T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YAPAY ARI KOLONİSİ TEMELLİ LOJİSTİK REGRESYON SINIFLAYICILARIN OPTİMAL TASARIMI VE TÜRKÇE SPAM MAİLLERİN FİLTRELENMESİNDE BAŞARIMLARININ İNCELENMESİ

Hazırlayan
Bilge Kağan DEDETÜRK

Danışman
Prof. Dr. Bahriye AKAY

Doktora Tezi

Aralık 2020
KAYSERİ

T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YAPAY ARI KOLONİSİ TEMELLİ LOJİSTİK REGRESYON SINIFLAYICILARIN OPTİMAL TASARIMI VE TÜRKÇE SPAM MAİLLERİN FİLTRELENMESİNDE BAŞARIMLARININ İNCELENMESİ

(Doktora Tezi)

Hazırlayan
Bilge Kağan DEDETÜRK

Danışman
Prof. Dr. Bahriye AKAY

Bu çalışma, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TUBİTAK) tarafından 116E947 kodlu proje ile desteklenmiştir

Aralık 2020
KAYSERİ

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Bilge Kağan DEDETÜRK

İmza: 

YÖNERGEYE UYGUNLUK

"Yapay Arı Kolonisi Temelli Lojistik Regresyon Sınıflayıcıların Optimal Tasarımı ve Türkçe Spam Maillerin Filtrelenmesinde Başarımlarının İncelenmesi" adlı Doktora tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi'ne uygun olarak hazırlanmıştır.

Tezi Hazırlayan
Bilge Kağan DEDETÜRK

Danışman
Prof. Dr. Bahriye AKAY

Elektrik ve Bilgisayar Mühendisliği ABD Başkanı
Prof. Dr. Veysel ASLANTAŞ

Prof. Dr. Bahriye AKAY danışmanlığında **Bilge Kağan DEDETÜRK** tarafından hazırlanan **"Yapay Arı Kolonisi Temelli Lojistik Regresyon Sınıflayıcıların Optimal Tasarımı ve Türkçe Spam Maillerin Filtrelenmesinde Başarımlarının İncelenmesi"** adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektrik ve Bilgisayar Mühendisliği Anabilim Dalında **Doktora** tezi olarak kabul edilmiştir.

14/12/2020

JÜRİ :

Danışman : Prof. Dr. Bahriye AKAY

Üye : Prof. Dr. Derviş KARABOĞA

Üye : Prof. Dr. Nurhan KARABOĞA

Üye : Doç. Dr. Mesut GÜNDÜZ

Üye : Doç. Dr. Mustafa Servet KIRAN

ONAY :

Bu tezin kabülü Enstitü Yönetim Kurulunun ..... tarih ve ..... sayılı kararı ile onaylanmıştır.

..... / 12 /2020
Prof. Dr. Mehmet AKKURT
Enstitü Müdürü

TEŞEKKÜR

Bana çalışmalarım süresince her türlü yardımı ve fedakârlığı sağlayan, başta doktora tez danışmanım Prof. Dr. Bahriye AKAY olmak üzere anneme, babama, ablama ve eşime teşekkürlerimi sunarım.

Türkiye Bilimsel ve Teknolojik Araştırma Kurumuna (TUBİTAK) "2228-B Yüksek Lisans Öğrencileri için Doktora Burs Programı" kapsamında doktora eğitimim boyunca verdikleri burstan dolayı teşekkür etmek isterim. Verdikleri destek sayesinde doktora eğitimime kesintisiz bir şekilde devam edebildim. Yine TUBİTAK'a 116E947 proje numarası altında tez çalışmalarıma verdikleri destekten dolayı minnetimi sunarım.

Tez İzleme Komitesinde yer alan Prof. Dr. Derviş KARABOĞA ve Prof. Dr. Nurhan KARABOĞA hocalarıma sağladıkları değerli bilgilerden ve yönlendirmelerden dolayı teşekkürlerimi sunarım.

YAPAY ARI KOLONİSİ TEMELLİ LOJİSTİK REGRESYON SINIFLAYICILARIN OPTİMAL TASARIMI VE TÜRKÇE SPAM MAİLLERİN FİLTRELENMESİNDE BAŞARIMLARININ İNCELENMESİ

Bilge Kağan DEDETÜRK
Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Doktora Tezi, Aralık 2020
Danışman : Prof. Dr. Bahriye AKAY

ÖZET

Spam e-posta, iletişim kalitesini düşüren, alıcıları rahatsız eden ve zamanlarını boşa harcayan ciddi bir problemdir. Bu problemin çözümü, e-postaları spam veya normal olarak sınıflandırmak ve spam e-postaları elemektir. Bu sınıflandırma problemini çözmek için pek çok yöntem önerilmiştir. Makine öğrenmesi yöntemleri bir e-postayı istenen veya istenmeyen olarak sınıflandırmada etkili olduklarından dolayı spam tespit sistemlerinde yaygın olarak kullanılmaktadır. Fakat mevcut spam tespit teknikleri genellikle düşük tespit oranlarından muzdarip olurlar ve etkili bir şekilde karmaşık ve büyük boyutlu verilerin üstesinden gelemezler. Bu problemlerin üstesinden gelmek amacıyla bu tez çalışmasında üç yeni yöntem (ABC-LR, CSA-LR, ABC-CSA-LR) önerilmiştir. Veri kümeleri üzerinde yapılan deneylerin sonuçları, önerilen yöntemlerin özellikle de ABC-LR ve ABC-CSA-LR yöntemlerinin sahip olduğu yüksek oranda etkili bölgesel ve küresel arama yeteneklerinden dolayı çok boyutlu verilerle başa çıkabildiğini göstermektedir. Önerilen yöntemlerin literatürde bilinen makine öğrenmesi algoritmalarıyla kıyaslanmalarına ek olarak geçmiş çalışmalardan literatürde bilinen ve güncel yöntemlerle de karşılaştırılmıştır. Sınıflandırma doğruluğu açısından önerilen yöntemlerin bu çalışmada göz önünde bulundurulan diğer spam tespit tekniklerinden daha üstün olduğu deneysel olarak gösterilmiştir. Ayrıca Türkçe ve İngilizce veri kümeleri üzerinde gösterdikleri yüksek başarıdan dolayı önerilen yöntemlerin diller arasındaki yapısal farklılıkların üstesinden gelebildikleri anlaşılmaktadır.

Anahtar Kelimeler: Spam e-posta, spam e-posta tespiti, spam filtreleme, Türkçe e-posta, makine öğrenmesi, lojistik regresyon, optimizasyon algoritmaları, yapay zeka, yapay arı kolonisi, yapay bağışıklık sistemi, klonal seçim algoritması

OPTIMAL DESIGN OF ARTIFICIAL BEE COLONY BASED LOGISTIC REGRESSION CLASSIFIERS AND ANALYSIS OF THEIR PERFORMANCES IN FILTERING TURKISH SPAM E-MAILS

Bilge Kağan DEDETÜRK
Erciyes University, Graduate School of Natural and Applied Sciences
Ph.D. Thesis, December 2020
Supervisor: Prof. Dr. Bahriye AKAY

ABSTRACT

Spam e-mail is a serious problem that reduces communication quality, annoys recipients and wastes their time. The solution to this problem is to classify e-mails as spam or normal and eliminate spam e-mails. Many methods have been proposed to solve this classification problem. Machine learning methods are widely used in spam detection systems because they are effective in classifying an email as desired or unsolicited. However, current spam detection techniques often suffer from low detection rates and cannot effectively deal with complex and high-dimensional data. In order to overcome these problems, three novel methods (ABC-LR, CSA-LR, ABC-CSA-LR) have been proposed in this study. The results of experiments on datasets show that the proposed methods, especially the ABC-LR and ABC-CSA-LR, are able to cope with multi-dimensional data due to their highly effective local and global search capabilities. In addition to comparing the proposed methods with the state-of-the-art machine learning algorithms, they were also compared with the state-of-the-art methods reported by previous studies. It has been experimentally demonstrated that the proposed methods outperforms other spam detection techniques considered in this study in terms of classification accuracy. In addition, it is understood that the proposed methods can efficiently overcome the structural differences between languages due to their quite successful classification performances on Turkish and English data sets.

Keywords: E-mail spam, e-mail spam detection, spam filtering, Turkish e-mail, machine learning, logistic regression, optimization algorithms, artificial intelligence, artificial bee colony, artificial immune system, clonal selection algorithm

İÇİNDEKİLER

YAPAY ARI KOLONİSİ TEMELLİ LOJİSTİK REGRESYON SINIFLAYICILARIN OPTİMAL TASARIMI VE TÜRKÇE SPAM MAİLLERİN FİLTRELENMESİNDE BAŞARIMLARININ İNCELENMESİ

KISALTMALAR

ABC	:	Yapay Arı Kolonisi (Artificial Bee Colony)
AIS	:	Yapay Bağışıklık Sistemi (Artificial Immune System)
NB	:	Naif Bayes (Naive Bayes)
SVM	:	Destek Vektör Makinesi (Support Vector Machine)
RBF	:	Radial Tabanlı Fonksiyon (Radial Basis Function)
CSA	:	Klonal Seçim Algoritması (Clonal Selection Algorithm)
NSA	:	Negatif Seçim Algoritması (Negative Selection Algorithm)
PSO	:	Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
DE	:	Diferansiyel Gelişim (Differential Evolution)
FN	:	Yanlış Negatif (False Negative)
FP	:	Yanlış Pozitif (False Pozitive)
FVS	:	Öznitelik Vektör Boyutu (Feature Vector Size)
LR	:	Lojistik Regresyon (Logistic Regression)
MCN	:	Maksimum Döngü Sayısı
MR	:	Modifikasyon oranı
ANOVA	:	Varyans Analizi (Analysis of Variance)
TUBİTAK	:	Türkiye Bilimsel ve Teknolojik Araştırma Kurumu

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

GİRİŞ

E-ticaret şirketlerinin sayısındaki artış reklam amaçlı gönderilen e-postaların (e-mail) sayısında artışa neden olmaktadır. Yine internet üzerinden alışverişin popülerliğinin artması müşterilere ait banka bilgilerinin ve şifre gibi gizli bilgilerinin çalınmasına yönelik e-postaları da arttırmaktadır. Ayrıca e-posta sayesinde neredeyse maliyetsiz bir şekilde çok sayıda insana ulaşma fikri spam gönderen kişileri ve reklamcıları ciddi manada cezbetmektedir. Bu tarz istenmeyen ve ayrım gözetmeksizin çok sayıda kişiye gönderilen e-postalar spam olarak adlandırılmaktadır [2]. Statista tarafından yayınlanan raporda [3] 2017 yılında dünya çapında 3.7 milyar e-posta kullanıcısı olduğu belirtilmiş ve bu sayının 2022 yılının sonuna kadar 4.3 milyara ulaşılacağı tahmin edilmiştir. Radicati tarafından yayınlanan rapora [4] göre 2019 yılında günlük 293 milyarın üzerinde e-postanın gönderileceği ve alınacağı belirtilmiş ve 2023 yılının sonuna kadar bu sayının 347 milyarı aşacağı tahmin edilmiştir. Bir başka rapora [5] göre 2017 ve 2018 yıllarında spam mesajların dünya çapında e-posta trafiğinin %55'ini oluşturduğu belirtilmiştir. Symantec tarafından yayınlanan raporda [6] ise 2015, 2016 ve 2017 yıllarında spam e-posta oranlarının sırasıyla %52.7, %53.4 ve %54.6 olduğu belirtilmiştir. Kaspersky [7] 2018'in üçüncü çeyreğinin sonuna kadar spam e-posta oranını %53.49 olarak raporlamıştır. Tüm bu yayınlanan istatistikler ışığında e-posta kullanıcıları tarafından alınan spam e-postaların çok büyük sayılara ulaştığı ve gönderilen her iki e-postadan en az birinin spam e-posta olduğu anlaşılmaktadır. Spam e-postalar hem alıcılarını rahatsız eden, hem de ağ trafiğinin, sınırlı e-posta kutusu depolama alanının, bellek ve işlemci gibi kaynakların boşa harcanmasına neden olan ciddi sorunlar oluşturabilmektedirler. [8]. Ayrıca eklerinde kötü amaçlı yazılımlar içerebilirler ve bu yazılımların açılmasıyla enfekte oldukları sistemdeki bilgilerin çalınmasından sistemin çökmesine kadar çok ciddi zararlar verebilmektedirler.

Spam e-postaları önlemeye yönelik pek çok yöntem geliştirilmiştir. Bhowmick ve

Hazarika [8] spam e-postaları azaltmak veya engellemek için kullanılan yöntemleri sezgisel filtreler, kara listeler, beyaz listeler, meydan okuma karşılık verme sistemleri, işbirlikçi spam filtreleme, bal kavanozları, imza şemaları ve makine öğrenmesi tabanlı yöntemler olmak üzere sekiz grupta kategorilendirmektedir. Daha genel bir kategorilendirme ise spam e-postaları önlemeye yönelik yöntemleri statik ve dinamik yöntemler olmak üzere iki gruba ayırmaktadır [9]. Statik yöntemler, önceden tanımlanan beyaz listeleri ve kara listeleri dikkate alan filtreleme yaklaşımlarına dayanmaktadır. Fakat spam e-posta gönderen kişilerin kolaylıkla farklı e-posta adreslerini kullanmaları bu yöntemleri etkisiz kılmaktadır. Dinamik yöntemler ise genellikle istatistiksel ya da makine öğrenmesi tekniklerine dayanan metin sınıflandırma yöntemlerini kullanarak bir e-postanın spam olup olmadığına karar vermek için e-posta içeriğini dikkate alırlar.

Bu tez çalışmasında, bir e-postanın spam olup olmadığının tespiti için optimize edilmiş sınıflandırma yöntemleri tasarlanmış ve bu yöntemlerin başarımları Türkçe ve İngilizce dillerindeki veri kümeleri üzerinde incelenmiştir. Bu yöntemlerden birincisinde, Lojistik Regresyon (Logistic Regression - LR) sınıflandırma yöntemini bal arılarının yiyecek arama davranışlarını taklit eden ve doğadan ilham alan bir topluluk zekâsı algoritması olan Yapay Arı Kolonisi (Artificial Bee Colony - ABC) [10] algoritması kullanılarak eğiten ABC-LR isimli yeni bir spam filtreleme yaklaşımı önerilmektedir. Bu spam filtreleme yaklaşımı, LR'nin yerel minimuma takılma probleminden kurtulmasını sağlarken aynı zamanda LR'nin basitlik, gerçek zamanlı uygulamalarda hızlı sınıflandırma ve kolay kodlanabilirlik gibi avantajlarını [11] korumaktadır. Bu çalışma literatür taramalarından edinilen bilgiye göre LR'nin öğrenme algoritması olarak ABC algoritmasını kullanan ilk spam filtreleme yöntemidir. ABC algoritması çok boyutlu ve kompleks problemlerde yüksek başarılar göstermesine ek olarak dengeli keşif ve faydalanma yeteneği sergilemektedir [12, 13]. Bu özellikleri onu spam filtreleme problemleri için iyi bir alternatif yapmaktadır. Ayrıca önerilen yöntem LR'nin denk öznitelik ağırlıklarına hassasiyetiyle ilişkili olan bir diğer dezavantajını terim frekansı-ters doküman frekansı ($tf - idf$) yöntemiyle entegrasyon sayesinde azaltmaktadır.

Bu tez çalışmasında önerilen ikinci yöntem, LR sınıflandırma modelinin eğitilmesi aşamasında Yapay Bağışıklık Sistemi (Artificial Immune System - AIS) algoritmasını

kullanmaktadır. AIS, memelileri patojenlere karşı koruyan gerçek bağışıklık sisteminden esinlenen bir yöntemdir [14]. Negatif seçim algoritması (negative selection algorithm - NSA) [15] ve klonal seçim algoritması (clonal selection algorithm - CSA) [16] AIS yönteminin literatürde bilinen algoritmalarıdır. Bu algoritmalardan NSA, sistemin kendine ait olmayanı tanıyabilen detektörler olan T-hücrelerinin yardımıyla kendine ait olan ve olmayan uzayı birbirinden ayırt edebilmektedir. Kendine ait olan uzay ile sistem hücreleri kastedilirken kendine ait olmayan uzay ile sistemdeki yabancı girdiler kastedilmektedir. Sistemin kendine ait olmayanlarının tespiti, spam tespitine oldukça benzediğinden dolayı AIS algoritmalarını kullanan e-posta spam tespiti ile alakalı çalışmaların çoğu NSA algoritmalarına dayanmaktadır. Fakat NSA algoritmasının, kendine ait olmayan uzayı tamamen kaplamak için çok fazla sayıda detektöre ihtiyaç duymasına ek olarak zaman ve uzay karmaşıklığı çok fazladır [15] ve düşük tespit oranlarına sahiptir [17]. CSA, benzerlik olgunlaşması (affinity maturation) prensibi sayesinde daha yüksek benzerliğe sahip antikorlar üreterek antijenik bir uyarana adaptif bağışıklık sisteminin verdiği cevabı taklit eder. Standart CSA ve çeşitli versiyonları optimizasyon, örüntü tanıma ve sınıflandırma gibi farklı tür problemlere uygulanmaktadır [16, 18–20]. Optimizasyon problemleri için önerilen CSA'nın temel versiyonu, hiper-mutasyon işlemlerini kullanarak çözüm uzayında yerel arama gerçekleştirir ve antikorları yerel olarak optimize eder. Ayrıca çözüm uzayının daha geniş keşfini sağlayan yeni antikorların üretildiği reseptör düzenleme (receptor editing) sürecini kullanarak küresel arama gerçekleştirir. Çözüm uzayında hem yerel hem de küresel çözüm arama özellikleri CSA'yı optimizasyon problemlerinin çözümü için uygun bir seçenek yapmaktadır [16]. CSA tabanlı yöntemlerin çeşitli uygulamalarda hem geleneksel optimizasyon tekniklerinden [16, 21] hem de birçok diğer bio-ilham algoritmalarından [22] daha üstün olduğu çalışmalarda belirtilmiştir. Bu çalışmada ikinci yöntem olarak LR sınıflandırma modelinin optimize edilmesinde CSA'yı kullanan CSA-LR isimli yeni bir hibrit sınıflandırma modeli önerilmektedir.

CSA, optimizasyon problemleri üzerinde kayda değer bir başarı göstermesine rağmen çözüm uzayını arama başarısı yeterli değildir [23] ve çözüm uzayını arama yetenekleri ile kararlılığı arttırılarak CSA'nın performansı iyileştirilebilmektedir [19, 24, 25]. Bu yüzden, ABC algoritmasının dikkat çeken yerel arama performansı göz

önünde bulundurularak ABC-CSA-LR isimli üçüncü bir sınıflandırma yöntemi bu tez çalışmasında önerilmektedir. Bu önerilen yeni yöntem, CSA'nın kararlılığını ve yerel arama yeteneğini iyileştirmek için CSA'da yer alan hiper-mutasyon aşamasının yerine ABC algoritmasında yer alan işçi ve gözcü arı aşamaları koyularak elde edilmiştir.

Tezin birinci bölümünde problemin tanımı, problemle alakalı literatür özeti, tez çalışmasının amacı ve önemi açıklanmaktadır. İkinci bölümde bu tez çalışmasında kullanılan literatürde bilinen makine öğrenmesi ve optimizasyon algoritmaları özet bir şekilde anlatılırken üçüncü bölümde önerilen yöntemler anlatılmaktadır. Dördüncü bölümde bu çalışmada kullanılan veri kümelerinden, veri ön-işleme adımlarından ve öznitelik seçme yöntemlerinden bahsedilmektedir. Beşinci bölümde deneylerin ve kıyaslamaların sonuçları paylaşılmaktadır. Son bölümde ise sonuç ve öneriler ifade edilmektedir.

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1. Problem Tanımı

Kolay, hızlı ve ucuz iletişim özelliklerinden dolayı e-postalar tüm dünyada bir iletişim aracı olarak yaygın bir şekilde kullanılmaktadır. Ayrıca e-posta sayesinde aynı içerik çok sayıda insana aynı anda zahmetsizce gönderilebilmektedir. Giriş bölümünde de bahsedildiği üzere milyarlarca insan iletişim yöntemi olarak e-posta kullanmaktadır ve bir günde yüz milyarlarca e-posta gönderilip alınmaktadır. Fakat e-posta kullanıcıları tarafından alınan e-postalar içerisinde onları rahatsız eden ve istemedikleri e-postalar da bulunmaktadır. Bu tarz e-postalara literatürde spam e-posta denilmektedir. Neredeyse maliyetsiz olarak çok sayıda insana ulaşma fikri, spam gönderen kişileri ve reklamcıları fazlasıyla cezbetmektedir. Reklamcılar ve spam gönderen kişiler aynı anda milyonlarca e-posta kullanıcısına çok kolay bir şekilde e-posta gönderebilmektedir. Spam e-postalardan bazıları reklam amacı taşırken bazıları ise oltalama adı verilen yöntemle kullanıcılara ait banka hesap bilgileri, kimlik bilgileri, parola bilgileri gibi çok önemli ve gizli bilgileri çalabilmekte veya eklerinde virüs içererek e-posta kullanıcılarına ciddi zararlar verebilmektedir. Yine giriş bölümünde verilen istatistikler ışığında kullanıcılara gelen e-postaların yarısından fazlasının spam e-posta olduğu anlaşılmaktadır ve gelen e-postalardan gerekli olanların tespit edilebilmesi için spam e-postaların ayıklanması gerekmektedir. Bu durum ise e-posta kullanıcılarının ciddi manada zaman kaybetmelerine neden olmaktadır. Spam e-postaların kullanıcılara verdiği zararlar ve zaman kayıpları göz önünde bulundurulduğunda bu tarz e-postaların otomatik bir şekilde ve yüksek başarıyla tespit edilmesinin önemi ve gerekliliği anlaşılmaktadır.

E-posta spam tespiti, bir e-postanın $d_i \in D = \{d_1, d_2, ..., d_{|D|}\}$ bir sınıfa $c_j \in C = \{c_{spam}, c_{normal}\}$ atandığı bir tür sınıflandırma problemidir. Burada D e-postaların

kümesini temsil ederken C spam ve normal olmak üzere iki sınıfı temsil etmektedir. Sınıflandırma algoritması çalıştırılmadan önce bir e-posta sırasıyla terimlere ayırma, metin olmayan terimleri çıkarma, kelimeleri yalın hale getirme, gereksiz terimleri çıkarma ve öznitelikleri belirleme işlemleri uygulanılarak elde edilen anlamlı bir öznitelik uzayı ile temsil edilir. Terimlere ayırma aşamasında bir e-postada yer alan tüm terimler (kelimeler, noktalama işaretleri, sayılar, html etiketleri vb.) birbirlerinden ayrılır. Metin olmayan terimleri çıkarma aşamasında noktalama işaretleri, html etiketleri ve boşluklar çıkarılır. Yalın hale getirme aşamasında kelimelerin anlamını değiştirmeyen çekim ekleri çıkarılarak kelimeler yalın hale getirilir. Gereksiz terimleri çıkarma aşamasında herhangi bir sınıf için ayırt edici özelliği olmayan kelimeler (ve, veya, bu vb.) çıkartılır. Son olarak tüm e-postalarda yer alan tüm benzersiz kelimeleri içeren kelime havuzu (bag of words) modeli oluşturulur. Daha sonra her bir e-posta $d_i \in D, \vec{d_i} = [w_1, w_2, \cdots, w_{|B|})]$ vektörü ile temsil edilir. Bu vektörde yer alan $w_1, w_2, \cdots, w_{|B|}$ değerleri $t_1, t_2, \cdots, t_{|B|}$ terimlerine karşılık gelen ağırlıklardır. Bu ağırlıklar ikili ağırlıklandırma (bw), terim frekansı (tf), terim frekansı – zıt doküman frekansı ($tf - idf$) gibi terim ağırlıklandırma yöntemleriyle hesaplanabilir. $|B|$ değeri ise kelime havuzunda yer alan toplam benzersiz kelime sayısını ifade etmektedir. Öznitelik vektör boyutu yüksek olabileceğinden ve bu durumun hesaplama karmaşıklığını ve maliyetini arttırabileceğinden dolayı öznitelik vektörünün boyutunu düşürmek için bir öznitelik seçim yöntemi uygulanılabilir. Öznitelik vektör boyutu uygun bir sayıya düşürüldükten sonra Eşitlik 1.1'de olduğu gibi bir e-postayı sınıflardan birine $c_j \in C = \{c_{spam}, c_{normal}\}$ atamak için bir sınıflandırma yöntemi (ϕ) kullanılır.

$$\phi(\vec{d_i'}, c_j) = \begin{cases} 1, & \vec{d_i'} \in c_j \quad \text{ise} \\ 0, & \text{aksi takdirde} \end{cases} \tag{1.1}$$

1.2. Literatür Özeti

NB sınıflandırıcılar, öznitelik vektöründeki her bir özniteliğin birbirinden bağımsız olduğu naif varsayımına dayanarak çok değişkenli problemi bir grup tek değişkenli problemlere dönüştüren olasılıksal bir sınıflandırma yöntemidir [26]. NB basitlik, hızlı yakınsama, lineer hesaplama karmaşıklığı ve kolay yorumlanma gibi

özelliklerinden dolayı spam filtreleme problemlerinde sıklıkla kullanılmaktadır [26–28]. Androutsopoulos ve arkadaşları [29] NB'nin anahtar terim tabanlı filtreleme (keyword-based filtering) yönteminden daha üstün olduğunu göstermişlerdir. Otomatik bir şekilde eğitilebilir anti spam filtrelerinin mümkün olduğunu ve sınıflandırma performansı üzerinde önemli bir etkiye sahip olabildiğini vurgulamışlardır. Fakat Rusland ve arkadaşları [30] e-posta türünün ve veri kümesindeki örnek sayısının NB performansını etkilediği sonucuna varmışlardır. Onlara göre NB sınıflandırıcılar, veri kümesi daha az sayıda örnek e-postadan oluştuğunda ve daha az sayıda özniteliğe sahip olduğunda yüksek başarı gösterebilmektedir. Almeida ve arkadaşları [31] Bayesian teorisine dayanan spam filtreleme alanı için boyut azaltmada terim seçim yöntemleri üzerine bir çalışma sunmuşlardır ve NB performansının seçilen öznitelikler ile sayısına yüksek derecede duyarlı olduğunu göstermişlerdir. Feng ve arkadaşları [32] NB'nin gerçek dünya uygulamalarında genellikle doğru olmayan eğitim kümesindeki özniteliklerin birbirinden bağımsız olduğu varsayımından dolayı uygulanabilirliğinin sınırlı olduğunu raporlamışlardır.

Bir diğer literatürde bilinen spam filtreleme tekniği, verileri analiz eden ve kategorilendirme için kullanılan örüntüleri belirleyen SVM sınıflandırıcıdır. SVM'ler ikinci dereceden problemleri çözerek değişkenler arasındaki ilişkiyi öğrenirler ve karar hiper düzlemine en yakın olan pozitif ve negatif veri noktaları arasındaki mesafeyi maksimum yaparlar [33]. SVM sınıflandırıcılar metin sınıflandırma ve spam filtreleme problemlerinde yüksek başarı sergilerler [34–36]. Amayri ve Bouguila [35] spam filtreleme için SVM sınıflandırıcılara yönelik detaylı bir çalışma gerçekleştirmiştir ve çeşitli mesafe tabanlı yöntemler önermişlerdir. Yu ve Xu [37] SVM'leri spam tespiti için uygulamışlardır ve karmaşıklıklarına rağmen sınıflandırma doğruluklarını ve hızlı test zamanlarını vurgulamışlardır. [38] çalışmasında iki farklı SVM yaklaşımı önerilmiştir. Klasik SVM'in yanlış pozitif (FP) oranını iyileştirmek için SVM, birinci yaklaşımda aşamalı olarak eğitilirken ikincisinde öznitelik kümesi sezgisel bir şekilde güncellenmiştir. Skulley ve Wachman [36] SVM'in hesaplama maliyetini düşürmek için maksimum marjin gereksinimini esneten çevrimiçi bir SVM yöntemi önermişlerdir ve benzer sonuçlar elde etmişlerdir. Bhowmick ve Hazarika [8] tarafından hazırlanan derleme makalesinde ise SVM'lerin yüksek başarılar göstermelerine rağmen hesaplama

maliyetlerinden ve FP oranlarından muzdarip oldukları ifade edilmiştir.

Spam filtrelemede kullanılan bir diğer yöntem olan LR, lojistik aktivasyon fonksiyonu yardımıyla hesaplanan çıktı ile gerçek sonuçlar arasındaki hatayı en aza indirir. LR e-posta sınıflandırma problemlerine uygulanmış olup spam filtrelemede iyi bir performans sergilemiştir [11, 39, 40]. Goodman ve Yih [39] çevrimiçi "gradient descent" algoritması ile eğitilen basit bir LR sınıflandırma modeli önermişlerdir. Modellerinin NB ile karşılaştırıldığında rekabetçi sonuçlar ürettiğini göstermişlerdir. Han ve arkadaşları [11] klasik LR'nin eğitimi aşamasında öznitelik ağırlıklarının denk ayarlanmasına yönelik hassasiyetini azaltan iyileştirilmiş bir LR modeli sunmuşlardır. Chang ve arkadaşları [40] hibrit bir LR ve NB sınıflandırma modeli geliştirmişlerdir. Bu model orijinal öznitelik uzayını birkaç gruba böler ve her bir gruba LR uygular. Önerilen modelin gerçek spam filtreleme verisi üzerinde hem LR'den hem de NB'den daha üstün performans sergilediğini göstermişlerdir.

Türkçe e-postaları sınıflandırmaya (spam veya normal) yönelik çok az sayıda çalışma bulunmaktadır. Bu çalışmalardan bir tanesinde kullanılan iki teknikten biri Bayesian algoritmasıdır ve bu algoritma ile %90 sınıflandırma doğruluğuna ulaşılarak önemli bir başarı elde edilmiştir [9]. Bu çalışma iki ana aşamadan oluşmaktadır. İlk aşamada e-postalardaki tüm kelimelere morfoloji işlemi uygulanmış ve kelimelerin kökleri bulunmuştur. İkinci aşamada kök haline getirilmiş bu kelimelerden öznitelik vektörünü oluşturan kelimeler belirlenmiş ve bu öznitelik vektörü kullanılarak e-postaları sınıflandırma işlemi yapılmıştır.

Türkçe e-postaları sınıflandırmaya yönelik bir başka çalışmada n-gram tekniği kullanılmıştır ve Türkçe e-postalar için yaklaşık %97, İngilizce e-postalar için %98'in üzerinde başarılı sonuçlar elde edilmiştir [41]. Ayrıca bu çalışmada insanların bir e-postanın spam e-posta olup olmadığını anlamak için e-postanın tamamını okumadıklarını sadece küçük bir kısmını okuyarak buna karar verebildikleri fikrinden yola çıkarak ilk n-kelime sezgiseli geliştirip çalışmalarına uygulamışlardır. Böylece başarı oranından fazla taviz vermeden çalışma zamanlarında iyileşme sağlamışlardır.

