



Research Paper
Journal of Smart Systems Research (JOINSSR) 3(2), 81-96, 2022
Received: 14-September-2022 Accepted: 22-October-2022



Yapay Sinir Ağları ile Hava Durumu Tahmini ve Çok Katmanlı Ağ Modeli için LabVIEW Arayüz Tasarımı

Serdar ÖZKAYA* , Gökhan ATALI² 

¹ Mekatronik Mühendisliği Bölümü, Lisansüstü Eğitim Enstitüsü, Sakarya Uygulamalı Bilimler Üniversitesi, Türkiye. y195010056@subu.edu.tr

² Mekatronik Mühendisliği Bölümü, Teknoloji Fakültesi, Sakarya Uygulamalı Bilimler Üniversitesi, Türkiye. gatali@subu.edu.tr

ÖZ

Yapay sinir ağları (YSA), insanın biyolojik sinir hücrelerinin çalışmasının taklit edilmesi sonucunda tahmin, sınıflandırma, optimizasyon, sinyal işleme, doğrusal olmayan sistemlerin modellenmesi gibi birçok alanda karşımıza çıkan problemlere çözüm üreten bir yöntemdir. YSA'nın tarihsel gelişiminde önce tek katmanlı algılayıcılar kullanılarak sadece doğrusal problemlere çözüm getirilmiştir. Doğrusal olmayan problemlerin çözümü için çok katmanlı algılayıcı (ÇKA) ağ modelleri geliştirilmiştir. Bir YSA'nın eğitiminde girdi ve çıktı değerlerinin birlikte verilmesi ile gerçekleşen öğrenme metodu gözetimli öğrenme metodu şeklinde adlandırılır. YSA'nın ÇKA modelinde gözetimli öğrenme metodu kullanılmaktadır. Bu çalışmada; doğrusal olmayan problemlerin çözümü için YSA'nın geri yayımlı ÇKA modeli kullanılarak LabVIEW ortamında bir ara yüz tasarlanmıştır. Gerçekleştirilen tasarım ile hava durumu tahmin problemine çözüm getirilmiştir. Tasarımda katmanlardaki nöron sayısı, öğrenme katsayısı, momentum katsayısı, iterasyon sayısı, başlangıç ağırlıkları gibi parametreler değiştirilerek ağın eğitilmesi sağlanmıştır ve test edilerek ağın performans ölçümü gerçekleştirilmiştir. Yapılan test işleminin sonucunda test verilerinin beklenen çıktı değerleri ile YSA'nın ürettiği değerler karşılaştırıldığında başarılı bir sonuç elde edilmiştir. Çalışmada Konya iline ait 45 günlük nem oranı (%), görüş mesafesi (km), basınç (mb), çığ noktası (°C), sıcaklık (°C) bilgilerini içeren bir veri seti kullanılmıştır.

Anahtar Kelimeler: Yapay Sinir Ağları, Çok Katmanlı Ağ, LabVIEW, Genelleştirilmiş Delta Öğrenme Metodu, Hava Tahmini

* Sorumlu yazarın e-posta adresi: y195010056@subu.edu.tr

Weather Forecasting with Artificial Neural Networks and LabVIEW Interface Design for Multi-Layer Network Model

ABSTRACT

Artificial neural networks (ANN) are a method that produces solutions to many problems by imitating the work of human biological nerve cells, problems such as prediction, classification, optimization, signal processing, and modeling of nonlinear systems. In the early stages of the historical development of ANN, only linear problems were solved by using single-layer perceptrons. Multilayer perceptron models have been developed to solve nonlinear problems. In the training of an ANN, the learning method that takes place by giving the input and output values together is called the supervised learning method. In the multilayer network model, the supervised learning method is used. In this study, for the solution of nonlinear problems, an interface is designed in the LabVIEW environment by using the Back Propagation model from ANN. With the emerging design, the weather forecast problem has been solved. By changing the parameters such as the number of neurons in the layers, the learning coefficient, the momentum coefficient, the number of iterations, and the initial weights, the network is trained, and the performance of the network is measured by testing it. As a result of the test process, a successful result was obtained by comparing the expected output values of the test data with the values produced by ANN. In the study, a dataset containing 45 daily humidity (%), visibility distance (km), pressure (mb), dew point (°C), and temperature (°C) information of Konya province was used

Keywords: Artificial Neural Networks (ANN), Multilayer Networks, LabVIEW, Generalized Delta Learning Method, Prediction Weather Forecasting

1 Giriş

İnsanlar genel olarak gördükleri ve yaşadıkları olaylar karşısında davranışlarında değişimler göstererek öğrenme işlemini gerçekleştirirler. Yaşadıkları her gün yeni tecrübeler elde ederler. Makine öğrenimi ile insanların tecrübelerini kullanarak doğru davranış gösterme özellikleri, makinelerle kazandırılmak istenmektedir.

Makine öğreniminde yaygın olarak kullanılan yöntemlerden biri yapay sinir ağlarıdır. YSA matematiksel olarak net çözüm getirilemeyen yani doğrusal olmayan problemlerde yaygın olarak kullanılmaktadır. YSA; tıp, mühendislik, uzay, üretim, ekonomi gibi birçok alanda görüntü ve sinyal işleme, tahmin, optimizasyon, sınıflandırma gibi farklı uygulamalarla kullanılmaktadır.

Bu konu hakkında literatür incelendiğinde; Bukhari ve arkadaşları çalışmalarında diyabet hastalığını etkili bir şekilde tahmin etmek için yirmi gizli katmandan oluşan bir yapay sinir ağı modeli geliştirmişlerdir [1]. Jiang ve Gao yapmış oldukları çalışmada yüksek hızlı tren ray profillerinin aşınımını azaltmak amacıyla tasarımlarında optimizasyon sağlamak için genetik algoritma ile bağlantılı yapay sinir ağı modeli oluşturmuşlardır [2]. Majoros ve arkadaşları PhysioNet üzerinden açık erişimli EEG (elektroensefalografi) veri setini kullanarak bir yapay sinir ağı modeli oluşturmuşlar ve bu sinyalleri deneklerin hayallerinde canlandırdıkları davranış biçimlerine göre sınıflandırmışlardır [3]. Thakur ve arkadaşları yapmış oldukları bir çalışmada Hindistan'ın Ocak 2000 ile Aralık 2012 arasındaki tarihsel aylık ekonomik verilerini analiz ederek enflasyon tahmini için geri yayımlı bir yapay sinir ağı modeli oluşturmuşlardır [4]. Atalı ve arkadaşları çalışmalarında tahribat işlemi uyguladıkları görüntüleri ve orijinal hallerini bir takım görüntü işleme teknikleri uygulayarak eğitim ve test verisi şeklinde bir yapay sinir ağına vermişlerdir. Tahribatın derecesine göre ağın ürettiği çıktıları ve başarı düzeylerini karşılaştırmışlardır [5]. Akyüz ve arkadaşları çalışmalarında Antalya'ya ait 2000-2016 yılları arasında

