Pamukkale Univ Muh Bilim Derg, 29(6), 596-606, 2023

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

Metasezgisel yöntemlerle öznitelik sayısını azaltarak diyabetin erken dönemde tespiti

Early stage diabetes prediction by features selection with metaheuristic methods

Tuğberk ÖZMEN[1], Üzeyir KUZU[2], Yücel KOÇYİĞİT[3], Haldun SARNEL[3]*

[1]Elektrik ve Enerji Bölümü, Manisa Teknik Bilimler Meslek Yüksekokulu, Manisa Celal Bayar Üniversitesi, Manisa, Türkiye.
tugberk.ozmen@cbu.edu.tr

[2]Elektronik ve Otomasyon Bölümü, Manisa Teknik Bilimler Meslek Yüksekokulu, Manisa Celal Bayar Üniversitesi, Manisa, Türkiye.
uzeyir.kuzu@cbu.edu.tr

[3]Elektrik Elektronik Mühendisliği Bölümü, Mühendislik Fakültesi, Manisa Celal Bayar Üniversitesi, Manisa, Türkiye.
yucel.kocyigit@cbu.edu.tr, haldun.sarnel@cbu.edu.tr

Geliş Tarihi/Received: 29.09.2022 Düzeltme Tarihi/Revision: 12.12.2022 doi: 10.5505/pajes.2022.82610
Kabul Tarihi/Accepted: 22.12.2022 Araştırma Makalesi/Research Article

Öz

Diyabet dünya çapında yaygın olarak görülen metabolik bir hastalıktır. Dünya genelinde her geçen yıl diyabet hastalığına yakalanan kişi sayısının artması beklenmektedir. Bu da hem kişilerin yaşam konforları hem de sağlık sistemi için olumsuz bir etki demektedir. Bu açıdan hastalığın erken dönemde teşhis edilmesi önem taşımaktadır. Teşhis amacıyla kullanılan verilerin yüksek boyutlu olması hesaplamanın maliyeti ve süresi üzerinde olumsuz etkiye sahiptir. Bunun önüne geçmek için, teşhis için en değerli olan özniteliklerin seçilmesi önem arz etmektedir. Bu çalışmada UCI (UCI Machine Learning Repository) veri deposundaki örnekler kullanılarak, Salp Sürü Algoritması, Yapay Arı Kolonisi Algoritması, Balina Optimizasyon Algoritması ve Karınca Kolonisi Algoritması kullanılarak öznitelik seçimi yapılmıştır. Seçilen özniteliklerin değerlendirilmesi için k-En Yakın Komşuluk (KNN), Naive Bayes (NB), Destek Vektör Makinası (DVM) ve Yapay Sinir Ağları (YSA) yöntemleri kullanılarak doğruluk, duyarlılık ve belirlilik parametreleri hesaplanmıştır. Diyabet hastası olma olasılığı için yapılan hesaplamalarda k-En Yakın Komşuluk yöntemiyle %99.04 doğruluk oranı elde edilmiştir.

Anahtar kelimeler: Diyabet, Metasezgisel, Salp sürü, Yapay arı kolonisi, Balina Sürüsü, Karınca Kolonisi, Öznitelik seçimi.

Abstract

Diabetes is a metabolic disease that is common worldwide. The number of people suffering from diabetes is expected to increase every year around the world. This means a negative impact on both the comfort of life of individuals and the health system. In this respect, it is important to diagnose the disease at an early stage. The high dimensionality of the data used for diagnostic purposes has a negative effect on the cost and time of the calculation. To avoid this, it is important to select the most valuable features for diagnosis. In this study, feature selection was made using Salp Swarm Algorithm, Artificial Bee Colony Algorithm, Whale Optimization Algorithm and Ant Colony Algorithm using the samples in the UCI (UCI Machine Learning Repository) data store. In order to evaluate the selected features, accuracy, sensitivity and specificity parameters were calculated using k-Nearest Neighborhood (KNN), Naive Bayes (NB), Support Vector Machine (SVM) and Artificial Neural Networks (ANN) methods. In the calculations for the probability of having diabetes, an accuracy rate of 99.04% was obtained with the k-Nearest Neighborhood method.

Keywords: Diabetes, Metaheuristic, Salp swarm, Artificial Bee colony, Whale swarm, Ant colony, Feature selection.

1 Giriş

Diyabet, kişilerde yetersiz insülin üretimi veya hücrelerde insüline düzgün cevap verilememesinden kaynaklı olarak kan şekeri değerinin sürekli olarak normalden daha yüksek olduğu dünya çapında yaygın olarak görülen bir tür metabolik hastalıktır [1]. Pankreas organının salgıladığı insülin hormonu tarafından kan şekerini dengelenirken diyabet hastaların bu hormon tam olarak işlev göremediği için kan şekeri değeri sürekli yüksek olmaktadır [2]. Çağımızın en ciddi ve yaygın kronik hastalıklarından biri olan diyabet, hayatı tehdit eden, yaşam kalitesini düşüren beraberinde hayati komplikasyonları da getiren tedavisi maliyetli bir hastalıktır [3]. Kan şekerinin sürekli olarak yüksek çıkması göz ve böbrek başta olmak üzere organlara zarar verebilmektedir. Diyabetin genel belirtileri arasında susuzluk ve açlık hissi, sık idrara çıkma isteği, yorgunluk ve zor iyileşen yaralar sayılabilir. Diyabet hastalığının İki türü vardır. Doğuştan itibaren gelen, pankreasın yeteri kadar veya hiç insülin üretememesi durumu Tip 1 diyabet olarak tarif edilirken; Tip 2 diyabet hastalığında hem insülin yetersizliği hem de insüline karşı direnç görülmektedir. Uluslararası Diyabet Federasyonu'na göre 2045 yılına kadar dünyadaki diyabetli kişi sayısının 700 milyona ulaşması beklenmektedir [4]. Bu nedenle, hastalığın erken dönemde teşhis edilmesi kişileri hastalığın zararlarından korumak için ve daha kaliteli bir yaşam sürmelerini sağlamak için önemlidir.

Günümüz teknolojisi sayesinde veri havuzları için çok miktarda veri elde edilebilmekte ve anlamlandırılmaktadır. Ancak bu verilerin boyutu arttıkça süreç karmaşık hale gelebilmekte ve hız düşebilmektedir. Büyük boyutlu verilerin incelenmesinin kolaylaştırılması için, verilerin doğruluğunu bozmayacak şekilde boyutunu azaltmaya yönelik yöntemler uygulanmaktadır. Öznitelik seçiminde amaç, veriyi analiz etmede daha güçlü olan özniteliklerin belirlenmesidir. Böylece öznitelik kümesi daraltılıp, performansın arttırılması sağlanır. Diyabet hastalığının erken dönemde teşhis edilebilmesi için,

*Yazışılan yazar/Corresponding author

kullanılacak veri setindeki özniteliklerden seçme işlemi yapılarak teşhisin doğruluk değeri arttırılabilmektedir.

Özlüer ve diğ.(2021), makine öğrenmesi algoritmalarından yararlanarak diyabet hastalığının sınıflandırılmasına yönelik bir çalışma yapmıştır. Bu çalışma için ABD'deki 130 hastaneden 70000 vakaya ilişkin verileri kullanmışlardır. 55 özelliğe sahip olan veriler öncelikle veri temizleme işlemine tabi tutulup biri sınıf olmak üzere toplam 23 değişkene indirilmiş ve ardından farklı sınıflandırma yöntemleri uygulanmıştır. Rastgele orman algoritmasında doğru sınıflandırma değeri en yüksek %84.78 olarak elde edilmiştir. Bu sonuç, sağlık kuruluşlarına başvuran kişilerde 22 değişkene verilen cevaplara göre %84.78 doğruluk oranında diyabet tahmininin yapılabileceğini göstermektedir [5].

Akyol ve Karacı (2021) UCI deposundan aldıkları veri setinin analizinde Rastgele Orman, K-En Yakın Komşu, Derin Sinir Ağları, Gradyan Arttırma ve Oylama topluluk algoritmaları ile oluşturulan modellerin dışarıda tutma ve 5-kat çapraz doğrulama yöntemlerini kullanmıştır. Çalışmada Oylama Topluluğu sınıflandırıcısı ile en yüksek doğruluk değerleri elde edilmiştir. Bu değerler dışarıda tutma yönteminde %100 ve 5 kat çapraz doğrulamalı yöntemde %97.31 bulunmuştur [6].

Nahzat ve Yağanoğlu (2021), diyabet tahmininde bulunmak için Pima Indian Diyabet Veri Kümesini kullanmıştır. Çalışmada, veri setindeki tüm öznitelikleri ve iki öznitelik çıkarılarak K-En Yakın Komşu, Rastgele Orman, Yapay Sinir Ağı, Karar Ağacı ve Destek Vektör Makinesi sınıflandırma algoritmaları uygulanmıştır. Her iki veri durumunda da diyabet tahmininde rastgele orman algoritması diğer algoritmalara üstünlük sağlamıştır [7].

Harman (2021), Pima Indian Diyabet Veri Kümesi üzerinden Naive Bayes ve Destek Vektör Makineleri algoritmalarını kullanarak Python dilinde diyabet tahminlemesi yapmıştır. Sınıflandırma algoritmalarının sonuçlarını arttırmak için veri setinde eksiklerin doldurulması, özniteliklerin ölçeklendirilmesi ve yeniden örnekleme işlemleri uygulanmıştır. Yapılan deneylerde doğruluk oranı, kesinlik, duyarlılık ve F1 skoru değerlerinin tamamında Destek Vektör Makineleri algoritmasının daha üstün olduğu görülmüştür [8].