Doğadan ilham alan algoritmalar sahip oldukları uyum sağlama yeteneklerinden dolayı spam filtreleme problemlerinde literatürde bilinen yöntemlerdendir. Oda ve

White [14] hayvanların sahip oldukları bağışıklık sistemleri sayesinde kendilerini korumalarından esinlenen bir bilgisayar güvenlik yaklaşımı olarak AIS algoritmasını kullanmışlardır. Önerdikleri AIS spam sistemi, lenfosit (beyaz kan hücrelerinin alt türü) üretimi, lenfositlerin uygulanması ve lenfosit değişimi olmak üzere üç aşamadan oluşmaktadır. Ölü hücreleri belirlemek için B hücrelerinin uyarlayıcı bir hafızası olarak ağırlıkları ve yaş mekanizmasını kullanmışlardır. Küçük boyutlu kütüphane uzayına rağmen gelecek vadeden sonuçlar elde etmişlerdir. Yue ve arkadaşları [42] spam e-postaları filtrelemek için artırımlı kümeleme yapay bağışıklık ağı (incremental clustering artificial immune network - ICAINet) olarak adlandırdıkları bağışıklık sisteminden esinlenen kümeleme algoritmasına dayandırdıkları davranış tabanlı işbirlikçi algoritma önermişlerdir. Algoritma spam mesajların davranış tabanlı karakteristiklerini çıkarmaktadır ve benzer spam metinlerini kümelemede bir girdi olarak vermek için spam skoru hesaplamaktadır. Guzella ve arkadaşları [43] spam e-posta mesajlarını belirlemek için doğal ve uyarlayıcı yapay bağışıklık sistemi (IA-AIS) olarak adlandırdıkları bağışıklık sisteminden esinlenen bir model sunmuşlardır. Bu model, hem doğuştan gelen hem de uyarlayıcı bağışıklık sistemlerini modelleyen B ve T lenfositlerine benzer girdileri entegre etmektedir. Parametre konfigürasyonuna bağlı olarak yüksek tespit oranlarına erişebilmektedir. IA-AIS e-postaların sınıflandırılmasında Naif Bayes sınıflandırıcıdan daha başarılıdır. Ruan ve Tan [44] spam tespiti için sistemin kendine ait olan ve olmayanları bir araya getiren vektörü oluşturan birleştirme tabanlı öznitelik vektörü oluşturma yaklaşımını kullanarak üç katmanlı geriye yayılım sinir ağını (BPNN) kullanmışlar ve BPNN, PU1 ve Ling veri kümelerine uygulamışlardır. Önerilen yöntemin deneysel olarak çalışma zamanı ve sınıflandırma doğruluğu bakımından etkili olduğu belirtilmiştir. Mohammed ve Zitar [45] genetik algoritma (GA) ile klasik AIS algoritmasını birleştiren bir spam tespit tekniği önermişlerdir. Yaptıkları çalışmada GA, standart AIS algoritmasındaki çıkarma ve yeniden inşa etme zamanlarını belirlemek için kullanılmıştır. SpamAssassin veri kümesi üzerindeki deneysel sonuçlar, standart AIS algoritması ile kıyaslandığında genetik olarak optimize edilmiş AIS algoritmasının yanlış pozitif ve yanlış negatif oranları bakımından daha iyi bir sınıflandırma performansı sergilediğini göstermiştir. Idris ve Selamat [46] literatürde bilinen AIS algoritmalarından biri olan Negatif Seçim Algoritması (NSA) ile Parçacık Sürü Optimizasyonu (Particle Swarm Optimization - PSO) algoritmasını birleştiren hibrit bir model önermişler ve

önerdikleri yöntemin spam e-postaları hem NSA'dan hem de PSO'dan daha iyi tespit edebildiğini göstermişlerdir. Ramdane ve Salim [17] k-ortalamalar kümeleme, NSA ve meyve sineği optimizasyon (Fruit Fly Optimization - FFO) algoritmasını birleştiren yeni bir hibrit model önermişler ve bu modelin hem NSA'dan hem de NSA-PSO'dan daha üstün olduğunu deneysel olarak göstermişlerdir.

1.3. Araştırmanın Amacı

Bu araştırma ile hem yerelde hem de küreselde çözümler arayabilen, yerel minimumlara takılmayan, karmaşık veya basit; lineer olan veya olmayan; Türkçe veya İngilizce; dengeli veya dengesiz; kişisel olan veya olmayan tüm verileri çok başarılı bir şekilde sınıflandırabilen optimize edilmiş sınıflandırma yöntemlerinin tasarlanması ve uygulanması amaçlanmaktadır. En yüksek başarılara ulaşabilen ve Türkçe e-postaları da dilin yapısını göz önünde bulundurarak çok başarılı bir şekilde sınıflandırabilen yöntemlerle literatüre katkı sağlamak hedeflenmektedir.

1.4. Araştırmanın Önemi

Literatürde spam e-postaların filtrelenmesine yönelik pek çok çalışma bulunmaktadır. Fakat yapılan çalışmaların neredeyse tamamı İngilizce metinleri sınıflandırmaya yöneliktir. Türkçe e-postaları sınıflandırmaya (spam veya normal) yönelik çalışmaların sayısı çok az olmakla birlikte sınıflandırma başarımları gelişime açık bir konudur. Eklemeli bir dil olan Türkçe yapısı gereği İngilizce diline oranla çok daha karmaşık bir dil olduğundan İngilizce metinleri sınıflandırmak için kullanılan yöntemler birebir uygulandığında çok daha düşük başarılar elde edilebilir [9]. Bu yüzden Türkçe metinleri sınıflandırmaya yönelik ayrıca yapılacak çalışmalara gerek duyulur. Türkçe yapısı gereği İngilizce dilinden farklıdır. İngilizce spam e-postaların filtrelenmesine yönelik çalışmalar birebir olarak Türkçe e-postalara uygulandığında çok daha düşük başarılar elde edilebilir. Türkçe eklemeli bir dil olup İngilizce'ye göre çok daha karmaşık bir dildir. Türkçe'de eklemelerle oluşturulan bir kelime İngilizce'de birden fazla kelimeyle oluşan bir cümleye karşılık gelebilir. Ya da Türkçe bir kelimeye eklemelerle çok sayıda kelime elde edilebilir (kitap, kitaplar, kitaplarım, kitaplarımda vb. kelimelerin yalın hali kitap kelimesidir). Yapısal karmaşıklığın yanı sıra başa çıkılması gereken bir diğer önemli

mesele özel Türkçe karakterlerdir (‘ç’, ‘ğ’, ‘ı’, ‘ö’, ‘ş’, ‘ü’). Çünkü tüm klavyeler özel Türkçe karakterleri desteklemez. Örneğin bir kişi “çalışmak” kelimesi yerine “calismak” yazabilir [9]. Spam e-postaların oluşturduğu ciddi sorunlar Türkçe e-postalar içinde geçerlidir. Bu yüzden bu konuda yapılacak çalışmalar önem arz etmektedir.

Bu araştırmayı önemli kılan bir diğer konu literatürde var olan çalışmaların e-posta sınıflandırma başarımlarının gelişime açık olması ve geliştirilecek yöntemlerin sınıflandırma başarımlarını yükseltmeyi hedeflemesidir. Spam e-postaları filtrelemeye yönelik çok etkili çalışmalar olmasına rağmen bu çalışmaların hesaplama karmaşıklıkları ve yanlış sınıflandırma oranları yüksek olabilmektedir. Yine bu çalışmalar, yerel minimuma takılma ve eğitim kümesindeki verileri ezberleme gibi problemlerden sorun yaşayabilmektedirler. Bu yüzden lineer olan veya olmayan; karmaşık veya basit; kişisel olan veya olmayan; Türkçe veya İngilizce; dengeli veya dengesiz tüm verileri hem yerelde hem de globalde çözümler arayıp oldukça karmaşık öznitelikleri verilerden çıkartarak çok başarılı bir şekilde sınıflandırabilen optimize edilmiş sınıflandırıcıları tasarlamak oldukça önemlidir.

Sonuç olarak Türkçe spam e-postaların tespitinin en az İngilizce spam e-postaların tespiti kadar önemli olması, Türkçe spam e-postaların tespitine yönelik çalışmaların çok az sayıda olması ve bu konuda elde edilen başarıların gelişime açık olması, Türkçe ve İngilizce dillerindeki yapısal farklılıklardan dolayı Türkçe e-postalar için ayrıca bir çalışmaya ihtiyaç duyulması, bu konuda yeni yöntemler geliştirilmesinin ve başarılarının incelenmesinin gerekliliği bu tezde yapılacak olan çalışmanın önemini arttırmaktadır.

2. BÖLÜM

YÖNTEM VE MATERYAL

2.1. Naive Bayes Yöntemleri

Makine öğrenmesinde Naive Bayes (NB) yöntemleri her bir çift özniteliğin birbirinden bağımsız olduğu gibi naif (naive) varsayımlara dayanan Bayes'in teoremini uygulamaya yönelik olasılıksal öğrenme algoritmalarıdır. Bayes'in teoremine göre, bir öznitelik vektörü $\vec{x} = [x_1, x_2, ..., x_n]$ ile temsil edilen bir mesajın c_j sınıfına ait olma olasılığı Eşitlik (2.1) kullanılarak hesaplanır.

$$p(c_j|\vec{x}) = \frac{p(c_j).p(\vec{x}|c_j)}{p(\vec{x})} \tag{2.1}$$

Eşitlik (2.1)'in paydası sabit olduğundan dolayı, NB $p(c_j).p(\vec{x}|c_j)$ değerini maksimize eden sınıfı seçer. Spam e-postaları filtreleme probleminde spam ve normal olmak üzere iki sınıf vardır. Eğer $p(c_{spam}|\vec{x})$ değeri $p(c_{normal}|\vec{x})$ değerinden büyük olursa NB, $\vec{x}$ öznitelik vektörünü spam olarak sınıflandırır. Aksi takdirde normal olarak sınıflandırır. Öncelikli olasılık $p(c_j)$, eğitim kümesindeki c_j sınıfına ait olan mesajların sayısının eğitim kümesindeki toplam mesaj sayısına bölünmesiyle elde edilir. $p(\vec{x}|c_j)$ olasılıklarının tahmini her bir NB yönteminde farklılık gösterir. Bu yöntemler aşağıda özet olarak verilmiştir.

2.1.1. Gaussian Naive Bayes

Gaussian NB yönteminde her bir özniteliğin normal (Gaussian) dağılıma göre dağıldığı varsayılır ve $p(\vec{x}|c_j)$ Eşitlik (2.2)'deki gibi hesaplanır.

$$p(\vec{x}|c_j) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_{i,j}^2}} e^{-\frac{(x_i-\mu_{i,j})^2}{2\sigma_{i,j}^2}} \tag{2.2}$$

Parametreler $\mu_{i,j}$ (ortalama) ve $\sigma_{i,j}$ (standard sapma) maksimum olabilirlik yöntemi kullanılarak hesaplanılır.

2.1.2. Multinomial Naive Bayes

Multinominal NB yönteminde bir e-posta, öznitelik vektörü $\vec{x} = [x_1, x_2, ..., x_n]$ ile temsil edilir ve bu öznitelik vektöründe her bir x_i kendisine karşılık gelen t_i teriminin bu e-postadaki görülme sayısıdır. $p(\vec{x}|c_j)$ olasılıkları Eşitlik (2.3) kullanılarak tahmin edilir.

$$p(\vec{x}|c_j) = \prod_{i=1}^{n} p(t_i|c_j), \tag{2.3}$$

$p(t_i|c_j)$ olasılığının değeri Eşitlik (2.4) kullanılarak elde edilir.

$$p(t_i|c_j) = \frac{N_{t_i,c_j} + \alpha}{N_{c_j} + \alpha n}, \tag{2.4}$$

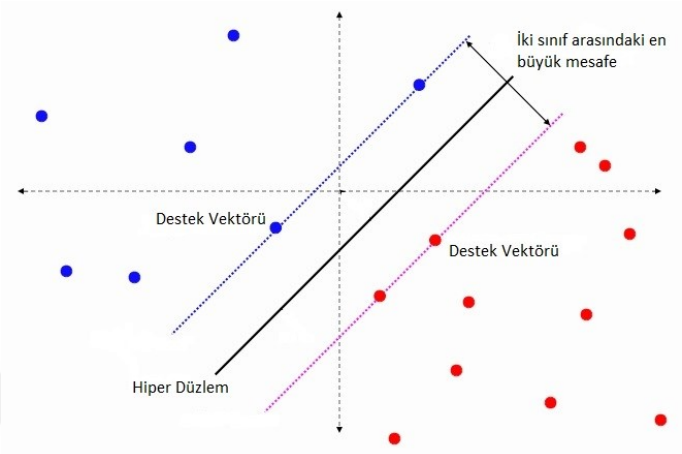
Eşitlik (2.4)'de N_{t_i,c_j} değeri eğitim kümesindeki c_j sınıfına ait mesajlarda t_i teriminin görülme sayısının toplamına eşitken N_{c_j} değeri c_j sınıfına ait mesajlardaki tüm terimlerin görülme sayılarının toplamına eşittir.

2.2. Destek Vektör Makineleri: SVM

Sınıflandırma problemlerinde sıklıkla kullanılan yöntemlerden bir tanesi Destek Vektör Makinesidir (Support Vector Machine - SVM). Sınıflandırma probleminde tam olarak iki sınıf varsa SVM doğrudan kullanılabilir. Eğer sınıf sayısı ikiden fazla ise bir sınıfa karşı diğer sınıflar (one versus all) yöntemi ile tüm sınıflar birbirleri arasında SVM ile ikişerli olarak sınıflandırılabilir.

SVM, bir sınıftaki tüm verileri diğer sınıftaki tüm verilerden ayıran en iyi hiper düzlemi bulmaya çalışarak verileri sınıflandırır. İki sınıfı birbirinden ayıran en iyi hiper düzlem, iki sınıfın destek vektörleri arasındaki en büyük mesafeyi (marjin) bulması anlamına gelir.

Şekil 2.1'de iki sınıfı birbirinden lineer olarak ayıran SVM örneği verilmiştir. Bu örnekten de görüldüğü üzere SVM'in amacı karar hiper düzlemine en yakın olan pozitif ve negatif veri noktaları arasındaki mesafeyi maksimize etmektir. Bu noktalar destek vektörleri olarak adlandırılırlar.



Şekil 2.1. Sınıfları birbirinden lineer olarak ayıran Destek Vektör Makinesi örneği

Bir eğitim kümesi verildikten sonra $\{(\vec{x}_1, y_1), ..., (\vec{x}_m, y_m)\}$, $\vec{x}_i \in R^n$ ve $y_i \in \{-1, 1\}$, $1 \leq i \leq m$, SVM Eşitlik (2.5)'de görülen optimizasyon problemini çözerek en iyi karar hiper düzlemini veren ağırlık vektörü $\vec{w}$ ve bias terimi b'yi belirler.

$$\begin{aligned} \min \quad & \frac{1}{2}\vec{w}^T\vec{w} \\ \text{subject to} \quad & y_i(\vec{w}^T\vec{x}_i + b) \geq 1 \end{aligned} \quad . \tag{2.5}$$

Yeni veri noktaları $\vec{x}_j$ Eşitlik (2.6) kullanılarak sınıflandırma yapılır.

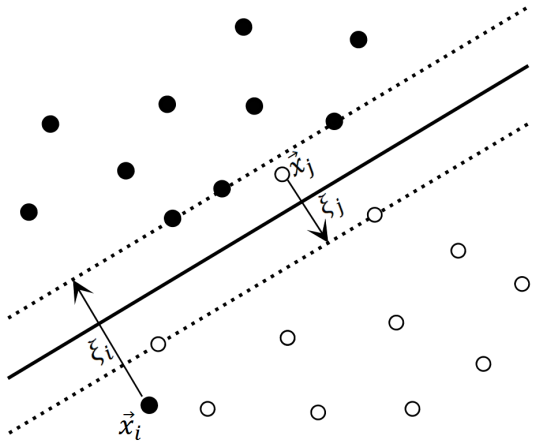
$$f(\vec{x}_j) = sign(\vec{w}^T\vec{x}_j + b). \tag{2.6}$$

2.2.1. Lineer SVM

Lineer bir SVM sınıflandırıcıda eğitim kümesi lineer bir şekilde ayrılabilir değilse SVM esnek bir marjin kullanır. Böyle durumlarda Şekil 2.2'de görüldüğü gibi yanlış sınıfa yerleştirilebilen bazı veri noktaları haricinde karar hiper-düzlemi eğitim kümesindeki veri noktalarını doğru bir şekilde sınıflandırır. SVM bu tür bir hiper-düzlemi bulmak için

bir ceza parametresi (C) ve yanlış sınıflandırılan örneklerin maliyetine karşılık gelen bir esneklik değişkeni (ξ_i) kullanır. Örnek doğru sınıfa atanırsa esneklik değişkeni sıfıra eşit olur. Esnek marjin SVM sınıflandırıcının formülasyonu Eşitlik (2.7)'de verilmektedir.

$$\begin{aligned} \min \quad & \frac{1}{2}\vec{w}^T\vec{w} + C\sum_{i=1}^{m}\xi_i \\ \text{subject to} \quad & y_i(\vec{w}^T\vec{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0. \end{aligned} \tag{2.7}$$



Şekil 2.2. Lineer olarak ayrılamayan bir durumda Lineer SVM sınıflandırıcı örneği

2.2.2. Radial Basis Fonksiyonlu SVM (RBF-SVM)

Veri kümesi lineer olmayan etkileşimlere sahip olduğunda ve öznitelik uzayındaki öznitelikler lineer olmayan bağımlılığa sahip olduğunda lineer ayırıcılar veya lineer olmayan karar sınırları kullanmak başarısız sınıflandırmaya neden olabilir. Bu durumda SVM sınıflandırıcılar veri kümesindeki lineer olmayan durumlara uyum sağlamak için lineer olmayan çekirdek (kernel) fonksiyonlarını (Eşitlik (2.8)) kullanır.

$$f(x) = sign\left\{\sum_{i=1}^{N}(w_i y_i K(x_i \cdot x_j) + b)\right\} \tag{2.8}$$

Bu çalışmada çekirdek fonksiyonu olarak Eşitlik (2.9)'da verilen RBF kullanılmıştır.

$$K_{RBF}(x_i, x_j) = exp(-\gamma||x_i - x_j||^2), \gamma > 0. \tag{2.9}$$

2.3. Lojistik Regresyon

Bir eğitim kümesi verildikten sonra $\{(\vec{x}_1, y_1), ..., (\vec{x}_m, y_m)\}$, $\vec{x}_i \in R^n$ ve $y_i \in \{0, 1\}, 1 \leq i \leq m$, LR bir $\vec{x}_i$ vektörünün sınıfını Eşitlik (2.10)'u kullanarak belirler.

$$y_i^{'} = \begin{cases} 0, & p_i < 0.5 \\ 1, & p_i \geq 0.5 \end{cases} \tag{2.10}$$

Eşitlik (2.10)'da yer alan p_i değeri Eşitlik (2.11) kullanılarak hesaplanır ve Eşitlik (2.12)'de verilen fonksiyon (σ) sigmoid fonksiyonu olarak adlandırılır. Sigmoid fonksiyonu eksi sonsuz ile artı sonsuz arasındaki tüm değerleri 0 ile 1 arasındaki değerlere dönüştürmek için kullanılır.

$$p_i = \sigma(\vec{w}\vec{x}_i), \tag{2.11}$$

$$\sigma(a) = \frac{1}{1 + e^{-a}}. \tag{2.12}$$

LR algoritmasının amacı maliyet fonksiyonunun verdiği değeri minimize eden ağırlıkları $(\vec{w})$ öğrenmektir. Çalışmalarımızda "cross-entropy" maliyet fonksiyonu ile "ridge" regresyon adı verilen yöntem kullanılmıştır. Ridge regresyon, modelin karmaşıklığını azaltır ve eğitim kümesinin ezberlenmesini (overfitting) önler. Cross-entropy maliyet fonksiyonu ile ridge regresyon yöntemi Eşitlik (2.13)'de verilmektedir.

$$J(\vec{w}) = -\sum_{i=1}^{m} y_i \log(p_i) + (1 - y_i) \log(1 - p_i) + \frac{\lambda}{2} \sum_{j=1}^{n} w_j^2, \tag{2.13}$$

Eşitlik (2.13)'de λ, regülisazyon parametresidir. LR'de ağırlıkların $(\vec{w})$ değerleri öğrenilirken "gradient descent" algoritması kullanılmaktadır. Gradient Descent algoritması başlangıç $\vec{w}$ değerleri ile başlar ve sonlandırma kriteri sağlanıncaya kadar Eşitlik (2.14)'de verilen güncellemeyi tekrarlar.

$$\vec{w} := \vec{w} + \alpha X^T(\vec{y} - \sigma(X\vec{w})) - \lambda\vec{w}, \tag{2.14}$$

Eşitlik (2.14)'de Gradient Descent algoritmasının vektör haline getirilmiş versiyonu görülmektedir ve bu eşitlikte yer alan α, öğrenme modelinin ne kadar hızlı bir şekilde yerel minimuma yakınsayacağını belirleyen öğrenme katsayısıdır. Eğer α çok küçük seçilirse yavaş yakınsama gerçekleşirken α çok büyük seçilirse yerel minimum noktası kaçırılabilir. Eşitlikteki α ve λ değerleri LR modelleri için önemli hiper-parametrelerdir. Eşitlikte görülen X ise $\{\vec{x}_1, \vec{x}_2, ..., \vec{x}_m\}$ vektörlerinden oluşan $m \times n$ boyutlu bir matrisi temsil etmektedir.

Tutun ve arkadaşları [47] LR'de yer alan ağırlıkları optimize etmek için evrimsel strateji ve simulated annealing algoritmasını kullanmışlardır. Önerdikleri online model erişime açık "diabetes readmission" veri kümesi üzerinde test edilmiştir ve sonuçlar önerilen yöntemin SVM, NB, LR ve karar ağaçlarından (decision trees) daha üstün olduğunu göstermiştir.

2.4. Optimizasyon algoritmaları ile eğitilen LR sınıflandırıcılar

Standart LR sınıflandırma yönteminde maliyet fonksiyonunun verdiği değeri en aza indirmek için gradient descent algoritması kullanılmaktadır. Bu algoritmanın bir çeşit optimizasyon yöntemi gibi davranması diğer optimizasyon algoritmalarını bu algoritmanın yerine kullanma fikrini doğurmuştur. Burada amaç LR sınıflandırma modelinde yer alan ağırlıkları maliyet fonksiyonun değerini minimize edecek şekilde hesaplamaktır. Optimizasyon algoritmaları ise bir fonksiyonun değerini minimum veya maksimum yapmak için gerekli olan parametreleri aramada sıklıkla ve başarıyla uygulanmaktadır. Ayrıca gradient descent algoritmasını kullanan standart LR sınıflandırma modelinin yerel minimuma takılabilmesi, lineer olmayan ve karmaşık/zor problemlerde başarısız olabilmesi gibi dezavantajlarından dolayı gradient descent algoritmasının yerine optimizasyon algoritmalarını kullanma fikri cezbedici olabilmektedir. Bu yüzden literatürde bilinen ve sıklıkla kullanılan optimizasyon algoritmaları olan Parçacık Sürü Optimizasyonu (Particle Swarm Optimization - PSO) ve Diferansiyel Gelişim (Differential Evolution - DE) algoritmaları bu tez çalışmasında önerilen yöntemlerle kıyaslama yapılması için seçilmiştir.

2.4.1. Parçacık Sürü Optimizasyonu algoritması ile eğitilen LR sınıflandırıcı: PSO-LR

Parçacık Sürü Optimizasyonu (PSO), sürü halinde hareket eden kuş veya balık gibi canlı topluluklarının yiyecek arama davranışlarından esinlenilerek Eberhart ve Kennedy tarafından geliştirilen ve sürü zekasına dayanan meta-sezgisel bir optimizasyon algoritmasıdır [48]. Sürüde (swarm) yer alan her bir bireye parçacık (particle) denir. Bir parçacığın pozisyonu optimize edilecek fonksiyonun olası bir çözümüne karşılık gelir. Her bir iterasyonda bir parçacığın pozisyonu, o parçacığın şimdiye kadar elde ettiği en iyi çözüm ($\vec{p}(i)$, kişisel en iyi çözüm) ile tüm sürüde şimdiye kadar elde edilen en iyi çözüm ($\vec{p}(g)$), küresel en iyi çözüm) kullanılarak güncellenir. Parçacıklar topluluğa yayılan bilgi sayesinde arama uzayında iyi çözümlerin yer aldığı bölgelere hareket etme eğilimindedirler. PSO algoritması aşağıdaki gibi özetlenebilir.

Algoritma 2.1 PSO algoritması

1: Sürüde yer alan her bir parçacığın başlangıç pozisyonlarını rastgele bir şekilde oluştur.

2: Sürüdeki parçacıkların uygunluk değerini hesapla.

3: Her bir parçacık için o parçacığa ait en iyi çözümü bul ve sakla ($\vec{p}(i)$, kişisel en iyi çözüm).

4: Tüm parçacıklar içerisindeki kişisel en iyi çözümler içerisinden küresel en iyi çözümü bul ve küresel en iyi çözüm olarak ata ($\vec{p}(g)$, küresel en iyi çözüm).

5: Her bir parçacığın pozisyonunu Eşitlik (2.15) kullanarak güncelle.

$$\vec{x}(t+1) = \vec{x}(t) + \vec{v}(t+1) \tag{2.15}$$

Burada $\vec{x}(t+1)$ parçacığın yeni pozisyonu iken $\vec{v}(t+1)$ parçacığa ait yeni hız değerini göstermektedir. Yeni hız değerini Eşitlik (2.16) kullanarak hesapla.

$$\vec{v}(t+1) = w\vec{v}(t) + c_1 rand(0,1)(\vec{p}(t) - \vec{x}(t) + c_2 rand(0,1)(\vec{g}(t) - \vec{x}(t))) \tag{2.16}$$

6: Madde 2,3,4 ve 5'i durdurma kriteri sağlanıncaya kadar tekrarla.

2.4.2. Diferansiyel Gelişim algoritması ile eğitilen LR sınıflandırıcı: DE-LR

Storn ve Price [49] tarafından önerilen Diferansiyel Gelişim (Differential Evolution - DE) algoritması çaprazlama, mutasyon ve seçim operatörlerini kullanan popülasyon tabanlı bir optimizasyon tekniğidir.

Algoritma bir arama mekanizması olarak mutasyon işlemini kullanırken arama uzayındaki muhtemel bölgelere aramayı yönlendirmek için seçim işlemini kullanır. Çaprazlama operatörü, çocuk vektörleri oluşturmak için mevcut popülasyon üyelerinin bileşenlerini kullanarak daha iyi bir çözüm uzayını aramayı mümkün kılan başarılı kombinasyonlar hakkındaki bilgiyi etkili bir şekilde karıştırır. DE algoritmasında P adet çözüm vektöründen oluşan bir popülasyon başlangıçta rastgele bir şekilde oluşturulur. Bu popülasyon sonlandırma kriteri sağlanıncaya kadar mutasyon, çaprazlama ve seçim operatörleri uygulanarak her bir iterasyonda geliştirilir. DE algoritmasının ana adımları Algoritma 2.2'de verilmektedir.

Algoritma 2.2 DE algoritması

1: Kontrol parametrelerinin değerlerini ata: popülasyon sayısı (P), mutasyon faktörü (F) ve çaprazlama olasılığı (CR)
2: P adet popülasyonu rastgele oluştur
3: Uygunluk değerlerini hesapla (değerlendirme)
4: **Tekrarla** {Sonlandırma kriteri sağlanıncaya kadar}
5: Mutasyon
6: Çaprazlama
7: Değerlendirme
8: Seçim

Mutasyon aşamasında popülasyon içerisinden rastgele üç vektör seçilir ve mutasyona uğramış vektör Eşitlik (2.17)'de gösterildiği gibi üretilir.

$$\vec{m}_i = \vec{x}_{r_1} + F(\vec{x}_{r_2} - \vec{x}_{r_3}) \tag{2.17}$$

Eşitlik (2.17)'de yer alan F değeri [0,1] aralığından seçilen mutasyon faktörünü ifade ederken $\vec{x}_{r_1}$, $\vec{x}_{r_2}$ ve $\vec{x}_{r_3}$ vektörleri $r_1 \neq r_2$, $r_1 \neq r_3$ ve $r_2 \neq r_3$ olması şartıyla popülasyon içerisinden rastgele seçilen çözüm vektörlerini ifade etmektedir. Son olarak i indeksi ise mevcut çözümün indeksini temsil etmektedir.

Çaprazlama aşamasında, çocuk vektörü elde etmek için ebeveyn vektör ile mutasyonlu vektör önceden tanımlanmış kontrol parametresi olan çaprazlama olasılığına (CR) bağlı olarak Eşitlik (2.18)'de görüldüğü gibi karıştırılır.

$$\vec{c}_i^{j} = \begin{cases} \vec{m}_i^{j}, & R_j \leq CR \\ \vec{x}_i^{j}, & R_j > CR \end{cases} \tag{2.18}$$

Eşitlik (2.18)'de R_j, [0,1] aralığında rastgele bir şekilde üretilen gerçek bir sayıyı temsil ederken j değişkeni, bağlı olduğu vektörün j. parametresini ifade etmektedir.

Popülasyondaki tüm çözümler uygunluk değerine bağlı olmaksızın ebeveynler olarak aynı şeçilme şansına sahiptir. Mutasyon ve çaprazlama işlemlerinden sonra üretilen çoçuk vektör değerlendirilir ve uygunluk değeri hesaplanır. Seçim aşamasında çocuk vektörle onun ebeveyninin uygunluk değeri karşılaştırılır ve daha yüksek uygunluk değerine sahip olan bir sonraki popülasyona aktarılır. Yani ebeveynin uygunluk değeri daha iyiyse popülasyonda kalır aksi takdirde çoçuk vektör ebeveyn yerine popülasyona dahil edilir. Mutasyon, çaprazlama ve seçim aşamaları sonlandırma kriteri sağlanıncaya kadar tekrar edilir.

3. BÖLÜM

ÖNERİLEN YÖNTEMLER VE BİLEŞENLERİ

3.1. Yapay Arı Kolonisi Algoritması

Karaboga [10] tarafından önerilen Yapay Arı Kolonisi (ABC) algoritması zeki optimizasyon algoritmalarından biri olup bal arısı topluluğunun yiyecek arama davranışlarını modeller. ABC algoritmasında işçi arı, gözcü arı ve kaşif arı olmak üzere üç tip yapay arı bulunmaktadır. Yiyecek kaynağının yeri olası bir çözüme karşılık gelmektedir ve yiyecek kaynağının nektar miktarı çözümün kalitesini göstermektedir. Algoritmanın amacı maksimum nektar miktarına sahip olan yiyecek kaynağını bulmaktır. ABC algoritması arıların yiyecek arama davranışlarını modellerken bazı kabullerde bulunmaktadır:

- Her kaynağın nektarı sadece bir görevli arı tarafından alınır.
- İşçi arıların sayısı yiyecek kaynağı sayısına eşittir.
- İşçi arıların sayısı, gözcü arıların sayısına eşittir.
- Nektarı tükenmiş kaynağın görevli arısı, kaşif arıya dönüşür.

Algoritmanın ana adımları Algoritma 3.1'de verilmektedir. İşçi arı aşamasında işçi arılar, hafızalarındaki yiyecek kaynağı pozisyonlarının çevresinde daha iyi bölgeler keşfetme ve dans bölgesinde bekleyen gözcü arılarla yiyecek kaynaklarının nektar miktarını ve pozisyon bilgisini paylaşma işlerinden sorumludur. Her bir gözcü arı işçi arılar tarafından gerçekleştirilen dansı izleyerek uçacağı bir yiyecek kaynağını belirler. ABC algoritması rulet tekerleği seçimi (roulette wheel selection) gibi stokastik bir seçim mekanizması ile dansı ve faydalı kaynak seçimini taklit eder. Stokastik mekanizma yiyecek kaynağının

nektar miktarı arttıkça daha fazla sayıda gözcü arının o kaynağı seçme ihtimalini artıran pozitif bir geri besleme verir.

Algoritma 3.1 ABC algoritması

1: Kontrol parametrelerine değerlerin atanması: yiyecek kaynağı sayısı (SN), maksimum döngü saysı (MCN) ve $limit$ değeri

2: Başlangıç yiyecek kaynağı popülasyonunu Eşitlik (3.1)'i kullanarak üret ve çözümleri değerlendir

$$x_{ij} = x_j^{min} + rand(0,1) \times (x_j^{max} - x_j^{min}) \tag{3.1}$$

Eşitlik (3.1)'de x_{ij}, $i.$ çözümün $j.$ parametresidir, $i \in \{1, 2, ..., SN\}$ ve $j \in \{1, 2, ..., D\}$, SN ve D sırasıyla yiyecek kaynağı ve optimizasyon parametrelerinin sayısını temsil etmektedir. $rand(0, 1)$, 0 ile 1 aralığında rasgele üretilen bir sayıdır. x_j^{max} ve x_j^{min} ise $j.$ parametrenin alt ve üst sınırlarıdır.

3: İşçi arı aşaması her bir çözüm üzerinde Eşitlik (3.2)'i kullanarak yerel bir arama yapar ve açgözlü bir seçim gerçekleştirir.

$$v_{ij} = x_{ij} + \varphi_{ij} \times (x_{ij} - x_{kj}), \tag{3.2}$$

φ_{ij}, -1 ile $+1$ aralığında rastgele bir sayıdır. $j \in \{1, 2, ..., D\}$ ve $k \in \{1, 2, ..., SN\}$, i ve k değerleri eşit olmamak şartıyla rastgele seçilen indekslerdir.

4: Her bir çözüme Eşitlik (3.3) kullanılarak kalitesiyle orantılı olasılıksal bir değer atanır.

$$p_i = \left(0.9 \times \frac{f_i}{max(\vec{f})}\right) + 0.1, \tag{3.3}$$

f_i, $i.$ çözümün uygunluk değerini ifade ederken $max(\vec{f})$, maksimum uygunluk değerini ifade eder.

