı ayıran optimum ayırıcıyı bulmayı amaç edinen, bunun için de eldeki örnek verilerden (eğitim seti) faydalanan bir tekniktir [32, 33].

DVM'yi üzerinde çalışılan verilerin lineer veya lineer olmama durumuna göre iki farklı türe ayırmamız mümkündür [34-37]:

- **Lineer DVM:** Lineer bir şekilde ayrılabilen (sınıflandırılabilen) verilere uygulanan temel DVM türüdür. Burada, $(x_1, \ldots\ldots, x_n)$ hedef veri seti, v normal vektör ve ed eşik değeri olmak üzere veriler Eşitlik 1'deki hiper-düzlem ile ayrılmaktadır.

$$h(x) = v^T x + ed = 0 \quad (1)$$

Hiper-düzlem'den yola çıkılacak olursa, $s_i \in (-1,+1)$ olmak üzere, dikkate alınan bir veri $v^T x_i + ed \geq +1$ ise Sınıf 1, $v^T x_i + ed \leq -1$ ise Sınıf 2'ye ait olmaktadır [35, 36].

- **Non-lineer DVM:** Hedef veriler, lineer olarak birbirinden ayrılamaz ise non-lineer DVM kullanılmaktadır. Bu durumda, eğitim verileri birbirlerinden ayrılabilecek daha yüksek uzayda haritalandırılmaktadır [37]. Söz konusu haritalandırma aşamasında kernel fonksiyon adı verilen çeşitli özel fonksiyonlar işe koşulmakta ve *n* boyuttaki veri kümesi daha yüksek boyuta taşınarak, ilgili boyutta lineer sınıflandırma yoluna gidilmektedir. Literatüre bakıldığında, DVM'nin uygulandığı problemdeki başarımını doğrudan etkileyen farklı kernel fonksiyonlar bulunmaktadır. Bu fonksiyonlardan sık kullanılan üç tanesi şu şekildedir [34]:

  Lineer fonksiyon: $K(x_i, x_j) = x_i^T x_j$ (2)

  Gaussian-RBF fonksiyon: $K(x_i, x_j) = e^{\frac{||x_i - x_j||^2}{2\sigma^2}}$ (3)

  Polinomal: $K(x_i, x_j) = (1 + x_i^T x_j)^p$ (4)

  Eşitliklerden de anlaşılacağı üzere, söz konusu kernel fonksiyonlarda değeri önem arz eden çeşitli parametreler bulunmaktadır. Bu parametrelerin alacakları değerler, DVM'nin uygulandığı problemi en iyi şekilde çözebilmesi adına oldukça önemlidir.

2.3. Zeki Optimizasyon ve Çalışmaya Konu Algoritmalar (Intelligent Optimization and the Algorithms Related to the Study)

Yapay Zekâ'nın geleneksel çözüm yaklaşımlarına olan üstünlüğü, bu bilim alanı kapsamındaki problem türlerinin kısa sürede artmasına sebep olmuştur. Çalışmanın Giriş bölümü altında da ifade edildiği üzere; birçok farklı problem türü Yapay Zekâ'nın kapsamı içerisindedir. Optimizasyon süreçleri de bu problem türleri içerisinde yer alan ve zamanla Yapay Zekâ altında alt ilgi alanlarının oluşmasına sebep olan bir problem türü olarak karşımıza çıkmaktadır.

Genel olarak bakıldığında zeki optimizasyon, belirli kıstaslar dahilinde, parçacık temelli olarak oluşturulmuş ve iteratif bir biçimde çalışabilen Yapay Zekâ algoritmaları ile yerine getirilmektedir [38, 39]. Mevcut gelişmeler de dikkate alındığında, zeki optimizasyon algoritmalarının (tekniklerinin), doğadan sıklıkla esinlendiğini ve doğadaki canlıların veya dinamiklerin temelinde tasarlandığını görebiliriz [38-40]. Özellikle Sürü Zekâsı (Swarm Intelligence) adı verilen alt ilgi alanı, bu bağlamda sürü içerisindeki etkileşim ve hareketlere odaklanarak optimizasyon süreçlerini içeren algoritmaları kapsamı içerisine almaktadır [41, 42]. Tipik bir zeki optimizasyon algoritması, matematiksel modellemesi yapılmış hedef optimizasyon problemi için *N* sayıda parçacığın aynı anda işe koşulduğu ve optimum sonuç için arama – hesaplama gibi süreçleri gerçekleştirdiği bir yapı üzerine kurulmaktadır. Burada önemli olan diğer bir husus da, zeki optimizasyon algoritmalarının, Makine Öğrenmesi tekniklerinden farklı olarak; eğitim süreçlerinden geçmeyen, doğrudan optimizasyon odaklı çözüm yapılarına karşılık gelmesidir.

Literatürde her ne kadar çok sayıda algoritma olsa da, bu çalışmada Sürü Zekâsı kapsamına giren, beş farklı algoritmanın diyabet teşhisi uygulamalarına katılması sağlanmıştır. Bu noktada, algoritmaların ikisi temel nitelikte, üçü ise güncel durumdaki algoritmalardır. Söz konusu algoritmalara ilişkin temel açıklamalar, ilerleyen alt-bölümler altında yapılmıştır.

2.3.1. Parçacık sürü optimizasyonu (Particle swarm optimization)

Parçacık Sürü Optimizasyonu (PSO), zeki optimizasyon amaçlı algoritmaların en temellerinden birisidir. Bu bağlamda, diyabet teşhisi problemi kapsamında çalışmaya dâhil edilen ilk temel algoritma olmuştur. PSO genel anlamda doğadaki kuş ve balık gibi hayvanların sürüler içerisindeki yiyecek arama hareketlerinden yola çıkarak geliştirilmiş bir algoritmadır [43, 44]. Algoritmada, çözüm uzayına dağılmış *N* parçacığın optimumu bulma çabasına yönelik iteratif adımlar yerine getirilmekte ve bu noktada her parçacığın bulunduğu konum (position) ve hız (velocity) gibi parametreler genel hareketlerin belirleyicisi olmaktadır. Hareketler neticesinde elde edilen parçacık konumları karşılık geldikleri değişkenler kapsamında problemin amaç fonksiyonu değer(ler)ini hesaplamakta kullanılmaktadır. Parçacıkların konum değişiklikleri sırasında her parçacığın o ana kadar elde ettiği en iyi kişisel konum (optimum değeri sağlayan) ve çözüm sürecinde o ana kadar elde edilmiş olan küresel konum değeri (küresel optimum değeri sağlayan konum) dikkate alınmakta ve parçacıklar bu değerler etkisinde küresel optimumu sağlayan parçacığı da takip ederek çözüme ulaşmaya çalışmaktadır [44, 45]. PSO'nun algoritmik adımlarına dair detaylar, [45] üzerinde ifade edilmektedir. Kaba-kod olarak ise Çizelge 2'de verilmiştir [38].