Ergün ve İlhan (2021), diyabet tahmini yapmak için 520 kişiden toplanan 16 özelliğe ilişkin verileri sekiz farklı makina öğrenmesi tekniklerine uygulamışlardır. -Model sonuçlarının doğrulanması için 10 kat çapraz doğrulama şeması kullanılmıştır. Çalışmada doğruluk, kesinlik, hatırlama ve F1 skoru ölçütleri ele alınmıştır. Kullanılan tekniklerden Evrişimli Sinir Ağı modeli her ölçüt için en yüksek değerlerin elde edilmesini sağlamıştır. Bu sayede sorulacak birkaç soru ile insanların diyabet olma durumunun kolayca tespit edilebileceği ortaya konmuştur [9]. Diyabete yakalanma olasılığını hesaplamak için makina öğrenmesi yöntemlerini kullanan Bilgin (2021), en yüksek doğruluk oranını k-NN Algoritması ile %99.81 olarak bulmuştur ve bu yöntemin kullanıldığı bir bilgisayar ara yüzü ile erken tanı kiti geliştirmiştir [10].

Hadeel Tariq ve diğ. (2017), öznitelik seçimi için literatürde ilk kez Salp Sürü Algoritmasını önermişlerdir. Meme, mesane ve kolon kanserlerine yönelik Irak hastanelerinden alınan verileri Salp Sürü Algoritmasına uygulayan Hadeel Tariq ve diğ. sonuçları Parçacık Sürü Optimizasyonu ve Diferansiyel Evrim Algoritmalarından elde edilen sonuçlarla kıyaslamışlardır. Sonuçlar, Salp Sürü Algoritmasının doğruluk ve süre açısından üstün olduğunu göstermiştir [11]. Can ve diğ. (2021) tarafından farklı veri grupları farklı sınıflandırıcılar üzerinden Salp Sürü Algoritmasına tabi tutulmuştur. -Sınıflandırıcıların, farklı ölçütler için farklı sonuçlar sunduğu çalışmada tespit edilmiştir. Ayrıca veri setinin, özniteliklerin ve örneklerin boyutunun sınıflandırıcıların performansını etkilediği belirtilmiştir [12]. Hegazy ve diğ. (2020), Salp Sürü Algoritmasının doğruluk, güvenirlik ve yakınsama hızını iyileştirmek için atalet ağırlığı denilen yeni bir kontrol parametresi eklemiştir. Geliştirdikleri bu yöntemin öz nitelik seçiminde daha üstün bir performans gösterdiğini ortaya koymuşlardır [13].

Kamel ve diğ. (2021) Çekirge Optimizasyon Algoritmasına (GOA) dayalı bir öznitelik seçim yöntemi önermektedir. Diyabet Tip-II testinin doğru sonuçlarını artırmak ve farklı makine öğrenme tekniklerini kullanmak için gelişmiş bir sınıflandırıcı ile çalışma gerçekleştirilmiştir. Bunu yaparken, testin güvenilirliğini elde etmek için 10 kat çapraz doğrulama yöntemi kullanılmıştır. Öznitelik seçme tekniği, bu çalışmada önemli öznitelikleri belirlemek için kullanılmıştır. Bu yaklaşım, MATLAB yazılımı kullanılarak Pima Indian Dataset üzerinde uygulanmıştır. Çalışma sonucunda Destek Vektör Makinesi (DVM) algoritması tarafından elde edilen %97'lik doğruluk değeri çalışmanın başarılı olduğunu göstermiştir [14].

Lukmanta ve diğ. (2019) diyabetin sınıflandırılmasında ve tespitinde Bulanık Mantık ve Destek Vektör Makinesi kullanmışlardır. Veri kümesindeki öznitelikler önce DVM daha sonra bulanık mantık yönteminde kullanılmak üzere eğitilmiştir. Bulanık çıkarım işlemi çıktıyı sınıflandırmak için kullanılmıştır. Yukarıda belirtilen metodoloji, Pıma Indian Diabetes (PID) veri setine uygulanmıştır. Sonuçta diyabetli hastaların tahmininde %89.02'lik umut verici bir doğruluk elde edilmiştir [15].

Garcia-Ordás ve diğ. (2021) tarafından diyabet tahmini için Derin Öğrenme Yöntemi önerilmiştir. Literatürde nadiren birlikte kullanılan Varyasyonel Otomatik Kodlayıcı (VAE), Seyrek Otomatik Kodlayıcı (SAE) ve Konvolüsyonel Sinir Ağı (CNN) yöntemleri bu çalışmada, birlikte kullanılmıştır. Ağın geri yayılımda birbirlerinden geri bildirim almalarına izin veren bir KNN Sınıflandırıcısı ile birlikte eğitilmiştir [16].

Omisore ve diğ. (2021) diyabetin tanımı için duyusal öğrenmeye dayalı bir sistem önermişlerdir. Diyabet teşhisi için çok modlu ANFIS (MANFIS) modeli kullanmışlardır. Belirlenen özniteliklere göre diyabetin teşhisi için nöro-bulanık bir sistem önermişlerdir [17].

Vaishali ve diğ. (2017) Tip 2 Diyabetin tahmini için Makine öğrenmesiyle birlikte mevcut tanı yönetmelerini kullanmışlardır. Önerilen algoritmada, Pima veri setinden diyabet için temel öznitelikler Genetik Algoritma ile ön işleme tabi tutularak seçilmiştir ve nesnel bir Evrimsel Bulanık Sınıflandırıcı veri kümesine uygulanmıştır. Algoritmanın çalışma prensibine göre maksimum sınıflandırıcı oranı ve minimum kurallar kullanılarak öznitelik seçimi Genetik Algoritma ile yapılmıştır ve öznitelik sayısı 8'den 4'e düşürülmüştür. Böylece başarı oranının %83.04'e yükseldiği görülmüştür [18].

Köse ve diğ. (2015) DVM ile Girdap Optimizasyonu (GO) tabanlı bir algoritma geliştirmişlerdir. Diyabet hastalığının teşhisi için ise geliştirilen bu karma yapay zekâ sistemi kullanılmıştır. Oluşturulan yapıda, DVM'yi oluşturmak için Gauss (RBF) çekirdek fonksiyonunda bulunan sigma (σ) parametresinin tespitine yönelik olarak GOA kullanılmıştır. Kurulan bu yapı Pima diyabet veri seti kullanılarak değerlendirilmiştir. Süreç sonunda kazanımı gerçekleştirilen bilgiler, önerilen GOA-DVM

sistemi ile diyabet hastalığının teşhisinin tatmin edici seviyede olduğunu göstermiştir [19].

Bhargava ve diğ. (2021) Diyabet hastalığının erken ve daha iyi tahmin edilebilmesi için doğruluk oranının kritik öneme sahip olduğunu belirtmişlerdir, araştırmacılar bir dizi derin öğrenme tekniği ile ve diyabet tahmini için makine öğrenimini kullanmışlardır. Yapılan çalışmada diyabeti erken bir aşamada, en yüksek doğruluk seviyesinde ve güvenilir bir şekilde tespit etmek için bir bilgisayar modeli oluşturulmuştur. Derin öğrenme tekniğinin çalışmada kullanılan sınıflandırıcılar arasından en yüksek doğruluğa sahip olduğu (%98.07) belirtilmiş ve diyabeti erken bir aşamada tahmin etmek için kullanılabileceği ifade edilmiştir. Önerilen sistemde derin öğrenme tekniği kullanılarak PIMA veri kümesini üzerinde sınıflama işlemi yapılmıştır [20].

Bu çalışmada UCI (UCI Machine Learning Repository) veri deposundaki örnekler kullanılarak, Salp Sürü Algoritması, Yapay Arı Kolonisi Algoritması, Balina Optimizasyon Algoritması ve Karınca Kolonisi Algoritması kullanılarak öznitelik seçimi yapılmıştır. Seçilen özniteliklerin değerlendirilmesi için k-En Yakın Komşuluk (KNN), Naive Bayes (NB), Destek Vektör Makinası (DVM) ve Yapay Sinir Ağları (YSA) yöntemleri kullanılarak doğruluk, duyarlılık ve belirlilik parametreleri hesaplanmıştır. Diyabet hastası olma olasılığı için yapılan hesaplamalarda k-En Yakın Komşuluk yöntemiyle %99.04 doğruluk oranı elde edilmiştir.

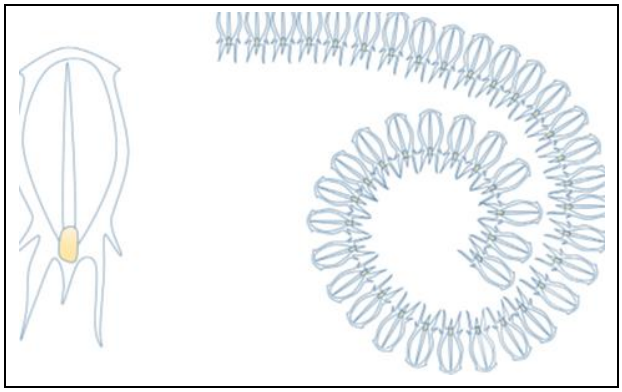
Makalenin ikinci kısmında öznitelik seçimi ve sınıflandırma için kullanılan yöntemlerden bahsedilirken üçüncü kısımda deneysel sonuçlar verilecektir. Dördüncü kısımda ise sonuçlar tartışılacaktır.