5: Gözcü arılar olasılıksal değerler ile doğru orantılı olarak çözümleri seçerler. Her bir seçilen çözümün etrafında Eşitlik (3.2) kullanılarak lokal arama gerçekleştirilir.

6: En iyi çözümü hatırla

7: Kaşif arılar eğer tükenmiş bir yiyecek kaynağı varsa Eşitlik (3.1)'i kullanarak yeni bir çözüm üretirler

8: Sonlandırma kriteri sağlanıncaya kadar 3, 4, 5, 6 ve 7 maddelerini tekrarla

Algoritma 3.1'de 3, 7 arasındaki adımlarda açgözlü seçim gerçekleşir. Açgözlü seçime göre yeni çözümün nektar miktarı mevcut çözümden daha yüksekse arı yeni çözümü hafızasına kaydeder ve eski çözümü unutur. Aksi takdirde mevcut çözümü hafızasında tutar ve mevcut çözümle ilgili sayacı 1 arttırır. Sayaçlar her bir kaynağın etrafındaki faydalanmaların sayısını tutar ve bir yiyecek kaynağının tükenip tükenmediğine karar vermek için kaşif arı aşamasında kullanılırlar. Bir çözüme karşılık gelen sayaç değeri "$limit$" adı verilen ve algoritmanın başında değeri atanmış sayıyı aşarsa o tükenmiş bir çözüm olur. Temel algoritmanın her bir döngüsünde en fazla bir işçi arı kaşif arıya dönüşebilir. Yiyecek kaynağı tükenmiş birden fazla işçi arı varsa algoritma en yüksek değere sahip sayaca karşılık gelen kaynağı seçer ve o kaynağın işçi arısı kaşif arıya dönüştürülür. Tükenmiş kaynaklar arılar tarafından terk edilirler ve kaşif arılar terk edilen kaynaklarla keşfedilmemiş kaynakları değiştirmek için arama yaparlar.

3.2. ABC algoritması ile eğitilen LR sınıflandırıcı: ABC-LR

LR'de kullanılan gradient descent algoritması türev hesaplamaları gerektirmesi ve maliyet fonksiyonunun sürekli olması gibi bazı sınırlamalara sahip olduğundan dolayı fonksiyon ve parametre arama uzayı hakkında herhangi bir varsayımda bulunmayan ve başarılı bir sezgisel olan ABC algoritması LR sınıflandırma modelini eğitmek için kullanılabilir. LR'nin ABC ile eğitilmesinde LR modelindeki optimize edilecek ağırlıklar, ABC algoritmasındaki yiyecek kaynağı pozisyonlarına karşılık gelir. İlk olarak ağırlıkların ve bias değişkeninin başlangıç değerleri algoritma tarafından üretilir. Daha sonra işçi arı, gözcü arı ve kaşif arı aşamaları gerçek çıktı ile sınıflandırıcının verdiği çıktı arasındaki farkın karelerinin toplamını en az indiren optimum ağırlık kümesini ($\vec{w}_i$) ve bias değerini bulmaya çalışır.

ABC algoritmasında bir yiyecek kaynağının pozisyonunu gösteren ağırlıklara ve bias değerine karşılık gelen vektör Eşitlik (3.4) kullanılarak oluşturulur.

$$w_{ij} = w_j^{min} + rand(0,1) \times (w_j^{max} - w_j^{min}), \tag{3.4}$$

Eşitlikte $\vec{w}_i$, $i.$ yiyecek kaynağı pozisyonunu ($i = 1...SN$) temsil ederken SN yiyecek kaynağı sayısını temsil eder. $rand(0,1)$ fonksiyonu 0 ile 1 aralığında homojen dağılmış rastgele bir sayı üretirken w_j^{max} ve w_j^{min} $j.$ parametre için ($j = 1...n$) sırasıyla üst ve

alt sınırları ifade etmektedir ve son olarak n ise optimize edilecek parametre sayısını göstermektedir.

Her bir ağırlık vektör kümesi LR modeline uygulanır ve çözüm vektöründeki ağırlıklar ve bias değeri kullanılarak bir çıktı hesaplanır. Eşitlik (3.5) ile ifade edilen bir çözümün uygunluk (fitness) değeri, elde edilmek istenen çıktı ile LR modelin verdiği çıktı arasındaki hata Eşitlik (3.6) kullanılarak hesaplanır. Uygunluk fonksiyonu ve seçim operatörü, arama uzayındaki daha iyi bölgeleri bulabilmek için algoritmaya rehberlik eder.

$$f_i = \frac{1}{1 + E_i} \tag{3.5}$$

$$E_i = \sum_{j=1}^{m} (p_j^i - y_j^i)^2 \tag{3.6}$$

Başlangıç popülasyonu değerlendirilir değerlendirilmez algoritma sonlandırma kriteri sağlanıncaya kadar işçi arı, gözcü arı ve kaşif arı aşamalarını tekrarlar. İşçi arı aşamasında her bir işçi arının hafızasındaki çözümün çevresinde bölgesel arama gerçekleştirilir ve Eşitlik (3.7) kullanılarak yeni bir çözüm elde edilir.

$$v_{ij} = w_{ij} + \varphi_{ij} \times (w_{ij} - w_{kj}), \tag{3.7}$$

Eşitlikte φ_{ij}, $[-1, 1]$ aralığında homojen dağılmış rastgele bir sayıyı temsil ederken j, $[1, n]$ aralığında rastgele bir boyut indeksidir ve son olarak k, $i \neq k$ olması şartıyla rastgele seçilen komşu çözümün indeksidir.

ABC algoritmasının Eşitlik (3.7) ile verilen standart versiyonunda yeni bir çözüm üretmek için sadece bir tane rastgele seçilen j parametresi değiştirilir. Bu çalışmada yakınsama oranını ve hızını iyileştirmek için değiştirilmiş bir bölgesel arama operatörü kullanılmıştır. Eşitlik (3.8) ile verilen bu operatöre göre her bir parametre, homojen dağılmış $[0, 1]$ aralığı içerisinden seçilen rastgele bir sayı önceden tanımlanmış kontrol parametresi olan MR değerinden küçükse değiştirilir.

$$v_{ij} = \begin{cases} w_{ij} + \varphi_{ij} \times (w_{ij} - w_{kj}), & R_{ij} < MR \\ w_{ij}, & \text{aksi takdirde} \end{cases} \tag{3.8}$$

Eşitlikte R_{ij}, homojen dağılmış $[0, 1]$ aralığı içerisinden seçilen rastgele bir sayıdır. Yeni bir çözüm üretilir üretilmez Eşitlik (3.9) kullanılarak parametre sınırlarının ihlal edilip edilmeği belirlenir ve ihlal varsa düzeltilir.

$$v_{ij} = \begin{cases} w_j^{max}, & v_{ij} > w_j^{max} \\ w_j^{min}, & v_{ij} < w_j^{min} \\ v_{ij} \quad , & \text{aksi takdirde} \end{cases} \tag{3.9}$$

Yeni çözüm uygunluk fonksiyonunda ölçülür ve hâlihazırdaki çözüm ile yönü çözüm arasında uygunluk değerine göre açgözlü bir seçim gerçekleştirilir.

Her bir çözüme kalitesine veya uygunluk değerine orantılı olacak şekilde Eşitlik (3.3) kullanılarak bir olasılık değeri atanır. Bir gözcü arı olasılık değerine bağlı olarak bir yiyecek kaynağı seçer ve her bir işçi arının yaptığı gibi Eşitlik (3.8)'i kullanan bir lokal arama ile yeni bir çözüm üretir. Daha sonra mevcut çözüm ile yeni çözüm arasında açgözlü seçim gerçekleştirilir.

Standart ABC algoritmasında olduğu gibi tükenmiş bir yiyecek kaynağı varsa kaşif arı aşaması o kaynağı terk eder ve yeni bir yiyecek kaynağı pozisyonunu (çözüm) Eşitlik (3.4)'ü kullanarak oluşturur.

3.3. CSA algoritması ile eğitilen LR sınıflandırıcı: CSA-LR

De Castro ve Von Zuben [16] tarafından önerilen klonal seçim algoritması, antijenik bir uyarana karşı uyarlanabilir bir bağışıklık sisteminin cevabını simüle eder. Bağışıklık cevabının klonal seçim (clonal selection) ve benzerlik olgunlaşması (affinity maturation) süreçlerine dayanan CLONALG adını verdikleri bir algoritma geliştirdiler. Örüntü tanıma ve optimizasyon problemlerini çözmek için klonal seçim algoritmasının iki farklı versiyonu önerilmiştir [16]. Bu çalışmada CSA'nın implementasyonunu yapmak için, optimizasyon problemlerini çözme amacıyla kullanılan uyarlanmış CLONALG sürümünden faydalanılmıştır. CSA'nın amacı maksimum benzerliğe sahip olan bir antikor

bulmaktır [16]. LR sınıflandırma modelinin eğitimi aşamasında, antikorlar ağırlıklara karşılık gelir ve CSA, Eşitlik (2.13)'de verilen maliyet fonksiyonunu en aza indirerek veya Eşitlik (3.11)'de verilen uygunluk değerini maksimize ederek optimal ağırlıkları öğrenir. Optimal ağırlıkları öğrenildikten sonra LR sınıflandırma modeli, Eşitlik (2.10), (2.11) ve (2.12)'i kullanarak bir e-postaya spam veya normal olarak bir sınıf atamak için kullanılır.

P adet antikor popülasyonu $Ab = \{Ab_1, Ab_2, ..., Ab_P\}$ Eşitlik (3.10) kullanılarak rastgele bir şekilde üretilir ve sonlandırma kriteri karşılanıncaya kadar seçim, klonlama, hiper-mutasyon, yeniden seçim ve reseptör düzenleme işlemleri her bir iterasyonda uygulanılarak bu popülasyon iyileştirilir [16]. Bu populasyonda ki her bir antikor $Ab_i = [Ab_{i,1}, Ab_{i,2}, ..., Ab_{i,D}] \in R^D$ olası bir çözüme karşılık gelir ve amaç tüm iterasyonlar tamamlandıktan sonra en yüksek benzerlik değerine sahip olan bir antikor bulmaktır.

$$Ab_{i,j} = lb_j + rand(0,1) \times (ub_j - lb_j) \tag{3.10}$$

Eşitlik (3.10)'da $rand(0,1)$, 0 ile 1 arasında düzgün dağıtılmış rasgele sayı üreten bir fonksiyondur ve lb_j değeri j. parametre için alt sınırı belirtirken ub_j değeri üst sınırı belirtir. P adet antikor popülasyonu üretildikten sonra her bir antikor Ab_i'nin uygunluk değeri Eşitlik (3.11)'de verildiği gibi hesaplanabilmektedir.

$$f(Ab_i) = \frac{1}{1 + J(Ab_i)} \tag{3.11}$$

$J(Ab_i)$ Eşitlik (2.13)'de verilen ve Ab_i'nin maliyet değerini dönderen maliyet fonksiyonuyken $f(Ab_i)$ ise Ab_i'nin uygunluk değerini dönderen uygunluk fonksiyonudur. Her bir seçilen antikor Ab_i için üretilen klon sayısı (α_i) aynı olabilir [16] ve klon popülasyonu C'deki toplam klon sayısı Eşitlik (3.12) kullanılarak hesaplanır.

$$|C| = \sum_{i=1}^{n} \alpha_i = \alpha \times n \tag{3.12}$$

Eşitlikteki α, pozitif bir tamsayı olan klonal çarpım faktörüdür. Bu çalışmada popülasyondaki tüm antikorlar klonlama için seçilir ($n = P$) ve birden fazla yerel optimum bulma amacıyla [16] her bir seçilen antikor için klon sayısı aynıdır ($\alpha_i = \alpha$).

Algoritma 3.2 CSA algoritması

Kontrol parametrelerine değerlerin atanması: antikor sayısı (P), maksimum iterasyon sayısı, B ve α
P adet antikordan oluşan başlangıç popülasyonunu Eşitlik (3.10)'u kullanarak oluştur.
Her bir antikor Ab için benzerlik (affinity) değerini hesapla
for herbir iterasyon **do**
    Antikorları klonla ve bu klonların benzerlik değerlerini hesapla
    for herbir klon C_i **do**
        C_i üzerinde ters mutasyon gerçekleştir ve mutasyonlu klonu (σ_i) elde et
        if $f(\sigma_i) > f(C_i)$ **then**
            $C_i := \sigma_i$
        else
            C_i üzerinde ikili mutasyon gerçekleştir ve mutasyonlu klonu (σ_i) elde et
            σ_i'nin benzerlik değerini hesapla
            if $f(\sigma_i) > f(C_i)$ **then**
                $C_i := \sigma_i$
            else
                $C_i := C_i$
            end if
        end if
    end for
    for herbir antikor Ab_i **do**
        Ab_i'nin klonları arasından en yüksek benzerlik değerine sahip olanı (C_j) bul
        $Ab_i := C_j$
    end for
    En kötü $\%B$ adet benzerlik değerine sahip antikorların yerine Eşitlik (3.10)'u kullanarak yenilerini üret.
end for

Klon popülasyonunu oluşturduktan sonra hiper mutasyon aşamaları olan ters mutasyon (inverse mutation) ve ikili mutasyon (pair-wise mutation) işlemleri kullanılarak antikorlar iyileştirilir. Ters mutasyon işleminde her bir klonun ($C_i = [C_1, C_2, ..., C_D]$) j ile l parametreleri $|j - l| > 2$ olması şartıyla rastgele bir şekilde seçilir ve C_i'nin j ile l

arasındaki parametreleri tersine çevrilir ve mutasyona uğramış klon olan σ_i elde edilir. Eğer σ_i'nin uygunluk değeri C_i'nin uygunluk değerinden büyükse σ_i, C_i'ye atanır. Aksi taktirde algoritma C_i üzerinde ikili mutasyon uygular. İkili mutasyonda C_i'nin j ile l parametreleri rastgele bir şekilde seçilir ve birbiriyle takas edilir. İkili mutasyon kullanılarak elde edilen mutasyona uğramış klon σ_i'nin uygunluğu değerlendirilir. C_i'nin uygunluk değeri σ_i'nin uygunluk değerinden küçükse C_i, σ_i ile değiştirilir. Aksi taktirde C_i değiştirilmez.

Hiper mutasyon sürecinden sonra antikor popülasyonun sayısını aynı tutmak için yeniden seçim işlemi gerçekleştirilir. Her bir antikor Ab_i için en yüksek uygunluk değerine sahip olan klon Ab_i'nin klonları arasından seçilir ve Ab_i'ye atanır [25]. Son olarak reseptör düzenleme işlemi gerçekleştirilir ve en kötü uygunluk değerine sahip olan $\%B$ adet antikor, Eşitlik (3.10) kullanılarak yeni üretilen antikorlarla değiştirilir. CSA optimizasyon algoritmasının adımları Algoritma 3.2'de verilmiştir.

3.4. ABC algoritması ile kombine edilen CSA algoritması tarafından eğitilen LR sınıflandırıcı: ABC-CSA-LR

Optimizasyon problemleri üzerinde CSA kayda değer bir başarı göstermesine rağmen yerel arama yeteneği, yakınsama oranı ve yakınsama hızı yeterli değildir [19, 23–25]. Bu yüzden bu çalışmada CSA'nın yerel arama performansını iyileştirmek için ABC algoritmasının çok başarılı yerel arama yeteneğinden faydalanan ve ABC-CSA adı verilen yeni bir yöntem geliştirilmiştir. Bu yöntemde CSA'nın hiper mutasyon aşamaları olan ters mutasyon ve ikili mutasyon işlemleri ile ABC algoritmasının arama uzayındaki yerel aramadan sorumlu olan işçi arı ve gözcü arı aşamaları değiştirilmiştir. ABC-CSA'nın sözde kodu Algoritma 3.3'de verilmektedir.

ABC algoritmasındaki işçi arılar hafızalarındaki yiyecek kaynakları etrafında Eşitlik (3.8)'i kullanarak yeni yiyecek kaynakları ararlar. İşçi arılar buldukları yiyecek kaynaklarının yeri ve kalitesi hakkında kovanda dans ederler. Gözcü arılar ise işçi arılar tarafından gerçekleştirilen dansları izlerler ve Eşitlik (3.3) ile hesaplanan olasılık değerlerine bakarak uçulacak bir yiyecek kaynağına karar verirler. Bu eşitliğe göre yüksek kaliteli çözümlerin bir gözcü arı tarafından seçilmesi daha muhtemeldir. Bir gözcü arı tarafından seçilen çözüm tekrar Eşitlik (3.8) kullanılarak sömürülür. Bir çözümün

sömürülmesinden sonra üretilen yeni çözüm eski çözümden daha iyiyse eski çözümün yerine popülasyona koyulur.

Algoritma 3.3 ABC-CSA algoritması

1: Kontrol parametrelerine değerlerin atanması: antikor sayısı (P), maksimum iterasyon sayısı, B, MR ve α
2: P adet antikordan oluşan başlangıç popülasyonunu Eşitlik (3.10)'u kullanarak oluştur.
3: Her bir antikor Ab_i için benzerlik değerini hesapla
4: **for** herbir iterasyon **do**
5: Antikorları Eşitlik (3.12)'ye göre klonla
6: klonların benzerlik değerlerini hesapla
7: **for** herbir klon C_i **do**
8: Eşitlik (3.13)'ü kullanarak işçi arıları gönder:

$$\hat{C}_{ij} = \begin{cases} C_{ij} + \phi_{ij}(C_{ij} - C_{kj}) & , \quad if\ R_j < MR \\ C_{ij} & , \quad \text{aksi takdirde} \end{cases} \tag{3.13}$$

/* Eşitlikte $k \in \{1, 2, ..., P \times \alpha\}$, i. klona karşılık gelen antikorun klonlarının indekslerinden farklı olması şartıyla i. klonun rasgele seçilen bir komşusudur */
9: Açgözlü seçim gerçekleştir
10: **end for**
11: **for** herbir antikor Ab_i **do**
12: Ab_i'nin klonları arasından en yüksek benzerlik değerine sahip olanı (C_j) bul
13: $Ab_i := C_j$
14: **end for**
15: **for** herbir antikor Ab_i **do**
16: Eşitlik (3.3)'ü kullanarak olasılık değerini hesapla
17: **end for**
18: **for** herbir gözcü arı **do** **//Gözcü arı ve antikorların sayısı eşittir**
19: Antikorların olasılık değerlerini dikkate alarak bir Ab_i antikorunu seç
20: Seçilen Ab_i antikoru üzerinde Eşitlik (3.14)'ü kullanarak yerel arama gerçekleştir.

$$\hat{A}b_{ij} = \begin{cases} Ab_{ij} + \phi_{ij}(Ab_{ij} - Ab_{kj}) & , \quad if\ R_j < MR \\ Ab_{ij} & , \quad \text{aksi takdirde} \end{cases} \tag{3.14}$$

/* Eşitlikte $k \in \{1, 2, ..., P\}$, $i \neq k$ olması şartıyla i. antikorun rastgele seçilen bir komşusudur */
21: Açgözlü seçim gerçekleştir
22: **end for**
23: En kötü $\%B$ adet benzerlik değerine sahip antikorların yerine Eşitlik (3.10)'u kullanarak yenilerini üret.
24: **end for**

4. BÖLÜM

DENEYLER

4.1. Veri Kümeleri

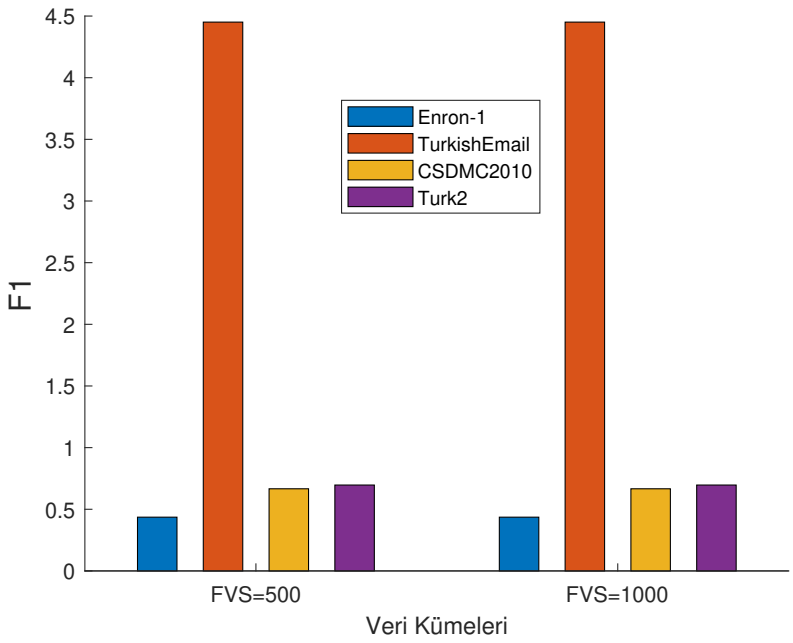
Deneylerde kullanılan veri kümelerinden birincisi 2949 (%68.15) adet normal e-posta ve 1378 (%31.85) adet spam e-posta olmak üzere toplamda 4327 adet İngilizce e-posta içermektedir. ICONIP 2010 organizasyonu ile ilişkili veri madenciliği yarışmasında kullanılan veri kümelerinden biri olan bu veri kümesi CSDMC2010 spam korpus olarak adlandırılmıştır. Bu veri kümesi bir e-posta standardı olan ".eml" uzantılı dosyalardan oluşmaktadır ve toplamda 82148 adet benzersiz terim içermektedir.

TurkishEmail olarak adlandırılan ikinci veri kümesi 400 (%50) adet spam e-posta ve 400 (%50) adet normal e-posta olmak üzere toplamda 800 adet Türkçe e-postadan oluşmaktadır [50]. Bu veri kümesi toplamda 25650 farklı terim içermektedir.

Üçüncü veri kümesi pek çok çalışmada bir ölçüt olarak kullanılan ve literatürde bilinen Enron-1 veri kümesidir [28]. Bu veri kümesi 3672 (%71) adet normal ve 1500 (%29) adet spam e-posta olmak üzere toplamda 5172 adet e-posta içermektedir. Bu veri kümesindeki normal e-postalar sadece bir kişiye ait olduğundan kişiselleştirilmiş veri kümesi olarak tanımlanabilir. Veri kümesindeki spam e-postalar makalenin üçüncü yazarı (Paliouras) tarafından toplanılmıştır [28] ve toplamda 47939 farklı terim içermektedir.

Türkçe e-postaları spam veya normal olarak sınıflandıran bir çalışmada [9] kullanılan veri kümesi makalenin ikinci yazarından istenmiş ve bu veri kümesinin çok az değiştirilmiş versiyonu deneylerde kullanılmıştır. Çalışmalarda kullanılan bu veri kümesi Türkçe e-postalardan oluşan ikinci veri kümesi olduğundan dolayı geri kalan kısımda Turk2 olarak adlandırılacaktır. Turk2 veri kümesi 405 adet spam e-posta ve 355 adet normal e-posta olmak üzere toplamda 760 adet e-postadan oluşmaktadır.

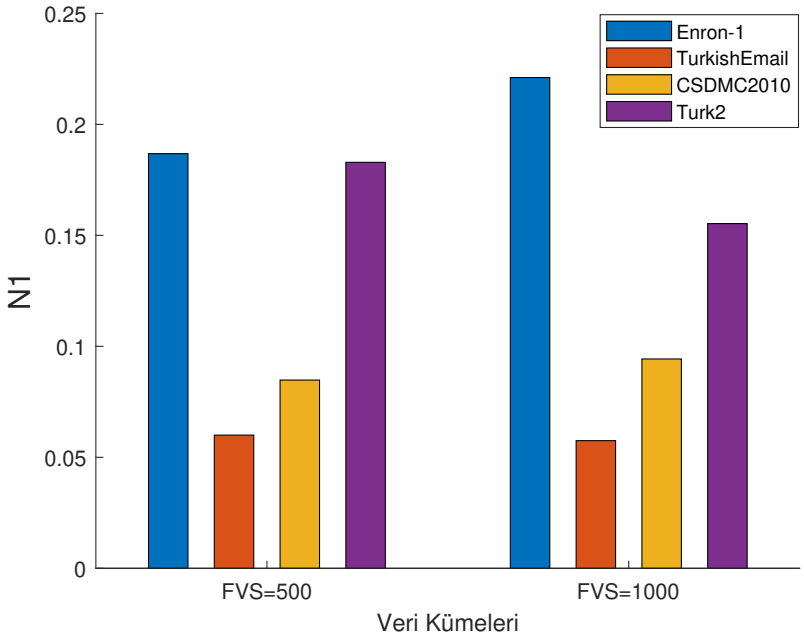
CSDMC2010, TurkishEmail, Enron-1 ve Turk2 veri kümeleri karmaşıklık, seyreklik ve dengesizlik oranları açılarından incelenmiştir. Veri kümelerinin karmaşıklığını ölçmek için Tin ve Basu [51] tarafından önerilen yöntemlerden bazıları kullanılmıştır. İlk karmaşıklık ölçütü, F1 (Fisher's discriminant ratio) adı verilen ve farklı sınıflardaki öznitelikler arasında üst üste binmeleri hesaplayan yöntemdir. Yüksek F1 değerleri sınıflar arasında küçük çakışmaya sahip en az bir öznitelik olduğunu gösterirken düşük F1 değerleri tek başına hiçbir özniteliğin sınıfları ayıramadığı daha karmaşık problemleri gösterir. Şekil 4.1'de veri kümelerinin F1 değerleri, TurkishEmail veri kümesinin en az bir tane ayırt edici özniteliğe sahip ve diğer iki veri kümesine kıyasla daha lineer bir problem olduğunu gösterirken Enron-1 veri kümesinin ise lineer olmayan bir problemi temsil ettiğini göstermektedir.



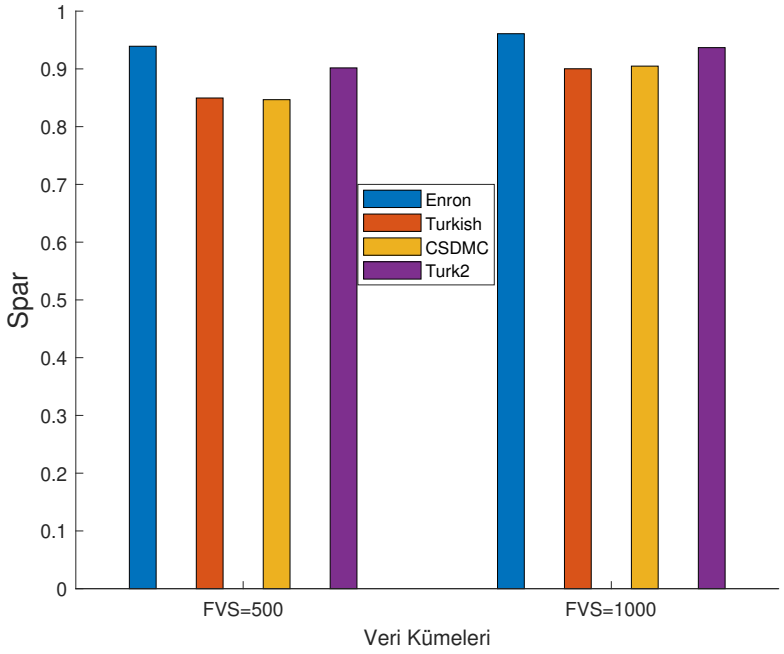
Şekil 4.1. F1 yöntemiyle veri kümelerinin karmaşıklık kıyaslaması

N1 adı verilen ikinci karmaşıklık ölçütü, sınıfların dağılımlarının ayrılabilirliğini ölçer. Bu yöntemde ilk olarak sınıfa bakılmaksızın veri kümesindeki tüm noktaları birbirine bağlayan minimum tarayan ağaç oluşturulur. Daha sonra bu minimum tarayan ağaçta bir kenarla karşı sınıfa bağlanan noktaların sayısı hesaplanır ve elde edilen bu sayı veri kümesindeki tüm noktaların sayısına bölünerek N1 değeri elde edilir. Yüksek N1 değerleri sınıfları birbirlerinden ayırt etmek için daha karmaşık sınırlara ihtiyaç duyulduğunu ve sınıflar arasında daha fazla üst üste binmelerin olduğunu anlatmaktadır.

Şekil 4.2'ye göre öznitelik vektör boyutu 500 için Enron-1 ve Turk2 veri kümeleri diğer iki veri kümesiyle kıyaslandığında sınıfları ayırmak için daha karmaşık sınırlara ihtiyaç duymaktadır. 4.2 üzerinde öznitelik vektör boyutu 1000 için N1 değerlerine bakıldığında Enron-1 veri kümesinin diğer üç veri kümesinden daha karmaşık olduğu görülmektedir.



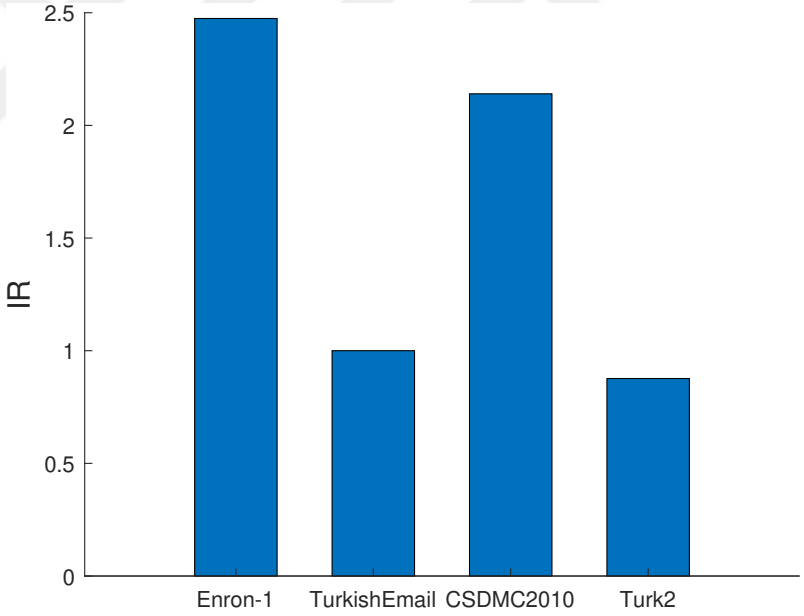
Şekil 4.2. N1 yöntemiyle veri kümelerinin karmaşıklık kıyaslaması



Şekil 4.3. Veri kümelerinin seyreklik değerleri

Bir başka veri kümesi değerlendirme kriteri veri kümelerinin seyrek olup olmadığını gösteren Spar yüzdesidir. Spar yüzdesi, veri kümesinin öznitelik vektörleri çıkarıldıktan sonra elde edilen matristeki 0 içeren hücrelerin toplam sayısının matristeki tüm hücrelerin sayısına bölünmesiyle elde edilmektedir. Öznitelik vektör boyutu 1000 iken Enron, TurkishEmail, CSDMC2010 ve Turk2 veri kümeleri için Spar yüzdeleri sırasıyla %96.09, %90.02, %90.48 ve %93.68'dir. Spar yüzdeleri bu veri kümelerinin oldukça seyrek olduğunu göstermektedir.

Son veri kümesi değerlendirme kriteri olan IR değeri normal e-posta sayısının spam e-posta sayısına bölünmesiyle elde edilmektedir ve bir veri kümesinin dengesiz olup olmadığını göstermektedir. Enron, TurkishEmail, CSDMC2010 ve Turk2 veri kümelerinin IR değerleri sırasıyla 2.47, 1, 2.14, 0.876'dır. Bu değerlere göre Enron, CSDMC2010 ve Turk2 veri kümeleri dengesizken TurkishEmail veri kümesi dengelidir.



