



## Yapay Sinir Ağları Kullanarak Kan Testi Sonuçlarının Sınıflandırılması ve Kullanıcı Ara Yüzünün Geliştirilmesi

Rıza İlhan<sup>1\*</sup>, Büşranur Gudar<sup>2</sup>

<sup>1\*</sup> Istanbul Aydın University, Faculty of Engineering, Department of Mechanical Engineering, İstanbul, Turkey, (ORCID: 0000-0001-8975-9942), [gilkhanisarkandi@aydin.edu.tr](mailto:gilkhanisarkandi@aydin.edu.tr)

<sup>2</sup> Istanbul Aydın University, Faculty of Engineering, Department of Mechatronics Engineering, İstanbul, Turkey, (ORCID: 0000-0003-0044-9011), [busranurgudar@stu.aydin.edu.tr](mailto:busranurgudar@stu.aydin.edu.tr)

(International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) 2021 – 21-23 October 2021)

(DOI: 10.31590/ejosat.1010484)

**ATIF/REFERENCE:** İlhan, R., Gudar, B. (2021). Yapay Sinir Ağları Kullanarak Kan Testi Sonuçlarının Sınıflandırılması ve Kullanıcı Ara Yüzünün Geliştirilmesi. *European Journal of Science and Technology*, (29), 1-5.

### Öz

Hastalıkların başarılı bir şekilde tedavi edilebilmesi için hızlı ve doğru tıbbi teşhis çok önemlidir. NHANES laboratuvarı analizleri ile elde edilen kan testi sonuçlarından hastalıkları tahmin etmek için MLP NN ve ChOA birleştirilerek bir model (Bileşik Model) geliştirilmiştir. Bileşik modelin geliştirilmesine ek olarak kan testi verisetlerinin model ile sınıflandırılarak sınıflandırma doğruluklarının diğer algoritmalar ile karşılaştırılmasını sağlayan bir kullanıcı arayüzü programı geliştirilmiştir. Öncelikle MLP Neural Network Algoritmasını, Chimp Optimization Algoritması ile birleştirerek MLP NN'ün en büyük dezavantajlarından olan aşırı yüklenme durumu giderilmiştir. Birleşik model'den elde edilen doğruluk değeri Rastgele Orman (Random Forest), Lojistik Regresyon (Logistic Regression) ve İki Katmanlı Yapay Sinir Ağı ile karşılaştırılmıştır. Sonuçlar, hazırlanan kullanıcı arayüzünde de görüldüğü gibi MLP NN-ChOA modelinin çoğu durumda diğer kıyaslama algoritmalarına kıyasla daha iyi veya karşılaştırılabilir bir performans sağladığını göstermektedir.

**Anahtar Kelimeler:** Kan testi, Yapay sinir ağları, Optimizasyon, Sınıflandırma, Teşhis.

## Classification of Blood Test Results Using Artificial Neural Networks and Improvement of User Interface

### Abstract

Rapid and accurate medical diagnoses are essential for the successful treatment of diseases. A model was developed by combining MLP NN and Chimp Optimization Algorithms to predict diseases from NHANES laboratory blood test results. In addition to the development of this model, a user interface program has been developed that allows datasets to be loaded and trained with the model and compared with other algorithms. First of all, by combining the MLP Neural Network Algorithm with the Chimp Optimization Algorithm, the overload situation, which is the biggest disadvantage of NN, is eliminated. The accuracy value obtained from the combined model was compared with Random Forest, Logistic Regression and Two-Layer Artificial Neural Network. The results show that the MLP NN-ChOA algorithm provides better or comparable performance compared to other benchmarking algorithms in most cases, as seen in the prepared user interface.