2 Yöntem

Bu çalışmada özniteliklerin seçimini sağlayan Salp Sürü Algoritması, Yapay Arı Kolonisi Algoritması, Balina Sürüsü Algoritması ve Yapay Karınca Kolonisi tanıtılmış, matematiksel modelleri ve sözde kodları verilmiştir. Sınıflandırma için kullanılan K-En Yakın Komşuluk, Destek Vektör Makinesi, Naive Bayes ve Yapay Sinir Ağları yöntemleri hakkında bilgi verilmiştir. Bu çalışmada yapılan deneyler için MATLAB R2016a yazılımı kullanılmıştır.

2.1 Salp sürü algoritması

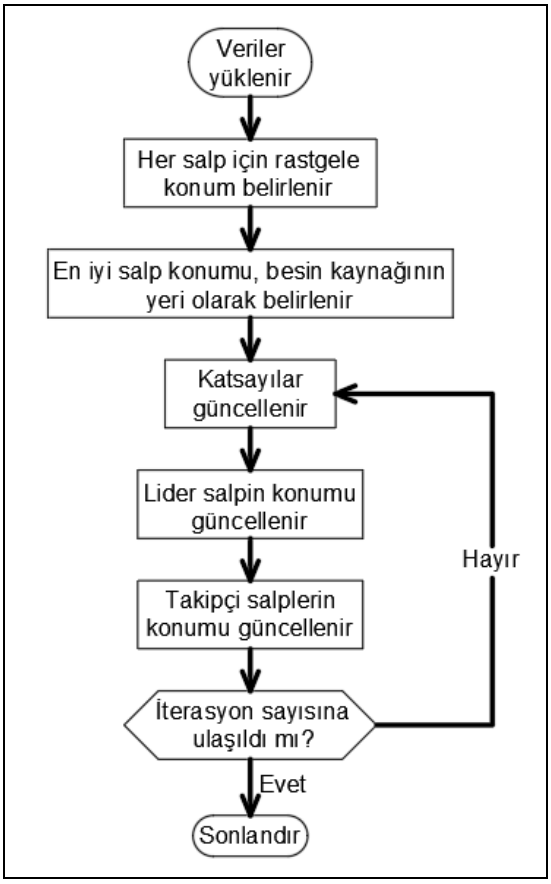
Salplar, derin denizlerde sürü halinde bulunan berrak görünüşlü denizanası benzeri bir canlıdır. Salp zinciri denen yapıyı oluşturup yiyecek aramaya yönelik hareketlerde bulunurlar (Şekil 1).



Şekil 1. Bireysel salp ve salp sürüsü [21].

Figure 1. Individual salp and salp swarm [21].

Bu hareketlerden yola çıkarak Salp Sürü Algoritması geliştirilmiştir. Sürü, hareket yönünde ilk sıradaki bir lider ve onun devamındaki takipçilerden oluşur. Bu sürünün hareketinden esinlenerek oluşturulan algoritmanın akış şeması Şekil 2'de verilmiştir.



Şekil 2. Salp sürü algoritması akış şeması.

Figure 2. Salp swarm algorithm flowchart.

Lider salpin konumunu güncellemek için kullanılan denklemler (1)'de verilmiştir.

$$X(i,d) = \begin{cases} X_f(d) + c_1 \cdot ((u_b - l_b) \cdot c_2 + l_b), & c_3 \geq 0.5 \\ X_f(d) - c_1 \cdot ((u_b - l_b) \cdot c_2 + l_b), & c_3 < 0.5 \end{cases} \quad (1)$$

Denklem (1)'deki $X(i,d)$ lider salpın konumunu, $X_f(d)$ besin kaynağının konumunu temsil etmektedir. c_2 ve c_3 rastgele üretilen değişkenler olup, u_b ve l_b sırasıyla üst ve alt sınır noktalarının değerleridir. Bu denklem lider salpın konumunun besin kaynağının konumuna bağlı olduğunu da göstermektedir. c_1, (2)'de verilen eşitlikle hesaplanan ve iterasyon sayılarına bağlı olarak bulunan bir değişkendir.

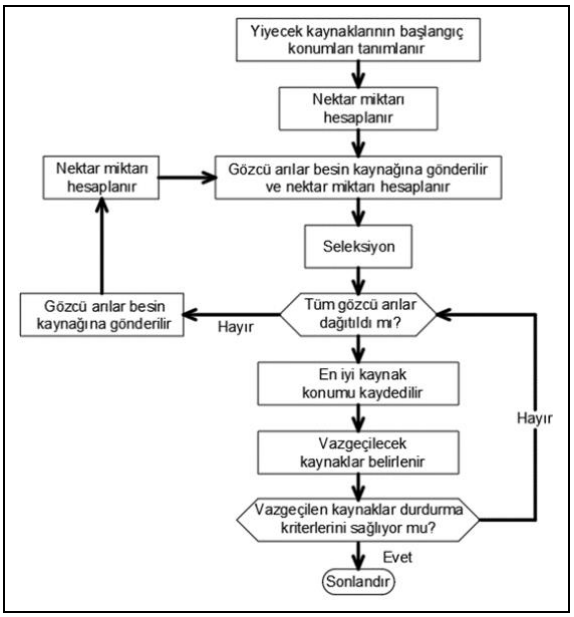
$$c_1 = 2 \cdot e^{\left(\frac{-4 \cdot t}{T}\right)^2} \quad (2)$$

Denklem (3)'teki t parametresi anlık iterasyon sayısını, T parametresi ise toplam iterasyon sayısını temsil etmektedir. Takipçi salplerin konumunu güncellemek için Denklem (3)'te verilen eşitlik kullanılır.

$$X(i,d) = \frac{1}{2} \cdot (X(i,d) + X(i-1,d)) \quad (3)$$

2.2 Yapay arı kolonisi algoritması

Yapay arı kolonisi, 2005 yılında Derviş Karaboğa ve diğ. tarafından geliştirilen, bal arılarının besin arama davranışından esinlenilen bir algoritmadır [22]. Arılar, arama uzayında besin kaynaklarının yerini tespit edip, bunu çeşitli dans hareketleri ile diğer arılara aktarmaktadır. Algoritmanın amacı en iyi besin kaynağına ulaşmaktır. Bu süreçteki görevlerine göre arılar işçi, gözcü ve izci olmak üzere üç türdür. İşçi arılar, rastgele olarak yiyecek arar. Yiyecek bulunduğunda görevli arı haline gelen işçi arılar kovana besin taşır. Görevli arılar besinleri bırakıp ya tekrar geri döner ya da çeşitli dans hareketleri ile kaynakların konumlarını tarif eder. Bu dansa bağlı olarak kovandaki izci arılar bir kaynağa yönelirler. Bu algoritmaya ilişkin akış şeması Şekil 3'te verilmiştir.



Şekil 3. Yapay arı kolonisi algoritması akış şeması.

Figure 3. Artificial bee colony algorithm flowchart.

Yiyecek kaynaklarının yerlerinin rastgele olarak belirlenmesi için Denklem (4) kullanılır.

$$X(i,d) = u_b + k_1 \cdot (u_b - l_b) \qquad (4)$$

Bu denklemdeki $X(i,d)$ yiyecek kaynaklarının iki boyutlu uzaydaki konumunu, $u_b$ve l_b sırasıyla alt ve üst sınırlarını ifade etmektedir. k_1 [0,1] aralığında üretilen rastgele bir katsayıdır. Başlangıçta yiyecek kaynaklarının konumunda bulunan işçi arılar, ardından bu konumlarının komşuluğunda besin kaynağı aramaya başlar. Bu aşamayı tarif eden Denklem (5)'te verilmiştir.

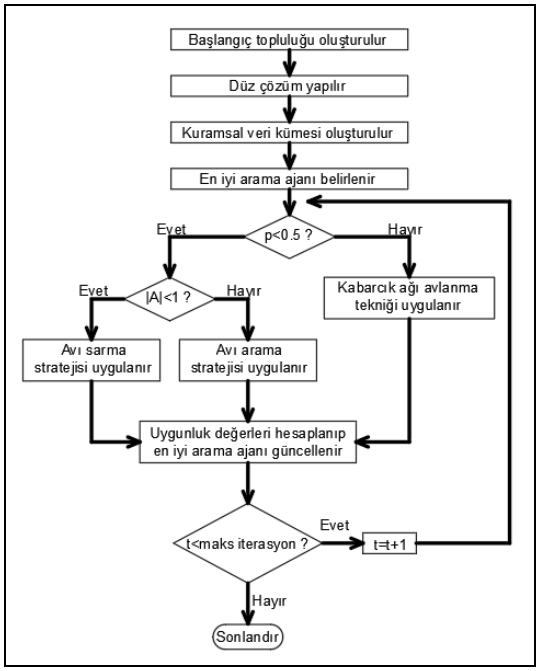
$$V(i,d) = X(i,d) + phi \cdot (X(i,d) - X(k,d)) \qquad (5)$$

$X(k,d)$ komşu besin kaynağının konumunu, phi [-1,1] aralığındaki ağırlıklandırma katsayısını ve $V(i,d)$ güncellenmiş besin kaynağını tarif etmektedir. Bu konumun alt ve üst sınırlar dahilinde olduğu teyit edildikten sonra uygunluk değerini hesaplanır. Hesaplanan değer daha iyi ise eski kaynağın yeri unutulup, yeni kaynak yeri hafızaya alınır. Kovana dönen arıların dans hareketine ve dolayısıyla nektar miktarına bağlı olarak olasılık hesabı yapılır ve bir seleksiyon işlemine tabi tutulur. Olasılık değerine bağlı olarak gözcü arılar da kaynak bölgesinde yeni çözüm üretir. Üretilen çözümlerin uygunluk değerleri hesaplanıp, öncekilerle kıyaslanarak daha iyi olan seçilir. İşçi ve gözcü arılar için bu işlemler limit değerlerine ulaştığında tekrar kâşif arı haline gelip döngü tekrarlanır.

