



A k-ELM approach to the prediction of number of students taking make-up exams

Mustafa Servet Kıran^{1,2*}, Eyüp Sıramkaya³, Engin Eşme²

¹Department of Computer Engineering, Konya Technical University, Konya, Turkey

²Department of Software Engineering, Konya Technical University, Konya, Turkey

³Sarayönü Vocational School, Selçuk University, Konya, Turkey

Highlights:

- Prediction of number of students taken make-up exams using extreme learning machine
- A novel approach named k-ELM based extreme learning machine
- Non-identical ELMs optimized by artificial bee colony algorithm

Keywords:

- Extreme learning machine
- multiple extreme learning machine
- artificial bee colony,
- make-up exam

Article Info:

Research Article
Received: 03.03.2021
Accepted: 31.05.2021

DOI:

10.17341/gazimmfd.890180

Acknowledgement:

The authors wish to thank Konya Technical University for providing data.

Correspondence:

Author: Mustafa Servet Kıran
e-mail: mskiran@ktun.edu.tr
phone: +90 332 205 1542

Graphical/Tabular Abstract

In this study, majority of voting-based extreme learning machines are used for prediction the number of students taken the make-up exams. The non-identical ELMs those join the voting are optimized by artificial bee colony algorithm.

Table A. The ELMs constructed by ABC algorithm for each dataset

Dataset	Params	1.ELM	2.ELM	3.ELM	4.ELM	5.ELM	6.ELM	7.ELM
D1	NS	14	27	38	28	23	50	21
	AF	TanSig	Tribas	Tribas	TanSig	TanSig	Hardlim	Hardlim
D2	NS	17	9	8	34	14	39	29
	AF	Hardlim	TanSig	Hardlim	Hardlim	Radbas	Hardlim	Hardlim
D3	NS	8	17	36	21	36	31	34
	AF	Tribas	Hardlim	Radbas	Sig	TanSig	TanSig	TanSig
D4	NS	42	34	28	43	38	33	36
	AF	Hardlim	TanSig	Sig	Sig	Radbas	Hardlim	Hardlim
D5	NS	48	23	26	50	18	28	12
	AF	TanSig	TanSig	Tribas	Tribas	TanSig	Hardlim	Tribas
D6	NS	49	15	44	35	31	40	44
	AF	TanSig	Radbas	TanSig	Hardlim	TanSig	Sig	Sig
D7	NS	40	19	50	45	35	33	29
	AF	Tribas	Hardlim	Hardlim	Tribas	Hardlim	TanSig	Tribas
D8	NS	15	32	6	27	5	23	42
	AF	Radbas	Hardlim	TanSig	Hardlim	Hardlim	TanSig	Hardlim
D9	NS	27	19	21	32	7	29	28
	AF	Sig	Sig	Hardlim	Sig	Hardlim	Hardlim	Hardlim
D10	NS	17	30	37	50	23	50	19
	AF	Tribas	Sig	Hardlim	Hardlim	TanSig	Sig	Tribas

Purpose: The main objective of this study is to present a novel problem, and novel methodology to solve this problem. The problem is to predict the number of students who fail the course and will join the make-up exams.

Theory and Methods:

The number of students who fail the course should take a make-up exam, but some of them do not join these exams due to internal or external motivations, and this causes waste of resources. Majority of voting-based extreme learning machines have been proposed to solve the problem, and the ELM parameters have been optimized by artificial bee colony algorithm.

Results:

The proposed approach shows better performance than the extreme learning machines in terms of classification accuracy.

Conclusion:

Before the scheduling make-up exams, the number of students who will join the exams should be predicted by the proposed or similar approaches in order to use resources efficiently.



Bütünleme sınavına girecek öğrenci sayısının tahmini için k-ELM yaklaşımı

Mustafa Servet Kıran^{1,2*}, Eyüp Sıramkaya³, Engin Eşme²

¹Bilgisayar Mühendisliği Bölümü, Konya Teknik Üniversitesi, Konya, Türkiye

²Yazılım Mühendisliği Bölümü, Konya Teknik Üniversitesi, Konya, Türkiye

³Sarayönü Meslek Yüksek Okulu, Selçuk Üniversitesi, Konya, Türkiye

ÖNEÇİKANLAR

- Aşırı öğrenme makinesi ile bütünleme sınavına girecek öğrenci sayısı tahmini
- Aşırı öğrenme makine tabanlı yeni bir yaklaşımın (k-ELM) önerilmesi
- Özdeş olmayan ELM'lerin yapay arı kolonisi ile optimizasyonu

Makale Bilgileri

Araştırma Makalesi

Geliş: 03.03.2021

Kabul: 31.05.2021

DOI:

10.17341/gazimmfd.890180

Anahtar Kelimeler:

Aşırı öğrenme makinesi,
çoklu aşırı öğrenme
makinesi,
yapay arı kolonisi,
bütünleme sınavı

ÖZ

Bütünleme sınavları, ara sınavlar ve genel sınavlar neticesinde dersi başaramayan öğrencilerin katıldığı bir sınavdır ve ülkemizde birçok üniversite tarafından uygulanmaktadır. Her ne kadar dersi başaramayan öğrencilerin bu sınavlara katılacağı varsayılsa da öğrencilerin bir kısmı çeşitli içsel (ara sınav notunun çok düşük olması, dersi öğrenme amacıyla yeniden alma isteği, tatil yapma ihtiyacı vb.) ve dışsal motivasyonlarla (memleketi ile üniversite arası mesafe, maddi imkânsızlıklar vb.) bütünleme sınavına katılmamaktadır. Tüm öğrencilerini sınava katılacağı varsayılarak yapılan sınav programları, gözetmen atama ve sınav kâğıdı çoğaltılması gibi süreçler ise maddi israflara neden olmaktadır. Bu çalışmada bütünleme sınavına kalan öğrencilerin bütünleme sınavına katılıp katılmayacağını tespit edilmesi amacıyla aşırı öğrenme makinesi tabanlı bir yaklaşım önerilmiştir. Önerilen yaklaşımda birden fazla aşırı öğrenme makinesi aynı veri setinin belirli bir oranındaki örnekler ile eğitime tabi tutulmuş ve test aşamasında oylama tabanlı bir yaklaşım ile sınıflandırma yapılmıştır. Oylamaya katılacak ELM karar parametrelerinin optimizasyonu ise yapay arı kolonisi algoritması ile sağlanmıştır. Önerilen yöntemlerin eğitimi ve test edilmesi 10 adet ders veri kümesi üzerinde yapılmıştır. Deneysel çalışmalarda aşırı öğrenme makinesi ve önerilen yöntemin performansı her ders için ayrı ayrı karşılaştırılmıştır. Önerilen yöntemin performansının sınıflandırma doğruluğu açısından temel aşırı öğrenme makinesinden daha yüksek olduğu gözlemlenmiştir.