ölçülen gerçek aylık ortalama buhar basıncı, aylık ortalama nisbi nem, ilgili ay ve yıl verilerini kullanarak geri beslemeli bir yapay sinir ağı modeli ile ortalama hava sıcaklığını %99 başarı oranıyla tahmin etmişlerdir [6]. Güç yapmış olduğu yüksek lisan tez çalışmasında Bilecik ilinin 2013 yılına ait saatlik ortalama rüzgar şiddeti, hava basıncı, su buharı basıncı, nispi nem ve hava sıcaklığı değerlerini barındıran 1794 adet veriyi tasarladığı YSA modelinde kullanarak %91 başarı oranı ile hava sıcaklığı tahmini yapmıştır. Verilerin %80'lik kısmını eğitim aşamasında kullanmıştır [7]. Yavuz ve arkadaşı çalışmalarında 1990 ile 2011 yılları arasındaki Meteoroloji Genel Müdürlüğünden Adana ilinin günlük hava durumu verileri alınarak elde edilen veri seti üzerinde çeşitli normalizasyon tekniklerini Matlab programı üzerinde tasarlanan bir yapay sinir ağı üzerinde kullanarak Ortalama Mutlak Yüzde Hata (MAPE, Mean Absolute Percentage Error) ve Hata Kareler Ortalaması (MSE, Mean Square Error) istatistiği yardımıyla değerlendirerek karşılaştırmışlardır [8]. Aydın ve arkadaşı çalışmalarında yerel saat zamanı ile her 15 dakika da Avrupa için uydu görüntüsü veren Eumetsat verilerinden elde edilen Kasım 2012'ye ait günlük 96 toplamda ise 2880 kızılötesi uydu görüntülerini dalgacık dönüşümü ve temel bileşenler analizi yöntemlerinden geçirerek yapay sinir ağı ve destek vektör makineleri algoritmaları ile sınıflandırmaya tabi tutmuşlardır. Yapay sinir ağı ile sınıflandırma sonucunda %84 ve destek vektör makineleri ile sınıflandırma sonucunda %93 oranında başarı elde etmişlerdir [9].

Bu çalışmada LabVIEW programı kullanılarak tahmin ve sınıflandırma problemlerinin uygulanabileceği görsel bir yazılım geliştirilmiştir. Yazılımda ÇKA modeli, Genelleştirilmiş Delta Algoritması (Geri Yayılım Algoritması) kullanılarak tasarlanmıştır. Yazılımda ağın yapısı, katmanlardaki nöron sayısı, öğrenme oranı, momentum katsayısı, iterasyon sayısı, tolerans miktarı, başlangıç ağırlıklarının seçimi kullanıcı tarafından belirlenmiştir. Eğitim esnasında oluşan hatanın grafiği gözlemlenmiştir ve test işleminden sonra ağın performansı ölçülerek beklenen değerler ile ağın ürettiği değerler karşılaştırılmıştır.

2 Yöntem ve Metot

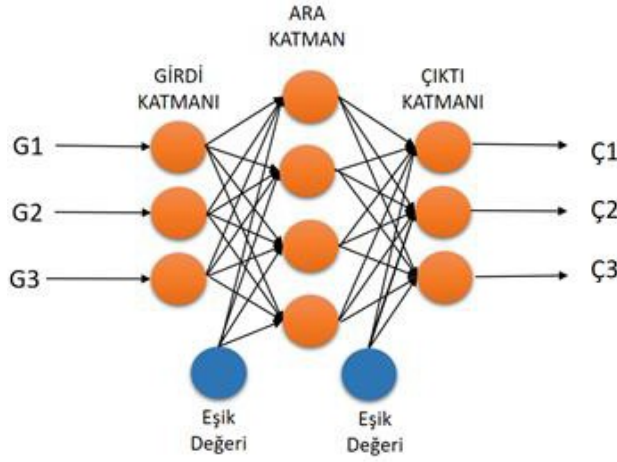
2.1 Yapay Sinir Ağları

YSA, insanın öğrenme özelliğinin taklit edilerek yeni bilgiler üretebilen, yeni bilgiler türetebilen, keşfedilme gibi yeteneklere sahip ve tüm bunları tek başına gerçekleştirebilen bilgisayar sistemleridir [10]. Paralel çalışma yapısına sahip basit işlemler gerçekleştiren bir işlemcidir ve deneyim ile elde edilmiş bilgiyi depolayan ve kullanılabilir kılmak için nöronlar arası bağlantı ağırlıklarını sürekli yineleyerek öğrenmeyi gerçekleştiren bir yapıdır [11]. Nöronlar arasında bulunan bağlantı ağırlıklarının sayısal bir değeri vardır ve bu değer bir nöronun girişinin gücünü veya önemini ifade eder. Bu ağırlıklar YSA da uzun süreli belleğin temel birimini oluşturmaktadır. Bir sinir ağı bu ağırlıkların değerini, sürekli olarak yineleyerek öğrenme işlemini gerçekleştirir. Bir YSA oluşturmak için ağda kullanılacak nöron sayısına, nöronlar arası bağlantıların nasıl olacağına yani ağın yapısına, hangi öğrenme algoritmasının kullanılacağına karar verilmesi gerekmektedir. Son olarak ağın eğitimi için ilk bağlantı ağırlıkları belirlenir ve ağın eğitimi başlatılır [12].

Bu çalışmada Konya ilinin 15 Haziran 2017 ve 29 Ağustos 2017 tarihleri arasındaki 45 günlük hava durumu verileri kullanılmıştır. Veri seti 15 günlük hava durumu verilerini içeren bir internet sitesinden alınan bilgilerle hazırlanmıştır [13]. Girdi verileri Nem Oranı (%), Görüş Mesafesi (km), Basınç (mb), Çiğ Noktası (°C) başlığı altında 4 adet öznitelik değerlerinden ve çıktı verileri Sıcaklık (°C) başlığı altında 1 adet öznitelik değerlerinden oluşmaktadır.

2.2 Çok Katmanlı Algılayıcılar

YSA'nın girdi ve çıktıları arasındaki ilişkilerin doğrusal olmadığı durumlarda kullanılan ÇKA, bilgi girişinin yapıldığı girdi katmanı, bir veya birden fazla ara (gizli) katman ve bir çıktı katmanından oluşmaktadır [14]. Şekil 1'de ÇKA modelinin yapısı gösterilmiştir.



Şekil 1: ÇKA modelinin yapısı

Girdi katmanında herhangi bir bilgi işleme olmaz. Gelen bilgiler olduğu gibi bir sonraki katmana aktarılır. Giriş ve çıkış katmanlardaki nöron sayısı problemin doğasına göre belirlenirken, gizli katmandaki nöron sayısı için herhangi bir yöntem bulunmamaktadır. Bu yüzden deneme yanılma yoluyla belirlenmektedir [15]. Her nöronun bir girdisi ve bir çıktısı vardır. Çıktılar bir sonraki katmanda bütün nöronlara gönderilir. Ara katmanda girdi katmanından gelen bilgiler işlenerek çıktı katmanına iletilir. Ara katmandaki nöronlar çıktı katmanındaki bütün nöronlar ile bağlantılıdır. Çıktı katmanı ara katmandan gelen bilgileri işleyerek ağıın girdi katmanından verilen girdilere karşılık ağıın ürettiği çıktıları dış dünyaya gönderir.