**Keywords:** Blood test, Neural networks, Optimization, Classification, Diagnosis.

\* Corresponding Author: Rıza İlhan İstanbul Aydın Üniversitesi, Mühendislik Fakültesi, Makina Mühendislik Bölümü, İstanbul, Türkiye, ORCID: 0000-0001-8975-9942, [gilkhanisarkandi@aydin.edu.tr](mailto:gilkhanisarkandi@aydin.edu.tr)

## 1. Introduction

Tıp bilimindeki en büyük sorun tanı koymaktır. Uzmanlar, hastalıklara teşhis koyabilmek için birçok farklı yöntemler geliştirmişler ve uygulamalar denemişlerdir (Maiellaro et al., 2005). Tıp alanındaki yapay zeka çalışmalarında literatür araştırmalarına örnek olarak, diyabet çalışmaları, akut hastalıkların risk tahmini ve sağlık kayıtlarına dayalı hibrit sinir ağı modellerinin geliştirilmesi verilmiştir. Örneğin, diyabet çalışmalarında insülin kavramının çok iyi anlaşılması gereklidir. Ancak tıp literatüründe insülinin yönetimiyle ilgili bilginin az olmasından dolayı doktorlar insülin rejiminin spesifikasyonu ve dozajı için kendi fikirlerini ve deneyimlerini kullanırlar. Örnek bir çalışmada insülin bilgilerinin doğru bir şekilde sınıflandırılması için yapay sinir ağı yaklaşımı kullanılmıştır (Gogou et al., 2001). Akut hastalık risklerinin tahminine yönelik çalışmalara örnek olarak Apandisit çalışmaları verilebilir. Akut apandisit için doğru ve hızlı teşhisi, hastalıktan ölüm oranını düşürmektedir. Literatür çalışmasında, akut apandisit zamanında teşhisi için MLP NN tabanlı sinir ağı algoritması kullanılmış olup diğer algoritmalara nazaran daha fazla duyarlılık, özgüllük ve doğruluk göstermiştir (Shahmoradi et al., 2018). Tıp alanındaki çalışmalarda teşhisin doğruluğu hastaların tedavisini de başarılı kılar. Bu nedenle hastalık türü ne kadar erken sınıflandırılırsa, tedavinin başarı şansı da o kadar yükselir. Genellikle, uzmanlar laboratuvar verilerinin güvenilir ve doğru analizinin eksikliğinden muzdariptir. Bu nedenle tanı koymada iyi ve doğru kararlar vermelerine yardımcı olacak araçlara ihtiyaçları vardır. Teşhisi doğru sınıflandırmak ve uzmanların hızlı ve doğru karar almaları için yapay zeka veya makine öğrenimi gereklidir (Maiellaro et al., 2005; Payandeh et al., 2009). Makine öğrenme algoritmaları denetimli ve denimsiz öğrenme olarak ikiye ayrılır (Mohammad Reza Mosavi et al., 2019). Denetimli öğrenmede öğrenme işlemi sağlayan yöntemler eğitmenler ile sağlanır. Eğitmenler, stokastik eğitmenler ve belirleyici eğitmenler olarak ikiye ayrılır. Deterministik eğitmenler, gradyan iniş ve geri yayılmaya dayalı yöntemlerdir. Ancak, yakınsama oranları basit ve hızlı yollarla elde edilmesine rağmen sonuçlar güvenilir değildir. Çünkü yerel optima(iyileştirme), öğrenmenin adım boyutlarına bağlıdır. Stokastik eğitmenlerse daha yavaştır, ancak sonuçlar daha güvenilirdir. Stokastik eğitmenlerde eğitim, başlangıç katsayısı rastgele olan bir parametre ile başlar. Bu rastgelelikle, hatadan maksimum kaçınma avantajı elde edilir (Afrakhteh et al., 2020; M. R. Mosavi & Khishe, 2017). Multi-Layer Perceptron Neural Network'te Stokastik Eğitmenler, öğrenme ve sınıflandırma(test edilme) süresi daha yavaş olmasına rağmen yerel optima'da takılmaz. Chimp Optimization Algoritması ise stokastik eğitmen temellidir. Ancak arama fazı içindeyken deterministik eğitmen davranışını seğileyerek yüksek doğruluk yakınsama oranına sahip olur (Stanford et al., 1994). Hem hızlı hemde yüksek doğruluk yakınsama oranına sahip olabilmek için bu iki model birleştirilmiştir. Ayrıca birleşik modelin diğer algoritmalar ile karşılaştırılabilmesi için bir kullanıcı arayüzü programı geliştirilmiştir. Çalışmada, Bölüm 2'de kullanıcı arayüzünde kullanılan tüm algoritmaların çalışma mantığı ve formülleri açıklanmıştır. Bölüm 3'de kullanıcı arayüzünün genel yapısı verilmiştir. Bölüm 4'te sonuçlar ve tartışmalar kısmına yer verilmiştir.