A k-ELM approach to the prediction of number of students taking make-up exams

HIGHLIGHTS

- Prediction of number of students taken make-up exams using extreme learning machine
- A novel approach named k-ELM based extreme learning machine
- Non-identical ELMs optimized by artificial bee colony algorithm

Article Info

Research Article

Received: 03.03.2021

Accepted: 31.05.2021

DOI:

10.17341/gazimmfd.890180

Keywords:

Extreme learning machine,
multiple extreme learning
machine,
artificial bee colony,
make-up exam

ABSTRACT

Make-up exam is an exam attended by students who fail the course at the end of midterm and general exams and are administered by many universities in our country. Although it is assumed that the students who fail the course will participate in these exams, some of the students may be affected by various internal (low midterm grade, desire to take the course again, need to take a vacation, etc.) and external motivations (distance between hometown and university, financial difficulties, etc.) does not take the make-up exams. Processes such as exam schedules, appointment of invigilators, copy of exam papers and toner, which are made by assuming that all students will take the exam, have caused tangible wastes. In this study, an extreme learning machine-based approach has been proposed in order to determine whether the students who they are expected to join the make-up exam will participate in the make-up exam. In the proposed approach, more than one extreme learning machine have been trained by using samples at a certain rate of the dataset, and the classification have been made with a voting-based approach in the testing phase. Furthermore, the ELM structures those join the voting, have been optimized by artificial bee colony algorithm. The training and testing of the proposed method have been conducted on 10 courses datasets. In the experimental studies, the performance of the extreme learning machine and the proposed method have been compared separately for each dataset, and it is seen that the performance of the proposed method is higher than the basic extreme learning machine in terms of classification accurac.

*Sorumlu Yazar/Yazarlar / Corresponding Author/Authors : *mskiran@ktun.edu.tr, eyup@selcuk.edu.tr, eesme@selcuk.edu.tr /

Tel: +90 332 205 1542

1. GİRİŞ (INTRODUCTION)

Bütünleme sınavı, ara sınavlar ve genel sınavlar neticesinde dersi başaramayan öğrenciler için yapılan bir sınavdır. Genellikle genel sınav notunun yerine bütünleme sınavı notu kullanılmaktadır ve yükseköğretim kurumunda bütünleme sınavı uygulanmakta ise her ders için bu sınav yapılmaktadır. Bütünleme sınavına kalan öğrencilerden bir kısmı ise ara sınav notunun çok düşük olması, dersi öğrenme isteği, tatilin bölünmemesi gibi içsel motivasyonlar ve memleketi ile üniversitesi arasındaki mesafe, ulaşım giderleri, kalacak yer sorunu vb. maddi imkânsızlıklar nedeniyle bu sınavlara katılmamaktadırlar. Bütünleme sınav programları ise bütünleme sınavına kalan bütün öğrenciler dikkate alınarak hazırlanmakta, sınav salonu tahsisi yapılmakta ve bu salonlara gözetmenler atanmaktadır. Ayrıca sınav kâğıdının çoğaltılması da bütünleme sınavına kalan öğrenci sayısına göre yapılmaktadır. Tüm bunlar neticesinde gereksiz salon tahsisi, fazla gözetmen atama, fazla kâğıt çoğaltma ve toner israfı gibi olumsuz sonuçlar ortaya çıkabilmektedir. Bu çalışmada bu olumsuzlukları azaltmak amacıyla yapay sinir ağı tabanlı bir çözüm önerilmiştir. İyi bir genelleştirme kabiliyetine sahip olması, son derece (extreme) hızlı öğrenme süreci ve iteratif olmayan eğitim algoritması olmasından dolayı aşırı öğrenme makinesi [1, 2], kısaca ELM, algoritması ve k-ELM adı verilen oylama tabanlı bir versiyonu yapay sinir ağı eğitimi için geliştirilmiş ve uygulanmıştır. ELM, Huang vd. [3] tarafından geliştirilmiş bir yapay sinir ağı eğitim algoritmasıdır. Bu ağ girdi katman ağırlıkları rastgele seçilen, tek gizli katmanlı ve ileri beslemeli bir sinir ağıdır. Geleneksel gradyan tabanlı öğrenme algoritmalarından daha yüksek genelleme performansı ile çok daha yüksek öğrenme hızına sahiptir. Ayrıca durdurma kriteri belirleme ve yerel minimumlara takılma gibi gradyan tabanlı öğrenme algoritmalarının karşılaştığı sorunlara sahip değildir. Bu avantajları nedeniyle birçok alanda yaygın şekilde kullanılmaktadır. Huang ve ark. giriş katmanındaki ağırlıkları rastgele seçmeyi önermiş ve gizli katman ile çıkış katmanı arasındaki ağırlıkların hesaplanmasında lineer bir yöntem uygulamışlardır. Bu yöntemi uygularken kare olmayan bir matrisin tersinin hesaplanması için Moore-Penrose genelleştirilmiş ters matris yaklaşımını önermişlerdir [1, 4, 5]. Literatürde ELM tabanlı algoritmalar hızlı eğitim süreci ve genelleştirme performansı nedeniyle araştırmacılar tarafından ilgi görmektedir ve bir kısmı şu şekilde özetlenebilir. Artırımlı aşırı öğrenme makinesi, I-ELM, gizli düğümleri rastgele üretir ve SLFNs nin çıkış ağırlıklarını analitik olarak hesaplar. Ancak I-ELM yeni bir düğüm eklendiğinde mevcut düğümlerin ağırlıklarını yeniden hesaplamaz [6]. Bundan dolayı, konveks artırımlı aşırı öğrenme makinesi ile rastgele yeni bir düğüm eklendiğinde mevcut düğümlerin ağırlıklarının yeniden hesaplanarak daha da iyileştirilebileceği [7] çalışmada gösterilmiştir. Miche vd. [8] optimal budanmış ELM (optimally pruned extreme learning machine - OP-ELM) yöntemini önermişlerdir. Bu yöntem klasik ELM'yi daha sağlam ve genel bir hale getirecek adımlar içermektedir. Yazarlar ELM'yi SVM ile karşılaştırmış ve