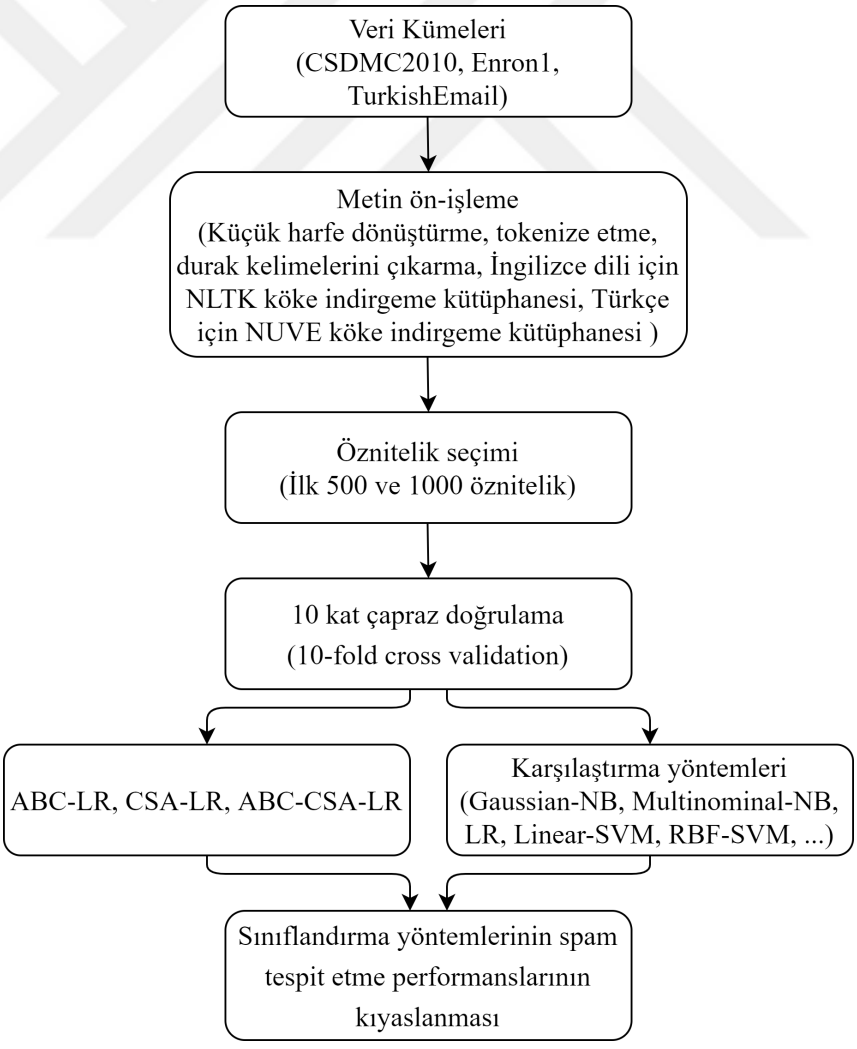
Şekil 4.4. Veri kümelerinin dengesizlik oranları

Bu tez çalışmasında kullanılan son veri kümesi Spambase [52] olarak adlandırılan veri kümesidir. Yapay bağışıklık sistemi algoritmalarını kullanarak spam e-posta tespiti yapan çalışmalarda [17, 46] bu veri kümesinin kullanılması ve bu çalışmada önerilen CSA-LR ve ABC-CSA-LR yöntemlerinin bir tür yapay bağışıklık sistemi algoritmasına dayanmasından dolayı önerilen yöntemlerin performansını kıyaslamalı bir şekilde değerlendirebilmek için bu veri kümesi deneysel çalışmalarda kullanılmıştır. Spambase veri kümesi 1813 (%39) adet spam e-posta ve 2788 (%61) adet normal e-posta

olmak üzere toplamda 4601 adet e-postadan oluşmaktadır. Bu veri kümesindeki e-postalar işlenmiş haldedir ve orijinal formlarında değillerdir. Bu yüzden veri ön-işlemeye ihtiyaç duymazlar. Bu veri kümesindeki her bir e-posta 58 boyutlu bir öznitelik vektörü ile temsil edilmektedir. Öznitelik vektörünün son elamanı 0 veya 1 değerinden oluşmaktadır ve e-postanın spam (1) veya normal (0) olduğunu göstermektedir.

4.2. Veri Ön İşleme

Spam e-posta tespit etme probleminde bir grup e-posta $D = \{d_1, d_2, ..., d_m\}$ ve iki sınıf $C = \{c_{spam}, c_{normal}\}$ verilmektedir. Problem, bir e-postayı $d_i \in D$ iki sınıftan birine atamaktır. Bu tez çalışmasındaki araştırma metodolojisinin ana adımları Şekil 4.5'de verilmektedir.



Şekil 4.5. Spam e-posta tespit etme modeli

Algoritma 4.1 Veri ön-işleme adımları

1: Veri kümesinde her bir dokümandaki bütün karakterler küçük harfe çevrildikten sonra tüm ifade kelimelere, sembollere ve anlamlı elemanlara ayrılır.

2: Aday terimleri belirlemek için noktalama işareti, boşluk, html etiketleri gibi metin olmayan ifadeler kaldırılır.

3: Kelimelere morfoloji işlemi uygulanarak çekim ekleri çıkartılır ve temel formuna dönüştürülür (stemming/lemmatizing).

4: Dillerde çok sık tekrarlayan ve herhangi bir bilgi sağlamayan durak kelimeleri (stopwords) çıkarılır ve kelime çantası (Bag Of Words - BOW) modeli kelimelerin sırasından ve gramerden bağımsız olarak elde edilir. BOW modeli tüm dokümanlardaki tüm eşsiz terimleri içeren bir terimler sözlüğünü oluşturur.

$$B = \{t_1, t_2, ..., t_{|B|}\}, \tag{4.1}$$

B terim kümesidir ve $|B|$ tüm dokümanlardaki eşsiz tüm terimlerin toplam sayısıdır.

5: Terim kümesi oluşturulduktan sonra her bir doküman $d_i \in D$ bir vektör $\vec{d_i} = [w_1, w_2, ..., w_{|B|}]$ ile temsil edilir. Bu vektörde $w_1, w_2, ..., w_{|B|}$ değerleri $t_1, t_2, ..., t_{|B|}$ terimlerine karşılık gelen ağırlıklardır. Ağırlıklar $tf - idf$ yöntemi kullanılarak hesaplanabilir.

$$tf - idf(t_j, d_i) =< w_j, d_i >= tf(t_j, d_i) \times log\frac{|D|}{|D_{t_j}|}, \tag{4.2}$$

$tf(t_j, d_i)$, d_i dokümanında geçen t_j terimlerinin toplam sayısıdır. $|D|$ toplam doküman sayısını ifade ederken $|D_{t_j}|$ ise t_j terimini içeren toplam doküman sayısıdır.

6: Eşsiz terim sayısı ne kadar fazla olursa vektör uzayı boyutu ve hesaplama karmaşıklığı o kadar fazla olur. Vektör uzayı boyutunu düşürmek için öznitelik seçimi (feature selection) yöntemleri uygulanır. Daha fazla bilgi sağlayan terimlerin seçimi tamamlanır tamamlanmaz bir d_i dokümanı bu seçilen terimlere karşılık gelen ağırlıklardan oluşan bir öznitelik vektörü ($\vec{d_i'}$) ile temsil edilir.

$$\vec{d_i'} = [w_{1'}, w_{2'}, ..., w_{|B'|}], |B'| << |B| \tag{4.3}$$

İngilizce ve Türkçe veri kümelerinde verilen bir e-postayı işlemek için ilk olarak e-postada yer alan metin ve konu kısımları alınır ve html etiketleri bilgi sağlayıcı olmadıklarından dolayı çıkartılırlar. Tüm harfler küçük harfe dönüştürülür ve tüm metin kelimelere, sembollere ve anlamlı alt elemanlara ayrılır (tokenize etme). Spam e-postalarda ayırt ediciliği olmayan ünlem işareti dışındaki noktalama işaretleri ve durak kelimeleri (stop-words) alt elamanlar içerisinden kaldırılır. Türkçe ve İngilizce dillerindeki dil yapıları ve kelime morfolojileri önemli ölçüde birbirlerinden farklı olduklarından dolayı dile özgü köke indirgeme (stemming/lemmatizing) kütüphanesi kullanılır. Bu kütüphanelerle kelimelere morfoloji işlemi uygulanarak çekim ekleri çıkartılır ve temel formuna dönüştürülür. İngilizce veri kümelerinde kelimelerden morfolojik ekleri çıkarmak ve sadece kelime kökünü bırakmak için NLTK köke indirgeme kütüphanesi [53] kelimelere uygulanılmıştır. Bu tez çalışmasındaki veri ön işleme adımları Algoritma 4.1'te olduğu gibi özetlenebilir [54].

Ön işleme adımları yapılır yapılmaz makine öğrenmesi algoritmaları öznitelik vektörü olarak ifade edilen veriye uygulanır ve sınıflandırıcı Eşitlik (4.4)'de gösterildiği gibi sınıflara dokümanları eşler.

$$\phi(\vec{d_i'}, c_j) = \begin{cases} 1, & if \quad \vec{d_i'} \in c_j \\ 0, & \text{aksi takdirde} \end{cases} \tag{4.4}$$

4.2.1. Morfoloji İşlemi

Türkçe yapısı gereği İngilizce dilinden farklıdır. İngilizce spam e-postaların filtrelenmesine yönelik çalışmalar birebir olarak Türkçe e-postalara uygulandığında çok daha düşük başarılar elde edilebilir. Türkçe eklemeli bir dil olup İngilizce'ye göre çok daha karmaşık bir dildir. Türkçe'de eklemelerle oluşturulan bir kelime İngilizce'de birden fazla kelimeyle oluşan bir cümleye karşılık gelebilir. Ya da Türkçe bir kelimeye eklemelerle çok sayıda kelime elde edilebilir (kitap, kitaplar, kitaplarım, kitaplarımda vb. kelimelerin yalın hali kitap kelimesidir).

Yapısal karmaşıklığın yanı sıra başa çıkılması gereken bir diğer önemli mesele özel Türkçe karakterlerdir ('ç', 'ğ', 'ı', 'ö', 'ş', 'ü'). Çünkü tüm klavyeler özel Türkçe

karakterleri desteklemez. Örneğin bir kişi “çalışmak” kelimesi yerine “calismak” yazabilir [9]. Bu problemin üstesinden gelmek için ilk önce Türkçe veri kümesindeki yalın hale dönüştürülememiş kelimelerin alternatifleri üretilmektedir [55]. Daha sonra bu alternatif kelimeler teker teker morfoloji işlemine tabi tutulmaktadır. Alternatif kelimelerden biri morfoloji işleminden sonra yalın hale dönüştüyse bu alternatif kelimenin yalın haliyle yalın hale dönüştürülememiş kelime değiştirilmektedir. Tüm alternatif kelimelere morfoloji işlemi uygulanmasına rağmen hiç biri yalın hale dönüştürülemediyse bu kelime olduğu gibi bırakılmaktadır. Örneğin “isciler” kelimesinin alternatif versiyonları “işciler”, “isçiler” ve “işçiler” olarak üretilmekte ve bu kelimelere sırasıyla morfoloji işlemi uygulandığında “işçiler” alternatif kelimesi “işçi” kelimesine dönüşmekte ve veri kümesindeki “isciler” kelimesinin yerine geçmektedir.

E-posta içeriğinde ki kelimelerin köklerini doğru bir şekilde bulabilmek spam e-postaların tespitine yönelik çalışmaların başarısına doğrudan katkı sağlar. Var olan bir araştırma, kelime köklerinin dokümanı iyi temsil ettiğini ve bu kelimelerin o dokümandaki sırasının daha küçük bir öneme sahip olduğunu belirtir [56]. Yani kısaca e-posta içerisindeki kelimelerin yalın hallerini bulmak çalışmalarımızın başarısı açısından hayati önem taşımaktadır. Bu durum Türkçe spam e-postalarının tespitine yönelik çalışmaları farklılaştırır ve Türkçe e-postalara yönelik ayrı bir çalışmayı zorunlu kılar.

Çalışmalarımızda, Türkçe kelimelere morfoloji analizi yapmak ve bu kelimelerin yalın hallerini bulmak için yazılmış Türkçe doğal dil işleme kütüphanesi kullanılmıştır [57]. Kullanılan bu kütüphane sayesinde kelimelerdeki yapım ekleri çıkarılmazken çekim ekleri çıkarılmış ve kelimelerin yalın halleri üretilmiştir.

4.3. Öznitelik Seçme Yöntemleri

Kelime çantasındaki terimler belirlendikten sonra her bir doküman bu terimlere karşılık gelen frekans değerleriyle temsil edilir. Bu tez çalışmasında terim frekanslarının hesaplanmasında iki yöntem kullanılmıştır. Birinci yöntemde, söz konusu terim dokümanda varsa 1 değerini alırken dokümanda yoksa 0 değerini alır. Terim frekanslarının hesaplanmasında kullanılan ikinci yöntem $tf - idf$ yöntemidir.

4.3.1. Terim Frekansı Ters Doküman Frekansı ($tf - idf$)

Metinlerden oluşan büyük bir veri kümesinde bazı kelimeler hemen hemen tüm metinlerde geçerler ve çok sık kullanılırlar. Bu tür kelimeler metnin asıl içeriği hakkında neredeyse hiç anlamlı bilgi taşımazlar. Eğer biz bu tür kelimeleri metinlerde geçtiği sayılara göre sınıflandırıcıya verirsek daha az sayıda kullanılan ama metinler için çok daha fazla anlam taşıyan kelimeleri gölgeleriz.

Kelimelerin metinlerde geçme sayılarını sınıflandırıcıların kullanımına uygun bir şekilde yeniden ağırlıklandırmada $tf - idf$ yöntemi çok yaygın olarak kullanılmaktadır. Bu yöntemde tf ifadesi, terim frekansı anlamına gelirken idf ifadesi, ters doküman frekansı anlamına gelir. Eşitlik (4.5)'de görüldüğü üzere $tf - idf$, terim frekansı ile ters doküman frekansının çarpımıyla elde edilen bir değerdir.

$$tf - idf(t, d) = tf(t, d) \times idf(t) \tag{4.5}$$

Eşitlik (4.5)'de $tf(t, d)$, t teriminin d dokümanında toplam görülme sayısını ifade ederken $idf(t)$ ifadesi ise t teriminin birbirinden farklı dokümanlarda görülme sayısı ile ters orantılı bir değerdir. Bir terim çok sayıda dokümanda görülüyorsa idf değeri düşük olurken az sayıda dokümanda görülüyorsa idf değeri yüksek olur [58].

$$idf(t, d) = log\frac{1 + n_d}{1 + df(d, t)} + 1 \tag{4.6}$$

Eşitlik (4.6)'da n_d, toplam doküman sayısını ve $df(d, t)$, t terimini içeren toplam doküman sayısını belirtmektedir. Son olarak elde edilen $tf - idf$ değeri Eşitlik (4.7)'de görüldüğü gibi normalize edilmektedir.

$$v_{norm} = \frac{v}{\|v\|_2} = \frac{v}{\sqrt{v_1^2 + v_2^2 + ... + v_n^2}} \tag{4.7}$$

Öznitelik vektörünü oluşturan kelimelerin seçimi (feature selection), eğitim kümesindeki kelimelerin bir alt kümesini seçme ve metin sınıflandırmada öznitelik vektörü olarak sadece bu alt kümeyi kullanma sürecidir. Eğitim kümesindeki tüm kelimelerle sınıflandırıcıyı eğitmek yerine az sayıda kelime kullanarak sınıflandırıcıyı eğitmek

özellikle eğitimi zahmetli olan ve uzun süren sınıflandırıcılarda çok verimli olmaktadır. Ayrıca öznitelik vektörünü oluşturan terimlerin seçimi sayesinde gereksiz terimler çıkartılarak sınıflandırıcının başarısı genellikle arttırılmaktadır.

Gereksiz terimleri çıkarmak, hesaplama karmaşıklığını düşürmek ve sınıflandırma başarısını arttırmak amacıyla özniteliklerin seçiminde $tf - idf$ yönteminden faydalanılmaktadır. Bu yöntemde her bir terimin tüm dokümanlarda ki normalize edilmiş $tf - idf$ değerleri hesaplandıktan sonra bu değerler toplanmakta ve her bir terim için toplam $tf - idf$ değeri elde edilmektedir. Daha sonra elde edilen değerler büyükten küçüğe doğru sıralanmakta ve en yüksek değere sahip öznitelik vektörünün boyutu kadar terim seçilmektedir. Bu seçilen terimlere karşılık gelen $tf - idf$ değerleriyle dokümanları temsil eden öznitelik vektörleri oluşturulmaktadır.

4.3.2. Karşılıklı Bilgi (Mutual Information)

Karşılıklı bilgi bir terimin (kelimenin) sınıf hakkında ne kadar bilgi taşıdığını ölçer. Yani bir kelimenin varlığı veya yokluğunun doğru sınıflandırma kararı vermede ne kadar bilgi sağladığını ölçer. Eğer bir kelime bir sınıfta sıklıkla görülürken diğer sınıfta görülmüyorsa bu kelimenin karşılıklı bilgi değerinin yüksek olması beklenir. Bir kelime iki sınıfta da sıklıkla kullanılıyorsa veya çok nadir kullanılıyorsa bu kelimenin karşılıklı bilgi değerinin düşük olması beklenir. Eşitlik (4.8) kelimelerin karşılıklı bilgi değerlerinin hesaplanmasında kullanılmaktadır.

$$KB(T;S) = \sum_{\substack{t\in\{0,1\},\\ s\in\{spam,normal\}}} P(T=t,S=s) \times \log \frac{P(T=t,S=s)}{P(T=t),P(S=s)} \tag{4.8}$$

Eşitlik (4.8)'de S karakteri, sınıfı (spam veya normal) temsil etmektedir ve eğer e-posta s sınıfına aitse 1 değerini alırken s sınıfına ait değilse 0 değerini almaktadır. T karakteri ise terimi temsil ediyor ve eğer e-posta t terimini içeriyorsa 1 değerini alırken, içermiyorsa 0 değerini almaktadır. Eşitlik (4.9), Eşitlik (4.8)'de verilen ifadeye denktir [59].

$$KB(T;S) = \frac{N_{11}}{N} log_2 \frac{NN_{11}}{N_{1.}N_{.1}} + \frac{N_{01}}{N} log_2 \frac{NN_{01}}{N_{0.}N_{.1}} + \frac{N_{10}}{N} log_2 \frac{NN_{10}}{N_{1.}N_{.0}} + \frac{N_{00}}{N} log_2 \frac{NN_{00}}{N_{0.}N_{.0}} \quad (4.9)$$

Eşitlik (4.9)'da N_{00}, t terimini içermeyen ve spam olan e-postaların sayısını; N_{10}, t terimini içeren ve spam olan e-postaların sayısını; N_{01}, t terimini içermeyen ve normal olan maillerin sayısını; N_{11}, t terimini içeren ve normal olan maillerin sayısını ifade etmektedir. Eşitlikte (4.9)'da yer alan $N_{1.}, N_{0.}, N_{.1}, N_{.0}$ ve N değerleri sırasıyla Eşitlik (4.10), (4.11), (4.12), (4.13) ve (4.14)'de verildiği gibi hesaplanılmaktadır.

$$N_{1.} = N_{10} + N_{11} \quad (4.10)$$

$$N_{0.} = N_{00} + N_{01} \quad (4.11)$$

$$N_{.1} = N_{01} + N_{11} \quad (4.12)$$

$$N_{.0} = N_{00} + N_{10} \quad (4.13)$$

$$N = N_{00} + N_{10} + N_{01} + N_{11} \quad (4.14)$$

Bu tez çalışmasında eğitim kümesindeki terimlerin karşılıklı bilgi değerlerini hesaplayabilmek için Eşitlik (4.9) kullanılmaktadır ve en yüksek karşılıklı bilgi değerine sahip belirli sayıda terim seçilmektedir. Örneğin öznitelik vektörünün 100 terimden oluşması belirlendiyse en yüksek karşılıklı bilgi değerine sahip 100 terim öznitelik vektörünü oluşturmaktadır.

Karşılıklı bilgi yöntemi kullanılarak oluşturulan öznitelik vektöründeki terimlerin bir sınıfı iyi temsil etmesi beklenir. Yani bir terim spam sınıfında yer alan metinlerde sıklıkla görülürken normal sınıfında yer alan metinlerde çok az görülüyorsa ya da hiç görülmüyorsa bu terim spam sınıfı için iyi bir temsilcidir. Aynı şekilde bir terim normal sınıfında sıklıkla görülürken spam sınıfında görülmüyorsa ya da çok az görülüyorsa bu terim normal sınıfı için iyi bir temsilcidir. Bu özelliklere sahip terimlerin karşılıklı bilgi değerlerinin yüksek olması beklenir. Karşılıklı bilgi yöntemi, tüm kelimelere kıyasla çok daha az sayıda kelime ile sınıflandırma işleminin başarılı bir şekilde yapılmasını sağlar. Öznitelik vektörünü oluşturan kelimeler ne kadar iyi temsil yeteneğine sahipse o kadar iyi sınıflandırıcı başarısı elde edilir. Yani öznitelik vektörünü oluşturan kelimelerin seçimi sınıflandırıcının başarısına doğrudan katkı sağlar.

5. BÖLÜM

BULGULAR

Deneylerin ilk aşamasında TurkishEmail ve CSDMC2010 veri kümeleri üzerinde Gaussian NB ve multinomial NB yöntemlerinin sınıflandırma sonuçları üretilmiş ve düzleştirme parametresine karşı multinomial NB yönteminin hassasiyeti analiz edilmiştir. İkinci aşamada lineer ve radial tabanlı SVM yöntemleri TurkishEmail ve CSDMC2010 veri kümelerine uygulanmış ve ceza parametresine karşı hassasiyetleri araştırılmıştır. Üçüncü aşamada sırasıyla gradient descent, PSO ve DE algoritmalarıyla eğitilen LR yöntemlerinin sınıflandırma performansları incelenmiştir. Dördüncü aşamada bu tez çalışmasında önerilen yöntemlerin (ABC-LR, CSA-LR ve ABC-CSA-LR) sınıflandırma performansları gözlemlenmiş ve öğrenme oranı, regülizasyon katsayısı, öznitelik vektör boyutu parametreleriyle ABC ve CSA algoritmalarına ait kontrol parametrelerinin sınıflandırma doğruluğu üzerindeki etkileri incelenmiştir. Son aşamada ise Gaussian NB, multinomial NB, lineer SVM, radial SVM, LR, PSO-LR, DE-LR ve bu tez çalışmasında önerilen yöntemler karşılaştırılmıştır. Ayrıca önerilen yöntemlerin Enron ve Spambase veri kümeleri üzerindeki sınıflandırma performanslarıyla literatürde bilinen çalışmalarda sunulan yöntemlerin sınıflandırma performansları kıyaslanılmıştır. Deneysel çalışmalarda beş farklı veri kümesi kullanılmıştır. Bu veri kümelerinden ikisi (TurkishEmail, Turk2) Türkçe dilinde e-postalar içerirken iki tanesi (CSDMC2010, Enron-1) İngilizce dilinde e-postalar içermektedir. Spambase veri kümesi ise diğer dört veri kümesinden farklı olarak öznitelik vektörleri çıkarılmış haldedir. Yani bu veri kümesindeki e-postalar metin içermemekte ve sayılardan oluşan bir vektörle temsil edilmektedirler. Bu yüzden Spambase veri kümesinde veri ön-işleme adımlarına ihtiyaç duyulmamaktadır. Spambase veri kümesi haricinde diğer dört veri kümesinde nihai sınıflandırma sonucunu elde etmek için 10-katmanlı çapraz doğrulama (10-fold cross validation) tekniği kullanılmıştır.

Bu teknik sayesinde verileri test ve eğitim sınıflarına bölerken sınıflandırma sonucuna yansıyabilecek avantajlı veya dezavantajlı etkiler ortadan kaldırılarak sınıflandırıcıların performansı hakkında oluşabilecek yanlış değerlendirmeler engellenmektedir. Bu teknikte veri kümesi ilk olarak 10 eşit parçaya bölünür ve her seferinde bir parçası test için geriye kalan 9 parçası eğitim için kullanılır. Bu işlem toplamda 10 kez yapılır ve elde edilen sonuçların ortalaması alınarak nihai sınıflandırma doğruluğu elde edilir.

5.1. NB Sınıflandırıcıların Başarımları

5.1.1. Gaussian NB sınıflandırıcı ile sınıflandırma

Gaussian NB sınıflandırıcı iki farklı öznitelik vektör boyutu {500, 1000} ile eğitilmiş ve her bir veri kümesi için scikit-learn kütüphanesi [60] kullanılarak iki deney gerçekleştirilmiştir. Tablo 5.1, TurkishEmail ve CSDMC2010 veri kümeleri üzerinde Gaussian NB sınıflandırıcının sınıflandırma doğruluklarını göstermektedir. Hem veri kümelerinin anlatıldığı bölümde (4.1 Veri Kümeleri) bahsedildiği hem de Şekil 4.4'de görüldüğü üzere CSDMC2010 veri kümesi TurkishEmail veri kümesinden daha karmaşık ve lineer olmayan bir yapıdadır. Ayrıca TurkishEmail veri kümesi CSDMC2010 veri kümesi ile karşılaştırıldığında çok daha az sayıda benzersiz terime sahiptir. Bu sebeblerden dolayı öznitelik vektör boyutları açısından değerlendirildiğinde sınıflandırma doğrulukları arasındaki fark TurkishEmail veri kümesinde daha azdır.

Tablo 5.1. Gaussian NB sınıflandırıcının sınıflandırma doğrulukları, FVS: Öznitelik vektör boyutu

FVS	TurkishEmail veri kümesi	CSDMC2010 veri kümesi
500	%95.38	%93.71
1000	%96.63	%95.92

5.1.2. Multinomial NB sınıflandırıcı ile sınıflandırma

Multinomial NB sınıflandırıcı iki farklı öznitelik vektör boyutu {500, 1000} ve beş farklı düzleştirme parametresi ($\alpha = \{1.0e-10, 0.001, 0.01, 0.1, 1\}$) ile eğitilmiştir. Scikit-learn kütüphanesi [60] kullanılarak her bir veri kümesi için 10 tane deney gerçekleştirilmiştir. Deneylerde normalize edilmiş $tf - idf$ değerli öznitelik vektörleri kullanılmıştır.

Multinomial NB'de genellikle öznitelik vektörlerinin terim sayılarından oluşmasına rağmen $tf - idf$ değerli öznitelik vektörlerinin pratikte başarılı sonuçlar verdiği bilinmektedir [58]. Tablo 5.2 ve Tablo 5.3, sırasıyla TurkishEmail ve CSDMC2010 veri kümeleri üzerinde multinomial NB sınıflandırıcının sınıflandırma doğruluklarını göstermektedir. Tablo 5.2 ve Tablo 5.3'den görüldüğü üzere en iyi sonuçlar, α sıfıra çok yakın olduğunda elde edilmiştir.

Tablo 5.2. TurkishEmail veri kümesi üzerinde Multinominal NB sınıflandırıcının verdiği sonuçlar, FVS: Öznitelik vektör boyutu

FVS	α (düzleştirme parametresi)				
	$1.0e-10$	0.001	0.01	0.1	1
500	**%96.88**	%96.75	%96.75	%96.50	%95.63
1000	**%97.62**	**%97.62**	**%97.62**	%97.38	%97.12

Tablo 5.3. CSDMC2010 veri kümesi üzerinde Multinominal NB sınıflandırıcının verdiği sonuçlar, FVS: Öznitelik vektör boyutu

FVS	α (düzleştirme parametresi)				
	$1.0e-10$	0.001	0.01	0.1	1
500	**%90.93**	%89.79	%89.51	%89.30	%88.89
1000	**%94.55**	%93.79	%93.50	%93.18	%92.02

5.2. SVM Sınıflandırıcıların Başarımları

5.2.1. Lineer SVM sınıflandırıcının başarımı

Lineer SVM sınıflandırıcı iki farklı öznitelik vektör boyutu $FVS = \{500, 1000\}$ ve dokuz farklı ceza parametresi $C = \{2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}\}$ ile eğitilmiştir. En iyi parametreleri belirleyebilmek için grid arama tekniği kullanılmıştır [61]. Her bir veri kümesi için LibSVM [62] kütüphanesi kullanılarak 18 deney gerçekleştirilmiştir.

Tablo 5.4 ve Tablo 5.5, lineer SVM sınıflandırıcının sırasıyla TurkishEmail ve CSDMC2010 veri kümeleri üzerinde sınıflandırma başarımlarını göstermektedir. TurkishEmail veri kümesi üzerinde en iyi sonuçlar öznitelik vektör boyutu 500 iken C katsayısı 2^4 olduğunda elde edilirken öznitelik vektör boyutu 1000 iken C katsayısı 2^5 olduğunda elde edilmiştir. CSDMC2010 veri kümesi üzerinde ise en iyi sonuçlar

öznitelik vektör boyutu 500 iken C katsayısı 2^2 olduğunda elde edilirken öznitelik vektör boyutu 1000 iken C katsayısı 2^4 olduğunda elde edilmiştir. Hem TurkishEmail hem de CSDMC2010 veri kümeleri için en kötü sonuçlar C katsayısı 2^{-3} olduğunda elde edilmiştir. Tablo 5.4 ve Tablo 5.5'deki sonuçlardan görüldüğü üzere lineer SVM sınıflandırıcının performansı C katsayısının değerine duyarlıdır.

Tablo 5.4. TurkishEmail veri kümesi üzerinde lineer SVM sınıflandırıcının sınıflandırma doğrulukları (%)

FVS \ C	2^{-3}	2^{-2}	2^{-1}	2^{0}	2^{1}	2^{2}	2^{3}	2^{4}	2^{5}
500	93.00	94.25	96.37	97.50	98.00	97.88	98.13	**98.63**	98.50
1000	93.75	95.38	96.37	97.75	98.00	98.38	98.50	98.75	**98.88**

Tablo 5.5. CSDMC2010 veri kümesi üzerinde lineer SVM sınıflandırıcının sınıflandırma doğrulukları (%)

FVS \ C	2^{-3}	2^{-2}	2^{-1}	2^{0}	2^{1}	2^{2}	2^{3}	2^{4}	2^{5}
500	92.30	95.36	96.64	97.49	97.87	**98.10**	97.96	98.07	97.82
1000	94.08	96.08	97.33	97.98	97.96	98.24	98.35	**98.42**	98.19

5.2.2. Radial Basis SVM sınıflandırıcının başarımı

RBF kernel (Radial Basis Function: $K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2), \gamma > 0$) kullanan SVM sınıflandırma yöntemi CSDMC2010 ve TurkishEmail veri kümeleri üzerinde iki farklı öznitelik vektör boyutu $FVS = \{500, 1000\}$ ve aşağıda verilen parametreler ile eğitilmiştir. Hangi parametrelerin en uygun olduğuna karar vermek için grid arama tekniği kullanılmıştır [61]. Her bir veri kümesi ve her bir öznitelik vektör boyutu için LibSVM [62] kütüphanesini kullanan 99 deney gerçekleştirilmiştir. Elde edilen sınıflandırma başarımları Tablo 5.6, Tablo 5.7, Tablo 5.8 ve Tablo 5.9'da paylaşılmıştır. TurkishEmail veri kümesi üzerinde %98.75 başarı oranıyla en iyi sonuç $C = 2^5$, $\gamma = 2^1$ ve $FVS = 1000$ olduğunda elde edilmiştir. CSDMC2010 veri kümesi üzerinde en iyi sonuç $C = 2^3$, $\gamma = 2^{-1}$ ve $FVS = 1000$ olduğunda elde edilmiştir. Deneysel sonuçlar, RBF kernelli SVM sınıflandırıcının CSDMC2010 veri kümesi üzerinde öznitelik vektör boyutları 500 ve 1000 için biraz daha iyi sınıflandırma başarısına sahip olduğunu gösterirken lineer kernelli SVM sınıflandırıcının TurkishEmail veri kümesi üzerinde

öznitelik vektör boyutları 500 ve 1000 için biraz daha iyi olduğunu göstermektedir. RBF kernel kullanan SVM sınıflandırma yöntemi aşağıda verilen parametreler ile eğitilmiştir.

$C = \{2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}\}$

$\gamma = \{2^{-15}, 2^{-13}, 2^{-11}, 2^{-9}, 2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 2^{1}, 2^{3}, 2^{5}\}$.

