
GAZİ ÜNİVERSİTESİ
Gazi
1926

İSTENMEYEN ELEKTRONİK POSTA SINIFLANDIRMA PROBLEMİNDE ETKİN ÖZELLİK SEÇİMİ

Nuriye BAKTIR

YÜKSEK LİSANS
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAZİRAN 2023

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Nuriye BAKTIR
16/06/2023

İSTENMEYEN ELEKTRONİK POSTA SINIFLANDIRMA PROBLEMİNDE ETKİN ÖZELLİK SEÇİMİ

(Yüksek Lisans Tezi)

Nuriye BAKTIR

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2023

ÖZET

Günümüzde hızlı ve güvenli iletişim büyük önem taşımaktadır. Bu bağlamda elektronik postalar, kullanım kolaylığı ve düşük maliyet gibi sebeplerle sıklıkla tercih edilmektedir. Ancak, e-posta kullanımının artmasıyla birlikte iletişim güvenliği riskleri de artmaktadır. En yaygın kötüye kullanım türlerinden olan istenmeyen e-postalar, kimlik avı, dolandırıcılık ve haksız kazanç gibi amaçlarla kullanılan zararlı elektronik iletilerdir. Güvenli iletişim için e-posta sistemlerini zararlı araçlardan korumak amacıyla sınıflandırma yöntemleri kullanılmakta ve sınıflandırma performansını artırmak için özellik seçimi çalışmaları yapılmaktadır. Özellik seçimi, sınıflandırma performansını etkileyen farklı tekniklerle en uygun ve önemli özellikleri seçerek başarılı sonuçlar elde etmeyi amaçlayan bir optimizasyon sürecidir. Bu çalışmada, spam sınıflandırma probleminin özellik seçim sürecinde ele alınabilecek filtreleme tabanlı tekniklerden birliktelik kuralı madenciliği, varyans eşiği gibi yöntemlerinin yanında; sarmal tekniklerden genetik algoritma ve karınca koloni optimizasyonu yaklaşımları kullanılmıştır. Farklı özellik alt kümelerine sahip veriler üzerinde yapılan deneylerde, özellik seçim süreçlerinden sonra yaklaşımların performans analizini yapabilmek için lojistik regresyon, rastgele orman, adaboost, karar ağacı ve derin sinir ağları gibi sınıflandırıcı algoritmaları kullanılmıştır. Genetik algoritmanın mutasyon oranı parametresi üzerinde yapılan deneyler, mutasyon oranının performansta etkili olduğunu ve %20 oranının kullanılmasıyla sonuçların iyileştirildiğini göstermektedir. Mutasyon oranı ‰1 olduğunda doğrusal destek vektör makineleri ile yaklaşık %87 doğruluk elde edilirken, %20 mutasyon oranında rastgele orman ile yaklaşık %96 doğruluk sağlanmıştır. Karınca koloni optimizasyonu ile doğrusal destek vektör makineleri birlikte kullanıldığında ise yaklaşık %99 duyarlılık oranı elde edilmiştir. Bu tez çalışması, özellik seçiminin sınıflandırma performansına doğrudan etki ettiğini ve hibrit yaklaşımların spam mesajlarının tespitinde başarıyı artırdığını ortaya koymaktadır.

Bilim Kodu	:	92402
Anahtar Kelimeler	:	Adaboost, derin sinir ağları, genetik algoritma, karınca koloni optimizasyonu, filtreleme yöntemleri, özellik seçimi
Sayfa Adedi	:	126
Danışman	:	Dr. Öğr. Üyesi Yılmaz ATAY

EFFECTIVE FEATURE SELECTION IN SPAM MAIL CLASSIFICATION PROBLEM

(M. Sc. Thesis)