sınıflandırma için ELM'nin geleneksel SVM'den daha iyi genelleme performansı elde etme eğiliminde olduğu ayrıca ELM'nin kullanıcı tarafından belirlenen parametrelere karşı daha az duyarlı olması ve kolay uygulanabilirliğini ortaya koymuşlardır. 2011'de Huang vd. [9] ELM'yi regresyon ve çok sınıflı sınıflandırma için kullanmışlardır. Bu çalışmada ELM'nin regresyon ve çok sınıflı sınıflandırma uygulamalarına doğrudan uygulanabildiğini, ELM'nin benzer yöntemlere nazaran daha az optimizasyon kısıtlamalarına sahip olduğunu, teorik olarak benzer yöntemlerin optimumun altında çözümler ile daha yüksek hesaplama karmaşıklığı gerektirdiğini, son olarak da teoride ELM'nin herhangi bir hedef fonksiyona yaklaşabildiğini ve herhangi bir ayrık bölgeyi sınıflandırabildiğini göstermişlerdir [9]. 2013 yılında Zong vd. [10] genelleştirilmiş tek katmanlı ileri beslemeli ağlar için dengeli hem de dengesiz veri dağılımı olan veri setlerini ikili/çoklu sınıflandırmada Ağırlıklı ELM (weighted ELM)'nin birleşik bir çözümünü önermişlerdir. Bir başka çalışmada yazarlar ELM'nin ağırlık güncellemesini lineer programlama problemi olarak ele almışlardır [11]. 2016'da Musikawan vd. [12] doğru sayıda gizli düğüm sayısının belirlenmesi sorununu çözmek için dışbükey artımlı aşırı öğrenme makinesini önermişlerdir. ELM algoritmasının iyileştirilmesinin yanı sıra bazı uygulamaları da literatürde bulunmaktadır. İnsan yüzü tanınması amacıyla ELM [13, 14] çalışmasında uygulanmıştır. Ayrıca 2011 yılında Mohammed ve ark. [15] insan yüzü tanıma algoritması sunmuşlar ve çalışmalarında çift yönlü iki boyutlu temel bileşen analizine ve ELM algoritmasına dayalı bir yöntem önermişlerdir. Yönlendirilebilir piramit dönüşümü ve ELM tabanlı bir algoritma renkli resimlerdeki yüz tanıma problemine [16] çalışmada uygulanmıştır. 2015 yılında, Ray vd. [17] yeraltı kablolarındaki arıza tespiti için s-dönüşümü ve ELM'yi birleştirmek suretiyle hibrit bir metod kullanmışlardır. 2016 yılında, hizmet ticareti, mal ve hizmet ihracatı, mal ve hizmet ithalatı, ticaret ve emtia ticaretine dayalı ekonomik büyümeyi öngörmek için ELM algoritması önerilmiştir [18]. 2017 yılında Kuang ve ark. [19] alt ekstremite hareketlerini doğru ve hızlı bir şekilde tanımlamak için, aşırı öğrenme makinesine dayalı bir tanıma yöntemi önermişlerdir. 2018 yılında ise Razavi vd. [20] lityum-iyon pillerin kalan ömürlerinin tahmini için, Hanbay ve Türkoğlu bitki türü tespiti için [21] ELM kullanmışlardır. Birçok farklı ELM versiyonu ve uygulaması literatürde bulunmasına karşın V-ELM olarak isimlendirilen çoklu ELM çalışması ve analizi [22] çalışmada ortaya konulmuştur. Bu çalışmada oylamaya katılacak ELM sayısı ve sınıflandırma performansları analiz edilmiştir. Oylamaya katılacak ELM sayısının 7 olduğunda kabul edilebilir ve performanslı çözümlerin elde edildiğini göstermişlerdir. V-ELM'de oylamaya katılacak ELM'lerin özdeş olduğu varsayımı yapılmıştır. Bu çalışmada ise ELM'lerin özdeş olması varsayımı yerine ELM'lerin oluşturulması için güncelleme ve uygulama kolaylığı, optimizasyon problemlerindeki iyi performansı ve kolay anlaşılabilirliği nedeniyle yapay arı kolonisi -ABC- [23-25] algoritması kullanılmıştır. Bu çalışmadaki yenilik ve literatüre katkı

kısaca yeni bir problem tanıtımı ve ELM ile için karar parametrelerinin belirlenmesi amacıyla yapay arı kolonisi algoritmasının önerilmesidir.

Çalışmanın organizasyonu şu şekildedir: 1. Bölümde çalışma tanıtılmış ve kısaca literatür özetlenmiştir. 2. Bölümde ELM anlatılmış ve 3. Bölümde önerilen yaklaşım ve ABC verilmiştir. 4. Bölümde deneylerde kullanılan veri setleri tanıtılmıştır. Deneyler ve deneysel sonuçlar 5. Bölümde verilmiştir. Son bölüm olan 6. Bölümde ise elde edilen sonuçlar tartışılmıştır.

2. AŞIRI ÖĞRENME MAKİNESİ (EXTREME LEARNING MACHINE)

Aşırı öğrenme makinesi, kısaca ELM, ileri beslemeli yapay sinir ağının eğitimi için kullanılan tek gizli katmana sahip bir yapay sinir ağı eğitim algoritmasıdır [1, 26]. ELM yapay sinir ağının gizli katmanı ile çıkış katmanı arasındaki ağırlıkların analitik bir yöntem ile tek adımda hesaplar. Eğitim süreci tek adımda hesaplandığından dolayı aşırı hızlı bir eğitim sürecine sahiptir. Bir ileri beslemeli yapay sinir ağında gizli katmanda N tane nöron olduğu ve aktivasyon fonksiyonunun $g(x)$ olduğu kabul edildiğinde M adet (x_i, t_i) eğitim örneği için çıkış hesabı Eş. 1 ile yapılır.

$$O_j = \sum_{i=1}^N \beta_i g_i(x_j) = \sum_{i=1}^N \beta_i g_i(w_i \cdot x_j + b_i), \quad j = 1..M \quad (1)$$

Eş. 1'de $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in R^n$ giriş vektörü, $t_i = [t_{i1}, t_{i2}, \dots, t_{in}]^T \in R^m$ giriş vektöründeki her bir eğitim örneğinin etiketi, $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ i. gizli düğüm ile giriş düğümü arasındaki ağırlık vektörü, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ i. gizli düğüm ile çıkış düğümü arasındaki ağırlık vektörü ve b_i gizli katman düğümünün biasıdır. Eş. 1 aşağıdaki şekilde tekrar yazılabilir.

$$H\beta = T \quad (2)$$

$$H = (w_1, \dots, w_N, b_1, \dots, b_N, x_1, \dots, x_N) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_N \cdot x_1 + b_N) \\ \vdots & & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_N \cdot x_N + b_N) \end{bmatrix} \quad (3)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{bmatrix} \text{ and } T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix} \quad (4)$$

Eş. 1 ve Eş. 2'deki H, yapay sinir ağının gizli katman çıkış matrisidir. İleri beslemeli yapay sinir ağını eğitmek için spesifik β_i' ve w_i' ağırlıkları gradyan tabanlı öğrenme algoritmaları ile Eş. 5 ile hesaplanmaktadır.

$$\|H(w'_1, \dots, w'_N, b'_1, \dots, b'_N)\beta - T\| \\ = \min_{w_i, b_i, \beta} \|(w_1, \dots, w_N, b_1, \dots, b_N, x_1, \dots, x_N)\beta - T\| \quad (5)$$

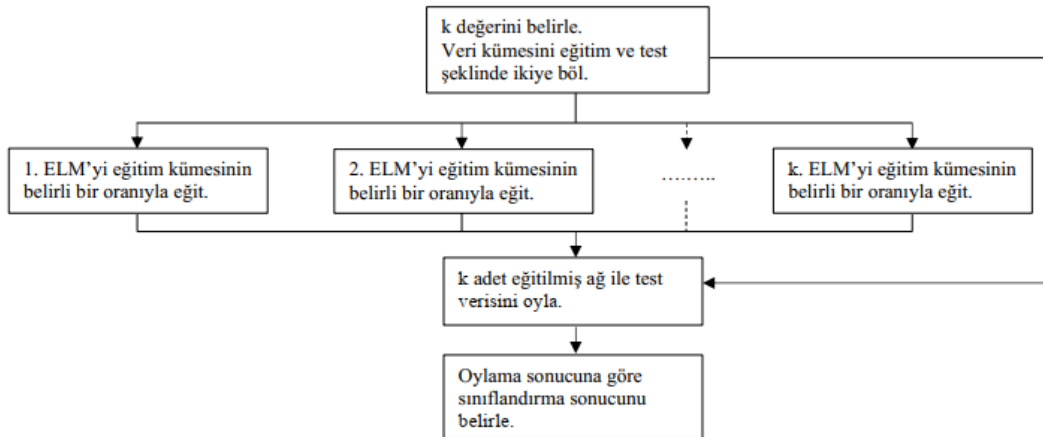
ELM'de giriş katmanı ile gizli katman arasındaki ağırlıklar rastgele atandıktan sonra gizli katman ile çıkış katmanı arasındaki ağırlıklar (β) Eş. 6 ile hesaplanabilir.

