T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ İLE KAN TAHLİL SONUÇLARININ SINIFLANDIRILMASI

YÜKSEK LİSANS TEZİ

Büşranur GÜDAR

Mekatronik Mühendisliği Ana Bilim Dalı

Mekatronik Mühendisliği Programı

EYLÜL, 2021

T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ İLE KAN TAHLİL SONUÇLARININ SINIFLANDIRILMASI

YÜKSEK LİSANS TEZİ

Büşranur GÜDAR

(Y1913.110001)

Mekatronik Mühendisliği Ana Bilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı : Dr Öğr. Üyesi Rıza İLHAN

EYLÜL, 2021

ONAYFORMU

ONUR SÖZÜ

Yüksek lisans tezi olarak sunduğum " Makine Öğrenmesi İle Kan Tahlil Sonuçlarının Sınıflandırılması" adlı çalışmanın, tezin proje safhasından sonuçlanmasına kadarki bütün süreçlerde bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurulmaksınızın yazıldığını ve yararlandığım eserlerin Kaynakça ‘da gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve onurumla beyan ederim. (17 / 09 / 2021)

Büşranur GÜDAR

ÖNSÖZ

Tez çalışması sırasında bilgi ve deneyimlerinden yararlandığım ve desteklerini hiçbir zaman onaylamadığım için kendimi çok şanslı hissediyorum, bu sayede birçok yeni bilgi öğrendim ve böyle değerli bir akademisyenle çalışma fırsatı buldum. Danışman Hocam Sayın Dr. Öğretim Üyesi Rıza İlhan'a çok değerli yardımları ve destekleri için en içten teşekkürlerimi sunarım.

Değerli hocalarıma, jüri üyelerime teşekkür ederim.

Çalışmalarım boyunca her zaman yanımda olan ve bana hem maddi hemde manevi destekleriyle yardımcı olan annem Nurcan Bayrakdar'a teşekkür ederim.

Eylül 2021 Büşranur GÜDAR

MAKİNE ÖĞRENMESİ İLE KAN TAHLİL SONUÇLARININ SINIFLANDIRILMASI

ÖZET

Hastalıkların tez ve başarılı bir şekilde tedavi edilebilmesi için öncelikle teşhislerin hızlı ve doğru olması gereklidir. Hastalıklar ve tıbbi teşhisler alanındaki en son ilerleme de ise teşhisler için makine öğrenmesini kullanarak sınıflandırılması üzerinedir.

Makine öğrenmesi ile teşhis tekniğinin kullanılabilmesi için öncelikle deney sonuçlarıyla elde edilen veri setlerinin doğru bir şekilde sınıflandırılması gerekmektedir. Bu tez çalışmasında, teşhislerin doğru bir şekilde sınıflandırılması için iki algoritma birleştirilerek yeni bir algoritma (bileşik) türetilmiştir. Önerilen bileşik algoritmanın hem Çok Katmanlı Algılayıcı Sinir Ağı (MLP NN) hem de Şempanze Optimizasyon Algoritmasının (ChOA) sahip olduğu dezavantajları ortadan kaldıracağı ve sonuçların sınıflandırılmasının daha doğru ve güvenilir hale getirmesi beklenmektedir. Bu tez çalışmasında önerilen algoritma, her ne kadar kan testi sonuçlarının sınıflandırılmasına yönelik kullanılsada kan testi veri setlerine benzer diğer veri setlerinin sınıflandırılmasında da kullanılabilir. Tez çalışmasında sınıflandırılma sonuçlarının ve diğer algoritmalarla karşılaştırılması amacıyla örnek olarak Kabakulak hastalığına ait veri setine MLP NN – ChOA uygulanmıştır. Kabakulak hastalığına ait veri setide kan testi sonucundan elde edilmektedir. Birleşik model ile elde edilen doğruluk değeri, MLP NN, Lojistik Regresyon (LR), Destek Vektör Makinesi (SVM), Rastgele Orman (RF) gibi diğer sınıflandırma amacıyla kullanılan algoritmalarla da karşılaştırılmıştır.

Sonuçlar, hazırlanan kullanıcı arayüzünde de görüldüğü gibi her ne kadar verisetine göre doğruluk oranı değişkenlik gösterebilse de MLP NN-ChOA algoritmasının çoğu durumda diğer kıyaslama algoritmalarına kıyasla karşılaştırılabilir bir iyi performans sağladığını göstermektedir.

Anahtar Kelimeler: Sinir ağları, Makine öğrenmesi, Optimizasyon, Kan testi.

CLASSIFICATION OF BLOOD ANALYSIS RESULTS WITH MACHINE LEARNING

ABSTRACT

In order for the diseases to be treated quickly and successfully, the diagnoses must be fast and accurate. The most recent progress in diseases and medical diagnoses is their classification using machine learning for diagnoses.

In order to use the diagnostic technique with machine learning, first of all, the data sets obtained from the experimental results must be classified correctly. In this thesis, a new algorithm (composite) is derived by combining two algorithms for the correct classification of diagnoses. It is expected that the proposed composite algorithm will eliminate the disadvantages of both Multilayer Perceptron Neural Network (MLP NN) and Chimpanzee Optimization Algorithm (ChOA) and make the classification of results more accurate and reliable. Although the algorithm proposed in this thesis is used to classify blood test results, it can also be used to classify other data sets similar to blood test data sets. In order to compare the classification results and other algorithms in the thesis study, MLP NN – ChOA was applied to the data set of Mumps disease as an example. The data set for mumps disease is obtained from the blood test result. The accuracy value obtained with the combined model was also compared with other algorithms used for classification purposes such as MLP NN, Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF).

The results show that although the accuracy rate may vary according to the dataset, as seen in the prepared user interface, the MLP NN-ChOA algorithm provides a comparable good performance compared to other benchmarking algorithms in most cases.

Key words: Neural network, Machine learning, Optimization, Blood test,

İÇİNDEKİLER

Sayfa

KISALTMALAR

ChOA : Şempanze Optimizasyon Algoritması

DR : Doğrusal Regresyon

DVM : Destek Vektör Makineleri

LR : Lojistik Regresyon

MLP NN : Çok Katmanlı Algılayıcı Sinir Ağları

MSE : Ortalama Kare Hatası

RF : Random Forest

SVM : Destek Vektör Makinesi

ÇİZELGE LİSTESİ

Sayfa

ŞEKİLLER LİSTESİ

Sayfa

I. GİRİŞ

Tıp alanındaki en büyük zorluk tanı koymaktır. Uzmanlar, hastalıklara tanı koyabilmek için birçok farklı yöntemler geliştirmişler ve uygulamalar denemişlerdir (Maiellaro et al., 2005).

Yapay zeka çalışmaları tıp alanındaki en sık çalışması yapılan örnek olarak, diyabet tanısı ve riski ile ilgili çalışmalar verilebilir. Örnek bir diyabet çalışmasında, bu akut hastalığın olası risk tahmini, teşhisi ve bilimsel sağlık kayıtlarına dayalı hibrit sinir ağı modellerinin geliştirilmiştir. Diyabet çalışmalarında tanı koyabilmek, sınıflandrabilmek, risk analizi yapabilme vb. gibi çalışmalar için öncelikle insülin kavramı çok iyi kavranılması gereklidir. Ancak literatürde insülin ile ilgili bilgi ve araştırma azdır. Çalışmaların az olmasından dolayı uzmanlar insülin kavramının anlaşılması, özelleştirilmesi, analizi ve dozajı için kendi fikirlerini ve deneyimlerini benimser ve kullanırlar. Örnek bu çalışmada insülin bilgilerinin doğru bir şekilde sınıflandırılması ve risk analizi yapılabilmesi için yapay sinir ağı modeli kullanılmıştır (Gogou et al., 2001).

Yine akut hastalıkların risklerine yönelik tahmin çalışmalarına başka bir örnek olarak Apandisit çalışmalarıdır. Akut apandisitin hızlı ve doğru teşhisi, ölüm riskinin fazla olmasından dolayı oldukça önemlidir. Genel olarak doğru ve hızlı teşhis, tüm ölümcül hastalıklar için hastalıktan ölüm oranını düşürmektedir. Akut apandisitin zamanında teşhisi ile ilgili bu örnek çalışmada, MLP NN tabanlı sinir ağı algoritması kullanılmıştır. MLP NN algoritması diğer algoritmalara nazaran daha fazla duyarlılık, özgüllük ve doğruluk gösterdiği saptanmıştır (Shahmoradi et al., 2018).

Tüm literature çalışmalarında, hastalıkların teşhisi ne kadar erken sınıflandırılırsa tedavinin başarı şansı da o kadar çabuk arttığı ortak olarak bildirilmiştir. Genellikle, tıp uzmanları laboratuar verilerinin hızlı, güvenilir ve doğru analizinin eksikliğinden yakınmaktadır. Bu nedenle tanının koyulması için hızlı, iyi ve doğru kararlar vermelerine yardımcı olacak araçlara gereksinim vardır. Teşhislerin hızlı, güvenilir ve doğru sınıflandırılması ve uzmanların da bu

sınıflandırılmış verileri değerlendirmesi ile hızlı ve doğru karar almaları için yapay zeka veya makine öğrenimi gereklidir (Gogou et al., 2001; Payandeh et al., 2009). Literatür araştırmalarının sonucunda modelin temel algoritmasının MLP NN olmasına karar verilmiştir. Makine öğrenme algoritmaları ikiye ayrılır: denetimli ve denetimsiz öğrenme (Khishe & Mosavi, 2020a). Denetimli öğrenme, öğrenme işlemini sağlayan yöntemler eğitmenler ile sağlanır. Eğitmenler ikiye ayrılır: stokastik eğitmenler ve belirleyici eğitmenler.

Deterministik eğitmenler, gradyan iniş ve geri yayılım temellidir. Ancak, basit ve hızlı yakınsama oranlarına sahip olsalarda sonuçlara güvenilemez. Güvenilememesinin temel nedeni yerel optima(iyileştirme)dır yani öğrenme işlemleri sırasında oluşturulan adım boyutlarına bağlıdır. Stokastik eğitmenlere sahip bir modelin öğrenme süresi ise daha yavaştır, ancak sonuçlar daha doğru ve güvenilirdir.

Öğrenme için eğitmen dışında diğer önemli kavram ise rastgeleliktir. Rastgelelikle, hatadan maksimum kaçınma avantajı elde edilir (Afrakhteh et al., 2020; Mosavi et al., 2019).

MLP NN' te stokastik eğitmenlere sahiptir. Bu nedenle daha yavaştır. Ancak yerel optima'da takılmaz. MLP NN ile birleştirilmesi istenen ChOA ise stokastik eğitmen temellidir. Ancak ChOA'nın arama ajanları öğrenme için arama fazı içesindeyken deterministik eğitmen davranışını sergiler. Bu nedenlede yüksek doğruluk yakınsama oranına sahiptir (Stanford et al., 1994).

