



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BÜTÜNLEME SINAVINA GİRECEK
ÖĞRENCİ SAYILARININ AŞIRI ÖĞRENME
MAKİNESİ TABANLI YAKLAŞIMLAR İLE
TAHMİN EDİLMESİ

Eyüp SİRAMKAYA

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Temmuz-2022
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Eyüp SİRANKAYA tarafından hazırlanan “Bütünleme Sınavına Girecek Öğrenci Sayılarının Aşırı Öğrenme Makinesi Tabanlı Yaklaşımlar İle Tahmin Edilmesi” adlı tez çalışması .../.../... tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof.Dr.Oğuz FINDIK

.....

Danışman

Prof.Dr. Mustafa Servet KIRAN

.....

Üye

Doç. Dr. Mesut GÜNDÜZ

.....

Üye

Doç. Dr. İsmail BABAOĞLU

.....

Üye

Doç. Dr. Mehmet Akif ŞAHMAN

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Eyüp SIRAMKAYA

03.06.2022

ÖZET

DOKTORA TEZİ

BÜTÜNLEME SINAVINA GİRECEK ÖĞRENCİ SAYILARININ AŞIRI ÖĞRENME MAKİNESİ TABANLI YAKLAŞIMLAR İLE TAHMİN EDİLMESİ

Eyüp SIRAMKAYA

**Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı**

Danışman: Prof.Dr. Mustafa Servet KIRAN

2022, 66 Sayfa

Jüri

Prof.Dr. Mustafa Servet KIRAN

Prof. Dr. Oğuz FINDIK

Doç Dr. Mesut GÜNDÜZ

Doç. Dr. İsmail BABAOĞLU

Doç. Dr. Mehmet Akif ŞAHMAN

Bütünleme sınavları genel sınavlar neticesinde başarısız olan öğrencilerin girebileceği bir ek genel sınavdır ve ülkemizde bu sınavlar farklı üniversitelerce uygulanmaktadır. Bütünleme sınavlarında elde edilen puan genel sınav puanı olarak değerlendirilmekte olup öğrencinin dersi başarıp başaramadığının değerlendirilmesi amacıyla genel sınavlardan sonra yapılmaktadır. Derslerden elde edilen notlara veya ağırlıklı not ortalamalarına bağlı olarak bir öğrenci, dersin bütünlemesine kalmasına rağmen bütünleme sınavına girmeyebilmektedir. Sınava girmeyecek öğrencilerin önceden belirlenmediği durumlarda bütünlemeye kalan tüm öğrencilerin sınava gireceği varsayımıyla sınav çizelgesi oluşturulmakta ve soru kâğıtları çoğaltılmaktadır. Bunun neticesinde ise hem emek israfı (gözetmenlerin sınavlara gönderilmesi) hem de kâğıt ve toner gibi kırtasiye malzemelerinin verimsiz kullanılması ortaya çıkmaktadır. Bu israfları önlemek amacıyla bu tez çalışmada her bir öğrenciye ait kişisel olmayan veriler Konya Teknik Üniversitesi Etik Kurulu'ndan alınan bir izinle toplanmış ve sınava girilmesine etki eden özellikler (mesafe, cinsiyet, genel ağırlıklı not ortalaması, dönem not ortalaması, dersin ara sınav notu, genel sınav notu vb.) çıkarılmıştır. Bunun sonucunda farklı derslere ait veriler kullanılarak veri setleri oluşturulmuş ve deneysel çalışmalarda kullanılmıştır. Bu veri kümelerindeki her bir kaydın bütünleme sınavına girip girmeyeceği sınıflandırma problemi olarak ele alınmış ve aşırı öğrenme makinesi tabanlı yaklaşımlar uygulanmıştır. Deneysel çalışmalarda ilk olarak problemin literatüre tanıtılması amacıyla temel aşırı öğrenme makinesi uygulaması yapılmıştır. Elde edilen sonuçlar sürü zekâsı algoritmaları ve hatanın geri yayılım algoritması ile eğitilmiş yapay sinir ağlarının eğitim ve test doğruluklarının karşılaştırması da bu çalışmada sunulmuştur. Elde edilen sonuçlar hatanın geri yayılım algoritması ile eğitimin yüksek başarı göstermesine karşın test başarısız da hem sürü zekâsı algoritmalarının hem de önerilen yaklaşımın daha iyi olduğunu göstermiştir. Daha sonraki deneysel çalışmalar ise yapay arı kolonisi ile optimize edilmiş aşırı öğrenme makinesinin bu veri setleri üzerindeki performans araştırmasını içermektedir. Bu çalışmada birden fazla aşırı öğrenme makinesinin gizli katman nöron sayısı ve kullanılan aktivasyon fonksiyonlarının belirlenmesi amacıyla yapay arı kolonisi algoritması önerilmektedir. Önerilen yaklaşımlar ve elde edilen sonuçlar genel olarak değerlendirildiğinde bütünlemeye girecek öğrenci sayısının belirlenmesi israfın önlenmesinde etkilidir ve bu sınıflandırma problemi için yapay sinir ağı tabanlı yaklaşımlar kabul edilebilir performans sunmaktadır.

Anahtar Kelimeler: Aşırı öğrenme makinesi, bütünleme, sınav, yapay arı kolonisi, yapay sinir ağı

ABSTRACT

PhD THESIS

PREDICTION OF THE NUMBER OF STUDENTS WHO WILL TAKE THE MAKE-UP EXAM BY EXTREME LEARNING MACHINE-BASED APPROACHES

Eyüp SIRAMKAYA

**Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering**

Advisor: Prof.Dr. Mustafa Servet KIRAN

2022, 66 Pages

Jury

**Prof.Dr. Mustafa Servet KIRAN
Prof. Dr. Oğuz FINDIK
Assoc. Prof. Dr. Mesut GÜNDÜZ
Assoc. Prof. Dr. İsmail BABAOĞLU
Assoc. Prof. Dr. Mehmet Akif ŞAHMAN**

Make-up exams are an additional general exam that students who fail as a result of the general exams take, and these exams are performed by different universities in our country. The score obtained in the make-up exams is evaluated as the general exam score and is made after the general exams in order to evaluate whether the student has succeeded in the course. Depending on the grades obtained from the courses or the weighted grade averages, a student may not take the make-up exam. In cases where the students who will not take the exam are not determined in advance, an exam schedule is created, and question papers are issued with the assumption that all students who fail the make-up exam will take the make-up exam. As a result of this, both the waste of labor (sending the examiners to the exams) and the inefficient use of stationery materials such as paper and toner occur. In order to avoid these wastes, in this thesis study, non-personal data (distance, gender, general weighted grade point average, semester grade point average, midterm exam grade of the course, general exam grade etc.) belonging to each student were collected with a permission from the Konya Technical University Ethics Committee and the features that affect taking the exam were extracted. As a result, data sets were created using data from different courses and used in experimental studies. Whether or not each record in these datasets will take the make-up test has been considered as a classification problem and extreme learning machine-based approaches have been applied. In the experimental studies, firstly, basic extreme learning machine application was made in order to introduce the problem to the literature. The results obtained are also presented in this study, comparing the training and test accuracies of artificial neural networks trained with swarm intelligence algorithms and back propagation algorithm. The results showed that although the training set with the back propagation algorithm showed high success, both the swarm intelligence algorithms and the proposed approach were better in test sets. An experimental study includes the performance research of the hyper learning machine optimized with artificial bee colony on these datasets. In this experimental study, it is aimed to determine the number of hidden layer neurons and the activation functions used in more than one extreme learning machine, and the artificial bee colony algorithm is recommended to handle this process. When the proposed approaches and the results obtained are evaluated in general, determining the number of students to be integrated is effective in preventing waste and artificial neural network-based approaches offer acceptable performance for this classification problem.

Keywords: Artificial bee colony, artificial neural network, exam, extreme learning machine, integration.

ÖNSÖZ

Çağımızın önemli problemlerinden bir tanesi kıt kaynakların kullanımındaki israftır. İsrâf tanım olarak eldeki bir şeyin gereksiz veya fazla kullanılmasıdır. Zaman, mekân, gıda ve eşya israfı günümüz toplumlarında önemli israf kalemlerini oluşturmaktadır. Bu noktada bütünleme sınavına kalan bir öğrenci için yapılan hazırlıklar zorunlu olarak görülmesine rağmen öğrenci bütünlemeye girmediği takdirde onun için yapılan hazırlıklardan kâğıt ve toner eşya israfına, fazladan atanan gözetmen zaman israfına, sınıf mekân israfına neden olmaktadır. Bu tez çalışmasında bu israfın minimizasyonu amacıyla bütünleme sınavına giriş bir sınıflandırma problemi olarak ele alınmış ve yapay sinir ağı tabanlı yaklaşımlar bu problemin çözümü amacıyla uygulanmıştır. Bu sayede bütünleme sınavı hazırlıklarında girecek öğrenci sayısının belirlenmesi ile çeşitli israfların önlenmesi hedeflenmiştir.

Tez çalışması sırasında Konya Teknik Üniversitesi Etik Kurulu'ndan 04/12/2020-E.16320 sayılı izinle Konya Teknik Üniversitesi Mühendislik ve Doğa Bilimleri Fakültesi Bilgisayar Mühendisliği Bölümü'nde okutulan 9 adet dersin verileri toplanmış ve 12/12/2019-E18053 sayılı izni ile de işlenmiştir. Bu dersler önceki yıllarda okutulduğundan dolayı veri kümeleri, bütünleme sınavına kalan ve giren, kalan ve girmeyen öğrencilere ait verileri içermekte olup sentetik veya hipotetik veri içermemektedir.

Verilerin toplanmasına izin veren ilgili kurula, verilerin toplanmasında yardımcı olan ilgili üniversitenin Bilgi-İşlem Daire Başkanlığı personeline teşekkür ederim. Tez çalışmam süresince farklı konularda yardımcı olan Bilgisayar Mühendisliği öğretim elemanlarına ayrıca minnettarım. Tezde kullanılan uygulamaların yazımında tecrübesi ve birikimini esirgemeyen Yazılım Mühendisliği Bölümü öğretim üyesi Dr. Öğretim Üyesi Engin EŞME'ye özellikle teşekkürlerimi sunarım. Tezin başından sonuna sürdürülmesinde yardımcı olan danışmanım Prof. Dr. Mustafa Servet KIRAN'a, tez izleme kurulu üyelerim Doç.Dr. Mesut GÜNDÜZ ve Doç.Dr. Mehmet Akif ŞAHMAN'a ve tez savunma jüri üyelerim Prof.Dr. Oğuz FINDIK ve Doç.Dr. İsmail BABAOĞLU'na teşekkürü bir borç bilirim.

Doktora tezimi, danışmanım Prof. Dr. Mustafa Servet KIRAN, kızım Ayşe ve oğlum Murat Gökhan'a ithaf ediyorum...

Eyüp SİRANKAYA
KONYA-2022

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	viii
1. GİRİŞ	1
1.1. Tezin Amacı.....	1
1.2. Tezin Önemi	3
1.3. Kaynak Araştırması	6
1.3.1. ELM’lerde Kullanılan Öğrenme Teorileri	11
2. MATERYAL VE YÖNTEM.....	19
2.1. Yapay Sinir Ağı	19
2.2. Aşırı Öğrenme Makinesi.....	20
2.3. Yapay Arı Kolonisi Algoritması.....	22
2.4. YSA Modeli ve Eğitim Algoritmaları	24
2.5. ABC ile optimize edilmiş Aşırı Öğrenme Makinesi.....	28
2.6. Veri Kümelerinin Özellikleri	28
2.7. Karışıklık Matrisleri.....	30
3. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	34
3.1. Aşırı Öğrenme Makinesi ile Edilen Sonuçlar ve Karşılaştırmalar	34
3.2. ABC ile Optimize Edilmiş Aşırı Öğrenme Makinesi ile Edilen Sonuçlar ve Karşılaştırmalar.....	37
3.2.1. Yaklaşımın Performans Analizi ve Karışıklık Matrisleri	42
4. SONUÇLAR VE ÖNERİLER	47
4.1 Sonuçlar	47
4.2 Öneriler	49
KAYNAKLAR	50

SİMGELER VE KISALTMALAR

Simgeler ve Kısaltmalar

BP	: Geri yayılım (Back Propagation)
SVM	: Destek vektör makinesi (Support Vector Machine)
RBF	: Radyal tabanlı fonksiyon (Radial Basis Function)
ELM	: Extreme Learning Machine
k-ELM	: k - Extreme Learning Machine
SLFN	: Single Layer Feed Forward Network
YSA	: Yapar Sinir Ağları
I-ELM	: Incremental Extreme Learning Machine
OP-ELM	: Optimally Pruned Extreme Learning Machine
FNN	: Fuzzy Neural Network
LS-SVMs	: Least Square Support Vector Machines
RKS	: Random Kitchen Sinks
LMS	: Least Mean Square
PSO	: Particle Swarm Optimization
GWO	: Gray Wolf Optimizer
ACO	: Ant Colony Optimization
ES	: Evolution Strategy
GA	: Genetic Algorithm
PBIL	: Population-Based Incremental Learning
ABC	: Artificial Bee Colony
BBO	: Biogeography-Based Optimization
TSA	: Tree Seed Algorithm
DNO	: Dönem Not Ortalaması
NO	: Not Ortalaması
ÖADS	: Öğrencinin Aldığı Ders Sayısı
VN	: Ara Sınav Notu
FN	: Final Notu
ADS	: Alttan Alınan Ders Sayısı
DO	: Öğrencinin Dönemi
KU	: Konyaya Uzaklık
C	: Cinsiyet
BSS	: Bütünleme Sınavı Sayısı
IoT	: Internet of Things (Nesnelerin İnterneti)
KELM	: Kernel Extreme Learning Machine

1. GİRİŞ

Zamanımızın en büyük sorunlarından biri de elimizdeki kaynakların hor kullanılması ve israfıdır. Kişisel olarak alınacak ufak tedbirler ya da küçük tasarruflar, aile, şehir, ülke ve dünya çapında ciddi miktarda gereksiz kaynak kullanımını engellediği yadsınamaz bir gerçektir. Bu tezde güdülen amaç ve gereksiz kaynak israflarından bir tanesi için uygulanan algoritmanın önemi bu bölümde verilecektir.

1.1. Tezin Amacı

Üniversitelerimizde her dönem uygulanan bütünleme sınavlarında ders sorumlusu öğretim elemanları, bütünleme sınavına girecek öğrenci sayılarını sınav öncesinde öngöremeyerek bütünleme sınavına girme hakkı olan tüm öğrenciler için sınav evrakı çoğaltmaktadırlar. Ancak, bütünlemeye kalan öğrencilerin bir kısmı, dersi şartlı geçmesi, yaşadığı şehrin okuduğu şehre olan uzaklığı, öncesinde girdiği başka bir bütünleme sınavından yüksek not almasıyla şartlı geçer duruma gelmesi gibi sebeplerden dolayı bütünleme sınavına girmemektedirler. Bu ve başka sebeplerle bütünleme sınavına girmeyen öğrenciler için çıktısı alınan sınav evrakları kâğıt, toner ve elektrik israfına neden olmaktadır. Ayrıca sınav programları hazırlanırken sınavda görevli gözetmenler için sınav görevlendirmeleri sınav hakkı olan öğrenci sayıları gözetilerek yapılmaktadır. Söz konusu akademisyenler bu görevleri için yapmakta oldukları çalışmalarına, muhtemel araştırmalarına ara vermektedirler. Yani akademik çalışmaları sekteye uğrayabilmektedir.

Tezde hedeflenen amaç üniversitelerimizde ders veren öğretim elemanlarımıza sorumlusu oldukları derslerin bütünleme sınavlarında, doğru bir tahminleme ile ilgili dersin bütünleme sınavına girmesi muhtemel öğrenci sayılarını sınav öncesinde sunmaktır. Bu sayede amaçlanan öncelikli olarak maliyet düşürerek israfı minimuma indirerek ulusal çapta bir katkı sağlamaktır. Ayrıca daha az kâğıt kullanılmasıyla çevre korumaya da katkı sağlamak diğer bir amaçtır. 1 ton kâğıt hamurundan fotokopide kullanılan 200.000 sayfa A4 kâğıdı elde edilmektedir. Bu da 80 koli A4'e denk gelmektedir ve bunun için de 20 tane ağaç kesilmek durumundadır, ayrıca ülkemizde yılda kâğıt, karton ve yaşamsal ihtiyaçlarımızı karşılamak amacıyla kişi başı ortalama 7 tane ağaç kullanılmaktadır. Bu da her bir bireyin yılda doğaya en az 7 fidan dikmesi gerektiği anlamına gelmektedir. Çevre ve Orman Bakanlığı gereksiz yapılan yazışmalar yüzünden her yıl 2 milyon ağacın harcandığını vurgulamaktadır (Avansas). Bir ofis çalışanı yılda ortalama 80 kg kâğıdı çöpe atmaktadır. 80kg kâğıt 1 ağaçtan elde edilen

miktara eşittir ve bu da gereksiz alınan her sınav evrakının önemini daha da ön plana çıkarmaktadır. 1 ağacın doğaya sağladığı oksijen kazanımını düşünülürse, 1 hektarlık bir alanda bulunan geniş yapraklı ağaçlardan oluşan bir ormanlığın 16 ton, iğne yapraklı ağaçlardan oluşan ormanlığın ise 30 ton oksijen ürettiği bilinmektedir. 1 hektar ormanlık alanda ortalama 10.000 ağaç bulunmaktadır ve bu 10.000 ağacın ürettiği oksijen tam 30.000 insanın oksijen tüketimine eşittir. Dünyada kağıt tüketen herkes tükettikleri kağıt miktarını yarı yarıya azalttığı düşünülduğünde bunun tam 8 milyon hektarlık ormanın koruyacağı ortadadır. Bu rakamlar göz önüne alındığından yapılacak tasarruf insanlığa ciddi faydalar sağlayabileceği ortadadır. Ağaçlarımızın hava kirliliğini önlemedeki önemini unutmamak gerekir, 24 saatlik zaman diliminde 1000 kişinin atmosfere çıkardığı karbondioksiti yetişkin bir kayın ağacının yok edebildiği bilinmektedir. Ayrıca bu yetişkin kayın ağaçları yılda ortalama 300kg zehir ve 7kg tozu da süzebilmektedir. Buna bağlı olarak ormanlarımız sayesinde hava kirliliğinin %50 si temizlenmektedir. Ayrıca ağaçlarımızın ülkemizde sıkça görülen doğal felaketlerden biri olan erozyonu da engellediği bir gerçektir, 100 yaşına gelmiş 25 mt boyu olan bir kayın ağacını 10 m2 içerisindeki 30.000 lt suyu emebilmekte ve bu da oluşabilecek toprak kaymalarının önüne geçerek verimli toprakların korunmasına yardımcı olmaktadır. Tez kapsamında maliyete etki eden elektrik ve toner masrafları, emek israfının önlenmesi amacıyla sınava görevlendirilen öğretim elemanı sayısının azaltılması gibi amaçların yanı sıra doğru tahminlemeyle azaltılması planlanan her bir kâğıdın aslında çevreye doğrudan katkısı olduğu da bir gerçektir (Avansas).

Ulusal kazanımda Konya Teknik Üniversitesi'nden yola çıkılarak yapılan hesaplamalarda sınavlarda kullanılan sınav evraklarının tek sayfadan oluştuğu varsayılmıştır. Ancak bilindiği gibi günümüz sınavlarında birçok öğretim elemanının hazırlamış oldukları sınavlar birden çok sayfadan oluşmaktadır. Yani yapılan hesaplardan çok daha fazla bir kazanım olacağı muhakkaktır. Bu kazanımlar ise kâğıt, toner ve elektrik için maliyetin minimizasyonu, çevreye fayda ve emek israfının önlenmesi olarak sıralanabilir.

Bu tez çalışmasının amacı, makine öğrenmesi ve evrimsel hesaplama algoritmalarıyla sunacağımız makine öğrenmesi sistemleri ile bütünleme sınavına girecek olan öğrenci sayıları için tahminleme yapmaktır. Bu bağlamda kullanacak metotlar, öğrenci verileri ile ilgili bilgiler toplamakta, genellemeler yapmakta ve daha sonra hiç görmediği yeni öğrenci verileri ile karşılaştırdığında öğrendiği bilgileri kullanarak yeni veriler hakkında tahmin yapmaktadır. Her ders özelinde, dersi alan öğrencilerin yukarıda

bahsedilen kriterleri göz önünde bulundurularak geçmiş yıllardaki öğrencilerin tutumları ile eğitilen sistem her bir ders için en doğru tahminlemeyi yapmak ve dersin öğretim elemanına bütünleme sınavına girmesi muhtemel öğrenci sayısını sunması amaçlanmaktadır. ELM(Extreme Learning Machine – Aşırı Öğrenme Makinesi) ve YSA(Yapar Sinir Ağları)’nın kullanılma sebebi, öğrenebilme ve genelleylebilme özellikleri sebebiyle günümüzde birçok bilim alanında geniş uygulama olanağı bulmakta ve karmaşık problemleri başarıyla çözebilme yeteneğine sahip algoritmalar oluşlarıdır. Ayrıca evrimsel hesaplama ve programlama tabanlı yaklaşımların da probleme uygulanması, ilk ve özgün bir katkı sağlamak amacı ile yapılmıştır. Bu yöntemler ile öğrenci davranışı modellenmeye çalışılmakta ve çıktı olarak herhangi bir dersin bütünleme sınavına girecek öğrenci sayısı tahmin edilmektedir.