Nuriye BAKTIR

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2023

ABSTRACT

In today's world, fast and secure communication is of great importance. In this context, electronic mails are frequently preferred due to reasons such as ease of use and low cost. However, with the increasing use of emails, communication security risks also increase. Unwanted emails, which are one of the most common forms of abuse, are harmful electronic messages used for purposes such as phishing, fraud, and unfair gain. Classification methods are employed to protect email systems from malicious tools and techniques, and feature selection studies are conducted to enhance classification performance. Feature selection is an optimization process that aims to select the most suitable and important features using different techniques that affect the classification performance. In this study, filtering-based techniques such as association rule mining and variance threshold, as well as spiral techniques such as genetic algorithms and ant colony optimization, were used in the feature selection process of the spam classification problem. Experiments were conducted on data with different subsets of features, and classifier algorithms such as logistic regression, random forest, adaboost, decision tree, and deep neural networks were utilized to analyze the performance of the approaches after the feature selection process. The experiments conducted on the mutation rate parameter of the genetic algorithm demonstrate that the mutation rate significantly affects the performance, and using a mutation rate of 20% improves the results. While a mutation rate of 1‰ yields an accuracy of approximately 87% with linear support vector machines, a mutation rate of 20% achieves an accuracy of approximately 96% with random forest. When ant colony optimization is combined with linear support vector machines, an approximate sensitivity rate of 99% is achieved. This thesis highlights the direct impact of feature selection on classification performance and demonstrates that hybrid approaches enhance the success of spam message detection.

Science Code	:	92402
Key Words	:	Adaboost, deep neural networks, genetic algorithm, ant colony optimization, filtering methods, feature selection
Page Number	:	126
Supervisor	:	Asst. Prof. Dr. Yılmaz ATAY

TEŞEKKÜR

Lisansüstü eğitim sürecince tez çalışmamın şekillenmesinde ve çalışmalarımın kontrolünde, yönlendirilmesinde, ihtiyaç halinde gerekli toplantıların yapılıp bilgilendirilmemde yardımcı olan Dr. Öğr. Üyesi Yılmaz ATAY'a sonsuz teşekkürlerimi sunuyorum.

Tez çalışmasının tamamlanmasında motive ederek dinamizm katan ve tez dokümanının dil üzerinde düzenlemelerde yardımcı olan sevgili eşim Ahmet Emin BAKTIR'a çok teşekkür ediyorum. Sadece çalışmam ve eğitim sürecimde değil her alanda yanımda olan benden desteklerini, ilgilerini, sevgilerini ve bana güvenlerini esirgemeyen aileme teşekkür ediyorum.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ x

ŞEKİLLERİN LİSTESİ xii

SİMGELER VE KISALTMALAR xiv

1. GİRİŞ 1

2. LİTERATÜR TARAMASI 5

2.1. Özellik Seçimi Gerçekleştirilmeden Yapılan Çalışmalar 6

2.2. Filtreleme Yöntemleri ile Yapılan Çalışmalar 11

2.3. Sarmal Yöntemler ile Yapılan Çalışmalar 14

2.3.1. Genetik algoritma ile yapılan çalışmalar 15

2.3.2. Karınca koloni optimizasyonu ile yapılan çalışmalar 18

3. METOT VE MATERYALLER 21

3.1. Önerilen Metot 21

3.2. Tez Çalışmasında Kullanılan Yaklaşımlar 23

3.3. Özellik Seçimi 25

3.3.1. Birliktelik kuralı madenciliği 30

3.3.2. Karşılıklı bilgi özellik seçimi 32

3.3.3. Varyans eşiği 34

3.3.4. Korelasyon özellik seçimi 34

3.3.5. Genetik algoritma 35

Sayfa

3.3.6. Karınca koloni optimizasyonu 39
3.4. Sınıflandırma Algoritmaları 44
3.4.1. Lojistik regresyon algoritması 44
3.4.2. Rastgele orman algoritması 46
3.4.3. Naive bayes algoritması 48
3.4.4. K-en yakın komşu algoritması 49
3.4.5. Adaboost sınıflandırma algoritması 51
3.4.6. Doğrusal destek vektör makineleri 52
3.4.7. Karar ağacı algoritması 54
3.4.8. Derin sinir ağları algoritması 55
3.5. Veri Seti 59
4. DENEYSEL ÇALIŞMALAR 61
4.1. Özellik Seçim Çalışması 61
4.1.1. Birliktelik kuralı madenciliği ile özellik seçimi 62
4.1.2. Karşılıklı bilgi özellik seçimi 64
4.1.3. Varyans eşik değeri ile özellik seçimi 65
4.1.4. Korelasyon özellik seçimi 66
4.1.5. Genetik algoritma ile özellik seçimi 67
4.1.6. Karınca koloni optimizasyonu ile özellik seçimi 73
4.2. Özellik Seçimi ile Sınıflandırma Çalışması 77
4.2.1. Lojistik regresyon ile sınıflandırma çalışması 78
4.2.2. Rastgele orman ile sınıflandırma çalışması 79
4.2.3. Naive bayes ile sınıflandırma çalışması 80
4.2.4. K-en yakın komşu ile sınıflandırma çalışması 81