2.3 ÇKA Modelinin Öğrenme Metodu

ÇKA gözetimli öğrenme stratejisine göre çalışır. Yani ağıın eğitiminde hem girdiler hem de o girdilere karşılık gelen çıktılar birlikte verilir ve ağıın ürettiği çıktı değerleri ile doğru sonuçlar karşılaştırılarak yeni ağıırlık değerleri belirlenir [16]. ÇKA modelinin öğrenme kuralı en küçük kareler yöntemine dayalı Delta Öğrenme Kuralının genelleştirilmiş halidir. Genelleştirilmiş Delta Kuralında öğrenme ileri ve geri hesaplama olmak üzere iki aşamada gerçekleşir. İleri hesaplamada ağıın çıktısı hesaplanır. Geri hesaplamada ise ağıın ağıırlık değerleri değiştirilir.

İleri hesaplama safhasında bilgi işleme, eğitim setindeki bir örneğın girdi katmanından ağıa verilmesiyle başlar. Daha öncede belirtildiği gibi girdi katmanında herhangi bir bilgi işleme olmaz. Gelen girdi olduğu gibi ara katmana gönderilir. Girdi katmanındaki bir proses elemanın çıktısı ζ_k^i Denklem 1'de gösterildiği gibi hesaplanır.

$$\zeta_k^i = G_k \quad (1)$$

Burada G_k girdi katmanındaki k. proses elemanın girdi değerini göstermektedir. Ara katmanındaki bir proses elemanın net girdisi NET_j^a Denklem 2'de gösterildiği gibi hesaplanır.

$$NET_j^a = \sum_{k=1}^a A_{kj} \zeta_k^i \quad (2)$$

Burada A_{kj} k. girdi katmanı proses elemanını j. ara katmanı proses elemanına bağlayan bağlantının ağırlık değerini göstermektedir. Ara katmandaki bir proses elemanın çıktısı gelen net girdinin bir aktivasyon fonksiyonundan geçirilmesi ile hesaplanır. Aktivasyon fonksiyonunun sigmoid seçilmesi durumunda j. ara katman proses elemanının çıktısı ζ_j^a Denklem 3'te gösterildiği gibi hesaplanmaktadır.

$$\zeta_j^a = \frac{1}{1 + e^{-(NET_j^a + \beta_j^a)}} \quad (3)$$

Burada β_j^a j. ara katman proses elemanına bağlanan eşik değerinin ağırlık değerini göstermektedir. Çıktı katman proses elemanlarının çıktı değerleri, ara katman proses elemanlarında olduğu gibi önce net girdi değerlerinin hesaplanması ve bir aktivasyon fonksiyonundan geçirilmesi elde edilir. Çıktı katmanının çıktı değerlerinin bulunması ile ileri hesaplama safhası tamamlanmış olur.

Geri hesaplama safhasında amaç; ağırlık çıktı değerleri ile olması gereken değerlerin farkı alınarak elde edilen hata değerlerini, bağlantı ağırlıklarına yayarak azaltmaktır. Çıktı katmanındaki m. proses elemanının hata değeri E_m Denklem 4'te gösterildiği gibi hesaplanır.

$$E_m = B_m - \zeta_m \quad (4)$$

Burada ζ_m m. proses elemanının çıktı değerini, B_m ise olması gereken değerini ifade etmektedir. Çıktı katmanındaki Toplam Hata (TH) değeri ise Denklem 5'te gösterildiği gibi hesaplanır.

$$TH = \frac{1}{2} \sum_m E_m^2 \quad (5)$$

Toplam hatayı en aza indirmek için yapılan ağırlık değişiminde iki durum söz konusudur. Bunlar;

- Ara katman ile çıktı katman arasındaki ağırlıkların değiştirilmesi
- Ara katmanlar arası veya ara katman ile girdi katman arasındaki ağırlıkların değiştirilmesidir.

Ara katman ile çıktı katman arasındaki ağırlıkların değiştirilmesi için öncelikle değişim miktarının hesaplanması gerekir. Ara katmandaki j. proses elemanı ile çıktı katmanındaki m. proses elemanı arasındaki bağlantının ağırlığındaki değişim miktarı ΔA_{jm}^a 'in t. iterasyondaki değeri Denklem 6'da gösterildiği hesaplanır.

$$\Delta A_{jm}^a(t) = \sigma \delta_m \zeta_j^a + \alpha \Delta A_{jm}^a(t-1) \quad (6)$$

Burada σ öğrenme katsayısını, α ise moment sayısını göstermektedir. δ_m ise aktivasyon fonksiyonun türevi ile m. çıktı değerine ait hata miktarının çarpımıyla elde edilir. δ_m değerinin hesaplanmasında Denklem 7'de sunulan formül kullanılmaktadır.

$$\delta_m = f'(NET) E_m \quad (7)$$

Burada aktivasyon fonksiyonu olarak sigmoid fonksiyonunun kullanılması durumunda δ_m değerine ait formül Denklem 8'de gösterilen hali alacaktır.

$$\delta_m = \zeta_m(1 - \zeta_m)E_m \quad (8)$$

Bu durumda ara katman ile çıktı katman arasındaki ağırlıkların t . iterasyondaki değeri $A_{jm}^a(t)$ Denklem 9'da gösterildiği gibi hesaplanır.

$$A_{jm}^a(t) = A_{jm}^a(t-1) + \Delta A_{jm}^a(t) \quad (9)$$

Eşik değeri bağlantı ağırlığının da benzer şekilde değiştirilmesi gerekir. Eşik değeri ile ilgili işlemler ağı yapısında eşik değeri kullanılıyorsa yapılmalıdır. Ara katman ile çıktı katman arası eşik değerinin bağlantı ağırlığındaki değişim miktarı $\Delta\beta_m^c$ t . iterasyondaki değeri Denklem 10'da gösterildiği gibi hesaplanır.

$$\Delta\beta_m^c(t) = \sigma\delta_m + \alpha\Delta\beta_m^c(t-1) \quad (10)$$

Ara katman çıktı katman arası eşik değeri bağlantı ağırlığının t . iterasyondaki yeni değeri Denklem 11'de gösterildiği gibidir.

$$\beta_m^c(t) = \beta_m^c(t-1) + \Delta\beta_m^c(t) \quad (11)$$

Ara katmanlar arası veya ara katman ile girdi katman arasındaki ağırlıkların değişim miktarı da benzer şekilde Denklem 12'de gösterildiği gibi hesaplanmaktadır.

$$\Delta A_{kj}^i(t) = \sigma\delta_j^a\zeta_k^i + \alpha\Delta A_{kj}^i(t-1) \quad (12)$$

Buradaki hata terimi δ_j^a aktivasyon fonksiyonunun sigmoid fonksiyonu seçilmesi durumunda Denklem 13'te gösterildiği gibi hesaplanmaktadır.