Tablo 5.6. RBF kernelli SVM sınıflandırıcının $FVS = 500$ iken TurkishEmail veri kümesi üzerinde sınıflandırma doğrulukları (%)

γ \ C	2^{-3}	2^{-2}	2^{-1}	2^{0}	2^{1}	2^{2}	2^{3}	2^{4}	2^{5}
2^{-15}	53.75	53.75	53.75	54.00	54.00	54.00	54.00	54.00	54.00
2^{-13}	53.75	53.75	53.75	54.00	54.00	54.00	54.00	54.00	54.00
2^{-11}	53.75	53.75	53.75	54.00	54.00	54.00	54.00	54.00	54.00
2^{-9}	53.75	53.75	53.75	54.00	54.00	54.00	54.00	54.00	54.00
2^{-7}	53.75	53.75	53.75	54.00	54.00	54.00	54.00	54.00	54.00
2^{-5}	54.00	54.00	54.00	59.37	61.37	61.37	61.37	61.37	61.37
2^{-3}	55.87	55.87	67.37	81.87	82.25	82.25	82.25	82.25	82.25
2^{-1}	94.50	95.87	97.25	97.87	98.12	98.12	98.25	98.37	98.37
2^{1}	90.12	92.87	94.00	96.00	97.25	98.00	98.00	98.25	**98.62**
2^{3}	87.50	87.50	87.87	88.87	90.62	93.00	94.25	96.37	97.50
2^{5}	87.50	87.50	87.50	87.50	87.50	87.50	87.87	88.87	90.62

Tablo 5.7. RBF kernelli SVM sınıflandırıcının $FVS = 1000$ iken TurkishEmail veri kümesi üzerinde sınıflandırma doğrulukları (%)

γ \ C	2^{-3}	2^{-2}	2^{-1}	2^{0}	2^{1}	2^{2}	2^{3}	2^{4}	2^{5}
2^{-15}	54.00	54.00	54.00	54.00	54.00	54.00	54.00	54.00	54.00
2^{-13}	54.00	54.00	54.00	54.00	54.00	54.00	54.00	54.00	54.00
2^{-11}	54.00	54.00	54.00	54.00	54.00	54.00	54.00	54.00	54.00
2^{-9}	54.00	54.00	54.00	54.00	54.00	54.00	54.00	54.00	54.00
2^{-7}	54.00	54.00	54.00	54.00	54.00	54.00	54.00	54.00	54.00
2^{-5}	54.00	54.00	54.00	56.62	57.75	57.75	57.75	57.75	57.75
2^{-3}	54.00	54.00	61.75	79.50	80.00	80.00	80.00	80.00	80.00
2^{-1}	89.62	94.00	96.00	96.50	97.25	97.25	97.25	97.25	97.25
2^{1}	90.25	93.62	95.00	96.00	97.62	98.00	98.25	98.50	**98.75**
2^{3}	87.75	87.75	88.00	89.25	91.12	93.62	95.37	96.37	97.75
2^{5}	87.75	87.75	87.75	87.75	87.75	87.75	88.12	89.50	91.25

Tablo 5.8. RBF kernelli SVM sınıflandırıcının $FVS = 500$ iken CSDMC2010 veri kümesi üzerinde sınıflandırma doğrulukları (%)

γ \ C	2^{-3}	2^{-2}	2^{-1}	2^{0}	2^{1}	2^{2}	2^{3}	2^{4}	2^{5}
2^{-15}	68.21	68.37	69.04	71.29	71.29	71.29	71.29	71.29	71.29
2^{-13}	68.21	68.37	69.04	71.29	71.29	71.29	71.29	71.29	71.29
2^{-11}	68.21	68.37	69.04	71.53	71.53	71.53	71.53	71.53	71.53
2^{-9}	68.21	68.37	69.21	72.06	72.11	72.11	72.11	72.11	72.11
2^{-7}	68.21	68.53	69.41	72.38	72.41	72.41	72.41	72.41	72.41
2^{-5}	68.21	68.56	69.97	73.61	73.94	73.94	73.94	73.94	73.94
2^{-3}	69.21	70.32	73.59	77.63	78.95	78.95	78.93	78.93	78.93
2^{-1}	96.96	97.56	98.02	98.28	98.44	98.51	98.56	**98.58**	**98.58**
2^{1}	85.82	92.13	95.05	96.84	97.54	98.00	98.30	98.12	98.16
2^{3}	68.21	68.21	74.17	78.86	86.89	92.29	95.40	96.65	97.47
2^{5}	68.21	68.21	68.21	68.21	68.21	68.21	74.33	78.90	86.91

Tablo 5.9. RBF kernelli SVM sınıflandırıcının $FVS = 1000$ iken CSDMC2010 veri kümesi üzerinde sınıflandırma doğrulukları (%)

γ \ C	2^{-3}	2^{-2}	2^{-1}	2^{0}	2^{1}	2^{2}	2^{3}	2^{4}	2^{5}
2^{-15}	68.21	68.37	69.04	71.29	71.29	71.29	71.29	71.29	71.29
2^{-13}	68.21	68.37	69.04	71.29	71.29	71.29	71.29	71.29	71.29
2^{-11}	68.21	68.37	69.04	71.48	71.48	71.48	71.48	71.48	71.48
2^{-9}	68.21	68.37	69.11	71.83	71.92	71.92	71.92	71.92	71.92
2^{-7}	68.21	68.53	69.39	72.11	72.20	72.20	72.20	72.20	72.20
2^{-5}	68.21	68.53	69.79	73.15	73.45	73.45	73.45	73.45	73.45
2^{-3}	68.81	69.97	72.94	76.58	77.44	77.44	77.44	77.44	77.44
2^{-1}	95.77	97.51	98.02	98.44	98.51	98.58	**98.63**	98.58	98.58
2^{1}	87.05	93.45	95.77	97.37	98.00	98.07	98.28	98.35	98.37
2^{3}	68.21	68.21	74.68	80.53	88.05	94.08	96.05	97.33	98.02
2^{5}	68.21	68.21	68.21	68.21	68.21	68.21	74.82	80.62	88.07

5.3. LR Sınıflandırıcı ile sınıflandırma başarımları

5.3.1. Gradient descent algoritması ile eğitilen LR sınıflandırıcının başarımı

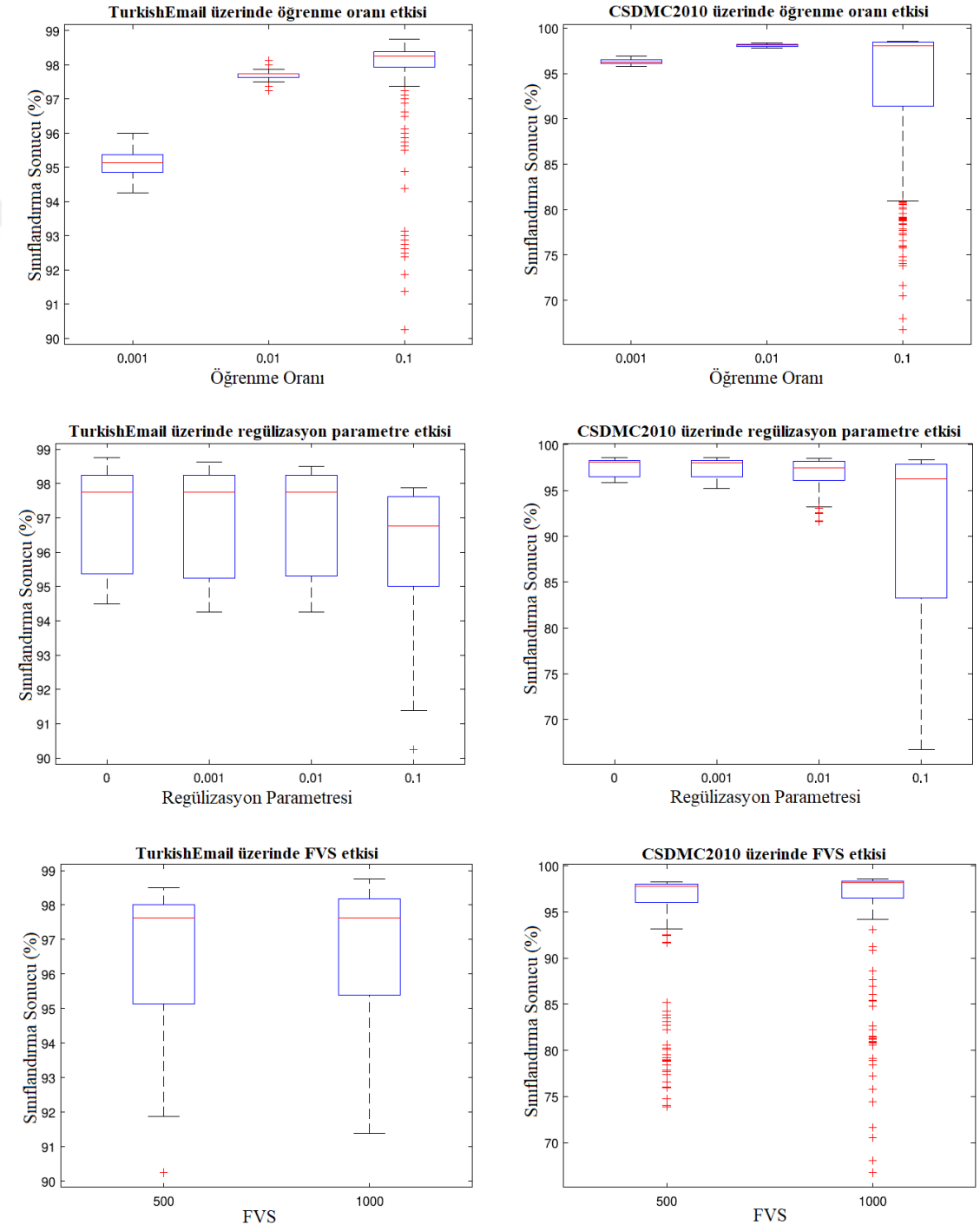
Geleneksel LR modelinde gradient descent algoritması Eşitlik (2.14)'de verilen maliyet fonksiyonunun değerini minimum yapmak için kullanılır. Standart LR sınıflandırıcı, iki farklı öznitelik vektör boyutu ($FVS = \{500, 1000\}$), üç farklı öğrenme katsayısı ($\alpha = \{0.001, 0.01, 0.1\}$) ve dört farklı regülizasyon parametresi ($\lambda = \{0, 0.001, 0.01, 0.1\}$) ile eğitilmiştir. En uygun parametreleri belirleyebilmek için kapsamlı grid yöntemi kullanılmıştır. Her bir veri kümesi için yirmi dört konfigürasyon oluşturulmuş ve her konfigürasyon farklı rastgele tohumlar ile 30 kez tekrarlanmıştır. Sırasıyla TurkishEmail ve CSDMC2010 veri kümeleri için her bir konfigürasyonun 30 kez koşulmasından elde edilen sınıflandırma doğruluklarının en iyi, en kötü, medyan (ortanca), ortalama ve standart sapma değerleri Tablo 5.10 ve Tablo 5.11'de verilmektedir. Şekil 5.1, kutu grafiklerini (box plot graphs) kullanarak tüm koşmaları özetlemektedir.

Tablo 5.10. TurkishEmail veri kümesi üzerinde LR sınıflandırma yönteminin istatistikleri

α	λ	$FVS = 500$					$FVS = 1000$				
		En iyi	En kötü	Medyan	Ort.	Std.	En iyi	En kötü	Medyan	Ort.	Std.
0.001	0	95.50	94.50	95.00	94.98	0.31	95.88	94.63	95.25	95.22	0.31
	0.001	95.50	94.25	95.00	94.97	0.34	96.00	94.25	95.06	95.13	0.37
	0.01	95.88	94.25	95.00	94.99	0.35	95.75	94.62	95.25	95.25	0.28
	0.1	95.63	94.37	95.00	95.01	0.30	95.75	94.38	95.25	95.15	0.38
0.01	0	98.00	97.50	97.75	97.78	0.12	98.00	97.25	97.75	97.72	0.17
	0.001	98.00	97.50	97.75	97.76	0.12	98.13	97.50	97.75	97.75	0.15
	0.01	98.00	97.50	97.75	97.72	0.11	98.00	97.37	97.75	97.73	0.16
	0.1	97.75	97.50	97.63	97.64	**0.07**	97.75	97.37	97.63	97.58	0.10
0.1	0	98.38	98.00	**98.25**	98.24	0.09	**98.75**	98.25	98.50	98.48	0.12
	0.001	**98.50**	98.13	**98.25**	98.29	0.08	98.63	**98.38**	**98.50**	**98.53**	**0.08**
	0.01	**98.50**	**98.25**	**98.25**	**98.31**	**0.07**	**98.50**	98.25	98.38	98.36	0.10
	0.1	97.63	90.25	96.38	95.32	2.32	97.88	91.38	96.88	95.82	2.14

Tablo 5.10, Tablo 5.11 ve Şekil 5.1'den görüldüğü üzere öğrenme katsayısı (α) ve regülizasyon parametresi (λ) 0.1 olduğunda hem TurkishEmail hem de CSDMC2010 veri kümeleri üzerinde algoritma kararsız sonuçlar üretmektedir. En kararlı sonuçlar

ise öğrenme katsayısı 0.01 olduğunda elde edilmekte ve küçük regülizasyon parametre değerleri ({0, 0.001}) performansı iyileştirmektedir. Yine Şekil 5.1'den görüldüğü üzere öznitelik vektör boyutu 1000 iken hesaplama karmaşıklığı göz önünde bulundurulursa öznitelik vektör boyutunun 500 olmasının verileri yeterince iyi temsil edebildiği sonucu çıkarılabilmektedir.



Şekil 5.1. TurkishEmail ve CSDMC2010 veri kümeleri üzerinde öznitelik vektör boyutu ve parametrelerin etkileri

Tablo 5.11. CSDMC2010 veri kümesi üzerinde LR sınıflandırma yönteminin istatistikleri

α	λ	$FVS = 500$					$FVS = 1000$				
		En iyi	En kötü	Medyan	Ort.	Std.	En iyi	En kötü	Medyan	Ort.	Std.
0.001	0	96.45	95.87	96.09	96.10	0.13	96.75	96.22	96.48	96.49	0.14
	0.001	96.38	95.78	96.11	96.13	0.16	96.80	96.22	96.51	96.52	0.16
	0.01	96.36	95.80	96.05	96.06	0.14	96.89	96.08	96.50	96.50	0.17
	0.1	96.31	95.80	96.03	96.06	0.12	96.82	96.29	96.50	96.53	0.14
0.01	0	98.14	97.87	98.03	98.02	0.06	98.40	98.14	98.26	98.27	0.06
	0.001	98.19	97.91	98.03	98.04	0.07	98.38	98.10	98.24	98.24	0.07
	0.01	98.12	97.84	97.99	97.99	0.07	98.38	98.17	98.26	98.27	0.05
	0.1	97.98	97.80	97.89	97.89	**0.05**	98.31	98.07	98.21	98.21	0.05
0.1	0	**98.26**	**98.10**	**98.17**	**98.17**	**0.05**	**98.56**	**98.42**	**98.49**	**98.49**	**0.03**
	0.001	98.17	95.20	98.03	97.94	0.53	**98.56**	98.40	98.47	98.47	0.04
	0.01	98.03	91.65	96.10	95.87	1.96	98.47	93.09	98.27	97.46	1.42
	0.1	85.15	73.85	78.93	79.32	3.23	91.21	66.75	81.24	80.87	6.17

5.3.2. PSO algoritması ile eğitilen LR sınıflandırıcının başarımı

PSO-LR sınıflandırma yöntemi, bu tez çalışmasında önerilen spam filtreleme yöntemleri olan CSA-LR ve ABC-CSA-LR'nin sınıflandırma performanslarıyla kıyaslama yapmak için tasarlanmıştır. Literatürde bilinen bir optimizasyon algoritması olan PSO algoritması LR sınıflandırma modelinin eğitimi aşamasında kullanılmıştır. Bölüm 5.4.2 ve Bölüm 5.4.3'den görüleceği üzere CSA-LR ve ABC-CSA-LR sınıflandırma yöntemleri Spambase veri kümesine uygulanıldığından dolayı PSO-LR sınıflandırma yöntemi de Spambase veri kümesine uygulanılmıştır.

PSO-LR sınıflandırma yöntemi kullanılarak Spambase veri kümesi üzerinde toplam 20 deney gerçekleştirilmiştir ve sınıflandırma performansı üzerinde rastgele veri etkisini azaltmak için her bir deney 30 kez tekrarlanılmıştır. PSO algoritmasının parametrelerine atanacak değerleri tavsiye eden çalışma [63] değerlendirildikten sonra PSO-LR sınıflandırma yönteminin eylemsizlik ağırlığı (inertia weight, w) parametresine 0.6 değeri atanırken hızlanma katsayıları olan c_1 ve c_2 parametrelerine 1.8 değeri atanmış ve 20 farklı popülasyon boyutu ile eğitilmiştir. Deneylerde maksimum döngü sayısına (MCN) 400 değeri verilmiştir. Eğitilecek parametrelerin alt sınır ve üst sınır değerleri sırasıyla -64 ve +64 olarak belirlenilmiştir. En iyi korelasyon katsayısı (CC), F ölçümü (F1), duyarlılık (SN), pozitif tahmin değeri (PPV), negatif tahmin değeri (NPV), özgüllük

(SP) ve saniye cinsinden ortalama harcanan zaman (eğitim ve test için harcanan toplam zamanların ortalama değeri) değerlerine ek olarak her bir deneyin 30 defa koşulmasıyla elde edilen sınıflandırma doğruluklarının (ACC) en iyi, en kötü, ortalama, medyan (ortanca) ve standart sapma değerleri Tablo 5.12 üzerinde gösterilmektedir.

Tablo 5.12. Spambase veri kümesi üzerinde PSO-LR sınıflandırma yönteminin sınıflandırma performansı

P	ACC					CC	SN	SP	PPV	NPV	F1	Süre(sn)
	En iyi	En kötü	Medyan	Ort.	Std.	En iyi						Ort.
10	92.46	72.61	87.72	86.66	4.15	84.18	89.52	94.38	91.20	93.26	90.35	**5.32**
20	91.81	83.41	89.13	88.78	2.19	82.86	90.44	93.18	89.54	93.70	89.62	10.64
30	91.88	85.22	89.64	89.47	1.79	82.99	89.52	94.50	91.07	93.12	89.67	15.88
40	92.10	84.71	89.57	89.19	1.85	83.45	90.63	93.66	90.20	93.80	89.95	21.20
50	92.68	85.80	90.04	90.00	1.40	84.72	**92.28**	93.66	90.28	94.80	90.78	26.54
60	93.04	86.09	90.18	90.13	1.48	85.44	91.36	95.45	92.42	94.36	**91.19**	31.76
70	91.59	88.12	90.40	90.12	1.09	82.38	90.26	94.02	90.48	93.48	89.32	37.00
80	91.59	87.61	90.72	90.26	1.08	82.38	90.44	93.78	90.15	93.59	89.30	46.10
90	92.90	88.12	90.83	90.65	1.29	85.09	89.89	94.86	91.92	93.51	90.89	53.74
100	92.68	85.72	90.72	90.42	1.44	84.61	91.18	95.45	92.68	94.03	90.50	53.00
110	92.83	86.88	90.72	90.59	1.20	84.94	89.89	94.74	91.74	93.51	90.81	58.84
120	92.32	87.75	90.91	90.64	0.95	84.08	**92.28**	94.38	91.12	**94.84**	90.45	63.46
130	92.83	88.41	90.76	90.61	0.94	84.98	90.99	94.62	91.30	94.13	90.91	68.92
140	92.54	87.97	90.72	90.53	1.16	84.32	89.89	94.86	91.84	93.32	90.38	74.16
150	92.97	87.97	90.54	90.56	1.07	85.30	91.36	95.57	92.73	94.36	91.11	79.20
160	91.81	87.17	**91.12**	90.64	1.11	82.79	91.73	95.57	92.64	94.40	89.52	84.68
170	92.32	89.13	91.05	90.99	0.77	83.88	89.89	95.33	92.10	93.41	90.17	89.98
180	92.03	87.97	90.91	90.82	0.98	83.36	90.99	94.50	90.93	94.03	89.96	95.18
190	92.46	**89.86**	90.94	**91.07**	**0.70**	84.16	90.07	**95.69**	**92.97**	93.56	90.24	101.38
200	**93.12**	89.28	90.80	90.87	0.94	**85.54**	90.99	95.45	92.61	94.01	91.16	106.26

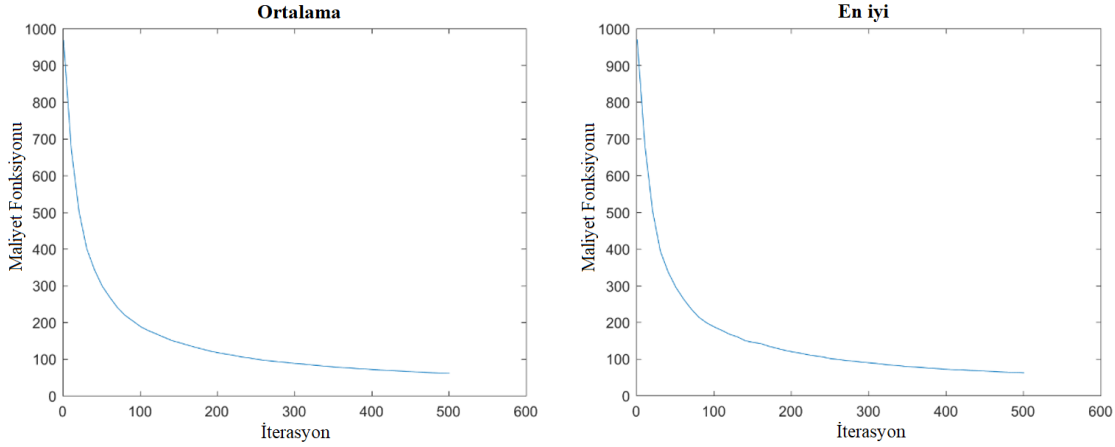
5.3.3. DE algoritması ile eğitilen LR sınıflandırıcının başarımı

Literatürde bilinen ve sıklıkla kullanılan bir optimizasyon yöntemi olan DE algoritması LR sınıflandırma modelinin eğitimi aşamasında kullanılmıştır. DE-LR sınıflandırma yöntemi kullanılarak TurkishEmail ve CSDMC2010 veri kümeleri üzerinde toplam 4 deney gerçekleştirilmiş ve sınıflandırma performansı üzerinde rastgele veri etkisini azaltmak için her bir deney 30 kez tekrarlanılmıştır. DE-LR sınıflandırma yönteminin mutasyon faktörü (F) parametresine 0.5 değeri atanırken geçiş olasılığı parametresine (CR) 0.8 değeri verilmiş ve popülasyon boyutu 50 olarak belirlenilmiştir. Eğitilecek

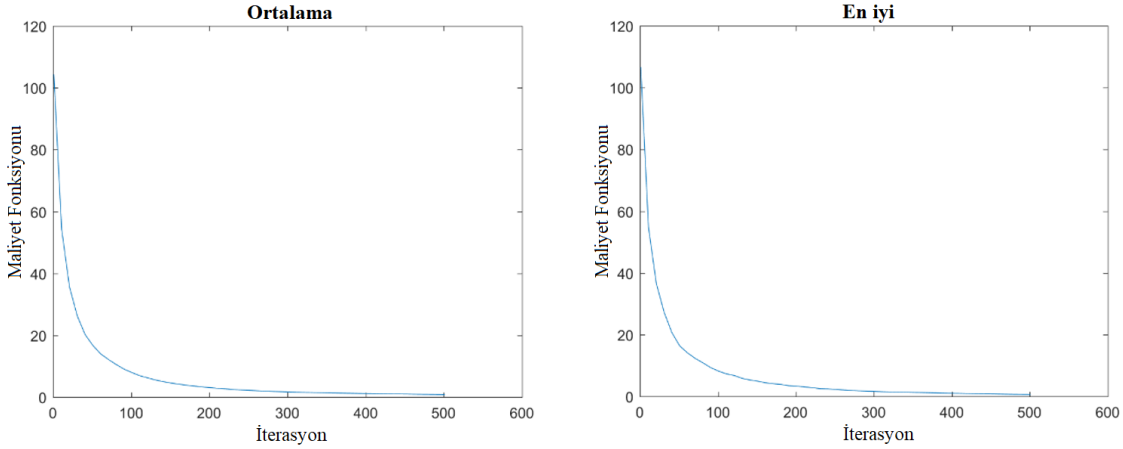
parametrelerin alt sınır ve üst sınır değerleri sırasıyla -8 ve +8 olarak seçilmiştir. Öznitelik vektör boyutu (FVS), 500 ve 1000 değerleri için analiz edilmiştir. Her bir deneyin 30 kez tekrarlanmasından elde edilen en iyi, en kötü, medyan, ortalama ve standart sapma değerleri Tablo 5.13 üzerinde gösterilmektedir. Şekil 5.2 ve Şekil 5.3, sırasıyla CSDMC2010 ve TurkishEmail veri kümeleri üzerinde ortalama ve en iyi sonuçların yakınsama grafiklerini göstermektedir.

Tablo 5.13. TurkishEmail ve CSDMC2010 veri kümeleri üzerinde DE-LR yönteminin sınıflandırma performansı

λ	TurkishEmail veri kümesi					CSDMC2010 veri kümesi				
	En iyi	En kötü	Medyan	Ort.	Std.	En iyi	En kötü	Medyan	Ort.	Std.
500	%97.88	%97.37	%97.38	%97.54	0.29	%97.26	%97.08	%97.22	%97.18	0.10
1000	%98.25	%97.25	%97.63	%97.71	0.51	%97.29	%96.75	%97.17	%97.07	0.28



Şekil 5.2. DE-LR sınıflandırma yönteminin CSDMC2010 veri kümesi üzerinde eğitim safhasındaki ortalama ve en iyi sonuçların yakınsama grafikleri



Şekil 5.3. DE-LR sınıflandırma yönteminin TurkishEmail veri kümesi üzerinde eğitim safhasındaki ortalama ve en iyi sonuçların yakınsama grafikleri

Tablo 5.14. Spambase veri kümesi üzerinde DE-LR sınıflandırma yönteminin sınıflandırma performansı

P	ACC					CC	SN	SP	PPV	NPV	F1	Süre(sn)
	En iyi	En kötü	Medyan	Ort.	Std.	En iyi						Ort.
10	90.58	63.48	86.09	84.33	6.45	80.42	89.89	**99.16**	89.42	93.15	88.25	**5.52**
20	93.26	90.43	91.81	91.76	0.79	85.84	92.28	96.17	93.79	94.90	91.27	10.98
30	93.19	91.30	92.50	92.44	**0.46**	85.69	90.63	95.81	93.22	93.91	91.25	16.38
40	93.84	91.16	92.36	92.45	0.70	87.07	92.65	95.93	93.50	95.15	92.11	23.90
50	93.77	91.16	92.54	92.51	0.59	86.92	91.18	96.29	93.97	94.33	92.02	28.50
60	93.77	**91.88**	92.68	**92.69**	0.51	86.94	91.73	96.17	93.87	94.64	92.07	34.08
70	93.77	91.59	92.68	92.65	0.53	86.91	91.73	96.17	93.76	94.58	91.96	37.52
80	93.33	91.59	92.64	92.58	0.51	86.00	91.54	95.81	93.26	94.49	91.46	45.12
90	93.62	90.94	92.61	92.47	0.68	86.63	91.36	95.93	93.45	94.42	91.87	51.06
100	93.33	91.01	92.39	92.41	0.56	86.07	92.10	96.05	93.52	94.82	91.59	56.68
110	**94.06**	91.09	92.25	92.38	0.63	**87.52**	91.36	96.29	94.08	94.33	92.32	61.68
120	93.48	90.58	92.46	92.44	0.60	86.39	92.46	96.17	93.76	95.05	91.79	65.34
130	93.99	91.45	92.28	92.39	0.63	87.44	**93.01**	95.57	93.01	**95.42**	**92.42**	71.76
140	93.99	91.30	92.43	92.52	0.71	87.37	91.36	96.29	93.95	94.44	92.26	78.10
150	93.91	91.59	**92.75**	**92.69**	0.65	87.23	91.91	96.65	94.43	94.66	92.24	82.26
160	93.41	91.45	92.54	92.57	0.49	86.19	91.54	96.17	93.73	94.50	91.63	89.82
170	93.91	90.72	92.32	92.31	0.69	87.21	91.18	96.29	94.06	94.17	92.12	95.14
180	93.77	91.16	92.50	92.52	0.63	86.92	92.10	96.65	**94.55**	94.83	91.87	99.04
190	93.33	91.38	92.64	92.57	0.55	86.00	91.18	96.65	94.33	94.28	91.43	105.10
200	93.70	91.59	92.50	92.50	0.55	86.76	91.73	96.17	93.80	94.57	91.89	109.36

DE-LR sınıflandırma yöntemi, Spambase veri kümesi üzerinde önerilen yöntemlerin sınıflandırma performanslarıyla kıyaslama yapmak için kullanılmıştır. DE-LR yöntemi kullanılarak Spambase veri kümesi üzerinde toplam 20 deney gerçekleştirilmiş ve sınıflandırma performansı üzerinde rastgele veri etkisini azaltmak için her bir deney 30 kez tekrarlanılmıştır. DE-LR sınıflandırma yönteminin mutasyon faktörü (F) parametresine 0.5 değeri atanırken geçiş olasılığı parametresine (CR) 0.8 değeri verilmiş ve 20 farklı popülasyon boyutu ile eğitilmiştir. Deneylerde maksimum döngü sayısına (MCN) 400 değeri verilmiştir. Eğitilecek parametrelerin alt sınır ve üst sınır değerleri sırasıyla -64 ve +64 olarak seçilmiştir. En iyi korelasyon katsayısı (CC), F ölçümü (F1), duyarlılık (SN), pozitif tahmin değeri (PPV), negatif tahmin değeri (NPV), özgüllük (SP) ve saniye cinsinden ortalama harcanan zaman (eğitim ve test için harcanan toplam zamanların ortalama değeri) değerlerine ek olarak her bir deneyin 30 defa koşulmasıyla

elde edilen sınıflandırma doğruluklarının (ACC) en iyi, en kötü, ortalama, medyan (ortanca) ve standart sapma değerleri Tablo 5.14 üzerinde gösterilmektedir.

5.4. Önerilen yöntemlerin sınıflandırma başarımları

5.4.1. ABC algoritması ile eğitilen LR sınıflandırıcının başarımı

ABC algoritması bazı kontrol parametrelerine sahip olduğundan dolayı bu çalışmadaki problem türü için en ideal performansı sergileyen uygun parametre aralıklarını önermek amaçlanılmaktadır [13, 64]. Çıkarılan uygun değerler çalışmanın sonraki aşamalarında kullanılmıştır.

Yiyecek kaynağı sayısı (SN), maksimum döngü sayısı (MCN), bir yiyecek kaynağının terk edilip edilmeyeceğine karar vermek için $limit$ değeri ve değiştirme oranı (MR) gibi ABC algoritmasının birkaç kontrol parametresi vardır. Standart LR sınıflandırma modelinde olduğu gibi bu sınıflandırma modelinde de en uygun parametreleri belirleyebilmek için kapsamlı grid yöntemi kullanılmıştır. Deneylerde SN parametresine {40,60,80} değerleri verilmiştir. Adil bir karşılaştırma yapabilmek için tüm deneylerde değerlendirme sayısı aynı (160.000) değere sabitlenilmiştir. Değerlendirme sayısı, koloni sayısı ile maksimum döngü sayısının çarpımına eşittir. Bu yüzden SN parametresine sırasıyla {40,60,80} değerleri verildiğinde maksimum döngü sayısına sırasıyla {2000,1333,1000} değerleri verilmiştir. MR kontrol parametresi için {0.05,0.08,0.1,0.2} değerleri incelenmiştir. $limit$ parametresi için {100,200} değerleri seçilmiştir. Öznitelik vektör boyutu (FVS) {500,1000} değerleri için analiz edilmiştir. 48 adet konfigürasyonun her biri TurkishEmail ve CSDMC2010 veri kümeleri için 30 kez tekrarlanılmıştır. Parametre sınırlarının minimum ve maksimum değerleri ($[w_j^{min}, w_j^{max}]$) sırasıyla -8 ve $+8$ olarak belirlenilmiştir. Her bir deneyin 30 kez tekrarlanmasından elde edilen en iyi, en kötü, medyan, ortalama ve standart sapma değerleri Tablo 5.15 ve Tablo 5.16'da görülmektedir. Şekil 5.4 ve Şekil 5.5, TurkishEmail ve CSDMC2010 veri kümeleri üzerinde ABC algoritmasındaki parametrelerin (SN, MR, $limit$) değerlerinin ve öznitelik vektör boyutlarının ({500,1000}) etkilerini göstermektedir.