$$\beta = H^+ T \quad (6)$$

Eş. 6'daki H^+ Moore-Penrose genelleştirilmiş ters matristir. ELM'de eğitim örnek sayısı ile gizli katmandaki nöron sayısı eşit olduğunda H kare matris olduğundan Moore-Penrose yaklaşımına ihtiyaç olmamasına rağmen genellikle eğitim örnek sayısı gizli katmandaki nöron sayısından fazla olduğundan dolayı matrisinin tersinin alınabilmesi için Moore-Penrose yaklaşımı kullanılmaktadır. ELM'nin eğitim sürecinin neden çok hızlı olduğu da Eş. 5 ile verilen ağırlıkların belirlenebilmesi için çevrimsel yöntemler kullanılırken Eş. 6'da tek adımda ağırlıkların belirlenmesi ile ortaya çıkmaktadır.

3. ÖNERİLEN YAKLAŞIM – k-ELM (PROPOSED APPROACH – k-ELM)

k-ELM yaklaşımı k sayıda ELM'yi içermektedir ve her bir ELM'nin eğitimi için eğitim örneklerinin oransal olarak kullanılması temel almaktadır. k-ELM'de veri kümesi eğitim ve test kümesi şeklinde ikiye ayrıldıktan sonra eğitim kümesi oransal olarak k adet ELM'nin eğitimi için kullanılmaktadır. k-ELM'nin eğitim süreci Şekil 1'de gösterilmiştir.



Şekil 1. k-ELM eğitimi ve testi (The training and test of k-ELM)

k-ELM'nin test edilmesi için şekilde verildiği üzere oylama tabanlı bir yaklaşım kullanılmıştır. Test örneğinin sınıflandırılması için öncelikle eğitilmiş her ağ için bir çıkış üretilir ve bu ağlardan sınıf tayinleri elde edilir. Elde edilen bu sonuçlara göre etiket değeri en fazla olan sınıf k-ELM'nin ilgili test örneğine karşılık çıktısı olarak kabul edilir. k-ELM'de eğitilen yapay sinir ağlarının her ne kadar giriş ve çıkış nöron sayılarının veri kümesine bağımlı olması zorunluluğu bulunmasına karşın gizli katman nöron sayısı farklı olabilmektedir. Ayrıca bu çalışmada veri kümeleri ikili sınıflandırma etiketleri barındırdığından dolayı k değerinin tek sayı olmamasına dikkat edilmesi gerekmektedir. Aksi takdirde k-ELM oylama sonucunda sınıf etiketi tayin edilirken eşitlik durumu ortaya çıkabilir. k adet ELM oluşturulduğunda her ELM'nin özdeş olma zorunluluğu bulunmamaktadır. Bu noktada her ELM için iki parametrenin optimum değerinin belirlenmesi gerekmektedir. Bu parametreler gizli katman nöron sayısı ve aktivasyon fonksiyonudur. Bu parametrelerin optimizasyonu için bu çalışmada yapay arı kolonisi, kısaca ABC, algoritması önerilmiştir. ABC algoritması, başlatma, işçi arı, gözcü arı, kâşif arı ve sonlandırma fazı olmak üzere 5 fazdan oluşmaktadır ve yöntemin k-ELM'nin parametrelerinin optimizasyonu için uyarlanmış sözde kodu aşağıda verilmiştir.

Başlatma Fazı

- k değerini belirle
- Eğitim kümesini yükle ve onu test ve eğitim olmak üzere ikiye ayır.
- Kovadaki arı sayısını (2N) belirle ve yarısını işçi (N), diğer yarısını gözcü (N) olarak ata.
- Kâşif arı oluşumunu kontrol etmek için kullanılan *limit* parametresini tanımla.
- Problem boyutunu $D = k \times 2$ olarak ata (tek indisli boyutlar nöron sayısını, çift indisli boyutlar aktivasyonu fonksiyonunu göstermektedir).
- Her bir işçi arı için yiyecek kaynağını Eş. 7 kullanarak oluştur.

$$F_{i,j} = L_j^{min} + r_{i,j} \times (H_j^{min} - L_j^{min}), \quad (7)$$

$i = 1, 2, \dots, N$ ve $j = 1, 2, \dots, D$

Eş. 7'de, $F_{i,j}$ i. yiyecek kaynağının j. boyutunu, L_j^{min} j. boyutunun alt sınırını, H_j^{min} j. boyutun üst sınırını ve $r_{i,j}$ [0,1] aralığında rastgele üretilmiş bir sayıdır.

- Yiyecek kaynaklarının kalitesini Eş. 8 kullanarak hesapla.

$$fit_i = \frac{1}{1+f_i} \quad (8)$$

Eş. 8'de, fit_i i. çözümün kalitesini ve f_i ise Şekil 1 ile verilen modeli göstermektedir.

- En iyi çözümü *gbest* olarak ata.
- Her bir yiyecek kaynağı için C sayacını oluştur ve 0'a eşitle.

- Durdurma kriterini belirle.

İşçi Arı Fazı

Her bir yiyecek kaynağı için (yiyecek kaynakları indisi i olmak üzere)

- Rastgele bir komşu yiyecek kaynağı seç (m)
- Rastgele bir boyutu seç (j)
- Eş. 9 kullanarak bir aday yiyecek kaynağı oluştur ($V=F_i$)

$$V_j = F_{i,j} + \varphi \times (F_{i,j} - F_{m,j}) \quad (9)$$

- Eş. 8 ile aday yiyecek kaynağını hesapla ve F_i 'nin kalitesi ile karşılaştır. Eğer daha iyiyse aday yiyecek kaynağını ezberle ($F_i=V$), değilse i. yiyecek kaynağının sayacını bir artır ($C_i = C_i + 1$)

Gözcü Arı Fazı

Yiyecek kaynaklarının seçilme ihtimalini Eş. 10'u kullanarak hesapla.

$$p_i = \left(0,9 \times \frac{fit_i}{\max(fit)}\right) + 0,1 \quad (10)$$

Her gözcü arı için (yiyecek kaynakları indisi i olmak üzere)

- Eş. 10'da verilen ihtimal ile bir yiyecek kaynağını seç.
- Rastgele bir komşu yiyecek kaynağı seç (m)
- Rastgele bir boyutu seç (j)
- Eş. 9 kullanarak bir aday yiyecek kaynağı oluştur ($V=F_i$)
- Eş. 8 ile aday yiyecek kaynağını hesapla ve F_i 'nin kalitesi ile karşılaştır. Eğer daha iyiyse aday yiyecek kaynağını ezberle ($F_i=V$), değilse i. yiyecek kaynağının sayacını bir artır ($C_i = C_i + 1$)

Kâşif Arı Fazı

- C ile belirlenen sayaçlardan en büyük içeriğe sahip olanını belirle.
- Bu değeri *limit* ile karşılaştır. Eğer limit değerinden büyükse bu yiyecek kaynağının yerine yeni bir yiyecek kaynağını Eş. 7 ile oluştur ve Eş. 8 ile bu yiyecek kaynağının kalitesini hesapla.
- Yeni yiyecek kaynağının sayacını sıfırla.