2.3 Balina optimizasyon algoritması

Avustralyalı bilim insanları tarafından 2016 yılında önerilen Balina Optimizasyon Algoritması yeni bir sürü en uygun şekle sokma algoritması olarak kullanılmaktadır. Algoritmanın çalışması basit ve parametre kullanımına gereksinimi daha düşükken aynı zamanda yerel optimumlara takılmadan global optimuma daha iyi bir şekilde yaklaşabilmektedir [23].

Kambur balinalar yapısı gereği avlarının konumlarını tahmin edebilmekle birlikte bu avların etrafı perdelenebilmektedir. Balina Optimizasyon Algoritması bulunan en iyi çözümü avın konumu olarak ya da avın konumuna en yakın konumu çözüm olarak kabul eder. Herhangi bir arama ajanının konumu en iyi ise diğer arama parametrelerinin yeri en iyi yer göz önüne alınarak tekrardan belirlenir. Balina Optimizasyon Algoritmasına ait akış diyagramı Şekil 4'te verilmiştir [24].



Şekil 4. Balina optimizasyon algoritması.

Figure 4. Whale optimization algorithm.

Balina Optimizasyon Algoritmasında yapılan başlıca işlemler avın etrafının sarılması, ava doğru hareket etme ve avı sarma şeklindedir.

2.3.1 Avın etrafının sarılması

Avların konumu balinalar tarafından yüksek bir olasılıkla belirlenebilir. Bu belirleme neticesinde belirlenen bu alan ya da avların etrafı hava kabarcıkları ile sarılmaktadır. BOA'da uygun değer noktası bu kabarcık içine alınan avın merkez noktasıdır. Optimizasyon problemlerinin temelinde optimum noktanın bilinmezliği yatmaktadır, bu problemlerde optimum nokta ise yaklaşılabilen en iyi sonuç ya da en iyi sonuca en yakın bir hedef yer kabul edilmektedir. En iyi arama değeri ya da buna en yakın yer tespit edildikten sonra bu değerler en iyi arama değeri kullanılarak güncellenir. Bu davranış matematiksel olarak modellenmek istenirse Denklem (6) ve Denklem (7)'deki gibi ifade edilir [25].

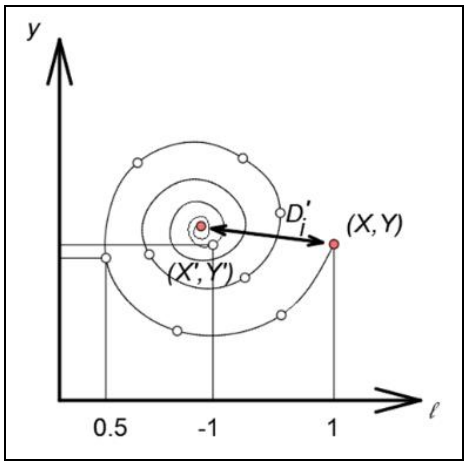
$$\vec{D} = |\vec{C}.\vec{X}(t) - \vec{X}(t)| \quad (6)$$

$$\vec{X}(t+1) = |\overrightarrow{X*}(t) \cdot \vec{A}.\vec{D}| \quad (7)$$

Denklemde verilen t gerçekleşen iterasyonu, D mesafe vektörünü A ve C katsayısı vektörlerini, X^* belirlenen en başarılı çözüm vektörünü gösterir.

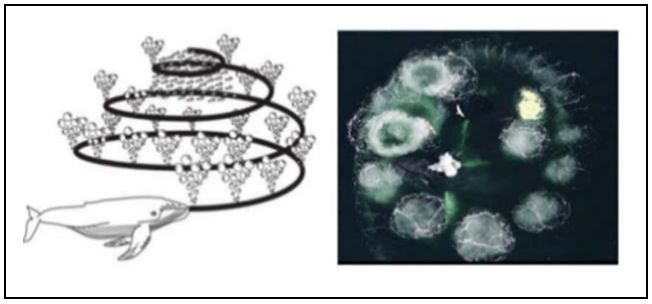
2.3.2 Ava yönelme

Balina Optimizasyonu belirlenen avın etrafındaki halkayı daraltma ve içe doğru daralan (spiral Hareket) hareket olarak iki şekilde modellenmiştir. Şekil 5'te içe doğru daralan hareket ve belirlenen en iyi sonucun konumu Şekil 6'da gösterilmiştir. Belirtilen bu yapı için hedef konum ve çözüm adayı arasındaki uzaklık belirlenerek Denklem (8) ve Denklem (9) oluşturulmuştur.



Şekil 5. Avın hareketi.

Figure 5. Movement of prey.



Şekil 6. balina optimizasyon algoritması sembolik ve gerçek gösterimi.

Figure 6. Symbolic and real representation of whale optimization algorithm.

$$\vec{x}(t+1) = \vec{D}'.e^{bl}.\cos(2\pi l) + \overrightarrow{X*}(t) \quad (8)$$

$$\vec{D}' = \vec{X}*(t) - \vec{X}(t) \quad (9)$$

2.3.3 Av arama

Global çözüm için, çözüme sahip değerlerin en son yerleri, belirlenen en iyi konum değerinin yerine, rastlantısal olarak belirlenen bir çözüm konumunun çevresinde belirlenir. İfade edilen bu işleme ait model Denklem (10) ve Denklem (11)'de gösterilmiştir [25].

$$\vec{D}' = \overrightarrow{C}.\overrightarrow{X_{\text{rand}}} - \vec{X} \quad (10)$$

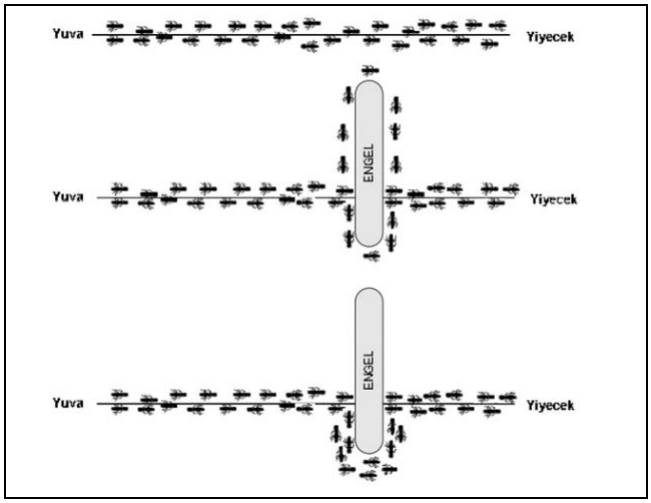
$$\vec{X}(t+1) = \overrightarrow{X_{\text{rand}}} - \vec{A} \cdot \vec{D} \quad (11)$$

X_{rand} vektörü rastgele tercih edilen bir sonuç değerini ifade etmektedir. X vektörünün değeri Küresel ya da yerel arama yapılacağına karar vermektedir. A vektörü değeri için, $x < 1$ veya $x > 1$ durumlarında belirlenen en iyi konumdan daha uzak bir nokta tercih edilebileceğinden, bunlar küresel arama olarak kabul edilmektedir.

2.4 Karınca kolonisi algoritması

Görme duyularına sahip olmayan karıncalar yiyecek kaynakları ile yuvaları arasında en kısa yolu bulma yeteneğine sahiptirler. Bu özelliğin akabinde değişen çevre koşullarına uyum sağlayabilme yetenekleri de vardır. Karıncaların takip ettikleri, kullandıkları en kısa yol dış etkenler sonucu artık en kısa yol değil ise karıncalar tekrardan seçenekler arasından en kısa mesafeyi tercih edebilmektedirler [26].

Şekil 7'de de olduğu gösterildiği haliyle karıncalar, ilk anda yuva ile yiyecek arasında düz bir çizgiyi takip etmekte iken feromon olarak bilinen bir hormonu salgılamakta ve kendinden sonraki karıncalar bu kimyasal izi takip ederek yollarını bulabilmektedirler. Takip edilen nu yol üzerinde bir engel yerleştirildiğinde kimyasal bir iz olan feromonu kaybettiklerinden dolayı, karıncalar tercih edebilecekleri iki seçenekten herhangi birini, gelişigüzel olarak seçmektedirler. Kullanılan bu iki seçenekten, kısa olan yol daha fazla geçişe imkân sağlamasından dolayı iz olarak kalan feromon miktarını da arttırmakta ve buharlaşmanın da daha az olmasını sağlamaktadır. Buna bağlı olarak, zaman içerisinde karıncalar artan bir şekilde kısa yolu tercih etmeye başlayacaklardır. Belirli bir süre sonunda ise tüm karıncalar kısa olan yolu tercih edeceklerdir.



Şekil 7. Karınca kolonisi algoritması.

Figure 7. Ant colony algorithm.