Tablo 5.15. TurkishEmail veri kümesi üzerinde ABC algoritması ile eğitilen LR sınıflandırma istatistikleri

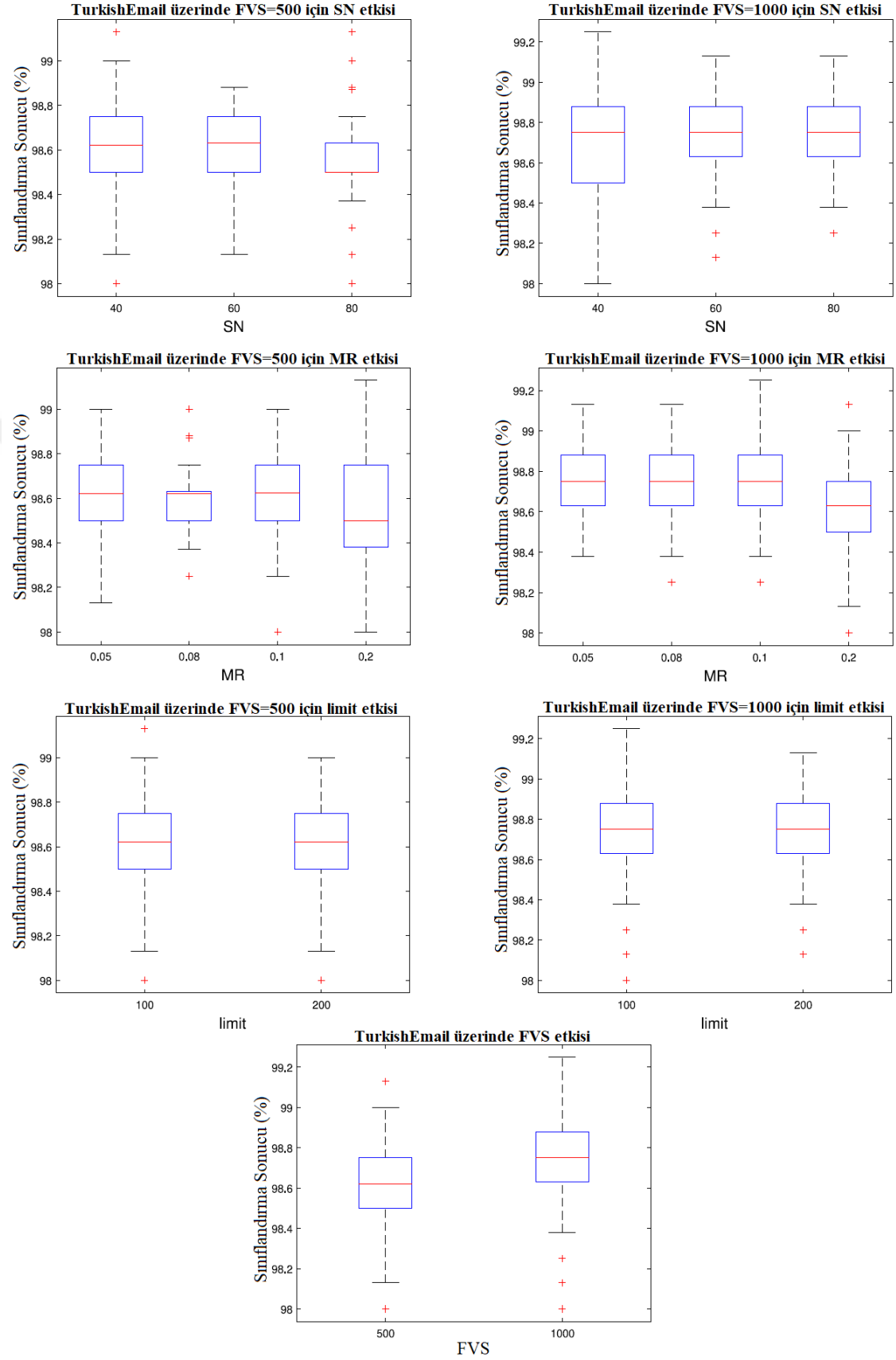
SN	MR	$limit$	$FVS = 500$					$FVS = 1000$				
			En iyi	En kötü	Medyan	Ort.	Std.	En iyi	En kötü	Medyan	Ort.	Std.
40	0.05	100	98.87	**98.38**	98.62	98.61	0.12	99.00	98.38	98.75	98.71	0.16
		200	98.87	98.37	98.62	98.60	0.17	99.13	98.38	**98.88**	**98.82**	0.20
	0.08	100	99.00	**98.38**	98.62	98.62	0.17	99.00	98.38	98.75	98.73	0.17
		200	99.00	**98.38**	98.62	98.62	0.14	99.00	98.25	98.75	98.68	0.19
	0.1	100	99.00	98.25	98.62	98.58	0.15	**99.25**	98.25	98.63	98.64	0.20
		200	98.88	98.00	98.62	98.58	0.20	99.13	98.50	98.81	98.76	0.20
	0.2	100	**99.13**	98.37	98.62	98.61	0.20	99.00	98.00	98.63	98.59	0.22
		200	98.88	98.13	**98.63**	98.57	0.21	99.00	98.13	98.56	98.56	0.25
60	0.05	100	98.87	98.25	**98.63**	98.61	0.19	99.00	98.50	**98.88**	**98.82**	**0.13**
		200	98.87	98.13	98.62	98.59	0.18	99.00	98.50	98.75	98.77	0.16
	0.08	100	98.87	98.37	98.62	98.60	0.13	99.13	98.25	98.75	98.73	0.19
		200	98.75	**98.38**	**98.63**	98.60	**0.11**	99.13	98.50	98.75	98.79	0.14
	0.1	100	98.88	**98.38**	**98.63**	**98.64**	0.16	99.00	98.38	98.75	98.75	0.16
		200	98.87	**98.38**	**98.63**	**98.64**	0.12	99.00	98.38	98.75	98.76	0.18
	0.2	100	98.75	98.13	98.50	98.48	0.19	99.13	98.13	98.63	98.64	0.22
		200	98.87	98.25	**98.63**	98.59	0.16	99.13	98.25	98.69	98.67	0.21
80	0.05	100	98.87	98.25	98.50	98.54	0.15	99.00	98.50	**98.88**	**98.82**	**0.13**
		200	99.00	98.25	98.50	98.57	0.19	99.00	98.50	98.75	98.78	0.14
	0.08	100	98.87	98.25	98.62	98.61	0.15	99.13	**98.63**	**98.88**	98.81	**0.13**
		200	98.88	**98.38**	98.50	98.56	0.14	99.00	98.50	**98.88**	98.83	0.14
	0.1	100	98.75	98.25	**98.63**	98.60	0.13	99.13	98.38	**98.88**	98.78	0.17
		200	98.88	98.25	98.50	98.56	0.15	99.00	98.38	98.75	98.77	0.16
	0.2	100	**99.13**	98.00	98.50	98.52	0.23	99.00	98.38	98.75	98.67	0.16
		200	99.00	**98.38**	98.50	98.60	0.16	99.00	98.25	98.75	98.69	0.20

Tablo 5.15 ve Tablo 5.16 ile Şekil 5.4 ve Şekil 5.5'e bakıldığında en iyi değerlerin beklenildiği üzere öznitelik vektör boyutu arttıkça elde edildiği görülmektedir. Fakat TurkishEmail veri kümesi ile kıyaslandığında CSDMC2010 veri kümesi üzerinde öznitelik vektör boyutu daha yüksek etkiye sahiptir. Bu farklılık CSDMC2010 veri kümesinin TurkishEmail veri kümesine göre çok daha fazla sayıda benzersiz terim içermesine ek olarak daha karmaşık ve lineer olmayan bir yapıda olmasından kaynaklanıyor olabilir. Tüm deneylerde değerlendirme sayısı aynı değere sabitlendiğinden dolayı farklı SN değerleri benzer sonuçlar üretmektedir. Fakat Şekil 5.4 ve Şekil 5.5'de görüldüğü üzere SN parametresi 60 değerini aldığında hem TurkishEmail

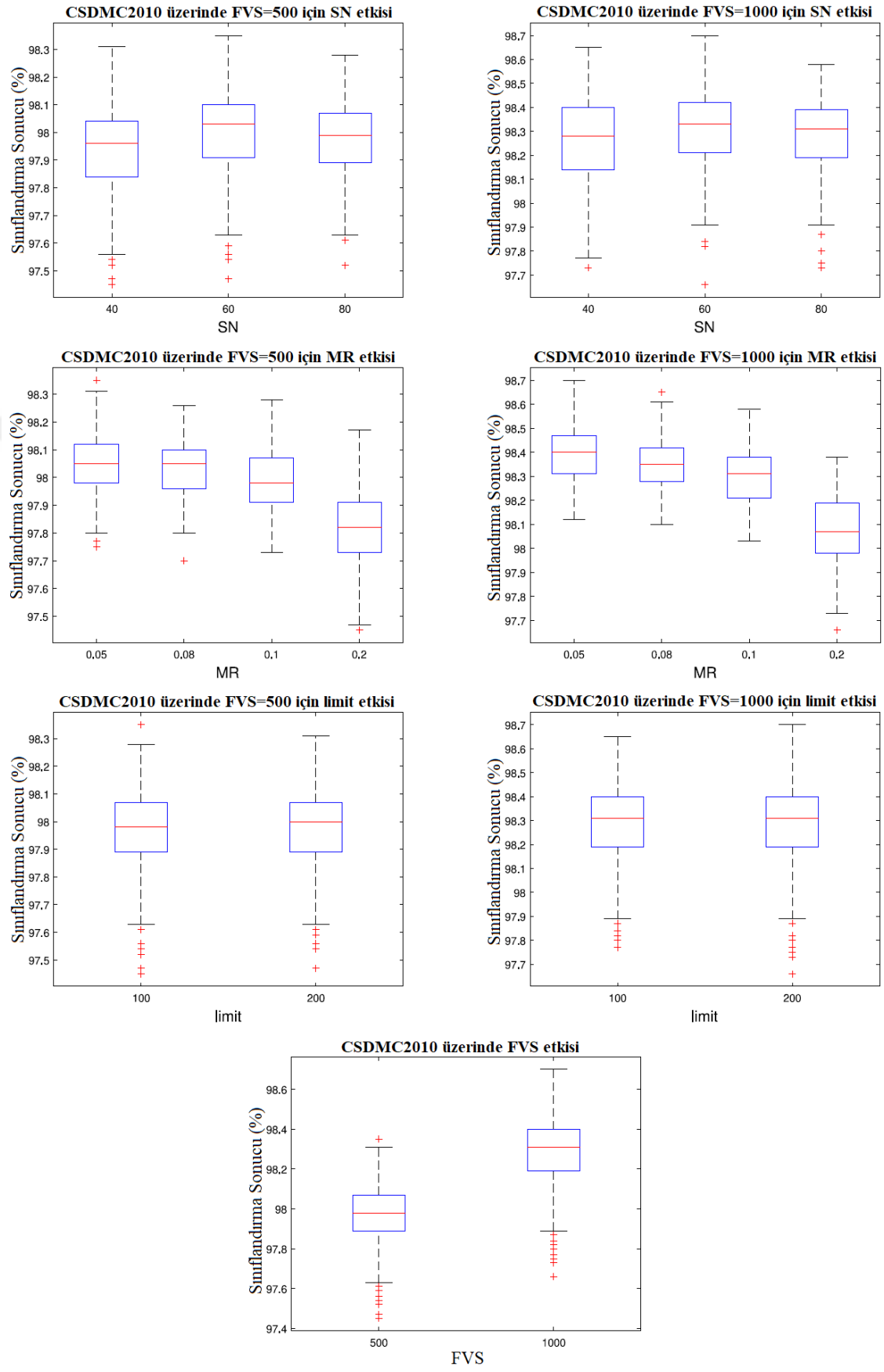
hem de CSDMC2010 veri kümeleri üzerinde ABC-LR algoritması biraz daha iyi ve kararlı sonuçlar üretmektedir. TurkishEmail veri kümesi üzerinde MR değişkeni $[0.05 - 0.1]$ aralığında değerler aldığında daha kararlı sonuçlar üretilmesine rağmen en iyi sonuçlar öznitelik vektör boyutu 500 için MR değişkeni 0.2 değerine eşit olduğunda elde edilirken öznitelik vektör boyutu 1000 için MR değişkeni 0.1 değerine eşit olduğunda elde edilmiştir. CSDMC2010 veri kümesi üzerinde ise MR değişkeni 0.05 değerini aldığında daha etkili ve kararlı sonuçlar üretilmiştir. Şekil 5.4 ve Şekil 5.5'den görüldüğü üzere $limit$ değeri 100 ve 200 olduğunda ABC-LR sınıflandırma yöntemi benzer sonuçlar üretmiştir.

Tablo 5.16. CSDMC2010 veri kümesi üzerinde ABC algoritması ile eğitilen LR sınıflandırma istatistikleri

SN	MR	$limit$	$FVS = 500$					$FVS = 1000$				
			En iyi	En kötü	Medyan	Ort.	Std.	En iyi	En kötü	Medyan	Ort.	Std.
40	0.05	100	98.19	97.80	98.03	98.03	0.11	98.56	98.19	98.39	98.39	**0.09**
		200	98.31	97.77	98.03	98.02	0.12	98.65	98.21	**98.45**	**98.43**	0.10
	0.08	100	98.17	97.87	97.99	98.00	**0.07**	98.65	98.14	98.30	98.32	0.11
		200	98.19	97.82	98.03	98.02	0.09	98.49	98.10	98.34	98.33	0.11
	0.1	100	98.19	97.73	97.93	97.94	0.11	98.52	98.03	98.27	98.26	0.13
		200	98.17	97.75	97.94	97.95	0.10	98.54	98.03	98.24	98.25	0.12
	0.2	100	98.05	97.45	97.75	97.74	0.14	98.35	97.77	98.04	98.04	0.14
		200	98.00	97.61	97.75	97.76	0.10	98.33	97.73	98.06	98.04	0.16
60	0.05	100	**98.35**	**97.96**	98.09	**98.10**	0.10	98.61	**98.24**	**98.45**	98.42	0.10
		200	98.21	97.91	**98.10**	98.08	**0.07**	**98.70**	**98.24**	**98.45**	**98.43**	0.11
	0.08	100	98.21	97.82	98.07	98.04	0.10	98.54	98.14	98.41	98.38	0.11
		200	98.26	97.91	98.06	98.09	0.08	98.61	98.19	98.37	98.37	**0.09**
	0.1	100	98.19	97.82	97.99	98.01	0.10	98.56	98.14	98.31	98.31	0.11
		200	98.26	97.87	98.05	98.04	0.11	98.58	98.12	98.32	98.34	0.12
	0.2	100	98.17	97.56	97.85	97.84	0.14	98.38	97.82	98.11	98.11	0.15
		200	98.12	97.47	97.82	97.82	0.15	98.38	97.66	98.12	98.10	0.15
80	0.05	100	98.17	97.75	98.03	98.04	0.09	98.58	98.21	98.33	98.34	**0.09**
		200	98.17	97.87	98.02	98.01	0.08	98.49	98.12	98.35	98.33	**0.09**
	0.08	100	98.24	97.80	98.03	98.02	0.11	98.56	98.14	98.38	98.37	0.11
		200	98.24	97.70	98.04	98.03	0.12	98.56	98.10	98.39	98.36	0.11
	0.1	100	98.28	97.82	98.04	98.01	0.13	98.54	98.19	98.34	98.33	**0.09**
		200	98.21	97.84	97.99	97.99	0.10	98.54	98.12	98.31	98.32	0.11
	0.2	100	98.07	97.52	97.83	97.83	0.15	98.31	97.87	98.10	98.09	0.11
		200	98.07	97.63	97.88	97.88	0.11	98.31	97.73	98.07	98.06	0.14



Şekil 5.4. TurkishEmail veri kümesi üzerinde öznitelik sayısının ve parametrelerin etkisi



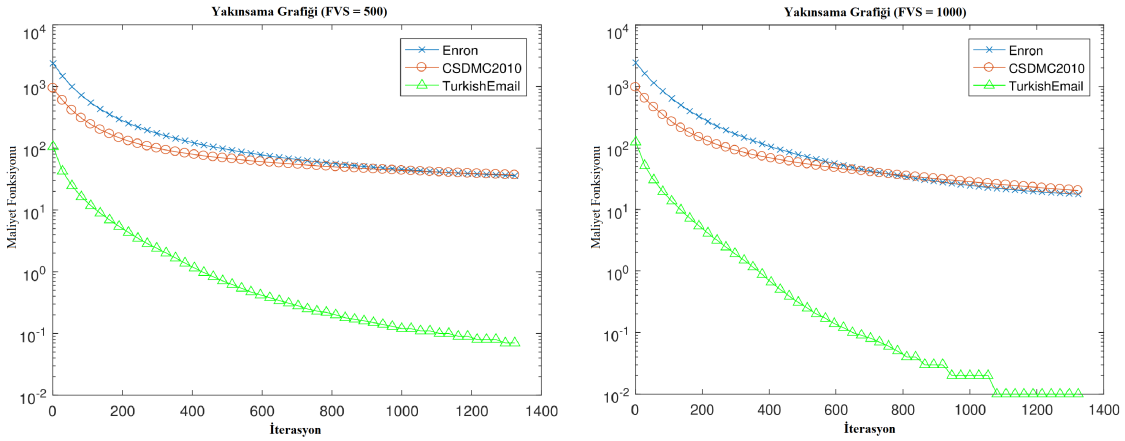
Şekil 5.5. CSDMC2010 veri kümesi üzerinde öznitelik sayısının ve parametrelerin etkisi

ABC-LR yönteminin sınıflandırma performansını önceki çalışmalarla kıyaslayabilmek için bu yöntem Enron-1 veri kümesine uygulanılmış ve üç farklı SN değeri ({40, 60, 80}), iki farklı MR değeri ({0.05, 0.1}) ve $limit$ parametresi 100 değeriyle eğitilmiştir. Değerlendirme sayısını 160.000 olarak sabitlemek için maksimum döngü sayısına {2000, 1333, 1000} değerleri verilmiştir. Ağırlıkların alt sınırı (w_j^{min}) ve üst sınırı üst sınırı (w_j^{max}) sırasıyla -64 ve $+64$ olarak ayarlanılmıştır. Toplam 12 deney gerçekleştirilmiştir. Her bir deneyin en iyi, en kötü, medyan, ortalama ve standart sapma değerleri Tablo 5.17'de görülmektedir.

Tablo 5.17. Enron-1 veri kümesi üzerinde ABC algoritması ile eğitilen LR'nin sınıflandırma istatistikleri

SN	MR	$FVS = 500$					$FVS = 1000$				
		En iyi	En kötü	Medyan	Ort.	Std.	En iyi	En kötü	Medyan	Ort.	Std.
40	0.05	%98.45	%98.02	**%98.21**	%98.22	0.11	%98.81	%98.42	%98.49	%98.55	0.11
	0.1	%98.24	%97.87	%98.07	%98.06	0.10	%98.53	%98.05	%98.24	%98.23	0.10
60	0.05	**%98.47**	**%98.05**	**%98.21**	**%98.24**	0.11	**%98.91**	**%98.49**	**%98.65**	**%98.66**	0.10
	0.1	%98.30	%97.87	%98.09	%98.09	0.11	%98.58	%98.12	%98.40	%98.40	0.12
80	0.05	%98.30	%97.97	%98.17	%98.15	0.10	%98.75	%98.47	%98.60	%98.61	**0.07**
	0.1	%98.20	%98.01	%98.12	%98.11	**0.05**	%98.62	%98.32	%98.47	%98.46	0.08

SN değişkeni 60 değerini, MR değişkeni 0.5 değerini ve $limit$ değişkeni 100 değerini aldığında Enron, CSDMC2010 ve TurkishEmail veri kümeleri üzerinde LR yönteminin eğitimi aşamasında ABC algoritmasının yakınsama grafikleri Şekil 5.6'da görülmektedir. Şekilde görüldüğü üzere ABC-LR yöntemi yerel minimuma takılmaksızın eğitim işlemine devam etmektedir.



Şekil 5.6. Enron, CSDMC2010 ve TurkishEmail veri kümeleri için eğitim aşamasında ABC-LR yönteminin yakınsama grafikleri

5.4.1.1. ABC algoritması üzerinde kontrol parametrelerinin etkileri

TurkishEmail ve CSDMC2010 veri kümeleri kullanılarak elde edilen sonuçlar üzerinde MR parametreleri arasındaki farkın istatistiksel olarak anlamlı olup olmadığını keşfetmek için ANOVA ile post-hoc testi uygulanılmıştır. MR değişkenleri dört farklı grup $(0.05, 0.08, 0.1, 0.2)$ olduğundan dolayı hangi çiftlerin ortalamaları arasındaki farkın önemli olduğunu belirlemek için post-hoc testine ihtiyaç duyulmuştur. Tablo 5.18 üzerinden TurkishEmail veri kümesi için sınıflandırma doğruluklarının grup ortalamaları arasında istatistiksel olarak önemli bir farkın $(Sig = 0.748 > 0.05)$ olmadığı görülürken CSDMC2010 veri kümesi için grup ortalamaları arasında istatistiksel olarak anlamlı bir farkın $(Sig. = 0.0 < 0.05)$ olduğu görülmektedir. Gruplar arası kareler toplamı, her bir grubun ortalamasının genel ortalamadan çıkarılmasıyla elde edilen değerlerin karesinin alınması ve bu değerlerin toplanarak serbestlik derecesine bölünmesiyle elde edilmektedir. Gruplar içi kareler toplamı ise her bir grup ve gruplardaki tüm elemanların değerinden dahil oldukları grubun ortalaması çıkartılıp karesi alındıktan sonra bu kareler toplanarak elde edilmektedir.

Tablo 5.18. MR parametrelerinin ANOVA test sonucu

Veri Kümesi		Karelerin top.	Karelerin ort.	$Sig.$
TurkishEmail	Gruplar arasında	0.000	0.000	0.748
	Gruplar içinde	0.000	0.000	
	Toplam	0.000		
CSDMC2010	Gruplar arasında	0.000	0.000	**0.000**
	Gruplar içinde	0.000	0.000	
	Toplam	0.000		

Tablo 5.19'dan görüldüğü üzere CSDMC2010 veri kümesi üzerinde $0.05-0.1$, $0.05-0.2$ ve $0.08-0.2$ çiftlerinin ortalamaları arasında istatistiksel olarak anlamlı farklılıklar varken $0.05 - 0.08$ ve $0.08 - 0.1$ çiftlerinin ortalamaları arasında anlamlı bir farklılık yoktur. Bu tabloda her grubun ikişerli karşılaştırmaları yapılmış ve bu karşılaştırılan grupların ortalamaları arasındaki farklar (Ortalama Farkı) sayısal olarak verilmiştir. Bu sayısal değerlerin yanında bir yıldız (*) işaretinin bulunması bu ikilinin ortalamaları arasında anlamlı bir farklılık olduğunu göstermektedir. Tablo incelendiğinde $0.05-0.1$, $0.05-0.2$, $0.08 - 0.2$ ve $0.1 - 0.2$ ikililerinin yanında bir yıldız (*) işareti olduğu görülür. Yani

bu ikililerin ortalamaları arasında anlamlı bir farklılık vardır. TurkishEmail veri kümesi üzerinde ise MR parametre çiftlerinin arasında anlamlı bir farklılık olmadığı yine Tablo 5.19'dan görülmektedir.

Tablo 5.19. MR parametrelerinin post-hoc test sonucu

Bağımlı değişken	(I)MR	(J)MR	Ort. Farkı (I–J)	$Sig.$
TurkishEmail Tukey HSD	0.05	0.08	-0.0002200	0.950
		0.10	0.0002333	0.941
		0.20	-0.0000167	1.000
	0.08	0.10	0.0004533	0.688
		0.20	0.0002033	0.960
	0.10	0.20	-0.0002500	0.929
CSDMC2010 Tukey HSD	0.05	0.08	0.0002867	0.746
		0.10	0.0008733*	**0.014**
		0.20	0.0028967*	**0.000**
	0.08	0.10	0.0005867	0.172
		0.20	0.0026100*	**0.000**
	0.10	0.20	0.0020233*	**0.000**
*. Ortalama farkı 0.05 düzeyinde anlamlıdır.				

İki grup FVS parametresi (500, 1000) olduğundan dolayı FVS parametreleri arasındaki farklılıkların istatistiksel olarak anlamlı olup olmadığını ölçmek için sadece ANOVA testi gerçekleştirilmiştir. Tablo 5.20'den görüldüğü üzere hem TurkishEmail hem de CSDMC2010 veri kümeleri üzerinde sınıflandırma doğruluklarının grup ortalamaları arasında istatistiksel olarak anlamlı farklılıklar bulunmaktadır.

Tablo 5.20. Öznitelik vektör boyutu (FVS) parametrelerinin ANOVA test sonucu

Veri Kümesi		Karelerin top.	Karelerin ort.	$Sig.$
TurkishEmail	Gruplar arasında	0.000	0.000	**0.035**
	Gruplar içinde	0.000	0.000	
	Toplam	0.000		
CSDMC2010	Gruplar arasında	0.000	0.000	**0.000**
	Gruplar içinde	0.000	0.000	
	Toplam	0.000		

İki grup $limit$ parametresi (100, 200) olduğundan dolayı $limit$ parametreleri arasındaki farklılıkların istatistiksel olarak anlamlı olup olmadığını görmek için sadece ANOVA testi uygulanılmıştır. Tablo 5.21'den görüldüğü üzere TurkishEmail ve CSDMC2010 veri

kümeleri üzerinde sınıflandırma doğruluklarının grup ortalamaları arasında istatistiksel olarak anlamlı farklılıklar bulunmamaktadır.

Tablo 5.21. limit parametrelerinin ANOVA test sonucu

Veri Kümesi		Karelerin top.	Karelerin ort.	$Sig.$
TurkishEmail	Gruplar arasında	0.000	0.000	0.913
	Gruplar içinde	0.000	0.000	
	Toplam	0.000		
CSDMC2010	Gruplar arasında	0.000	0.000	0.354
	Gruplar içinde	0.000	0.000	
	Toplam	0.000		

İki gruptan fazla SN/MCN parametresi olduğundan dolayı SN/MCN parametreleri arasındaki farklılıkların istatistiksel olarak önemli olup olmadığını görebilmek için ANOVA ile Post-Hoc testi gerçekleştirilmiştir. Tablo 5.22 ve Tablo 5.23'den görüldüğü üzere SN/MCN parametreleri için hem TurkishEmail hem de CSDMC2010 veri kümeleri üzerinde sınıflandırma doğruluklarının grup ortalamaları arasında istatistiksel olarak anlamlı farklılıklar bulunmamaktadır.

Tablo 5.22. SN/MCN parametrelerinin ANOVA test sonuçları

Veri Kümesi		Karelerin top.	Karelerin ort.	$Sig.$
TurkishEmail	Gruplar arasında	0.000	0.000	0.799
	Gruplar içinde	0.000	0.000	
	Toplam	0.000		
CSDMC2010	Gruplar arasında	0.000	0.000	0.276
	Gruplar içinde	0.000	0.000	
	Toplam	0.000		

Tablo 5.23. SN/MCN parametrelerinin Post-Hoc test sonuçları

Bağımlı değişken	(I)MR	(J)MR	Ort. Farkı (I–J)	$Sig.$
TurkishEmail Tukey HSD	40	60	0.0002533	0.789
		80	0.0001733	0.895
	60	80	-0.0000800	0.977
CSDMC2010 Tukey HSD	40	60	-0.0004000	0.248
		80	-0.0002400	0.602
	60	80	0.0001600	0.797

5.4.2. CSA-LR sınıflandırıcının başarımı

Yapay bağışıklık sistemi algoritmalarını kullanan güncel ve literatürde bilinen spam filtreleme çalışmaları [17, 46] deneylerini Spambase veri kümesi üzerinde gerçekleştirdiğinden dolayı bu tez çalışmasında önerilen ve bir yapay bağışıklık sistemi algoritması olan CSA ile geliştirilen CSA-LR spam filtreleme yöntemi Spambase veri kümesine uygulanmıştır. Önerilen yöntemin aynı veri kümesine uygulanılmasındaki amaç, literatürde bilinen yapay bağışıklık sistemi algoritmalarına dayandırılan çalışmalardaki yöntemlerle önerilen yöntemi kıyaslayabilmektir. CSA-LR sınıflandırma yöntemi, [17, 46] çalışmalarında olduğu gibi Spambase veri kümesi katmanlı bir örnekleme yaklaşımıyla %70 eğitim kümesine ve %30 test kümesine bölünerek değerlendirilmiştir. Katmanlı örnekleme yaklaşımı, veri kümesinin farklı alt gruplara bölündüğü ve finalde oluşacak grupların farklı alt gruplardan rastgele bir şekilde seçildiği bir tür olasılıksal örnekleme tekniğidir.

Tablo 5.24. Spambase veri kümesi üzerinde CSA-LR sınıflandırma yönteminin sınıflandırma performansı

P	α	ACC					CC	SN	SP	PPV	NPV	F1	Süre(sn)
		En iyi	En kötü	Medyan	Ort.	Std.	En iyi						Avg
10	1	92.46	85.94	90.40	90.08	1.52	84.17	91.18	95.81	93.01	94.00	90.32	**11.12**
	2	92.39	89.78	91.41	91.23	0.72	84.02	90.07	96.17	93.44	93.54	90.25	21.76
	3	92.61	90.29	91.81	91.72	0.65	84.50	90.63	96.05	93.35	93.86	90.57	32.52
	4	93.84	90.65	92.10	92.08	0.75	87.07	91.73	**96.53**	**94.39**	94.62	91.99	43.60
	5	93.55	90.87	91.81	91.94	0.59	86.47	91.18	95.69	92.98	94.31	91.77	55.02
20	1	92.83	89.35	90.65	90.75	0.87	85.02	91.54	95.22	91.77	94.45	90.96	24.22
	2	93.41	89.86	91.85	91.75	0.71	86.15	90.44	95.93	93.20	93.88	91.53	43.30
	3	93.04	90.80	91.67	91.84	**0.56**	85.39	92.28	96.29	93.92	94.83	91.08	68.22
	4	93.55	90.72	91.88	91.96	0.65	86.47	91.18	95.33	92.56	94.20	91.75	88.84
	5	**94.20**	90.51	92.10	**92.19**	0.99	**87.87**	**92.83**	96.17	93.83	**95.32**	**92.66**	110.54
30	1	93.48	89.42	90.65	90.89	0.99	86.34	91.54	95.10	92.12	94.51	91.71	32.34
	2	92.83	90.14	91.63	91.67	0.70	84.93	90.26	95.22	92.14	93.69	90.79	65.58
	3	93.04	89.49	91.74	91.75	0.90	85.39	90.99	95.22	92.11	94.13	91.08	98.36
	4	93.41	90.65	91.99	92.04	0.71	86.17	91.18	95.45	92.63	94.19	91.58	131.6
	5	93.41	90.65	91.92	92.01	0.67	86.15	90.81	95.45	92.82	94.04	91.52	165.26
40	1	93.62	88.91	91.16	91.19	0.96	86.61	90.99	95.57	93.02	94.01	91.81	43.00
	2	93.33	90.72	91.74	91.93	0.68	86.12	**92.83**	95.45	92.75	95.26	91.65	87.52
	3	93.84	90.51	**92.21**	92.12	0.71	87.09	91.91	95.69	92.97	94.76	92.17	131.44
	4	93.62	**91.01**	91.81	92.00	0.64	86.65	91.91	96.29	93.90	94.74	91.91	175.6
	5	93.33	90.94	91.88	91.92	0.63	86.00	90.99	95.81	93.12	94.15	91.42	228.48

CSA-LR dört farklı antikor sayısı ($P = \{10, 20, 30, 40\}$), beş farklı klonal çarpım faktörü ($\alpha = \{1, 2, 3, 4, 5\}$) ile değiştirme oranına 0.1 değeri ($B = 0.1$) ve maksimum döngü sayısına 400 değeri ($MCN = 400$) atanarak eğitilmiştir. Li ve arkadaşları çalışmalarında [25] klonal çarpım faktörü α parametresine 7'den daha küçük bir değerin atanmasını önerdiğinden dolayı α değeri 1 ile 5 arasından seçilmiştir. Toplam 20 deney gerçekleştirilmiş ve sınıflandırma performansı üzerinde rastgele veri etkisini azaltmak için her bir deney 30 kez tekrarlanmıştır. En iyi korelasyon katsayısı (CC), F ölçümü (F1), duyarlılık (SN), pozitif tahmin değeri (PPV), negatif tahmin değeri (NPV), özgüllük (SP) ve eğitim ve test aşamasında saniye cinsinden ortalama harcanan zaman değerlerine ek olarak her bir deneyin 30 defa koşulmasıyla elde edilen sınıflandırma doğruluklarının (ACC) en iyi, en kötü, ortalama, medyan (ortanca) ve standart sapma değerleri Tablo 5.24 üzerinde görülmektedir.