Yapay zekanın tıptaki uygulamaları içinde en sık kullanım alanları genom dizilimi veya DNA gen ekspresyonu mikrodizileri, gen ağlarının modellenmesi, gen ekspresyonu il ilgili verilerin analizi ve kümelenmesi, DNA ve proteinlerde örüntü tanıma, protein yapısı tahmini gibi uygulamalar yer alır. Hematoloji alanında yapay zeka uygulamaları ilk olarak rutin olarak kullanılan cihazların laboratuvar veri yönetiminde ve verilerin sınıflandırılmasında kullanılmıştır. Yapay zekanın hematolojik alanında kullanılmış yeni cihazlardaysa, periferik kan analizinden elde edilen verilerle eğitilmiş sinir ağlarına dayanan anemi, talasemi ve lösemi gibi belirli hastalıklarda ayırıcı tanı ile ilgilidir. Yapay zekanın kullanıldığı hematolojik çalışmalara örnek olarak hematolojik malignitelerin teşhisi, kanser teşhisi, moleküler teşhis içinse ilk mikroarray tabanlı ve

biyoinformatik yaklaşımın tanıtılması dahil olmak üzere bir çok çalışma yapılmıştır. Örnek olan bir diğer çalışmada ise DNA mikroarray kullanarak binlerce genin eş zamanlı ifadesinin, ilk biyolojik özelliklerden bağımsız olarak izlenmesine dayanan sistematik bir yaklaşım geliştirilmiştir. Hesaplamalı yapay zeka yöntemleri, parametrik olmayan veri modelleri sağlar ve yeni verileri önceden tanımlanmış kategorilere göre sınıflandırmaya izin vererek tanı ve prognozu destekler. Ayrıca yapay zeka ile yeni kategoriler araştırılır, mantıksal kurallar oluşturulur ve verileri anlamaya ve çok boyutlu ilişkileri görselleştirmeyi sağlarlar (Zini, 2005).

Neural network çalışmaları örnek olarak diğer bir çalışma ise derin evrişimli sinir ağı kullanılarak trikrom boyalı dışkı örneklerinde bağırsak protozoalarının tespitidir. Yapay zeka ve dijital slayt tarama, bir konvolüsyonel sinir ağı (CNN) modeli kullanarak parazitlerin tespitini ve slayt yorumlamasını artırarak klinik parazitoloji laboratuvarında önemli bir çalışma olarak yer almaktadır. Bu çalışmanın amacı ise, manuel doğrulama için potansiyel parazitleri işaretlerken negatif trikrom slaytları tarayabilen hassas bir model geliştirmektir. Öncelikle geleneksel protozoalar, derin bir CNN'de “sınıflar” olarak eğitilmiş, daha sonrada veri etiketleme yapılarak arayüzü geliştirilmiştir. Bu çalışma modelin ve tarayıcının seri olarak seyreltilmiş dışkı kullanılarak tespit limiti, 4 benzersiz slayt seti kullanılarak birden fazla parazitolog tarafından yapılan manuel incelemelerden 5 kat daha hassas olduğu belirtilmiş ve çıkan sonuçlara göre dijital slayt tarama ve bir CNN modelinin, bağırsak protozoalarının geleneksel tespitini artırmak için sağlam araçlar olduğunu kanıtlanmıştır (Mathison et al., 2020).

Destek vektör makinesi ile yapılan bir diğer çalışmada ise ABD içerisinde diyabetli ve prediyabetli kişileri tespit etme yöntemi çalışmasıdır. İki sınıflandırma şeması için SVM modellerini geliştirmek ve doğrulamak için 1999-2004 yılında National Health and Nutrition Examination Survey (NHANES) tarafından elde edilen verileri kullanılmıştır. Sınıflandırma Şeması 1, tanısı konulmuş veya teşhis edilmemiş diyabet ile prediyabet veya diyabetsiz temsil etmektedir. Sınıflandırma Şeması 2 ise tanısı konmamış diyabet veya prediyabet durumları temsil etmektedir. SVM modelleri, bireylerin bu diyabet kategorilerine göre en iyi şekilde sınıflandırılmasını sağlayacak değişken kümelerini seçmek

için kullanılmış olup popülasyondaki diyabet ve prediyabet gibi yaygın hastalıkları olan kişileri tespit etmek için umut verici bir sınıflandırma yaklaşım olduğu sonucu çıkartılmıştır (Yu et al., 2010).

Güncel hastalıkların teşhis edilmesinde de iyi bir araç olduğunu kanıtlayan makine öğrenmesi çalışmalarına bir diğer örnek ise SARS-CoV-2 hastalığının pozitif olduğu bireylerin belirlenmesi için yapılmış çalışmadır. Hastalığın tespiti için RT-PCR ile viral RNA tespiti gereklidir. Çalışmada rutin laboratuvar testleri ile genellikle 1-2 saat içinde bir sonuçlara ait bir geri dönüş süresi (TAT) kolayca elde edilebilir. Bireyin SARS-CoV-2 enfeksiyon durumunu tahmin etmek için 27 rutin laboratuvar testinden alınan sonuçlara göre hastanın demografik özelliklerini (yaş, cinsiyet, ırk) birleştiren bir makine öğrenimi modeli geliştirilmiştir. Çalışmada destek vektör makineleri kullanılarak AUC eğrisi çizdirilmiştir. Sonuç olarak destek vektör makinesinin kullanılmasıyla 0,838 gibi bir AUC sonucu ile sonuçlanmıştır (Yang et al., 2020).

Elektronik sağlık kayıtları gibi büyük, heterojen veri setleri ile çoklu değişkenler, değişkenlerin etkileşimleri ve zamanı işleyebilen ve bireysel düzeyde kişinin klinik risk tahminini ölçen yöntemler geliştirilmiştir. Örnek bir diğer çalışmada sol ventriküler yapısal (LV) ani kardiyak ölüm (SCD) kayıt defterindeki ani kardiyak arresti (SCA) tahmin ederek hayatta kalma verileri için standart rastgele orman yöntemleri kullanılmıştır. Bu yöntem ile hayatta kalma verileri için standart rastgele orman yöntemleri üstün bir performans göstermiştir (Wongvibulsin et al., 2019).

Kan tahlili ile hastalık veya risk tahmini veya teşhisine en sık olarak yapılan çalışmalara örnek olarak kardiyovasküler hastalıklar için risk tahmini modeli uygulamaları verilebilir. Yine 2011 ve 2018 yılları arasında Xi'an Tıp Üniversitesi'nde 498 denek içeren retrospektif bir çalışma ile elde edilen verilerle rastgele orman algoritması kullanılarak kardiyovasküler hastalık riskini rahmin eden bir model geliştirilmiştir. Ayrıca birçok öenmli değişkeni lojistik regresyon ile analiz etmişlerdir. Daha sonra elde edilen sonuçlarla lojistik regresyon ve rastgele orman algoritması AUC eğrisinden elde edilen çıkarımlarla karşılaştırılmıştır.Çıkan sonuçlara göre çalışmadan kardiyovasküler risk tahmini için rastgele orman algoritmasının kullanılmasının daha uygun olduğu sonucu çıkartılmıştır (Su et al., 2020).

Kan tahili ile elde edilen parametrelerde WBC değeri beyaz kan hücrelerinin oranını göstermektedir. Beyaz kan hücreleri (WBC'ler), bağışıklık sisteminde enfeksiyonlara karşı koruma sağlayan önemli bir unsurdur. Bir bireyin sağlık durumunu, hastalık riskini ve hastalıklara tepki verme işlevini WBC'lerden öğrenilebilir. Ancak WBC'leri sınıflandırma hızını ve doğruluğunu sınırlayan bir zaman kısıtlanması sorunu ve büyük miktarda kan örneğinin işlenmesinde zorluklar vardır. Yapılan örnek bir çalışmada WBC' lerin sınıflandırılması için Destek Vektör Makinesi (SVM) ve Evrişimli Sinir Ağı (CNN) tekniklerinin karşılaştırmalı bir analizini yapmaktadır. Renk, doku ve şekil çıkarılarak WBC'lerin özelliklerini analiz etmek için bir özellik çıkarma işlemi gerçekleştirilmektedir. Her tekniğin sınıflandırma performansı, 200 WBC görüntüsüyle test edilir. Çıkan sonuçlarca CNN'nin SVM'ye kıyasla daha iyi bir WBC sınıflandırma sonucu verdiği gözlemlenmiştir (Ibrahim et al., 2019).

Bu çalışmadaki temel amaç, teşhisin hem hızlı hemde yüksek doğruluk yakınsama oranına sahip olabilmesi için bu iki model birleştirilmiştir. Ayrıca kullanıcı arayüzü programı geliştirilmiştir. Böylece kullanıcı arayüzü programı ile birleşik modelin diğer algoritmalar ile karşılaştırılarak diğer algoritmalardan daha güvenilir ve doğru sonuçlara sahip olduğu ispatlanmıştır.

Bu tez çalışmasında, Bölüm 2'de kullanılan tüm algoritmaların çalışma mantığı ve formülleri açıklanmıştır. Bölüm 3'de kullanıcı arayüzünün genel yapısı verilmiştir. Bölüm 4' teyse sonuçlar ve tartışmalar kısmına yer verilmiştir.

II. KULLANILAN ALGORİTMALAR

Bu bölümde kan tahlili sonuçlarının sınıflandırılması için kullanılan algoritmalar ve bu algoritmalar için kullanılan formüller hakkında bilgiler verilmektedir.

A. Lojistik Regresyon (LOGISTIC REGRESSION)

LR (Logistic Regression, Logit Regression, Log-Linear Classifier) modeli regresyon analizi için kullanılıyor gibi gözüksede gerçekte verileri sınıflandırma amacıyla kullanılmaktadır (Sinan, 2021).

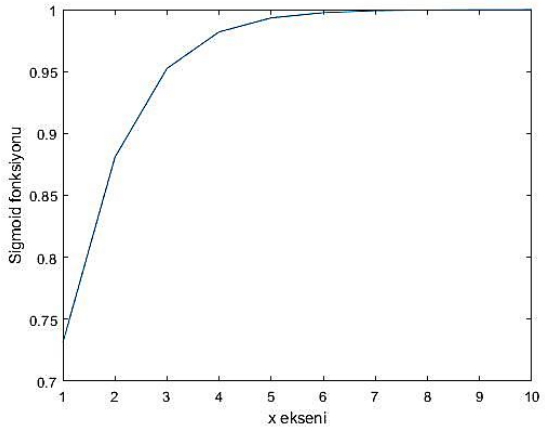
DR, 1 ve 0 aralığında olan çıktı değerlerinin (Bağımsız değer) sınıflandırılmasında doğru sonuçlar verirken LR, 1 ve 0 olan doğruluk değerlerinin sınıflandırılmasında oldukça iyi bir sonuç vermektedir (Sinan, 2021). O halde LR, DR bağımlı değişkenlerinin iki düzeyli (1/0, Doğru/Yanlış, Evet/Hayır, Var/Yok) olduğu özel bir durum olarak düşünülebilir (Alzen et al., 2018; Gürsakal, 2017; Saavedra-García et al., 2019).