$$\delta_j^a = \zeta_j^a(1 - \zeta_j^a)\sum_m \delta_m A_{jm}^a \quad (13)$$

Bu durumda ağırlıkların t . iterasyondaki yeni değeri Denklem 14'te gösterildiği gibi hesaplanır.

$$A_{kj}^i(t) = A_{kj}^i(t-1) + \Delta A_{kj}^i(t) \quad (14)$$

Ara katmanlar arası veya girdi katman ara katman arası eşik değeri bağlantı ağırlığı değişiminin de gerçekleşmesi gerekmektedir. Eşik değeri bağlantı ağırlığının değişim miktarı β_j^a t . iterasyondaki değeri Denklem 15'te gösterildiği gibi hesaplanır.

$$\Delta\beta_j^a(t) = \sigma\delta_j^a + \alpha\Delta\beta_j^a(t-1) \quad (15)$$

Eşik değeri bağlantı ağırlığının t . iterasyondaki yeni değeri Denklem 16'da gösterildiği gibi hesaplanır.

$$\beta_j^a(t) = \beta_j^a(t-1) + \Delta\beta_j^a(t) \quad (16)$$

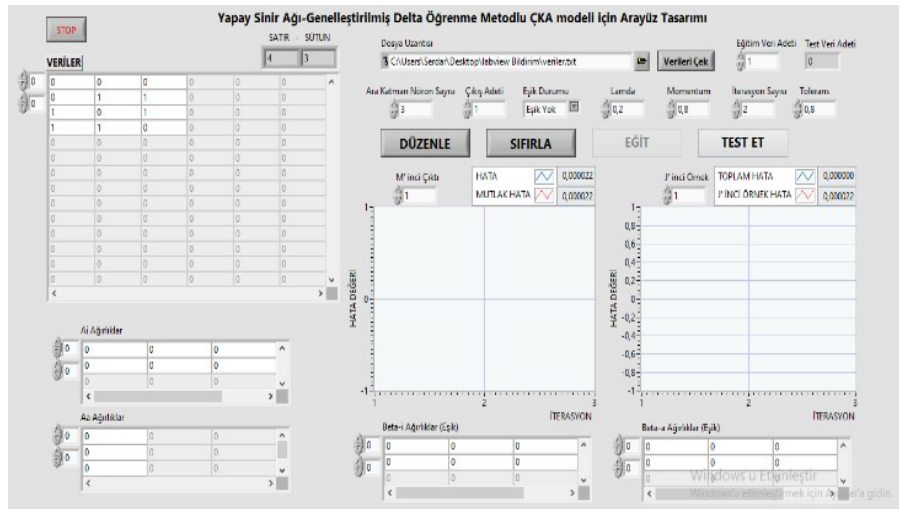
Ağırlık değerlerinin değiştirilmesi ile bir iterasyondaki ileri ve geri hesaplama aşamaları tamamlanmış olur. İkinci bir örnek verilmesiyle sonraki iterasyon başlar ve aynı işlemler öğrenme işlemi gerçekleşinceye kadar tekrarlanır.

3 Bulgular ve Tartışma

Bu çalışmada Konya iline ait 15 Haziran 2017 ve 29 Temmuz 2017 tarihleri arasındaki nem oranı (%), görüş mesafesi (km), basınç (mb), çiğ noktası ($^{\circ}\text{C}$), sıcaklık ($^{\circ}\text{C}$) bilgilerini içeren bir veri seti kullanılarak LabVIEW ortamında tasarlanan bir ara yüz ile geleceğe yönelik hava tahmini uygulaması gerçekleştirilmiştir.

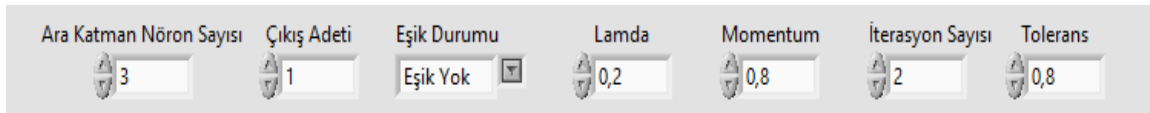
3.1 LabVIEW Kullanıcı Arayüz Tasarımı

Grafiksel gösterimle yazılım yapılan LabVIEW programında öncelikli olarak Front panel üzerine konumlandırılan bilgilerin boyutlandırılmasına ve görünümüne dikkat edilmiştir. Tasarlanan ara yüz Şekil 2’de gösterilmiştir.



Şekil 2: LabVIEW yazılımı front panel görünümü

Tasarlanan ara yüzün dosya uzantısı kısmında YSA’ya verilecek eğitim ve test verilerinin hangi dosyadan yükleneceği seçilmektedir. Yüklenen veri setinde girdi ve çıktı değerleri birlikte olmalıdır. Veri seti aynı zamanda eğitim ve test verilerinin ikisini birden kapsamalıdır. Dosya uzantısının yanında yer alan “Eğitim Veri Adeti” kısmında, yüklenen veri setinin ilk kaç satırının eğitim verisi şeklinde kullanılacağı seçilmektedir. Girilen değerın altında kalan satırlar otomatik olarak test verisi şeklinde kabul edilecektir.



Şekil 3: YSA parametreler

YSA’nın öğrenme ve topolojik yapısını belirleyen parametreleri Şekil 3’te gösterilmiştir. “Ara Katman Nöron Sayısı”, “Çıkış Adeti”, “Eşik Durumu” gibi değerler ağın topolojik yapısını belirlemektedir. “Lamda”, “Momentum”, “İterasyon Sayısı”, “Tolerans” gibi değerler ağın öğrenme parametreleridir. “Ara Katman Nöron Sayısı” kısmında ağın gizli katmanında kaç adet proses elemanın kullanıldığı girilmektedir. “Çıkış Adeti” kısmında ağın kaç adet çıktısı olduğu girilmektedir. Ağda eşik değeri kullanılıp kullanılmaması durumu “Eşik Durumu” kısmından belirtilmektedir. “Lamda” kısmında ağın öğrenme katsayısı, “Momentum” kısmında ise momentum katsayısı girilmektedir. Ağın öğrenme işleminin kaç iterasyonda gerçekleşeceği “İterasyon Sayısı” kısmından belirtilmektedir. Test işlemi

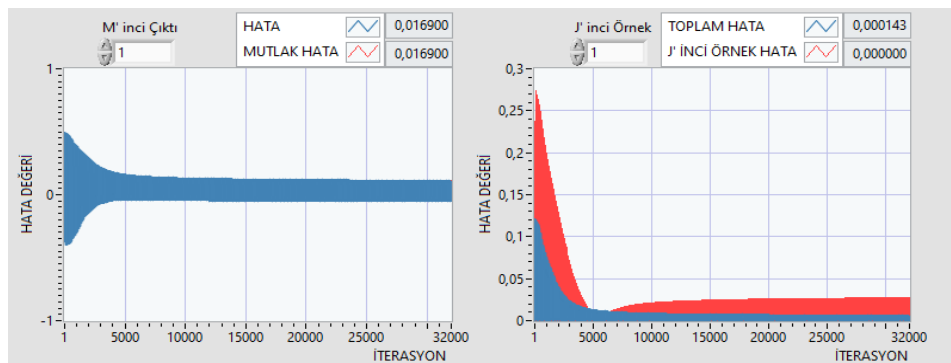
esnasında ağın ürettiği çıktı değerlerinin doğru kabul edilebilmesi için belirlenen tolerans değeri “Tolerans” kısmından girilmektedir.