2.3.2. Genetik algoritma (Genetic algorithm)

Evrim Teorisi'nden esinlenerek işe koşulan çaprazlama, mutasyon ve nihayetinde doğal seleksiyon gibi çeşitli mekanizmaları temel alarak geliştirilmiş olan Genetik Algoritma (GA), bu çalışmaya dâhil edilen ikinci temel algoritma olmuştur. Genel hatlarıyla GA, çeşitli kodlama yaklaşımlarıyla (her bir kod gen anlamı gelmektedir) oluşturulan *N* adet parçacıktan oluşmaktadır (Her bir parçacık, genlerden oluşan, kromozom yapısı altında temsil edilen bireylerdir) [46, 47]. İteratif algoritmik adımlar, hesaplanan amaç fonksiyonu değerine göre seçilen, nispeten iyi durumdaki parçacıkların karşılıklı çaprazlanması (karşılık parçacık kodlarının-genlerinin takası) ve hatta elde edilen yeni parçacıkların bazılarına mutasyon (bazı kodların-genlerin değiştirilmesi) uygulanması suretiyle yenilenen popülasyon ile optimizasyon sürecinin gerçekleştirilmesine dayanmaktadır. GA'ya ilişkin adımların detayları [48]'de

açıklanmaktadır. Kaba-kod ise Çizelge 3'deki gibidir [38].

Çizelge 2. Parçacık Sürü Optimizasyonu algoritması kaba-kodu (Pseudo-code of the Particle Swarm Optimization algorithm) [38].

- **1. Adım (Kurulum):** *N* parçacığı çözüm uzayına rastgele dağıt ve her parçacık için başlangıç *Hız* (*v: velocity*) değerini ata. Her parçacığın başlangıç konumunu aynı zamanda kişisel en iyi konumu yap. Algoritma parametreleri değerlerini ata. Çözülecek probleme yönelik düzenlemeleri yap.
- **2. Adım:** İteratif adımlar (her parçacık ve her amaç fonksiyon boyutu için):
- **2.1. Adım:** Parçacığın konumuna göre amaç fonksiyon değeri hesapla.
- **2.2. Adım:** Parçacığın konumu o ana kadarki kişisel en iyi konumundan daha iyi ise, güncelle. Konum aynı zamanda küresel en iyi konumdan daha iyi ise, küresel en iyi konumu da güncelle.
- **2.3. Adım:** Parçacığın hızını güncelle.
- **2.4. Adım:** Güncel hız ile parçacığın konumunu güncelle.
- **3. Adım:** İteratif süreç sonunda optimum çözüm, küresel en iyi konum ve değer(ler)dir.

Çizelge 3. Genetik Algoritma kaba-kodu (Pseudo-code of the Genetic Algorithm) [38].

- **1. Adım (Kurulum):** *N* bireyli (parçacık) popülasyonu hedef problem ve seçilen kodlama düzenine göre oluştur. Başlangıç algoritma parametre değerlerini ve seçilim, çaprazlama, mutasyon gibi işlemlerde uygulanacak yöntemleri belirle.
- **2. Adım:** İteratif adımlar (her parçacık ve her amaç fonksiyon boyutu için):
- **2.1. Adım:** Birey (parçacık) ile amaç fonksiyon değeri hesapla.
- **2.2. Adım:** Amaç fonksiyon değerleri bağlamında üreme sürecine girecek bireyleri seç.
- **2.3. Adım:** Seçilen bireylere çaprazlama işlemini uygula.
- **2.4. Adım:** Bazı bireylere mutasyon işlemini uygula.
- **3. Adım:** İteratif süreç sonunda optimum çözüm, küresel en iyi konum ve değer(ler)dir.

2.3.3. Guguk kuşu araması (Cuckoo search)

Guguk Kuşu Araması (GKA), doğadaki guguk kuşlarının yuva ve yumurtalarını sahiplenme ve başka yuvaları parazitsel olarak kullanma gibi davranışlarından esinlenerek geliştirilmiş bir zeki optimizasyon algoritmasıdır [49]. GKA'da çözüm uzayına dağıtılmış olan *N* adet parçacık, Levy Uçuşları [50] adı verilen hareketler gerçekleştirerek, potansiyel çözümleri temsil eden konumlara (yuvalara) konumlanmakta, bu noktada problem amaç fonksiyonu üzerinden iyi değerler üreten yumurtalar – yuvalar gelecek nesillere aktarabilmekte, parazitsel davranışlar neticesinde yuvalarını başka parçacıklara (guguk kuşlarına) kaptıran parçacıklar yine yeni yuva arayışlarına girerek iteratif çözüm arama ve optimizasyon sürecini devam ettirmektedir [49, 51]. GKA'nın genel algoritmik adımlarına ilişkin bilgiler, [51] üzerinde detaylı bir şekilde verilmiştir. Algoritmanın kaba-kodu Çizelge 4'te verilmiştir [38].

Çizelge 4. Guguk Kuşu Araması algoritması kaba-kodu (Pseudo-code of the Cuckoo Search Algorithm) [38].