## 2. Algoritmalar

Bu bölümde kullanıcı arayüzünde tanımlanmış algoritmalar ve kullanılan formüller kısaca açıklanacaktır.

### 2.1. Şempanze Optimizasyon Algoritması (Chimp Optimization Algorithm - ChOA)

Şempanze optimizasyon algoritması, şempanzelerin grup bileşimini örnek alarak oluşturulmuş bir algoritmadır. Şempanzelerin grup bileşimi dinamik bir özelliğe sahiptir. Dinamik grup içerisinde, gruptaki üyeler zamana ve çevreye göre değişir (Khishe & Mosavi, 2020b). Bir şempanze kolonisinde 4 tip şempanze grubu bulunur. Bunlar; engelleyici, takipçi, sürücü ve saldırgan. Başarılı bir av(doğru sınıflandırma ve sonuç) için dört grubunda dahil edilmesi gereklidir. Sürücüler, avı hapsedmeden veya başka bir girişimde bulunmadan sadece avı izlerler. Engelleyiciler, avın kaçmasını önlemek için ağaç üzerinde konumlanır ve avlanma sürecinde onu takipçilere yönlendirir. Takipçiler, avı tuzağa düşürmek için avı kovalar ve saldırganı yönlendirir. Saldırgan, avın kaçış yolunu tahmin eder ve onu avın bölgelerine ve takipçilerine yönlendirir, ardından avı yakalar. ChOA için matematiksel formüller aşağıdaki gibi yazılabilir:

$$d = |c x_{prey}(t) - m x_{chimp}(t)| \quad (2.1.1)$$

$$x_{chimp}(t + 1) = x_{prey}(t) - ad \quad (2.1.2)$$

$$a = 2fr_1 - a, \quad c = 2r_2 \quad (2.1.3)$$

2.1.1 ve 2.1.2 formüllerinde yer alan t değeri iterasyonun güncel sayısını göstermektedir. a vektörü -1 ile 1 arasında rastgele sayılardan oluşan bir arama faktörüdür. c vektörü avcılığın kolay olması için atanan rastgele sayılardan oluşmuş bir vektördür.  $x_{prey}$ , avın pozisyonuna ait konum vektörü,  $x_{chimp}$  ise şempanzenin konum vektörünü göstermektedir. Burada f değeri bütün iterasyonlarda 2.5 tan 0'a nonlinear olarak azalmaktadır (Aljarah et al., 2018; Khishe & Mosavi, 2020b).

$r_1$  ve  $r_2$  değerleri ise 0 ile 1 aralığında rasgele sayılardan oluşturulmuş vektörlerdir. m değeri kaotik bir değerdir. Şempanzenin avı bulması için iyileştirme ve güncelleme görevi görür.

### 2.2. Çok Katmanlı Algılayıcı Sinir Ağları (Multi-Layer Perceptron Neural Network – MLP NN)

Çok Katmanlı Algılayıcı Sinir Ağı, ileri beslemeli, basit ve güvenilir bir Yapay Sinir Ağları(YSA) türüdür. MLP NN'un çıkış değerleri ikilidir. Bu ikili sonuçlar Evet/Hayır, 0/1, Doğru/Yanlış vb. anlamına gelmektedir. YSA'nın genel olarak en büyük dezavantajı aşırı uyum durumudur. İterasyonlar sırasında oluşan bu durumda belli bir iterasyondan sonra hata oranı tekrarlanmaktadır. Bu aşırı uyum, MLP NN' un gizli katmanı içerisinde bir optimizasyon algoritması kullanılarak önlenebilir. Literatürde, MLP NN genellikle Lojistik Regresyon, Destek Vektör Makineleri, Parçacık Sürü Optimizasyonu ve Karar Ağacı gibi sınıflandırma amaçları için kullanılan modellerden daha doğru performans sergilemektedir (Aefinfar et al., 2009; Khishe & Mosavi, 2020a; Shariati et al., 2019; Sinan, 2021).

$$(I_1)_j = b + (W_1)_{10}X_0 + (W_1)_{11}X_1 + (W_1)_{12}X_2 + \dots + (W_1)_{1n}X_n \quad (2.2.1)$$