Karınca Kolonisi Algoritması yapay olarak temsil edilen karıncalardan oluşturulmakla birlikte, oluşturulan feromon izinin yenilenmesiyle devamlı olarak yinelenen oluşumdur. Algoritma kullanılırken, yapay karıncalar ile yenilenen feromon kalıntılarıyla birlikte istenen en ideal çözümün bulunması için amaç fonksiyonu yeni değerler ile birlikte güncellenmektedir. İlgili yapıya ait algoritma Şekil 8'deki gibidir.

Algoritmanın çalışmasında uygulanan adımlar, yapay karıncaların gezinmeleri sonucunda üzerinden tekrar geçilen yollar üzerinde kalan feromon miktarlarının arttırılması ancak

yapısı gereği feromonun bir miktarının uçarak azaltılmasının gerçekleştirilmesi bunun sonucunda en iyi değerin bulunması ve bu değere bağlı olarak global feromon değerinin yeniden belirlenmesinin yapılması ve yapay karıncaların değeri yeniden belirlenen bu feromon izlerine olan duyarlılıkları geri yeni hareketlerini gerçekleştirmeleridir. Bu algoritma Şekil 8'de gösterilmiştir.

Şekil 8. Karınca kolonisi algoritması.

Figure 8. Ant colony algorithm.

Adım : Feromon için ilk değerler belirlenir,
Adım : Karıncalar rastgele olarak düğümlere atanır,
Adım : Belirlenen her bir karınca, başlangıçta verilen yerel arama olasılığına göre hareketini tamamlanır,
Adım : Karıncalar için gidilen yol her bir karınca için hesaplanarak yerel feromon değeri belirlenir,
Adım : En başarılı değer bulunur ve global feromon değeri tekrar hesaplanır
Adım : Başarım değeri sağlanana kadar Adım 2'ye gidilir [27].

2.4.1 Geçiş kuralı

KKA' da iterasyon sırasında k karıncası için i noktasından j noktasına gitmek için iki seçenek bulunmaktadır. Birinci seçenek, seçenek olarak bulunan yollar içerisinde feromon miktarı dikkate alınarak hesaplanan değere göre seçim yapmaktır. Feromon miktarına bağlı olarak tercih yama olasılığı yüksektir. Diğer seçenek ise yollardaki feromon değerine dikkat edilerek belirlenen olasılık dağılımına göre tercih edilir.

Yukarıda bahsedilen geçiş kurulanına göre oluşturulan Denklem (12) aşağıda verilmiştir.

$$j = \max_{u \in j_k(i)} (\mathrm{i})\{[\tau(i,u)]^a \, x[\eta(\mathrm{i},u)]^\beta\} \;\; eğer \; q \leq q_0 \qquad (12)$$

Denklemde kullanılan $\tau(i,u)$, (i,u) yolundaki feromon miktarı, $h(i,u)$ i noktası ile u noktası arasındaki mesafenin tersidir. $j_k(i)$, i noktasında bulunan k karıncasının gitmediği noktaları, $\beta(b>0)$feromon güncellemesinde uzaklığın etkisini ve $q < q_0$ çözüm arama parametresini göstermektedir.

2.4.2 Feromon güncellemesi

Feromon miktarının güncellenmesi tüm karıncalar etaplarını tamamladıktan sonra gerçekleştirilir. İlk adımda tüm yollarda bulunan feromonlar, belirlenen bir katsayı ile buharlaştırılır. İkinci adımda ise karıncaların kullandıkları yolda bulunan feromon miktarı, bu yoldan geçen karıncanın kat ettiği toplam mesafe tersi ile arttırılmaktadır [28]. Bunun sonucunda karıncaların kullandıkları daha kısa olan yollar üzerinde feromon miktarının arttığı görülmektedir. Feromon güncellemesi Lokal feromon güncellemesi ve Global Feromon değerinin belirlenmesi iki şekilde yapılabilmektedir.

2.4.2.1 Lokal feromon güncellemesi

Bütün karıncaların etabı tamamlaması sonrasında, feromon değeri belirli bir miktarda uçucu etki sonucunda azaltılır, karıncaların etap boyunca geçtiği konumlarda belirlenen bir değerde feromon miktarının artması sağlanabilir. Bahsedilen bu yapılar Denklem (13)'e göre yapılmaktadır:

$$t_{ij}(t+1) = (1-r)t_{ij}(t) + \sum_{k=1}^{m} \Delta t_{ij}^{k}\,(t+1) \qquad (13)$$

Denklemde verilen $t_{ij}(t+1)$ başlangıç feromon düzeyidir ve $\sum_{k=1}^{m} \Delta t_{ij}^{k}\,(t+1)$ lokal feromon güncelleme parametresidir.

2.4.2.2 Global feromon güncellemesi

Global feromon güncellemesi, etabı tüm karıncaların tamamlaması sonucunda gerçekleştirilir. Her bir karıncanın gittiği mesafe kaydedilir be mesafeler içerisinde en az mesafe giden karınca tespit edilir. Tespit edilen bu karıncanın tercih ettiği yollardaki feromon miktarları Denklem (14)'e göre arttırılmaktadır

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + \Delta\tau_{ij}^{k}(t+1) \qquad (14)$$

2.5 Sınıflandırma algoritmaları

Çalışmada, sınıflandırma işlemleri K-En Yakın Komşuluk, Destek Vektör Makinesi, Naive Bayes ve Yapay Sinir Ağları Algoritmalarına dayalı olarak yapılmıştır.

2.5.1 K-En yakın komşu algoritması

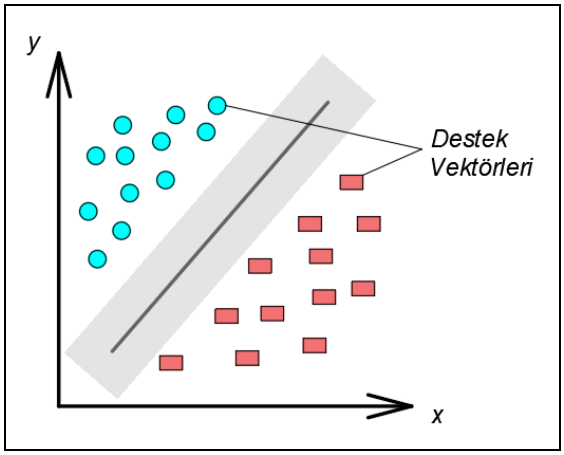
K-En Yakın Komşu Algoritması (K-Nearest Neighbor Algorithm -KNN) veri madenciliğinde kullanılan bir sınıflandırma yöntemidir. KNN sınıflandırması veri setinde bulunan verilere bağlı olarak öğrenme gerçekleştirir. KNN denetimli öğrenme ile nesne sınıflandırılmasında kullanılmaktadır. Bu sınıflama en yakın nesneye göre yapılmaktadır. Sınıflamaya giren yeni örnek veri, KNN içerisindeki önceden belirlenen ya da algoritma tarafından oluşturulan sınıflara bağlı olarak k sayısı değerince en yakın komşu konumuna bakılarak sınıflama yapar. KNN ayrıca eski ve yeni durumlar için yakınlık hesaplamalarında da kullanılır. Sınıflandırma KNN sadece bir model kullanmaz hafızaya dayalıdır. Kullanılan algoritma şu şekilde tarif edilmektedir:

1. Uygun K parametresini belirlemek için; kullanılan verilere bağlı olarak K değeri minimum değer olarak 1 veya eğitim verisinin değeri maksimum değer olarak alınır,
2. Test verileri ile veriler arasındaki mesafe sıralı bir şekilde hesaplanır. Mesafeyi hesaplamak için kullanılan Öklid mesafesi Denklem (15)'te verilmiştir,
3. Hesaplanan mesafeler küçükten büyüğe doğru sıralanır,
4. K parametresine bağlı olarak k adet En yakın mesafe belirlenir,
5. K adet sınıftan çoğunlukta olan sınıf belirlenir ve uygun sınıf ile eşleştirilir.

$$\text{Öklid mesafesi} = \sqrt{\sum_{i=1}^{n}(pi - qi)^2} \quad (15)$$

2.5.2 Destek vektör makinası

Corinna ve Vladimir tarafından geliştirilen Destek Vektör Makinası yöntemi denetimli makine öğrenme yöntemlerinden biridir [29]. Bu yöntemdeki temel hedef, sınıf etiketlerinin kullanılmasıyla eğitim verilerini ayrıştırılabilecek bir fonksiyonun bulunmasıdır. Sınıf etiketleri genellikle "evet-hayır" veya "pozitif-negatif" gibi adlandırıldığından, Destek Vektör Makinası yöntemi ile bu ifadeler arasındaki en uygun ayırıcı yani hiperdüzlem bulunmaktadır. Yani, DVM, iki karar sınırı arasındaki mesafeyi maksimize ederek optimal ayırıcı hiperdüzlemi bulur. Matematiksel olarak, $w^T x + b = 0$ ile gösterilen optimal ayırıcı hiperdüzlemini bulmak için $w^T x + b = -1$ hiperdüzlemi ile $w^T x + b = 1$ hiperdüzlemi arasındaki $\frac{2}{\|w\|}$ mesafesini maksimize eder. Başka bir ifadeyle $\frac{\|w\|}{2}$ terimini minimize etmeye çalışır. Bunun için her iki sınıftan ve hiperdüzleme yakın verileri destek vektörü olarak kullanır. Şekil 9'da hiperdüzlem ve destek vektörlerine ilişkin görsel verilmiştir.



Şekil 9. Destek vektör makinası.

Figure 9. Support vector machine.