- **1. Adım (Kurulum):** *N* yuvayı (amaç değişken değerlerini) hedef probleme göre, çözüm uzayında rastgele oluştur. Fraksiyon parametresi değerini belirle ve her yuvaya karşılık birer guguk kuşu olduğunu kabul et.
- **2. Adım:** İteratif adımlar (her parçacık ve her amaç fonksiyon boyutu için):
- **2.1. Adım (Levy Uçuşları):** Her guguk kuşu (parçacık) için Levy Uçuş'u gerçekleştir ve ulaştığı konuma göre amaç fonksiyon değeri hesapla.
- **2.2. Adım (Yuva Tespit ve Değerlendirme):** İlgili guguk kuşu için rastgele yuva seç ve karşılıklı amaç fonksiyon değerlerini karşılaştır. Eğer guguk kuşunun daha optimum bir amaç fonksiyon değeri söz konusu ise, guguk kuşu ile yuva değerlerini takas yap.
- **2.3. Adım (Seçilim):** Fraksiyon değerine göre kötü düzeydeki yuvaları ele ve rastgele yeni yuvalar oluştur. İyi yuvaları gelecek nesillere aktar.
- **3. Adım:** İteratif süreç sonunda optimum çözüm, küresel en iyi konum ve değer(ler)dir.

2.3.4. Bakteriyel yiyecek arama optimizasyonu (Bacterial foraging optimization)

Bakteriyel Yiyecek Arama Optimizasyonu (BYAO), Koli Basili adı verilen bakterinin hareketlerinden ve yaşam süreçlerinden esinlenilerek ortaya konulmuş bir zeki optimizasyon algoritmasıdır [52, 53]. BYAO'nun algoritmik adımları, *N* adet, bakterileri temsil eden parçacıkların, çözüm uzayı ortamında gerçekleştirdikleri; kemotaksis, yüzme, yuvarlanma, üreme ve gelecek iteratif süreçlere ulaşmaya çalışma çabalarını temel alan bir yapı ortaya koymaktadır. Algoritmada tıpkı GA'da olduğu gibi bir tür seleksiyon (iyi olanların devamı, kötü olanların elenmesi) mekanizması işletilmiş olmaktadır [53, 54]. Parçacıkların bütün hareket ve yaşam süreçleri, ilgili problemin amaç fonksiyonu üzerinden hesaplanan değer(ler)in uygunluğuna göre devam etmektedir. BYAO'nun algoritmik adımlarına dair detaylar [54] üzerinde açıklanmaktadır. Kaba-kod ise Çizelge 5'te gösterildiği şekildedir [38].

Çizelge 5. Bakteriyel Yiyecek Arama Optimizasyonu algoritması kaba-kodu (Pseudo-code of the Bacterial Foraging Optimization algorithm) [38].

- **1. Adım (Kurulum):** N bakteriyi (parçacığı) çözüm uzayına rastgele dağıt. Algoritma parametreleri değerlerini belirle. Hedef probleme dair düzenlemeleri gerçekleştir.
- **2. Adım:** Bakterilerin konumlarına göre amaç fonksiyon değerini hesapla.
- **3. Adım:** Aşağıdaki adımları, N_e algoritma parametresi kadar, her boyut için tekrarla:
- **3.1. Adım (Kemotaksis):** Aşağıdaki adımları N_k algoritma parametresi kadar, her bakteri için gerçekleştir:
- **3.1.1. Adım:** Bakterinin konumuna göre amaç fonksiyon değerini hesapla.
- **3.1.2. Adım:** Bakterinin amaç fonksiyon değerini hücreler arası çekici etkisi ile güncelle. Yüzme sürecine kadar bu değeri sakla.
- **3.1.3. Adım (Yuvarlanma):** [-1, 1] aralığında amaç fonksiyon boyutu kadar rastgele sayı üret. İlgili bakteri için yuvarlanma işlemini gerçekleştir.
- **3.1.4. Adım:** Bakterinin konumuna göre amaç fonksiyon değerini hesapla ve yine ilgili bakterinin amaç fonksiyon değerini hücreler arası çekici etkisi ile güncelle.
- **3.1.5. Adım (Yüzme):** Aşağıdaki adımları $N_{yüz}$ algoritma parametresi kadar, ilgili bakteri için gerçekleştir:
- **3.1.5.1. Adım:** Bakterinin son amaç fonksiyon değeri, Yüzme sürecinden önce saklanandan daha iyi ise güncelle.
- **3.1.5.2. Adım:** Hesaplanan yer değiştirme değerine göre bakterinin saklanan amaç fonksiyon değerini güncelle.
- **3.2. Adım (Üreme):** Her bakterinin sağlık durumunu hesapla ve hepsini, bu değere göre büyükten küçüğe sırala.
- **3.3. Adım:** İlgili kritere göre en kötü bakterileri ele. En iyi durumdakilerin ise üremesini sağla (Yeni bakteriler ebeveynlerin yerlerine geçer).
- **3.4. Adım:** $N_{ü}$ algoritma parametre değerine ulaşılmadıysa, bu değeri bir artır ve 3.1. Adım'a dön (Yeni nesil bakteriler ile devam).
- **3.5. Adım (Eliminasyon – Dağılım).** o_{ed} algoritma değerine göre her bakteriyi rastgele yeni bir konuma geçir.
- **Adım 4.** İteratif süreç sonunda optimum çözüm, küresel en iyi konum ve değer(ler)dir.

2.3.5. *Çiçek tozlaşma optimizasyonu (Flower pollination optimization)*

Adından da anlaşılacağı üzere; doğadaki tozlaşma süreçlerinden esinlenerek geliştirilmiş olan Çiçek Tozlaşma Optimizasyonu (ÇTO), N adet parçacığın, birer çiçeği temsilen çözüm uzayına konumlandığı ve tozlaşma süreçlerine dair çeşitli mekanizmaların işletilerek, amaç fonksiyon değer(ler)i üzerinden optimizasyon süreçlerinin yerine getirildiği bir zeki optimizasyon algoritması olarak literatürde yer almaktadır [55]. ÇTO'nun algoritmik adımları, küresel tozlaşma adı verilen ve Levy Uçuşları yaklaşımına dayanan çapraz konumlandırmalar ile yerel tozlaşma olarak ifade edilen, kendi kendine konumlandırmalara ve çiçek istikrarı kriteri üzerinden çiçek benzerliklerinin değerlendirilmesi – yeniden üretim gibi süreçleri içermektedir. Bütün bu süreçler her algoritmada olduğu gibi iteratif bir şekilde gerçekleştirilmektedir [55, 56]. ÇTO'nun algoritma adımlarına ilişkin detaylar [56]'da ifade edilmektedir. Kaba-kod ise Çizelge 6'da gösterilmiştir [38].