1.2. Tezin Önemi

Bütünleme sınavı (ek genel sınav) Türkiye’de birçok yükseköğretim kurumunda uygulanmakta olan bir sınav türüdür ve bu sınava genel (final) sınavlar neticesinde başarısız olan öğrenciler girebilmektedir. Öğrencilerin bütünleme sınavına girmesi zorunlu değildir ve final sınavları neticesinde öğrencilerin ilgili dersten geçmesini sağlayacak farklı başarı kriterleri bulunmaktadır. Örneğin bazı üniversitelerde dönem ağırlıklı not ortalaması 2 ve üzeri olan öğrencilerin ilgili dersten DD (Şartlı Geçer) veya DC (Şartlı Geçer) notunu almış olması dersin başarıldığı anlamına gelmektedir. Ayrıca verilerin alındığı Konya Teknik Üniversitesi’nde ise transkriptinde 6 tane DC notu olan öğrenciler diğer başarı kriterlerini sağladıysa bu derslerden başarılı sayılmaktadırlar. Bunlara ek olarak dersten direkt olarak kalan öğrenciler de bütünleme sınavlarına çeşitli içsel ve dışsal motivasyonlar neticesinde girmeyebilmektedirler. Fakat öğrenci işleri otomasyon sistemlerinde sınava girecek öğrenci sayısı olarak dersten direkt kalmış ve şartlı geçer durumundaki öğrencilerin tümü listelenmektedir. Bu noktada dersi yürüten öğretim elemanları sınav evraklarını otomasyondaki sayıya göre çoğaltmakta olup bu da toner, kağıt, elektrik ve emek israfına neden olmaktadır. Bu israflardan tasarruf edilmesi amacıyla öğrencilerin ders notları, aile ev adreslerinin il-il mesafesi, öğrencinin geçmiş dönem başarısı, vize, final vb. notları gibi öğrenciye ait bilgiler ile dersin kredisi, ortalaması, dersi alan öğrencisi sayısı, dersten bütünlemeye kalan öğrenci sayısı gibi derse ait bilgiler dikkate alınarak makine öğrenmesi yöntemleriyle sınıflandırma ve

evrimsel hesaplama tabanlı modelleme ile bu sınavlara girecek öğrenci sayılarının tespiti üzerine çalışmalar yürütülecektir. Doğru bir tahminleme neticesinde kâğıt, toner ve diğer israf edilen kaynakların minimizasyonu öngörülmektedir. Toner ve kâğıt masrafı ve israfı doğrudan hesaplanabilir olmasından dolayı ulusal kazanım ve tezin önemini vurgulamak üzere Selçuk Üniversitesi (S.Ü.) örnek üniversite olarak seçilmiş ve bazı hesaplamalar yapılmıştır. S.Ü. ülkemizin en çok öğrencisi olan üniversitelerinden birisidir ve mühendislik, tıp, diş, turizm, hukuk, iletişim, teknoloji, veterinerlik, ziraat, İİB, eğitim, sanat, spor, MYO'lar gibi bünyesinde çok fazla fakülte ve yüksekokul bulundurmaktadır. Ayrıca konum olarak da Anadolu'nun merkezi konumunda olduğu söylenebilir. Bunun neticesinde de ülkemizin her yerinden öğrencisi olan, ülkenin mozağini gösteren bir üniversitemizdir. S.Ü.'de 2015–2018 yılları arasında final/bütünleme sınavı yapılan toplam öğrenci sayıları, bütünleme sınavına girme hakkı olan öğrenci sayıları, bütünleme sınavına giren öğrenci sayıları 09/06/2022-E301179 sayılı belge ile elde edilmiştir. Bu sayılar Çizelge 1.1.'de verilmiştir.

Çizelge 1.1. Selçuk Üniversitesi Bütünleme Sınavına Giren/Girmeyen Öğrenci Sayıları

YIL	Toplam Sınav Yapılan Öğrenci Sayısı	Bütünleme Hakkı Olan Öğrenci Sayısı	Bütünlemeye Girmeyen Öğrenci Sayısı
2018	871.301	265.951	111.510
2017	950.038	301.928	126.415
2016	951.373	341.534	135.495
2015	882.836	300.293	107.464

Çizelge 1.1.'de de görüleceği gibi bütünlemeye girme hakkı olduğu halde bütünleme sınavına girmeyen öğrencilerin oranı yaklaşık %39,81'dir. Bu da bütünleme sınav döneminde çıkartılan sınav evraklarının yaklaşık %40'ının kullanılmadığını gösterir. Bu yaklaşımdan yola çıkarak S.Ü. bünyesinde sadece son 4 yılda 480.884 adet sınav evrakı kullanılmadan atıl olmuştur. Ülkemizde sık kullanılan yazıcı, fotokopi cihazı markaları ve piyasadaki A4 fotokopi kağıdı fiyatları dikkate alındığında yaklaşık olarak bir sayfanın toner ve kâğıt masrafı 0,0822 TL (1,3237 Cent) civarında olmaktadır. Bu hesaplama Mart 2020 tarihinde yapılmıştır. Bu bilgiler ışığında 4 yıllık bütünleme sınavlarındaki Selçuk Üniversitesi'nin israfı;

$$480884 \text{ (syf)} \times 1,3237 \text{ (Cent/syf)} = 6.365,46 \text{ USD (4 Yıl için)}$$

$$480884 / 10000 = 48,09 \text{ Adet Ağaç}$$

olarak hesaplanmaktadır.

Bu hesap ülke geneli için oluşturulmaya çalışılırsa, Yüksek Öğretim Kurulu'nun kendi verilerinden sağladıkları rapor sayfasından alınan öğrenci sayıları Çizelge 1.2'de verilmiştir.

Çizelge 1.2. Yüksek Öğretim Kurulu'na göre Öğrenci Sayısı

YIL	Toplam Öğrenci Sayısı	Lisans Öğrenci Sayısı	Ön Lisans Öğrenci Sayısı
2018	7.010.598	4.241.841	2.768.757
2017	6.627.505	4.071.579	2.555.926
2016	6.186.007	3.900.601	2.285.406
2015	5.642.562	3.628.800	2.013.762

Örnek olarak kullanılan S.Ü.'den yola çıkarak sadece 2015-2018 yılları için tüm üniversitelerimizde 25.466.472 öğrenci sınava girmiş ve bunların yaklaşık %33,05'i bütünlemeye kaldığı varsayılmaktadır. Aynı şekilde S.Ü.'den yola çıkarak bu öğrencilerin %39,81'inin bütünleme sınavına girmediği öngörüldüğünde, bu da yaklaşık olarak 3.350.702 öğrencinin olduğu anlamına gelmektedir. Bunun yaklaşık maliyeti ise;

$$3.350.702 \text{ (syf)} \times 1,3237 \text{ (Cent/syf)} = 44.353,24 \text{ USD}$$

$$3.350.702 / 10000 = 335,07 \text{ Adet Ağaç}$$

olarak hesaplanabilir.

Yukarıdaki hesaplamalar ışığında görülmektedir ki sadece kağıt ve toner israfı yüksek öğretim kurumları için ayrı bir gider kalemi oluşturmaktadır. Bunun sıfıra indirgenmesi mevcut sınav sistemi ve mevzuat kapsamında imkansız olmasına rağmen öğrenci işleri otomasyon sistemlerinin biraz akıllı davranacak şekilde geliştirilmesi ile minimizasyonu oldukça mümkün görünmektedir. Çizelge 1.1 dikkate alındığında bütünleme sınavına girme hakkı olan öğrencilerin %35 ile %42 arasında bir sınava girmeme söz konusu olmasına ve bu istatistik üzerinden girmeyecek öğrenci sayısına yönelik bir varsayım oluşturulabilmesine rağmen bu oranlar ders bazında dikkate alındığında yıl bazında %20 ile %50 arasında değişebilmekte ve istatistiki açıdan bir

değerlendirme yapılabilmesi zorlaşmaktadır. Bu kapsamda tarihsel verinin kullanımı ile makine öğrenmesi yöntemlerinden faydalanılarak daha doğru bir tahmin yapılması öngörülmektedir. Tez önerisinde makine öğrenmesi ve evrimsel hesaplama yöntemleri ile bütünleme sınavına girecek öğrencilerin tespit edilmesi amacıyla araştırma ve geliştirme yapılacak ve prototip ürün geliştirilecektir. Geliştirilecek prototip ürün sınıflandırma ve tahmin motorlarını içerecek şekilde yapılandırılacak ve gerekli verinin tedarik edildiği bütün tahmin ve sınıflandırma uygulamaları için modüler bir yapıda olacaktır.

Tezin maliyet açısından önemi vurgulandıktan sonra ülkemizin 11. Kalkınma Planı doğrultusunda Yapay Zekâ kritik alanında araştırmacı yetiştirme ve ürün geliştirme konusu da dikkate alınmalıdır. İlgili alanda bir doktora öğrencisinin tez çalışmasına katkıda bulunulacak ve bu da tezin bir diğer açıdan önemini arttıracaktır.

1.3. Kaynak Araştırması

Bu tez çalışmasında giriş bölümünde anlatılan olumsuzlukları azaltmak amacıyla yapay sinir ağı tabanlı çözümler önerilmiştir. İlk önerilen çözüm iyi bir genelleştirme kabiliyetine sahip olması, son derece (extreme) hızlı öğrenme süreci ve iteratif olmayan eğitim süreci avantajlarından dolayı aşırı öğrenme makinesinin, kısaca ELM algoritması, tez çalışmasında ele alınan sınıflandırma problemlerine uygulanmasıdır (Kiran ve ark., 2022). İkinci çalışma ise ABC ile optimize edilmiş k-ELM (k - Extreme Learning Machine – k-Aşırı Öğrenme Makinesi) adı verilen oylama tabanlı bir versiyonun yapay sinir ağının eğitimi için geliştirilmiş ve probleme uygulanmıştır (Kiran ve ark., 2022). Tez kapsamında önerilen yöntemler ELM ile yapay sinir ağının eğitimi üzerine yoğunlaştığından dolayı literatür özeti de ELM algoritması üzerine yoğunlaşmıştır.

ELM, Huang ve ark. (2004b) 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 hesaplaması için Moore-Penrose genelleştirilmiş ters matris

yaklaşımını önermişlerdir (Moore, 1920; Penrose, 1955; Huang ve ark., 2006b). ELM'ler orijinalinde SLFN(Single Layer Feed Forward Network – Tek Gizli Katmanlı İleri Beslemeli Yapay Sinir Ağı)'ler için geliştirilmiştir. Daha sonra genelleştirilmiş SLFN'lere genişletilmiştir. ELM, özünde genel eğitim anlayışından farklı olarak SLFN'lerin gizli katman ayarlamasına ihtiyaç duymamaktadır. ELM gizli katmandaki düğümlere rastgele değerler vererek uygulanmaktadır. Sinir ağlarında uygulanan geleneksel eğitim algoritmalarından farklı olarak ELM, sadece en küçük eğitim hata oranıyla kalmaz aynı zamanda çıktı ağırlıklarının da en küçük normda olmasını sağlamaktadır. YSA teorisine göre bir ileri beslemeli ağda daha küçük eğitim hata oranına sahip ve çıkış ağırlık normları ne kadar küçükse genelleştirme performansı o kadar iyidir. Bu sebeple ELM'nin çok daha iyi bir genelleştirme performansına sahip olduğu söylenebilir. ELM'de gizli katman ağırlıkları rastgele verilip daha sonra tekrar ayarlanmadığından, çıktı ağırlıkları lineer bir denklemle çözümlenebilmektedir.

Literatürde ileri beslemeli yapay sinir ağlarının eğitiminde sıklıkla Hatanın Geriye Yayılımı Algoritması (BP) kullanılmaktadır (Rumelhart ve ark., 1986). Geleneksel BP (Back Propagation – Hatanın Geriye Yayılma Algoritması) algoritması, parametreleri optimize etmek için kullanılabilecek birinci dereceden bir gradyan yöntemi olarak kabul edilir. Ancak local minimum ve yavaş yakınsama sorunları vardır. Araştırmacılar, FNN(Fuzzy Neural Network - Bulanık Sinir Ağı) eğitiminde optimallliği ve verimliliği artırmak için alt küme seçim metodu (Li ve ark., 2005; Chen ve ark., 2012), ikinci dereceden optimizasyon yöntemleri (Hagan ve Menhaj, 1994; Wilamowski ve Yu, 2010), global optimizasyon yöntemleri (Yao, 1993; Branke, 1995) gibi farklı teknikler önerdiler. Artırımlı aşırı öğrenme makinası, I-ELM(Incremental Extreme Learning Machine - Artımlı Aşırı Öğrenme Makinesi), gizli düğümleri rastgele üretir ve SLFN'nin çıkış ağırlıklarını analitik olarak hesaplar. Ancak I-ELM yeni bir düğüm eklendiğinde mevcut düğümlerin ağırlıklarını yeniden hesaplamaz (Huang ve ark., 2006a). Bundan dolayı, konveks artırımlı aşırı öğrenme makinesi ile rastgele yeni bir düğüm eklendiğinde mevcut düğümlerin ağırlıklarının yeniden hesaplanarak daha da iyileştirilebileceğini (Huang ve Chen, 2007b) çalışmasında gösterilmiştir. Miche ve ark. (Miche ve ark., 2010) optimal budanmış ELM (OP-ELM - optimally pruned extreme learning machine - Optimal Budanmış Aşırı Öğrenme Makinesi) yöntemini önermişlerdir. Bu yöntem klasik ELM'yi daha sağlam ve genel bir hale getirecek adımlar içermektedir. Yazarlar ELM'yi SVM(Support Vector Machine – Destek Vektör Makineleri) 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 ve ark. (Huang ve ark., 2012) 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 ayırık bölgeyi sınıflandırabildiğini göstermişlerdir (Huang ve ark., 2012). 2013 yılında Zong ve ark. (Zong ve ark., 2013) genelleştirilmiş tek katmanlı ileri beslemeli ağlar için hem 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 (Balasundaram ve ark., 2014). 2016'da Musikawan ve ark. (Musikawan ve ark., 2016) 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 (Zong ve Huang, 2011b; Zong ve ark., 2011) çalışmalarında ELM uygulamışlardır. Ayrıca 2011 yılında Mohammed ve ark. (Mohammed ve ark., 2011) 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 (Ucar, 2014) uygulanmıştır. Ray ve Mishra (2015) yeraltı kablolarındaki arıza tespiti için s-dönüşümü ve ELM'yi birleştirmek suretiyle hibrit bir metot 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 (Sokolov-Mladenovic ve ark., 2016). Kuang ve ark. (2017) alt ekstremitte hareketlerini doğru ve hızlı bir şekilde tanımlamak için, aşırı öğrenme makinesine dayalı bir tanıma yöntemi önermişlerdir. 2018 yılında ise lityum-iyon pillerin kalan ömürlerinin tahmini için (Razavi-Far ve ark., 2018) ve bitki türü tespiti için (Turkoglu ve Hanbay, 2019) ELM kullanılmıştır.

Birçok farklı ELM versiyonu ve uygulaması literatürde bulunmasına karşın V-ELM olarak isimlendirilen çoklu ELM çalışması ve analizi bir başka çalışmada (Cao ve ark., 2012) ortaya konulmuştur. İlgili çalışmada oylamaya katılacak ELM sayısı ve

sınıflandırma performansları analiz edilmiştir. Oylamaya katılacak ELM sayısının 7 olduğunda kabul edilebilir ve performanslı çözümlerin elde edildiğini göstermişlerdir ve V-ELM’de oylamaya katılacak ELM’lerin özdeş olduğu varsayımı yapılmıştır (Cao ve ark., 2012). Bu tez kapsamında önerilen 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 (Artificial Bee Colony - ABC) algoritması kullanılmıştır (Karaboga ve Basturk, 2007; Eke ve ark., 2011; Ozturk ve ark., 2014).

BP algoritmasına kıyasla daha iyi genelleme performansı ve daha hızlı eğitim hızı sergilemesine rağmen, bu yöntemlerin çoğu hala küresel bir optimal çözümü garanti edemez. ELM’nin, tek gizli katmanlı ileri beslemeli sinir ağlarını (SLFN’ler) eğitmek için kullanılabileceği öne sürülmüştür. ELM’de, gizli düğümlerin değer ataması, yinelemeli ayarlama olmadan sabitlenmeden önce rastgele yapılır. Ayrıca, ELM’nin gizli düğümlerinin aynı nöron olmasına bile gerek yoktur. Öğrenmesi gereken serbest parametre, çıktı katmanı ile gizli katman arasındaki bağlantılardır (veya ağırlıklardır). Bu nedenle, ELM, nihayetinde doğrusal bir sistemin çözümüyle ilgilenen, parametre içinde doğrusal bir model olarak geliştirilmiştir. Geleneksel FNN öğrenme yöntemlerinden farklı olarak, ELM önemli ölçüde daha verimlidir ve küresel bir optimuma ulaşma eğilimi daha yüksektir. ELM’nin rastgele oluşturulmuş gizli düğümlerle çalışsa bile SLFN’lerin evrensel yaklaşım yeteneğini koruyabildiği teorik çalışmalarla gösterilmiştir (Huang ve ark., 2006a; Huang ve Chen, 2007a; 2008). Sık kullanılan aktivasyon fonksiyonları ile ELM, tüm parametreleri öğrendiği geleneksel FNN’nin neredeyse optimal genelleme sınırına ulaşma yeteneğine sahiptir (Liu ve ark., 2014). ELM’nin geleneksel FNN algoritmalarına göre genelleştirme ve verimlilik performansındaki avantajları, çeşitli alanlardan çok çeşitli problemlerde gözlemlenmiştir (Huang ve ark., 2006a; Huang ve ark., 2011). ELM’nin genellikle en küçük kare destek vektör makinelerine (LS-SVM’ler) (Suykens ve Vandewalle, 1999), destek vektör makinelerine (SVM’ler) (Cortes ve Vapnik, 1995) ve diğer gelişmiş algoritmalara kıyasla daha fazla verimliliğe sahip olduğu gözlemlenmiştir. Deneysel çalışmalar, ELM’nin genelleme yeteneğinin, SVM tabanlı sınıflandırma yöntemlerinden üstün olduğunu ortaya koymuştur (Huang ve ark., 2006a; Huang ve ark., 2011; Fernández-Delgado ve ark., 2014; Huang ve ark., 2014). Son onbeş yılda, ELM uygulamaları ve teorileri kapsamlı bir şekilde araştırılmıştır. Öğrenme verimliliği açısından, ELM’nin orijinal tasarımında yüksek öğrenme doğruluğu, en az insan katkısı ve hızlı öğrenme olmak üzere kritik faktörler öne çıkarılmıştır (Huang ve

ark., 2004b; Huang ve ark., 2006b). Orijinal ELM modeli, belirli uygulamalar için daha uygun ve verimli hale getirmek için çeşitli uzantılarla donatılmıştır. Huang ve ark. (2015) ELM hakkında geniş kapsamlı bir çalışma yapan bir inceleme yazısı yazdılar. Ayrıca (Huang ve ark., 2011), ELM'nin uygulamaları ve teorileri hakkında bir literatür araştırması da yaptılar. O zamandan beri, ELM hakkında daha aktif araştırmalar yapıldı. Teorik bir bakış açısından, ELM'nin evrensel yaklaşım yeteneği daha ayrıntılı olarak (Huang ve ark., 2011) araştırılmıştır.

Spesifik uygulama gereksinimlerini karşılamak için birçok ELM varyantı önerilmiştir. Örneğin kompakt bir ağa ihtiyaç duyan maliyete duyarlı öğrenmede test zamanı minimuma indirilmelidir ki test zamanı ihtiyaç duyulan seviyeye gelsin. Bu bağlamda ELM yüksek seviyede kompaktlık içeren ağlarda başarılı bir şekilde adapte olabilmektedir. (Deng ve ark., 2011; He ve ark., 2011; Martínez-Martínez ve ark., 2011; Yang ve ark., 2012; Du ve ark., 2013; Lahoz ve ark., 2013; Li ve ark., 2013; Yang ve ark., 2013; Bai ve ark., 2014; Wang ve ark., 2014). ELM uzantıları gürültülü/eksik verilerde (Miche ve ark., 2009; Man ve ark., 2011; Horata ve ark., 2013; Yu ve ark., 2013), çevrimiçi sıralı verilerde (Liang ve ark., 2006; Lan ve ark., 2009; Rong ve ark., 2009; Zhao ve ark., 2012; Ye ve ark., 2013), dengesiz verilerde (Zong ve Huang, 2011a; Horata ve ark., 2013; Huang ve ark., 2014), vb. başarıyla uygulanmıştır. Ayrıca, regresyon ve geleneksel sınıflandırma görevlerindeki kullanımlarının yanı sıra, ELM uygulamaları son zamanlarda özellik seçimi, kümeleme ve temsili öğrenmeye de başarıyla uygulanmıştır. (Benoît ve ark., 2013; Kasun ve ark., 2013; Huang ve ark., 2014). Şu da belirtilmelidir ki doğrusal olmayan özellik seçimi için ELM framework'ün rastgele stratejileri, makine öğrenmesi ve hesaplamalı zeka topluluğunda büyük ilgi çekmiştir (Rahimi ve Recht, 2007; 2008a; 2008b; Saxe ve ark., 2011; Le ve ark., 2013; Widrow ve ark., 2013). Bu yaklaşımların ELM ile yakın bir ilişkisi vardır ve birçok ortak özelliğe sahip oldukları için bir kısmı özel durumlar olarak kabul edilebilir. Örneğin, (Rahimi ve Recht, 2008b) gizli katmanının yapısını Fourier temeli ile sınırlayan özel bir ELM türü olan Rastgele Mutfak Evyelerini (Random Kitchen Sinks (RKS)) tanıttı. (Widrow ve ark., 2013)'de gösterilen No-Prop algoritması, ELM'ninkine benzer bir ruha sahiptir, Ancak, çıktı ağırlıklarının En Küçük Ortalama Kare (Least Mean Square - LMS) yöntemini kullanarak eğitir.