2.5.3 Naive bayes

Olasılık tabanlı olarak sınıflandırma işlemi yapan bu yöntem Bayes teoremini temel almaktadır. Görüntü işleme ve sinyal işleme alanlarında tercih edilen bir yöntem olarak bilinmektedir. Bu yöntemde sınıf üyelik olasılıkları kullanılarak hangi sınıfta olma olasılığı tahmini yapılır. Kolay kullanımı ve iyi sonuç vermesi yöntemin olumlu yanları iken olumsuz yanı, sınıf ve nitelikler bakımından bağımsızlık varsayımı gerektirmesidir. Bu yöntemde, olayların birbirinden bağımsız olduğu düşünülerek işlem yapılmaktadır.

Bayes teoremine göre, x değerine bağlı olarak y değerinin rastgele olasılığını hesaplamak için kullanılan Denklem (16):

$$\mathrm{P(y|x)} = \frac{\mathrm{P(x|y)P(y)}}{\mathrm{P(x)}} \quad (16)$$

Burada; $= (x_1, x_2, x_3, \dots, x_n)$ şeklinde yazılabilen bir vektördür. Bu durumda Bayes denklemi şöyle olur:

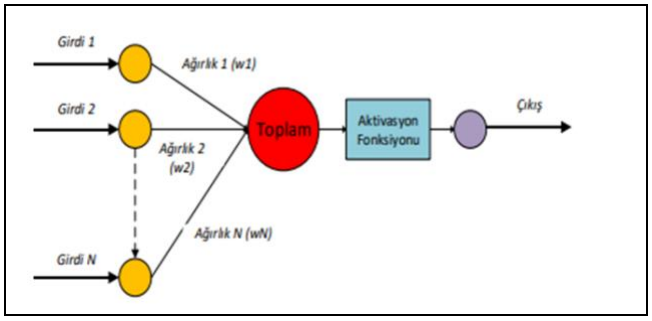
$$\mathrm{P}(y|x_1, \dots, x_n) = \frac{\mathrm{P}(x_1|\mathrm{y})\ \mathrm{P}(x_2|\mathrm{y}) \dots \mathrm{P}(x_n|\mathrm{y})\mathrm{P(y)}}{\mathrm{P}(x_1)\mathrm{P}(x_2) \dots \mathrm{P}(x_n)} \quad (17)$$

Bu durumda, $P(y/x)$'in maksimum değerini veren y hedef sonucu şöyle yazılır:

$$y = argmax_y P(y) \prod_{i=1}^{n} p(x_i|y) \quad (18)$$

2.5.4 Yapay sinir ağları

YSA problemlerin çözümünde insan beyninin anatomisinden faydalanılarak geliştirilen bir modeldir. YSA temek olarak insan beyninde öğrenmenin gerçekleştiği kısımlar olan sinir hücrelerinin yapısı baz alarak geliştirilmiştir. Benzetilen bu yapı sonucunda yüksek miktardaki verilerin işlenerek doğru sonuçlar elde edilmesi gerçekleştirilmiştir. YSA kullanımı için sinir hücrelerine benzer yapılar eğitilerek problem çözme işlemi gerçekleştirilmektedir. Yapay sinir ağına ait temel yapı Şekil 10' da gösterilmiştir.



Şekil 10. YSA yapısı.

Figure 10. ANN structure.

YSA'yı meydana getiren yapay sinir hücrelerinin genel yapısı; girdi (x_i), ağırlık (w_i) toplam fonksiyonu, aktivasyon fonksiyonu ve çıkış ile birlikte beş bölümden meydana gelmektedir. Sisteme ait olan girdiler eğitim esnasında belirlenen kendi ağırlık değerleri ile çarpılarak toplanır. Oluşturulan bu değerin bir çıkış değerini ifade etmesi amacıyla bir aktivasyon değeri ile çarpılır ve bir sonuç elde edilir. YSA modelinin başarımını belirleyen parametrelerden biri öğrenme esnasında oluşturulan ağırlık değerlerine bağlıdır. Bir diğer başarım parametresi ise yapı oluşturulurken belirlenen gizli katman sayısıdır. Gizli katman sayısının az ya da çok olması çıkış değerini doğrudan etkilemektedir [30]. YSA'da girdiler, daha önce belirlenen ağırlık değerleri ile çarpılır, elde edilen bu çarpım değerlerinin toplamları neticesinde net girdi değeri hesaplanmaktadır. Net değeri hesaplamaya ilişkin ağırlıklı toplam fonksiyonu Denklem (19)'da verilmiştir.

$$NET = \sum_{i}^{n} w_i * x_i \quad (19)$$

Bu denklemde x_i hücreye gelen girdileri, w_i girdilerin ağırlıklarını ve n hücreye gelen toplam girdi sayısını ifade etmektedir. Elde edilen toplam değerin anlamlı bir çıktı olarak aktarılması için aktivasyon fonksiyonu olarak isimlendirilen bir fonksiyona tabi tutulması gerekmektedir. Aktivasyon fonksiyonu sonucunda giriş değerlerine karşılık anlamlı bir çıkış değeri elde edilmektedir. En çok sigmoid fonksiyonu aktivasyon fonksiyonu olarak kullanılmaktadır.

2.6 Veri Seti

Çalışma kapsamında University of California Irvine (UCI) tarafından sağlanan veri deposu kullanılmıştır. Veri deposundaki erken dönem diyabet risk tahmini veri seti seçilmiştir. Bu veri setinde 17 öznitelik ve 520 örnek bulunmaktadır. Tablo 1'de kullanılan veri setindeki özniteliklere ait açıklamalar verilmiştir.

Tablo 1. Özniteliklere ilişkin açıklamalar.

Table 1. Descriptions of attributes.

No	Öznitelik	Değer
1	Yaş	[16, … ,90]
2	Cinsiyet	Erkek:1, Kadın:0
3	Poliüri	Evet:1, Hayır:0
4	Polidipsi	Evet:1, Hayır:0
5	Hızlı kilo kaybı	Evet:1, Hayır:0
6	Halsizlik	Evet:1, Hayır:0
7	Polifagi	Evet:1, Hayır:0
8	Genital Pamukçuk	Evet:1, Hayır:0
9	Bulanık görme	Evet:1, Hayır:0
10	Kaşıntı	Evet:1, Hayır:0
11	Sinirlilik	Evet:1, Hayır:0
12	Geç iyileşme	Evet:1, Hayır:0
13	Kısmi parezi	Evet:1, Hayır:0
14	Kas sertliği	Evet:1, Hayır:0
15	Alopesi	Evet:1, Hayır:0
16	Obezite	Evet:1, Hayır:0
17	Sınıf	Pozitif:1, Negatif:0

3 Deneysel sonuçlar

Çalışmada yapılan işlemlere ilişkin akış şeması Şekil 11'de verilmiştir. Verilerin yüklenmesiyle başlayan deneysel çalışmalar, verilerin 0 ve 1'ler cinsinden ifade edilmesinden sonra özniteliklerin seçilmesiyle devam etmiştir. Öznitelik seçiminde Yapay Arı Kolonisi Algoritması, Salp Sürü Algoritması, Balina Optimizasyon Algoritması ve Karınca Kolonisi Algoritması kullanılmıştır. Seçilen özniteliklere göre sınıflandırma başarısının hesaplanmasında diğer değerlere karşı daha iyi bir sonuç verdiği için $K = 3$ seçilerek k-En Yakın Komşuluk, Destek Vektör Makinaları, Naive Bayes ve Yapay Sinir Ağları yöntemleri kullanılmıştır. İterasyon sayıları tüm algoritmalar için aynı tutularak 200 seçilmiştir. Metasezgisel algoritma ile sınıflama başarısı bir optimizasyon problemi olarak kabul edilmiştir. Doğası gereği rastgele bir şekilde başlayan öznitelik seçim işlemi hatanın sıfır olduğu ya da sıfır değerine en yakın olduğu durumda sonlanmıştır ve o anda elde edilen öznitelikler sınıflama algoritmasında kullanılmıştır. Bu sayede daha az öznitelik ile daha başarılı bir sınıflama başarısının elde edilmesi istenmiştir.

Şekil 11. Çalışmada kullanılan yöntemin akış şeması.

Figure 11. Flow chart of the method used in the study.

Çalışmada kullanılan modelin performansını ölçmek için doğruluk, duyarlılık ve belirlilik parametreleri hesaplanmıştır. Denklem (20)'de olduğu gibi modelin yaptığı sınıflandırma sonuçlarıyla gerçek verilerin sonuçlarının aynı olduğu örnek sayısının toplam örnek sayısına oranı doğruluk parametresini vermektedir.

$$Doğruluk = \frac{TP + TN}{TP + TN + FP + FN} \quad (20)$$

Denklem 20'de verilen TP pozitif doğru, TN negatif doğru, FP pozitif yanlış ve FN negatif yanlış sonuçları temsil etmektedir. Duyarlılık parametresi, gerçek pozitiflik oranı olarak da bilinmektedir. Denklem (21)'de verildiği gibi pozitif doğru örnek sayısının pozitif doğru ve negatif yanlış örnek sayılarının toplamına oranı duyarlılığı ifade etmektedir.

$$Duyarlılık = \frac{TP}{TP + FN} \quad (21)$$

Çalışmada incelenen bir diğer parametre olan kesinlik; doğru pozitif örnek sayısının, toplam pozitif örnek sayısına oranı şeklinde Denklem (22)'de verildiği gibi hesaplanmaktadır.

$$Kesinlik = \frac{TP}{TP + FP} \quad (22)$$

Duyarlılık ve kesinlik parametrelerini daha anlamlı hale getirebilmek için kullanılan $F1$ skoru parametresi, Denklem (23)'te olduğu gibi kesinlik ve duyarlılık parametreleri üzerinden formülleştirilmektedir.

$$F1\ skor = \frac{2 * Kesinlik * Duyarlılık}{Kesinlik + Duyarlılık} \quad (23)$$

Diyabet tahmininde kullanılmak üzere yapılan bu çalışmada özniteliklerin azaltılması ile elde edilen sonuçlar Tablo 2'de ve tüm özniteliklerin kullanılması ile elde edilen sonuçlar Tablo 3'te verilmiştir. Tablo 4'te bu çalışmadaki veri setinin kullanıldığı literatürdeki bazı çalışmalar özet olarak verilmiştir.