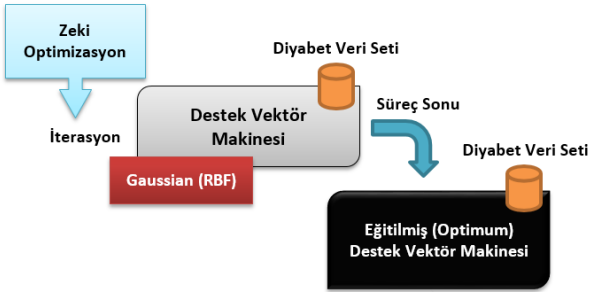
Çizelge 6. Çiçek Tozlaşma Optimizasyonu algoritması kaba-kodu (Pseudo-code of the Flower Pollination Optimization algorithm) [38].

- **1. Adım (Kurulum).** N çiçeği (parçacığı) çözüm uzayına rastgele dağıt. Algoritma parametre değerlerini ata ve hedef probleme dair düzenlemeleri gerçekleştir.
- **2. Adım:** Çiçeklerin konumlarına göre amaç fonksiyon değeri hesapla ve en iyi olanı belirle.
- **3. Adım:** İteratif adımlar (her parçacık ve her amaç fonksiyon boyutu için):
- **3.1. Adım (Küresel – Yerel Tozlaşma).** Rastgele bir değer üret. Eğer üretilen değer *go* algoritma parametresinden düşük ise Levy Uçuşları üzerinden küresel tozlaşma gerçekleştir. Aksi halde, [0, 1] aralığında uniform dağılım ve yerel tozlaşma yap.
- **3.2. Adım:** Çiçeklerin güncel konumlarına göre amaç fonksiyon değerini hesapla. En iyi amaç fonksiyon değerinden daha iyisi bulunmuşsa küresel en iyi değerleri güncelle.
- **4. Adım:** İteratif süreç sonunda optimum çözüm, küresel en iyi konum ve değer(ler)dir.

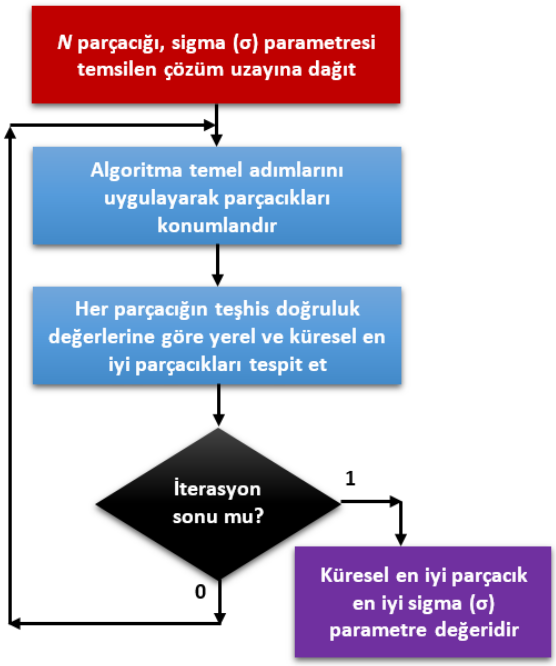
İlgili zeki optimizasyon algoritmaları, bir sonraki alt-bölüm altında açıklanan hibrit sistem yaklaşımı üzerinden DVM'yi eğitmek (optimize etmek) adına kullanılmıştır. Bu bağlamda dikkate alınan çözüm süreci, ilerleyen paragraflar altında açıklanmıştır.

2.4. Diyabet Teşhisi İçin Zeki Optimizasyon Tabanlı DVM Sistemi (Intelligent Optimization Based SVM System for Diabetes Diagnosis)

Çalışma kapsamında oluşturulan zeki optimizasyon-DVM hibrit sistem yaklaşımı, genel hatlarıyla parametre optimizasyonunun parçacık temelli algoritmalarla gerçekleştirilmesi sürecine dayanmaktadır (Şekil 1).



Şekil 1. Diyabet teşhisi için zeki optimizasyon tabanlı DVM sistemi (Intelligent optimization based SVM system for diabetes diagnosis).



Şekil 2. Geliştirilen sistemde eğitim – optimizasyon süreci (Training – optimization process in the developed system).

Şekil 1'de gösterilen sistemde, hedef diyabet veri seti kapsamında alternatif optimize DVM'lerin başarımlarını ölçmek için aşağıdaki adımlar (Şekil 2), aynı parçacık sayısı ve eğitim (optimizasyon) iterasyon sayısına tabi, seçili zeki optimizasyon algoritmalarına uygulanmıştır:

- Seçilen zeki optimizasyon algoritmasının *N* parçacığı, DVM'deki Gaussian-RBF kernel fonksiyonunda işe koşulacak sigma (σ) parametresinin değerini taşıma görevindedir. İlgili parçacıklar bu amaçla çözüm uzayına konumlandırılmaktadır.
- Anlaşılacağı üzere, algoritmanın kendi spesifik adımları doğrultusunda yönlendirdiği parçacıkların konum değerleri değiştikçe, DVM kernel fonksiyondaki sigma (σ) parametresi olarak kullanılarak teşhis başarımı ölçülebilecektir.
- Algoritma belirli bir iterasyon sayısına göre, eğitim verisi üzerinde çalıştırılmakta ve her iterasyonda, parçacıkların sigma (σ) parametresi değerleri DVM'de kullanılmaktadır. Burada, Eşitlik 5 aracılığıyla hesaplanan teşhis doğruluk değerlerine göre en iyi parçacıklar [yani sigma (σ) parametre değerleri] tespit edilmektedir.

$$\frac{dnt + dpt}{ynt + ypt + dnt + dpt} * 100 \qquad (5)$$

Eşitlik 5'te, *dnt* doğru teşhis edilen negatif (diyabet olmayan) kişi sayısı, *dpt* doğru teşhis edilen pozitif (diyabet) kişi sayısı, *ynt* yanlış teşhis edilen negatif (diyabet olmayan) kişi sayısı ve *ypt* yanlış teşhis edilen pozitif (diyabet) kişi sayısıdır.