Bu formülde  $W_1, X_1$  giriş öngerilim değerine rastgele atanan ağırlık değerini temsil eder ve  $j_i$ , giriş katmanından gizli katmana ait nöron sayısını gösterir. Formül 2.2.2' nin açılmış hali formül 2.2.1'de açıkça gösterilmiştir. Formül 2.2.2, gizli katman sonucunu göstermektedir.

$$(Out)_1 = f((I_1)_j) \quad (2.2.2)$$

Formül 2.2.2'nin sonucu, aktivasyon fonksiyonu tarafından belirlenen eşik değerine göre 0 veya 1 olarak sınıflandırılır. Ağırlıklı toplam değer; eşik değerine eşit veya büyük ise 1, eşik değerden küçük ise 0 yapılır. Problemin tipine ve ağırlık yapısına göre Lojistik, Unit Step, Sign, Linear, Piece-Wise, Hyperbolic Tangent, Rectified Linear Unit (RELU) yada Softplus fonksiyonları kullanılmaktadır.

### 2.3. Lojistik Regresyon (Logistic Regression - LR)

Lojistik Regresyon (Logistic Regression, Logit Regression, Log-Linear Classifier) modeli verileri sınıflandırma amacıyla kullanılmaktadır. Lojistik Regresyon 1 ve 0 olan doğruluk değerlerinin sınıflandırılmasında oldukça iyi bir sonuç vermektedir. Ancak sadece karmaşık olmayan veri setleri için yapılabilir. Veri setindeki çıktı değerleri kararsız, evet, hayır vs. gibi ikiden fazla durumu kapsadığında sınıflandırma doğruluğuna güvenilemez. Lojistik Regresyona Sigmoid fonksiyonu uygulanması halinde bağımsız değer tahmin doğruluğu daha da artmaktadır. Veri setindeki çıktı değerlerinin 1 veya 0 değerlerine ait olma olasılığının hesabı aşağıdaki formül 2.3.1'de gösterilmektedir (Sinan, 2021).

$$h_\theta(x) = P(y = 1|x) = 1 - P(y = 0|x) \quad (2.3.1)$$

Sigmoid fonksiyonu formül 2.3.2'de gösterilmiştir (Sinan, 2021).

$$\sigma(x) = \text{Sigmoid Fonksiyonu} = \frac{1}{1+e^{-x}} \quad (2.3.2)$$

### 2.4. Rastgele Orman (Random Forest - RF)

Rastgele orman algoritması, böl ve fethet yaklaşımına sahip özel bir topluluk yaklaşımıdır (Breiman, 2001). Rastgele orman algoritması birçok karar verme ağacı veya regresyonu ile oluşturulur. Algoritmanın en büyük avantajı, herbir ağaç veya regresyon zayıf bir öğrenme özelliğine sahip olduğu halde ağaçların tümü algoritmayı daha güçlü öğrenme özelliğine sahip kılar. Ayrıca herhangi bir parametre ayarı olmaksızın hem eğitim hemde tahminde çok iyi bir performans gösterir. En büyük dezavantajı, kayıp yada düzensiz verilere sahip veri setlerinde yada çok fazla veriye sahip veri setlerinde tahmin doğruluğu ve eğitim performansı düşmektedir. Ancak Destek Vektör Makinesi, Karar Ağaçları, Naive Bayes Algoritmasına kıyasla daha yüksek tahmin doğruluğu elde edilmektedir (Gunçar et al., 2018).

### 2.5. Şempanze Optimizasyon Algoritması ile Eğitilmiş Çok Katmanlı Aygılayıcı Sinir Ağı (MLP NN + ChOA)

Bu bölümde, ChOA kullanılarak MLP Sinir Ağı'nın eğitimi için önerilen metodoloji açıklanmış ve kullanılan formüller

eklenmiştir. Çalışmada, ChOA, MLP NN algoritmasının bir adet gizli katmanına yerleştirilmiştir.