Sayfa

4.2.5. Adaboost ile sınıflandırma çalışması 82
4.2.6. Doğrusal destek vektör makinesi ile sınıflandırma çalışması 83
4.2.7. Karar ağacı ile sınıflandırma çalışması 84
4.2.8. Derin sinir ağları ile sınıflandırma çalışması 85
4.2.9. Özellik seçimi yapılmadan sınıflandırma çalışması 89
4.3. Başarım Durumlarının Değerlendirilmesi 90
5. BULGULAR 93
5.1. Genetik Algoritma Farklı Mutasyon Değerlerinin Sonuçları 93
5.2. Karınca Koloni Optimizasyonu ile Alınan Sonuçlar 98
5.3. Çalışma İçerisindeki Tüm Uygulamaların Sonuçları 99
6. SONUÇ VE ÖNERİLER 103
KAYNAKLAR 109
EKLER 121
EK-1. Karşılıklı bilgi özellik seçim yöntemi ile özeliklerin sıralanışı 122
EK-2. Korelasyon özellik seçiminde niteliklerin birbiriyle ilişkisi 123
EK-3. Sarmal yöntemler ile sınıflandırma sonuçları 124
EK-4. Filtreleme yöntemleri ile sınıflandırma sonuçları 125
ÖZGEÇMİŞ 126

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 2.1. Özellik seçimi yapılmadan performans karşılaştırması..... 9

Çizelge 2.2. Makale çalışması sonuçları 11

Çizelge 2.3. Filtreleme yöntemleri ile özellik seçim çalışmaları 13

Çizelge 2.4. Genetik algoritma ile özellik seçim çalışmaları 17

Çizelge 2.5. Karınca koloni optimizasyonu ile özellik seçim çalışmaları..... 19

Çizelge 2.6. Farklı algoritmaların özellik seçim çalışmaları..... 20

Çizelge 3.1. Filtreleme yöntemleri 27

Çizelge 3.2. Özellik seçim yöntemleri..... 29

Çizelge 3.3. Genetik algoritmanın sözde kodu..... 39

Çizelge 3.4. Karınca koloni optimizasyonu sözde kod 43

Çizelge 3.5. Prova edilen k değerleri..... 50

Çizelge 3.6. Spambase veri setinin özellikleri..... 60

Çizelge 4.1. Varyans eşik değerine göre seçilen özellikler..... 65

Çizelge 4.2. Korelasyon eşik değerine göre seçilen özellikler..... 67

Çizelge 4.3. Genetik algoritmada kullanılan parametreler..... 68

Çizelge 4.4. ‰1 mutasyon ile elde edilen değerler..... 71

Çizelge 4.5. %1 mutasyon ile elde edilen değerler..... 72

Çizelge 4.6. %20 mutasyon ile elde edilen değerler..... 72

Çizelge 4.7. Karınca koloni optimizasyonunda kullanılan parametreler..... 74

Çizelge 4.8. Karınca koloni optimizasyonu ile elde edilen değerleri..... 77

Çizelge 4.9. Derin sinir ağları çoklu parametre listesi 86

Çizelge 4.10. Filtreleme yöntemleri ile DNN hibrit çalışmasının parametreleri..... 87