- Algoritmanın iteratif adımları sonunda, elde edilen optimum (küresel en iyi) parçacık değeri, DVM'nin Gaussian-RBF kernel fonksiyonundaki sigma (σ) parametresi için, diyabet teşhisinde en yüksek başarımı sağlayan değerdir. Bütün bu süreç, DVM'nin bir tür eğitimi (optimizasyonu) anlamına gelmektedir.

3. BULGULAR (FINDINGS)

Bir önceki bölüm altında anlatılan hibrit sistem düzeneği, çalışmaya dahil edilen beş farklı zeki optimizasyon algoritması üzerinden, hedef diyabet veri seti dikkate alınarak çalıştırılmıştır. Bu noktada, aşağıdaki kıstaslar da uygulamalarda dikkate alınmıştır:

- Her algoritmada parçacık sayısı N sırasıyla 50, 75 ve 100 olmak üzere, 5000 iterasyonda eğitim (optimizasyon) sürecinden geçirilerek, üç farklı parçacık sayısının süreci nasıl etkilediğine ilişkin bulguların elde edilmesi sağlanmıştır.
- Daha önce de belirtiği üzere; toplamda 768 örnekten oluşan diyabet veri setinin yarısı (384 örnek) eğitim için (eğitim veri seti) kullanılmıştır.
- Her zeki optimizasyon algoritmasının ilgili parametrelerine, literatürdeki kaynaklarda [44, 47, 49, 54, 56] geçen ve genel kabul gören, varsayılan değerler verilmiştir.
- İşe koşulan zeki optimizasyon-DVM sistemlerinin tutarlılığını yeter düzeyde ölçmek adına, her farklı parçacık düzeni için sistemler 30'ar kez çalıştırılmış ve elde edilen optimum sigma (σ) parametrelerinin ortalaması dikkate alınarak, eğitim için ayrılan 384 örnek haricindeki, geriye kalan 384 örnek (test veri seti) kullanılarak, diyabet teşhis süreci değerlendirmesine geçilmiştir.

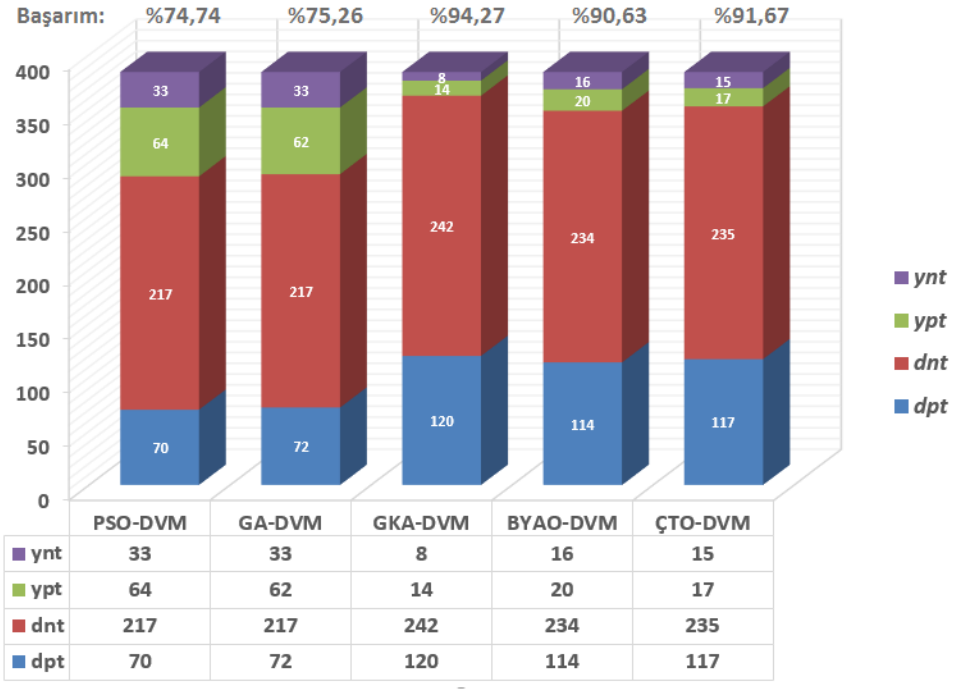
Her hibrit sistem için farklı parçacık sayılarıyla ulaşılan ortalama sigma (σ) parametresi ve başarım oranı değerleri Çizelge 7'de verilmiştir (En iyi bulgular koyu olarak gösterilmiştir).

Çizelge 7. Farklı hibrit sistemler ve parçacık düzenleri ile elde edilen sigma (σ) parametresi ortalamaları (Obtained average value of sigma (σ) parameter on different hybrid systems and particle set-ups).

Sistem	Parçacık Sayısı*	Ortalama sigma (σ)*	Başarım Oranı*
PSO-DVM	50	0,6584	%72,12
	75	0,6550	%73,15
	100	**0,6482**	**%74,61**
GA-DVM	50	0,6604	%71,60
	75	0,6590	%72,14
	100	**0,6472**	**%75,22**

GKA-DVM	50	0,5599	%87,74
	75	0,5017	%93,49
	100	**0,4889**	**%94,42**
BYAO-DVM	50	0,6008	%81,77
	75	0,5402	%89,31
	100	**0,5321**	**%90,59**
ÇTO-DVM	50	0,5798	%85,39
	75	0,5330	%89,81
	100	**0,5204**	**%91,77**
*** Her sistem için en iyi bulgular koyu olarak gösterilmiştir**			

Söz konusu ortalama sigma (σ) parametresi değerleri kullanılarak, hedef diyabet veri setinin test veri seti kısmı üzerinden (384 örnek), her bir sistemin elde ettiği (*Bkz.* Eşitlik 5) doğru teşhis edilen negatif (*dnt*: diyabet olmayan) kişi sayısı, doğru teşhis edilen pozitif (*dpt*: diyabet) kişi sayısı, yanlış teşhis edilen negatif (*ynt*: diyabet olmayan) kişi sayısı ve yanlış teşhis edilen pozitif (*ypt*: diyabet) kişi sayısı bağlamında başarım grafikleri Şekil 3'te gösterilmiştir.