Şekil 4: Bağlantı Ağırlıkları

Şekil 2’de görüldüğü üzere ağın parametre değerlerinin altında kalan kısımda “DÜZENLE”, “SIFIRLA”, “EĞİT” ve “TEST ET” şeklinde isimlendirilmiş 4 adet buton bulunmaktadır. “DÜZENLE” butonuna tıklanmadan önce ağın veri setinin yüklenmesi ve parametre değerlerinin girilmesi gerekmektedir. Butona tıkladığında yazılım, girilen parametre değerlerine göre ağın yapısını düzenlemektedir. “SIFIRLA” butonuna basıldığında girilen tüm parametre ve ağırlık değerleri sıfırlanmaktadır. “EĞİT” butonu ile ağın eğitimine başlanmakta, “TEST ET” butonu ile ağın test işlemine başlanmaktadır. “EĞİT” butonuna basmadan önce ilk ağırlık değerlerinin girilmesi gerekmektedir.

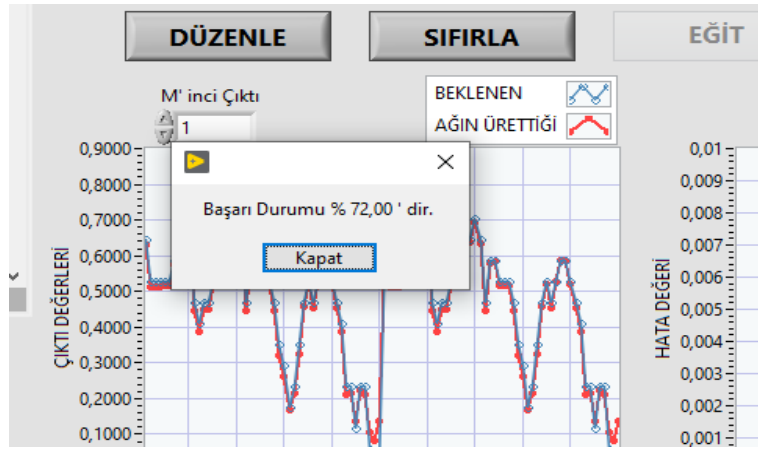
Bağlantı ağırlıklarının girildiği kısım Şekil 4’te gösterilmiştir. Bağlantı ağırlıklarını girmeden önce veri setinin yüklenmesi, çıkış adetinin belirtilmesi ve ağın topolojik yapısını belirleyen parametrelerin girilmesinin ardından düzenle butonuna tıklanması gerekmektedir. Butona tıkladığında yazılım, ağın birtakım parametrelerine ve topolojik yapısına göre hangi katmanda ne kadar bağlantı ağırlığı olacağını Şekil 4’te gösterilen diziler üzerinde otomatik olarak düzenlemektedir. Şekil 4 üzerinde görülen “Ai Ağırlıklar” kısmı giriş katmanı ile ara katman arasındaki, “Aa Ağırlıklar” kısmı ise ara katman ile çıktı katmanı arasındaki ağırlıkları ifade etmektedir.



Şekil 5: Hata grafikleri

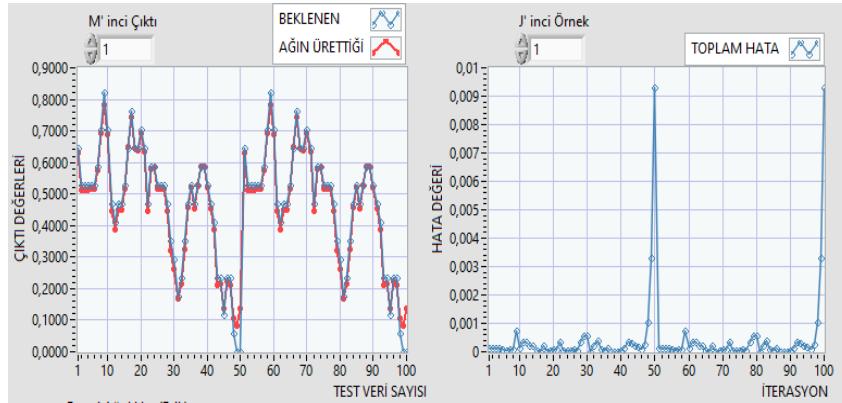
Ağın eğitimi esnasında iterasyonlar boyunca oluşan hataların grafikleri Şekil 5’te gösterilmiştir. Bu grafiklerden soldaki hata ve mutlak hata değerlerinin iterasyonlar boyunca değişimini göstermektedir. Sağdaki grafik ise toplam hata ve her bir örneğin hata değerlerinin iterasyonlar boyunca değişimini göstermektedir. Grafikler üzerine sağ tıkladığında “Export” yazan kısım üzerine gelince açılan alt menüde grafik verilerini resim veya excel ortamında kayıt altına alabileceğimiz seçenekler

bulunmaktadır.



Şekil 6: Dialog Box

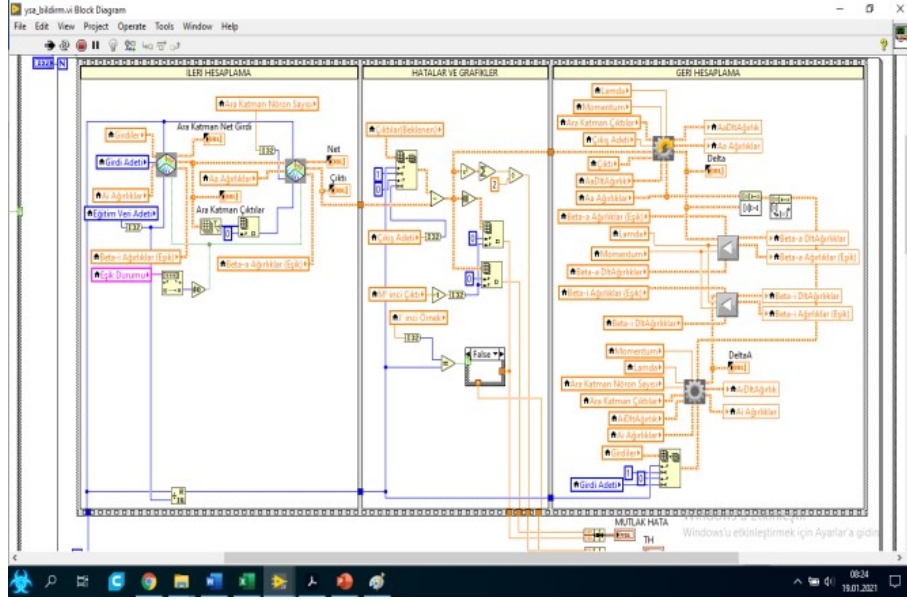
Test işleminin bitmesinin ardından başarı durumunun belirtildiği Dialog Box Şekil 6'da görülmektedir. Şekil üzerinde görülen “Kapat” butonuna tıkladığında Şekil 5'te görülen hata grafiklerinin yerini karşılaştırma ve toplam hata grafikleri almaktadır. Karşılaştırma ve toplam hata grafikleri Şekil 7'de gösterilmektedir.