LR' ye sigmoid fonksiyonu uygulanması halinde bağımsız değerin tahmin doğruluğu daha da artmaktadır. Bir veri setinde çıktı değerlerin 1 ve 0'lardan ($y \in \{0,1\}$) oluşsun. Veri setindeki çıktı değerlerinin 1 veya 0 değerlerine ait olma olasılığının hesabı aşağıdaki Denklem 1'de gösterilmektedir (Korkmaz et al., 2012; Sinan, 2021).

$$h_\vartheta(x) = P(y = 1|x) = 1 - P(y = 0|x) \qquad \text{(Denklem 1)}$$

Sigmoid fonksiyonu Denklem 2' de gösterilmiştir (Sinan, 2021).

$$\sigma(x) = Sigmoid\ Fonksiyonu = \frac{1}{1+e^{-x}} \qquad \text{(Denklem 2)}$$



Şekil 1. Sigmoid fonksiyonu grafiği

Sigmoid fonksiyonunun kullanımına örnek verilirse: iki adet çıktı değerine sahip problem olsun. Örneğe ait çıktı değerleri; şampiyon ve şampiyon değildir, şeklindedir. Bu çıktı değerleri Olasılık hesabı denklem 1' e göre yazılırsa denklem 3' teki gibi bir formül elde edilir.

$$h_{\vartheta}(x) = P(y = Şampiyon|Pozisyon\ Sayısı) + P(y = Şampiyon\ Değil|Pozisyon\ Sayısı) = 1 \quad \text{(Denklem 3)}$$

Bir Regresyon Modeli, $\boldsymbol{\theta}$= Ağırlıklar vektörü, $\boldsymbol{\theta^T}$ = Ağırlıklar vektörünün transpozu (devriği), $\boldsymbol{x}$ = Girdi vektörü ve $\boldsymbol{\varepsilon}$ = Hata olmak üzere aşağıdaki Denklem 4 ile ifade edilmektedir (Sinan, 2021).

$$y = h_{\vartheta}(x) = \vartheta_0 + \vartheta_1 * x_1 + \vartheta_2 * x_2 + \cdots + \vartheta_n * x_n + \varepsilon = \boldsymbol{\vartheta^T} * \boldsymbol{x} \quad \text{(Denklem 4)}$$

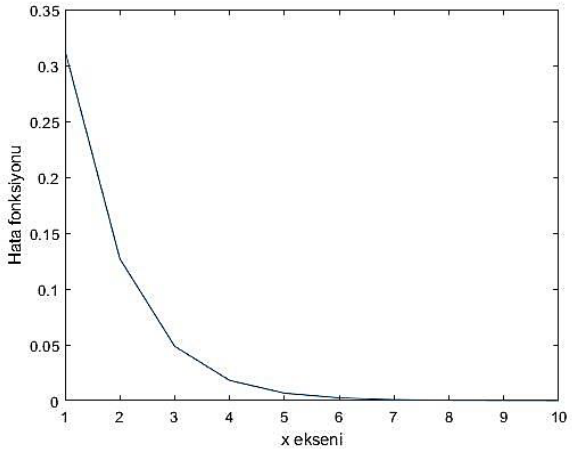
LR, sigmoid fonksiyonu ile kullanılırsa y çıktısı aşağıdaki Denklem 5 ile hesaplanmaktadır (Sinan, 2021).

$$y = h_{\vartheta}(x) = \sigma(\boldsymbol{\vartheta^T} * \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\vartheta^T} * \boldsymbol{x}}} \quad \text{(Denklem 5)}$$

Hata fonksiyonunun hesaplanabilmesi için Sigmoid fonksiyonunun logaritma işlemine tabii tutulması gerekmektedir. Bunun nedeni, x değerleri $-\infty$ değerine yaklaşırsa fonksiyon 1' e yakınsayacaktır. x degerleri $+\infty$ değerine yaklaşırsa fonksiyon 0' a yansıyacaktır. Fonksiyonun 0 olması veri ilişkilerinin tamamen kusursuz olduğunu 1 ise hepsinin hatalı olduğunu göstermektedir. Ancak hiçbir LR modeli ile sınıflandırılmış veriseti, hata fonksiyonlarının çıktısı

1 ve 0 olamaz. Hata fonksiyonunu küçültebilmek için sigmoid fonksiyonun logaritması alınmalıdır. Logaritması alındığında hata fonksiyonu denklemi Denklem 6' da gösterilmiştir (Sinan, 2021).

$$Hata(h_\vartheta(x), y) = \begin{cases} -log(h_\vartheta(x)), & y = 1 \\ -log(1 - h_\vartheta(x)), & y = 0 \end{cases} \quad \text{(Denklem 6)}$$



Şekil 2. y=1 için Logaritması alınmış hata fonksiyonu grafiği

LR (Logistic Regression, Logit Regression, Log-Linear Classifier) modeli daha çok bilimsel verilerin sınıflandırılması amacıyla kullanılmaktadır. Ancak 1 ve 0 olan doğruluk değerlerinin sınıflandırılmasında oldukça iyi bir sonuç vermektedir. Bu nedenle sadece karmaşık olmayan veri setleri için yapılabilir. Veri setindeki çıktı değerlerinin kararsız, evet, hayır vs. gibi ikiden fazla durumu kapsadığında sınıflandırma doğruluğuna güvenilemez. LR' ye sigmoid fonksiyonu uygulanması halinde bağımsız değerin tahmin doğruluğu daha da artmaktadır.

Bu çalışmanın sonucunda Tahlil sonuçlarının değerlendirilmesi için LR algoritmalarının karmaşık veri setleri için sınıflandırma doğruluğuna güvenilemeyeceği çıkarımı yapılmıştır.

B. Şempanze Optimizasyon Algoritması (CHIMP OPTIMIZATION ALGORITHM)

Bu bölümde şempanze grubu ve üyelerinin fonksiyonları, kullanım amaçları ve algoritmadaki önemi anlatılmaktadır. Ayrıca, ChOA'nın matematiksel modeli

ve MLP NN algoritması ile birleştirilebilmesi için gerekli bilgiler ve aşamalar verilmiştir. ChOA, Keşif ve Sömürü olarak iki aşamada hesaplanmaktadır. Her iki aşamada yerel ve küresel arama araçlarına sahiptir. Ancak ChOA kullanan birey bu iki aşamayı da kullanmak konusunda serbesttir. Kulanım amaçlarına göre; küresel arama araçları, tüm veri kümesinin doğruluğunu gerçek verilerle karşılaştırırken, yerel arama araçları ise numuneler için optimum doğruluk sağlamak amacıyla kullanılır. Arama araçları yani arama ajanları şempanzelerdir. Bir şempanze kolonisinde 4 tip şempanze grubu vardır: engelleyici, takipçi, sürücü ve saldırgan. Başarılı bir av (doğru ve güvenilir sınıflandırma oranı) için dört grubun da algoritma içinde yer alması gereklidir.

Eğer şempanzeler gerçek hayattaki görevlerine göre yazılırsa; Sürücüler, avı hapsetmeden 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 saldırgana yönlendirir. Saldırgan, avın kaçış yolunu tahmin eder ve onu avın bölgelerine ve takipçilerine yönlendirir, ardından avı yakalar.

ChOA için matematiksel model aşağıdaki gibi yazılabilir:

$$d = \left| c\, x_{Prey}(t) - m\, x_{Chimp}(t) \right| \qquad \text{(Denklem 7)}$$

$$x_{Chimp}(t+1) = x_{Prey}(t) - ad \qquad \text{(Denklem 8)}$$

$$a = 2fr_1 - f,\ c = 2r_2 \qquad \text{(Denklem 9)}$$

m= Chaotic Value

t değeri, algoritma çalıştırıldığında program içindeki geçerli yineleme sayısını gösterir. *a* değeri bir arama faktörüdür ve –1 ile 1 arasındaki rasgele sayılardan oluşmaktadır. Denklem 9'da da görüldüğü gibi f ve r_1 değerlerine bağlıdır. f vektörü arama ajanlarının ava göre konumlarını iyileştirmek için kullanılır. *c*, rasgele sayılardan oluşan bir vektördür. x_{Prey}, avın konumunu gösteren bir konum vektörüdür, x_{Chimp}, saldırgan rolündeki şempanzenin konumu gösteren bir konum vektörüdür. Keşif aşamasında, Denklem 9 kullanılarak veri setindeki örneklerin (a ve c değerleri) giriş ve çıkışları bulunur.

f vektörü, x_{Chimp} değerini ve konumunu optimize etmek ve güncellemek için kullanılır. Bu nedenle f vektörü, dinamik (şempanzenin konumuna göre

güncellenen) bir katsayıdır. f vektörü, tüm yinelemelerde (iterasyonlarda) doğrusal olmayan şekilde (explonensiyel olarak) 2.5' tan 0' a düşürülür (Aljarah et al., 2018; Khishe & Mosavi, 2020a). Başka bir deyişle, sömürü aşamasındaki şempanzeler, Denklem 8' i kullanarak giriş ve çıkış değerleri için kendi x_{Chimp} değerlerini, yani $x_{Attacker}, x_{Barrier}, x_{Driver}$ ve x_{Chaser}vektörlerini üretirler.

x_{Prey} normalde avın optimum konumudur (örnekteki giriş ve çıkış değerleri) ve çıkış değerlerine göre yaklaşık değerler ancak rastgele bir şekilde seçilir. Ancak bu çalışmada pasif veri setteki gerçek giriş/çıkış değerleri optimum konumu en iyi olacak şekilde seçilmiştir. r_1 ve r_2 değerleri 0 ile 1 arasındaki rastgele sayılardan oluşan vektörlerdir. m değeri ise kaotik bir değerdir. Şempanzenin av bulması için kullanılan bir optimizasyon değeridir ve global olarak kullanılan iyileştirme amaçlı bir değerdir.

Denklem 9' da da görüldüğü gibi a' nın değeri f' ye bağlı olduğundan f vektörü gibi 2,5' tan 0' a düşme eğiliminde olacaktır.

Denklem 13' te görülen $|a| < 1$ duruunda şempanzeler avına saldırır. $|a| > 1$ durumundaysa, a' nın değeri yeniden hesaplanır. Avın yerini bilmek için 4 grup şempanzenin yerini ve uzaklığını bilmek gerekir. Av ve şempanzeler arasındaki mesafe $d_{Attacker}, d_{Barrier}, d_{Chaser}$ ve d_{Driver} değerleri denklem 10' daki gibi hesaplanır. Sonuçlara göre şempanzelerin ava göre konumları ise denklem 11 ile hesaplanır ve denklem 12' de de görüldüğü gibi ortalaması alınır:

$$\begin{aligned} d_{Attacker} &= |c_1 x_{Attacker} - m_1 x| \\ d_{Barrier} &= |c_2 x_{Barrier} - m_2 x| \\ d_{Chaser} &= |c_3 x_{Chaser} - m_3 x| \\ d_{Driver} &= |c_4 x_{Driver} - m_4 x| \end{aligned} \qquad \text{(Denklem 10)}$$

$$x_1 = x_{Attacker} - a_1 \, d_{Attacker}$$
$$x_2 = x_{Barrier} - a_2 \, d_{Barrier}$$
$$x_3 = x_{Chaser} - a_3 \, d_{Chaser}$$
$$x_4 = x_{Driver} - a_4 \, d_{Driver} \quad \text{(Denklem 11)}$$

$$x = \frac{x_1 + x_2 + x_3 + x_4}{4} \quad \text{(Denklem 12)}$$

Şekil 4‘ de görülen MLP NN gizli katmanındaki nöronların *x* değerleri, denklem 11 ve denklem 12 ile bulunur.

Ancak *x* ($x_{Attacker}, x_{Barrier}$, x_{Chaser} ve x_{Driver}) değeri pasif veri setindeki her bir örnek için ayrı olarak hesaplanmalıdır. Bu nedenle Denklem 12 kullanılarak tek bir değere dönüştürülmesi gerekir.

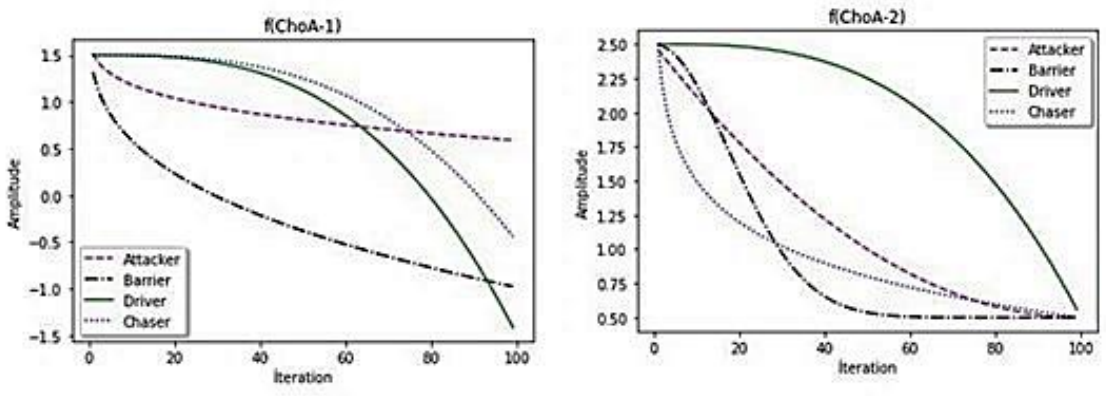
Ortak x_{Chimp} değeri, MLP NN - ChOA'nın gizli katmanının çıktısıdır. x_{Chimp}, D bölümünde de gösterildiği gibi denklem 7 içerisindeki *out* değerine yerleştirilir ve ardından da MSE değeri hesaplanır. D bölümünde de anlatılacaktır.