Şekil 3. Farklı hibrit sistemlerin diyabet teşhisindeki başarımları (Success of different hybrid systems on diabetes diagnosis).

Şekil 3'ten hareketle, her bir hibrit sistemin diyabet teşhis başarımlarına ilişkin şu yorumları yapmak mümkündür:

- PSO-DVM sisteminde doğru tespit edilen birey verileri negatif tespit için 217, pozitif tespit için 70 olmuştur. Böylece, toplamda doğru tespit sayısı 287 olup, geriye kalan 97 birey verisinin (*ynt:* 33 ve *ypt:* 64) diyabet yönünden tespiti yanlış yapılmıştır. Genel anlamda sistem, negatif diyabetleri kendi içinde oransal anlamda daha doğru tespit etmiştir. Sistemin tespit durumu sayısal olarak GA-DVM ile oldukça yakındır. Sonuç itibariyle ise PSO-DVM sisteminin genel başarım oranı %74,74 bulunmuştur.
- GA-DVM sistemi, 217 adet doğru negatif tespit ve 72 adet doğru pozitif tespit ile toplamda 289 doğru diyabet tespitinde bulunmuştur. Yanlış yapılan tespit sayısı ise toplamda 95 (*ynt:* 33 ve *ypt:* 62) şeklindedir. Sistemin kendi içlerinde oransal anlamda negatif diyabetleri daha doğru tespit ettiği görülmekle birlikte, sayısal anlamdaki değer, PSO-DVM ile aynıdır. Sadece, pozitif birey tespiti PSO-DVM sistemine göre 2 birey verisi daha fazladır (72). Genel başarım oranı ise %75,26'dır.
- GKA-DVM sistemindeki doğru tespit edilen birey verisi sayısı toplamda 362 olarak elde edilmiştir (*dnt:* 242 ve *dpt:* 120). Geriye kalan 22 adet birey verisi ise yanlış tespit edilmiştir (*ynt:* 33 ve *ypt:* 62). Oransal anlamda bu sistemde de negatif diyabetleri tespit etme oranı yüksek olsa da, pozitif tespit oranında da, özellikle PSO-DVM ve GA-DVM'ye göre çok daha fazla artış görülmektedir. GKA-DVM sisteminin genel başarım oranı %94,27'dir.
- BYAO-DVM sisteminde birey verileri için doğru tespit sayısı negatif olanlar için (*dnt*) 234, pozitif olanlar için (*dpt*) 114 olarak elde edilmiştir. Geriye kalan 36 birey verisinin (*ynt:* 16 ve *ypt:* 20) diyabet yönünden tespiti ise yanlış yapılmıştır. Oransal anlamda sistemin kendi içlerinde negatif diyabet tespit oranı diyabet pozitif tespitinden yaklaşık %8 daha iyidir. Sistemin genel başarım oranı %90,63'tür.

- ÇTO-DVM sistemi için elde edilen doğru tespit sayısı toplamda 352 bulunmuştur (*dnt:* 235 ve *dpt:* 117). Yanlış yapılan diyabet teşhisi ise birey verisi kapsamında 32'dir (*ynt:* 15 ve *ypt:* 17). Oransal anlamda bakıldığında ise sistemin kendi içlerinde negatif diyabet tespit oranı diyabet pozitif tespitinden yaklaşık %7 daha iyi durumdadır. Genel başarım oranı bu sistemde %91,67'dir.
- Genel çerçevede bakıldığında, PSO-DVM ile GA-DVM sistemlerinin birbirine yakın olduğu performanslara karşı, GKA-DVM, ÇTO-DVM ve BYAO-DVM'nın %90'ın üstünde birbirine yakın, uzak ara sayılacak noktalarda bir başka grup halinde temsil edebileceğimiz bir performans durumu göze çarpmaktadır.

Açıklamaları ve bulguları bir bütün halinde incelediğimizde, tespit edilen ortalama sigma (σ) parametresi değerleri üzerinden, hedef diyabet veri seti üzerinde en yüksek oranda tespit başarımını GKA-DVM hibrit sistemi göstermiş, bu sistemi sırasıyla ÇTO-DVM, BYAO-DVM, GA-DVM ve PSO-DVM sırası izlemiştir. Dolayısıyla seçilen zeki optimizasyon algoritmalarının her birinin diyabet veri seti bağlamında DVM eğitimi (optimizasyonu) etkinliğinin de bu sırada olduğu bulgusuna ulaşabiliriz. Genel hatlarıyla bakıldığında ise, her sistemin, kabul edilebilir düzeylerde diyabet teşhisi performansına sahip olduğunu ifade etmek de yanlış bir tespit olmayacaktır.

6. SONUÇ VE TARTIŞMA (CONCLUSION and DISCUSSION)

Medikal teşhis, Yapay Zekâ'nın yoğun bir şekilde uygulandığı problem kapsamlarından birisidir. Bu doğrultuda kurulan zeki sistemler, özellikle sağlık alanındaki karar destek süreçleri için kritik önem arz etmektedir. Bu çalışmada, söz konusu problem kapsamında diyabet teşhisine odaklanılmış ve bu amaçla, zeki optimizasyon tabanlı Destek Vektör Makineleri (DVM) yaklaşımına dayalı bir süreç izlenmiştir. Non-lineer yapıdaki, Gaussian-RBF kernel fonksiyonu kullanan ve zeki optimizasyon algoritmalarıyla eğitilen (optimize edilen) DVM sistemleri Pima Yerlileri veri seti üzerinde uygulanmış ve elde edilen bulgular değerlendirilmiştir. Bu noktada, zeki optimizasyon algoritmalarının rolü, Gaussian-RBF kernel fonksiyonunun sigma (σ) parametresini optimize etmek olmuştur.