Tablo 2. Özniteliklerin azaltılması ile elde edilen sonuçlar.

Table 2. Obtained results by reducing the features.

		Doğruluk	Duyarlılık	Kesinlik	F1 Skoru	Seçilen öznitelikler
Salp Sürü Algoritması	KNN	%95.19	%98.43	%94.02	0.96	2-3-4-8-9-10-12-13-14-16
	NB	%91.34	%89.06	%96.61	0.92	1-2-3-6-8-9-11-14
	DVM	%95.19	%93.75	%90.90	0.92	1-2-3-4-6-9-10-11-12-13
	YSA	%96.63	%95.70	%98.79	0.97	2-3-4-7-8-10-11-12-13-15-16
Yapay Arı Kolonisi Algoritması	KNN	%98.07	%98.43	%98.43	0.98	2-3-4-5-7-8-10-11-12
	NB	%95.19	%95.31	%96.82	0.96	2-3-4-7-8-9-14-15
	DVM	%93.26	%95.31	%93.84	0.94	2-3-4-5-11-12-15-16
	YSA	%95.67	%94.92	%97.98	0.96	2-3-4-5-7-9-10-11-12-14-15-16
Balina Optimizasyon Algoritması	KNN	%98.07	%98.43	%98.43	0.98	2-3-5-7-8-9-10-12-14
	NB	%91.34	%93.75	%92.30	0.93	2-4-5-7-10-11-14-15-16
	DVM	%93.26	%95.31	%93.84	0.94	2-3-4-5-8-9-10-11-12-14-15-16
	YSA	%94.23	%97.26	%93.60	0.95	2-3-4-5-6-7-8-10-11-12-15-16
Karınca Kolonisi Algoritması	KNN	%99.04	%100	%98.46	0.99	2-3-4-5-7-8-10-11-12-14-15-16
	NB	%90.38	%92.18	%92.18	0.92	1-2-3-4-5-7-8-9-13
	DVM	%96.15	%95.31	%98.38	0.96	1-2-3-4-5-6-8-9-13-14-15-16
	YSA	%95.43	%96.48	%96.10	0.96	2-3-4-5-7-8-9-10-14-16

Tablo 3. Tüm özniteliklerin kullanılması ile elde edilen sonuçlar.

Table 3. Obtained results using all features.

		Doğruluk	Duyarlılık	Kesinlik	F1 Skoru
Salp Sürü Algoritması	KNN	%88.46	%89.06	%96.42	0.90
	NB	%89.42	%89.06	%93.44	0.91
	DVM	%90.38	%93.75	%90.90	0.92
	YSA	%93.99	%94.92	%95.29	0.95
Yapay Arı Kolonisi Algoritması	KNN	%89.42	%85.93	%96.49	0.90
	NB	%90.38	%92.18	%92.18	0.92
	DVM	%88.46	%90.62	%90.62	0.90
	YSA	%93.02	%94.14	%94.50	0.94
Balina Optimizasyon Algoritması	KNN	%94.23	%90.62	%100	0.95
	NB	%88.46	%90.62	%90.62	0.90
	DVM	%89.42	%90.62	%92.06	0.91
	YSA	%82.69	%94.14	%80.87	0.87
Karınca Kolonisi Algoritması	KNN	%89.42	%85.93	%96.49	0.90
	NB	%85.57	%92.18	%85.50	0.88
	DVM	%92.30	%89.06	%98.27	0.93
	YSA	%86.05	%87.50	%89.60	0.88

Tablo 4. Literatürdeki bazı çalışmaların özeti.

Table 4. Summary of some studies in the literature.

Çalışma	Sınıflandırma Yöntemi	Veri Seti Bölme Tekniği	Doğruluk Sonucu
Le ve diğ. [1]	Adaptif Parçacık Gri Kurt Optimizasyonu ve Çok Katmanlı Algılayıcı	Dışarıda tutma	%97.00
Al-Behadili ve Ku-Mahamud [31]	Öznitelik Seçimi: Açgözlü Tepe Tırmanması Algoritması Sınıflandırma: Bulanık Mantık	Eğitim: %80, Test: %20	%97.69
Özer [32]	Uzun-Kısa Dönem Bellek Ağları Yaklaşımı	10-kat çapraz doğruluma	%98.65
Chaves [33]	Yapay Sinir Ağları	10-kat çapraz doğruluma	%98.10
Akyol ve Karacı [6]	Oylama Topluluk Sınıflandırıcısı	10-kat çapraz doğrulama	%97.31
Ergün ve İlhan [9]	Evrişimsel Sinir Ağı	5-kat çapraz doğrulama	%99.04
Bu çalışma	Öznitelik seçimi: Karınca Kolonisi Algoritması Sınıflandırma: KNN	Eğitim: %80, Test: %20	%99.04

Tablo 2'deki metasezgisel yöntemler kullanılarak öznitelik seçilmesiyle elde edilen başarı ölçütlerinin ve Tablo 3'teki öznitelik seçimi yapmaksızın bütün özniteliklerin kullanılmasıyla elde edilen başarı ölçütlerinin üstünde sonuçlar verdiğini göstermektedir. Öznitelik azaltılmasının başarıyı ve diğer başarı ölçütlerini artırdığı görülmüştür.

Tablo 2 ve Tablo 3'te incelenen diğer başarı kriterleri duyarlılık, kesinlik ve $F1$ skorudur. Tablo 2'de $F1$ skorunun en yüksek değerinin 0.99 ile Karınca algoritması ve KNN ikilisine ait olduğu görülmektedir. Öznitelik sayısının azaltılması yanında sınıflama yöntemlerinden KNN $K = 3$ değeriyle tüm sınıflama yöntemleri içinde en başarılı sonuçları vermiştir. Tablo 4'te literatürdeki diğer çalışmalarla yapılan kıyaslamaya göre %99.04 olarak elde edilen doğruluk oranının diğer çalışmalarla karşılaştırılabilecek düzeyde yüksek bir değerde olduğu görülmektedir.

4 Sonuçlar

Büyük boyutlu verilerin herhangi bir kayba uğramadan analiz edilmesi için günümüzde makine öğrenmesi teknikleri uygulanmaktadır. Bu çalışmada diyabet hastalığının erken teşhis edilebilmesi için bir inceleme yapılmıştır. Veri seti olarak, University of California Irvine (UCI) tarafından sağlanan açık kaynaklı veri deposu kullanılmıştır. Seçilen sette 16 öznitelik ve 520 örnek bulunmaktadır. Özniteliklerin seçilmesinde metasezgisel yöntemlerden en çok kullanılan Yapay Arı Kolonisi Algoritması ve Karınca Kolonisi Algoritması ile birlikte Salp Sürü Algoritması ve Balina Optimizasyon Algoritması da kullanılmıştır ve 4 adet sınıflama algoritmasından faydalanılmıştır. Farklı sınıflandırma yöntemleri kullanılarak hastalığın erken teşhisinde kullanılacak parametrelerin performansları tek tek hesaplanmıştır. En yüksek doğruluk, duyarlılık ve kesinlik değerleri sırasıyla %99.04, %100 ve %98.46 olarak Karınca Kolonisi Algoritması ve KNN ikilisi ile bulunmuştur.

Öznitelik sayısının azaltılması başarıyı ve diğer başarı ölçütlerini arttırmıştır. Sonuçların başarısını desteklemek için incelenen duyarlılık ve kesinlik ölçütleri ve bunlara bağlı olarak $F1$ skoru da yüksek çıkmıştır.

Eldeki 16 öznitelikten oluşan diyabet verileriyle yapılan sınıflama sonuçları gösteriyor ki özniteliklerin azaltılması başarı oranını artırıyor. Bu amaçla hangi öznitelik kombinasyonunun en başarılı sonucu verdiğini araştırmak için $2^{16} = 65535$ adet kombinasyon ele alınması gerekmektedir. Bunun araştırılması için gerekli işlem ve zaman oldukça fazla olacaktır. Ortalama olarak öngörülen süre yaklaşık 30 dakikadır. Buna karşılık metasezgisel yöntemleri kullanarak elde edilen sonuçlar birkaç adımda ve yaklaşık 20 saniye içinde bulunmuştur.

Literatürde aynı veri setiyle yapılan diğer çalışmalar Tablo 4'te verilmiştir. Buna göre neticede bu çalışmada elde edilen sonuçlar, sağlık kuruluşuna başvuran bir hasta için diyabet hastası olma olasılığının %99.04 doğruluk oranında tahmin edilebileceğini göstermektedir. Bu sayede diyabet teşhisi için gereken sürenin kısalmasına ve dolayısıyla sağlık kuruluşlarındaki iş yükünün azalmasına katkı sağlanacaktır.