Yapay sinir ağlarının en önemli dezavantajı uygun ağ yapısının belirlenememesi, sadece sayısal bilgilerle çalışması ve ağırlık işlem süresinin bilinmemesidir. Sinir ağlarının yapısını belirlemek için belirli bir kural yoktur. Bu nedenle de uygun ağ yapısı, deneyim ve deneme yanılma yoluyla elde edilir. Örnekteki hatanın belirli bir değere indirilmesi eğitimin tamamlandığı anlamına gelir. Ancak bu değer bize optimum sonuçları vermemektedir. Ayrıca algoritma belirli bir kurala tabi değilse işlem süresi de öngörülemez sorunlara neden olabilir. Örneğin, çıktının belirli sayıda yinelemesinden sonra aşırı uyum (overfitting) en önemli öngörülemez sonuçlardandır (Demirci et al., 2016).

Öncelikle Bölüm 2.1'de kullanılmış formüller herbir şempanze için özelleştirilmiştir. Özelleştirilmiş formüller aşağıdaki gibidir.

$$d_{Attacker} = |c_1x_{Attacker} - m_1x| \quad (2.5.1)$$

$$d_{Barrier} = |c_2x_{Barrier} - m_2x| \quad (2.5.2)$$

$$d_{Chaser} = |c_3x_{Chaser} - m_3x| \quad (2.5.3)$$

$$d_{Driver} = |c_4x_{Driver} - m_4x| \quad (2.5.4)$$

$$x_1 = x_{Attacker} - a_1 d_{Attacker} \quad (2.5.5)$$

$$x_2 = x_{Barrier} - a_2 d_{Barrier} \quad (2.5.6)$$

$$x_3 = x_{Chaser} - a_3 d_{Chaser} \quad (2.5.7)$$

$$x_4 = x_{Driver} - a_4 d_{Driver} \quad (2.5.8)$$

$$x = \frac{x_1+x_2+x_3+x_4}{4} \quad (2.5.9)$$

$d_{Attacker}, d_{Barrier}, d_{Chaser}$  ve  $d_{Driver}$  formülleri ile av ile şempanzeler arasındaki mesafe hesaplanır. Çıkan sonuçlara göre de şempanzelerin ava göre konumları hesaplanıp ortalaması alınır. 2.5.1 - 2.5.8 formülleri kullanılarak MLP NN gizli katmanındaki nöronların  $x$  değerleri bulunur. Ancak  $x$  ( $x_{Attacker}, x_{Barrier}, x_{Driver}$  and  $x_{Chaser}$ ) veri kümesindeki örneklerin her bir değeri için hesaplandığından tek bir değere dönüştürülmesi gerekir. Tek bir değere dönüştürmek için formül 2.5.9 kullanılır. Ortak  $x_{chimp}$  değeri, MLP NN - ChOA'nın gizli katmanının çıktısıdır ve aslında MLP NN gizli katmanının ağırlık değeridir.  $x_{chimp}$ , out olarak formül 2.2.1'e yerleştirilir ve ardından MSE değeri hesaplanır. Gizli katmandan elde edilen çıkış değeri Lojistik aktivasyon formülü ile hesaplanır. Aşağıda formül 2.5.10'da gösterilmiştir.

$$f = \frac{1}{1+e^{-Out}} \quad (2.5.10)$$

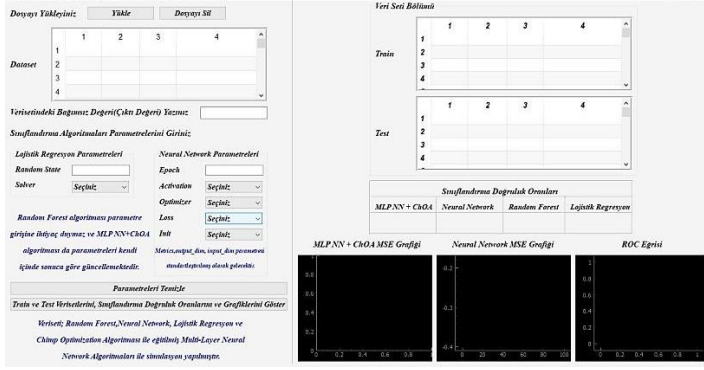
Ortalama Kare Hatası (MSE), tüm eğitim örnekleri için oluşturulan arama araçları (MLP NN'ler) tarafından istenen ve değerlendirilen değerler arasındaki farkını hesaplamak için kullanılır.

$$MSE = 1/m \sum_{i=1}^m (f - f')^2 \quad (2.5.11)$$

Formül 2.5.11'de  $f$  istenen değer,  $f'$  ise değerlendirilen değer ve  $m$  eğitim veri kümesindeki örnek sayısıdır.