5.4.3. ABC-CSA-LR sınıflandırıcının başarımı

ABC-CSA-LR sınıflandırma yöntemi kullanılarak Spambase veri kümesi üzerinde toplam 20 deney gerçekleştirilmiş ve sınıflandırma performansı üzerinde rastgele veri etkisini azaltmak için her bir deney 30 kez tekrarlanılmıştır. Deneylerde maksimum döngü sayısına (MCN) 400 değeri verilirken değiştirme oranı (B) parametresine 0.1 değeri verilmiştir. ABC-CSA-LR sınıflandırma yöntemi dört farklı antikor sayısı ($P = \{10, 20, 30, 40\}$), beş farklı klonal çarpım faktörü ($\alpha = \{1, 2, 3, 4, 5\}$) ve bir ABC algoritması parametresi olan MR kontrol parametresine 0.4 değeri atanarak eğitilmiştir. En iyi korelasyon katsayısı (CC), F ölçümü (F1), duyarlılık (SN), pozitif tahmin değeri (PPV), negatif tahmin değeri (NPV), özgüllük (SP) ve saniye cinsinden ortalama harcanan zaman (eğitim ve test için gerekli toplam zaman) değerlerine ek olarak her bir deneyin 30 defa koşulmasıyla elde edilen sınıflandırma doğruluklarının (ACC) en iyi, en kötü, ortalama, medyan (ortanca) ve standart sapma değerleri Tablo 5.25 üzerinde gösterilmektedir.

ABC-CSA-LR sınıflandırma yöntemi Spambase veri kümesine ek olarak spam filtreleme alanında bir kıyaslama veri kümesi olarak literatürde oldukça kullanılan Enron-1 veri kümesine uygulanılmıştır. Enron-1 veri kümesi 1500 (%29) adet spam e-posta ve 3672 (%71) adet normal e-postadan oluşmaktadır. Normal e-postalar ile spam e-postaların

Tablo 5.25. Spambase veri kümesi üzerinde ABC-CSA-LR sınıflandırma yönteminin sınıflandırma performansı

P	α	ACC					CC	SN	SP	PPV	NPV	F1	Süre(sn)
		En iyi	En kötü	Medyan	Ort.	Std.	En iyi						Ort.
10	1	93.19	91.01	92.32	92.24	0.6	85.71	90.81	95.81	93.19	94.06	91.31	**7.30**
	2	93.99	90.80	92.57	92.45	0.76	87.42	92.65	96.65	94.30	95.20	92.39	12.92
	3	93.84	90.94	92.43	92.52	0.61	87.07	91.36	96.05	93.64	94.44	92.12	18.62
	4	**94.49**	91.09	92.39	92.38	0.69	**88.33**	**93.20**	96.05	93.54	**95.56**	**92.94**	23.94
	5	93.41	90.94	92.46	92.42	0.57	86.16	91.54	95.81	93.27	94.49	91.57	30.18
20	1	93.55	90.80	92.39	92.39	0.64	86.47	92.10	96.17	93.79	94.83	91.77	13.28
	2	94.35	91.52	92.46	92.49	0.61	88.13	92.83	96.17	93.83	95.26	92.75	24.64
	3	93.41	**91.67**	92.50	92.56	0.54	86.14	92.46	95.93	93.47	95.04	91.62	36.40
	4	94.06	91.59	92.61	92.51	0.59	87.52	91.91	96.05	93.58	94.72	92.36	48.36
	5	94.06	91.45	**92.75**	92.71	0.72	87.53	91.73	96.41	94.17	94.67	92.41	59.48
30	1	93.84	91.23	92.54	92.50	0.61	87.09	92.28	96.17	93.59	94.88	92.17	19.22
	2	93.77	91.59	92.72	92.65	**0.50**	86.94	91.91	95.45	92.83	94.75	92.08	37.12
	3	93.77	91.09	92.50	92.49	0.71	86.94	91.91	96.53	94.29	94.75	92.08	54.12
	4	94.06	91.52	92.57	92.56	0.56	87.53	91.91	95.81	93.42	94.66	92.38	71.16
	5	93.77	91.23	92.72	92.68	0.61	86.94	91.91	**96.77**	**94.69**	94.75	92.08	88.20
40	1	93.55	91.30	92.46	92.48	0.53	86.45	91.18	96.29	94.00	94.28	91.61	25.28
	2	94.35	91.16	92.83	**92.72**	0.67	88.13	92.46	96.17	93.96	95.05	92.74	48.28
	3	93.55	91.38	92.64	92.54	0.63	86.46	91.73	96.53	94.35	94.60	91.70	71.74
	4	94.06	91.45	92.64	92.65	0.63	87.52	92.10	96.29	94.02	94.79	92.35	97.22
	5	93.62	91.09	92.64	92.64	0.62	86.62	90.99	96.17	93.77	94.21	91.84	118.04

sayıları arasında bir denge olmadığı için dengesiz bir veri kümesidir. Dengesiz bir veri kümesinde parametrelerin öğrenilmesi aşamasında veri sayısı fazla olan sınıfa doğru eğilim oluşmaktadır ve bu durum sınıflandırma başarısını düşürebilmektedir. Bu yüzden dengesizlik probleminin üstesinden gelmek sınıflandırma başarısının yükselmesi açısından önemlidir. Bu tez çalışmasında dengesizlik probleminin üstesinden gelmek için aşırı-örnekleme (over-sampling) yöntemi kullanılmıştır. Bu yöntemde az sayıda örnek içeren sınıf ile çok sayıda örnek içeren sınıftaki örnek sayısını eşitlemek için az sayıda örnek içeren sınıftan rasgele seçilen örnekler tekrarlanır. Aşırı örnekleme yöntemi uygulanıldıktan sonra veri kümesinin boyutu 7344$\times$47939 olmuştur. Bu boyutta yer alan 7344 sayısı toplam e-posta sayısını ifade ederken 47939 sayısı ise veri kümesinde yer alan toplam benzersiz terim sayısını ifade etmektedir. ABC-CSA-LR sınıflandırma yöntemi üç farklı antikor sayısı ($P = \{40, 60, 80\}$), iki farklı öznitelik vektör boyutu

($FVS = \{500, 1000\}$), üç farklı klonal çarpım faktörü ($\alpha = \{1, 2, 3\}$), değiştirme oranı (B) parametresine 0.1 değeri ve maksimum döngü sayısına 1000 değeri atanarak eğitilmiştir.

Enron-1 veri kümesi üzerinde toplam 18 deney gerçekleştirilmiş ve sınıflandırma performansı üzerinde rastgele veri etkisini azaltmak için her bir deney 20 kez tekrarlanılmıştır. Yanlış negatif oranının (FN) ve yanlış pozitif oranının (FP) en iyi değerleriyle saniye cinsinden ortalama harcanan eğitim zamanı değerlerine ek olarak her bir deneyin 20 defa koşulmasıyla elde edilen sınıflandırma doğruluklarının (ACC) en iyi, en kötü, ortalama, medyan (ortanca) ve standart sapma değerleri Tablo 5.26 üzerinde verilmektedir. ABC-CSA-LR sınıflandırma yöntemi Enron veri kümesi üzerinde önemli bir başarı göstermiştir. Tablo 5.26'dan görüldüğü üzere tüm deneyler için 20 koşmanın standart sapma değerleri oldukça düşüktür. Deneysel sonuçlar, ABC-CSA-LR'nin spam filtreleme görevi için verimli, güçlü ve güvenilir bir sınıflandırma modeli olduğunu göstermektedir.

Tablo 5.26. Enron-1 veri kümesi üzerinde ABC-CSA-LR'nin performans ölçümleri

P	FVS	α	ACC					FN	FP	Süre(sn)
			En iyi	En kötü	Medyan	Ort.	Std.	En iyi		Ort.
40	500	1	%98.15	%97.86	%98.02	%98.02	0.0868	0.0038	0.0319	**27.38**
		2	%98.39	%98.07	%98.22	%98.21	0.0912	0.0019	0.0297	31.49
		3	%98.47	%98.08	%98.27	%98.27	0.0933	0.0014	0.0283	35.63
	1000	1	%98.60	%98.30	%98.46	%98.44	0.1073	0.0014	0.0256	49.09
		2	%98.71	%98.47	%98.60	%98.59	0.0677	**0.0003**	0.0251	54.16
		3	%98.75	%98.58	%98.71	%98.69	0.0649	0.0005	0.0245	58.81
60	500	1	%98.24	%97.97	%98.09	%98.09	0.0905	0.0025	0.0319	31.06
		2	%98.39	%98.23	%98.30	%98.30	0.0591	0.0019	0.0297	36.42
		3	%98.56	%98.15	%98.31	%98.32	0.0879	0.0011	0.0267	42.44
	1000	1	%98.69	%98.47	%98.56	%98.57	0.0661	0.0008	0.0245	54.32
		2	%98.83	%98.57	%98.67	%98.69	0.0979	0.0005	0.0226	60.02
		3	**%98.91**	%98.54	%98.70	%98.69	0.0854	**0.0003**	**0.0213**	67.27
80	500	1	%98.30	%98.11	%98.19	%98.19	**0.0537**	0.0025	0.0308	34.79
		2	%98.51	%98.23	%98.30	%98.34	0.0914	0.0011	0.0281	42.82
		3	%98.49	%98.26	%98.34	%98.35	0.0600	0.0011	0.0283	50.71
	1000	1	%98.77	%98.60	%98.69	%98.69	0.0691	0.0008	0.0226	55.21
		2	%98.90	%98.58	%98.71	%98.71	0.0864	**0.0003**	0.0215	69.11
		3	%98.88	**%98.62**	**%98.77**	**%98.75**	0.0888	**0.0003**	0.0215	78.96

5.5. Sınıflandırıcıların Karşılaştırılması

Tablo 5.27, öznitelik vektör boyutları 500 ve 1000 için TurkishEmail ve CSDMC2010 veri kümeleri üzerinde yedi farklı sınıflandırıcıdan elde edilen en iyi ve en kötü sınıflandırma doğruluklarını göstermektedir. Tablo 5.27'de görüldüğü üzere ABC algoritmasını kullanan LR sınıflandırıcı (ABC-LR) TurkishEmail ve CSDMC2010 veri kümeleri üzerinde sırasıyla %99.25 ve %98.70 başarı oranlarıyla en iyi sonuçları elde etmiştir.

Tablo 5.27. TurkishEmail ve CSDMC2010 veri kümeleri için en iyi ve en kötü sınıflandırma doğruluklarının karşılaştırılması

Sınıflandırma Yöntemi	TurkishEmail Veri Kümesi				CSDMC2010 Veri Kümesi			
	FVS=500		FVS=1000		FVS=500		FVS=1000	
	Best	Worst	Best	Worst	Best	Worst	Best	Worst
Gaussian NB	%95.38	%95.38	%96.63	%96.63	%93.71	%93.71	%95.92	%95.92
Multinomial NB	%96.88	%95.63	%97.62	%97.12	%90.93	%88.89	%94.55	%92.02
Linear SVM	%98.63	%93.00	%98.88	%93.75	%98.10	%92.30	%98.42	%94.08
RBF Kernel SVM	%98.62	%53.75	%98.75	%54.00	**%98.58**	%68.21	%98.63	%68.21
LR (Gradient Descent)	%98.50	%90.25	%98.75	%91.38	%98.26	%73.85	%98.56	%66.75
DE-LR	%97.88	%97.37	%98.25	%97.25	%97.26	%97.08	%97.29	%96.75
ABC-LR*	**%99.13**	**%98.00**	**%99.25**	**%98.13**	%98.35	**%97.45**	**%98.70**	**%97.66**
*. Bu tez çalışmasında önerilen yöntem								

Bu tez çalışmasında önerilen ABC-LR ve ABC-CSA-LR yöntemlerinin sınıflandırma performansları ile Barushka ve arkadaşları [1] tarafından yapılan bir çalışmada yer alan en güncel spam filtreleme yöntemlerinin sınıflandırma performansları karşılaştırılmıştır. Enron-1 veri kümesi üzerinde önceki çalışmalar tarafından rapor edilen en güncel spam filtreleme yöntemlerinin sınıflandırma performanslarıyla ABC-LR ve ABC-CSA-LR yöntemlerinin sınıflandırma performansları Tablo 5.28'de görülmektedir. Bu tabloda saniye cinsinden ortalama geçen eğitim süresine ek olarak spam filtrelerinin en iyi doğruluk yüzdeleriyle bu yüzdelere karşılık gelen FN ve FP oranları verilmektedir. Ayrıca Tablo 5.28'de on katmanlı çapraz doğrulama (ten-fold cross validation) yöntemi kullanılarak elde edilen her bir sonucun standart sapma değerleri gösterilmektedir.

Sonuçlar, sınıflandırma doğruluğu ve FN oranı açılarından Enron-1 veri kümesi üzerinde ABC-LR ve ABC-CSA-LR sınıflandırma yöntemlerinin diğer sınıflandırma yöntemlerinden üstün olduğunu göstermektedir. Ayrıca Enron-1 veri kümesi üzerinde ABC-LR ve ABC-CSA-LR, diğer yöntemlerden kayda değer ölçüde daha iyi FN

oranları elde etmiştir. FP oranları karşılaştırıldığında ise FDA+NB, random forest ve AdaBoost algoritmaları dışında ABC-LR algoritması diğer algoritmalardan daha iyi performans göstermiştir. Fakat bu üç yöntem (FDA+NB, random forest, AdaBoost) FP oranları açısından iyi performans göstermelerine rağmen ABC-CSA-LR ve ABC-LR ile kıyaslanıldığında başarısız yanlış negatif (FN) oranlarından ve düşük sınıflandırma doğruluklarından dolayı spam filtreleme yöntemi olarak bu tez çalışmasında önerilen yöntemlerin çok gerisinde sınıflandırma performansı göstermişlerdir.

Tablo 5.28. Enron-1 veri kümesi üzerinde doğruluk yüzdesi, FN, FP ve saniye cinsinden ortalama geçen eğitim süresi yönlerinden önceki çalışmada [1] tanımlanan algoritmalarla bu tez çalışmasında önerilen yöntemlerin kıyaslaması

Yöntem	ACC $\pm$ Std	FN $\pm$ Std	FP $\pm$ Std	Ort. Süre $\pm$ Std
MDL [1, 65]	95.67 $\pm$ 1.13	0.0107 $\pm$ 0.0105	0.0566 $\pm$ 0.0165	10.1864 $\pm$ 1.4689
FDA + NB [1, 66]	91.29 $\pm$ 1.18	0.1209 $\pm$ 0.0163	0.0045 $\pm$ 0.0057	0.8592 $\pm$ 0.0374
FDA + SVM [1, 66]	96.66 $\pm$ 0.84	0.0459 $\pm$ 0.0178	0.0283 $\pm$ 0.0092	5.4755 $\pm$ 1.1081
IL-C4.5 [1, 67]	93.35 $\pm$ 1.29	0.0608 $\pm$ 0.0207	0.0688 $\pm$ 0.0159	40.0021 $\pm$ 3.6369
Voting [1, 68]	97.20 $\pm$ 1.06	0.0247 $\pm$ 0.0118	0.0294 $\pm$ 0.0130	34.3022 $\pm$ 2.0173
Random forest [1, 69]	98.05 $\pm$ 0.57	0.0178 $\pm$ 0.0121	0.0201 $\pm$ 0.0073	28.0200 $\pm$ 0.3813
AdaBoost [1]	78.76 $\pm$ 1.15	0.6838 $\pm$ 0.0340	0.0200 $\pm$ 0.0800	2.7871 $\pm$ 0.0530
LR [1]	94.54 $\pm$ 1.04	0.0668 $\pm$ 0.0200	0.0496 $\pm$ 0.0125	4.7599 $\pm$ 0.8967
AIRS2Parallel [1]	71.36 $\pm$ 7.65	0.1220 $\pm$ 0.1679	0.3535 $\pm$ 0.1519	21.2569 $\pm$ 0.5605
MLP [1]	96.29 $\pm$ 2.84	0.0501 $\pm$ 0.0823	0.0318 $\pm$ 0.0361	347.943 $\pm$ 9.1491
kNN [1]	91.36 $\pm$ 1.34	0.0378 $\pm$ 0.0158	0.1062 $\pm$ 0.0177	**0.0018 $\pm$ 0.0039**
CNN [1]	97.47 $\pm$ 0.87	0.0347 $\pm$ 0.0193	0.0215 $\pm$ 0.0079	170.738 $\pm$ 28.478
DBB-RDNN-Rel [1]	98.76 $\pm$ 0.57	0.0017 $\pm$ 0.0018	0.0212 $\pm$ 0.0101	183.586 $\pm$ 28.398
ABC-LR* [70]	**98.91 $\pm$ 0.87**	0.0005 $\pm$ 0.0011	**0.0210 $\pm$ 0.0176**	113.565 $\pm$ 1.2632
ABC-CSA-LR*	**98.91 $\pm$ 0.76**	**0.0003 $\pm$ 0.0009**	0.0213 $\pm$ 0.0157	67.274 $\pm$ 0.6204
*. Bu tez çalışmasında önerilen yöntem				

Enron-1 veri kümesi üzerinde önerilen modellerin sınıflandırma doğrulukları sadece sınıflandırma doğruluğu paylaşılan önceki çalışmalardan başka yöntemlerin sonuçlarıyla ayrıca kıyaslanılmıştır. Tablo 5.29, önerilen yöntemlerin diğer yöntemlerden daha iyi sınıflandırma doğruluğuna ulaştığını göstermektedir. Bu tez çalışmasında önerilen ve AIS (yapay bağışıklık sistemi) tabanlı bir sınıflandırma yöntemi olan ABC-CSA-LR ile önceki çalışmalarda önerilen ve yine AIS tabanlı olan yöntemler kıyaslanıldığında ABC-CSA-LR sınıflandırma yönteminin hem AIRS2Parallel [1] yönteminden hem de AIS [71] yönteminden sınıflandırma doğruluğu açısından çok daha üstün olduğu Tablo 5.28 ve Tablo 5.29'dan görülmektedir. ABC-CSA-LR ile AIRS2Parallel yöntemleri

yanlış negatif (FN) ve yanlış pozitif (FP) oranları açılarından kıyaslanıldığında yine ABC-CSA-LR yönteminin çok daha üstün olduğu Tablo 5.28 üzerinde görülmektedir. Fakat ABC-CSA-LR yönteminin çalışma zamanı AIRS2Parallel yönteminin yaklaşık üç katıdır. Adından da anlaşılacağı üzere AIRS2Parallel yöntemi paralel hesaplama yöntemi ile çalışma zamanı iyileştirilmiş bir yöntemdir. Fakat AIRS2Parallel yönteminin sınıflandırma doğruluğu, standart sapma değerleri, FP ve FN oranları açılarından çok kötü bir performans gösterdiği Tablo 5.28 üzerinden görülebilmektedir. Paralel hesaplama yöntemi kullanılarak bu tez çalışmasında önerilen ABC-CSA-LR, ABC-LR ve CSA-LR yöntemlerinin çalışma zamanları iyileştirilebilir.

Tablo 5.29. Enron-1 veri kümesi üzerinde geçmiş çalışmalardaki yöntemlerle bu tez çalışmasında önerilen yöntemlerin doğruluk yüzdelerinin karşılaştırılması

Yöntem	Doğruluk
Deep belief networks [72]	%97.43
AIS [71]	%90.00
Multivariate Bernoulli NB [31]	%94.79
Distinguishing feature selector [73]	%94.35
Minimum description length [74]	%95.56
Bagged RF [75]	%97.75
Enhanced genetic programming [76]	%94.10
RF [77]	%96.39
Relief + NB [78]	%96.30
k means + SVM [79]	%97.35
Natural language toolkit NB [80]	%94.70
Incremental SVM [38]	%96.86
Boosted NB + SVM [81]	%95.60
DBB-RDNN-ReL [1]	%98.76
ABC-LR* [70]	**%98.91**
ABC-CSA-LR*	**%98.91**
*. Bu tez çalışmasında önerilen yöntem	

Spambase veri kümesi üzerinde CSA-LR, ABC-CSA-LR, PSO-LR ve DE-LR yöntemlerinin aynı popülasyon boyutlarında göstermiş oldukları sınıflandırma performansları Tablo 5.30 üzerinde görülmektedir. Tablo 5.30, CSA-LR'nin sınıflandırma performansının PSO-LR'den çok daha iyi olduğunu gösterirken ABC-CSA-LR'nin ise hem CSA-LR yönteminden hem de PSO-LR ve DE-LR yöntemlerinden daha iyi sınıflandırma performansına sahip olduğunu göstermektedir.

Tablo 5.30. Spambase veri kümesi üzerinde CSA-LR, ABC-CSA-LR ve PSO-LR yöntemlerinin performans ölçümleri

P	Model	α	ACC (%)					CC	SN	SP	PPV	NPV	F1	Süre
			En iyi	En kötü	Medyan	Ort.	Std.	En iyi						Ort.
10	CSA	1	92.46	85.94	90.40	90.08	1.52	84.17	91.18	95.81	93.01	94.00	90.32	11.12
		2	92.39	89.78	91.41	91.23	0.72	84.02	90.07	96.17	93.44	93.54	90.25	21.76
		3	92.61	90.29	91.81	91.72	0.65	84.50	90.63	96.05	93.35	93.86	90.57	32.52
		4	93.84	90.65	92.10	92.08	0.75	87.07	91.73	96.53	94.39	94.62	91.99	43.60
		5	93.55	90.87	91.81	91.94	0.59	86.47	91.18	95.69	92.98	94.31	91.77	55.02
	ABC-CSA	1	93.19	91.01	92.32	92.24	0.6	85.71	90.81	95.81	93.19	94.06	91.31	7.30
		2	93.99	90.80	92.57	92.45	0.76	87.42	92.65	96.65	94.30	95.20	92.39	12.92
		3	93.84	90.94	92.43	92.52	0.61	87.07	91.36	96.05	93.64	94.44	92.12	18.62
		4	**94.49**	91.09	92.39	92.38	0.69	**88.33**	**93.20**	96.05	93.54	**95.56**	**92.94**	23.94
		5	93.41	90.94	92.46	92.42	0.57	86.16	91.54	95.81	93.27	94.49	91.57	30.18
	PSO-LR		92.46	72.61	87.72	86.66	4.15	84.18	89.52	94.38	91.20	93.26	90.35	**5.32**
	DE-LR		90.58	63.48	86.09	84.33	6.45	80.42	89.89	**99.16**	89.42	93.15	88.25	5.52
20	CSA	1	92.83	89.35	90.65	90.75	0.87	85.02	91.54	95.22	91.77	94.45	90.96	24.22
		2	93.41	89.86	91.85	91.75	0.71	86.15	90.44	95.93	93.20	93.88	91.53	43.30
		3	93.04	90.80	91.67	91.84	0.56	85.39	92.28	96.29	93.92	94.83	91.08	68.22
		4	93.55	90.72	91.88	91.96	0.65	86.47	91.18	95.33	92.56	94.20	91.75	88.84
		5	94.20	90.51	92.10	92.19	0.99	87.87	92.83	96.17	93.83	95.32	92.66	110.54
	ABC-CSA	1	93.55	90.80	92.39	92.39	0.64	86.47	92.10	96.17	93.79	94.83	91.77	13.28
		2	94.35	91.52	92.46	92.49	0.61	88.13	92.83	96.17	93.83	95.26	92.75	24.64
		3	93.41	**91.67**	92.50	92.56	0.54	86.14	92.46	95.93	93.47	95.04	91.62	36.40
		4	94.06	91.59	92.61	92.51	0.59	87.52	91.91	96.05	93.58	94.72	92.36	48.36
		5	94.06	91.45	92.75	92.71	0.72	87.53	91.73	96.41	94.17	94.67	92.41	59.48
	PSO-LR		91.81	83.41	89.13	88.78	2.19	82.86	90.44	93.18	89.54	93.70	89.62	10.64
	DE-LR		93.26	90.43	91.81	91.76	0.79	85.84	92.28	96.17	93.79	94.90	91.27	10.98
30	CSA	1	93.48	89.42	90.65	90.89	0.99	86.34	91.54	95.10	92.12	94.51	91.71	32.34
		2	92.83	90.14	91.63	91.67	0.70	84.93	90.26	95.22	92.14	93.69	90.79	65.58
		3	93.04	89.49	91.74	91.75	0.90	85.39	90.99	95.22	92.11	94.13	91.08	98.36
		4	93.41	90.65	91.99	92.04	0.71	86.17	91.18	95.45	92.63	94.19	91.58	131.6
		5	93.41	90.65	91.92	92.01	0.67	86.15	90.81	95.45	92.82	94.04	91.52	165.26
	ABC-CSA	1	93.84	91.23	92.54	92.50	0.61	87.09	92.28	96.17	93.59	94.88	92.17	19.22
		2	93.77	91.59	92.72	92.65	0.50	86.94	91.91	95.45	92.83	94.75	92.08	37.12
		3	93.77	91.09	92.50	92.49	0.71	86.94	91.91	96.53	94.29	94.75	92.08	54.12
		4	94.06	91.52	92.57	92.56	0.56	87.53	91.91	95.81	93.42	94.66	92.38	71.16
		5	93.77	91.23	92.72	92.68	0.61	86.94	91.91	96.77	**94.69**	94.75	92.08	88.20
	PSO-LR		91.88	85.22	89.64	89.47	1.79	82.99	89.52	94.50	91.07	93.12	89.67	15.88
	DE-LR		93.19	91.30	92.50	92.44	**0.46**	85.69	90.63	95.81	93.22	93.91	91.25	16.38
40	CSA	1	93.62	88.91	91.16	91.19	0.96	86.61	90.99	95.57	93.02	94.01	91.81	43.00
		2	93.33	90.72	91.74	91.93	0.68	86.12	92.83	95.45	92.75	95.26	91.65	87.52
		3	93.84	90.51	92.21	92.12	0.71	87.09	91.91	95.69	92.97	94.76	92.17	131.44
		4	93.62	91.01	91.81	92.00	0.64	86.65	91.91	96.29	93.90	94.74	91.91	175.6
		5	93.33	90.94	91.88	91.92	0.63	86.00	90.99	95.81	93.12	94.15	91.42	228.48
	ABC-CSA	1	93.55	91.30	92.46	92.48	0.53	86.45	91.18	96.29	94.00	94.28	91.61	25.28
		2	94.35	91.16	**92.83**	**92.72**	0.67	88.13	92.46	96.17	93.96	95.05	92.74	48.28
		3	93.55	91.38	92.64	92.54	0.63	86.46	91.73	96.53	94.35	94.60	91.70	71.74
		4	94.06	91.45	92.64	92.65	0.63	87.52	92.10	96.29	94.02	94.79	92.35	97.22
		5	93.62	91.09	92.64	92.64	0.62	86.62	90.99	96.17	93.77	94.21	91.84	118.04
	PSO-LR		92.10	84.71	89.57	89.19	1.85	83.45	90.63	93.66	90.20	93.80	89.95	21.20
	DE-LR		93.84	91.16	92.36	92.45	0.70	87.07	92.65	95.93	93.50	95.15	92.11	23.90

Tablo 5.30'da sınıflandırma yöntemlerinin elde ettikleri en iyi ve en kötü sınıflandırma doğruluğu sonuçları (ACC) Tablo 5.31'de verilmektedir. Tablo 5.31'de paylaşılan sonuçlardan, PSO-LR, DE-LR ve CSA-LR ile kıyaslanıldığında tüm deneylerin en iyi ve en kötü sınıflandırma doğrulukları arasındaki fark ABC-CSA-LR yönteminde daha düşük olduğundan dolayı ABC-CSA-LR'nin daha kararlı ve güvenilir bir sınıflandırma yöntemi olduğu anlaşılmaktadır. Ayrıca Tablo 5.30'dan görüldüğü üzere çok sayıda koşmanın sonucunda hem CSA-LR'den hem de PSO-LR'den daha düşük standart sapma değerlerine sahip olduğundan dolayı ABC-CSA-LR yönteminin daha kararlı olduğu söylenebilir. Fakat CSA-LR ve ABC-CSA-LR ile kıyaslanıldığında PSO-LR ve DE-LR yöntemleri daha düşük çalışma zamanına ihtiyaç duymaktadırlar.

Tablo 5.31. Spambase veri kümesi üzerinde CSA-LR, ABC-CSA-LR ve PSO-LR yöntemlerinin en iyi ve en kötü sınıflandırma doğrulukları

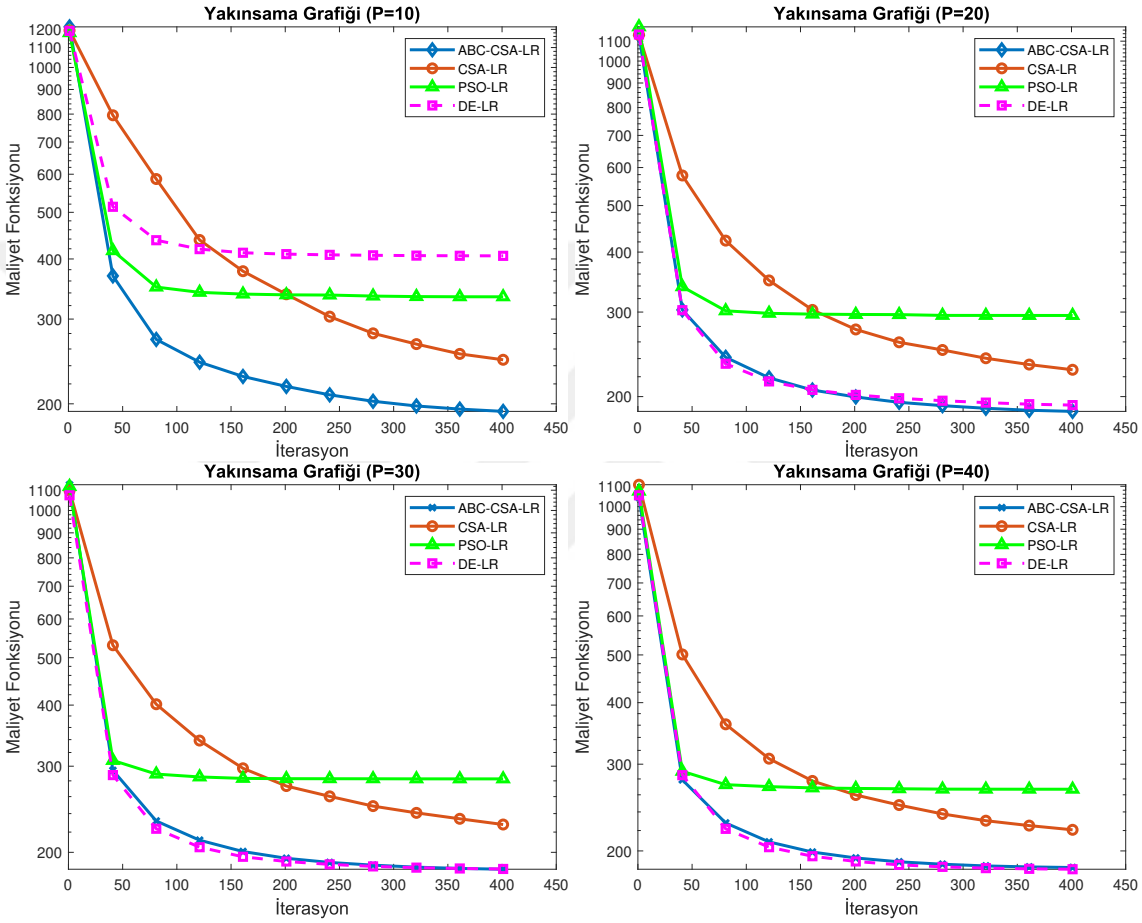
Yöntem	En iyi	En kötü
ABC-CSA-LR*	**%94.49**	**%90.80**
CSA-LR*	%94.20	%85.94
PSO-LR	%92.46	%72.61
DE-LR	%93.84	%63.48

Tablo 5.32. Spambase veri kümesi üzerinde önceki çalışmalarda önerilen sınıflandırma yöntemleriyle CSA-LR, ABC-CSA-LR ve PSO-LR sınıflandırma yöntemlerinin performans ölçümleri

Yöntem	En iyi									En kötü		
	ACC	Ort.	Std	CC	F1	SN	PPV	SP	NPV	ACC	Ort.	Std
NSA [46]	68.86	-	-	48.33	36.01	22.24	94.53	**99.16**	66.24	-	-	-
PSO [46]	81.32	-	-	60.95	71.84	60.48	88.44	94.86	78.69	-	-	-
NSA-PSO [46]	91.22	-	-	63.37	74.95	65.99	86.72	93.43	80.86	-	-	-
CNSA-FFO [17]	93.88	-	-	86.29	45.34	87.28	94.38	97.31	93.66	-	-	-
PSO-LR	93.12	91.07	0.70	85.54	91.19	92.28	92.97	95.69	94.84	72.61	86.66	4.15
DE-LR	94.06	92.69	**0.46**	87.52	92.42	93.01	94.55	**99.16**	95.42	63.48	84.33	6.45
CSA-LR*	94.20	92.19	0.56	87.87	92.66	92.83	94.39	96.53	95.32	85.94	90.08	1.52
ABC-CSA-LR*	**94.49**	**92.72**	0.50	**88.33**	**92.94**	**93.20**	**94.69**	96.77	**95.56**	**90.80**	**92.24**	**0.76**
*. Bu tez çalışmasında önerilen yöntem												

Bu tez çalışmasında önerilen CSA-LR ve ABC-CSA-LR yöntemleri ve bu yöntemlerle kıyaslama yapabilme amacıyla tasarlanan PSO-LR ve DE-LR sınıflandırma yöntemleri, Spambase veri kümesi üzerinde daha önce yapılmış literatürde bilinen çalışmalarla

kıyaslanılmış ve sonuçlar Tablo 5.32'de paylaşılmıştır. Tabloda yer alan en iyi sonuçlar koyu renkle vurgulanılmış ve bu tablodan NSA ve DE-LR yöntemlerinin daha iyi özgüllük (SP) değerine sahip olmalarına rağmen diğer performans ölçütleri (ACC, CC, F1, SN, NPV) açılarından bu tez çalışmasında önerilen yöntemlerin diğer sınıflandırma modellerinden üstün olduğu görülmektedir.