Vektör *f*‘ yi hesaplamak için formüller, aşağıdaki Çizelge 1' de (Khishe & Mosavi, 2020a) yer almaktadır.

Çizelge 1. f vektörünün dinamik katsayıları

Gruplar	**ChOA 1**	**ChOA 2**
Grup 1: Saldırgan	$1.95 + 2t^{1/4}/T^{1/3}$	$2.5 - (2logt/logT)$
Grup 2: Engelleyici	$1.95 + 2t^{1/3}/T^{1/4}$	$(-2t^3/T^3) + 2.5$
Grup 3: Sürücü	$(-3t^3/T^3) + 1.5$	$0.5 + 2\exp[-(4t/T)^2]$
Grup 4: Takipçi	$(-2t^3/T^3) + 1.5$	$2.5 + 2(t/T)^2 - 2(2t/T)$

T , toplam iterasyon sayısını gösterir. Şekil 3, Çizelge 1' deki denklemler [11] kullanılarak hesaplanan f vektörünün grafikleri gösterilmektedir. T, yani toplam iterasyon değeri 100 alınmıştır. Şekil 3' te de görüldüğü gibi f vektörü 2.5' tan 0' a explonensiyel (doğrusal olmayan bir şekilde) azalma eğilimi göstermektedir. f (ChOA-1), Çizelge 1' deki 'ChOA 1' sütunundaki f vektörünün formüllerine göre hesaplanır. f (ChOA-2), Çizelge 1' deki 'ChOA 2' sütunundaki f vektörü formüllerine göre hesaplanır.



Şekil 3. Dinamik katsayıların matematiksel modelleri

f (ChOA) değerleri, kaotik haritalarında çıkarılacak en iyi puan, sınıflandırma veya hata değerlerinin performansını artırmak için kullanılır. Kaotik haritalama ise m değerinin bulunması için kullanılan ve grafikle çizdirilmiş halidir. Gizli katmanın çıktı değerinin performansını optimize etmek için kullanılan en iyi bilinen kaotik harita formüllerinden bu çalışma içerisinde kullanılanları Çizelge 2' de verilmiştir (Khishe & Mosavi, 2020b).

Bu kaotik haritalama formülleri aynı zamanda ChOA içerisinde en yaygın bir şekilde kullanılan formüllerdir.

Çizelge 2. Çalışma'da kullanılan kaotik haritalama formülleri

İsim	Formüller	Aralık
Quadratic	$x_{i+1} = {x_i}^2 - c, \quad c = 1$	(0,1)
Gause/Mouse	$x_{i+1} = \begin{cases} 1, & x_i = 0 \\ \frac{1}{mod(x_i, 1)}, & otherwise \end{cases}$	(0,1)
Logistic	$x_{i+1} = \propto x_i(1 - x_i), \quad \propto = 4$	(0,1)
Bernoulli	$x_{i+1} = 2x_i(mod1)$	(0,1)
Sine	$x_{i+1} = \frac{a}{4}\sin(\pi x_i), \quad a = 4$	(0,1)

Yerel sınıflandırıcı olarak kullanılan μ değeri x_{Chimp}' i sınıflandırmak için kullanılır. Denklem 13' te kullanılan μ değeri, şempanzenin ava yaklaşma olasılığını göstermektedir (Khishe & Mosavi, 2020a, 2020b). μ değeri 0 ile 1 arasında rastgele belirlenen bir değerdir. μ>0.5 ise şempanzenin av pozisyonu manuel olarak güncellenir. μ>0.5 ise kaotik değer verilir. Bu çalışmada şempanzenin konumu için başlangıç kaotik değeri 0,7 alınarak hesaplanmıştır (Khishe & Mosavi, 2020a, 2020b).

$$x_{Chimp}(t+1) = \begin{cases} x_{Prey}(t) - ad \; if \; \mu > 0.5 \\ Chaotic \; Value \; if \; \mu < 0.5 \end{cases} \quad \text{(Denklem 13)}$$

Şempanze Optimizasyon Algoritması, tek başına verileri sınıfandırmak için kullanılmaz. İsminden de anlaşılacağı gibi sınıflandrma doğruluğu, performans arttırma(güvenirlilik), hızlı sonuçlandırma vs amaçlarla kullanılan bir iyileştirme algoritmasıdır ve kullanım amacına göre başka algoritmalarla birleştirilerek kullanılır. Bu çalışmada Çok Katmanlı Algılayıcı Sinir Ağları ile birlikte birleştirilerek kullanılmıştır.

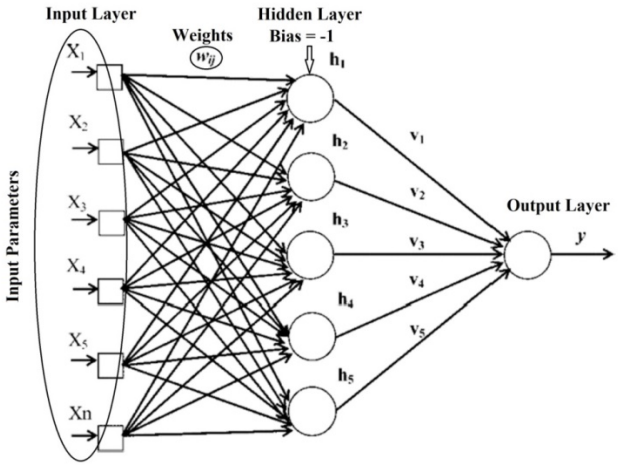
C. Çok Katmanlı Algılayıcı Sinir Ağları Algoritması (MLP NN)

Yapay sinir ağları (YSA), insanların sinir sistemini oluşturan nöronların oluşturduğu ağa benzer yapıdadır. YSA, en bilinen türleri olarak ikiye ayrılır. YSA'da üç girdi katmanı ve bir çıktı katmanından oluşan modele Tek Katmanlı Sinir Ağları (SLNN), girdi, çıktı ve gizli katmanlardan oluşan modele Çok Katmanlı Algılayıcı Sinir Ağları (MLP NN) denir. Bu çalışmada MLP NN kullanılmılmıştır.

Çok Katmanlı Algılayıcı Sinir Ağları, ileri beslemeli bir yapıya sahip olup basit ve güvenilir bir YSA türüdür. MLP NN' den gelen çıkış değerleri ikili omak zorundadır. Sonuçların ikili olması demek, çıkış katmanındaki çıktı değerlerinin Evet/Hayır, 0/1, Doğru/Yanlış vb. anlamına gelmektedir. MLP NN, ikiye ayrılır: Sığ Sinir Ağları ve Derin Sinir Ağları.

Eğer yalnızca bir gizli katman varsa bu modele Sığ Sinir Ağları, birden fazla gizli katman varsa bu modele Derin Sinir Ağları denir. Bu ağlar arasındaki temel fark, aşırı uyum durumudur. Program çalıştırıldığında yinelemeler yani iterasyonların işlenmesi devam ederken, belli bir yinelemeden sonra hata oranı

tekrarlanır. Bu tekrarlanma durumuna aşırı uyum (overfitting) denir. Bu olumsuz durum MLP NN için MLP NN' nin gizli katmanı içerisine başka bir regresyon veya sınıflandırma algoritması birleştirilerek önlenebilir. Literatürde, MLP NN genellikle Lojistik Regresyon, Destek Vektör Makineleri, Parçacık Sürü Optimizasyonu (PSO) ve Karar Ağacı gibi sınıflandırma amaçları için kullanılan modellerden daha doğru performans sergiler (Gogou et al., 2001; Mathison et al., 2020; Shahmoradi et al., 2018).



Şekil 4. MLP NN mmari yapısı

MLP NN' ün çalışma mantığı basittir. Öncelikle girdi katmanı girdilerin değerlerini alır, gizli katmandaki nöronlara gönderir. Gizli katmanın her nöron' u içerisinde ağırlıklı bir girdi toplamı hesaplanır. Hesaplanan bu değer, bias değeri ve aktivasyon fonksiyonu ile çıkış değerine döndürülür. Bu çıkış değeri 0 ile 1 arasındadır.

Bu çalışmada kullanılmaktadır. Sığ türde Çok Katmanlı Algılayıcı Sinir Ağları için mimari yapı Şekil 4' te gösterilmiştir (Yao et al., 2019; Yılmaz & Bozkurt, 2012). Gizli katman için formüller denklem 14'teki gibidir.

$$(I_1)_j = \sum_{i=1}^{n} (W_1)_{ji} * X_i \quad \text{(Denklem 14)}$$

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

Bu denklemde W_1, X_1 giriş öngerilim değerine rastgele atanan ağırlık değerini temsil eder ve ji, giriş katmanından gizli katmana ait nöron sayısını gösterir.

Aşağıda denklem 15' in açılmış hali Denklem 14'te gösterilmektedir. Denklem 15, gizli katman sonucunun çıktı değerini göstermektedir (Gomes & Ludermir, 2013; Kazerouni et al., 2020; Vecci et al., 1998).

$$(Out)_1 = f((I_1)_j) \qquad \text{(Denklem 15)}$$

$(Out)_1$ sonucu, ağırlıklı toplam değeri aktivasyon fonksiyonu tarafından belirlenen eşik değerine göre 0 veya 1 olarak sınıflandırılır. Ağırlıklı toplam değer eşik değerine eşit veya büyük ise 1, eğik değerden küçük ise 0 yapılır. En sık kullanılan aktivasyon fonksiyonu Lojistik (Sigmoid) fonksiyonudur. . Bunun dışında problemin tipine ve ağın yapısına göre Unit Step, Sign, Linear, Piece-Wise, Hyperbolik Tanjant, Rectified Linear Unit (RELU) veya Softplus fonksiyonları kullanılabilir. Sigmoid fonksiyonlarının çıkış değerleri her zaman (0,1) aralığındadır. Bu nedenle iyi bir sınıflandırıcı olarak kabul edilebilir (Sinan, 2021). Aşağıda çizelge 3' te aktivasyon fonksiyonlarının formülleri verilmektedir (Sinan, 2021).

Çizelge 3. Aktivasyon fonksiyonlarının formülleri

Aktivasyon Fonksiyonu İsmi	Formüller
Birim Basamak (Unit Step)	$f(u) = \begin{cases} 0, & u < 0 \\ 1, & u \geq 0 \end{cases}$
İşaret (Sign)	$f(u) = \begin{cases} -1, & u < 0 \\ 0, & u = 0 \\ 1, & u > 0 \end{cases}$
Doğrusal (Linear)	$f(u) = u$
Parçalı (Piece – Wise)	$f(u) = \begin{cases} 1, & u < {}^1/_2 \\ u + {}^1/_2, & -{}^1/_2 < u < {}^1/_2 \\ 0, & u \leq -{}^1/_2 \end{cases}$
Lojistik (Logistic, Sigmoid)	$f(u) = \frac{1}{1 + e^{-u}}$
Hiperbolik Tanjant (Hyperbolic Tangent)	$f(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$
Doğrultucu (ReLU, Rectified Linear Unit)	$f(u) = max(0, u)$
Doğrultucu (Softplus)	$f(u) = ln(1 + e^u)$

Bu çalışmadaysa, bütün aktivasyon fonksiyonları arasında en sık kullanılan ve sınıflandırma doğruluk oranı en yüksek aktivasyon fonksiyonu olan Sigmoid

fonksiyonu kullanılmıştır. Bu fonksiyonun kullanılmasının nedeni en çok bilinmesinden dolayı değil bu çalışmada olduğu gibi sınıflandırma doğruluğunun en yüksek olması istenen durumlarda en yüksek skoru vermesindendir.