1.3.1. ELM'lerde Kullanılan Öğrenme Teorileri

Ara katmanlardaki ağırlık değerlerini bulma kabiliyeti (the Interpolation Capability) ve evrensel yaklaşım yeteneği Huang ve Ark. tarafından araştırılmıştır (Huang ve ark., 2006b; Huang ve Chen, 2008). L gizli düğüme sahip ileri beslemeli bir sinir ağının çıkış fonksiyonu şu şekilde ifade edilebilir:

$$f_L(x) = \sum_{i=1}^L \beta_i g_i(x) = \sum_{i=1}^L \beta_i G(a_i, b_i, x), \quad x \in R^d, \beta_i \in R^m \quad (1.1)$$

g_i , $G(a_i, b_i, x)$ çıkış fonksiyonunun i . gizli düğümünü belirtir. Aktivasyon fonksiyonu g olan ek düğümler için g_i şu şekilde tanımlanır:

$$g_i = G(a_i, b_i, x) = g(a_i \cdot x + b_i), \quad a_i \in R^d, b_i \in R \quad (1.2)$$

Ve g aktivasyon fonksiyonuna sahip RBF(Radial Basis Function – Radyal Temel Fonksiyonu) düğümleri için g_i şu şekilde tanımlanır:

$$g_i = G(a_i, b_i, x) = g(b_i \| x - a_i \|), \quad a_i \in R^d, b_i \in R^+ \quad (1.3)$$

Geçtiğimiz yıllarda, SLFN'lerin enterpolasyon ve evrensel yaklaşım yetenekleri kapsamlı bir şekilde araştırılmıştır. N adet rastgele farklı örneğin, N eşik gizli düğümü olan SLFN'ler tarafından kesin olarak öğrenilebileceği kanıtlanmıştır (Huang ve Huang, 1991; Satorri ve Antsaklis, 1991). Sonraki çalışmalardan (Huang ve Babri, 1998), SLFN'lerin enterpolasyon kabiliyeti hakkında daha eksiksiz ve en fazla N gizli düğümü olan ve bir sonsuzda bir limiti olan herhangi bir keyfi sınırlı doğrusal olmayan aktivasyon fonksiyonuna sahip bir SLFN'nin, herhangi bir N rastgele farklı örneği öğrenebileceğini sıfır hata ile kanıtladı. Bu tür aktivasyon fonksiyonları, eşik, rampa ve sigmoid fonksiyonlarının yanı sıra radyal temel, “kosinüs ezici (cosine squasher)” ve birçok düzensiz fonksiyonu içerir (Gallant ve White, 1988). Pek çok araştırmacı (Stinchcombe, 1989; Hornik, 1991) verilen herhangi bir sürekli hedef fonksiyona f , herhangi bir $\epsilon > 0$ hata oranı ile yaklaşan bir dizi network fonksiyonu $\{f_L\}$ içeren ve birçok basit koşulları sağlayan $g(x)$ aktivasyon fonksiyonu tatmin edici sonuçlar alındığını kesin olarak kanıtlamışlardır. Tüm bu geleneksel sinir ağı teorilerinde, ağ dizisinin herhangi bir $\{f_L\}$ fonksiyonuna ait tüm parametrelerin (örneğin gizli katman parametreleri (a_i, b_i) ve çıkış

ağırlıkları β_i) serbestçe ayarlanabilmesi gerekir. Bu geleneksel sinir ağı teorilerine göre, (a_i, b_i) gizli katman parametrelerinin uygun şekilde ayarlanması ve verilen herhangi bir hedef fonksiyon $\{f\}$ için ağ parametrelerinin (örneğin (a_i, b_i) ve β_i) uygun değerlerinin bulunması gerekir. Son yirmi yılda gizli katman parametreleri (a_i, b_i) 'yi ayarlamak için harcanan çabayı en aza indirmek için çalışıldı. Bazı araştırmacılar (Barron, 1993; Romero Merino, 2001), ağ dizisinin tüm $\{f_L\}$ 'lerinde gizli katmanların tüm parametrelerini ayarlamak yerine, yeni eklenen gizli düğümlerin parametrelerini ayarlayan ve daha sonra ayarladıktan sonra bunları düzelteren SLFN'ler için artımlı yöntemler önerdiler. Mevcut gizli düğümlerin parametreleri sabit kalacak ve sonraki öğrenme prosedüründe asla güncellenmeyecektir. Bu geleneksel öğrenme modellerindeki gizli katman parametrelerinin eğitim örneklerine göre en az bir kez ayarlanması gerekir. Buna karşılık, ELM'lerdeki gizli katmanın tüm parametrelerinin ayarlanması gerekmez ve eğitim örneklerinden bağımsız olabilir (Huang ve ark., 2006b; Huang ve Chen, 2008). ELM'lerin tipik uygulamalarından biri, ELM'nin gizli düğüm parametreleri (a_i, b_i) 'nin rastgele oluşturulabilmesidir. Aşırı öğrenme makinelerinin öğrenme yeteneği teorik açıdan enterpolasyon yeteneği, evrensel yaklaşım yeteneği ve ELM'nin genelleme sınırı olmak üzere 3 kısımda incelenmiştir (Huang ve ark., 2006b).

Liu ve ark. (2014) ELM'nin sınıflandırma görevlerinde genelleme performansı hakkında bir analiz sunmuştur. Regresyon görevleriyle ilgili olarak, ELM'nin genelleme yeteneği (Lin ve ark., 2014; Liu ve ark., 2014)'de kapsamlı bir şekilde incelenmiştir. Bu iki parçalı makalede, polinomlar, sigmoid ve Nadaraya-Watson fonksiyonları gibi bazı uygun aktivasyon fonksiyonlarıyla ELM'nin, tüm parametrelerin ayarlanabildiği bir SLFN ile aynı optimal genellemeyi başarabileceği gösterilmiştir. Lin ve ark. (2014) ayrıca Gauss tipi işlevler gibi belirli etkinleştirme işlevlerinin ELM'nin genelleştirme performansını düşürebileceğini de bulmuşlardır. Bununla birlikte, bu sorun, birden fazla ELM modelinin eğitilmesi veya düzenlileştirme tekniklerinin tanıtılmasıyla aşılabildiği gösterilmiştir (Huang ve ark., 2011; Lin ve ark., 2014).

Xi-Zhao ve ark. (2013), ilk yerelleştirilmiş hata modelini (LGEM) kullanarak ELM'nin genelleme yeteneğini araştırmışlardır. Bu çalışmada genelleme hatasının üç bölümden oluştuğu gösterilmiştir: eğitim hatası, stokastik duyarlılık ölçüsü ve bir sabit. Elde edilen sonuçlara dayanarak, yazarlar, ELM için gerçek veri setleri üzerinde tatmin edici sonuçlar veren otomatik model seçimi (gizli nöronların sayısını seçme) yöntemi için bir algoritma önermişlerdir.

Chen ve ark. (2014) "sıralamayı öğrenme" görevleri için kullanılan bir ELM varyantının genelleme sınırını oluşturdu. Makalelerinde, hipotez uzayının kapasitesini ölçmek için kaplama sayısı tanıtıldı.

(Rahimi ve Recht, 2007; 2008a), ELM rastgele özellik eşlemelerinden biri olan rasgele Fourier özellikleri üzerinde teorik bir analiz sağladı ve dönüştürülmüş verilerin iç çarpımının, özellik alanındaki bazı popüler çekirdek fonksiyonlarıninkilere düzgün bir şekilde yaklaşabileceğini gösterdi. Daha sonra, rastgele gizli katmana sahip SLFN'ler için bir genelleme test hatası sınırı verdiler (Rahimi ve Recht, 2008b). Rastgele SLFN'lerin ampirik riskinin, yakınsama oranı $O(1/\sqrt{N} + 1/\sqrt{L})$ olan bir fonksiyon ailesi tarafından elde edilen en düşük gerçek riske yakınsadığı kanıtlanmıştır.

2022'de Sahu ve Ark. insanlardaki beyin tümörünü tespit etme ve sınıflandırma için derin öğrenme metoduna dayalı bir ELM yöntemi kullanmışlardır. Ham verilere makine öğrenmesi yöntemleri uygulamışlar ve ardından sınıflandırma işlemi için ELM kullanmışlardır. Sahu ve Ark. çalışmalarında sınıflandırma doğrulukları için %98.09'luk bir yaklaşıklık üretmişlerdir (Sahu ve ark., 2022). Li ve Ark. 2022'de süper kapasitörlerin kapasitesini tahmin etmek için Sezgisel Kalman Filtresi algoritması ile birleştirilmiş bir ELM tahmin modeli önermişlerdir (Li ve ark., 2022). Önerdikleri bu tahmin modelinde araştırmacılar, geleneksel veriye dayalı tahmin modellerine kıyasla, ELM'nin hızlı hesaplama avantajı ile, çok daha iyi performans gösterdiğini ortaya koymuşlardır. Chakravarthy ve Rajaguru 2022 yılında yaptıkları çalışmalarında meme kanserinin erken tanı ve sınıflandırılması için basit karga arama algoritması kullanarak optimize edilen bir ELM önermişlerdir (Chakravarthy ve Rajaguru, 2022). 2022 yılındaki bir diğer tahmin çalışmasında ise Kaloop ve Ark. Uluslararası Pürüzlük İndeksi tahmini için hibrit bir ELM modeli önermişlerdir. Önerdikleri modelde Kaloop ve Ark. Optimal Pruned Extreme Learning Machine ve Wavelet Analizini entegre etmişlerdir. Önerdikleri model ile Uluslararası Pürüzlük İndeksini tahmin etme etkinlikleri açısından Optimal Pruned Extreme Learning Machine, geleneksel ileri beslemeli Yapay Sinir Ağları ve regresyon modeli ile istatistiksel olarak karşılaştırmışlardır. Bu model ile yalnızca %7 tahmin hatasıyla regresyona kıyasla sağlam ve daha doğru bir modelleme tekniği olduğunu ortaya koymuşlardır (Kaloop ve ark., 2022). Bir başka tahmin çalışmasında Gao ve Ark. 2022 yılında, ELM'yi büyük ticari binaların soğutma yükünü tahmin etmek için hibritleyerek kullanmışlardır. Gao ve Ark. büyük ticari binaların soğutma yükünü tahmin etmek için rastgele orman iyileştirme paralel balina optimizasyonu ile aşırı öğrenme makinelerine dayalı yapay sinir ağına dayalı hibrit bir tahmin modeli önermişlerdir (Gao

ve ark., 2022). 2018 yılında Chen ve Ark. cirRNA'lar ve IncRNA'lar arasında sınıflandırma yapmak için ELM tabanlı bir yaklaşım sundular. İlk önce bir özellik seçme işlemi uyguladıktan sonra sınıflandırma için hiyerarşik uç öğrenme makinesi kullandılar ve ikna edici sonuçlar elde ettiklerini iddia ettiler (Chen ve ark., 2018). Daha sonra 2020'de Niu ve Ark. cirRNA'ların tanınması için gelişmiş Aşırı Öğrenme Makinesine dayalı bir yöntem önerdiler. Yöntemlerinde ELM sınıflandırıcısının performansını yükseltmek amacıyla parçacık sürü optimizasyon algoritması uygulamışlardır (Niu ve ark., 2020). 2018'de You ve Ark. yaptıkları çalışmada protein ikincil yapısını tahmin etmek için farklı sınıflandırıcı türlerinin performanslarını deneyler ile karşılaştırmışlardır. Aldıkları sonuçlara göre ELM, en hızlı eğitim hızına sahip olmasına rağmen en doğru sonuçları verememiştir. En doğru sonuçları destek vektör makinesi ile elde etmişlerdir (You ve ark., 2016). Cao ve Ark. da aynı yıl yaptıkları çalışmada iyonik sıvıların toksisitesini tahmin etmek için çeşitli yaklaşımları karşılaştırmışlar ve ELM'nin en iyi performansa sahip olduğunu, çoklu doğrusal regresyon ve destek vektör makinesine üstünlük sağladığını iddia etmişlerdir (Cao ve ark., 2018). 2018 yılında bu kez de Kang ve Ark. Henry Kanunu Sabitindeki CO₂ miktarını tahmin etmek için ELM ve diğer yöntemleri uygulamışlardır. Kang ve Ark. göre ELM, çoklu doğrusal regresyon ve destek vektör makinesi de dahil olmak üzere birçok yöntem arasından en iyisi performansı elde etmiştir (Kang ve ark., 2018). 2019'da Lei ve Ark. protein-protein etkileşimlerini tahmin etmek için düzenleştirilmiş aşırı öğrenme makinesine dayalı bir tahmin yöntemi önerdiler ve umut verici performans elde ettiklerini iddia ettiler (Lei ve ark., 2018) daha sonra aynı konuda Li ve Ark. 2020'de Ağırlıklı Aşırı Öğrenme Makinesi yöntemlerini Protein-Protein Etkileşimlerini tahmin etmek için kullandılar (Li ve ark., 2020). Ouyang ve Ark. 2020 yılındaki çalışmalarında araçların egzozundaki NO_x ölçümünde ELM'yi uygulamışlar ve nitelikli deney sonuçları elde etmişlerdir (Ouyang ve ark., 2020). 2019'da Cai ve Ark. gri kurt optimizasyon stratejisine dayalı bir çekirdek aşırı öğrenme makinesi geliştirdiler. Amaçları stokastik davranışı artırmaktı. Deneysel sonuçları verimliliğini ve ilerlemesini göstermiştir (Cai ve ark., 2019). 2019'da Shouma ve Ark. kredi riski değerlendirmesinde birçok makine öğrenmesi modelinin karşılaştırmasını gerçekleştirdi ve ELM, bu çeşitli yöntemler arasında hızı ve basit oluşuyla ne kadar avantajlı olduğunu gösterdi (Shouma ve ark., 2019).

Son yıllarda makine öğrenmesi uygulamaları ulaştırma alanında oldukça popüler hale gelmiştir. Örneğin bilim adamları tehlikeli sürüşten kaçınmak ve insanların hayatlarını kurtarmak için sürücü uyumsuzluk izleme sistemi kurmak için makine

öğrenmesi tekniklerini kullanmışlardır (Jiang ve ark., 2020b). Sun ve Ark. taşıma sistemini optimize etmek için doğrusal programlama ve aşırı öğrenme makinesini birleştiren iki aşamalı bir yaklaşım önermişlerdir (Sun ve ark., 2011). Yaptıkları birleşik yaklaşımlarının ulaşım sisteminin ömrünü uzatabileceğini ve güvenilirliğini de artıracığını kanıtlamışlardır. 2016 yılında Liu ve Ark. gerçek zamanlı sürücü dikkat dağınıklığı tespiti için yarı denetimli yöntemler üzerinde çalıştılar ve laplacian destek vektör makinesini ve yarı denetimli aşırı öğrenme makinesi analiz ederek bu iki yarı denetimli yöntemin sürücü dikkat dağınıklığı algılama sistem oluşturmada etiketlenmiş eğitim verilerinin maliyetini azaltabileceği sonucuna vardılar (Liu ve ark., 2015). Liu ve Ark. 2017 yılında yol yüzeyi sıcaklığını tahmin etmek için aşırı öğrenme makinesine radyal temel işlevi uyguladılar ve umut verici deneysel sonuçlar elde ettiler (Liu ve ark., 2017a). Oneto ve Ark. büyük ölçekli demiryolu ağının dinamik gecikmesini tahmin etmek için ELM'yi uygulamışlardır (Oneto ve ark., 2017). Benzer şekilde Zeng ve Ark. da trafik işaretlerini derin özelliklerle tanımak için ELM'yi benimsemişlerdir (Zeng ve ark., 2016). Zhang ve Ark. video verilerine dayalı trafik kazalarını analiz etmek için ELM'yi kullanmışlardır (Zhang ve ark., 2017). 2018'de Wang ve Ark. taksi sürücülerinin en iyi rotaları aramasına yardımcı olmak için büyük ölçekli GPS verileriyle ELM'yi uyguladılar (Wang ve ark., 2017). Bu araştırmalara benzer şekilde bir çok bilim insanı sürücüler ve hükümetler için trafik akışını tahmin etmek amacıyla ELM'yi kullanmışlardır (Liu ve ark., 2018; Xing ve ark., 2018). Ve ELM ulaşım alanında popüler bir trend haline gelmiştir.

ELM, yüksek hızda eğitim avantajı sebebiyle robotik uygulamalarda da yaygın olarak kullanılmıştır. 2015 yılında Sun ve Ark. nesne kavrama algılamasını gerçekleştirmek için ELM tabanlı bir yöntem önerdiler. ELM'yi kademeli sınıflandırıcılarının temeli olarak uyguladılar ve özellik olarak yönlendirilmiş gradyanların histogramlarını benimsediler. Yöntemleri diğer kıyaslama yöntemlerinden daha iyi performans elde ettiğini gösterdiler (Sun ve ark., 2015). 2016'da Alçın ve Ark. robotik silah kontrolünde ELM'yi de uygulamış ve yaklaşımlarının yapay sinir ağı modelinden daha üstün olduğunu iddia etmişlerdir (Alcin ve ark., 2016). She ve Ark. 2018 yılındaki çalışmalarında elektroensefalografik sinyalleri sınıflandırmak için aşırı öğrenme makinesini kullandılar ve umut verici deneysel sonuçlar elde ettiler (She ve ark., 2018). 2019 yılında Duan ve Ark. robotik hareket kontrolünde dinamik sistemleri oluşturmak için ELM'yi uygulamışlardır. ELM'nin kendi yaklaşımlarına verimlilik ve

doğruluk özellikleri bakımından diğer yaklaşımlara göre daha iyi performans getirdiğini deneyleri ile iddia etmişlerdir (Duan ve ark., 2017).

Son yıllarda hem akademik hem de endüstriyel alanda sürekli ilgi çeken Nesnelerin İnterneti (IoT), giderek daha fazla bilim insanı, modern bilgi teknolojileri ile IoT’de çok sayıda yaklaşım veya uygulama geliştirildi (Jiang ve ark., 2020a). IoT uygulamalarında da ELM’yi uygulayan birçok yaklaşım bulunmaktadır. Rathore ve Ark. siber saldırıların tespiti için ELM tabanlı bir yaklaşım önerdiler. Araştırmacılar sis hesaplamaya dayalı bir saldırı algılama çerçevesi oluşturdular ve saldırıyı normal ziyaretten ayırt etmek için sınıflandırıcı olarak geliştirilmiş ELM kullanmışlar ve yaptıkları çalışmanın deneysel sonuçlarına göre önerdikleri bu yaklaşım başarılı sonuçlar elde etmiştir (Rathore ve Park, 2018). Yin ve Ark. mühendislerin hata düzeltme işlemleri için ELM tabanlı bir algoritma sundular. Sınıflandırıcı oluşturmak için çeşitlendirilmiş özellik seçimini aşırı öğrenme makinesiyle birleştirdiler ve umut verici sonuçlar elde ettiklerini iddia ettiler (Yin ve ark., 2018). Li ve Ark. 2019’daki çalışmalarında ortak entropi özelliklerini önermek için DDoS saldırısını analiz ettiler. DDoS saldırısının tespiti için aşırı öğrenme makinesini kullandılar. Deneysel sonuçlar, bu yöntemin verimli ve hızlı olduğunu kanıtladı (Li ve ark., 2019). 2020 yılında Zheng ve Ark. aşırı öğrenme makinesi de dahil olmak üzere birkaç yararlı şemayı birleştiren izinsiz giriş tespiti için yeni bir yaklaşım ortaya koydular. İlk olarak özellik boyutlarının azaltılması için geliştirilmiş lineer diskriminant analizini uyguladılar. Daha sonra yaklaşım boyutu azaltılmış verileri ELM optimizasyonu ile işlemek için sınıflandırıcı olarak tek gizli katmanlı ileri beslemeli sinir ağını kullandılar. Deneysel sonuçlar, yaklaşımlarının olağanüstü gerçek zamanlı özellikler ve genelleme performansı elde ettiğini gösterdi (Zheng ve ark., 2020).

ELM’nin son yıllarda uygulandığı bir diğer alan da coğrafyadır. Huang ve Ark. 2017’deki çalışmalarında heyelan duyarlılık indekslerini oluşturma için ELM’yi kullanmışlar (Huang ve ark., 2017). Çalışmalarında ilk olarak kendi kendini organize eden harita ağı ile kaba bir indeks haritası elde edildi ve çok düşük duyarlılığı olan alanlar kaldırıldı. Daha sonra, ELM kullanılarak nihai heyelan duyarlılık indeks haritası oluşturulmuştur. Wei ve Ark., 2019’daki çalışmalarında ELM kullanarak moloz yığını dalgakıranların stabilitesini değerlendirmeye çalıştılar (Wei ve ark., 2019b). Çalışmalarında test performansına göre en iyisini seçmek için 5000 farklı ELM’yi eğittiler. Wei ve Ark. 2019 yılındaki çalışmalarında tortu taşıma kapasitesini tahmin etmek için ELM’yi kullandılar (Wei ve ark., 2019a). Bu çalışmada araştırmacılar eğitim

için 30 set veri ve doğrulama için 4 set veri kullandılar. ELM'nin girdi vektörü, yüzey hızı, eğim ve çökme hızını içeriyordu ve çıktı, tortu taşıma kapasitesiydi. Niu ve Ark. 2019 yılındaki çalışmalarında çoklu doğrusal regresyon, yapay sinir ağı, SVM ve ELM'ye dayalı hidroelektrik rezervuarının çalışma kuralını türetmek için dört akıllı sistem geliştirdiler (Niu ve ark., 2019). Çalışmanın simülasyon sonuçlarına bakıldığında, yapay sinir ağı, SVM ve ELM'nin çoklu doğrusal regresyondan daha iyi performans gösterdiğini buldular. Li ve Ark. 2015 yılındaki çalışmalarında ise hiperspektral görüntü sınıflandırması için verimli bir yöntem sundular (Li ve ark., 2015). Aşırı öğrenme makinesi tarafından oluşturulan sınıflandırıcıyı kullanarak, diğer geleneksel yöntemlerden daha iyi umut verici sonuçlar elde ettiklerini çalışmalarında sundular.