Şekil 5.7. Spambase veri kümesi üzerinde eğitim aşamasında dört farklı popülasyon boyutu ($P = \{10, 20, 30, 40\}$) için CSA-LR, ABC-CSA-LR, PSO-LR ve DE-LR yöntemlerinin yakınsama grafikleri

Spambase veri kümesi üzerinde eğitim aşamasında CSA algoritmasının yakınsama yeteneğinin ABC algoritmasının kullanımıyla nasıl etkilendiğini görmek için CSA-LR, ABC-CSA-LR, PSO-LR ve DE-LR sınıflandırma yöntemlerinin yakınsama grafikleri analiz edilmiştir. Şekil 5.7, Spambase veri kümesi üzerinde eğitim aşamasında farklı popülasyon boyutları için CSA-LR, ABC-CSA-LR, PSO-LR ve DE-LR yöntemlerinin yakınsama grafiklerini göstermektedir. PSO-LR, DE-LR ve önerilen yöntemlerin yakınsama grafiklerini adil bir şekilde kıyaslayabilmek için CSA-LR ve ABC-CSA-LR

yöntemlerindeki α değeri 1 ve $P = \{10, 20, 30, 40\}$ olduğunda elde edilen sonuçlar seçilmiştir. Şekil 5.7'den görüldüğü üzere ABC-CSA-LR yöntemi eğitim aşamasında CSA-LR ve PSO-LR yöntemlerinden daha hızlı yakınsamaktadır. Popülasyon boyutu 30 ve 40 için ABC-CSA-LR ve DE-LR yöntemleri benzer yakınsama oranlarına sahipken özellikle popülasyon boyutu 10 için ABC-CSA-LR yöntemi çok daha iyi yakınsamaktadır. Ek olarak ABC-CSA-LR ve CSA-LR yöntemleri popülasyon boyutu ne olursa olsun iterasyon ilerledikçe yerel minimuma takılmadan öğrenmeyi sürdürmektedirler. Fakat PSO-LR yöntemi az sayıda iterasyondan sonra bulduğu en iyi çözümü iyileştirememekte ve yerel minimuma takılmaktadır. CSA'nın mutasyon işlemlerini ABC algoritmasının aşamalarıyla değiştirmenin daha hızlı yakınsama sağladığı ve arama yeteneğini iyileştirdiği grafiklerden anlaşılmaktadır.

Sınıflandırma yöntemlerinin performanslarını kıyaslama amacıyla yapılan başka bir çalışmada Turk2 veri kümesi kullanılarak LR, ABC-LR, DE-LR ve PSO-LR sınıflandırma yöntemleri beş farklı öznitelik vektör boyutu ($FVS = \{50, 100, 150, 200, 300\}$) ve maksimum iterasyon sayısına 1000 atanarak eğitilmiştir. ABC-LR yönteminde yer alan MR ve $limit$ parametrelerine sırasıyla 0.1 ve 100 değerleri verilmiştir. PSO-LR yönteminin w, c_1 ve c_2 parametrelerine sırasıyla 0.6, 2 ve 2 değerleri verilirken DE-LR yönteminin cr ve f parametrelerine sırasıyla 0.8 ve 0.5 değerleri verilmiştir. LR sınıflandırma modelinin α ve λ parametrelerine 0.001 ve 0 değerleri verilmiştir. Yöntemlerin parametreleri kapsamlı grid arama tekniği kullanılarak belirlenilmiştir. ABC-LR, PSO-LR ve DE-LR yöntemleri tarafından öğrenilen her bir parametrenin alt sınır (x_j^{min}) ve üst sınır (x_j^{max}) değerleri sırasıyla -64 ve $+64$ olarak belirlenilmiştir. Deneylerde iki farklı öznitelik seçim yöntemi ($tf - idf$, karşılıklı bilgi) kullanılmıştır. En iyi parametreler belirlendikten sonra her bir sınıflandırma modeli ve her bir öznitelik seçim yöntemi için toplam beş deney gerçekleştirilmiştir. Sınıflandırma performansı üzerinde rastgele veri etkisini azaltmak için her bir deney 20 kez tekrarlanılmış ve verilerin sırası ile bölünmesinden kaynaklı etkileri engellemek için 10-katmanlı çapraz doğrulama tekniği uygulanılmıştır.

Her bir deneyin 20 kez tekrarlanmasından elde edilen saniye cinsinden ortalama harcanan eğitim zamanına ek olarak en iyi, en kötü, medyan, ortalama ve standart sapma değerleri Tablo 5.33 ve Tablo 5.34 üzerinde görülmektedir. Tablo 5.33'de görülen

sonuçlar $tf - idf$ tabanlı öznitelik seçme yaklaşımı kullanılarak elde edilirken Tablo 5.34'de görülen sonuçlar karşılıklı bilgi yöntemi kullanılarak elde edilmiştir. Deneysel sonuçlar sınıflandırma doğruluğu açısından ABC-LR sınıflandırma yönteminin LR, PSO-LR ve DE-LR yöntemlerinden daha iyi sınıflandırma performansına sahip olduğunu göstermektedir. Diğer sınıflandırma yöntemleriyle kıyaslanıldığında özellikle de $tf - idf$ tabanlı öznitelik seçme yaklaşımı kullanıldığı zaman ABC-LR yönteminin daha düşük standart sapma değerlerine sahip olduğu görülmektedir. Ayrıca LR, PSO-LR ve DE-LR yöntemleriyle karşılaştırıldığında ABC-LR yönteminin en iyi ve en kötü sınıflandırma doğruluğu arasındaki fark daha düşüktür. Tüm bu veriler ABC-LR'nin daha kararlı ve güvenilir bir sınıflandırma modeli olduğunu göstermektedir. Turk2 veri kümesi üzerinde ABC-LR yöntemi %96.13 başarı oranıyla en iyi sınıflandırma doğruluğunu elde etmiştir.

Tablo 5.33. Turk2 veri kümesi üzerinde öznitelik seçme yaklaşımı olarak $tf - idf$ yöntemini kullanan LR, ABC-LR, DE-LR ve PSO-LR sınıflandırma yöntemlerinin performans ölçümleri

Yöntem	FVS	En iyi	En kötü	Medyan	Ort.	Std.	Süre
LR	50	%92.67	%90.67	%92.07	%91.93	0.57	**0.061**
	100	%93.07	%91.60	%92.33	%92.26	0.47	0.080
	150	%94.13	%91.60	%92.60	%92.58	0.63	0.122
	200	%93.87	%92.13	%93.33	%93.16	0.48	0.136
	300	%94.27	%91.73	%93.47	%93.43	0.63	0.154
ABC-LR	50	%92.40	%92.40	%92.40	%92.40	**0.00**	3.76
	100	%93.60	%93.07	%93.20	%93.28	0.13	4.71
	150	%94.40	%93.47	%94.00	%93.96	0.18	5.19
	200	%95.47	%94.80	%95.07	%95.07	0.18	5.65
	300	**%96.13**	**%95.07**	**%95.47**	**%95.53**	0.25	6.67
DE-LR	50	%92.40	%91.07	%92.00	%91.89	0.47	12.64
	100	%93.33	%91.07	%92.07	%92.04	0.58	66.92
	150	%93.60	%92.13	%93.27	%93.04	0.53	93.12
	200	%94.53	%93.07	%93.80	%93.77	0.52	117.54
	300	%94.80	%92.67	%94.07	%93.91	0.68	165.57
PSO-LR	50	%92.00	%90.80	%91.27	%91.31	0.41	10.78
	100	%90.13	%87.87	%89.47	%89.33	0.81	61.01
	150	%91.33	%88.13	%89.40	%89.51	1.09	88.70
	200	%91.07	%88.00	%89.07	%89.24	1.01	114.13
	300	%91.07	%87.20	%88.87	%89.01	1.16	161.63

Tablo 5.34. Turk2 veri kümesi üzerinde öznitelik seçme yaklaşımı olarak karşılıklı bilgi yöntemini kullanan LR, ABC-LR, DE-LR ve PSO-LR sınıflandırma yöntemlerinin performans ölçümleri

Yöntem	FVS	En iyi	En kötü	Medyan	Ort.	Std.	Süre
LR	50	%92.27	%90.00	%91.27	%91.29	0.67	**0.102**
	100	%93.33	%90.53	%91.33	%91.53	0.80	0.115
	150	%92.93	%90.80	%91.67	%91.69	0.65	0.126
	200	%92.53	%89.73	%91.13	%91.13	0.78	0.139
	300	%93.07	%89.33	%91.27	%91.17	0.83	0.120
ABC-LR	50	%93.60	%92.80	%93.20	%93.17	0.32	4.91
	100	%94.67	%92.93	%93.67	%93.67	0.49	6.57
	150	%94.27	%92.93	%93.40	%93.44	0.41	7.64
	200	%94.40	%93.73	%94.13	%94.12	**0.22**	8.63
	300	**%95.47**	**%94.00**	**%94.80**	**%94.89**	0.47	10.61
DE-LR	50	%93.07	%92.00	%92.67	%92.60	0.40	11.94
	100	%93.60	%92.00	%92.80	%92.76	0.46	62.61
	150	%94.53	%93.07	%94.07	%93.87	0.55	91.95
	200	%93.60	%92.13	%93.13	%92.97	0.59	115.62
	300	%94.40	%93.20	%93.73	%93.75	0.40	163.03
PSO-LR	50	%92.53	%91.20	%91.40	%91.67	0.46	10.76
	100	%93.20	%90.27	%92.00	%91.89	0.80	66.78
	150	%92.27	%91.07	%91.87	%91.79	0.36	92.96
	200	%92.00	%90.27	%91.73	%91.41	0.68	120.37
	300	%93.20	%90.27	%91.53	%91.55	0.80	176.85

6. BÖLÜM

SONUÇ VE ÖNERİLER

6.1. Katkılar

Bu çalışmada önerilen ABC-LR ve ABC-CSA-LR yöntemleri, çözüm uzayında bölgesel ve küresel aramalar yapabilen ve verilerden oldukça karmaşık özelliklerin öğrenilmesini mümkün kılan öğrenme algoritmalarından dolayı lineer olmayan ve çok boyutlu verilerin üstesinden gelebilmektedirler. Önerilen yöntemler lineer olan veya olmayan, dengeli veya dengesiz, kişisel olan veya olmayan, İngilizce veya Türkçe olmak üzere geniş bir yelpazede spam veri kümeleri için etkilidir. Çevrimiçi olarak erişilebilen veri kümeleri üzerinde yapılan deneysel çalışmalar önerilen yöntemlerin dil farklılıklarının ve karmaşıklığının üstesinden gelebildiğini göstermektedir. Tez çalışmasında kullanılan aşırı örnekleme yöntemiyle veri dengesizliği probleminin üstesinden gelinmiştir. Öznitelik ağırlıklarına karşı hassasiyet ile ilişkili sınırlama $tf - idf$ yönteminin entegrasyonuyla azaltılmıştır. Tezde önerilen sistemler literatüre yeni yaklaşımlar kazandırmış ve mevcut çalışmalara göre daha başarılı sonuçlar üretmiştir.

6.2. Sonuç ve Öneriler

LR sınıflandırma yöntemi gerçek zamanlı uygulamalar için uygun olan kolay ve etkili bir sınıflandırıcıdır. Fakat, optimum ağırlıkları elde etmek için kullanılan eğitim algoritması lokal minimuma takılabilir ve denk öznitelik ağırlıklarına karşı hassastır. Bu tez çalışmasında LR'nin, ABC'nin, CSA'nın ve $tf-idf$ yöntemlerinin avantajlarını bir araya getiren üç yeni spam filtreleme modeli (ABC-LR, CSA-LR, ABC-CSA-LR) önerilmiştir. Önerilen yöntemlerin Türkçe ve İngilizce dillerindeki dört farklı (TurkishEmail, CSDMC2010, Enron ve Spambase) erişime açık veri kümesine uygulanılmasına ek olarak çevrimiçi erişime açık olmayan Turk2 veri kümesine uygulanılmıştır. Dil

farklılıklarının üstesinden gelmek ve kelimelerin yalın halini elde etmek için iki farklı kök bulma kütüphanesi kullanılmıştır. Önerilen yöntemlerle kıyaslama yapabilme amacıyla literatürde bilinen makine öğrenmesi yöntemleri olan iki farklı NB, iki farklı SVM, "gradient descent" algoritması kullanılarak eğitilen LR sınıflandırma yöntemlerinin programları yazılmış ve veri kümelerine uygulanılmıştır. Yine önerilen yöntemlerle kıyaslamak için literatürde bilinen optimizasyon teknikleri olan PSO ve DE algoritmaları LR sınıflandırma modelinin eğitilmesi aşamasında kullanılmıştır. Programlanan PSO-LR ve DE-LR sınıflandırma modelleri veri kümelerine uygulanılmıştır. Önerilen yöntemler ile bu tez çalışmasında programlanan diğer yöntemlerin sınıflandırma performansları kıyaslanılmıştır. Deneysel sonuçlar önerilen yöntemlerin sınıflandırma doğruluğu açısından diğer yöntemlerden üstün olduğunu göstermektedir. Yukarıda söylenenlere ek olarak deneysel sonuçlar, kök bulma işleminde dilin morfolojik yapısı dikkate alınarak dil karmaşıklığının kolaylıkla aşılabileceğini göstermektedir.

Ayrıca, önceki çalışmalarda yer alan en başarılı yöntemlerle önerilen yöntemlerin (ABC-LR, ABC-CSA-LR) sınıflandırma performanslarını karşılaştırmak için literatürde bilinen Enron-1 veri kümesi kullanılmıştır. Önerilen yöntemlerin sınıflandırma başarısı ve FN değeri açısından diğer yöntemlerden daha üstün performans sergilediği deneysel olarak gösterilmiştir. Önerilen ABC-LR ve ABC-CSA-LR yöntemleri açık bir şekilde dengesiz, lineer olmayan ve karmaşık spam veri kümeleri üzerinde etkili bir performans sergilemiştir.

Önerilen yöntemlerin başlıca dezavantajı bazı yöntemler dışındaki diğer yöntemlerle kıyaslandığında hesaplama karmaşıklığının yüksek olmasıdır. ABC-LR, CSA-LR ve ABC-CSA-LR yöntemlerinin hesaplama karmaşıklığı paralel hesaplama teknikleri kullanılarak iyileştirilebilir ve bu yönde yapılacak çalışmalar önerilir. İleride yapılacak bir başka çalışma SVM, ANN ve CNN gibi farklı sınıflandırma modellerini eğitmek için ABC algoritmasını kullanmak olabilir. Önerilen yöntemlerin bir başka dezavantajı ise kontrol parametrelerinin sayısının fazla olmasından kaynaklı en iyi sonucu veren parametreleri bulmanın zor ve zahmetli olmasıdır. Bu yüzden otomatik parametre ayarlaması veya çözüm uzayını arama esnasında uyarlamalı parametre düzenleme faydalı bir çalışma olabilir. Önerilen yöntemler sadece metin içerikli spam e-postaları filtrelemeye yöneliktir. Bu yüzden Türkçe veya İngilizce dillerinde metinler barındıran

resimler içeren veya metinsiz resimler içeren veya hem metin hem de resim içeren çok daha geniş kapsamlı spam e-postaları tespit etmeye yönelik çalışmalar önemlidir. Bu çalışmada önerilen yöntemlerin ikiden fazla sınıftan oluşan Türkçe ve İngilizce metinleri sınıflandıracak şekilde uyarlanması bir başka önemli çalışma olabilir. Son olarak bu çalışmada önerilen yöntemlerin milli işletim sistemlerine veya açık kaynak kodlu e-posta yazılım araçlarına entegresine yönelik çalışmalar önemlidir.

KAYNAKLAR

1. Barushka, A. and Hajek, P., 2018. Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks, **Applied Intelligence**, (10):3538–3556.

2. V. Cormack, G., 2008. Email Spam Filtering: A Systematic Review, **Foundations and Trends in Information Retrieval**, 335–455.

3. Statista, 2020. Number of e-mail users worldwide from 2017 to 2024, Technical report, Statista.

4. Radicati, 2019. Email Statistics Report, 2019-2023, Executive summary, The Radicati Group.

5. Statista, 2018. Global e-mail spam rate from 2012 to 2018, Technical report, Statista.

6. Symantec, 2018. Internet Security Threat Report, Technical report, Symantec.

7. Kaspersky, 2018. Spam and phishing in Q3 2018, Spam and phishing reports, Kaspersky.

8. Bhowmick, A. and Hazarika, S.M., 2017. E-Mail Spam Filtering: A Review of Techniques and Trends, Lecture Notes in Electrical Engineering, 583–590, Springer Singapore.

9. Ozgur, L., Gungor, T., and Gurgen, F., 2004. Adaptive anti-spam filtering for agglutinative languages: a special case for Turkish, **Pattern Recognition Letters**, (16):1819 – 1831.

10. Karaboga, D., 2005. An Idea Based On Honey Bee Swarm For Numerical Optimization, Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.

11. Han, Y., Yang, M., Qi, H., He, X., and Li, S., 2009. The Improved Logistic Regression Models for Spam Filtering, *2009 International Conference on Asian Language Processing*, 314–317.

12. Karaboga, D. and Akay, B., 2009. A comparative study of Artificial Bee Colony algorithm, **Applied Mathematics and Computation**, 108–132.

13. Akay, B. and Karaboga, D., 2012. A modified Artificial Bee Colony algorithm for real-parameter optimization, **Information Sciences**, 120–142.

14. Oda, T. and White, T., 2003. Developing an Immunity to Spam, *Proceedings of the 2003 International Conference on Genetic and Evolutionary Computation: PartI*, GECCO'03, 231–242, Springer-Verlag, Berlin, Heidelberg.

15. Forrest, S., Perelson, A.S., Allen, L., and Cherukuri, R., 1994. Self-nonself discrimination in a computer, *Proceedings of 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, 202–212.

16. de Castro, L.N. and Von Zuben, F.J., 2002. Learning and optimization using the clonal selection principle, **IEEE Transactions on Evolutionary Computation**, (3):239–251.

17. Chikh, R. and Chikhi, S., 2019. Clustered negative selection algorithm and fruit fly optimization for email spam detection, **Journal of Ambient Intelligence and Humanized Computing**, (1):143–152.

18. Ulutas, B.H. and Islier, A.A., 2009. A clonal selection algorithm for dynamic facility layout problems, **Journal of Manufacturing Systems**, 123 – 131.

19. Gong, M., Jiao, L., and Zhang, L., 2010. Baldwinian learning in clonal selection algorithm for optimization, **Information Sciences**, (8):1218 – 1236.

20. Sharma, A. and Sharma, D., 2011. Clonal Selection Algorithm for Classification, *P. Liò, G. Nicosia, and T. Stibor, Eds:, Artificial Immune Systems*, 361–370, Springer Berlin Heidelberg, Berlin, Heidelberg.

21. d. S. Batista, L., Guimaraes, F.G., and Ramirez, J.A., 2009. A Distributed Clonal Selection Algorithm for Optimization in Electromagnetics, **IEEE Transactions on Magnetics**, 1598–1601.

22. Haktanirlar Ulutas, B. and Kulturel-Konak, S., 2011. A review of clonal selection algorithm and its applications, **Artificial Intelligence Review**, (2):117–138.

23. Lining Zhang, Maoguo Gong, Licheng Jiao, and Jie Yang, 2008. Optimal Approximation of Linear Systems by an Improved Clonal Selection Algorithm, *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 527–534.

24. Xu, N., Ding, Y., Ren, L., and Hao, K., 2018. Degeneration Recognizing Clonal Selection Algorithm for Multimodal Optimization, **IEEE Transactions on Cybernetics**, (3):848–861.

25. Li, Z., Xia, Y., and Sahli, H., 2019. CSA-DE/EDA: a Novel Bio-inspired Algorithm for Function Optimization and Segmentation of Brain MR Images, **Cognitive Computation**, (6):855–868.

26. Heckerman, D., Horvitz, E., Sahami, M., and Dumais, S., 1998. A Bayesian Approach to Filtering Junk E-Mail, *AAAI Workshop on Learning for Text Categorization*, 55–62.

27. Androutsopoulos, I., Koutsias, J., Chandrinos, K., Paliouras, G., and Spyropoulos, C., 2000. An Evaluation of Naive Bayesian Anti-Spam Filtering, **CoRR**.

28. Metsis, V., Androutsopoulos, I., and Paliouras, G., 2006. Spam Filtering with Naive Bayes - Which Naive Bayes?, *THIRD CONFERENCE ON EMAIL AND ANTI-SPAM (CEAS*.

29. Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., and Spyropoulos, C.D., 2000. An Experimental Comparison of Naive Bayesian and Keyword-based Anti-spam Filtering with Personal e-Mail Messages, *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, 160–167, ACM, New York, NY, USA.

30. Rusland, N.F., Wahid, N., Kasim, S., and Hafit, H., 2017. Analysis of Naive Bayes Algorithm for Email Spam Filtering across Multiple Datasets, **IOP Conference Series: Materials Science and Engineering**.

31. Almeida, T.A., Almeida, J., and Yamakami, A., 2011. Spam filtering: how the dimensionality reduction affects the accuracy of Naive Bayes classifiers, **Journal of Internet Services and Applications**, (3):183–200.

32. Feng, W., Sun, J., Zhang, L., Cao, C., and Yang, Q., 2016. A support vector machine based naive Bayes algorithm for spam filtering, *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*, 1–8.

33. N. Vapnik, V., 1995. The Nature Of Statistical Learning Theory, Springer.

34. Drucker, H., Wu, D., and Vapnik, V., 1999. Support vector machines for spam categorization, **IEEE transactions on neural networks**, 1048–54.

35. Amayri, O. and Bouguila, N., 2010. A study of spam filtering using support vector machines, **Artificial Intelligence Review**, (1):73–108.

36. Sculley, D. and Wachman, G.M., 2007. Relaxed Online SVMs for Spam Filtering, *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, 415–422, ACM, New York, NY, USA.

37. Yu, B. and ben Xu, Z., 2008. A comparative study for content-based dynamic spam classification using four machine learning algorithms, **Knowledge-Based Systems**, (4):355 – 362.

38. Sanghani, G. and Kotecha, K., 2016. Personalized spam filtering using incremental training of support vector machine, *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, 323–328.

39. Goodman, J. and Yih, S.W.t., 2006. Online Discriminative Spam Filter Training, *Proceedings of the 3rd Conference on Email and Anti-Spam*, CEAS.

40. Chang, M.w., Yih, W.t., and Meek, C., 2008. Partitioned Logistic Regression for Spam Filtering, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, 97–105, ACM, New York, NY, USA.

41. Gungor, T. and Ciltik, A., 2007. Developing Methods and Heuristics with Low Time Complexities for Filtering Spam Messages, *Natural Language Processing and Information Systems*, 35–47, Springer Berlin Heidelberg, Berlin, Heidelberg.

42. Yue, X., Abraham, A., Chi, Z.X., Hao, Y.Y., and Mo, H., 2006. Artificial immune system inspired behavior-based anti-spam filter, **Soft Computing**, (8):729–740.

43. Guzella, T., Mota-Santos, T., Uchôa, J., and Caminhas, W., 2008. Identification of SPAM messages using an approach inspired on the immune system, **Biosystems**, 215–225.

44. Ruan, G. and Tan, Y., 2010. A Three-Layer Back-Propagation Neural Network for Spam Detection Using Artificial Immune Concentration, **Soft Comput.**, 139–150.

45. Mohammad, A.H. and Zitar, R.A., 2011. Application of genetic optimized artificial immune system and neural networks in spam detection, **Applied Soft Computing**, (4):3827–3845.

46. Idris, I. and Selamat, A., 2014. Improved email spam detection model with negative selection algorithm and particle swarm optimization, **Applied Soft Computing**, 11 – 27.

47. Tutun, S., Khanmohammadi, S., He, L., and Chou, C.A., 2016. A Meta-heuristic LASSO Model for Diabetic Readmission Prediction, *Conference: Industrial and Systems Engineering Research Conference (ISERC)*.

48. Kennedy, J. and Eberhart, R., 1995. Particle swarm optimization, *Proceedings of ICNN'95-International Conference on Neural Networks*, 1942–1948, IEEE.

49. Storn, R. and Price, K., 1997. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, **Journal of Global Optimization**, 341–359.

50. Ergin, S., Gunal, E.S., Yigit, H., and Aydin, R.S.T., 2012. Turkish anti-spam filtering using binary and probabilistic models, *2nd World Conference on Information Technology, WCIT-2011*, 1007–1012.

51. Tin Kam Ho and Basu, M., 2002. Complexity measures of supervised classification problems, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, (3):289–300.

52. Bache, K. and Lichman, M., 2013, UCI Machine Learning Repository.

53. Bird, S., Klein, E., and Loper, E., 2009. Natural language processing with Python: analyzing text with the natural language toolkit, O'Reilly Media, Inc., 1st edition.

54. Golub, T., 2019. Modernized Mathematical Model of Text Document Classification, *CMIS*, 607–617.

55. Çıltık, A., 2006. TIME EFFICIENT SPAM E-MAIL FILTERING FOR TURKISH, Yüksek Lisans Tezi, Boğaziçi University.

56. Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features, *European conference on machine learning*, 137–142, Springer.

57. Zafer, H.R., Natural Language Processing Library for Turkish in C#.

58. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python, **Journal of Machine Learning Research**, 2825–2830.

59. Manning, C.D., Schütze, H., and Raghavan, P., 2008. Introduction to information retrieval, Cambridge university press.

60. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G., 2013. API design for machine learning software: experiences from the scikit-learn project, *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108–122.

61. Hsu, C.W., Chang, C.C., and Lin, C.J., 2003. A Practical Guide to Support Vector Classification, Technical report, Department of Computer Science, National Taiwan University.

62. Chang, C.C. and Lin, C.J., 2011. LIBSVM: A Library for Support Vector Machines, **ACM Trans. Intell. Syst. Technol.**, (3):1–27.

63. Vesterstrom, J. and Thomsen, R., 2004. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems, *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, 1980–1987 Vol.2.

64. Akay, B. and Karaboga, D., 2009. *Parameter Tuning for the Artificial Bee Colony Algorithm*, 608–619.

65. Almeida, T.A. and Yamakami, A., 2016. Compression-based spam filter, **Security and Communication Networks**, 327–335.

66. Aragão, M.V., Frigieri, E.P., Ynoguti, C.A., and Paiva, A.P., 2016. Factorial design analysis applied to the performance of SMS anti-spam filtering systems, **Expert Systems with Applications**, 589 – 604.

67. Sheu, J.J., Chu, K.T., Li, N.F., and Lee, C.C., 2017. An efficient incremental learning mechanism for tracking concept drift in spam filtering, **PLOS ONE**, (2):1–17.

68. Najadat H, Abdulla N, A.R.N.S., 2016. Spam detection for mobile short messaging service using data mining classifiers, **International Journal of Computer Science and Information Security (IJCSIS)**, 511–517.

69. Khorshidpour, Z., Hashemi, S., and Hamzeh, A., 2017. Evaluation of random forest classifier in security domain, **Applied Intelligence**, 558–569.

70. Dedeturk, B.K. and Akay, B., 2020. Spam filtering using a logistic regression model trained by an artificial bee colony algorithm, **Applied Soft Computing**, 106229.

71. Abi-Haidar, A. and Rocha, L., 2008. *Adaptive Spam Detection Inspired by the Immune System*.

72. Tzortzis, G. and Likas, A., 2007. Deep Belief Networks for Spam Filtering, *19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)*, 306–309.

73. Uysal, A.K. and Gunal, S., 2012. A novel probabilistic feature selection method for text classification, **Knowledge-Based Systems**, 226 – 235.

74. Almeida, T.A. and Yamakami, A., 2012. Occam's razor-based spam filter, **Journal of Internet Services and Applications**, 245–253.

75. Shams, R. and Mercer, R.E., 2013. Personalized Spam Filtering with Natural Language Attributes, *2013 12th International Conference on Machine Learning and Applications*, 127–132.

76. Trivedi, S.K. and Dey, S., 2013. An Enhanced Genetic Programming Approach for Detecting Unsolicited Emails, *2013 IEEE 16th International Conference on Computational Science and Engineering*, 1153–1160.

77. Mishra, R. and Thakur, R., 2013. Analysis of Random Forest and Naive Bayes for Spam Mail using Feature Selection Catagorization, **International Journal of Computer Applications**, 42–47.

78. Trivedi, S.K. and Dey, S., 2016. A Comparative Study of Various Supervised Feature Selection Methods for Spam Classification, *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, ICTCS '16, Association for Computing Machinery, New York, NY, USA.

79. Hassan, D., 2017. Investigating the Effect of Combining Text Clustering with Classification on Improving Spam Email Detection, *A.M. Madureira, A. Abraham, D. Gamboa, and P. Novais, Eds:, Intelligent Systems Design and Applications*, 99–107, Springer International Publishing, Cham.

80. Chhogyal, K. and Nayak, A., 2016. An Empirical Study of a Simple Naive Bayes Classifier Based on Ranking Functions, *B.H. Kang and Q. Bai, Eds:, AI 2016: Advances in Artificial Intelligence*, 324–331, Springer International Publishing, Cham.

81. Trivedi, S.K. and Dey, S., 2016. A Combining Classifiers Approach for Detecting Email Spams, *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 355–360.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı, Soyadı	:	Bilge Kağan DEDETÜRK
Uyruğu	:	Türkiye (T.C.)
Doğum Tarihi ve Yeri	:	11.06.1988 - KAYSERİ
Medeni Durum	:	Evli
E-posta	:	kagandedeturk@gmail.com
Adres	:	Yakut Mah. 3852 Sok 7/36 Kocasinan/KAYSERİ

EĞİTİM

Derece	**Kurum**	**Mez.Yılı**
Lise	Nuh Mehmet Küçükçalık Anadolu Lisesi, Kayseri	2006
Lisans	Gazi Üniversitesi, Bilgisayar Sistemleri Öğretmenliği	2010
Yüksek Lisans	Melikşah Üniversitesi, Bilgisayar Mühendisliği	2014
Doktora	Erciyes Üniversitesi, Bilgisayar Mühendisliği	2020

İŞ DENEYİMİ

Yıl	**Kurum**	**Görev**
2018-Halen	Kayseri Ulaşım AŞ.	Yazılım Geliştirme Şefi
2012-2016	Melikşah Üniversitesi	Araştırma Görevlisi

YABANCI DİL

İngilizce

YAYINLAR

1. B. K. Dedeturk, B. Akay, Spam filtering using a logistic regression model trained by an artificial bee colony algorithm, Applied Soft Computing (2020)

2. B. K. Dedeturk, B. Akay, A novel logistic regression classifier trained by an improved clonal selection algorithm for e-mail spam detection, Applied Soft Computing, In Revision

3. B. K. Dedeturk, B. Akay, D. Karaboga, Chapter: Artificial Bee Colony Algorithm and Its Application on Spam Mail Detection, In Book:Nature-Inspired Metaheuristic Algorithms for Engineering Optimization Applications, Springer, In Press

BURSLAR ve PROJELER

1. Tubitak Doktora Bursu (2228-B) 2014/1

2. Tubitak Proje Bursu (3501) - Yüksek Lisans - Proje Kodu: 112E192, Proje Konusu: Steiner Ormanı Ve İlişkili Problemler İçin Yeni Yaklaştırma Algoritmaları

3. Tubitak Proje Bursu (3001) - Doktora - Proje Kodu: 116E947, Proje Konusu: Optimize Edilmiş Sınıflayıcıların Tasarımı ve Türkçe Spam Mail Filtreleme Üzerinde Başarımlarının İncelenmesi