Şekil 7: Karşılaştırma ve Toplam Hata grafikleri

Yazılımın Blok Diyagram tarafında toplamda 4 adet alt program (SubVI) oluşturulmuştur. Alt programların kullanılmasıyla yazılımın karmaşıklığı giderilmiş ve tekrar eden işlemler için kolaylık sağlanmıştır. Şekil 8'de yazılıma ait Blok Diyagram gösterilmiştir.

Yapay Sinir Ağları ile Hava Durumu Tahmini ve Çok Katmanlı Ağ Modeli için LabVIEW Arayüz Tasarımı



Şekil 8: Blok Diyagram

3.2 Hava Durumu Tahmini için ÇKA Modelinin LabVIEW de Gerçekleştirilmesi

Çalışmada toplam 30 adet eğitim verisi 15 adet test verisi kullanılmıştır. Kullanılan veriler öncelikle Denklem 17’de gösterilen formül kullanılarak Excel ortamında normalize edilmiştir.

$$X_{normal_i} = \frac{X_i - X_{min}}{X_{max} - X_{min}} \quad (17)$$

Veri seti Nem Oranı (%), Görüş Mesafesi (km), Basınç (mb), Çiğ Noktası (°C) başlığı altında dört adet girdi ve Sıcaklık (°C) başlığı altında bir adet çıktıdan oluşmaktadır. Uygulamada kullanılan eğitim verileri Tablo 1’de, test verileri ise Tablo 2’de sunulmuştur.

Tablo 1: Eğitim verisi

Tarih	Girdi Katmanı Değerleri				Çıktı Katmanı Değerleri
	Nem Oranı (%)	Görüş Mesafesi (km)	Basınç (mb)	Çiğ Noktası (°C)	Sıcaklık (°C)
15.06.2017	67,5	31	36	17	27
16.06.2017	70	32	37	17	28
17.06.2017	72,5	33	38	18	29
18.06.2017	75	34	39	17	30
19.06.2017	75	34	39	19	30
20.06.2017	80	36	41	19	32
21.06.2017	77,5	35	40	19	31
22.06.2017	85	38	43	21	34
23.06.2017	82,5	37	42	23	33
24.06.2017	80	36	41	22	32
25.06.2017	82,5	37	42	23	33
26.06.2017	82,5	37	42	19	33

27.06.2017	85	38	43	21	34
28.06.2017	80	36	41	22	32
29.06.2017	70	32	37	22	28
30.06.2017	62,5	29	34	19	25
1.07.2017	75	34	39	20	30
2.07.2017	77,5	35	40	19	31
3.07.2017	82,5	37	42	20	33
4.07.2017	75	34	39	21	30
5.07.2017	72,5	33	38	20	29
6.07.2017	75	34	39	19	30
7.07.2017	82,5	37	42	19	33
8.07.2017	70	32	37	17	28
9.07.2017	72,5	33	38	16	29
10.07.2017	72,5	33	38	16	29
11.07.2017	70	32	37	16	28
12.07.2017	70	32	37	16	28
13.07.2017	82,5	37	42	18	33
14.07.2017	90	40	45	21	36

Tablo 2: Test verisi

Tarih	Girdi Katmanı Değerleri				Çıktı Katmanı Değerleri
	Nem Oranı (%)	Görüş Mesafesi (km)	Basınç (mb)	Çiğ Noktası (°C)	Sıcaklık (°C)
15.07.2017	90	40	45	22	36
16.07.2017	97,5	43	48	23	39
17.07.2017	97,5	43	48	24	39
18.07.2017	97,5	43	48	23	39
19.07.2017	82,5	37	42	22	33
20.07.2017	75	34	39	21	30
21.07.2017	67,5	31	36	20	27
22.07.2017	75	34	39	20	30
23.07.2017	80	36	41	20	32
24.07.2017	80	36	41	19	32
25.07.2017	77,5	35	40	19	31
26.07.2017	80	36	41	20	32
27.07.2017	85	38	43	20	34
28.07.2017	82,5	37	42	22	33
29.07.2017	82,5	37	42	21	33

Ağın topolojik yapısını belirleyen hususlardan biri katmanlardaki proses eleman sayısıdır. Çalışmada kullanılan YSA'nın topolojik yapısı girdi katmanında dört, ara katmanda üç, çıktı katmanında ise bir proses elemanından oluşmaktadır. Proses elemanı net girdilerinin hesaplanmasında ağırlıklı toplam

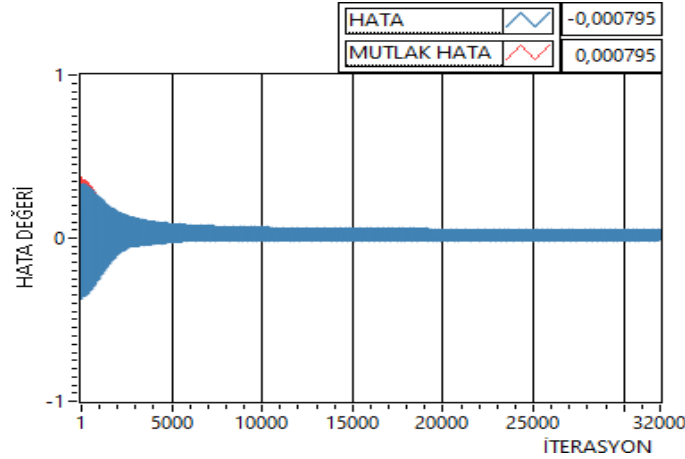
formülü kullanılmıştır. Aktivasyon fonksiyonu olarak Sigmoid fonksiyonu kullanılmıştır. Katmanlar arası bağlantı ağırlıklarına ilave eşik değeri kullanılmamıştır.