D. Şempanze Optimizasyon Algoritması İle Eğitilen Çok Katmanlı Algılayıcı Sinir Ağları Algoritması (MLP NN - ChOA)

Algoritmaları birleştirmek için öncelikle istenilen durumlar (hız, doğruluk, güvenirlilik) ve algoritmaların avantajları ve dezavantajları incelenmelidir. Bu çalışmada her iki algoritmayı birleştirmek için öncelikle temel algoritmanın (MLP NN) genel dezavantajları incelenmiştir. MLP NN'nin genel dezavantajlarını incelemedeki amaç, tahlil çıktılarının sınıflandırılmasında doğru ve güvenilir sonuç vermesini engelleyen durumları ortadan kaldırmaktır.

Yapay sinir ağlarının genel olarak en önemli dezavantajı uygun ağ yapısının belirlenememesi, sadece sayısal bilgilerle çalışılabilmesi ve ağın sonuçlanmasındaki işlem süresinin bilinememesidir. Ayrıca sinir ağlarının yapısını belirlemek için belirli bir kural yoktur. Uygun ağ yapısı, deneyim ve deneme yanılma yoluyla elde edilir.

Verisetindeki örneklerdeki hataların belirli bir değere indirilmesi eğitimin tamamlandığı anlamına gelir. Ancak bu değer sınıflandırma sırasında optimum sonuçları vermez. Ayrıca algoritma belirli bir kurala tabii değilse işlem süresi de öngörülemeyen sorunlara neden olabilir. Örneğin, çıktının belirli sayıda yinelemesinden sonra gerçekleşen aşırı uyum durumu, en önemli öngörülemeyen sonuçlardandır (Livingstone et al., 1997).

Aşırı uyum sorununu gidermek için giriş bölümünde stokastik eğiticilere sahip olduğu açıklanan ChOA algoritması, MLP NN' nin gizli katmanına yerleştirilmiştir.

Algoritmanın modellenmesi ve sonuçların sınıflandırması ve tahmini işlemlerinden önce pasif veri seti eğitim ve test seti olarak ikiye ayrılmalıdır. Böylece sonuçların daha güvenilir ve doğru olması sağlanır. Öncelikle eğitim seti üzerinde modelin eğitilmesi işlemleri yapılır, ardından test seti üzerinde de tahmin süreci tamamlanarak sınıflandırılır. Eğitim sürecindeki amaç, modeli eğitmek için gerçek veri kümesinden başka bir veri kümesi oluşturmaktır.

Böylece eğitim süreci ile veriseti algoritmanın uygulanmasına hazır hale getirilir. Test sürecinde ise eğitim setinde geliştirilen model değerlendirilir. Önerilen eğitim yönteminde iki önemli faktör dikkat edilmelidir: birincisi, ChOA'da MLP NN algoritmasının (arama araçları) temsili ve ikincisi, maliyet fonksiyonunun (iyileştirme, optimizasyon) seçimidir. Arama araçları yada ajanları ve iyileştirme faktörleri Şempanze Optimizasyon Algoritması bölümünde detaylı bir şekilde anlatılmıştır.

MLP NN - ChOA algoritmasının gizli katmanından çıkan sonuçlar, eğitim sürecindeki her bir şempanzenin ağırlığı (W_{ij}) ve bias ile oluşturulan tek boyutlu bir vektör şeklindedir. Her vektörün uzunluğu, MLP NN'deki ağırlıkların ve sapmaların toplamına eşittir ve aşağıdaki gibi tanımlanabilir (Khishe & Mosavi, 2020a):

$$Vector\ Length = (n * h) + (2 * h) + 1 \qquad \text{(Denklem 16)}$$

Denklem 16'da görülen n degeri, giriş sayısı ve h degeri ise gizli katmandaki nöron sayısıdır (Aljarah ve diğerleri, 2018). Gizli katmandan elde edilen çıktı değeri, denklem 17' de gösterilen Lojistik aktivasyon formülü ile hesaplanır:

$$f = \frac{1}{1 + e^{-Out}} \qquad \text{(Denklem 17)}$$

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

$$MSE = 1/m \sum_{i=1}^{m} (f - f')^2 \qquad \text{(Denklem 18)}$$

Denklem 18' de görülen f değeri istenen sonuçtur, f' değeri ise değerlendirilen sonuçtur ve m değeri ise eğitim veri setindeki örnek sayısıdır. Eğitim veri setindeki örnek sayısı bazen gerçek veri setindeki örnek sayısı ile uyuşmuyor gözükebilir. Bunun nedeni gerçek veri setinde girilen yanlış değerlerdir (Örneklerin boş bırakılması yada yarım doldurulmuş olması vs.). Bu nedenle program akışı yazılırken bu yanış değerlerin filtrelenmesi gereklidir.

Kısaca bileşik modelin oluşturulma aşamaları özetlenirse, öncelikle algoritmanın daha iyi anlaşılabilmesi, arama ajanlarına göre güncellenip daha doğru snıflandırmanın yapılabilmesi için denklem 7 deki formüller her bir şempanzenin sonucu daha doğru arayabilmesi için denklem 11 de de görüldüğü gibi özelleştirilir. Denklem 10'da görülen $d_{Attacker}, d_{Barrier}, d_{Chaser}$ ve d_{Driver} formülleri ile av ile şempanzeler arasındaki mesafe hesaplanp, Çıkan sonuçlara göre de şempanzelerin ava göre konumları hesaplanır ve ortalaması alınır. Denklem 11 ve 12'deki formüller kullanılarak MLP NN gizli katmanındaki nöronların x değerleri bulunur. Ancak x değeri ($x_{Attacker}, x_{Barrier}, x_{Driver}\ and\ x_{Chaser}$) veri kümesindeki örneklerin her bir değeri için hesaplandığından tek bir değere dönüştürülmesi gerekir. Tek bir değere dönüştürmek içinde denklem 13 kullanılır. Ortak x_{Chimp} değeri, MLP NN - ChOA'nın gizli katmanının çıktısıdır. x_{Chimp}, out olarak denklem 17' ye yerleştirilir ve ardından denklem 18 kullanılarak MSE değeri hesaplanır. Gizli katmandan elde edilen çıkış değeri Lojistik aktivasyon formülü ile hesaplanır.

Daha önce de belirtildiği gibi, bu çalışmada örnek olarak sınıflandırma sonuçlarının kullanıcı arayüzünde gösterilmesi amacıyla Kabakulak hastalığı antikorlarına ait sonuçlarının bulunduğu pasif veri seti, ChOA ile eğitilmiş MLP NN tabanlı bir bileşik model (Ensemble Model) tasarlanmıştır. ChOA ile eğitilen MLP NN algoritmasının sözde kodu ise , aşağıdaki Şekil 5' te yer almaktadır (Khishe & Mosavi, 2020a).

Pseude Code of Training an MLP NN using ChOA

Get your dataset.
Split your dataset into Dependent and Independent vector
Multiply dependent vector by weight value
Calculate the position of each chimpanzee belonging to the vector entering the hidden layer.
Calculate the search agent to be created based on the number of instances in the dependent vector.
Until the stop condition
Calculate the fitness of each chimpanzee.
$X_{Attacker}$ = The best search agent
X_{Chaser} = The second search agent
$X_{Barrier}$ = The third search agent
X_{Driver} = The fourth search agent
While (m< Max_iter)
  For each chimp
    Divide the chimpanzees into groups.
    Optimize chimpanzees using f, m and c
    Using f, m, and c, calculate a first, then d
  End For
  For each search chimp
    if ($\mu < 0.5$)
      if ($|a| < 1$)
        Update each search agent according to the formula: $x_{Chimp}(t+1) = x_{Prey}(t) - ad$
      Else if ($|a| > 1$)
        Select randomly search agent
      End if
    Else if ($\mu > 0.5$)
      Update position of each search agent according to the Chaotic Map
    End if
  End For
  Update f, m, c ve a
  Update $X_{Attacker}$, X_{Chaser}, $X_{Barrier}$ ve X_{Driver}
  Calculate X positions according to the formula: $X = (X_{Attacker} + X_{Chaser} + X_{Barrier} + X_{Driver})/4$
  Create X_{Chimp_X}, X_{Chimp_y} vector for each Dependent and Independent vector respectively
  Calculate X_{Chimp_X} vektor with Sigmoid activation function
  Calculate the error value E
  Calculate MSE.
  if (MSE <MSETarget)
    Draw the log of the MSE values
  End İf
End While

Şekil 5. ChOA kullanarak eğitilen MLP NN algoritması için sözde kod

E. Rastgele Orman (RANDOM FOREST)

Rastgele orman algoritması, genel olarak sınıflandırma amacıyla kullanılan ve böl - fethet yaklaşımına sahip bir algoritmadır (Breiman, 2001). Rastgele orman algoritması, karar verme ağaçlarından veya regresyonlardan oluşmaktadır. Ancak bu agaçlar ve regresyonlar zayıf öğrenme durumuna sahiptir. Bu durum RF' nin en büyük dezavantajıdır.

Ancak bu ağaçların ve regresyonların herbirisi birleştirildiğinde algoritmanın öğrenme özelliğini güçlü hale getirir. Ayrıca herhangi bir parametre ayarı olmaksızın hem eğitim hemde tahminde çok iyi bir performans gösterir. Bütün bu özelliklerse RF algoritmasının en büyük avantajlarındandır. Bir diğer avantajı ise sınıflandırmada hızlı sonuç vermesidir.

Ancak bu avantajlı durum düzenli ve basit veri setleri için geçerlidir. Bu durumda RF algoritmasının en büyük dezavantajı, kayıp yada düzensiz verilere sahip veri setlerinde yada çok fazla veriye sahip verisetlerinde tahmin doğruluğunun ve eğitim perfomansının düşmesidir. Rastgele orman algoritması ile Destek Vektör Makinesi, Karar Ağaçları, Naive Bayes Algoritmasına kıyasla daha yüksek tahmin doğruluğu elde edilmektedir (Gunčar et al., 2018).

İncelenen literatür araştırmalarına göre RF, tüm deneysel aktiviteler sonucunda ulaşılan ve eksik veri bulunmayan verisetlerinin sınıflandırılması için kullanılır. Örneğin, COVID-19' un hızlı testi için yapılan bir çalışmada RF algoritması verisetindeki çıktıların sınıflandırılması ve bu sınıflandırılmaya göre sonucun yorumlanması amacı ile kullanılmıştır.

Bu çalışmada veriseti NHANES'ten hazır olarak alınmış olduğundan (Herhangi bir deneysel süreç ile elde edilmeyen verisetlerine pasif verisetleri denmektedir) deneysel bir süreç ve bu süreç sonucunda elde edilen veri sonuçları için RF başka bir algoritma ile birleştirilmesi söz konusu değildir.

NHANES'ten alınan verisetleri, tahlil sonuçlarının gerçek hayatta labaratuvar testlerinden elde edilmiş olduğu ve verilerin bir veri kümesi haline getirilmiş halidir.

Bu çalışmada RF sadece sınıflandırmada çalışmamızın ana algoritması olan ChOA ile eğitilmiş MP NN algoitması ile karşılaştırılmak için kullanılmıştır.

F. Destek Vektör Makineleri

Destek vektör makineleri genellikle sınıflandırma ve regresyon amacı ile kullanılan denetimli öğrenme yöntemlerine sahip algoritma çeşitlerindendir.

SVM birçok tahlil sonuçlarının yada görüntüleme teknikleri ile elde edilmiş verilerin sınıflandırılmasında kullanılmıştır.