Sonlandırma

- Kovadaki en iyi yiyecek kaynağını belirle ve *gbest* ile karşılaştır. Hangisi daha iyiyse *gbest* olarak ata.
- Durdurma kriteri sağlanmadıysa İşçi Arı Fazı'na git, sağlandıysa *gbest*'i raporla.

ABC algoritmasının k-ELM parametrelerinin optimizasyonu için uygulanmasında iki önemli nokta bulunmaktadır. Bir

tanesi Eş. 8 ile kalite hesaplanırken eğitim işleminin her ELM için ayrı ayrı uygulanması ve tümünün test verisi için oylama yapmasıdır. Diğer önemli nokta ise algoritmanın sürekli optimizasyon için önerilmiş olması nedeniyle yiyecek kaynaklarının ondalık sayılar ile ifade edilmesidir. ELM'de ise hem gizli katmandaki nöron sayıları hem de aktivasyon fonksiyonu belirlenmesi için tamsayı gösterimi kullanılmaktadır. Yani Eş. 8 ile kalitenin hesabı yapılmadan hemen önce yiyecek kaynaklarının eşit veya büyük en yakın tamsayıya yuvarlanması gerekmektedir. k-ELM'de, k değeri 7 alındığında her ELM'nin 2 karar değişkeni (aktivasyon fonksiyonu-AF ve nöron sayısı NS) olduğundan dolayı 14 boyutlu bir optimizasyon problemi ortaya çıkmaktadır. Bu problemi çözmek için az önce değinildiği üzere ABC algoritmasının tamsayı versiyonu bu çalışmada kullanılmıştır. ABC algoritmasında bir çözümün temsili Şekil 2'de gösterilmiştir.

Şekil 2'de NS gizli katman nöron sayısını ve AF ELM'nin aktivasyon fonksiyonudur. Bu bağlamda yapay sinir ağırları oluşturulmadan önce karar değişkenleri eşit veya büyük en yakın tamsayıya yuvarlanmaktadır. ($X = [X]$). Bu dönüşümden sonra AF eğer 1 ise sigmoid (Sig), 2 ise sinüs (Sin), 3 ise katı sınırlayıcı (Hardlim), 4 ise üçgensel temel (triangular basis- Tribas), 5 ise radyal temel (radial basis-Radbas), 6 ise tanjant sigmoid (TanSig) aktivasyon fonksiyonu ilgili ELM'de kullanılmaktadır. ABC algoritmasında Eş. 9 ile oluşturulan aday çözüm eğer sınırların dışında bir değer alırsa (NS için 5'ten küçük veya 50'den büyük olması, AF için 1'den küçük veya 6'dan büyük olması) kelepçeleme ile sınır değerlere çekilmektedir. Yani NS 5'ten küçük ise 5'e, 50'den büyük ise 50 değerini almaya zorlanmaktadır.

4. DENEYLERDE KULLANILAN VERİ KÜMELERİ (DATASETS USED IN EXPERIMENTS)

Deneyleerde kullanılan veri kümeleri Konya Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü'nde okutulan Bilgisayar Mühendisliğine Giriş (D1), Atatürk İlke ve İnkılap Tarihi (D2), Fizik 1 (D3), İşletim Sistemleri (D4), Matematik 1 (D5), Matematik 2 (D6), Türk Dili 1 (D7), Türk Dili 2 (D8) ve Veri Yapıları (9) derslerinden bütünlümeye kalan öğrencilerin kişisel olmayan verileri kullanılmıştır. Ayrıca bu 9 dersin birleştirilmesiyle oluşan bir veri kümesi (D10) daha deneysel çalışmalarda kullanılmıştır. Veri kümelerinde eksik veri bulunmamaktadır ve her veri kümesi 10 farklı özellik içermektedir. Bu özellikler öğrencinin dönem ortalaması, ağırlıklı not ortalaması, öğrencinin aldığı ders sayısı, dersten aldığı ara sınav puanı, final puanı, tekrar aldığı ders sayısı, öğrencinin dönemi, öğrencinin memleketi ile Konya arası mesafe, cinsiyet ve öğrencinin kaldığı bütünlümeye sınavı sayısıdır. Bu veri kümeleri için iki adet

sınıf etiketi oluşturulmuştur. Sınıf etiketi 0 ise öğrenci bütünlümeye sınavına kaldığı halde girmemiştir ve 1 ise öğrenci bütünlümeye sınavına kalmıştır ve bütünlümeye sınavına girmiştir. Her veri kümesi için sınıflardaki örnek sayıları aşağıdaki gibidir. D1 veri kümesindeki toplam örnek sayısı 124, 0 sınıfında 52 ve 1 sınıfında 72 örnek; D2 veri kümesindeki toplam örnek sayısı 71, 0 sınıfında 32 ve 1 sınıfında 39 örnek; D3 veri kümesindeki toplam örnek sayısı 391, 0 sınıfında 132 ve 1 sınıfında 259 örnek; D4 veri kümesindeki toplam örnek sayısı 196, 0 sınıfında 69 ve 1 sınıfında 127 örnek; D5 veri kümesindeki toplam örnek sayısı 391, 0 sınıfında 137 ve 1 sınıfında 254 örnek; D6 veri kümesindeki toplam örnek sayısı 497, 0 sınıfında 253 ve 1 sınıfında 244 örnek; D7 veri kümesindeki toplam örnek sayısı 84, 0 sınıfında 33 ve 1 sınıfında 51 örnek; D8 veri kümesindeki toplam örnek sayısı 62, 0 sınıfında 22 ve 1 sınıfında 40 örnek; D9 veri kümesindeki toplam örnek sayısı 292, 0 sınıfında 140 ve 1 sınıfında 152 örnek; D10 veri kümesindeki toplam örnek sayısı 2108, 0 sınıfında 870 ve 1 sınıfında 1238 örnek bulunmaktadır.