Ai Ağırlıklar				
0	0,129952	0,570345	0,631894	▲
0	-0,923123	0,328932	-0,256987	▼
	0,214896	0,525896	-0,325734	
	0,98321	0,84374	0,741369	
< >				
Aa Ağırlıklar				
0	0,164732	0	0	▲
0	0,752624	0	0	▼
	-0,56987	0	0	
< >				

Şekil 9: Başlangıç Ağırlıkları

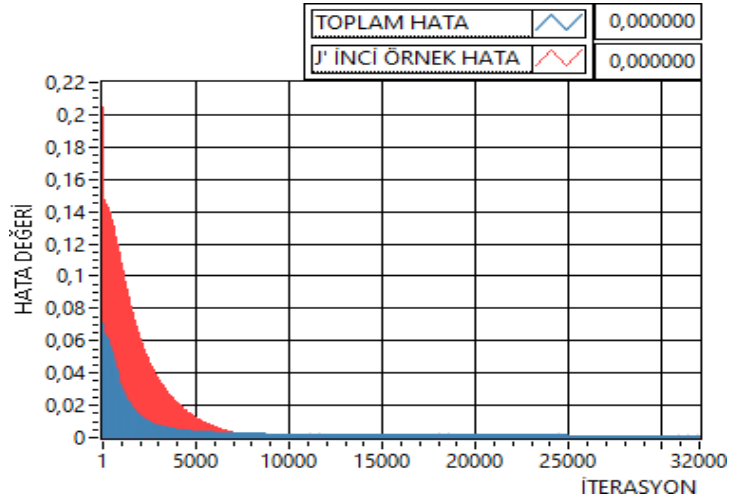
Öğrenme parametrelerinden Lamda değeri 0,8 ve Momentum değeri 0,2 olarak belirlenmiştir. Tolerans değeri 0,03 ve İterasyon sayısı 32000 olarak belirlenmiştir. Ağın ilk başlangıç ağırlık değerleri Şekil 9'da gösterilmiştir.

Eğitimin sonunda elde edilen Hata ve Mutlak Hata değerlerinin iterasyon boyunca değişimini gösteren grafik Şekil 10'da gösterilmiştir. Hata değeri, ağın ürettiği çıktı ile beklenen çıktı değeri arasındaki fark alınarak bulunur. Şekil 10'da gösterilen grafikten de anlaşıldığı üzere hata değeri belirli bir iterasyon boyunca sürekli düşmüş daha sonra sabit bir çizgi izlemeye başlamıştır. Yani hata değişim miktarı azalmıştır.



Şekil 10: Hata ve Mutlak Hata Grafığı

Ağa verilen örneklere ait hata değerleri ve toplam hata değerlerinin iterasyon boyunca değişimini gösteren grafik Şekil 11'de gösterilmiştir.



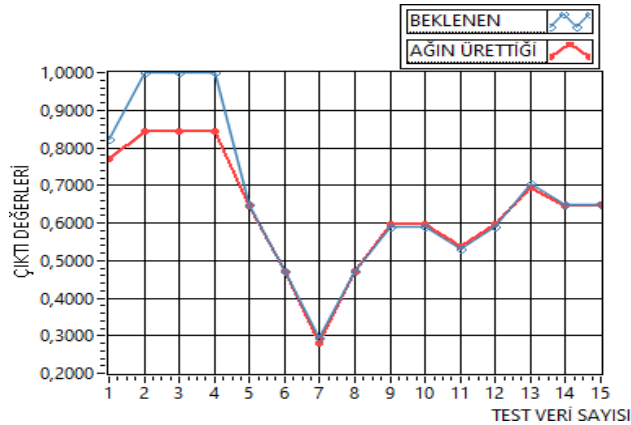
Şekil 11: Örnek ve Toplam Hata Grafiği

Eğitimin sonunda elde edilen yeni ağırlık değerleri Şekil 12’de belirtilmiştir.



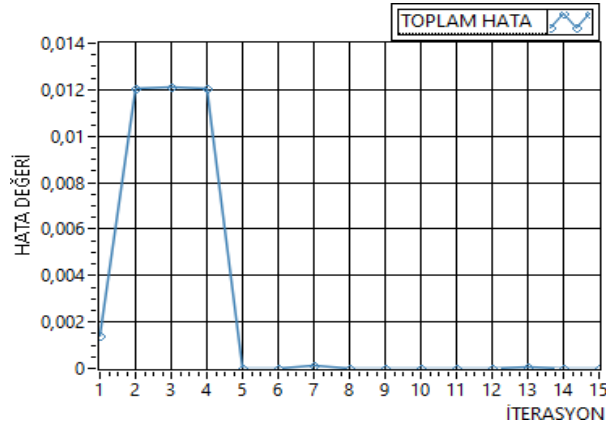
Şekil 12: Eğitim Sonunda Elde Edilen Ağırlıklar

Yeni ağırlıklar kullanılarak yapılan test işlemi sonucunda ağın başarı durumu yüzde 73,33 çıkmıştır. Beklenen değerler ile ağın ürettiği çıktı değerlerinin karşılaştırıldığı grafik Şekil 13’te belirtilmiştir. Grafiği inceleyecek olursak ilk dört değer için aradaki farkın diğerlerine nazaran daha çok olduğu görülmektedir.



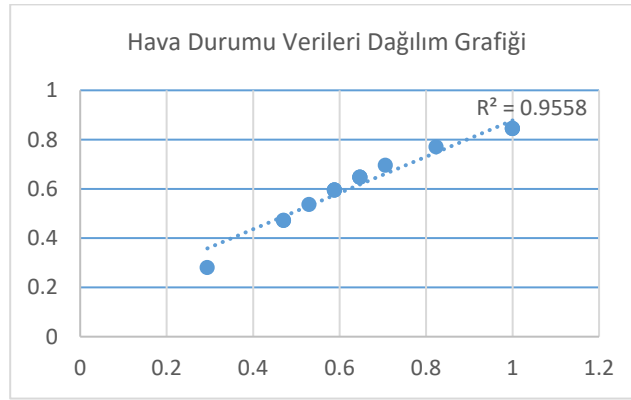
Şekil 13: Karşılaştırma Grafiği

Test verilerine ait toplam hata grafiği Şekil 14’te gösterilmiştir.



Şekil 14: Toplam Hata Grafiği

Ağın ürettiği çıktı değerleri ile beklenen değerler Tablo 3'te gösterilmiştir. Bu veriler kullanılarak Microsoft Excel ortamında dağılım grafiği çizdirilip eğilim çizgisi eklenmesiyle oluşan R^2 değerinin 0,9558 olduğu Şekil 15'te görülmektedir.



Şekil 15: Dağılım Grafiği

Tablo 3: Beklenen ve Ağın Ürettiği Çıktı Değerleri

Beklenen Sıcaklık Değeri (°C)	Normalize Edilmiş Beklenen Çıktı Değeri	Ağın Ürettiği Çıktı Değeri	Ağın Ürettiği Sıcaklık Değeri (°C)	Yüzde Başarı (%)
36	0,8235	0,7702	35,0934	93,52763
39	1	0,8447	36,3599	84,47
39	1	0,8444	36,3548	84,47
39	1	0,8447	36,3599	84,47
33	0,6471	0,6468	32,9956	99,9534
30	0,4706	0,4711	30,0087	99,8935
27	0,2941	0,2804	26,7668	95,3412
30	0,4706	0,4722	30,0274	99,6601
32	0,5882	0,5949	32,1133	98,8603
32	0,5882	0,5955	32,1235	98,7583
31	0,5294	0,5364	31,1188	98,6775
32	0,5882	0,5949	32,1133	98,8603
34	0,7059	0,6955	33,8235	98,5267
33	0,6471	0,6468	32,9956	99,9534
33	0,6471	0,6477	33,0109	99,9078

4 Sonuçlar

LabVIEW programında tasarlanan ara yüz kullanılarak Genelleştirilmiş Delta Öğrenme Kurallı YSA modeli ile doğrusal olmayan hava durum tahmin problemine çözüm getirilmiştir. Tasarlanan ara yüzün kullanıcılara sunduğu imkanlar şöyle listelenebilir.