ELM'nin gıda güvenliği alanında da oldukça popüler olduğu söylenebilir. Geng ve Ark. 2017 yılındaki çalışmalarında ELM'ye dayalı bir gıda güvenliği izleme sistemi önerdiler (Geng ve ark., 2017). Analitik hiyerarşi sürecini, süreç karakteristik bilgilerini oluşturmak için kullandılar. Ardından, süreç karakteristik bilgilerini tahmin için ELM'yi kullandılar. Süreç karakteristik bilgileriyle birlikte tahmin sonuçları, gıda güvenliği izlemesi için daha iyi bilgiler sağlayabildiğini gösterdiler. Ouyang ve Ark. soya sosundaki amino asit nitrojen içeriğini saptamak için ELM'yi sınıflandırıcı olarak kullandıkları bir yaklaşım geliştirdiler (Ouyang ve ark., 2013). Çalışmalarında yakın kızılötesi spektroskopi uygulayıp ardından özellik seçimini kullandılar. Sınıflandırıcı olarak ise ELM'yi kullandılar ve soya sosunun kalite değerlendirmesi için amino asit nitrojen tespit sisteminin yapımında kendi yaklaşımlarının diğer popüler yöntemlerden daha iyi performans gösterdiğini iddia ettiler. Da Costa ve Ark. 2018 yılındaki çalışmalarında farklı şarap türlerinin sınıflandırılması için özellik seçimi ve ELM'yi birleştirerek benzer bir yaklaşım kullandılar (da Costa ve ark., 2018). Liu ve Ark, 2017'deki çalışmalarıyla ELM'nin büyük ölçekli gıda numunesi analizi sırasında tahmin için de uyarlanabilir olduğunu gösterdiler (Liu ve ark., 2017b). Zhang ve Ark. 2018 yılındaki çalışmalarında büyük verilerden süt ürünleri güvenliği tahmininde ELM'yi uygulamışlar ve ELM'nin BP sinir ağı ve destek vektör makinesi gibi diğer yöntemlere üstünlüğünü kanıtlamışlardır (Zhang ve ark., 2018). Guo Ark. yaptıkları çalışmada buğday tanesinin özelliklerini toplamak için darbe akustik sinyali kullandılar ve daha sonra, çekirdeği hasarlı buğdayı çekirdeksiz buğdaydan ayırt etmek sınıflandırıcı olarak ELM'yi uyguladılar (Guo ve ark., 2019). Deneysel sonuçlar, ELM tabanlı yaklaşımlarının bir buğday çekirdeği sınıflandırma sistemini başarıyla oluşturabileceğini kanıtladı.

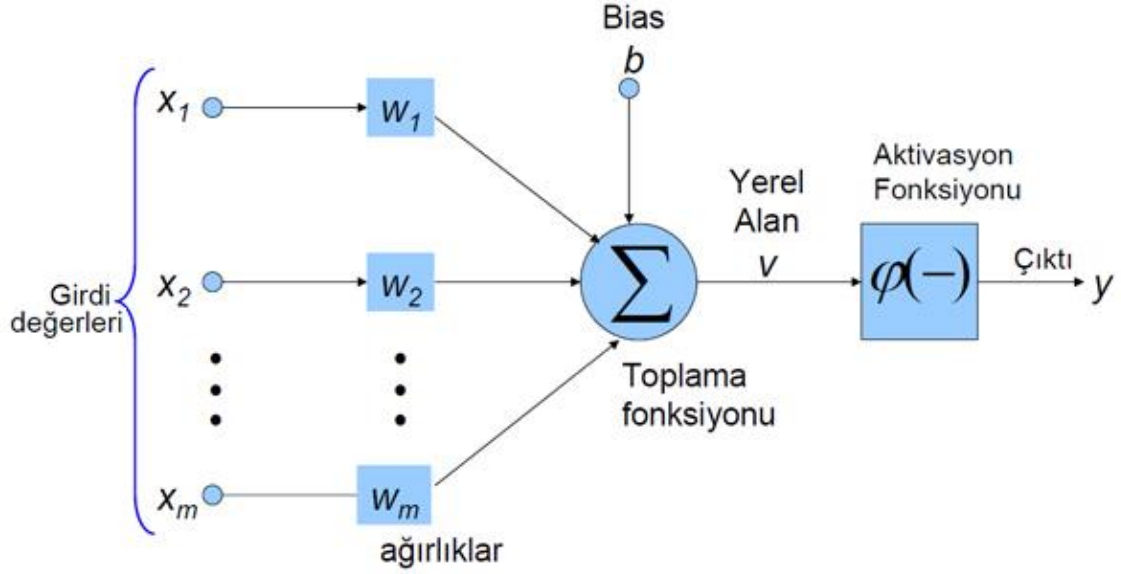
ELM diğer disiplinler arası uygulama alanlarında da kullanılmaktadır. Kontrol sistemleri alanında Zhang ve Ark. 2019 yılındaki çalışmalarında, modelsiz uyarlamalı kontrolde zamanla değişen parametreleri tahmin etmek için hatası minimize edilmiş çevrimiçi düzenli ELM ortaya koydular (Zhang ve Ma, 2019). Geliştirdikleri ELM algoritması, parametreleri eğitebilmekte ve ağ yapısını eşzamanlı optimize edebilmektedir. Diğer bir kontrol sistemi uygulamasında Xu ve Ark. 2019'daki çalışmalarında ELM'ye dayalı saf geri besleme sistemleri için uyarlanabilir bir sinirsel kontrol yöntemi önerdiler (Xu ve ark., 2019). Önerdikleri yöntemde sistemin bilinmeyen terimini tahmin etmek için ELM kullandılar. Wong Ark. 2020 yılındaki çalışmalarında otomotiv motor rölanti hızı kontrolü için ELM'den yararlanmışlardır (Wong ve ark., 2020). Video Analizi alanında da ELM'nin kullanılabilir olduğunu görebiliriz. Yi ve Ark. 2019 yılında, ELM'ye dayalı hareketli bir gölge algılama yöntemi önermişlerdir (Yi ve ark., 2019). İlk olarak, farklı kanalların piksel düzeyindeki öznitelikleri ve bölge düzeyindeki öznitelikleri birleştirilmişler ve daha sonra ELM, pikselleri gölgeli veya gölgesiz olarak sınıflandırmak üzere eğitilmiştir. Rajpal ve Ark. çift yönlü ELM'ye dayalı kendi damgalama algoritmalarını önerdiler (Rajpal ve ark., 2019). Tek veya çift sayıda gizli düğüm için, çift yönlü ELM, hatayı sıfıra indirmek için parametreleri güncellemek için iki farklı yöntem kullandılar. Anahtar çerçeveleri seçmek için eşik tabanlı bir bulanık algoritma kullandılar ve filigran için çift yönlü ELM eğittiler. Çalışmaları gerçek dünya uygulamaları için hız bakımından oldukça tatmin edici sonuçlar verdiğini gösterdiler. Yu Ark. 2019'da KELM(Kernel Extreme Learning Machine - Çekirdek Aşırı Öğrenme Makinesi)'ye dayalı otomatik bir şiddet sahnesi algılama sistemi geliştirdiler (Yu ve ark., 2019). Özellik çıkarıcı olarak gradyan oryantasyonunun 3D histogramı kullandılar ve görsel kelimeleri K ortalama kümeleme ile ürettiriler. Sınıflandırma için KELM'yi kullanmışlar. Deney sonuçları yöntemin etkili ve verimli olduğunu göstermiştir.

2. MATERYAL VE YÖNTEM

Tezin bu bölümünde Yapay Sinir Ağları, Aşırı Öğrenme Makinesi ve Yapay Arı Kolonisi algoritmaların hakkında özet bilgi verilecektir. Bölümün sonunda ABC ile Optimize Edilmiş Aşırı Öğrenme Makinesi çalışması anlatılacaktır. 3. bölümde yapılan araştırma sonuçları ve tartışma, 4. ve son bölümde sonuçlar ve öneriler sunulmuştur.

2.1. Yapay Sinir Ağı

Yapay zekâ uygulamalarından biri olan yapay sinir ağları, insan beyninin çalışma yapısını taklit ederek mevcut verileri analiz edip, bu verilerden farklı öğrenme algoritmaları ile yeni bilgiler oluşturan bilgi işlem teknolojisidir. Yapay sinir ağlarının temelleri (McCulloch ve Pitts, 1943) ile atılmıştır. Ancak XOR problemi olarak bilinen problemi çözmedeki başarısızlığı nedeniyle bir süre Yapay Sinir Ağlarına olan ilgi azalmıştır. 1980'li yıllarda günümüzde bahsi geçen yapay sinir ağlarının devrim niteliğindeki ilk çalışmalarını (Rumelhart ve ark., 1988)'de yapmıştır. (Rumelhart ve ark., 1988) çalışmasında geliştirdiği Hatanın Geriye Yayılma Algoritmasını (Back Propagation Algorithm) tanıtmıştır. Bu çalışma ile yapay sinir ağlarına olan ilginin azalmasına sebep olan XOR problemi gibi problemlerin çözülebileceğini ispatlamıştır. Yapay Sinir Ağları insan beyninin çalışma prensibinden esinlenilerek bulunmuştur. İnsan sinir sisteminin bir parçası olan nöronlar YSA'nın ilham kaynağı olmuştur. Nöronların birbirlerine olan bağlantıları, nörona giriş ve çıkış işlemleri taklit edilmeye çalışılmıştır. Şekil 2.1'de YSA modeli gösterilmiştir.



Şekil 2.1. YSA Modeli (<https://www.veribilimiokulu.com/>)

Bir YSA'nın öğrenme işlemi nöronları arasındaki ağırlık vektörlerinin farklarının en aza indirgenmesiyle mümkündür. YSA'daki eğitimin amacı ağırlık değerlerinin bulunmasıdır. Bu değerlerin bulunmasına ağırlık öğrenmesi denilmektedir. Bu noktada geri yayılım ağırlık eğitimi için verilen örnekler için doğru çıktıya ulaşılması amacıyla ağırlıkların tekrar tekrar ayarlanması işlemidir.

2.2. Aşırı Öğrenme Makinesi

Aşırı öğrenme makinesi, kısaca ELM, ileri beslemeli yapay sinir ağının eğitimi için kullanılan tek gizli katmana sahip bir yapay sinir ağı eğitim algoritmasıdır (Huang ve ark., 2004a; 2006b). ELM yapay sinir ağının gizli katmanı ile çıkış katmanı arasındaki ağırlıklarını 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 Denklem 2.1 ile yapılır.

$$O_j = \sum_{i=1}^N \beta_i g_i(x_j) = \sum_{i=1}^N \beta_i g_i(w_i \cdot x_j + b_i), \quad j = 1..M \quad (2.1)$$

Denklem 2.1’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. Denklem 2.1, Denklem 2.2 şeklinde tekrar yazılabilir.

$$H\beta = T \quad (2.2)$$

$$H = (w_1, \dots, w_N, b_1, \dots, b_N, x_1, \dots, x_N) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_N \cdot x_1 + b_N) \\ \vdots & & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_N \cdot x_N + b_N) \end{bmatrix} \quad (2.3)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{bmatrix} \text{ and } T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix} \quad (2.4)$$

Denklem 2.1 ve Denklem 2.2’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 Denklem 2.5 ile hesaplanmaktadır.

$$\|H(w'_1, \dots, w'_N, b'_1, \dots, b'_N)\beta - T\| = \min_{w_i, b_i, \beta} \|(w_1, \dots, w_N, b_1, \dots, b_N, x_1, \dots, x_N)\beta - T\| \quad (2.5)$$

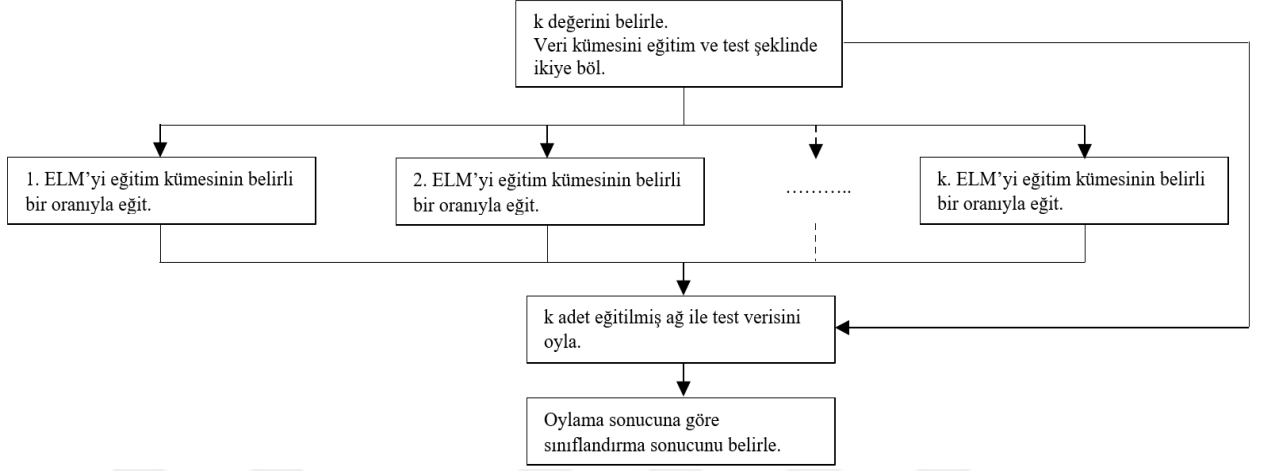
ELM’de giriş katmanı ile gizli katman arasındaki ağırlıklar rastgele atandıktan sonra gizli katman ile çıkış katmanı arasındaki ağırlıklar (β) Denklem 2.6 ile hesaplanabilir.

$$\beta = H^+ T \quad (2.6)$$

Denklem 2.6’deki H^+ Moore-Penrose genelleştirilmiş ters matristir. ELM’de eğitim örnek sayısı ile gizli katmandaki nöron sayısı eşit olduğunda H kare matris olmaktadır ve Moore-Penrose yaklaşımına ihtiyaç olmamaktadır. Ancak genellikle eğitim örnek sayısı gizli katmandaki nöron sayısından fazladır. Bu sebepten dolayı matrisinin tersinin alınabilmesi için Moore-Penrose yaklaşımı kullanılmaktadır. ELM’nin eğitim sürecinin neden çok hızlı olduğu da Denklem 2.5 ile verilen ağırlıkların belirlenebilmesi için çevrimsel yöntemler kullanılırken Denklem 2.6’de tek adımda ağırlıkların belirlenmesi ile ortaya çıkmaktadır.

k-ELM yaklaşımı k sayıda ELM’yi içermektedir ve her bir ELM’nin eğitimi için eğitim örneklerinin oransal olarak kullanılmasını 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-2.2'de gösterilmiştir.



Şekil-2.2. k -ELM Eğitimi ve Testi

k -ELM'nin test edilmesi için Şekil 2.2'de 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 olmasına dikkat edilmiştir. 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.

2.3. Yapay Arı Kolonisi Algoritması

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

1. k değerini belirle.
2. Eğitim kümesini yükle ve onu test ve eğitim olmak üzere ikiye ayır.
3. Kovandaki arı sayısını ($2N$) belirle ve yarısını işçi (N), diğer yarısını gözcü (N) olarak ata.
4. Kâşif arı oluşumunu kontrol etmek için kullanılan limit parametresini tanımla.
5. Problem boyutunu $D = k \times 2$ olarak ata (tek indisli boyutlar nöron sayısını, çift indisli boyutlar aktivasyonu fonksiyonunu göstermektedir).
6. Her bir işçi arı için yiyecek kaynağını Denklem 2.7 kullanarak oluştur.

$$F_{i,j} = L_j^{min} + r_{i,j} \times (H_j^{min} - L_j^{min}), i = 1, 2, \dots, N \text{ ve } j = 1, 2, \dots, D \quad (2.7)$$

Denklem 2.7’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.

7. Yiyecek kaynaklarının kalitesini Denklem 2.8 kullanarak hesapla.

$$fit_i = \frac{1}{1+fit_i} \quad (2.8)$$

Denklem 2.8’de, fit_i i. çözümün kalitesini ve f_i ise Şekil-2.2 ile verilen modeli göstermektedir.

8. En iyi çözümü gbest olarak ata.
9. Her bir yiyecek kaynağı için C sayacını oluştur ve 0’a eşitle.
10. Durdurma kriterini belirle.

İşçi Arı Fazı

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

1. Rastgele bir komşu yiyecek kaynağı seç (m).
2. Rastgele bir boyutu seç (j).
3. Denklem 2.9 kullanarak bir aday yiyecek kaynağı oluştur ($V=Fi$).

$$V_j = F_{i,j} + \varphi \times (F_{i,j} - F_{m,j}) \quad (2.9)$$

4. Denklem 2.8 ile aday yiyecek kaynağını hesapla ve F_i ’nin kalitesi ile karşılaştır. Eğer daha iyiyse aday yiyecek kaynağını ezberle ($Fi=V$), değilse i. yiyecek kaynağının sayacını bir artır ($C_i = C_i + 1$).

Gözcü Arı Fazı

Yiyecek kaynaklarının seçilme ihtimalini Denklem 2.10'u kullanarak hesapla.

$$p_i = \left(0,9 \times \frac{fit_i}{\max(fit)}\right) + 0,1 \quad (2.10)$$

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

1. Denklem 2.10'da verilen ihtimal ile bir yiyecek kaynağını seç.
2. Rastgele bir komşu yiyecek kaynağı seç (m).
3. Rastgele bir boyutu seç (j).
4. Denklem 2.9 kullanarak bir aday yiyecek kaynağı oluştur ($V=Fi$).
5. Denklem 2.8 ile aday yiyecek kaynağını hesapla ve F_i 'nin kalitesi ile karşılaştır. Eğer daha iyiye aday yiyecek kaynağını ezberle ($Fi=V$), değilse i. yiyecek kaynağının sayacını bir artır ($C_i = C_i + 1$).

Kâşif Arı Fazı

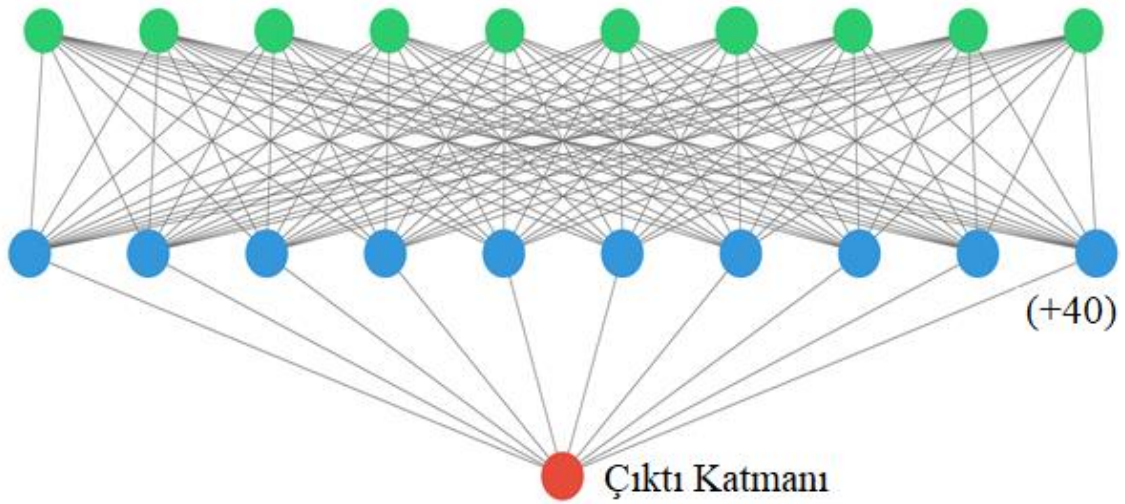
1. C ile belirlenen sayaçlardan en büyük içeriğe sahip olanını belirle.
2. Bu değeri limit ile karşılaştır. Eğer limit değerinden büyükse bu yiyecek kaynağının yerine yeni bir yiyecek kaynağını Denklem 2.7 ile oluştur ve Denklem 2.8 ile bu yiyecek kaynağının kalitesini hesapla.
3. Yeni yiyecek kaynağının sayacını sıfırla.

Sonlandırma

1. Kovandaki en iyi yiyecek kaynağını belirle ve gbest ile karşılaştır. Hangisi daha iyiye gbest olarak ata.
2. Durdurma kriteri sağlanmadıysa İşçi Arı Fazı'na git, sağlandıysa gbest'i raporla.

2.4. YSA Modeli ve Eğitim Algoritmaları

Bütünleme sınavına giren öğrencilerin sınıflandırılması için 10x50x1 ağ yapısı kurulmuştur. Gizli katmanın tek olarak seçilmesinin nedeni, ELM'nin tek bir gizli katman ağı eğitebilmesi ve metaheursitcis algoritmaları ile optimize edilecek ağırlık sayısının 550 olmasıdır. Ağın yapısı Şekil 2.3'deki gibidir:



Şekil 2.3. YSA Modeli (10x50x1)

Şekil 2.3'de verilen ağı eğitmek için ELM, EBP ve sürü zekası algoritmaları kullanılmıştır.