5. DENEYSEL SONUÇLAR (EXPERIMENTAL RESULTS)

k-ELM için ELM'lerin özdeş olmadığı durumda k, gizli katman nöron sayısı ve aktivasyon fonksiyonu olmak üzere üç algoritmik parametre bulunmaktadır. Cao vd. [22] özdeş ELM'ler için k değerinin 7 alındığında 19 veri kümesi üzerinde önerdikleri yaklaşımın temel ELM'den daha iyi olduğunu göstermişlerdir. Bundan dolayı deneysel çalışmada k değeri 7 olarak alınmıştır. k değeri 7 alındığında her ELM'nin 2 karar değişkeni olduğundan dolayı 14 boyutlu bir optimizasyon problemi ortaya çıkmaktadır. Bu problemi çözmek için ABC algoritmasının tamsayı versiyonu bu çalışmada kullanılmıştır. ABC algoritmasında algoritmanın kontrol parametreleri kovan popülasyonu, limit değeri ve durdurma kriteridir. Literatürdeki çalışmalara göre ABC algoritması yüksek popülasyon boyutlarına ihtiyaç duymamaktadır [27-29]. Bundan dolayı kovandaki popülasyon sayısı literatüre uygun olarak yiyecek kaynağı sayısının 2 katı olarak seçilmiş ve 40 olarak alınmıştır. Literatürde limit değeri popülasyona ve karar değişkeni sayısına bağlı olarak aşağıdaki şekilde hesaplanmaktadır [28, 30, 31].

$$\text{limit} = (N \times D)/2 \quad (11)$$

Eş. 11'de N popülasyondaki arı sayısını ve D karar değişkeni sayısını göstermektedir. Bu çalışmada popülasyon boyutu 40, karar değişkeni sayısı 14 olduğundan dolayı limit değeri 280 olarak hesaplanmaktadır. Durdurma kriteri ise maksimum çevrim sayısıdır ve 500 olarak alınmıştır. ELM'lerin eğitim ve testi için rasgele örneklem (random sampling) metodu kullanılmıştır ve veri kümesinin %90'ı ile

1. ELM		2. ELM		3. ELM		4. ELM		5. ELM		6. ELM		7. ELM	
NS	AF	NS	AF	NS	AF	NS	AF	NS	AF	NS	AF	NS	AF
16,7	3,8	40,1	3,5	28,3	1,6	31,6	4,8	47,2	5,2	10,	1,5	10,8	4,2

Şekil 2. ABC'de çözümün temsili (Representation of a solution in ABC)

eğitim ve kalan %10 ile eğitilmiş ağ test edilmiştir. Optimizasyon safhasında her bir çözümün kalitesini belirlemek amacıyla 30 defa ELM'ler eğitilmiş ve test edilmiştir. Ortalama hata ise çözümün kalitesi hesaplamak amacıyla kullanılmıştır dolayısıyla problem bir minimizasyon problemi (ELM sınıflandırma hatasının minimizasyonu) olarak ele alınmıştır. Eş. 8 ile de hata ne kadar küçükse uygunluk o kadar yüksek, hata ne kadar büyükse uygunluk o kadar düşük olacak şekilde hesaplama yapılmıştır. k-ELM'deki gizli katman nöron sayısı karar değişkeni 5 ile 50 arasında ve aktivasyon fonksiyonu karar değişkeni ise 1 ile 6 arasında tamsayı değerler alabilir: 1 ise sigmoid (Sig), 2 ise sinüs (Sin), 3 ise katı sınırlayıcı (Hardlim), 4 ise üçgensel temel (triangular basis- Tribas), 5 ise radyal temel (radial basis-Radbis), 6 ise tanjant sigmoid (TanSig) aktivasyon fonksiyonu ilgili ELM'de ayarlanmış anlamına gelmektedir. Bu şartlar altında ABC ile oluşturulan ağlara ait bilgiler Tablo 1'de verilmiştir. Tablo 1'den görüldüğü üzere Sin aktivasyon fonksiyonu hiçbir ELM'de kullanılmamıştır. Diğer aktivasyon fonksiyonları ise Sig 10 defa, Hardlim 26 defa, Tribas 11 defa, Radbas 5 defa ve TanSig 18 defa kullanılmıştır.

ABC ile tek katmanlı ileri beslemeli yapay sinir ağları oluşturulduktan sonra ELM'ler her veri kümesi için 100 defa eğitime ve teste tabi tutulmuştur. 100 defa çalıştırmanın her birinde verinin %90'ı eğitim için ayrılmış kalan %10'u ile test yapılmıştır. Ayrıca k değerinin 1 alındığı özel durumda ELM algoritması temel ELM olarak çalışmaktadır. Temel ELM algoritması da gizli katman nöron sayısı 50 ve 6 adet aktivasyon fonksiyonu ile ayrı ayrı her veri kümesi için çalıştırılmış ve sonuçlar raporlanmıştır. Tablolarda 100 defa çalıştırma sonucunda elde edilen en iyi sınıflandırma performansı, ortalama test performansı ve standart sapmalar verilmiştir. Tüm sonuçlar bir tabloda verildiğinde analiz etmek zorlaştığından ve sayfa yapısı bozulduğundan dolayı sonuçlar 3 tabloya bölünerek verilmiştir. Bu bağlamda Tablo 2'de önerilen yaklaşım Sigmoid, Sinüs aktivasyon fonksiyonlarına sahip temel ELM ile, Tablo 3'te Hardlim ve Tribas aktivasyon fonksiyonlarına sahip ELM ile ve Tablo 4'te ise Radbas ve Tanjant Sigmoid aktivasyon fonksiyonlarına sahip temel ELM algoritması ile karşılaştırılmıştır. Tablo 2, Tablo 3 ve Tablo 4'ten görüldüğü üzere önerilen yaklaşım tüm veri setlerinde karşılaştırılan ELM versiyonlarından daha başarılı sonuçlar üretmektedir.

Tablo 1. ABC ile oluşturulan ELM'ler (ELM optimized by ABC)

Veri Kümesi	Parametre	1.ELM	2.ELM	3.ELM	4.ELM	5.ELM	6.ELM	7.ELM
D1	NS	14	27	38	28	23	50	21
	AF	TanSig	Tribas	Tribas	TanSig	TanSig	Hardlim	Hardlim
D2	NS	17	9	8	34	14	39	29
	AF	Hardlim	TanSig	Hardlim	Hardlim	Radbis	Hardlim	Hardlim
D3	NS	8	17	36	21	36	31	34
	AF	Tribas	Hardlim	Radbis	Sig	TanSig	TanSig	TanSig
D4	NS	42	34	28	43	38	33	36
	AF	Hardlim	TanSig	Sig	Sig	Radbis	Hardlim	Hardlim
D5	NS	48	23	26	50	18	28	12
	AF	TanSig	TanSig	Tribas	Tribas	TanSig	Hardlim	Tribas
D6	NS	49	15	44	35	31	40	44
	AF	TanSig	Radbis	TanSig	Hardlim	TanSig	Sig	Sig
D7	NS	40	19	50	45	35	33	29
	AF	Tribas	Hardlim	Hardlim	Tribas	Hardlim	TanSig	Tribas
D8	NS	15	32	6	27	5	23	42
	AF	Radbis	Hardlim	TanSig	Hardlim	Hardlim	TanSig	Hardlim
D9	NS	27	19	21	32	7	29	28
	AF	Sig	Sig	Hardlim	Sig	Hardlim	Hardlim	Hardlim
D10	NS	17	30	37	50	23	50	19
	AF	Tribas	Sig	Hardlim	Hardlim	TanSig	Sig	Tribas