### 3. Kullanıcı Arayüzü

Geliştirilen Kullanıcı arayüzü programı ile pasif veri setleri algoritmalarla modellenerek algoritmaların sınıflandırma doğruluğu, ROC eğrisi ve ChOA ile MLP NN birleştirilerek elde edilen birleşik modele ait Mean Squared Error (MSE) – Ortalama Kare Hatası – grafiği görüntülenmektedir. Kullanıcı bu sonuçlara göre aktif veri setlerini seçtiği algoritma ile hastalıkları sınıflandırabilir. Şekil 1’de boş Kullanıcı arayüzü ekranı verilmiştir.



Şekil 1: Kullanıcı arayüzü programına ait boş arayüz ekranı

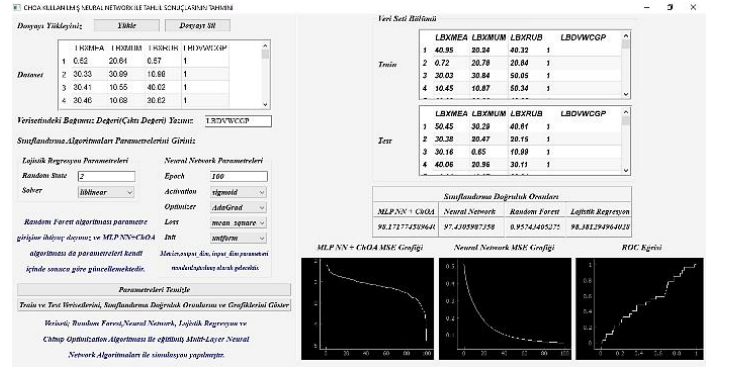
Kullanıcı arayüzü ekranında, veriseti yüklendiğinde program içinde gömülen “Dataset” olarak etiketlenen excel tablosunda gösterilecektir. Kullanıcı, veri seti içindeki sonuç değerlerini/değerini “Bağımsız değer”e yazmalıdır. RF algoritması herhangi bir parametre ayarına gerek duymaz. MLP NN + ChOA birleşik (ensemble) algoritması ise yüklenen verisetine uygun olarak parametrelerini kendisi günceller. Bu nedenle hem RF hemde birleşik algoritma için kullanıcıdan herhangi bir parametre ayarı istenmemiştir. Lojistik Regresyonda en önemli iki parametre solver ve random state’tir. Solver, çözümleyici fonksiyonlar anlamına gelmektedir ve kullanıcı arayüzü ekranında en sık kullanılan 5 fonksiyon çeşidi seçilmektedir: Liblinear, Sag, Saga, Newton-cg, Lbfgs. Random state ise olası durumlar anlamına gelmektedir. Kullanıcı istediği değeri verebilir. MLP NN’te girilmesi gereken 5 önemli parametre ayarı vardır: Epoch, Activation, Optimizer, Loss ve Init’tir. Epoch, iterasyon sayısıdır. Activation, gizli katmandan çıkan sonuca uygulanan fonksiyon çeşitleridir. Arayüz için Relu, Sigmoid, Tanh, Gaussian ve Softplus fonksiyonları seçilebilmektedir. Optimizer parametresi ile en iyileme algoritmasının belirlenir. En iyi 3 parametre seçilebilmektedir: Adam, AdaGrad, RMSprop. Loss parametresi ile hata fonksiyonları tanımlanır ve Binary crossentropy, Mean absolute error, Mean squared error ve Categorical hinge olarak 4 tip hata fonksiyonu çeşidi seçilebilir. Init parametresi ile başlangıç ağırlık değerlerinin belirleyen seçenekler tanımlanır ve arayüzden Uniform, Lecun uniform, Identity ve Orthogonal olarak 4 çeşit ağırlık belirleme seçeneği seçilebilir. Metrics parametresi bu çalışma için varsayılan olarak doğruluk değeri (accuracy) olarak tanımlanmıştır.