(Huang ve ark., 2006b), 2005 yılında yaptığı çalışmada ELM tekniğini önermiş ve bu çalışmada tek bir gizli katmana sahip aşırı öğrenme makinesi adı verilen yeni bir ileri beslemeli YSA algoritmasını sunmuştur. Bu ELM'de, geleneksel ileri beslemeli YSA'lardan farklı olarak, giriş ağırlıkları ve gizli eşik değeri rastgele verilir ve çıkış katmanındaki nöronların ağırlıkları bir matris işlemi olarak hesaplanır. ELM'de girdi ve gizli katmanlar arasındaki ağırlıklar rastgele üretildikten sonra gizli ve çıktı katmanları arasındaki ağırlıklar Denklem 2.11'deki gibi hesaplanır:

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_M \end{bmatrix}_{M \times 1} \quad \text{and} \quad \gamma = \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_n \end{bmatrix}_{N \times 1} \quad (2.11)$$

Burada $g(\cdot)$ aktivasyon fonksiyonudur, w ve b sırasıyla giriş ve çıkış katmanları arasındaki ağırlık ve sapmadır. β , gizli ve çıktı katmanları arasındaki ağırlık vektörüdür ve γ , hedef vektördür. β ağırlık değerleri bulunduğunda denklem, Denklem 2.12 gibi çözümlenir:

$$H \times \beta = \gamma \quad (2.12)$$

Denklem 2.12'i çözüldüğünde, gizli ve çıkış katmanları arasındaki optimum ağırlık değerleri elde edilmektedir. Eğitim örneklerinin sayısı gizli katmanlardaki nöronların sayısına eşit olduğunda, H bir kare matristir ve denklem temel matris işlemleriyle çözülebilir. Ancak genel olarak eğitim örneklerinin sayısı gizli katmandaki nöronlardan daha fazladır ve dolayısıyla H dikdörtgen bir matristir. Bu sorunun üstesinden gelmek için (Huang ve ark., 2006b)'te H matrisinin Moore-Penrose genelleştirilmiş tersi kullanılmış ve Denklem 2.12, Denklem 2.13'deki gibi yeniden tanımlanmıştır:

$$\beta' = H^{\dagger} \gamma \quad (2.13)$$

ELM'nin algoritmik adımları aşağıdaki gibidir:

- i) Girdi ve gizli katman arasındaki ağırlıkları ve biasları rastgele üret ve ata.
- ii) H matrisini hesapla.
- iii) Denklem'i kullanarak gizli ve çıktı katmanları arasındaki ağırlıkları hesapla.

Gizli katmandaki nöron sayısı, (Huang ve ark., 2006b) 'te verilen farklı eğitim örneklerinin sayısı ile sınırlıdır. ELM'de algoritmanın performansını doğrudan etkileyen iki önemli seçim vardır. Bunlardan biri gizli katmandaki nöron sayısı diğeri ise aktivasyon fonksiyonudur. Bu çalışmada, gizli katmandaki nöron sayısı ve aktivasyon fonksiyon parametreleri sırasıyla 50 ve tanjant sigmoid olarak ayarlanmıştır.

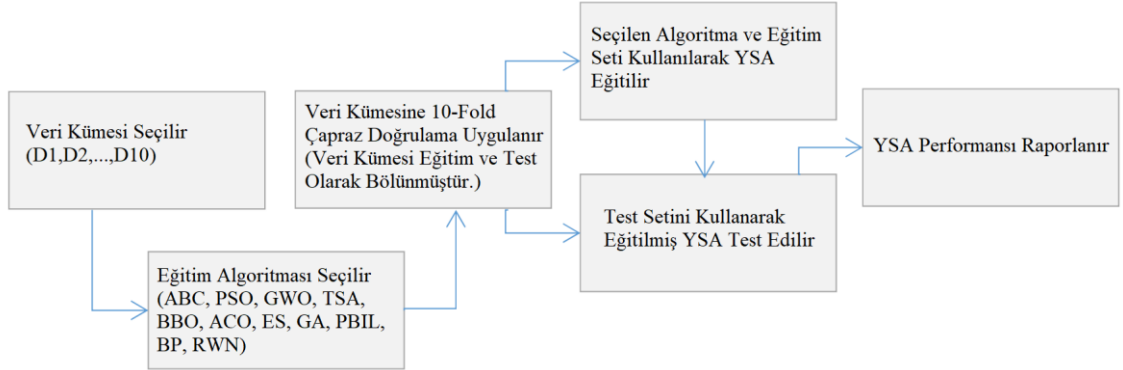
Bu çalışmada YSA modelini eğitmek için kullanılan metasezgisel algoritmalar, parçacık sürüsü optimizasyonu (Particle Swarm Optimization - PSO), gri kurt optimizer(Gray Wolf Optimizer - GWO), karınca kolonisi optimizasyonu(Ant Colony Optimization - ACO), evrim stratejisi(Evolution Strategy - ES), genetik algoritma(Genetic Algorithm - GA), popülasyon tabanlı artımlı öğrenme (Population-Based Incremental Learning - PBIL), yapay arı kolonisi(ABC) algoritması, biyocoğrafya tabanlı optimizasyon (Biogeography-Based Optimization - BBO), ağaç tohumu algoritması(Tree Seed Algorithm - TSA)'dır. Tüm bu algoritmaların ileri beslemeli YSA modeli eğitimi üzerindeki performansı, (Cinar, 2020)'da en çok bilinen veri kümeleri üzerinde incelenmiştir. Ancak (Cinar, 2020)'da önemli bir konu vardır. O da eğitim ve test setlerinin aynı olması ve metasezgisellerin gerçek performansının gösterilmiş olmasıdır. Bu çalışmada, YSA modelini eğitmek ve test etmek için k'nin 10 olarak

seçildiği k-kat (k-fold) çapraz doğrulama kullanılmıştır. Metasezgisellerin ortak özellikleri, popülasyona sahip olmaları ve bir çözüm güncelleme mekanizması (aritmetik veya genetik operatörler) kullanmaları ve çözümler arasında açgözlü seçim yapmalarıdır. YSA eğitimi için genel bir metaherustik modeli aşağıdaki gibidir:

- i) Çözüm popülasyonu oluştur.
- ii) YSA ağırlıklarına metaherustik yöntemin çözüm güncelleme mekanizması uygula.
- iii) Bu YSA ile eğitim örneklerini sınıflandır.
- iv) Gelecek popülasyona daha iyi çözümleri aktar.
- v) Bir sonlandırma koşulu karşılanmaz ise (ii) adımına git; aksi takdirde en iyi çözümü bildir.

Optimize edilmiş ağırlık vektörü olan en iyi çözüm kullanılarak test örnekleri sınıflandırılmış ve böylece algoritmanın performansı araştırılmıştır.

Algoritmaların bu çalışma ile ilgili veri kümelerine uygulanması Şekil 2.4’de verilmiştir.



Şekil 2.4. YSA eğitimi ve testi üzerine algoritmaların uygulama framework’ü.

Önerilen framework’te, örnekler öncelikle k-fold çapraz doğrulama şeması kullanılarak eğitim ve test olmak üzere iki gruba ayrılmıştır. k, 10 olarak seçildiğinden, verilerin dokuz bölümü YSA modelinin eğitimi için, geri kalan bölümü ise YSA modelini test etmek için kullanılmıştır. Bu işlem verilerin tüm bölümleri için uygulanmış ve k-kat çapraz doğrulama ile algoritmalar için eşit bir eğitim ve test süreci elde edilmiştir.

2.5. ABC ile optimize edilmiş Aşırı Öğrenme Makinesi

ABC algoritmasının k-ELM parametrelerinin optimizasyonu için uygulanmasında iki önemli nokta bulunmaktadır. Bir tanesi Denklem 2.8 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 Denklem 2.8 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 ABC algoritmasının tamsayı versiyonu bu çalışmada kullanılmıştır. ABC algoritmasında bir çözümün temsili Şekil 2.5’de gösterilmiştir.

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-2.5. ABC’de Çözümün Temsili

Şekil-2.5’de NS gizli katman nöron sayısını ve AF ELM’nin aktivasyon fonksiyonudur. Bu bağlamda yapay sinir ağları 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 Denklem 2.9 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.

2.6. Veri Kümelerinin Özellikleri

Deneylerde kullanılan veri setleri 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 (D9) derslerini almış ve bütünleme sınavına girme hakkı bulunan öğrencilerin kişisel olmayan verileri ile hazırlanmıştır. 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 bütünleme sınavı sayısıdır. Bu veri kümeleri için iki adet sınıf etiketi oluşturulmuştur. Sınıf etiketi 0 ise öğrenci bütünleme sınavına kaldığı halde girmemiştir ve 1 ise öğrenci bütünleme sınavına kalmıştır ve bütünleme sınavına girmiştir. Veri setlerindeki toplam örneklem sayıları verilecek olursa; D1’de 124, bunun 52’si 0 sınıfında ve 72’si 1 sınıfında bulunmaktadır. D2’de 71, bunun 32’si 0 sınıfında ve 39’u 1 sınıfında bulunmaktadır. D3’de 391, bunun 132’si 0 sınıfında ve 259’u 1 sınıfında bulunmaktadır. D4’de 196, bunun 69’u 0 sınıfında ve 127’si 1 sınıfında bulunmaktadır. D5’de 391, bunun 137’si 0 sınıfında ve 254’ü 1 sınıfında bulunmaktadır. D6’da 497, bunun 253’ü 0 sınıfında ve 244’ü 1 sınıfında bulunmaktadır. D7’de 84, bunun 33’ü 0 sınıfında ve 51’i 1 sınıfında bulunmaktadır. D8’de 62, bunun 22’si 0 sınıfında ve 40’ı 1 sınıfında bulunmaktadır. D9’da 292, bunun 140’ı 0 sınıfında ve 152’si 1 sınıfında bulunmaktadır. D10’da 2108, bunun 870’i 0 sınıfında ve 1238’i 1 sınıfında bulunmaktadır. Küme tanımı çizelge 2.1’de gösterilmiştir. Ayrıca veri setlerindeki özniteliklere ait istatistiksel bilgiler Çizelge 2.2’de verilmiştir.

Çizelge 2.1. Veri Küme Tanımı

Veri Kümesi Adı	Özellik Sayısı	Sınıf Sayısı	Kayıp Değer	Örneklem Sayısı	1. Sınıftaki Örneklem Sayısı	2. Sınıftaki Örneklem Sayısı
D1: Bilgisayar Mühendisliğine Giriş	10	2	Yok	124	52	72
D2: Atatürk İlke ve İnkılap Tarihi	10	2	Yok	71	32	39
D3: Fizik 1	10	2	Yok	391	132	259
D4: İşletim Sistemleri	10	2	Yok	196	69	127
D5: Matematik 1	10	2	Yok	391	137	254
D6: Matematik 2	10	2	Yok	497	253	244
D7: Türk Dili 1	10	2	Yok	84	33	51
D8: Türk Dili 2	10	2	Yok	62	22	40
D9: Veri Yapıları	10	2	Yok	292	140	152
D10: Tüm Dersler	10	2	Yok	2108	870	1238

Çizelge 2.2. Veri kümelerinde kullanılan özelliklerin değerleri

Özellikler	Değer	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
DNO	Min	0	0	0	0	0	0	0	0	0	0
	Maks	2.55	2.84	3.41	2.7	3.28	3.22	2.88	3.33	3.41	3.41
	Ort	0.922	0.975	1.22	1.2	1.254	1.124	1.193	0.877	1.422	1.193
	StnSap	0.634	0.806	0.699	0.768	0.701	0.69	0.793	0.699	0.85	0.742
NO	Min	0	0	0	0	0	0	0	0.17	0	0
	Maks	3.07	3.6	3.57	3.3	3.57	3.57	3.35	3.57	3.29	3.6
	Ort	1.974	1.755	2.304	2.246	2.279	2.376	2.108	2.121	2.429	2.277
	StnSap	0.974	1.121	0.774	0.64	0.767	0.645	0.993	0.97	0.521	0.761
ÖADS	Min	1	1	0	2	1	0	1	3	1	0
	Maks	10	10	10	7	10	9	10	9	8	10
	Ort	6.863	6.169	5.506	4.776	5.215	4.59	6.75	6.065	4.021	5.13
	StnSap	1.399	2.104	2.388	1.289	2.532	2.014	1.803	1.129	2.051	2.232
VN	Min	0	0	0	0	0	0	0	0	0	0
	Maks	96	88	82	75	69	67	90	75	91	96
	Ort	4.685	34.507	33.642	28.883	29.668	20.312	41.357	40.71	34.822	30.442
	StnSap	22.476	23.035	17.19	19.54	17.95	15.124	24.307	23.328	22.746	20.12
FN	Min	0	0	0	0	0	0	0	0	0	0
	Maks	68	92	73	75	65	61	70	80	85	92
	Ort	35.419	29.704	25.514	23.204	11.834	12.553	33.69	22.097	26.13	20.741
	StnSap	25.254	28.692	18.51	24.097	13.086	15.107	20.727	23.321	23.234	20.877
ADS	Min	0	0	0	0	0	0	0	0	0	0
	Maks	10	10	10	6	10	8	10	6	6	10
	Ort	3.452	3.493	2.125	1.929	2.107	1.934	2.917	3.323	1.986	2.23
	StnSap	2.054	2.623	1.685	1.684	1.702	1.33	2.48	1.725	1.801	1.797
DÖ	Min	1	1	0	1	0	0	1	0	1	0
	Maks	9	9	9	11	11	12	5	8	11	12
	Ort	2.145	2.803	2.949	5.378	3.345	4.209	2.143	3.032	4.18	3.667
	StnSap	1.596	1.947	2.008	1.995	2.152	2.204	1.253	1.568	2.225	2.238
KU	Min	0	0	0	0	0	0	0	1	1	0
	Maks	1134	1225	1256	1256	1256	1256	1225	99	99	1256
	Ort	300.032	263.507	338.442	343.393	339.65	351.801	393.202	40.581	40.274	289.612
	StnSap	288.074	336.053	322.95	365.496	317.494	324.083	377.095	25.517	19.717	319.856
C	Min	0	0	0	0	0	0	0	0	0	0
	Maks	1	1	1	1	1	1	1	1	1	1
	Ort	0.621	0.592	0.657	0.704	0.706	0.728	0.762	0.823	0.695	0.697
	StnSap	0.487	0.495	0.475	0.458	0.456	0.445	0.428	0.385	0.461	0.46
BSS	Min	1	1	0	1	1	0	1	1	1	0
	Maks	10	10	10	7	10	9	10	9	7	10
	Ort	5.573	5.338	3.941	3.566	3.754	3.087	5.167	4.71	3.223	3.785
	StnSap	1.702	2.229	1.886	1.328	1.922	1.531	2.127	1.464	1.929	1.92

DNO(Dönem Not Ortalaması), NO(Not Ortalaması), ÖADS(Öğrencinin Aldığı Ders Sayısı), VN(Ara Sınav Notu), FN (Final Notu), ADS (Alttan Alınan Ders Sayısı), DO (Öğrencinin Dönemi), KU (Konyaya Uzaklık), C (Cinsiyet), BSS (Bütünleme Sınavı Sayısı)

2.7. Karışıklık Matrisleri

Bu çalışmada ELM ile bütünleme sınavına girme hakkı olan öğrencilerin ilgili dersin sınavına girip girmeyeceği ile ilgili bir tahmin yapılmaktadır.

Çıktı, ya sınava girecek (+sg) yada sınava girmeyecek (-sg) şeklindedir.

Dolayısıyla herhangi bir X öğrencisi için 4 farklı durum bulunmaktadır. Bu durumlar:

- Gerçek Pozitif (TP) : Tahminimiz +sg ve X sınava giriyor, istenilen durum.
- Gerçek Negatif (TN): Tahminimiz -sg ve X sınava girmiyor.
- Yanlış Pozitif (FP): Tahminimiz -sg ve X sınava giriyor.
- Yanlış Negatif (FN): Tahminimiz +sg ve X sınava girmiyor.

Buradaki önemli husus tahminin doğru verildiği durumlardır. Hedeflenen, gerçekten sınava giren öğrencilere tahminin de pozitif olmasıdır. Bu durumun oranının yüksek olması en çok beklenen olaydır. Diğer bir istenilen durum ise sınava girmeyen öğrenciler için tahminin pozitif olduğu durumun oranının sıfıra yakın olmasıdır. Aynı şekilde sınava girmeyen öğrenciler için ise sınava girecek tahmininin de sıfıra yakın olması istenilen durumdur.

Doğruluk (Accuracy)

Doğru etiketlenmiş örneklemelerin tüm örneklem havuzuna oranıdır. Performans kriterleri içerisinde en sezgisel olanıdır. Doğruluk şu sorunun cevabını vermektedir:

Tüm öğrenciler içerisinde kaç öğrenciyi doğru sınıflandırdık?

$$\text{Doğruluk (Accuracy)} = \frac{(TP+TN)}{(TP+FP+FN+TN)}$$

Pay: Tümü doğru etiketlenmiş öğrenciler.

Payda: Tüm öğrenciler.

Kesinlik – Hassasiyet (Precision)

Kesinlik, önerilen yaklaşım tarafından doğru olarak sınıflandırılan +sg'nin tüm +sg sınıflara oranıdır. Kesinlik şu sorunun cevabını vermektedir:

Sınava girecek olarak tahmin edilen öğrencilerin kaç tanesi aslında sınava girmiştir?

$$\text{Kesinlik (Precision)} = \frac{TP}{(TP+FN)}$$

Pay: +sg olarak sınıflandırılan öğrenciler

Payda: pozitif sınıflandırdığımız tüm öğrenciler

Geri Çağırma – Hatırlama (Recall)

Hatırlama, pozitif olarak önerilen öğrencilerin gerçekte pozitif olan tüm öğrencilere oranıdır. Hatırlama şu sorunun cevabını vermektedir:

Tüm sınava giren öğrencilerden kaç tanesi doğru tahmin edildi?

$$\text{Hatırlama (Recall)} = \frac{TP}{(TP+FP)}$$

Pay: Pozitif önerilen sınava giren öğrenciler

Payda: Sınava giren tüm öğrenciler

F1-Skoru (F1-score/aka F-Score/F-Measure)

F1 Skoru hem kesinliği hem de hatırlamayı dikkate almaktadır. Kesinlik ve hatırlamanın harmonik ortalamasıdır. Sistemde kesinlik (p) ve geri çağırma (r) arasında bir tür denge varsa F1 Skoru en iyisidir. Tam tersine eğer aralarından birinin iyileştirmek amacıyla ötekisinden ödün veriliyorsa yani aralarındaki denge bozulursa o zaman F1-Skorunun çok da iyi olduğu söylenemez.

Örneğin, P=1 ve R=0 ise, F1 puanı 0'dır.

$$\text{F1 Puanı} = 2 * (\text{Hatırlama} * \text{Kesinlik}) / (\text{Hatırlama} + \text{Kesinlik})$$

Özgünlük (Specifity)

Özgünlük, gerçekte sınava girmeyen öğrencilerin tahmin olarak da doğru şekilde sınıflandırılmasıdır. Özgünlük şu sorunun cevabını vermektedir: Sınava girmeyen öğrencilerden kaç tanesini doğru tahmin edildi?

$$\text{Özgünlük} = \text{TN} / (\text{FN} + \text{TN})$$

Pay: Sınava girmediği tahmin edilen öğrenciler

Payda: Gerçekten sınava girmeyen tüm öğrenciler

Karışıklık Matrisleri İle İlgili Genel Notlar

Doğruluk, veri kümelerinin simetrik olduklarında gerçekten harika bir ölçüttür (yanlış negatifler ve yanlış pozitif sayıları birbirine yakındır). Yanlış pozitifler ve yanlış negatiflerin dağılımları farklıysa, F1 iyi bir ölçüttür. Eşit olmayan bir sınıf dağılımı varsa ise F1 en iyi seçim olacaktır. Kesinlik, gerçek pozitiflerden ne kadar emin olduğudur, Hatırlama ise herhangi bir pozitifliğin kaçırılmadığından ne kadar emin olduğudur. Eğer yanlış pozitifler yanlış negatiflerden çok daha iyiye, başka bir deyişle, yanlış negatiflerin oluşumu kabul edilmiyorsa/tahammül edilemezse, bazı yanlış negatifleri kaydetmek yerine fazladan yanlış pozitifler (yanlış alarmlar) tercih ediliyorsa, o zaman Hatırlama'nın seçilmesi en doğrusu olacaktır.

Gerçek pozitiflerinizden daha emin olmak istiyorsanız o zaman kesinliği seçmelisiniz. Örneğin, Spam e-postalar. İstenmeyen posta kutusundaki bazı normal e-postalar yerine, gelen kutusunda bazı istenmeyen e-postalar olmasını tercih edilir. Bu nedenle, e-posta şirketi, bir e-postayı spam kutusuna koymadan önce spam olduğundan emin olmak ister.

Tüm gerçek negatifler kapsanmak istenirse, yani herhangi bir yanlış alarm istenmiyorsa, herhangi bir yanlış pozitif istenmez ise Özgünlük seçilmesi en doğrusu

olacaktır. Örneğin, pozitif çıkan herkesin hemen hapse gireceği bir uyuşturucu testi yapılmak istendiğinde, uyuşturucu kullanmayan kimsenin hapse girmemesi gerekmektedir. Burada yanlış pozitifler kabul edilemez.

Sonuç:

- Örneğin, %90 doğruluk değeri, her 10 etiketten 1'inin yanlış ve 9'unun doğru olduğu anlamına gelir.
- Örneğin, %80 kesinlik değeri, ortalama olarak sınıflandırıcı tarafından sınava girecek şeklinde sınıflandırılan her 10 öğrenciden 2'sinin sınava girmediği, 8'inin sınava girdiği anlamına gelir.
- Örneğin, hatırlama değerinin %70 olması, gerçekte her 10 sınava giren öğrenciden 3'ünün sınıflandırıcı tarafından kaçırıldığı ve 7'sinin sınava girecek şeklinde sınıflandırıldığı anlamına gelir.
- Örneğin, özgünlük değerinin %60 olması, gerçekte her 10 sınava girmeyen öğrencinin 4'ünün sınava girecek şeklinde yanlış sınıflandırıldığı ve 6'sının sınava girmeyecek şeklinde doğru olarak sınıflandırıldığı anlamına gelir.

3. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

3.1. Aşırı Öğrenme Makinesi ile Edilen Sonuçlar ve Karşılaştırmalar

Bu çalışmada, eğitim algoritmalarının karşılaştırılması için normalize edilmiş ve edilmemiş on adet veri seti kullanılmış olup, YSA modeli giriş katmanı (10 nöron), bir gizli katman (50 nöron) ve çıkış katmanından (1 nöron) oluşmaktadır. Kullanılan modelde eğitim algoritmaları tarafından optimize edilen ağırlıkların sayısı 550 ve bias 51'dir. Metasezgisel algoritmaların ortak kontrol parametreleri sonlandırma koşuludur ve metasezgisel algoritmaları sonlandırmak için maksimum fonksiyon değerlendirme sayısı kullanılır. Metasezgisel algoritmalara özgü diğer kontrol parametreleri (Cinar, 2020)'daki gibi kullanılır. ELM ve BP'de nöronların aktivasyon fonksiyonu teğet sigmoid'dir. Adil bir karşılaştırma için, k'nin 10 olarak seçildiği k-kat çapraz doğrulama şeması kullanılır ve rapor edilen sonuçlar tüm çizelgelerdeki katların ortalamasıdır.

Çizelge 3.1. Normalleştirilmemiş Veri Kümeleri Üzerindeki Algoritmaların Eğitim Doğruluğu

Veri Kümesi	BBO	PSO	GA	ACO	ES	PBIL	GWO	TSA	ABC	BP	ELM
D1	0.945	0.784	0.862	0.757	0.790	0.797	0.877	0.790	0.865	1.000	0.791
D2	0.995	0.822	0.933	0.789	0.795	0.847	0.946	0.827	0.933	1.000	0.881
D3	0.878	0.782	0.841	0.775	0.766	0.795	0.854	0.782	0.837	0.980	0.770
D4	0.936	0.841	0.898	0.822	0.804	0.848	0.913	0.845	0.894	1.000	0.814
D5	0.898	0.822	0.866	0.797	0.791	0.830	0.893	0.824	0.872	0.998	0.777
D6	0.883	0.791	0.850	0.756	0.771	0.803	0.874	0.798	0.851	0.999	0.730
D7	0.946	0.792	0.869	0.770	0.796	0.806	0.894	0.806	0.875	1.000	0.809
D8	0.991	0.832	0.953	0.807	0.799	0.843	0.946	0.843	0.961	1.000	0.960
D9	0.912	0.777	0.856	0.754	0.763	0.779	0.866	0.783	0.867	1.000	0.806
D10	0.835	0.776	0.824	0.758	0.754	0.777	0.852	0.779	0.814	0.959	0.722

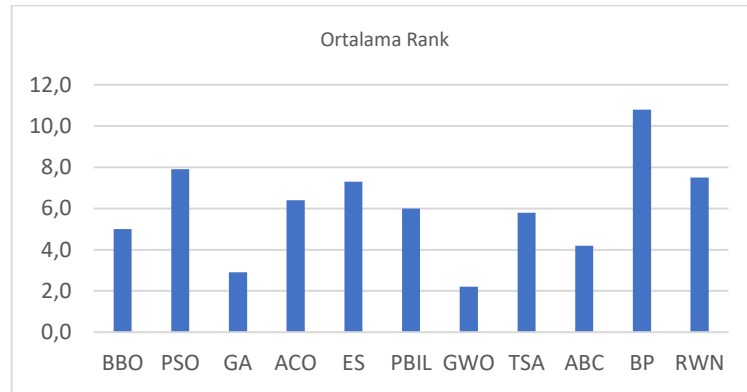
Çizelge 3.2. Normalleştirilmemiş Veri Kümelerinde Algoritmaların Test Doğruluğu

Veri Kümesi	BBO	PSO	GA	ACO	ES	PBIL	GWO	TSA	ABC	BP	ELM
D1	0.698	0.745	0.739	0.755	0.772	0.746	0.766	0.729	0.767	0.553	0.682
D2	0.694	0.724	0.795	0.697	0.756	0.764	0.692	0.782	0.769	0.610	0.650
D3	0.791	0.748	0.812	0.761	0.759	0.769	0.815	0.765	0.785	0.627	0.768
D4	0.796	0.800	0.835	0.815	0.779	0.813	0.851	0.806	0.792	0.612	0.805
D5	0.828	0.794	0.836	0.774	0.778	0.805	0.854	0.795	0.811	0.598	0.809
D6	0.787	0.769	0.815	0.787	0.741	0.786	0.835	0.777	0.779	0.532	0.779
D7	0.814	0.722	0.753	0.742	0.768	0.761	0.775	0.772	0.762	0.555	0.617
D8	0.749	0.704	0.790	0.748	0.702	0.658	0.784	0.680	0.754	0.625	0.564
D9	0.728	0.717	0.754	0.745	0.744	0.719	0.822	0.749	0.759	0.672	0.745
D10	0.799	0.773	0.819	0.754	0.749	0.774	0.845	0.774	0.786	0.753	0.781

Çizelge 3.3. Normalleştirilmemiş Veri Kümeleri Üzerinde YSA Eğitimi İçin Algoritmaların Çalışma Süresi

Veri Kümesi	BBO	PSO	GA	ACO	ES	PBIL	GWO	TSA	ABC	BP	ELM
D1	100.373	81.397	79.547	119.088	99.219	37.766	127.782	194.364	81.491	1.751	0.005
D2	92.925	74.861	74.371	117.862	93.917	31.747	124.266	191.813	76.419	0.403	0.004
D3	127.268	108.941	105.944	140.681	130.974	68.692	155.888	218.450	108.005	9.166	0.008
D4	109.372	90.410	88.804	127.469	109.593	46.695	137.752	203.575	90.429	3.475	0.006
D5	127.740	109.534	106.978	142.816	131.875	68.386	157.060	219.294	108.121	3.742	0.006
D6	138.407	119.374	116.882	149.798	142.555	78.901	164.679	229.407	118.660	2.686	0.007
D7	93.999	76.089	74.232	117.353	95.141	32.879	125.113	191.335	76.970	2.780	0.002
D8	92.046	73.664	72.684	114.781	93.019	30.870	122.021	189.515	75.669	0.352	0.002
D9	120.805	103.250	101.353	134.467	124.441	61.223	146.528	213.676	102.622	1.198	0.007
D10	283.415	267.748	257.219	266.589	303.320	226.039	305.739	374.603	255.262	18.648	0.026

Çizelge 3.1, 3.2 ve 3.3'e göre veri setleri üzerinde daha iyi eğitim doğruluğu BP algoritması ile elde edilmiştir. BP algoritmasının eğitim hatası, 10 veri kümesinin 6'sında 0'dır. Ancak, BP algoritması test senaryolarında benzer performans göstermedi. Test senaryolarında, GWO algoritması, ortalama test doğruluğu açısından diğer algoritmalarından daha iyi performans göstermektedir. YSA'yı eğitecek algoritmaların çalışma sürelerini göz önüne alılınca, ELM daha iyi performans gösteriyor ve beklendiği gibi YSA'yı bir saniyeden daha kısa sürede eğitiyor çünkü diğer algoritmaların YSA'yı eğitmek için yinelemeli bir süreci bulunmaktadır. Test doğruluğuna göre algoritmaların ortalama sıralaması Şekil 3.1'de verilmiştir.



Şekil 3.1. Test Doğruluğu İçin Normalleştirilmemiş Veri Kümelerindeki Algoritmaların Ortalama Sıralaması

Çizelge 3.4. Normalleştirilmiş Veri Kümeleri Üzerindeki Algoritmaların Eğitim Doğruluğu

Veri Kümesi	BBO	PSO	GA	ACO	ES	PBIL	GWO	TSA	ABC	BP	ELM
D1	0.933	0.787	0.895	0.751	0.776	0.800	0.879	0.796	0.903	1.000	0.907
D2	0.980	0.834	0.953	0.773	0.805	0.847	0.941	0.848	0.955	1.000	0.993
D3	0.861	0.771	0.850	0.769	0.759	0.788	0.859	0.783	0.844	1.000	0.826
D4	0.918	0.834	0.902	0.788	0.815	0.849	0.909	0.843	0.916	1.000	0.900
D5	0.904	0.816	0.880	0.778	0.798	0.831	0.882	0.833	0.888	1.000	0.852
D6	0.883	0.797	0.857	0.757	0.771	0.802	0.868	0.796	0.862	1.000	0.829
D7	0.926	0.777	0.889	0.753	0.773	0.795	0.887	0.792	0.920	1.000	0.957
D8	0.970	0.840	0.926	0.773	0.820	0.853	0.936	0.855	0.951	1.000	0.985
D9	0.885	0.762	0.864	0.750	0.747	0.775	0.870	0.772	0.872	1.000	0.844
D10	0.833	0.762	0.834	0.754	0.737	0.774	0.853	0.774	0.817	0.970	0.793

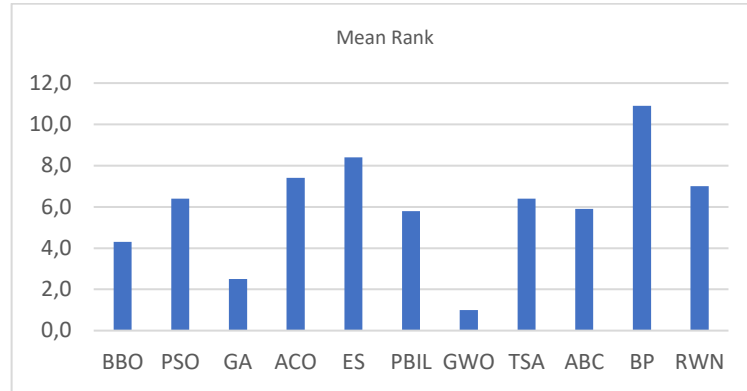
Çizelge 3.5. Normalleştirilmiş Veri Kümeleri Üzerindeki Algoritmaların Test Doğruluğu

Veri Kümesi	BBO	PSO	GA	ACO	ES	PBIL	GWO	TSA	ABC	BP	ELM
D1	0.743	0.700	0.758	0.710	0.733	0.736	0.787	0.677	0.675	0.556	0.682
D2	0.769	0.734	0.715	0.598	0.720	0.728	0.794	0.698	0.636	0.626	0.650
D3	0.780	0.754	0.805	0.770	0.728	0.772	0.840	0.749	0.785	0.634	0.768
D4	0.807	0.812	0.842	0.793	0.768	0.824	0.849	0.812	0.799	0.662	0.805
D5	0.820	0.791	0.851	0.714	0.762	0.783	0.852	0.811	0.802	0.569	0.809
D6	0.795	0.760	0.819	0.748	0.740	0.782	0.851	0.765	0.783	0.493	0.779
D7	0.674	0.664	0.759	0.710	0.607	0.676	0.796	0.696	0.709	0.604	0.617
D8	0.642	0.720	0.770	0.675	0.708	0.658	0.786	0.678	0.775	0.498	0.564
D9	0.778	0.745	0.809	0.750	0.700	0.735	0.834	0.742	0.734	0.624	0.745
D10	0.802	0.755	0.830	0.751	0.731	0.769	0.850	0.774	0.800	0.725	0.781

Çizelge 3.6. Normalleştirilmiş Veri Kümeleri Üzerinde YSA Eğitimi İçin Algoritmaların Çalışma Süresi

Veri Kümesi	BBO	PSO	GA	ACO	ES	PBIL	GWO	TSA	ABC	BP	ELM
D1	100.851	82.020	80.145	120.771	101.698	39.384	126.992	195.623	82.686	0.145	0.005
D2	92.530	74.060	73.343	114.167	92.833	31.848	120.561	187.887	75.119	0.073	0.004
D3	130.070	113.073	110.172	148.517	138.178	72.593	151.029	220.587	111.585	0.450	0.008
D4	109.214	90.942	89.400	128.106	112.012	48.804	131.888	201.042	90.874	0.143	0.006
D5	130.339	113.405	110.798	148.524	138.432	73.019	150.277	220.892	111.353	0.602	0.006
D6	139.762	123.282	120.632	155.028	151.030	85.420	160.878	231.876	122.243	1.085	0.007
D7	95.007	76.706	75.096	117.318	96.227	34.016	124.124	188.324	77.076	0.094	0.002
D8	91.175	73.391	72.289	114.552	92.478	30.469	120.680	187.739	75.228	0.098	0.002
D9	118.793	102.766	100.492	138.675	124.902	61.331	141.953	211.654	101.710	0.323	0.007
D10	304.049	290.832	277.314	303.019	338.719	249.556	297.863	399.621	277.724	18.881	0.026

Çizelge 3.4, 3.5 ve 3.6'ya göre, BP algoritmaları, D10 hariç normalleştirilmiş veri kümeleri üzerinde ANN modelini mükemmel bir şekilde eğitir ve BP algoritmasının eğitim hatası, 10 veri kümesinin 9'unda 0'dır. Ancak BP tarafından eğitilen YSA eğitim örneklerini ezberlediğinden, BP tarafından eğitilen YSA'nın test performansı diğer eğitim algoritmalarına göre çok daha düşüktür. Test senaryolarında, GWO algoritması, ortalama test doğruluğu açısından diğer algoritmalarından daha iyi performans göstermektedir. Metasezgisel algoritmaların çalışma süreleri normalleştirilmiş veri setlerinde biraz azalır, ancak ELM, karşılaştırılan algoritmalar arasında daha iyi performans gösterir. Test doğruluğuna göre algoritmaların ortalama sıralaması Şekil 3.2'da verilmiştir.



Şekil 3.2. Test Doğruluğu İçin Normalleştirilmiş Veri Kümeleri Üzerindeki Algoritmaların Ortalama Sıralaması

3.2. ABC ile Optimize Edilmiş Aşırı Öğrenme Makinesi ile Edilen Sonuçlar ve Karşılaştırmalar

Bu çalışmada, k-ELM’de 3 adet algoritmik parametre bulunmaktadır. Bu parametreler, k (ELM’lerin özdeş olmadığı durumda), gizli katmandaki nöron sayısı ve aktivasyon fonksiyonudur. Cao ve ark. (Cao ve ark., 2012) yaptıkları çalışmalarında k değerini 7 almışlar ve 19 veri seti üzerinde önerdikleri yaklaşım ile temel ELM’ye göre daha iyi sonuçlar elde etmişlerdir. Bu sebeple yapılan deneysel çalışmada k değeri 7 olarak belirlenmiş ve kullanılmıştır. k değerinin 7 alındığı durumda her ELM için 2 karar değişkeni olmasından dolayı 14 boyutlu bir optimizasyon problemiyle karşı karşıya kalınmaktadır. Bu optimizasyon problemini çözmek amacıyla ABC algoritmasının tamsayı versiyonu bu çalışmada kullanılmıştır. Kullanılan ABC algoritmasındaki kontrol parametreleri ise kovan popülasyonu, limit değeri ve durdurma kriteridir. (Akay ve Karaboga, 2009; Kiran ve Findik, 2015; Akay ve Karaboga, 2020) yaptıkları çalışmalarda ABC algoritmasının yüksek popülasyona ihtiyaç duymadığını göstermişlerdir. Bu sebeple kovandaki popülasyon sayısı literatüre uygun şekilde yiyecek kaynağı sayısının 2 katı olacak şekilde seçilerek 40 şeklinde belirlenmiştir. Literatürde limit değeri popülasyona ve karar değişkeni sayısına bağlı olarak Denklem 3.1’deki gibi hesaplanmaktadır (Karaboga ve Basturk, 2008; Karaboga ve Akay, 2009; Kiran ve Findik, 2015).

$$limit = (N \times D)/2 \quad (3.1)$$

Denklem 3.1.'de verilen N değeri popülasyon kullanılan arı sayısı, D ise karar değişkeni sayısıdır. Yapılan çalışmada popülasyon boyutu olarak 40 alınmıştır ve karar değişkeninin sayısı ise 14'tür. Kullanılan bu değerlere göre limit değeri 280 bulunmaktadır. Durdurma kriteri ise 500'dür ve maksimum çevrim sayısına eşittir. ELM test ve eğitimleri için ise rastgele örneklem yöntemi kullanılmış ve veri kümesinin %90'ı ağı eğitmek için, %10'u ise ağı eğitiminin sona ermesi için test etmek için kullanılmıştır. Optimizasyon aşamasında ELM'ler 30'ar kere eğitilmiş ve teste tabi tutulmuştur. Bunun amacı ise bu aşamada tüm çözümlerin kalitesini belirlemektir. Çözümlerin kalitelerinin hesaplanması amacıyla ortalama hata kullanılmaktadır. Bunun neticesinde problem bir minimizasyon problemi (ELM sınıflandırma hatasının minimizasyonu) şeklinde düşünülmüştür. Denklem 2.2'de yapılan hesaplamada hata büyüdükçe uygunluğun hata ile orantılı şekilde düşük olacağı, hata küçüldükçe ise uygunluğun artacağı gösterilmiştir. k -ELM'nin gizli katmanındaki nöron sayısı karar değişkeni değeri 5 ile 50 arasında ve aktivasyon fonksiyonundaki karar değişkeni sayısının 1 ile 6 arasında bir tamsayı değer alacaktır. Bu bağlamda farklı ELM kümesi $1.2200105E+17$ eleman içermektedir ve bunların içerisinde 7 tanesini belirlemek üstel zaman gerektirmektedir. Bundan dolayı bu çalışmada ABC algoritması bu problemin üstesinden gelmek amacıyla uygulanmıştır. Aktivasyon fonksiyonlarının indekslenmesinde sigmoid (Sig) için 1, sinüs (Sin) için 2, katı sınırlayıcı (Hardlim) için 3, üçgensel temel (triangular basis-Tribas) için 4, radyal temel (radial basis-Radbas) için 5, tanjant sigmoid (TanSig) için 6, ilgili ELM'nin ayarlarında kullanılan değerlerdir. Çizelge 3.7'de, bu ayarlar ve şartlar ile ABC kullanılarak oluşturulan ağlara ait bilgiler bulunmaktadır.

Çizelge 3.7. ABC ile oluşturulan ELM'ler

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	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

Çizelge 3.7'ye baktığımız zaman aktivasyon fonksiyonları içerisinde Sig'in 10 kere, Hardlim'in 26 kere, Tribas'ın 11 kere, Radbas'ın 5 kere ve Tansig'in 18 kere kullanıldığı görülürken Sin fonksiyonunun 0(sıfır) kere yani hiçbir ELM'de kullanılmadığı görülmektedir.

ABC kullanılarak tek katmanlı ileri beslemeli YSA oluşturulmuştur. Daha sonra bu YSA'larda ELM'ler her bir veri kümeleri için 100'er kere test ve eğitime tabi tutulmuştur. Bu 100 kere çalıştırmada da elde edilen verilerin %90'ı ağı eğitimi için kullanılmıştır. Kalan %10'luk veri ise ağı test etmek için kullanılmıştır. k-ELM'de k değerinin 1 olarak alındığı özel bir durum bulunmaktadır. Bu durumdaki ELM algoritması temel ELM şeklinde oluşturulmuştur. k değerinin 1 alındığı temel ELM algoritmasında gizli katmandaki nöron sayısı 50'dir. Temel ELM, 6 tane aktivasyon

fonksiyonu için de ayrı ayrı tüm veri kümeleri için koşturulmuştur. Sonuçları ayrıca çizelgelerde raporlanmıştır. Verilen çizelgelerde 100 kere çalıştırılıp elde edilmiş sonuçlar içerisinde sınıflandırma performansı, ortalama test performansı ve standart sapma değerleri en iyi olanlar verilmektedir. Elde edilen sonuçların hepsi aynı çizelgede verildiğinde analiz edilmesi oldukça zorlaşmıştır ve ayrıca sayfa yapısında da bozulmalar görülmüştür. Bu sebeple elde edilen sonuçları üç çizelgeye bölünüp o şekilde verilmiştir. Bu üç çizelgeden Çizelge 3.8, Sigmoid ve Sinüs fonksiyonlarının aktivasyon olarak kullanıldığı Temel ELM'lerin, önerilen yaklaşım ile karşılaştırılmasını içermektedir. Çizelge 3.9, ise Hardlim ve Tribas fonksiyonlarının aktivasyon olarak kullanıldığı Temel ELM'lerin, önerilen yaklaşım ile karşılaştırılmasını içermektedir. Üç çizelgeden sonuncusu olan Çizelge 3.10 ise Radbas ve Tanjant Sigmoid fonksiyonlarının aktivasyon olarak kullanıldığı Temel ELM'lerin, önerilen yaklaşım ile karşılaştırılmasını içermektedir.

Çizelge 3.8. Önerilen yaklaşımın Sigmoid ve Sinüs aktivasyon fonksiyonlarına sahip temel ELM ile karşılaştırılması

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

Çizelge 3.9. Önerilen yaklaşımın Hardlim ve Tribas aktivasyon fonksiyonlarına sahip temel ELM ile karşılaştırılması

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

Çizelge 3.10. Önerilen yaklaşımın Radbas ve TanSig aktivasyon fonksiyonlarına sahip temel ELM ile karşılaştırılması

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

Çizelge 3.8, 3.9 ve 3.10'dan anlaşıldığı gibi önerilen yaklaşımın uygulandığı tüm veri setlerinde karşılaştırılan ELM yapılarının tümünden daha iyi sonuçlar üretmiştir. Temel ELM'de en kötü performans, aktivasyon fonksiyonu olarak Tribas ve Sin kullanıldığında ortaya çıkmaktadır. Önerilen yöntem en yakın sonuçlar, temel ELM'de aktivasyon fonksiyonu olarak Hardlim kullanıldığında üretilmiştir.