Tablo 2. Önerilen yaklaşımın Sigmoid ve Sinüs aktivasyon fonksiyonlarına sahip temel ELM ile karşılaştırılması
(The comparison of the proposed approach with basic ELM with the Sigmoid and Sinus activation functions)

Veri Seti	Sig			Sin			Önerilen Yaklaşım		
	En İyi	Ortalama	Std. Sapma	En İyi	Ortalama	Std. Sapma	En İyi	Ortalama	Std. Sapma
D1	0,9167	0,6058	0,1391	0,8333	0,4950	0,1610	1,0000	0,6583	0,1439
D2	1,0000	0,5729	0,1984	1,0000	0,5629	0,2008	1,0000	0,6529	0,1702
D3	0,8974	0,7210	0,0703	0,7179	0,5167	0,0770	0,8718	0,7182	0,0698
D4	0,9000	0,6935	0,1160	0,7500	0,5180	0,1088	0,9500	0,7435	0,0934
D5	0,8462	0,7169	0,0822	0,7436	0,5159	0,0873	0,9231	0,7462	0,0843
D6	0,9400	0,7044	0,0749	0,7200	0,4990	0,0698	0,8600	0,7116	0,0745
D7	1,0000	0,6013	0,2054	0,8750	0,5250	0,1713	1,0000	0,6425	0,1732
D8	1,0000	0,5133	0,2256	1,0000	0,5150	0,1897	1,0000	0,7200	0,1622
D9	0,8621	0,7100	0,0798	0,7241	0,5341	0,0933	0,8966	0,7428	0,0801
D10	0,8009	0,7127	0,0407	0,6161	0,5225	0,0361	0,7962	0,7392	0,0322

Tablo 3. Önerilen yaklaşımın Hardlim ve Tribas aktivasyon fonksiyonlarına sahip temel ELM ile karşılaştırılması
(The comparison of the proposed approach with basic ELM with the Hardlim and Tribas activation functions)

Veri Seti	Hardlim			Tribas			Önerilen Yaklaşım		
	En İyi	Ortalama	Std. Sapma	En İyi	Ortalama	Std. Sapma	En İyi	Ortalama	Std. Sapma
D1	0,9167	0,6417	0,1374	0,8333	0,4533	0,1373	1,0000	0,6583	0,1439
D2	1,0000	0,6100	0,1881	1,0000	0,4857	0,1827	1,0000	0,6529	0,1702
D3	0,8462	0,7121	0,0700	0,7179	0,4041	0,0836	0,8718	0,7182	0,0698
D4	0,9500	0,7040	0,1105	0,6500	0,4305	0,1070	0,9500	0,7435	0,0934
D5	0,9231	0,7146	0,0896	0,6154	0,4508	0,0799	0,9231	0,7462	0,0843
D6	0,8600	0,6758	0,0813	0,7400	0,5436	0,0689	0,8600	0,7116	0,0745
D7	1,0000	0,6413	0,1596	0,8750	0,4338	0,1816	1,0000	0,6425	0,1732
D8	1,0000	0,6300	0,2046	1,0000	0,4733	0,2006	1,0000	0,7200	0,1622
D9	0,8966	0,7076	0,0847	0,7931	0,5645	0,1037	0,8966	0,7428	0,0801
D10	0,8104	0,7154	0,0440	0,5687	0,4856	0,0317	0,7962	0,7392	0,0322

Tablo 4. Önerilen yaklaşımın Radbas ve TanSig aktivasyon fonksiyonlarına sahip temel ELM ile karşılaştırılması
(The comparison of the proposed approach with basic ELM with the Radbas and TanSig activation functions)

Veri Seti	Radbas			TanSig			Önerilen Yaklaşım		
	En İyi	Ortalama	Std. Sapma	En İyi	Ortalama	Std. Sapma	En İyi	Ortalama	Std. Sapma
D1	0,8333	0,5008	0,1569	1,0000	0,6042	0,1467	1,0000	0,6583	0,1439
D2	1,0000	0,5329	0,1998	1,0000	0,5900	0,1812	1,0000	0,6529	0,1702
D3	0,8205	0,6203	0,0922	0,8462	0,7008	0,0637	0,8718	0,7182	0,0698
D4	0,9500	0,5940	0,1175	0,9500	0,7070	0,0987	0,9500	0,7435	0,0934
D5	0,8718	0,6403	0,0961	0,9487	0,7185	0,0842	0,9231	0,7462	0,0843
D6	0,8200	0,6052	0,0855	0,8600	0,6700	0,0862	0,8600	0,7116	0,0745
D7	0,8750	0,5413	0,1804	0,8750	0,5600	0,1614	1,0000	0,6425	0,1732
D8	1,0000	0,4967	0,2132	0,8333	0,5367	0,1904	1,0000	0,7200	0,1622
D9	0,8621	0,6293	0,0896	0,9310	0,7059	0,0902	0,8966	0,7428	0,0801
D10	0,7299	0,6027	0,0576	0,7820	0,7110	0,0398	0,7962	0,7392	0,0322

Sin ve Tribas aktivasyon fonksiyonu kullanan temel ELM en kötü performansı sergilemiştir. Önerilen yaklaşıma en yakın sonuçlar ise Hardlim aktivasyon fonksiyonu temel ELM’de kullanıldığında elde edilmiştir.

5. SONUÇLAR VE TARTIŞMALAR (RESULTS AND DISCUSSIONS)

Bu çalışmada yapay sinir ağlarının eğitiminde ELM metodunun çoklu versiyonu olan k-ELM algoritması geliştirilmiştir. k-ELM algoritması bütünleme sınavına girecek öğrenci sayılarının tespiti için sınıflandırma yöntemi olarak kullanılmıştır. 10 veri seti üzerinde k-ELM algoritmasında k değeri 7 olarak alınarak sınıflandırma oylamasına katılacak 7 ELM için aktivasyon fonksiyonlarının ve gizli katman nöron sayılarının optimizasyonu için ABC algoritması önerilmiştir. ABC algoritması gizli katman nöron sayısı 5 ile 50 arasında ve 6 adet aktivasyon fonksiyonu olmak üzere toplam $1,22E+17$ farklı ELM içerisinden en yüksek performans sağlayacak 7 tanesini bulması için kullanılmıştır. ABC algoritması 500 iterasyonda 40 arıyla çözüm uzayını araştırmış, en yüksek performansa sahip 7 ELM yapısını kurmuştur ve elde edilen sonuçlar Tablo-1’de sunulmuştur. 6 adet aktivasyon