Bu parametreler girildiğinde eğitim (train), test veri setleri, algoritmaların doğruluk oranları ve grafikler gösterilecektir. Train ve Test verisetlerinin algoritmalarda kullanılma yüzdeliği önemlidir. Eğer algoritmanın öğrenme oranının yüksek olmasını istiyorsak bu durumda Train verisetinin algoritmadaki yüzdeliği en az %20 ve hatta daha düşük bir yüzdeliğe sahip olmalıdır. Sınıflandırmaya önem verilmesi durumunda ise Train verisetinin

yüzdeliği %60 olabilir. Ancak bu çalışmada sınıflandırmaya önem verildiği için Train veriseti için %67, Test veriseti için %33 olması uygun görülmüştür. Bu değerler standart olarak belirlenmiştir.

### 4. Tartışma ve Sonuçlar

Bu çalışmadaki amacımız, MLP NN modelimizde gizli katmana ChOA ekleyerek oluşturduğumuz yeni birleşik algoritmayı hazırladığımız Kullanıcı Arayüzü programında göstererek hata değerinin azaltılarak tahmin doğruluğunu arttırdığını göstermektir. Böylece MLP NN’nin en büyük dezavantajı, yani belirli bir iterasyon veya aşırı yüklenme durumunda oluşan hata oranının düşmemesi sorunu çözülmüştür. Sınıflandırma oranı ise tahmin doğruluğu hakkında bilgi verir. Hata grafiği ve sınıflandırma doğruluğunun gösterilmesi kullanıcıya hangi algoritmanın hangi veri setine uygun olarak kullanılacağına rehberlik eder ve bu algoritmalar aktif verisetlerine uygulanarak hastalık tahmini daha kolay ve hızlı yapılabilir.



Şekil 2: NHANES’ten alınmış kan tahlili sonuçlarına göre Kızamık, Kabakulak, Suçiçeği ve Kızamıkçık hastalıklarının hastalardaki antikorlara göre sonuçlarını içeren veri setinin kullanıcı arayüzü programında modellenmesi

Kızamık, Kabakulak, Suçiçeği ve Kızamıkçık hastalıklarının kan testi sonuçlarına göre hazırlanmış herhangi bir pasif veri setine ait tahmin doğruluğunu, kullanıcının programda belirlediği 4 çeşit algoritma ile gösterilmiştir. Bu karşılaştırma sonuçları Şekil 2’de de yer aldığı gibi Kullanıcı Arayüzünde gösterilmektedir. Modellerden elde edilen tahmin doğruluğu sonuçları; MLP NN için %97,43, RF için %95,92, LR için %98,38 ve ChOA-MLP NN algoritması için %98,17’dir. Bu değerler programa yüklenen verisetine göre değişkenlik göstermektedir. %98,38 değeri tek başına LR algoritmasının doğruluğundan daha küçüktür, ancak LR’nin karmaşık verisetinde uygulanmasından dolayı daha güvenilirdir. LR algoritmasına ait olan ROC eğrisinden de görüldüğü gibi sonuçlar exponensiyel yükselmemesi sınıflandırma doğruluğuna güvenilmemesi gerektiğinin bir göstergesidir.

Ayrıca MLP NN Algoritmasının en büyük dezavantajı olan aşırı uyum durumu Chimp optimizasyon algoritması kullanılarak Şekil 2’de yer alan MLP NN + ChOA grafiğinden de görüldüğü gibi MSE oranı sürekli azalarak ortadan kaldırılmıştır. Sonuç olarak, MLP NN - ChOA’nın sınıflandırma oranı diğer bölümlerde açıklanan algoritmaların dezavantajı olduğu durumları dikkate alındığında çok daha güvenilirdir ve daha doğru sonuçlar vermektedir.

### Teşekkür

İstanbul Aydın Üniversitesi Yazılım Mühendisliği Bölümü'nden Dr. Sedjad Eyni ve Dr. Elnaz Pashaei'ye faydalı yorumları ve tartışmaları için teşekkür ederiz.