3.2.1. Yaklaşımın Performans Analizi ve Karışıklık Matrisleri

Karışıklık matrisleri oluşturulurken ABC ile optimize edilmiş olan ELM'ler için rastgele örneklem yaklaşımıyla veri seti önce %90 eğitim ve %10 test kümesi olarak ikiye ayrılmıştır. Daha sonra önerilen yaklaşım ile ELM'ler eğitilmiş ve sınıflandırma yapılmıştır. Bu işlem 100 defa tekrarlanmış olup Çizelge 3.11'de genel metrikler, Çizelge 3.12-3.21'de de karışıklık matrisleri raporlanmıştır. Çizelgelerden görüleceği üzere doğruluk değerleri karşılaştırmalar ile yakın olmasına rağmen birebire aynı değildir. Bunun sebebi optimizasyon sürecinde tamamlandıktan sonra karışıklık matrislerinin elde edilen optimum değerler ile tekrar eğitim ve teste tabi tutulmasından kaynaklanmaktadır. Bilindiği üzere ELM'de giriş katmanı ile gizli katman arasındaki ağırlıklar rastgele atanmakta olduğundan dolayı bu rastgelelik sınıflandırıcının performansını çok az da olsa etkilemektedir.

Çizelge 3.11. Önerilen Yaklaşımın Performans Analiz Tablosu

	Doğruluk (Accuracy)	Özgünlük (Specificity)	Kesinlik (Precision)	Hatırlama (Recall)	F1-Skoru (F-Measure)
D1	0,6583	0,8735	0,6872	0,3767	0,5066
D2	0,6386	0,8410	0,7387	0,4459	0,5873
D3	0,7351	0,9629	0,7856	0,2811	0,4023
D4	0,7650	0,9452	0,8277	0,4467	0,5662
D5	0,7403	0,8375	0,6902	0,5748	0,5973
D6	0,6990	0,6202	0,6883	0,7816	0,7223
D7	0,6013	0,6678	0,5215	0,5229	0,5484
D8	0,7117	0,8928	0,6715	0,3880	0,6616
D9	0,7407	0,8870	0,8195	0,5844	0,6713
D10	0,7391	0,8716	0,7649	0,5533	0,6338

Çizelge 3.12. Bilgisayar Mühendisliğine Giriş (D1)'in Karışıklık Matrisi

		Girmedi	Girdi	
		0	1	
Girmedi	0	0,1525	0,2658	Gerçek
Girdi	1	0,0758	0,5058	
		Tahmin Edilen		

Oransal Gösterim

		Girmedi	Girdi	
		0	1	
Girmedi	0	19	33	Gerçek
Girdi	1	9	63	
		Tahmin Edilen		

Sayısal Gösterim

Çizelge 3.13. Atatürk İlke ve İnkılap Tarihi (D2)'in Karışıklık Matrisi

		Girmedi	Girdi	
		0	1	
Girmedi	0	0,2025	0,2705	Gerçek
Girdi	1	0,0999	0,4272	
		Tahmin Edilen		

Oransal Gösterim

		Girmedi	Girdi	
		0	1	
Girmedi	0	14	19	Gerçek
Girdi	1	7	30	
		Tahmin Edilen		

Sayısal Gösterim

Çizelge 3.14. Fizik 1 (D3)'ün Karışıklık Matrisi

		Girmedi	Girdi	
		0	1	
Girmedi	0	0,0921	0,2400	Gerçek
Girdi	1	0,0249	0,6431	
		Tahmin Edilen		

Oransal Gösterim

		Girmedi	Girdi	
		0	1	
Girmedi	0	36	94	Gerçek
Girdi	1	10	251	
		Tahmin Edilen		

Sayısal Gösterim

Çizelge 3.15. İşletim Sistemleri (D4)'ün Karışıklık Matrisi

		Girmedi	Girdi	
		0	1	
Girmedi	0	0,1505	0,2005	Gerçek
Girdi	1	0,0345	0,6145	
		Tahmin Edilen		

Oransal Gösterim

		Girmedi	Girdi	
		0	1	
Girmedi	0	29	39	Gerçek
Girdi	1	7	120	
		Tahmin Edilen		

Sayısal Gösterim

Çizelge 3.16. Matematik 1 (D5)'in Karışıklık Matrisi

		Girmedi	Girdi	
		0	1	
Girmedi	0	0,2074	0,1549	Gerçek
Girdi	1	0,1049	0,5328	
		Tahmin Edilen		

Oransal Gösterim

		Girmedi	Girdi	
		0	1	
Girmedi	0	81	61	Gerçek
Girdi	1	41	208	
		Tahmin Edilen		

Sayısal Gösterim

Çizelge 3.17. Matematik 2 (D6)'nın Karışıklık Matrisi

		Girmedi	Girdi	
		0	1	
Girmedi	0	0,3944	0,1122	Gerçek
Girdi	1	0,1888	0,3046	
		Tahmin Edilen		

Oransal Gösterim

		Girmedi	Girdi	
		0	1	
Girmedi	0	196	56	Gerçek
Girdi	1	94	151	
		Tahmin Edilen		

Sayısal Gösterim

Çizelge 3.18. Türk Dili 1 (D7)'nin Karışıklık Matrisi

		Girmedi	Girdi	Gerçek
		0	1	
Girmedi	0	0,2038	0,1900	
Girdi	1	0,2088	0,3975	
		Tahmin Edilen		

Oransal Gösterim

		Girmedi	Girdi	Gerçek
		0	1	
Girmedi	0	17	16	
Girdi	1	18	33	
		Tahmin Edilen		

Sayısal Gösterim

Çizelge 3.19. Türk Dili 2 (D8)'in Karışıklık Matrisi

		Girmedi	Girdi	Gerçek
		0	1	
Girmedi	0	0,1563	0,2262	
Girdi	1	0,1027	0,5149	
		Tahmin Edilen		

Oransal Gösterim

		Girmedi	Girdi	Gerçek
		0	1	
Girmedi	0	10	14	
Girdi	1	6	32	
		Tahmin Edilen		

Sayısal Gösterim

Çizelge 3.20. Veri Yapıları (D9)'un Karışıklık Matrisi

		Girmedi	Girdi	Gerçek
		0	1	
Girmedi	0	0,2779	0,1979	
Girdi	1	0,0614	0,4628	
		Tahmin Edilen		

Oransal Gösterim

		Girmedi	Girdi	Gerçek
		0	1	
Girmedi	0	81	58	
Girdi	1	18	135	
		Tahmin Edilen		

Sayısal Gösterim

Çizelge 3.21. Tüm Dersler (D10)'un Karışıklık Matrisi

		Girmedi	Girdi	
		0	1	
Girmedi	0	0,2286	0,1852	Gerçek
Girdi	1	0,0757	0,5105	
		Tahmin Edilen		

Oransal Gösterim

		Girmedi	Girdi	
		0	1	
Girmedi	0	482	390	Gerçek
Girdi	1	160	1076	
		Tahmin Edilen		

Sayısal Gösterim

4. SONUÇLAR VE ÖNERİLER

4.1 Sonuçlar

Aşırı öğrenme makinesi ile yapılan ilk çalışmada, yeni bir sınıflandırma problemine ELM, BP ve metasezgisel algoritmalar tarafından eğitilmiş bir YSA modeli uygulanmıştır. YSA, 10x50x1 tek gizli katmanlı ileri beslemeli ağıdır. Toplam optimize edilmiş ağırlıklar ve sapmalar 601'dir (550 ağırlık, 51 sapma). Deneyler hem normalleştirilmiş hem de normalleştirilmemiş veri kümeleri üzerinde gerçekleştirilmiştir. BP algoritması ağı hemen hemen tüm veri setlerinde mükemmel şekilde eğitmesine rağmen, test performansı Şekil 3.1 ve Şekil 3.2'ye göre diğer algoritmalarından çok daha düşüktür. ELM'nin eğitim performansı, metasezgisel algoritmaların çoğundan daha iyidir, test performansı çoğu metasezgisel algoritmadan daha iyi değildir. Bu test sonuçları, BP ve ELM'nin eğitim örneklerini ezberlediğini ve test örneklerini tanıyamadıklarını göstermektedir. Metasezgisel algoritmalar eğitim YSA üzerinde kabul edilebilir bir performans sergilemekte ve özellikle GWO, GA, BBO testlerindeki performansları Şekil 3.1 ve Şekil 3.2'ye göre BP ve ELM'den daha iyidir. Algoritmaların eğitim süresi göz önüne alındığında, ELM, karşılaştırılan algoritmalarından daha üstündür çünkü karşılaştırılan algoritmalar, YSA modelini eğitmek için yinelemeli bir süreç kullanır. Çalışma süresi performansı, eğitim süresi tüm veri kümelerinde bir saniyeden az olduğu için çok daha iyidir.

Bu çalışmada, bütünleme sınavına giren öğrenci sayısının tahmini yeni bir sınıflandırma problemi olarak sunulmuş ve bu probleme yapay sinir ağı tabanlı algoritmalar uygulanmıştır. Veriler Konya Teknik Üniversitesi'nden toplanmıştır ve her veri seti bir dersin öğrencilerinin verilerinden oluşmaktadır. Bu şekilde dokuz veri seti oluşturulmuş ve onuncu veri seti tüm verilerden oluşmakta ve tüm veri setleri normalize edilmiştir. Toplanan ve normalize edilen veri seti üzerinde BP, ELM ve metasezgisel algoritmalar ile eğitilmiş bir YSA modeli, öğrencilerin bütünleme sınavına katılıp katılmadıklarını sınıflandırmak için uygulanmıştır. Elde edilen sonuçlar eğitim doğruluğu, test doğruluğu ve eğitim süresi açısından birbirleriyle karşılaştırılmıştır. BP algoritmalarının eğitim doğruluğu açısından karşılaştırmalı algoritmalarından daha iyi olduğu, GWO algoritmasının test doğruluğu açısından karşılaştırmalı algoritmalarından daha iyi olduğu ve veri setleri üzerinde eğitim süresi açısından ELM'nin diğer algoritmalarından daha iyi olduğu gösterilmiştir. Daha yüksek düzeyde eğitim ve test

doğruluğu gerektirdiğinden, yakın gelecekte bu sorunu bir topluluk stratejisi içinde aşmak için bu algoritmaları geliştirip değiştirilmesi hedeflenmiştir.

ABC ile optimize edilmiş aşırı öğrenme makinesi çalışmasında k-ELM adı verilen ELM'nin çoklu bir versiyonu geliştirilmiş ve Yapay Sinir Ağının eğitiminde kullanılmıştır. Ek genel sınavlara girmesi beklenen öğrenci sayılarını bulmak için sınıflandırmada k-ELM algoritması kullanılmıştır. k-ELM algoritmasındaki k değeri 7 olarak alınmıştır. Buradaki 7 değeri sınıflandırma oylamasına katılacak ELM sayısını belirtmektedir. k-ELM algoritmamız 10 veri setimize de uygulanmıştır ve gizli katmandaki nöron sayılarını optimize etmek için ABC algoritması önerilmiştir. Optimizasyonda kullanılan ABC algoritmasında 6 tane aktivasyon fonksiyonu ve gizli katmanındaki nöron sayıları 5—50 aralığında olmak suretiyle toplamda $1,22E+17$ farklı ELM'den 7 tanesi bulmak amacıyla kullanılmıştır. Bulunan 7 ELM performansı en yüksek olan 7 tanesidir. Çizelge 3.7 de sunulduğu üzere, ABC algoritması maksimum performansa sahip olan 7 ELM yapısı kurmuş ve 500 iterasyonda 40 arıyla çözüm uzay araştırması yapmıştır. Gizli katmandaki nöron sayısını, 6 farklı aktivasyon fonksiyonu için 50 olarak aynı veri kümelerinde temel ELM algoritması çalıştırılmıştır. Bu çalıştırılma sonucunda ulaştığımız sonuçlar önerilen yaklaşım ile karşılaştırılmıştır. Karşılaştırma sonuçları Çizelge 3.8, 3.9 ve 3.10'da verilmiştir. ABC algoritmasının başarısı Çizelge 3.7, 3.8, 3.9 ve 3.10 beraber değerlendirildiğinde görülmektedir. Bu başarının sebebi Hardlim'dir. ABC algoritması Hardlim'i, 10 veri kümesinde 7 farklı ELM toplamda oluşturulan 70 ELM yapısından 26'sında kullanmaktadır. Çizelge 3.9'da Hardlim'in temel ELM algoritmasında kullanıldığında da önerilen yöntemle en yakın sonuçları elde ettiği görülmektedir. Çizelge 3.7'de aktivasyon fonksiyonu olarak Tribas'ın 11 defa kullanıldığı görülmektedir. Buna rağmen Tribas ile oluşturulan ELM yapısında en kötü sonuçların alındığı Çizelge 3.7'de görülmektedir. Buna rağmen Çizelge 3.7'deki satırlara bakıldığında Sin, Radbas ve Tribas aktivasyon fonksiyonları ile oluşturulmuş ELM yapılarının, aktivasyon fonksiyonu olarak önerilen yöntemle en yakın sonuçları üreten Sig, Tansig ve Hardlim'in kullanılarak oluşturulmuş ELM'lere kıyasla ağırlıklı olarak yer almadıkları görülmektedir. Sonuç olarak ABC ile oluşturulmuş ELM'lerin sınıflandırma performansının, temel ELM'lerde aktivasyon fonksiyonunun ve gizli katman sayısının sabit atanarak oluşturulan ELM'lerin sınıflandırma performansından daha iyi olduğu görülmektedir.

4.2 Öneriler

Bu tezde daha önce değinilmemiş bir problem olan bütünleme sınavına girecek öğrenci sayılarının sınıflandırma yaklaşımı ile tahminlemesi yapılmış ve literatüre yeni bir problem önerilmiştir. Önerilen problemin çözümü için yüksek genelleme kabiliyetine sahip ELM tabanlı çözümler önerilmiştir. İlk çalışmada temel ELM algoritması ve diğer algoritmalar ile eğitilmiş yapay sinir ağlarının performansları oluşturulmuş olan veri setleri üzerine kıyaslanmıştır. İkinci çalışmada ise çoklu ELM tabanlı bir yaklaşımın parametreleri ABC ile optimize edilmiş ve farklı aktivasyon fonksiyonlarına sahip ELM'ler ile karşılaştırılmıştır. Gelecek çalışmalarda araştırmacılar için aşağıda verilen çalışmaların yapılabilceğı önerilmektedir.

- ✓ Veri setinin kayıt sayısının artmasıyla performansın iyileştiğı gözlemlendiğinden dolayı daha büyük veri setlerinin oluşturulması ile daha yüksek performanslı sonuçlar elde edilebilir.
- ✓ Veri setindeki özellik sayısı artırılarak sınıflandırma performansı yükseltilebilir.
- ✓ ELM yerine farklı sınıflandırıcıların uygulanması ile kapsamlı bir performans karşılaştırması yapılabilir.
- ✓ Farklı evrimsel hesaplama yöntemlerinin performansı tezde öne sürülen problem üzerinde araştırılabilir.
- ✓ Bu tez çalışmasında sınıflandırma tabanlı bir yaklaşım ile öğrencilerin bilgileri kullanarak bütünlemeye girecek öğrenci sayısı tespit edilmeye çalışılmıştır. Ders (dersi daha önce alan öğrenci sayısı, dersin zorluk derecesi, dersin kredisi vb. öznitelikler) verileri kullanılarak programlama tabanlı bir yaklaşım da aynı problemin çözümü için geliştirilebilir.
- ✓ Bu çalışmada önerilen yaklaşımlar teknolojiye dönüştürülerek öğrenci bilgi sistemlerine entegre edilebilir.

KAYNAKLAR

- Akay, B. ve Karaboga, D., 2009, Parameter tuning for the artificial bee colony algorithm, *International conference on computational collective intelligence*, 608-619.
- Akay, B. ve Karaboga, D., 2020, Artificial Bee Colony Algorithm, *Swarm Intelligence Algorithms: A Tutorial*, 17.
- Alcin, O. F., Ucar, F. ve Korkmaz, D., 2016, Extreme learning machine based robotic arm modeling, *2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, 1160-1163.
- Avansas.
- Bai, Z., Huang, G.-B., Wang, D., Wang, H. ve Westover, M. B., 2014, Sparse extreme learning machine for classification, *IEEE transactions on cybernetics*, 44 (10), 1858-1870.
- Balasundaram, S., Gupta, D. ve Kapil, 2014, 1-Norm extreme learning machine for regression and multiclass classification using Newton method, *Neurocomputing*, 128, 4-14.
- Barron, A. R., 1993, Universal approximation bounds for superpositions of a sigmoidal function, *IEEE Transactions on Information theory*, 39 (3), 930-945.
- Benoît, F., Van Heeswijk, M., Miche, Y., Verleysen, M. ve Lendasse, A., 2013, Feature selection for nonlinear models with extreme learning machines, *Neurocomputing*, 102, 111-124.
- Branke, J., 1995, Evolutionary algorithms for neural network design and training, *In Proceedings of the first Nordic workshop on genetic algorithms and its applications*.
- Cai, Z., Gu, J., Luo, J., Zhang, Q., Chen, H., Pan, Z., Li, Y. ve Li, C., 2019, Evolving an optimal kernel extreme learning machine by using an enhanced grey wolf optimization strategy, *Expert Systems with Applications*, 138, 112814.
- Cao, J., Lin, Z., Huang, G.-B. ve Liu, N., 2012, Voting based extreme learning machine, *Information Sciences*, 185 (1), 66-77.
- Cao, L., Zhu, P., Zhao, Y. ve Zhao, J., 2018, Using machine learning and quantum chemistry descriptors to predict the toxicity of ionic liquids, *Journal of hazardous materials*, 352, 17-26.
- Chakravarthy, S. S. ve Rajaguru, H., 2022, Automatic detection and classification of mammograms using improved extreme learning machine with deep learning, *IRBM*, 43 (1), 49-61.
- Chen, H., Peng, J., Zhou, Y., Li, L. ve Pan, Z., 2014, Extreme learning machine for ranking: generalization analysis and applications, *Neural Networks*, 53, 119-126.
- Chen, L., Zhang, Y.-H., Huang, G., Pan, X., Wang, S., Huang, T. ve Cai, Y.-D., 2018, Discriminating cirRNAs from other lncRNAs using a hierarchical extreme learning machine (H-ELM) algorithm with feature selection, *Molecular genetics and genomics*, 293 (1), 137-149.
- Chen, X., Dong, Z. Y., Meng, K., Xu, Y., Wong, K. P. ve Ngan, H., 2012, Electricity price forecasting with extreme learning machine and bootstrapping, *IEEE Transactions on Power Systems*, 27 (4), 2055-2062.
- Cinar, A. C., 2020, Training feed-forward multi-layer perceptron artificial neural networks with a tree-seed algorithm, *Arabian Journal for Science and Engineering*, 45 (12), 10915-10938.
- Cortes, C. ve Vapnik, V., 1995, Support vector machine, *Machine learning*, 20 (3), 273-297.

- da Costa, N. L., Llobodanin, L. A. G., de Lima, M. D., Castro, I. A. ve Barbosa, R., 2018, Geographical recognition of Syrah wines by combining feature selection with Extreme Learning Machine, *Measurement*, 120, 92-99.
- Deng, J., Li, K. ve Irwin, G. W., 2011, Fast automatic two-stage nonlinear model identification based on the extreme learning machine, *Neurocomputing*, 74 (16), 2422-2429.
- Du, D., Li, K., Irwin, G. W. ve Deng, J., 2013, A novel automatic two-stage locally regularized classifier construction method using the extreme learning machine, *Neurocomputing*, 102, 10-22.
- Duan, J., Ou, Y., Hu, J., Wang, Z., Jin, S. ve Xu, C., 2017, Fast and stable learning of dynamical systems based on extreme learning machine, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49 (6), 1175-1185.
- Eke, I., Taplamacioglu, M. C. ve Kocaarslan, I., 2011, 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.
- Fernández-Delgado, M., Cernadas, E., Barro, S., Ribeiro, J. ve Neves, J., 2014, Direct Kernel Perceptron (DKP): Ultra-fast kernel ELM-based classification with non-iterative closed-form weight calculation, *Neural Networks*, 50, 60-71.
- Gallant, A. R. ve White, H., 1988, There exists a neural network that does not make avoidable mistakes, *ICNN*, 657-664.
- Gao, Z., Yu, J., Zhao, A., Hu, Q. ve Yang, S., 2022, A hybrid method of cooling load forecasting for large commercial building based on extreme learning machine, *Energy*, 238, 122073.
- Geng, Z., Zhao, S., Tao, G. ve Han, Y., 2017, Early warning modeling and analysis based on analytic hierarchy process integrated extreme learning machine (AHP-ELM): Application to food safety, *Food Control*, 78, 33-42.
- Guo, M., Ma, Y., Yang, X. ve Mankin, R. W., 2019, Detection of damaged wheat kernels using an impact acoustic signal processing technique based on Gaussian modelling and an improved extreme learning machine algorithm, *biosystems engineering*, 184, 37-44.
- Hagan, M. T. ve Menhaj, M. B., 1994, Training Feedforward Networks with the Marquardt Algorithm, *Ieee Transactions on Neural Networks*, 5 (6), 989-993.
- He, Q., Du, C., Wang, Q., Zhuang, F. ve Shi, Z., 2011, A parallel incremental extreme SVM classifier, *Neurocomputing*, 74 (16), 2532-2540.
- Horata, P., Chiewchanwattana, S. ve Sunat, K., 2013, Robust extreme learning machine, *Neurocomputing*, 102, 31-44.
- Hornik, K., 1991, Approximation capabilities of multilayer feedforward networks, *Neural Networks*, 4 (2), 251-257.
- <https://www.veribilimiokulu.com/>.
- Huang, F., Yin, K., Huang, J., Gui, L. ve Wang, P., 2017, Landslide susceptibility mapping based on self-organizing-map network and extreme learning machine, *Engineering Geology*, 223, 11-22.
- Huang, G.-B. ve Babri, H. A., 1998, Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions, *Ieee Transactions on Neural Networks*, 9 (1), 224-229.
- Huang, G.-B., Zhu, Q.-Y. ve Siew, C.-K., 2004a, Extreme learning machine: a new learning scheme of feedforward neural networks, *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, 985-990.