fonksiyonunun her biri için gizli katman nöron sayısı 50 alınarak temel ELM algoritması aynı veri kümesi üzerinde koşurulmuş ve elde edilen sonuçlar ile önerilen yaklaşımın sonuçları Tablo-2, Tablo 3 ve Tablo 4’te karşılaştırılmıştır. Tablo 1, Tablo 2, Tablo 3 ve Tablo 4 birlikte değerlendirildiğinde ABC algoritmasının başarılı olduğu gözlemlenmiştir. Bunun sebebi ise ABC algoritması 10 veri kümesi üzerinde 7 ELM’den yani toplam 70 ELM yapısı içerisinde 26’sında kullanılan Hardlim’dir. Temel ELM algoritmasında da Hardlim kullanıldığında önerilen yonteme en yakın sonuçların elde edildiği Tablo 3’te görülmektedir. Tribas aktivasyon fonksiyonunun kullanıldığı ELM yapısı en kötü sonuçlar üretmesine rağmen Tablo 1’de verilen ABC ile oluşturulan ELM’lerde 11 defa kullanılmaktadır. Buna rağmen Sin, Tribas ve Radbas aktivasyon fonksiyonlarının kullanıldığı ELM yapıları hiçbir veri kümesi için önerilen yonteme en yakın sonuçlar üreten Hardlim, Sig ve Tansig aktivasyon fonksiyonlarının kullanıldığı ELM yapılarından Tablo 1’deki satırlarda ağırlıklı olarak yer almamaktadır. Bu durumdan anlaşılacağı üzere ABC tarafından oluşturulan ELM’lerin performansı gizli katman nöron sayısının ve aktivasyon fonksiyonunun sabit olarak atandığı temel ELM’lerden daha iyi sınıflandırma performansından daha iyidir.

TEŞEKKÜR (ACKNOWLEDGEMENT)

Yazarlar, Konya Teknik Üniversitesi'ne verileri sağladığı için teşekkür eder.

KAYNAKLAR (REFERENCES)

1. Huang G.B., Zhu Q.Y., Siew C.K., Extreme learning machine: theory and applications, *Neurocomputing*, 70 (1-3), 489-501, 2006.
2. Alcin O.F., Sengur A., Ince M.C., Forward-Backward Pursuit Based Sparse Extreme Learning Machine, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 30 (1), 111-117, 2015.
3. Huang G.B., Zhu Q.Y., Siew C.K., Extreme learning machine: A new learning scheme of feedforward neural networks, *IEEE International Joint Conference on Neural Networks*, Budapest-Hungary, 985-990, 25-29 July, 2004.
4. Moore E.H., On the reciprocal of the general algebraic matrix, *Bulletin of the American Mathematical Society*, 26, 394-395, 1920.
5. Penrose R., A generalized inverse for matrices, *Mathematical proceedings of the Cambridge philosophical society*, 51 (3), 406-413, 1955.
6. Huang G.B., Chen L., Siew, C.K. Universal approximation using incremental constructive feedforward networks with random hidden nodes, *IEEE Transactions on Neural Networks*, 17 (4), 879-892, 2006.
7. Huang G.B., Chen L., Convex incremental extreme learning machine, *Neurocomputing*, 70 (16-18), 3056-3062, 2007.
8. Miche Y., Sorjamaa A., Bas P., Simula O., Jutten C., Lendasse A., OP-ELM: optimally pruned extreme learning machine, *IEEE Transactions on Neural Networks*, 21 (1), 158-162, 2010.
9. Huang G.B., Zhou H., Ding X., Zhang R., Extreme learning machine for regression and multiclass classification, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42 (2), 513-529, 2011.
10. Zong W., Huang G.B., Chen Y., Weighted extreme learning machine for imbalance learning, *Neurocomputing*, 101, 229-242, 2013.
11. Balasundaram S., Gupta D., 1-Norm extreme learning machine for regression and multiclass classification using Newton method, *Neurocomputing*, 128, 4-14, 2014.
12. Musikawan P., Sunat K., Chiewchanwattana S., Horata P., Kongsorot Y., Improved convex incremental extreme learning machine based on ridgelet and PSO algorithm, *13th International Joint Conference on Computer Science and Software Engineering*, Khon Kaen - Thailand, 1-6, 13-15 July, 2016.
13. Zong W., Huang G.B., Face recognition based on extreme learning machine, *Neurocomputing*, 74 (16), 2541-2551, 2011.
14. Zong W., Zhou H., Huang G.B., Lin Z., Face recognition based on kernelized extreme learning machine, *International Conference on Autonomous and Intelligent Systems*, Burnaby, BC - Canada, 263-272, 22-24 June, 2011.
15. Mohammed A.A., Minhas R., Wu Q.J., Sid-Ahmed M.A., Human face recognition based on multidimensional PCA and extreme learning machine, *Pattern recognition*, 44 (10-11), 2588-2597, 2011.
16. Ucar A., Color face recognition based on steerable pyramid transform and extreme learning machines, *The Scientific World Journal*, 2014, 2014.
17. Ray P., Mishra D., Application of extreme learning machine for underground cable fault location, *International Transactions on Electrical Energy Systems*, 25 (12), 3227-3247, 2015.
18. Sokolov-Mladenovic S., Milovancevic M., Mladenovic I., Alizamir M., Economic growth forecasting by artificial neural network with extreme learning machine based on trade, import and export parameters, *Computers in Human Behavior*, 65, 43-45, 2016.
19. Kuang Y., Wu Q., Shao J., Wu J., Wu X., Extreme learning machine classification method for lower limb movement recognition, *Cluster Computing*, 20 (4), 3051-3059, 2017.
20. Razavi-Far R., Chakrabarti S., Saif M., Zio E., Palade V., Extreme learning machine based prognostics of battery life, *International Journal on Artificial Intelligence Tools*, 27 (08), 2018.
21. Turkoglu M., Hanbay D., Plant recognition system based on extreme learning machine by using shearlet transform and new geometric features, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 34 (4), 2097-2112, 2019.
22. Cao J., Lin Z., Huang G.B., Liu N., Voting based extreme learning machine, *Information Sciences*, 185 (1), 66-77, 2012.
23. Karaboga D., Basturk B., A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, 39 (3), 459-471, 2007.
24. Ozturk C., Hancer E., Karaboga D., Automatic clustering with global best artificial bee colony algorithm, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 29 (4), 667-687, 2014.
25. Eke I., Taplamacioglu M., Kocaarslan I., Design of robust power system stabilizer based on artificial bee colony algorithm, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 26 (3), 683-690, 2011.
26. Huang G.B., Zhu Q.Y., Siew C.K., Extreme learning machine: a new learning scheme of feedforward neural networks, *IEEE International Joint Conference on Neural Networks*, Budapest-Hungary, 985-990, 25-29 July, 2004.
27. Akay B., Karaboga D., Artificial Bee Colony Algorithm. In *Swarm Intelligence Algorithms*, Editör: Slowik A., CRC Press, Boca Raton, 17-30, 2020.

28. Kiran M.S., Findik O., A directed artificial bee colony algorithm, *Applied Soft Computing*, 26, 454-462, 2015.
29. Akay B., Karaboga D., Parameter tuning for the artificial bee colony algorithm, *International conference on computational collective intelligence*, Wroclaw-Poland, 608-619, 5-7 October, 2009.
30. Karaboga D., Basturk B., On the performance of artificial bee colony (ABC) algorithm, *Applied soft computing*, 8 (1), 687-697, 2008
31. Karaboga D., Akay B., A comparative study of artificial bee colony algorithm, *Applied mathematics and computation*, 214 (1), 108-132, 2009.