Çalışmada zeki optimizasyon kapsamında seçilen iki temel ve üç güncel algoritmanın [Parçacık Sürü Optimizasyonu (PSO), Genetik Algoritma (GA), Guguk Kuşu Araması (GKA), Bakteriyel Yiyecek Arama Optimizasyonu (BYAO) ve Çiçek Tozlaşma Optimizasyonu (ÇTO)] diyabet teşhisi doğrultusunda performansları dikkate alındığında; genel olarak bütün algoritmaların diyabet teşhisinde istendik düzeylerde başarım gösterdiği, spesifik olarak bakıldığında ise, GKA ile desteklenmiş DVM'nin (hibrit sistemin) diğer algoritmalarla kurulmuş hibrit sistemlere göre daha iyi başarım gösterdiği görülmüştür (Bu algoritmayı sırasıyla ÇTO, BYAO, GA ve PSO takip etmiştir). İşe koşulan DVM, ortaya koyduğu başarılı sonuçlar nedeniyle, farklı gerçek dünya problemleri için sıklıkla kullanılagelen bir teknik konumundadır. Bu çalışmada ulaşılan sonuçlar, gerek DVM'nin, gerekse bu teknik ile kurulabilecek hibrit sistemlerin teşhis ve hatta medikal teşhis yönelimli problemlere dair potansiyelini kanıtlamıştır. Yine ulaşılan sonuçlar, Yapay Zekâ'nın potansiyelini de ortaya koymaktadır.

Elde edilen sonuçlar, farklı gelecek çalışmaların düşünülmesine de yol açmıştır. Buna göre, aynı problem – sistem düzeninde farklı zeki optimizasyon algoritmalarının performansları da değerlendirilecek, DVM'nin farklı kernel fonksiyonlarla uygulamaları ve yine kernel fonksiyonlardaki farklı parametrelerin optimizasyonuna yönelik alternatif çalışmalar yerine getirilecektir. Yine ilgili zeki optimizasyon tabanlı DVM yaklaşımının farklı gerçek dünya problemlerine yönelik uygulamalar da gelecek çalışmalar kapsamındadır.

KAYNAKLAR (REFERENCES)

[1] Russell, S.J., Norvig, P., Canny, J.F., Malik, J.M. and Edwards, D.D., ***"Artificial Intelligence: A Modern Approach"***, 2(9): Upper Saddle River: Prentice Hall, (2003).

[2] Nabiyev, V.V., ***"Yapay Zeka: Problemler-Yöntemler-Algoritmalar"***, Seçkin Yayıncılık, (2012).

[3] Goertzel, B., "Artificial general intelligence: concept, state of the art, and future prospects", ***Journal of Artificial General Intelligence***, 5(1): 1-48, (2014).

[4] Allahverdi, N., ***"Uzman Sistemler: Bir Yapay Zeka Uygulaması"***, Atlas Yayın Dağıtım, (2002).

[5] Alpaydın, E., ***"Yapay Öğrenme"***, İstanbul, Boğaziçi Üniversitesi Yayınevi, (2011).

[6] Aydın, A.O., ***"Yapay Zekâ: Bütünleşik Bilişe Doğru"***, İstanbul, İstanbul Gelişim Üniversitesi Yayınları, (2013).

[7] Hamet, P. and Tremblay, J., "Artificial intelligence in medicine", ***Metabolism***, 69: 36-40, (2017).

[8] Jaeger, H., "Artificial intelligence: Deep neural reasoning", ***Nature***, 538(7626): 467-468, (2016).

[9] Strong, A., "Applications of artificial intelligence and associated technologies", ***Science (ETEBMS-2016)***, 5(6), (2016).

[10] Szolovits, P., Patil, R.S. and Schwartz, W.B., "Artificial intelligence in medical diagnosis", ***Annals of Internal Medicine***, 108(1): 80-87, (1988).

[11] Kononenko, I., "Machine learning for medical diagnosis: history, state of the art and perspective", ***Artificial Intelligence in Medicine***, 23(1): 89-109, (2001).

[12] Abbass, H.A., "An evolutionary artificial neural networks approach for breast cancer diagnosis", ***Artificial Intelligence in Medicine***, 25(3): 265-281, (2002).

[13] Al-Shayea, Q.K., "Artificial neural networks in medical diagnosis", ***International Journal of Computer Science Issues***, 8(2): 150-154, (2011).

[14] Dhar, J. and Ranganathan, A., "Machine learning capabilities in medical diagnosis applications:

Computational results for hepatitis disease", ***International Journal of Biomedical Eng and Tech***, 17(4): 330-340, (2015).

[15] Botero-Rosas, D., Leon-A,J., Reina, J.M., Obando, A. and Bastidas, A.R., "Use of artificial ıntelligence in the diagnosis of chronic obstructive pulmonary disease (COPD)", ***Am. J. Respir. Crit. Care. Med.***, 191, (2015).

[16] Thong, N.T., "HIFCF: An effective hybrid model between picture fuzzy clustering and intuitionistic fuzzy recommender systems for medical diagnosis", ***Expert Systems with Applications***, 42(7): 3682-3701, (2015).

[17] Faris, H., Hassonah, M.A., Ala'M, A.Z., Mirjalili, S. and Aljarah, I., "A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture", ***Neural Computing and Applications***, 1-15, (2017).

[18] Das, S.P. and Padhy, S., "A novel hybrid model using teaching–learning-based optimization and a support vector machine for commodity futures index forecasting", ***International Journal of Machine Learning and Cybernetics***, 9(1): 97-111, (2018).

[19] Pradhan, B., Jebur, M.N. and Abdullahi, S., "Spatial Prediction of Landslides Along Jalan Kota in Bandar Seri Begawan (Brunei) Using Airborne LiDAR Data and Support Vector Machine". In ***Laser Scanning Applications in Landslide Assessment***, 167-178: Springer, Cham, (2017).

[20] Ling, X., Feng, X., Chen, Z., Xu, Y. and Zheng, H., "Short-term traffic flow prediction with optimized Multi-kernel Support Vector Machine". In ***Evolutionary Computation (CEC), 2017 IEEE Congress on***, 294-300: IEEE, (2017).

[21] Tharwat, A. and Hassanien, A.E., "Chaotic antlion algorithm for parameter optimization of support vector machine", ***Applied Intelligence***, 48(3): 670-686, (2018).

[22] Frank, A. and Asuncion, A., ***"UCI Machine Learning Repository"***, University of California Irvine, School of Information and Computer Science, http://archive.ics.uci.edu/ml, [Erişim: 11.01.2018], (2010).

[23] Turney, P., "Pima Indians Diabetes Data Set", UCI M.L. Repository. Originally from: ***National Ins. of Diabetes and Digestive and Kidney Diseases***, https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes, [Erişim: 11.01.2018], (1990; updated: 2011).