- Huang, G.-B., Chen, L. ve Siew, C. K., 2006a, Universal approximation using incremental constructive feedforward networks with random hidden nodes, *IEEE Trans. Neural Networks*, 17 (4), 879-892.
- Huang, G.-B., Zhu, Q.-Y. ve Siew, C.-K., 2006b, Extreme learning machine: theory and applications, *Neurocomputing*, 70 (1-3), 489-501.
- Huang, G.-B. ve Chen, L., 2007a, Convex incremental extreme learning machine, *Neurocomputing*, 70 (16-18), 3056-3062.
- Huang, G.-B. ve Chen, L., 2008, Enhanced random search based incremental extreme learning machine, *Neurocomputing*, 71 (16-18), 3460-3468.
- Huang, G.-B., Zhou, H., Ding, X. ve Zhang, R., 2011, Extreme learning machine for regression and multiclass classification, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42 (2), 513-529.
- Huang, G., Song, S., Gupta, J. N. ve Wu, C., 2014, Semi-supervised and unsupervised extreme learning machines, *IEEE transactions on cybernetics*, 44 (12), 2405-2417.
- Huang, G., Huang, G.-B., Song, S. ve You, K., 2015, Trends in extreme learning machines: A review, *Neural Networks*, 61, 32-48.
- Huang, G. B., Zhu, Q. Y. ve Siew, C. K., 2004b, Extreme learning machine: A new learning scheme of feedforward neural networks, *2004 Ieee International Joint Conference on Neural Networks, Vols 1-4, Proceedings*, 985-990.
- Huang, G. B. ve Chen, L., 2007b, Convex incremental extreme learning machine, *Neurocomputing*, 70 (16-18), 3056-3062.
- Huang, G. B., Zhou, H. M., Ding, X. J. ve Zhang, R., 2012, Extreme Learning Machine for Regression and Multiclass Classification, *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, 42 (2), 513-529.
- Huang, S.-C. ve Huang, Y.-F., 1991, Bounds on the number of hidden neurons in multilayer perceptrons, *Ieee Transactions on Neural Networks*, 2 (1), 47-55.
- Jiang, Y., Bi, A., Xia, K., Xue, J. ve Qian, P., 2020a, Exemplar-based data stream clustering toward Internet of Things, *The Journal of Supercomputing*, 76 (4), 2929-2957.
- Jiang, Y., Zhang, Y., Lin, C., Wu, D. ve Lin, C.-T., 2020b, EEG-based driver drowsiness estimation using an online multi-view and transfer TSK fuzzy system, *IEEE Transactions on Intelligent Transportation Systems*, 22 (3), 1752-1764.
- Kaloop, M. R., El-Badawy, S. M., Ahn, J., Sim, H.-B., Hu, J. W. ve Abd El-Hakim, R. T., 2022, A hybrid wavelet-optimally-pruned extreme learning machine model for the estimation of international roughness index of rigid pavements, *International Journal of Pavement Engineering*, 23 (3), 862-876.
- Kang, X., Liu, C., Zeng, S., Zhao, Z., Qian, J. ve Zhao, Y., 2018, Prediction of Henry's law constant of CO₂ in ionic liquids based on SEP and S σ -profile molecular descriptors, *Journal of Molecular Liquids*, 262, 139-147.
- Karaboga, D. ve Basturk, B., 2007, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, 39 (3), 459-471.
- Karaboga, D. ve Basturk, B., 2008, On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing*, 8 (1), 687-697.
- Karaboga, D. ve Akay, B., 2009, A comparative study of Artificial Bee Colony algorithm, *Applied Mathematics and Computation*, 214 (1), 108-132.
- Kasun, L. L. C., Zhou, H., Huang, G.-B. ve Vong, C. M., 2013, Representational learning with ELMs for big data.

- Kiran, M. S. ve Findik, O., 2015, A directed artificial bee colony algorithm, *Applied Soft Computing*, 26, 454-462.
- Kiran, M. S., Siramkaya, E., Esme, E. ve Senkaya, M. N., 2022, Prediction of the number of students taking make-up examinations using artificial neural networks, *International Journal of Machine Learning and Cybernetics*, 13 (1), 71-81.
- Kuang, Y. X., Wu, Q., Shao, J. K., Wu, J. F. ve Wu, X. H., 2017, Extreme learning machine classification method for lower limb movement recognition, *Cluster Computing-the Journal of Networks Software Tools and Applications*, 20 (4), 3051-3059.
- Lahoz, D., Lacruz, B. ve Mateo, P. M., 2013, A multi-objective micro genetic ELM algorithm, *Neurocomputing*, 111, 90-103.
- Lan, Y., Soh, Y. C. ve Huang, G.-B., 2009, Ensemble of online sequential extreme learning machine, *Neurocomputing*, 72 (13-15), 3391-3395.
- Le, Q., Sarlós, T. ve Smola, A., 2013, Fastfood-approximating kernel expansions in loglinear time, *Proceedings of the international conference on machine learning*.
- Lei, H., Wen, Y., You, Z., Elazab, A., Tan, E.-L., Zhao, Y. ve Lei, B., 2018, Protein-protein interactions prediction via multimodal deep polynomial network and regularized extreme learning machine, *IEEE journal of biomedical and health informatics*, 23 (3), 1290-1303.
- Li, B., Li, Y. ve Rong, X., 2013, The extreme learning machine learning algorithm with tunable activation function, *Neural Computing and Applications*, 22 (3), 531-539.
- Li, D., Li, S., Zhang, S., Sun, J., Wang, L. ve Wang, K., 2022, Aging state prediction for supercapacitors based on heuristic kalman filter optimization extreme learning machine, *Energy*, 250, 123773.
- Li, J., Shi, X., You, Z.-H., Yi, H.-C., Chen, Z., Lin, Q. ve Fang, M., 2020, Using weighted extreme learning machine combined with scale-invariant feature transform to predict protein-protein interactions from protein evolutionary information, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17 (5), 1546-1554.
- Li, K., Peng, J. X. ve Irwin, G. W., 2005, A fast nonlinear model identification method, *Ieee Transactions on Automatic Control*, 50 (8), 1211-1216.
- Li, W., Chen, C., Su, H. ve Du, Q., 2015, Local binary patterns and extreme learning machine for hyperspectral imagery classification, *IEEE Transactions on Geoscience and Remote Sensing*, 53 (7), 3681-3693.
- Li, Z., Wei, L., Li, W., Wei, L., Chen, M., Lv, M., Zhi, X., Wang, C. ve Gao, N., 2019, Research on DDoS attack detection based on ELM in IoT environment, *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, 144-148.
- Liang, N.-Y., Huang, G.-B., Saratchandran, P. ve Sundararajan, N., 2006, A fast and accurate online sequential learning algorithm for feedforward networks, *Ieee Transactions on Neural Networks*, 17 (6), 1411-1423.
- Lin, S., Liu, X., Fang, J. ve Xu, Z., 2014, Is extreme learning machine feasible? A theoretical assessment (Part II), *IEEE Transactions on Neural Networks and Learning Systems*, 26 (1), 21-34.
- Liu, B., Yan, S., You, H., Dong, Y., Li, J., Li, Y., Lang, J. ve Gu, R., 2017a, An ensemble RBF extreme learning machine to forecast road surface temperature, *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 977-980.
- Liu, J., Li, Q., Chen, W., Yan, Y. ve Wang, X., 2018, A fast fault diagnosis method of the PEMFC system based on extreme learning machine and Dempster-Shafer

- evidence theory, *IEEE Transactions on Transportation Electrification*, 5 (1), 271-284.
- Liu, T., Yang, Y., Huang, G.-B., Yeo, Y. K. ve Lin, Z., 2015, Driver distraction detection using semi-supervised machine learning, *IEEE Transactions on Intelligent Transportation Systems*, 17 (4), 1108-1120.
- Liu, X., Lin, S., Fang, J. ve Xu, Z., 2014, Is extreme learning machine feasible? A theoretical assessment (Part I), *IEEE Transactions on Neural Networks and Learning Systems*, 26 (1), 7-20.
- Liu, Y., Li, X., Wang, J., Chen, F., Wang, J., Shi, Y. ve Zheng, L., 2017b, Pattern Discovery from Big Data of Food Sampling Inspections Based on Extreme Learning Machine, *International Conference on Research and Practical Issues of Enterprise Information Systems*, 132-142.
- Man, Z., Lee, K., Wang, D., Cao, Z. ve Miao, C., 2011, A new robust training algorithm for a class of single-hidden layer feedforward neural networks, *Neurocomputing*, 74 (16), 2491-2501.
- Martínez-Martínez, J. M., Escandell-Montero, P., Soria-Olivas, E., Martín-Guerrero, J. D., Magdalena-Benedito, R. ve Gómez-Sanchis, J., 2011, Regularized extreme learning machine for regression problems, *Neurocomputing*, 74 (17), 3716-3721.
- McCulloch, W. S. ve Pitts, W., 1943, A logical calculus of the ideas immanent in nervous activity, *The bulletin of mathematical biophysics*, 5 (4), 115-133.
- Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C. ve Lendasse, A., 2009, OP-ELM: optimally pruned extreme learning machine, *Ieee Transactions on Neural Networks*, 21 (1), 158-162.
- Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C. ve Lendasse, A., 2010, OP-ELM: Optimally Pruned Extreme Learning Machine, *Ieee Transactions on Neural Networks*, 21 (1), 158-162.
- Mohammed, A. A., Minhas, R., Wu, Q. M. J. ve Sid-Ahmed, M. A., 2011, Human face recognition based on multidimensional PCA and extreme learning machine, *Pattern Recognition*, 44 (10-11), 2588-2597.
- Moore, E. H., 1920, On the reciprocal of the general algebraic matrix, *Bull. Am. Math. Soc.*, 26, 394-395.
- Musikawan, P., Sunat, K., Chiewchanwattana, S., Horata, P. ve Kongsorot, Y., 2016, Improved Convex Incremental Extreme Learning Machine Based on Ridgelet and PSO Algorithm, *2016 13th International Joint Conference on Computer Science and Software Engineering (Jcsse)*, 189-194.
- Niu, M., Zhang, J., Li, Y., Wang, C., Liu, Z., Ding, H., Zou, Q. ve Ma, Q., 2020, CirRNAPL: a web server for the identification of circRNA based on extreme learning machine, *Computational and structural biotechnology journal*, 18, 834-842.
- Niu, W.-J., Feng, Z.-K., Feng, B.-F., Min, Y.-W., Cheng, C.-T. ve Zhou, J.-Z., 2019, Comparison of multiple linear regression, artificial neural network, extreme learning machine, and support vector machine in deriving operation rule of hydropower reservoir, *Water*, 11 (1), 88.
- Oneto, L., Fumero, E., Clerico, G., Canepa, R., Papa, F., Dambra, C., Mazzino, N. ve Anguita, D., 2017, Dynamic delay predictions for large-scale railway networks: Deep and shallow extreme learning machines tuned via thresholdout, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47 (10), 2754-2767.
- Ouyang, Q., Chen, Q., Zhao, J. ve Lin, H., 2013, Determination of amino acid nitrogen in soy sauce using near infrared spectroscopy combined with characteristic

- variables selection and extreme learning machine, *Food and Bioprocess Technology*, 6 (9), 2486-2493.
- Ouyang, T., Wang, C., Yu, Z., Stach, R., Mizaikoff, B., Huang, G.-B. ve Wang, Q.-J., 2020, NOx measurements in vehicle exhaust using advanced deep ELM networks, *IEEE Transactions on Instrumentation and Measurement*, 70, 1-10.
- Ozturk, C., Hancer, E. ve Karaboga, D., 2014, Automatic Clustering with Global Best Artificial Bee Colony Algorithm, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 29 (4), 677-687.
- Penrose, R., 1955, A generalized inverse for matrices, *Mathematical proceedings of the Cambridge philosophical society*, 406-413.
- Rahimi, A. ve Recht, B., 2007, Random Features for Large-Scale Kernel Machines, *NIPS*, 5.
- Rahimi, A. ve Recht, B., 2008a, Uniform approximation of functions with random bases, *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, 555-561.
- Rahimi, A. ve Recht, B., 2008b, Weighted sums of random kitchen sinks: replacing minimization with randomization in learning, *Nips*, 1313-1320.
- Rajpal, A., Mishra, A. ve Bala, R., 2019, A novel fuzzy frame selection based watermarking scheme for MPEG-4 videos using bi-directional extreme learning machine, *Applied Soft Computing*, 74, 603-620.
- Rathore, S. ve Park, J. H., 2018, Semi-supervised learning based distributed attack detection framework for IoT, *Applied Soft Computing*, 72, 79-89.
- Ray, P. ve Mishra, D., 2015, Application of extreme learning machine for underground cable fault location, *International Transactions on Electrical Energy Systems*, 25 (12), 3227-3247.
- Razavi-Far, R., Chakrabarti, S., Saif, M., Zio, E. ve Palade, V., 2018, Extreme Learning Machine Based Prognostics of Battery Life, *International Journal on Artificial Intelligence Tools*, 27 (8).
- Romero Merino, E., 2001, Function aproximation with SAOCIF: a general sequential method and a particular algorithm with feed-forward neural networks.
- Rong, H.-J., Huang, G.-B., Sundararajan, N. ve Saratchandran, P., 2009, Online sequential fuzzy extreme learning machine for function approximation and classification problems, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39 (4), 1067-1072.
- Rumelhart, D. E., Hinton, G. E. ve Williams, R. J., 1986, Learning Representations by Back-Propagating Errors, *Nature*, 323 (6088), 533-536.
- Rumelhart, D. E., McClelland, J. L. ve Group, P. R., 1988, Parallel distributed processing, IEEE New York, p.
- Sahu, P., Sarangi, P. K., Mohapatra, S. K. ve Sahoo, B. K., 2022, Detection and Classification of Encephalon Tumor Using Extreme Learning Machine Learning Algorithm Based on Deep Learning Method, In: *Biologically Inspired Techniques in Many Criteria Decision Making*, Eds: Springer, p. 285-295.
- Sartori, M. A. ve Antsaklis, P. J., 1991, A simple method to derive bounds on the size and to train multilayer neural networks, *Ieee Transactions on Neural Networks*, 2 (4), 467-471.
- Saxe, A. M., Koh, P. W., Chen, Z., Bhand, M., Suresh, B. ve Ng, A. Y., 2011, On random weights and unsupervised feature learning, *Icml*.
- She, Q., Hu, B., Gan, H., Fan, Y., Nguyen, T., Potter, T. ve Zhang, Y., 2018, Safe semi-supervised extreme learning machine for EEG signal classification, *IEEE Access*, 6, 49399-49407.

- Shoumo, S. Z. H., Dhruva, M. I. M., Hossain, S., Ghani, N. H., Arif, H. ve Islam, S., 2019, Application of machine learning in credit risk assessment: a prelude to smart banking, *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, 2023-2028.
- Sokolov-Mladenovic, S., Milovancevic, M., Mladenovic, I. ve Alizamir, M., 2016, 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.
- Stinchcombe, M., 1989, Universal approximation using feed-forward networks with nonsigmoid hidden layer activation functions, *Proc. IJCNN, Washington, DC*, 1989, 161-166.
- Sun, C., Yu, Y., Liu, H. ve Gu, J., 2015, Robotic grasp detection using extreme learning machine, *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1115-1120.
- Sun, Z.-L., Ng, K. M., Soszynska-Budny, J. ve Habibullah, M. S., 2011, Application of the LP-ELM model on transportation system lifetime optimization, *IEEE Transactions on Intelligent Transportation Systems*, 12 (4), 1484-1494.
- Suykens, J. A. ve Vandewalle, J., 1999, Least squares support vector machine classifiers, *Neural processing letters*, 9 (3), 293-300.
- Turkoglu, M. ve Hanbay, D., 2019, 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.
- Ucar, A., 2014, Color Face Recognition Based on Steerable Pyramid Transform and Extreme Learning Machines, *Scientific World Journal*.
- Wang, N., Er, M. J. ve Han, M., 2014, Parsimonious extreme learning machine using recursive orthogonal least squares, *IEEE Transactions on Neural Networks and Learning Systems*, 25 (10), 1828-1841.
- Wang, R., Chow, C.-Y., Lyu, Y., Lee, V. C., Kwong, S., Li, Y. ve Zeng, J., 2017, TaxiRec: Recommending road clusters to taxi drivers using ranking-based extreme learning machines, *IEEE Transactions on Knowledge and Data Engineering*, 30 (3), 585-598.
- Wei, X., Li, C., Huang, W., Gu, J. ve Zhu, M., 2019a, Sediment-carrying capacity prediction using extreme learning machine, *IOP Conference Series: Earth and Environmental Science*, 042006.
- Wei, X., Liu, H., She, X., Lu, Y., Liu, X. ve Mo, S., 2019b, Stability Assessment of Rubble Mound Breakwaters Using Extreme Learning Machine Models, *Journal of Marine Science and Engineering*, 7 (9), 312.
- Widrow, B., Greenblatt, A., Kim, Y. ve Park, D., 2013, The no-prop algorithm: A new learning algorithm for multilayer neural networks, *Neural Networks*, 37, 182-188.
- Wilamowski, B. M. ve Yu, H., 2010, Neural Network Learning without Backpropagation, *Ieee Transactions on Neural Networks*, 21 (11), 1793-1803.
- Wong, P. K., Huang, W., Vong, C. M. ve Yang, Z., 2020, Adaptive neural tracking control for automotive engine idle speed regulation using extreme learning machine, *Neural Computing and Applications*, 32 (18), 14399-14409.
- Xi-Zhao, W., Qing-Yan, S., Qing, M. ve Jun-Hai, Z., 2013, Architecture selection for networks trained with extreme learning machine using localized generalization error model, *Neurocomputing*, 102, 3-9.
- Xing, Y.-m., Ban, X.-j. ve Liu, R., 2018, A short-term traffic flow prediction method based on kernel extreme learning machine, *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 533-536.

- Xu, C., Lei, H., Li, J., Ye, J. ve Zhang, D., 2019, Adaptive neural control for nonaffine pure-feedback system based on extreme learning machine, *Mathematical Problems in Engineering*, 2019.
- Yang, Y., Wang, Y. ve Yuan, X., 2012, Bidirectional extreme learning machine for regression problem and its learning effectiveness, *IEEE Transactions on Neural Networks and Learning Systems*, 23 (9), 1498-1505.
- Yang, Y., Wang, Y. ve Yuan, X., 2013, Parallel chaos search based incremental extreme learning machine, *Neural processing letters*, 37 (3), 277-301.
- Yao, X., 1993, A Review of Evolutionary Artificial Neural Networks, *International Journal of Intelligent Systems*, 8 (4), 539-567.
- Ye, Y., Squartini, S. ve Piazza, F., 2013, Online sequential extreme learning machine in nonstationary environments, *Neurocomputing*, 116, 94-101.
- Yi, Y., Dai, J., Wang, C., Hou, J., Zhang, H., Liu, Y. ve Gao, J., 2019, An effective framework using spatial correlation and extreme learning machine for moving cast shadow detection, *Applied Sciences*, 9 (23), 5042.
- Yin, Y., Dong, X. ve Xu, T., 2018, Rapid and efficient bug assignment using ELM for IOT software, *IEEE Access*, 6, 52713-52724.
- You, Z.-H., Zhou, M., Luo, X. ve Li, S., 2016, Highly efficient framework for predicting interactions between proteins, *IEEE transactions on cybernetics*, 47 (3), 731-743.
- Yu, J., Song, W., Zhou, G. ve Hou, J.-j., 2019, Violent scene detection algorithm based on kernel extreme learning machine and three-dimensional histograms of gradient orientation, *Multimedia Tools and Applications*, 78 (7), 8497-8512.
- Yu, Q., Miche, Y., Eirola, E., Van Heeswijk, M., Séverin, E. ve Lendasse, A., 2013, Regularized extreme learning machine for regression with missing data, *Neurocomputing*, 102, 45-51.
- Zeng, Y., Xu, X., Shen, D., Fang, Y. ve Xiao, Z., 2016, Traffic sign recognition using kernel extreme learning machines with deep perceptual features, *IEEE Transactions on Intelligent Transportation Systems*, 18 (6), 1647-1653.
- Zhang, R., Zhou, L., Zuo, M., Zhang, Q., Bi, M., Jin, Q. ve Xu, Z., 2018, Prediction of dairy product quality risk based on extreme learning machine, *2018 2nd international conference on data science and business analytics (ICDSBA)*, 448-456.
- Zhang, X., He, T., Lu, L., Yue, S., Cheng, D. ve Xu, X., 2017, Video analysis of traffic accidents based on projection extreme learning machine, *2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 149-154.
- Zhang, X. ve Ma, H., 2019, Data-Driven Model-Free Adaptive Control Based on Error Minimized Regularized Online Sequential Extreme Learning Machine, *Energies*, 12 (17), 3241.
- Zhao, J., Wang, Z. ve Park, D. S., 2012, Online sequential extreme learning machine with forgetting mechanism, *Neurocomputing*, 87, 79-89.
- Zheng, D., Hong, Z., Wang, N. ve Chen, P., 2020, An improved LDA-based ELM classification for intrusion detection algorithm in IoT application, *Sensors*, 20 (6), 1706.
- Zong, W. ve Huang, G.-B., 2011a, Face recognition based on extreme learning machine, *Neurocomputing*, 74 (16), 2541-2551.
- Zong, W. W. ve Huang, G. B., 2011b, Face recognition based on extreme learning machine, *Neurocomputing*, 74 (16), 2541-2551.

- Zong, W. W., Zhou, H. M., Huang, G. B. ve Lin, Z. P., 2011, Face Recognition Based on Kernelized Extreme Learning Machine, *Autonomous and Intelligent Systems*, 6752, 263-272.
- Zong, W. W., Huang, G. B. ve Chen, Y. Q., 2013, Weighted extreme learning machine for imbalance learning, *Neurocomputing*, 101, 229-242.