Yapay Sinir Ağları (YSA) ile SVM arasındaki temel farklardan birisi SVM'in tek bir çözüme odaklı olarak (tek bir optimal çözüm) herhangi bir sistemi eğitirken, YSA birden fazla çözümü karşılaştırır ve en iyi optimal çözümü bularak eğitir. Bu durum SVM için sonucu daha pratik ve hızlı ulaşma açısından avantaj hemde tek bir çözüme odaklandığı için bir dezavantajdır. SVM de MLP NN çalışma mantığına benzer çalışmaktadır. Bu benzerlik daha detaylı anlatılırsa, öncelikle veriseti (ister pasif ister deneysel sonuçlarla elde edilen dinamik) algoritmaya tanıtılır. Ardından sırasıyla öznitelikler değerlendirilir. Özniteliklerin belirlenmesi MLP NN için olmasada bu algoritmanın çıktıları sınıflandırabilmesi için oldukça önem taşıdığı görülmüştür. Daha sonra gerçek verisetinden yola çıkılarak eğitim ve test veri setleri oluşturulur.Ayrıca bir diğer benzerlikte MLP NN algoritmasında olduğu gibi sistem performansını denetlemek ve değerlendirmek amacıyla MSE hesaplanmaktadır (Karamizadeh et al., 2014; Sinan, 2021).

Destek vektör regresyonu için verisetinin sürekli (boş ve eksik girilen örnek olmaksızın) verilerden oluşması gereklidir. SVM için tahmin edilmek istenen veriler önceden veriseti içerisinde kategorik hale getirilmiş olmalıdır (Sinan, 2021). Bu durumda SVM verisetinin sınıflandırılmasını yapabilir. Bu durum aynı zamanda bu çalışma için bir dezavantaj niteliğindedir. Çünkü NHANES'ten alınan verisetleri her nekadar pasif verisetleri olsada verisetlerinin içerisinde eksik girilen veya boş bırakılmış birçok örnek olduğu görülmüştür. Bu nedenle de bileşik modelin ana algoritması olarak seçilmemiştir.

SVM için ROC eğrisinin çizdirilmesi AUC kavramnın bilinmesi açısından oldukça büyük önem taşıdığı anlaşılmıştır. ROC eğrisi, makine öğrenmesinde algoritmanın gerçek performansını gösteren asıl faktördür. ROC eğrisi kavramı, aslında ayırıcı veya karar değişkeni kavramına dayanmaktadır (Hajlan-Tilaki, 2013). AUC ise ROC eğirisinin altında kalan alandır.

AUC değeri ise algoritmanın sınıflandırma doğruluğunu ne kadar iyi performans doğruluğu ile verdiğini göstermektedir. Yani SVM sonucunda veriseti sınıflandırma doğruluğu %90 olan bir makine öğrenmesi sistemi için AUC değeri %60'larda çıkabilmektedir. Bunun anlamı %90 olan bu sınıfladırma doğruluğunun güvenirlilik derecesinin aslında %60 olduğunu gösterir, denilebilir. Ancak bu durum Destek Vektör Makineleri başlığının ilk paragraflarındada

açıklanan verisetinin SVM' e göre uygunluk derecesine bağlıdır. ROC eğrisi bu çalışmada Kullanıcı arayüzü ekranında da SVM' in sınıflandırma performansını göstermek amacıyla konulmuştur ve diğer algoritmalarla karşılaştırması Sonuçlar ve Geliştirmeler bölümünde açıklanacaktır (Hajlan-Tilaki, 2013; Oğuz, 2019).

O halde algoritmalar probleme ve problemde istenen sonuca göre seçilmelidir. SVM 'in en önemli avantajlarından diğeri de her bir parametre ile ayrı bir çözüm elde edilir. Bunun anlamı, her bir parametre seçimi ile benzersiz bir optimal sonucu elde edilir. Yine de bu yöntem, YSA gibi birden fazla optimal çözüme ulaşabilmesi için global SVM yerine birden fazla SVM kullanılır. Böylece sınıflandırma oranı ve performans artar. Ama birden fazla SVM kullanılması da sistemi ağırlaştırır.

Ayrıca verisetinin karmaşık olduğu problemlerde hızlı çözüme ulaşılması isteniyorsa sistemi ağırlaştırmasından dolayı istenmeyen bir durum meydana getirecektir (Anguita ve diğerleri, 2010).

Kullanıcı arayüzünde her ne kadar SVM için sınıflandırma doğruluğu paylaşılmasa sonuçlar bölümünde sınıflandırma doğruluğu oranı ayrıca gösterilecek ve açıklanacaktır.

III. KULLANICI ARAYÜZÜ TASARIMI

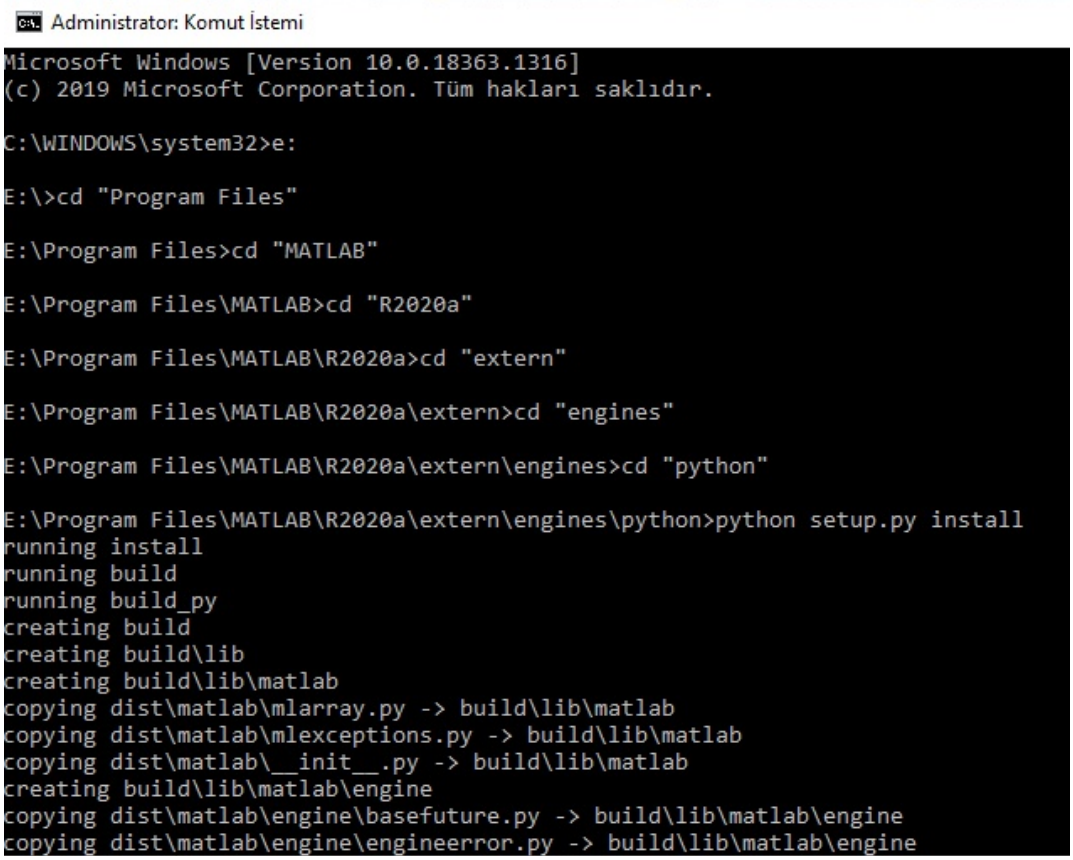
MLP NN + ChOA bileşik modelinin programlaması MATLAB 2020 programına yapılmıştır. Ancak Kullanıcı Arayüzünün tasarımı ve MLP NN + ChOA bileşik algoritması ile sınıflandırma doğruluğunun karşılaştırılmasına ilişkin programlama kodları ise Anaconda Navigator 3 paket programı içerisinde Python tabanlı olarak çalışan Jupyter Notebook programı ile çalışılmıştır. Dolayısıyla MATLAB kodlarının Python ile uyumulu çalışmasını sağlayan ara bağdaştırıcının kurulumuna ihtiyaç vardır. Bu ara bağdaştırıcı ise MATLAB API' dir. Ayrıca bu bölümde Kullanıcı Arayüzü Tasarımında kullanıcıların girmesi gereken parametreler ve bu parameterelerin tanmları verilecektir. Ayrıca grafiklerin içeriği hakkında da ön bilgiler paylaşılacaktır.

A. Matlab Apı Kurulumu

Bu çalışma hazırlanırken Anaconda (Notebook 6.3.0 versiyon) ve Matlab 2020 programlarından ve Matlab API ara bağdaştırıcısından yararlanılmıştır.

MATLAB motorunu bir Python oturumunda başlatmak için önce MATLAB API' si bir Python paketi olarak kurulmalıdır. Anaconda programı python'un 3.8' lik sürümünü kullandığından MATLAB 2020 programı kurulmuştur.

Kurulum için öncelikle Command prompt dosyası yönetici olarak açılmaldır ve Command (cmd) penceresi içerisine MATLAB programının dosyalarının bulunduğu klasör içerisinde python.py dosyasının bulunduğu dosya adresi girilmelidir. Şekil 6'da kurulum için Command penceresine girilen kodların ekran görüntüsü yer almaktadır.



Şekil 6. Command penceresine girilen kodlara ait ekran görüntüsü

Python oturumunda başlatılabilmesi için MATLAB API' nin temel kurulumu tamamlandıktan sonra MATLAB programının editor command penceresinde Python ve API dosyasının kurulumu tanıtılır.

Şekil 7' de Matlab içinde python'un tanıtılmasına ait kod örneği bulunmaktadır. Kurulum için yardım sayfasına MATLAB' ın kendi sitesi olan Mathworks' ten de ulaşılabilmektedir.

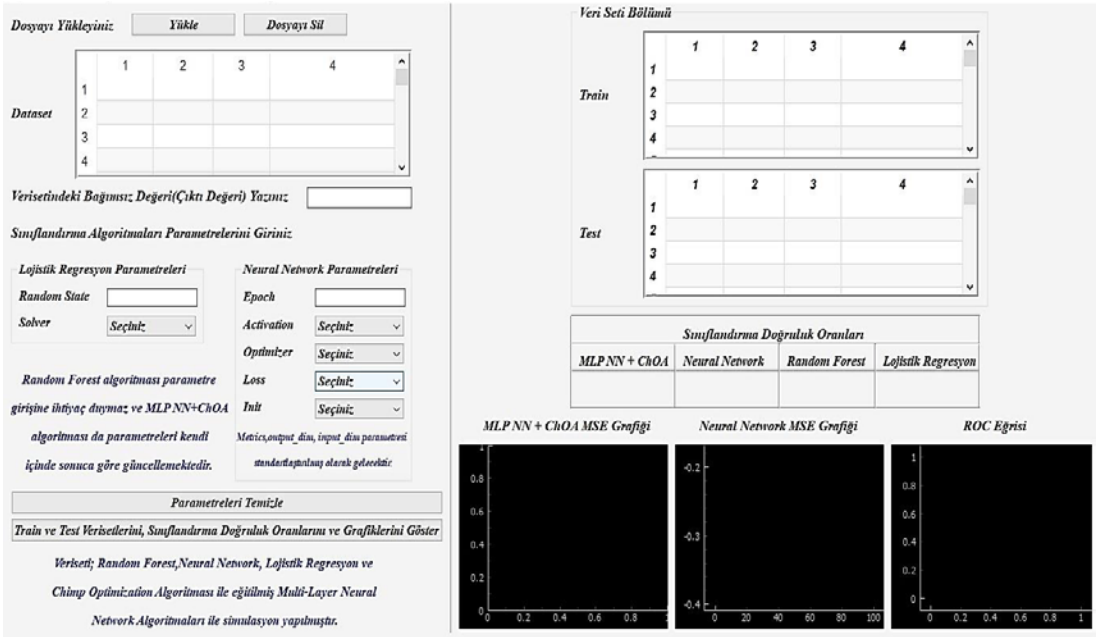
```
cd (fullfile('e:\Program Files\MATLAB\R2020a','extern','engines','python'));system('python setup.py install');
```

Şekil 7. Matlab içinde python'un tanıtılmasına ait kod örneği

B. Kullanıcı Arayüzü Parametreleri Ve Genel Tasarım

Kullanıcı arayüzü programı ile pasif yada dinamik verisetleri (Dinamik versetinin oluşturulabilmesi için sistem kurulumu gereklidir) arayüze yüklenen algoritma modelleriyle modellenerek algoritmalara ait sınıflandırma doğruluğu, ROC eğrisi ve ChOA-MLP NN birleştirilerek elde edilen birleşik modele ait MSE

(Mean Squared Error–Ortalama Kare Hatası) grafiği görüntülenmektedir. Şekil 1'de boş Kullanıcı arayüzü ekranı verilmiştir.