5 Conclusions

Machine learning techniques are applied to analyze large-scale data without any loss. In this study, an examination was made for early prediction of diabetes. The open-source data repository provided by the University of California Irvine (UCI) was used as the dataset. The selected set contains 16 features and 520 samples. In the selection of the features, Artificial Bee Colony Algorithm and Ant Colony Algorithm, which are the most used metaheuristic methods, along with Salp Swarm Algorithm and Whale Optimization Algorithm were also used and 4 classification algorithms were used. The performances of the parameters to be used in the early diagnosis of the disease were calculated one by one using different classification methods. The highest accuracy, sensitivity and precision values were found with the Ant Colony Algorithm and KNN pair as 99.04%, 100% and 98.46%, respectively.

Reducing the number of features increased the success and other success criteria. The criteria of sensitivity and precision examined to support the success of the results and the corresponding F1 score was also high.

The classification results made with the diabetes data consisting of 16 features show that reducing the features increases the success rate. For this purpose, $2^{16} = 65535$combinations should be considered in order to investigate which feature combination gives the most successful result. The process and time required to investigate this will be quite a lot. The estimated time on average is about 30 minutes. In contrast, results obtained using metaheuristic methods were found in a few steps and in about 20 seconds.

Other studies with the same data set in the literature are given in Table 4. Accordingly, the results obtained in this study show that the probability of having diabetes for a patient applying to a health institution can be predicted with an accuracy rate of 99.04%. In this way, it will contribute to the shortening of the time required for the diagnosis of diabetes and thus to the reduction of the workload in health institutions.

6 Yazar katkı beyanı

Gerçekleştirilen çalışmada Tuğberk ÖZMEN ve Haldun SARNEL literatür taramasının yapılması, Üzeyir KUZU deneysel sonuçların elde edilmesi aşamalarında, Yücel KOÇYİĞİT ise fikrin oluşması, sonuçların incelenmesi ve içerik açısından makalenin kontrol edilmesi aşamalarında katkı sunmuştur.

7 Etik kurul onayı ve çıkar çatışması beyanı

Hazırlanan makalede etik kurul izni alınmasına gerek yoktur. Ayrıca hazırlanan makalede herhangi bir kişi/kurum ile çıkar çatışması bulunmamaktadır.

8 Kaynaklar

[1] Le TM, Vo TM, Pham TN, Dao SVT. "A novel wrapper–based feature selection for early diabetes prediction enhanced with a metaheuristic". *IEEE Access*, 9, 7869-7884, 2021.

[2] Kurt MS, Ensarı T. "Diabet diagnosis with support vector machines and multi layer perceptron". *Electric Electronics, Computer Science, Biomedical Engineerings' Meeting*, İstanbul, Turkey, 20-21 April 2017.

[3] Sun H, Saeedi P, Karuranga S, Pinkepank M, Ogurtsova K, Duncan BB, Magliano DJ. "IDF diabetes atlas: global, regional and country-level diabetes prevalence estimates for 2021 and projections for 2045". *Diabetes Research and Clinical Practice*, 183, 109-119, 2022.

[4] International Diabetes Federation. "International Diabetes Federation-Facts & Figures". https://www.idf.org/ (24.06.2022).
[5] Özlüer Başer B, Yangın M, Sarıdaş ES. "Makine öğrenmesi teknikleriyle diyabet hastalığının sınıflandırılması". *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 25(1), 112-120, 2021.
[6] Akyol K, Karacı A. "Diyabet hastalığının erken aşamada tahmin edilmesi için makine öğrenme algoritmalarının performanslarının karşılaştırılması". *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 9, 123-134, 2021.
[7] Nahzat S. Yağanoğlu M. "Diabetes prediction using machine learning classification algorithms". *Avrupa Bilim ve Teknoloji Dergisi Özel Sayı*, 24, 53-59, 2021.
[8] Harman G. "Destek vektör makineleri ve naive bayes sınıflandırma algoritmalarını kullanarak diabetes mellitus tahmini". *Avrupa Bilim ve Teknoloji Dergisi*, 32, 7-13, 2021.
[9] Ergün ÖN, İlhan HO. "Early-stage diabetes prediction using machine learning methods". *European Journal of Science and Technology*, 29, 52-57, 2021.
[10] Bilgin G. "Makine öğrenmesi algoritmaları kullanarak erken dönemde diyabet hastalığı riskinin araştırılması". *Zeki Sistemler Teori ve Uygulamaları Dergisi*, 4(1), 55-64, 2021.
[11] Tarik IH, Mazher WJ, Uçan ON, Bayat O. "Feature selection using salp swarm algorithm for real biomedical datasets". *IJCSNS International Journal of Computer Science and Network Security*, 17(12), 13-20, 2017.
[12] Can C, Kaya Y, Kılıç F. "Salp sürü algoritması ile öznitelik seçimi ve sınıflandırıcı performans değerlendirmesi". *Avrupa Bilim ve Teknoloji Dergisi*, 30, 12-16, 2021.
[13] Hegazy AhE, Makhlouf MA, El-Tawelb GhS. "Improved salp swarm algorithm for feature selection". *Journal of King Saud University-Computer and Information Sciences*, 32(3), 335-344, 2020.
[14] Kamel SR, Yaghoubzadeh R. "Feature selection using grasshopper optimization algorithm in diagnosis of diabetes disease". *Informatics in Medicine Unlocked*, 26, 1-9, 2021.
[15] Lukmanto RB, Nugroho A, Akbar H. "Early detection of diabetes mellitus using feature selection and fuzzy support vector machine". *Procedia Computer Science*, 157, 46-54, 2019.
[16] García-Ordás MT, Benavides C, Benítez-Andrades JA, Alaiz-Moretón H, García-Rodríguez I. "Diabetes detection using deep learning techniques with oversampling and feature augmentation". *Computer Methods and Programs in Biomedicine*, 202, 1-11, 2021.
[17] Omisore OM, Ojokoh BA, Babalola AE, Igbe T, Folajimi Y, Nie Z, Wang L. "An affective learning-based system for diagnosis and personalized management of diabetes mellitus". *Future Generation Computer Systems*, 117, 273-290, 2021.
[18] Vaishali R, Sasikala R, Ramasubbareddy S, Remya S, Nalluri S. "Genetic algorithm based feature selection and MOE Fuzzy classification algorithm on Pima Indians Diabetes dataset". *International Conference on Computing Networking and Informatics (ICCNI)*, Lagos, Nigeria, 29-31 October 2017.
[19] Köse U, Güraksın GE, Deperlioğlu Ö. "Diabetes determination via vortex optimization algorithm based support vector machines". *Medical Technologies National Conference (TIPTEKNO)*, Bodrum, Turkey, 15-18 October 2015.
[20] Bhargava R, Dinesh J. "Deep learning based system design for diabetes prediction". *International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, Pune, India, 29-30 October 2021.
[21] Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM. "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems". *Advances in Engineering Software*, 114, 163-191, 2017.
[22] Karaboğa D. "An idea based on honey bee swarm for numerical optimization". Kayseri, Turkey, TR06, 2005.
[23] Zhao Y, He Y, Chen B, Xue X. "An improved Whale Swarm Algorithm with nonlinear weighting and convergence factor". *2nd International Conference on Safety Produce Informatization (IICSPI),* Chongqing, Çhina, 28-30 November 2019.
[24] Doğan C. Balina Optimizasyon Algoritması ve Gri Kurt Optimizasyonu Algoritmaları Kullanılarak Yeni Hibrit Optimizasyon Algoritmalarının Geliştirilmesi. Yüksek Lisans Tezi, Erciyes Üniversitesi, Kayseri, Türkiye, 2019.
[25] Ahmetoğlu H, Resul DAŞ. "Makine öğrenmesi yöntemleri kullanarak web uygulama saldırılarının tespitinde genetik öznitelik seçimi yaklaşımı". *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 14(2), 109-119, 2021.
[26] Moesinger L, Dorigo W, de Jeu R, van der Schalie R, Scanlon T, Teubner I, Forkel M. "The global long-term microwave vegetation optical depth climate archive (VODCA)". *Earth System Science Data*, 12(1), 177-196, 2020.
[27] Söyler H, Keskintürk T. "Karınca kolonisi algoritması ile gezen satıcı probleminin çözümü". *Türkiye Ekonometri ve İstatistik Kongresi,* Malatya, Türkiye, 24-25 Mayıs 2007.
[28] Hoos HH, Stützle T. *SATLIB: An online resource for research on SAT*. Editors: Maaren HV, Gent IP, Walsh T. SAT2000, 283-292, Amsterdam, Netherlands, IOS Press, 2000.
[29] Cortes C, Vapnik V. "Support-vector networks". *Mach Learn*, 20, 273-297, 1995.
[30] Ataseven B. "Yapay sinir ağları ile öngörü modellemesi". *Öneri Dergisi*, 10(39), 101-115, 2013.
[31] Al-Behadili HNK, Ku-Mahamud KR. "Fuzzy unordered rule using greedy hill climbing feature selection method: an application to diabetes classification". *Journal of Information and Communication Technology*, 20(3), 391-422, 2021.
[32] Özer İ. "Uzun kısa dönem bellek ağlarını kullanarak erken aşama diyabet tahmini". *Mühendislik Bilimleri ve Araştırmaları Dergisi,* 2(2), 50-57, 2020.
[33] Chaves L, Marques G. "Data mining techniques for early diagnosis of diabetes: a comparative study". *Applied Sciences*, 11(5), 1-12, 2021.