[24] Çomak, E., Arslan, A. and Türkoğlu, İ., "A decision support system based on support machines for diagnosis of the heart valve diseases", ***Computers in Biology and Medicine***, 37: 21-27, (2007).

[25] Vapnik, V., Golowich, S. and Smola, A., "Support vector method for function approximation, regression estimation, and signal processing", ***Advances in Neural Information Processing Systems***, 9: 281–287, (1997).

[26] Akşehirli, Ö.Y., Ankaralı, H., Aydın, D. and Saraçlı, Ö., "Tıbbi tahminde alternatif bir yaklaşım: Destek vektör makineleri", ***Turkiye Klinikleri Journal of Biostatistics***, 5(1), (2013).

[27] Chung, S.S. and Zhang, S., "Volatility estimation using support vector machine: Applications to major foreign exchange rates", ***Electronic Journal of Applied Statistical Analysis***, 10(2): 499-511, (2017).

[28] Güner, N. and Çomak, E., "Mühendislik öğrencilerinin matematik I derslerindeki başarısının destek vektör makineleri kullanılarak tahmin edilmesi", ***Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi***, 17(2): 87-96, (2011).

[29] Rastogi, R., Sharma, S. and Chandra, S., "Robust parametric twin support vector machine for pattern classification", ***Neural Processing Letters***, 1-31, (2017).

[30] Kavzoğlu, T. and Çölkesen, İ., "Destek vektör makineleri ile uydu görüntülerinin sınıflandırılmasında kernel fonksiyonlarının etkilerinin incelenmesi", ***Harita Dergisi***, 144(7): 73-82, (2010).

[31] Hasni, H., Alavi, A.H., Jiao, P. and Lajnef, N., "Detection of fatigue cracking in steel bridge girders: a support vector machine approach", ***Archives of Civil and Mechanical Engineering***, 17(3): 609-622, (2017).

[32] Tamura, H. and Tanno, K., "Midpoint validation method for support vector machines with margin adjustment technique", ***International Journal of Innovative Computing, Information and Control***, 5: 4025-4032, (2009).

[33] Çomak, E., ***"Destek Vektör Makineleri Çoklu Sınıf Problemleri için Çözüm Önerileri"***, Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği ABD., (2004).

[34] Özkaya, A.U., Kaya, M.E. and Gürgen, F. "Destek Vektör Makineleri Kullanılarak Aritmi Sınıflandırması". ***Biyomedikal Müh. Ulusal Toplantısı***, (2005).

[35] Karaç, E.I., ***"Model Selection for Multi-Class Support Vector Machines"***, Yüksek Lisans Tezi, Boğaziçi Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği ABD., (2005).

[36] Özkan, Y., ***"Veri Madenciliği Yöntemleri"***, Papatya Yayıncılık, (2008).

[37] Özkaya, A.U., ***"Intelligent Arrhythmia Classification Based on Support Vector Machines"***, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümü, (2003).

[38] Köse, U., ***"Yapay Zekâ Tabanlı Optimizasyon Algoritmaları Geliştirilmesi"***, Doktora Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği ABD., (2017).

[39] Karaboğa, D., ***"Yapay Zekâ Optimizasyon Algoritmaları"***, Ankara, Nobel Akademik Yayıncılık, (2014).

[40] Yang, X.-S., ***"Nature-Inspired Metaheuristic Algorithms"***, Luniver Press, (2010).

[41] Yang, X.-S., Cui, Z., Xiao, R., Gandomi, A.H. and Karamanoglu, M., ***"Swarm Intelligence and Bio-Inspired Computation: Theory and Applications"***, Newnes, (2013).

[42] Hassanien, A.E. and Emary, E., ***"Swarm Intelligence: Principles, Advances, and Applications"***, CRC Press, (2016).

[43] Eberhart, R. and Kennedy, J., "A new optimizer using particle swarm theory", ***Micro Machine and Human Science***, 1995. MHS'95., Proceedings of the Sixth International Symposium on, 39-43, (1995).

[44] Shi, Y. and Eberhart, R., "A modified particle swarm optimizer", Evolutionary Computation Proceedings, ***1998. IEEE World Congress on Computational***

Intelligence., The 1998 IEEE International Conference on, 69-73, (1998).

[45] Kennedy, J., "Particle Swarm Optimization", In: ***Encyclopedia of Machine Learning***, Eds: Springer, 760-766, (2011).

[46] Back, T., "Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming", ***Genetic Algorithms***, Oxford University Press, (1996).

[47] Goldberg, D.E., ***"Genetic Algorithms"***, Pearson Education India, (2006).

[48] Holland, J.H., "Genetic algorithms", ***Scholarpedia***, 7(12): 1482, http://www.scholarpedia.org/article/Genetic_algorithms, [Erişim: 20.01.2018], (2012).

[49] Yang, X.-S. and Deb, S., "Cuckoo search via Lévy flights", Nature & Biologically Inspired Computing, ***2009. NaBIC 2009. World Congress on***, 210-214, (2009).

[50] Mandelbrot, B.B., ***"The Fractal Geometry of Nature"***, New York, W. H. Freeman, (1982).

[51] Yang, X.-S. and Deb, S., "Engineering optimisation by cuckoo search", ***International Journal of Mathematical Modelling and Numerical Optimisation***, 1(4): 330-343, (2010).

[52] Passino, K.M., "Biomimicry of bacterial foraging for distributed optimization and control", ***IEEE Control Systems***, 22(3): 52-67, (2002).

[53] Passino, K.M., "Bacterial foraging optimization", ***Innovations and Developments of Swarm Intelligence Applications***, 219-233, (2012).

[54] Gazi, V. and Passino, K.M., "Bacteria Foraging Optimization", In: ***Swarm Stability and Optimization***, Eds: Springer, 233-249, (2011).

[55] Yang, X.-S., "Flower Pollination Algorithm for Global Optimization", ***International Conference on Unconventional Computing and Natural Computation***, 240-249, (2012).

[56] Yang, X.-S., Karamanoglu, M. and He, X., "Flower pollination algorithm: A novel approach for multiobjective optimization", ***Engineering Optimization***, 46(9): 1222-1237, (2014).