Şekil 8. Kullanıcı arayüzü programına ait boş arayüz ekranı

Kullanıcı arayüzü ekranında veriseti "Dataset" başlığı altındaki excel tablosunda gösterilecektir. Kullanıcı, veri seti içindeki sonuç değerlerini/değerini "Bağımsız değer"e yazmalıdır.

Kullanıcı arayüzü programında RF algoritması kullanıcının girmesi gereken herhangi bir parametre ayarına gerek olmadığı için RF' e ait herhangi bir parametre ayar girişi de istenmemiştir. RF için Random state değeri LR ile aynı olacaktır. Bunun nedeni algoritmalara ait sınıflandırma doğrulupu oranı hesaplanırken adil bir şekilde karşılaştırabilmektir.

MLP NN + ChOA birleşik (ensemble) algoritması ise yüklenen verisetine uygun olarak parametrelerini kendisi günceller. Bu parametreler chaos değeri (m), verisetindeki örneklerin toplam satır ve sütn sayısı ve kullanıcı tarafından girilen verisetidir. Bu nedenle hem RF hemde geliştirilen birleşik model için kullanıcıdan herhangi bir parametre ayarı istenmemiştir.

Lojistik Regresyonda kullanıcı tarafından belirlenen iki önemi parametre vardır: solver ve random state. Random state, rastegele durum anlamına gelir. Kullanıcı istediği değeri integer yani tamsayı olması şartıyla girebilmektir. Ancak

random state parametresi girilirken dikkat edilmesi gereken en önemli husus; girilecek değerin verisetindeki bağımlı değerlerden küçük olması gerektiğidir.

Solver, çözümleyici fonksiyon anlamına gelir ve kullanıcı arayüzü ekranında en sık kullanılan 5 fonksiyon çeşidi seçilmektedir:

- Liblinear,
- Sag,
- Saga,
- Newton-cg,
- Lbfgs

'Newton-cg', 'sag' ve 'lbfgs' çözümleyicileri, yalnızca birincil formülasyonla L2 düzenlileştirmeyi destekler veya hiçbir düzenlileştirme yapmaz. 'liblinear' çözücü, yalnızca L2 cezası için ikili bir formülasyonla hem L1 hem de L2 düzenlemesini destekler. Elastic-Net düzenlemesi yalnızca 'destan' çözücüsü tarafından desteklenir. L1 düzenlemesi, Lasso Regresyonu, L2 düzenlenmesi ise Ridge Regresyonu olarak bilinir. Bu iki parametre LR regülasyon fonksiyonları arasında en bilinen foksiyonlardır (Hoerl, 1970). L1 regresyonu, model parametrelerinin mutlak değerlerinin toplamını gerçek fonksiyona eklerken, L2 regresyonu, bunların karelerinin toplamını ekler (Zou, 2005; ZongBen, 2010).

MLP NN' te ise kullanıcı tarafından girilmesi istenen 5 önemli parametre ayarı vardır: Epoch, Activation, Optimizer, Loss ve Init'tir. Epoch, iterasyon sayısıdır. Activation, gizli katmandan çıkan sonuca uygulanan fonksiyon çeşitleridir ve Arayüz için;

- Relu,
- Sigmoid,
- Tanh,
- Gaussian
- Softplus

fonksiyonları seçilebilmektedir. Optimizer parametresi ile en iyileme algoritmasının belirler. En iyi 3 parametre seçilebilmektedir:

- Adam,
- AdaGrad,
- RMSprop

Bu üç iyileme parametreleri en sık kullanılan paramaetrelerdir. Literatür çalışmlarına göre en iyi iyileme parametresi Adam' dır. Hem AdaGrad hemde RMSProp parametreleri, iterasyon sayısı artıkça sınıflandırma doğruluğunun ve dolayısıyla öğrenme oranının azaldığı gözlemlenmiştir.

Loss parametresi ile hata fonksiyonları tanımlanır:

- Binary crossentropy,
- Mean absolute error (MAE),
- Mean squared error (MSE),
- Categorical hinge

olarak 4 tip hata fonksiyonu çeşidi seçilebilir. En uygun ve en sık kullanılan loss parametresi MSE (Mean squared error)' dir. Ancak bu hata parametreleri diğer algoritmalar içerisinde de kullanılabilir. Binary crossentropy, ikili çıktılar (Evet/Hayır, 1/0 vb gibi) için kullanılır. MSE, L2 kaybını gözlemlemek amacıyla kullanılır. MAE, L1 kaybını gözlemlemek amcıyla kullanılır. Categorical hinge, Binary crossentropy gibidir. Tek fark Binary Crossentropy $y_{gercek(1)} - y_{Doğru}$ arasında yapılırken Categorical Hinge $y_{gercek(0)} - y_{yanlis}$ arasında yapılmasıdır. Y çıkış demektir.

Init parametresi ile başlangıç ağırlık değerlerinin belirleyen seçenekler tanımlanır ve arayüzden:

- Uniform,
- Lecun uniform,
- Identity,
- Orthogonal

olarak 4 çeşit ağırlık belirleme seçeneği seçilebilir. Metrics parametresi bu çalışma için varsayılan olarak doğruluk değeri (accuracy) olarak tanımlanmıştır.

Kullanıcı kendisinden istenen bu parametreleri girmesi sonucunda eğitim (train)-test veri setleri, algoritmaların sınıflandırma doğruluk oranları ve grafikler gösterilecektir.

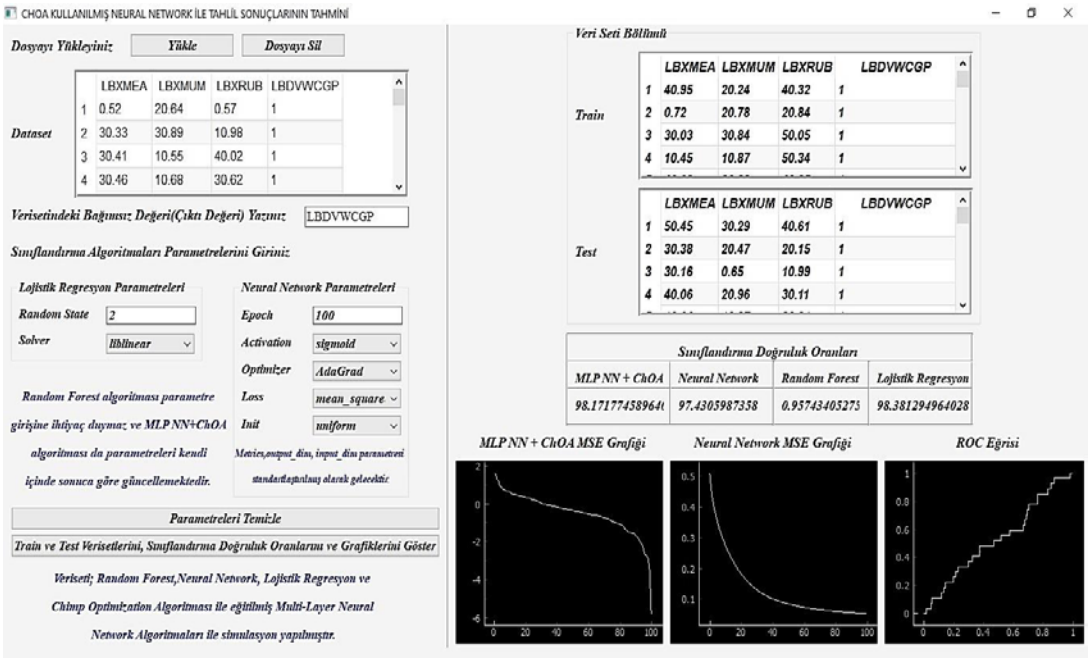
IV. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasındaki amaç, MLP NN algoritmasının gizli katmanına ChOA ekleyerek oluşturulan birleşik modeli, kan tahlili sonuçlarının değerlendirilmesi sonucunda elde edilen sınıflandırma doğruluğunun ve sonuca ait güvenilirlik derecesinin diğer algoritmalara nazaran daha yüksek olduğunu hazırlanan Kullanıcı Arayüzü programında gösterek kanıtlamaktır.

İki algoritmanın harmanlanması ile elde edilen bu bileşik model ile MLP NN algoritmasının en büyük dezavantajı olan aşırı yükleme yada aşırı uyum sorunu çözümlenmiştir.

Verisetlerinin işlenmesi sonucu elde edilen sınıflandırma doğruluğu oranı ise algoritmanın tahmin doğruluğu hakkında bilgi vermektedir. Hata grafiği (MSE 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.

NHANES sitesinden hazır olarak alınan ve kan testleri ile elde edilen Kızamık, Kabakulak, Suçiçeği ve Kızamıkçık hastalıklarının örnek sonuçlarına göre hazırlanmış ancak yine hastalıklarla ilgili herhangi bir pasif veri setine de uygulanabilecek algoritmaların yüklendiği Kullanıcı arayüzü programıyla hastalığın tahmin doğruluğu oranları ve grafikler 4 çeşit algoritma ile gösterilmiştir. Bu algoritmaların karşılaştırmasına ait sonuçlar Şekil 9'da gösterilmektedir.



Şekil 9. NHANES'ten alınmış kan tahlili ile elde edilen pasif verisetinin Kullanıcı Arayüzü ekranındaki sonuçları

Algoritmalarla elde edilen tahmin doğruluğu sonuçları; Şekil 9' da da görüldüğü üzere MLP NN için %97.4, RF için %95.9, LR için %98.3 ve ChOA-MLP NN algoritması için %98.1' dir. Ancak bu sınıflandırma doğruluğuna ait oranlar kullanıcı arayüzüne yüklenen verisetine göre değişkenlik göstermektedir. Ancak her ne kadar burada verisetine göre değişkenlik gösteren sınıflandırma doğruluğu yüzdeleri verilemesede hastalık veri setleri için tahmin/sınıflandırma doğruluğu yüzdesi/oranı değişiklik oranı ∓ 5 olduğu gözlemlenmiştir.

Sınıflandırma/ tahmin doğruluğu yüzdeleri bakımından şekil 9' da görüldüğü gibi sırasıyla LR>ChOA-MLP NN>NN>RF şeklindedir. LR algoritması, ikili sonuçlara ait verisetlerinin sınıflandırılma doğruluğu yüksektir. Ancak veriseti karmaşıklaştıkça (Örneğin, örnek çıktılarının/sonuçlarının Hastalıklı(1), Hastalıklı Değil(0), Eksik Bilgi(2) şeklinde tanımlanması gibi) bu sınıflandırma doğruluğu sonucuna güvenilemez.