1. Eğitim ve Test verilerinin ayrı ayrı yüklenmesine gerek duyulmamaktadır. Eğitim veri adedinin girilmesi ile bu ayırımın otomatik olarak yapılması sağlanmıştır.
2. Ana ekranda bulunan hata grafikleri ile eğitim esnasında eş zamanlı olarak hataların iterasyon boyunca değişimlerinin izlenebilmesine olanak sağlanmıştır.
3. Ara yüzde YSA'nın oluşturulması, eğitimi ve test işlemine dair tüm parametreler, grafikler ve diziler tek bir sayfada görülebilecek şekilde tasarlanmıştır.
4. Grafikler üzerine sağ tıklandığında "Export" yazan kısım üzerine gelince açılan alt menüde grafik verilerini resim veya excel ortamında kayıt altına alabileceğimiz seçenekler bulunmaktadır.
5. Geliştirilen tasarım ile doğrusal olmayan sınıflandırma, tahmin gibi problemlerine çözüm getirmeye olanak sağlanmıştır.

LabVIEW üzerinde nesneye dayalı programlama teknikleri kullanılması, ileriki çalışmalarda çeşitli algoritma ve çözüm yöntemlerinin kolaylıkla eklenebilmesine olanak tanır. Farklı öğrenme algoritmaları ve aktivasyon fonksiyonları eklenerek çalışmanın geliştirilmesi mümkündür.

5 Beyanname

5.1 Çalışma sınırları

Çalışmada herhangi bir sınır bulunmamaktadır.

5.2 Teşekkür

Çalışmaya olan katkılarından ötürü emeği geçen Elektrik Elektronik Mühendisi Şevket ÖZKAYA'ya ve Sakarya Uygulamalı Bilimler Üniversitesi Robot Teknolojileri ve Akıllı Sistemler Uygulama Araştırma Merkezine (ROTASAM) de çalışmaya sağladığı katkılardan dolayı teşekkür ederiz.

5.3 Finansman Kaynağı

Çalışmada herhangi bir finans kaynağı kullanılmamıştır.

5.4 Rakip Çıkarlar

Çalışma rakip çıkarlar içermemektedir.

5.5 Yazarların Katkıları

Serdar ÖZKAYA: Makale için fikir geliştirme, sonuçlara ulaşma yöntemlerini planlama, sonuçların açıklanması ve sunumu için sorumluluk alma, literatür taraması sorumluluğunu alma, makalenin tamamının oluşturulmasında katkı sağlama.

Gökhan ATALI: Makale için fikir geliştirme, sonuçlara ulaşma yöntemlerini planlama, sonuçların açıklanması ve sunumu için sorumluluk alma, literatür taraması sorumluluğunu alma, makalenin tamamının oluşturulmasında katkı sağlama.

6 Veri Kullanılabilirliği

Bu çalışmada 15 günlük hava durumu verilerini içeren bir internet sitesinden çekilerek hazırlanmış veri seti kullanılmıştır. [17].

Kaynakça

- [1] M. M. Bukhari, B. F. Alkhamees, S. Hussain, A. Gumaedi, A. Assiri ve S. S. Ullah, "An Improved Artificial Neural Network Model for Effective", *Complexity*, cilt 2021, 2021.
- [2] H. Jiang ve L. Gao, "Optimizing the Rail Profile for High-Speed Railways", *Sustainability*, cilt 12, no. 2, p. 658, 2020.
- [3] T. Majoros, S. Oniga ve Y. Xie, "Motor imagery EEG classification using feedforward neural network", *Annales Mathematicae Et Informaticae*, cilt 53, pp. 235-244, 2021.
- [4] M. Thakur, R. Bhattacharyya ve S. S. Mondal, "Artificial Neural Network Based Model for Forecasting of Inflation in India", *Fuzzy Information And Engineering*, cilt 8, no. 1, pp. 87-100, 2016.
- [5] G. Atalı, S. S. Özkan ve D. Karayel, "Image Damage Analysis With Morphological Image Processing Technique Using Artificial Neural Networks", *Academic Platform - Journal of Engineering and Science*, cilt 4, no. 1, pp. 0-0, 2016.
- [6] A. Ö. Akyüz, K. Kumaş, M. Ayan ve A. Güngör, "Antalya İli Meteorolojik Verileri Yardımıyla Hava Sıcaklığının Yapay Sinir Ağları Metodu ile Tahmini", *Gümüşhane Üniversitesi Fen Bilimleri Dergisi*, cilt 10, no. 1, pp. 146-154, 2020.
- [7] R. Güç, "Bilecik ili için güneş enerjisi analizi ve yapay sinir ağları ile hava sıcaklığı tahmini (Yayımlanmamış yüksek lisans tezi)", *Bilecik: Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü - Tez Koleksiyonu*, 2016.
- [8] S. Yavuz, M. Deveci, "İstatistiksel Normalizasyon Tekniklerinin Yapay Sinir Ağın Performansına Etkisi", *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, cilt 0, no. 40, pp. 167-187, 2015.
- [9] M. Aydın, E. Çelik, "Early detection of hazardous weather conditions in Turkey with satellite images using Support Vector Machines and Artificial Neural Networks", *2013 21st Signal Processing and Communications Applications Conference (SIU)*, Girne, Kıbrıs, 2013.
- [10] E. Öztemel, *Yapay Sinir Ağları*, İstanbul: Papatya Yayıncılık, 2006.
- [11] S. Haykin, *Neural Networks and Learning Machines*, New Jersey: Pearson Education, 2009.
- [12] M. Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems. Second Edition*, England: Pearson Education, 2005.
- [13] "Hava Durumu 15 Günlük", [Çevrimiçi]. Available: <https://www.havadurumu15gunluk.net/>.
- [14] A. Arı ve M. E. Berberler, "Yapay Sinir Ağları ile Tahmin ve Sınıflandırma Problemlerinin Çözümü İçin Arayüz Tasarımı", *Acta Infologica*, cilt 1, no. 2, pp. 55-73, 2017.
- [15] E. N. Dizdaar ve O. Koçar, "İş Sağlığı Ve Güvenliği Yönetim Sistemlerinde Risklerin Yapay Sinir Ağlarıyla Değerlendirilmesi", *Academic Platform - Journal of Engineering and Science*, cilt 6, no. 3, pp. 73-83, 2018.
- [16] D. Kriesel, *A Brief Introduction to Neural Networks.*, 2007.
- [17] M. Kalaycı, "YSA Hava Durumu Tahmin", *GitHub*, Konya, 2018.



© 2020 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).