## **Kaynaklar**

- Aeinfar, V., Mazdarani, H., Deregeh, F., Hayati, M., & Payandeh, M. (2009). Multilayer perceptron neural network with supervised training method for diagnosis and predicting blood disorder and cancer. *IEEE International Symposium on Industrial Electronics*, 2075–2080. <https://doi.org/10.1109/ISIE.2009.5213591>
- Afrakhteh, S., Mosavi, M. R., Khishe, M., & Ayatollahi, A. (2020). Accurate classification of EEG signals using neural networks trained by hybrid population-physic-based algorithm. *International Journal of Automation and Computing*, 17(1), 108–122. <https://doi.org/10.1007/s11633-018-1158-3>
- Aljarah, I., Faris, H., & Mirjalili, S. (2018). Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Computing*, 22(1). <https://doi.org/10.1007/s00500-016-2442-1>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Demirci, F., Akan, P., Kume, T., Sisman, A. R., Erbayraktar, Z., & Sevinc, S. (2016). Artificial neural network approach in laboratory test reporting: Learning algorithms. *American Journal of Clinical Pathology*, 146(2), 227–237. <https://doi.org/10.1093/ajcp/aqw104>
- Gogou, G., Maglaveras, N., Ambrosiadou, B. V., Goulis, D., & Pappas, C. (2001). A neural network approach in diabetes management by insulin administration. In *Journal of Medical Systems* (Vol. 25, Issue 2, pp. 119–131). <https://doi.org/10.1023/A:1005672631019>
- Gunčar, G., Kukar, M., Notar, M., Brvar, M., Černelč, P., Notar, M., & Notar, M. (2018). An application of machine learning to haematological diagnosis. *Scientific Reports*, 8(1). <https://doi.org/10.1038/s41598-017-18564-8>
- Khishe, M., & Mosavi, M. R. (2020a). Classification of underwater acoustical dataset using neural network trained by chimp optimization algorithm. *Applied Acoustics*, 157. <https://doi.org/10.1016/j.apacoust.2019.107005>
- Khishe, M., & Mosavi, M. R. (2020b). Chimp optimization algorithm. *Expert Systems with Applications*, 149. <https://doi.org/10.1016/j.eswa.2020.113338>
- Maiellaro, P., Cozzolongo, R., & Marino, P. (2005). Artificial neural networks for the prediction of response to interferon plus ribavirin treatment in patients with chronic hepatitis c. *Current Pharmaceutical Design*, 10(17), 2101–2109. <https://doi.org/10.2174/1381612043384240>
- Mosavi, M. R., & Khishe, M. (2017). Training a feed-forward neural network using particle swarm optimizer with autonomous Groups for sonar target classification. *Journal of Circuits, Systems and Computers*, 26(11). <https://doi.org/10.1142/S0218126617501857>
- Mosavi, Mohammad Reza, Khishe, M., Naseri, M. J., Parvizi, G. R., & Ayat, M. (2019). Multi-layer perceptron neural network utilizing adaptive best-mass gravitational search algorithm to classify sonar dataset. *Archives of Acoustics*, 44(1), 137–151. <https://doi.org/10.24425/AOA.2019.126360>
- Payandeh, M., Aeinfar, M., Aeinfar, V., & Hayati, M. (2009). A new method for diagnosis and predicting blood disorder and cancer using artificial intelligence (Artificial neural networks). *International Journal of Hematology-Oncology and Stem Cell Research*, 3(4), 25–33.
- Shahmoradi, L., Safdari, R., Mirhosseini, M. M., Arji, G., Jannat, B., & Abdar, M. (2018). Predicting risk of acute appendicitis: A comparison of artificial neural network and logistic regression models. *Acta Medica Iranica*, 56(12), 784–795.
- Shariati, M., Mafipour, M. S., Mehrabi, P., Bahadori, A., Zandi, Y., Salih, M. N. A., Nguyen, H., Dou, J., Song, X., & Poingnian, S. (2019). Application of a hybrid artificial neural network-particle swarm optimization (ANN-PSO) model in behavior prediction of channel shear connectors embedded in normal and high-strength concrete. *Applied Sciences (Switzerland)*, 9(24). <https://doi.org/10.3390/app9245534>
- Sinan, U. (2021). Makine öğrenmesi teorik yönleri ve python uygulamaları ile bir yapay zeka ekolü (Atalay Mat). Nobel Akademik Yayıncılık.
- Stanford, C. B., Goodall, J., Wallis, J., Mpongo, E., Wallis, J., & Goodall, J. (1994). Hunting decisions in wild chimpanzees. *Behaviour*, 131(1–2), 1–18. <https://doi.org/10.1163/156853994X00181>