ChOA-MLP NN'nin sınıflandırma oranı %98.1 değeri tek başına LR algoritmasının doğruluk oranından daha küçüktür, ancak LR'nin karmaşık verisetinde uygulanamasından dolayı daha güvenilirdir. LR algoritmasına ait olan ROC eğirisinden de görüldüğü gibi sonuçların explonensiyel yükselmemesi sınıflandırma doğruluğuna güvenilmemesi gerektiğinin bir göstergesidir.

RF, sadece verileri sınıflandırmak amacı ile çalışan bir algortima modelidir. Kullanıcı arayüzü içerisine dahil edilmesinin nedeni bu sadece sınıflandırma amacıyla kullanılan algoritma ile oluştrulan bileşik model arasındaki olumlu farkı gösterebilmektir. RF ‘e ait bir grafik yoktur.

SVM, her ne kadar kullanıcı arayüzünde sınıflandırma oranı karşılaştırılmasını görmek amacıyla eklenmese de şekil 10, SVM doğruluk tablosuna ait grafikte sınıflandırma doğruluk oranı yer almaktadır. SVM doğruluk oranı da %98.3 olarak LR gibi yüksek bir sınıflandırma doğruluk oranına sahiptir. Ancak “macro avg / f1-score” kısmında da görüldüğü gibi bu veriseti için bu sınıflandırma doğruluğu %50’ dir.

```
Destek Vektör Makinesi Dogruluk Değeri = 0.9838129496402878
[[1641    0]
 [  27    0]]
              precision    recall  f1-score   support

           1       0.98      1.00      0.99      1641
           2       0.00      0.00      0.00        27

    accuracy                           0.98      1668
   macro avg       0.49      0.50      0.50      1668
weighted avg       0.97      0.98      0.98      1668
```

Şekil 10. SVM sınıflandırma doğruluğu tablosu

NN algoritmasının tek başına doğruluk oranı dahi diğer algoritmalara kıyasla yüksektir. Ancak aşırı uyum sorunu yani belli bir iterasyon sayısından sonra hatanın aynı değerde görülmesi güvenlik sorunudur. Yinede diğer algoritmalara nazaran daha güvenilirdir.

MLP NN Algoritmasının en büyük dezavantajı olan aşırı uyum sorunu ChOA kullanılarak Şekil 9’ da yer alan MLP NN + ChOA MSE grafiğinden de görüldüğü gibi MSE oranı 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.

Bu çalışma her ne kadar kan testine göre yapılmış olsada parametrelerin ayarlanması ile diğer verisetlerinin incelenmesinde de kullanılabilecektir. Ayrıca MLP NN algoritması yapısı gereği diğer algoritmalara nazaran verisetin boyutuna

göre işlem süresi biraz daha uzundur. İşlem süresini kısaltabilmek için MLP NN algoritmasının gizli katmanına gömülü olan ChOAi RF ile birleştirilebilir. Ayrıca tek bir gizli katman ile çalışılmıştır. Kullanıcı arayüzünde MLP NN için katman sayısı seçtirilebilir. Yada başka bir optimizasyon algoritması ile de birleştirilebilir.

V.KAYNAKÇA

MAKALELER

AFRAKHTEH, S., MOSAVİ, M. R., KHİSHE, M., & AYATOLLAHİ, 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., FARİS, H., & MİRJALİLİ, 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

ALZEN, J. L., LANGDON, L. S., & OTERO, V. K. (2018). A logistic regression investigation of the relationship between the learning assistant model and failure rates in introductory STEM courses. **International Journal of STEM Education**, 5(1). https://doi.org/10.1186/S40594-018-0152-1

BREİMAN, L. (2001). Random forests. **Machine Learning**, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

GOGOU, G., MAGLAVERAS, N., AMBROSİADOU, B. V., GOULİS, D., & PAPPAS, C. (2001). A neural network approach in diabetes management by insulin administration. **In Journal of Medical Systems**, 25(2), 119-131. https://doi.org/10.1023/A:1005672631019

GOMES, G. S. D. S., & LUDERMİR, T. B. (2013). Optimization of the weights and asymmetric activation function family of neural network for time series forecasting. **Expert Systems with Applications**, 40(16), 6438–6446. https://doi.org/10.1016/j.eswa.2013.05.053

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

IBRAHİM, S., ROZAN, M. H. C., & SABRİ, N. (2019). Comparative analysis of support vector machine (SVM) and convolutional neural network (CNN) for white blood cells' classification. **International Journal of Advanced Trends in Computer Science and Engineering**, 8(1.3 S1), 394–399. https://doi.org/10.30534/ijatcse/2019/6981.32019

KAZEROUNİ, F., BAYANİ, A., ASADİ, F., SAEİDİ, L., PARVİZİ, N., & MANSOORİ, Z. (2020). Type2 diabetes mellitus prediction using data mining algorithms based on the long-noncoding RNAs expression: A comparison of four data mining approaches. **BMC Bioinformatics**, 21(1). https://doi.org/10.1186/s12859-020-03719-8

KHİSHE, M., & MOSAVİ, 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

KHİSHE, M., & MOSAVİ, M. R. (2020b). Chimp optimization algorithm. **Expert Systems with Applications**, 149. https://doi.org/10.1016/j.eswa.2020.113338

KORKMAZ, M., GÜNEY, S., & YÜKSEL YİĞİTER, Ş. (2012). The importance of logistic regression implementations in the turkish livestock sector and logistic regression implementations/fields. **Journal of the Faculty of Agriculture of Harran University**, 16(2), 25–36. https://app.trdizin.gov.tr/makale/TVRjeE56UTJOZz09/the-importance-of-logistic-regression-implementations-in-the-turkish-livestock-sector-and-logistic-regression-implementations-fields

LİVİNGSTONE, D. J., MANALLACK, D. T., & TETKO, I. V. (1997). Data modelling with neural networks: Advantages and limitations. **Journal of Computer-Aided Molecular Design**, 11(2), 135–142. https://doi.org/10.1023/A:1008074223811

MAİELLARO, P., COZZOLONGO, R., & MARİNO, 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

MATHİSON, B. A., KOHAN, J. L., WALKER, J. F., SMİTH, R. B., ARDON, O., ARDON, O., COUTURİER, M. R., & COUTURİER, M. R. (2020). Detection of intestinal protozoa in trichrome-stained stool specimens by use of a deep convolutional neural network. **Journal of Clinical Microbiology**, 58(6). https://doi.org/10.1128/JCM.02053-19

MOSAVİ, M. R., KHİSHE, M., NASERİ, M. J., PARVİZİ, 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., AEİNFAR, M., AEİNFAR, V., & HAYATİ, 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. https://www.google.com/search?q=A+New+Method+for+Diagnosis+and+Predicting+Blood+Disorder+and+Cancer+Using+Artificial+Intelligence+%28Artificial+Neural+Networks%29%2C+IJHOSCR&ei=R35QYcWEAaGB9u8PrJCFKA&oq=A+New+Method+for+Diagnosis+and+Predicting+Blood+Diso

SAAVEDRA-GARCÍA, M., MATABUENA, M., MONTERO-SEOANE, A., & FERNÁNDEZ-ROMERO, J. J. (2019). A new approach to study the relative age effect with the use of additive logistic regression models: A case of study of FIFA football tournaments (1908-2012). **PLoS ONE**, 14(7). https://doi.org/10.1371/journal.pone.0219757

SHAHMORADİ, L., SAFDARİ, R., MİRHOSSEİNİ, M. M., ARJİ, 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. https://www.google.com/search?q=Predicting+Risk+of+Acute+Appendicitis%3A+A+Comparison+of+Artificial+Neural+Network+and+Logistic+Regression+Models%2C+Acta+medica+Iranica+%2C&ei=3XlQYd_UMc-0kwWY-

6vIBA&oq=Predicting+Risk+of+Acute+Appendicitis%3A+A+Comparison

STANFORD, C. B., GOODALL, J., WALLİS, J., MPONGO, E., WALLİS, J., & GOODALL, J. (1994). Hunting decisions in wild chimpanzees. **Behaviour**, 131(1–2), 1–18. https://doi.org/10.1163/156853994X00181

SU, X., XU, Y., TAN, Z., WANG, X., YANG, P., SU, Y., JİANG, Y., QİN, S., & SHANG, L. (2020). Prediction for cardiovascular diseases based on laboratory data: An analysis of random forest model. **Journal of Clinical Laboratory Analysis**, 34(9). https://doi.org/10.1002/jcla.23421

VECCİ, L., PİAZZA, F., & UNCİNİ, A. (1998). Learning and approximation capabilities of adaptive spline activation function neural networks. **Neural Networks**, 11(2), 259–270. https://doi.org/10.1016/S0893-6080(97)00118-4

WONGVİBULSİN, S., WU, K. C., & ZEGER, S. L. (2019). Clinical risk prediction with random forests for survival, longitudinal, and multivariate (RF-SLAM) data analysis. **BMC Medical Research Methodology**, 20(1). https://doi.org/10.1186/s12874-019-0863-0

YANG, H. S., HOU, Y., VASOVİC, L. V, STEEL, P. A. D., CHADBURN, A., RACİNE-BRZOSTEK, S. E., VELU, P., CUSHİNG, M. M., LODA, M., KAUSHAL, R., ZHAO, Z., & WANG, F. (2020). Routine Laboratory Blood Tests Predict SARS-CoV-2 Infection Using Machine Learning. **Clinical Chemistry**, 66(11), 1396–1404. https://doi.org/10.1093/CLINCHEM/HVAA200

YAO, Y., CİFUENTES, J., ZHENG, B., & YAN, M. (2019). Computer algorithm can match physicians' decisions about blood transfusions. **Journal of Translational Medicine**, 17(1). https://doi.org/10.1186/s12967-019-2085-y

YILMAZ, Z., & BOZKURT, M. R. (2012). Determination of women iron deficiency anemia using neural networks. **Journal of Medical Systems**, 36(5), 2941–2945. https://doi.org/10.1007/s10916-011-9772-4

YU, W., LİU, T., VALDEZ, R., GWİNN, M., & KHOURY, M. J. (2010). Application of support vector machine modeling for prediction of common diseases: The case of diabetes and pre-diabetes. **BMC Medical Informatics and Decision Making**, 10(1), 1–7. https://doi.org/10.1186/1472-6947-10-16

ZİNİ, G. (2005). Artificial intelligence in hematology. **Hematology**, 10(5), 393–400. https://doi.org/10.1080/10245330410001727055

KİTAPLAR

GÜRSAKAL, N. (Ed.). (2017). **Makine Öğrenmesi & Derin Öğrenme**. Dora Yayınları. Bursa(Basım Yeri). 50-380.

SİNAN, U. (2021). **Makine öğrenmesi teorik yönleri ve python uygulamaları ile bir yapay zeka ekolü** (Atalay Mat). Nobel Akademik Yayıncılık. 1-265.

KONFERANSLAR

KARAMİZADEH, S., ABDULLAH, S. M., HALİMİ, M., SHAYAN, J., & RAJABİ, M. J. (2014). Advantage and drawback of support vector machine functionality. **I4CT 2014 - 1st International Conference on Computer**, **Communications, and Control Technology, Proceedings,** 63–65. https://doi.org/10.1109/I4CT.2014.6914146

ÖZGEÇMİŞ

Ad-Soyad **:** Büşranur Güdar

ÖĞRENİM DURUMU

- **Lisans :** 2015, Kocaeli Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği
- **Yükseklisans :** 2021, İstanbul Aydın Üniversitesi, Mekatronik Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- İlhan R., Güdar B., 2021, Yapay Sinir Ağları Kullanarak Kan Testi Sonuçlarının Sınıflandırılması ve Kullanıcı Ara Yüzünün Geliştirmesi, **International Symposium on Multidisciplinary Studies and Innovative Technologies**, 21-23 Ekim 2021