Çizelge 4.11. GA ile DNN hibrit çalışmasının parametreleri..... 88

Çizelge **Sayfa**

Çizelge 4.12. ACO ile DNN hibrit çalışmasının parametreleri 89

Çizelge 4.13. Özellik seçimi olmadan algoritmaların başarım değerleri 90

Çizelge 4.14. Performans değerlendirmesi için kullanılan metrikler 91

Çizelge 4.15. Performans değerlendirmesi fonksiyonları 92

Çizelge 5.1. Tüm algoritmaların f1-skor sonuç değerleri..... 100

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 3.1. Çalışma metodu 22

Şekil 3.2. Popülasyon 36

Şekil 3.3. Çaprazlama noktası 37

Şekil 3.4. Gen değişimi 38

Şekil 3.5. Çaprazlama sonrası yeni bireylerin belirlenmesi 38

Şekil 3.6. Genlerin mutasyona uğratılması 38

Şekil 3.7. Karınca koloni optimizasyonu yapısal gösterimi 41

Şekil 3.8. Lojistik regresyon sigmoid fonksiyonu 46

Şekil 3.9. Rastgele orman algoritması 48

Şekil 3.10. K-en yakın komşu algoritması 50

Şekil 3.11. Prova edilen k değerlerine karşılık doğruluk değerleri 50

Şekil 3.12 Adaboost algoritması 52

Şekil 3.13. Destek vektör makineleri algoritması 53

Şekil 3.14. Karar ağacı algoritması 55

Şekil 3.15. Derin sinir ağları algoritması 56

Şekil 3.16. Derin sinir ağları katmanlı yapısı 57

Şekil 4.1. Genetik algoritmasının akış diyagramı 69

Şekil 4.2. Karınca koloni optimizasyonu akış diyagramı 75

Şekil 4.3. Lojistik regresyon başarım değerlendirme 79

Şekil 4.4. Rastgele orman başarım değerlendirme 80

Şekil 4.5. Naive bayes başarım değerlendirme 81

Şekil 4.6. K-en yakın komşu başarım değerlendirme 82

Şekil 4.7. Adaboost başarım değerlendirme 83

Şekil **Sayfa**

Şekil 4.8. Doğrusal destek vektör makinesi başarım değerlendirme 84

Şekil 4.9. Karar ağacı başarım değerlendirme 85

Şekil 4.10. Derin sinir ağları başarım değerlendirme 89

Şekil 5.1. Genetik algoritma ‰1 mutasyon sonuçları 94

Şekil 5.2. Genetik algoritma %1 mutasyon sonuçları 95

Şekil 5.3. Genetik algoritma %20 mutasyon sonuçları 96

Şekil 5.4. GA doğruluk değerleri 96

Şekil 5.5. GA kesinlik değerleri 97

Şekil 5.6. GA duyarlılık metriği 97

Şekil 5.7. GA f1-skor metriği 98

Şekil 5.8. Karınca koloni optimizasyonu sonuçları 99

Şekil 5.9. Tüm algoritmaların f1-skor sonuçları 100

SİMGELER VE KISALTMALAR

Bu çalışma içerisinde kullanılmış olan kısaltmalar ve açıklamaları aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
Adaboost	Adaptive Boost Classifier- Adaboost Classifier
ANN	Yapay Sinir Ağı- Artificial neural network
BFA	İkili Ateşböceği Algoritması- Binary Firefly Algorithm
BNB	Bernoulli Naive Bayes
CSA	Klonal Seçim Algoritması- Clonal Selection Algorithm
C-SVC	C-Destek Vektör Sınıflandırıcı
DE	Diferansiyel Gelişim- Differential Evolution
DNN	Derin Sinir Ağları- Deep Neural Networks
DT	Karar Ağacı- Decision Tree
FA	Ateşböceği Algoritması- Firefly Algorithm
GA	Genetik Algoritma- Genetik Algorithm
GA‰1	Genetik Algoritma ‰1 mutasyon sonucu
GA%1	Genetik Algoritma %1 mutasyon sonucu
GA%20	Genetik Algoritma 5’ te 1 mutasyon sonucu
HHO	Harris Hawks Optimize Edici
HS	Harmoni Arama- Harmony Search
InfoGain	Bilgi Kazanımı- Information Gain
KNN	K-En Yakın Komşu- K-Nearest Neighbors
LR	Lojistik Regresyon- Logistic Regression
LSTM	Uzun Kısa Süreli Bellek- Long Short Term Memory
ML	Makine Öğrenimi- Machine Learning
NB	Naive Bayes
NN	Sinir Ağları- Neural Networks
OR	Olma Olasılığı- Odds Ratio
ÖSY	Özellik seçimi yapılmadan
RF	Rastgele Orman- Random Forest
RT	Rastgele Ağaç- Random Tree
SA	Tavlama Simülasyon- Simulating Annealing

Kısaltmalar	Açıklamalar
SVM	Destek Vektör Makineleri- Support Vector Machine
TS	Tabu Arama- Tabu Search
VT	Varyans Eşiği- Variance Threshold
WCFS	Su Döngüsü Özellik Seçimi- Water Cycle Feature
XGBoost	Aşırı Gradyan Artırma- Extreme Gradient Boosting

1. GİRİŞ

Elektronik posta, kullanımının kolaylığı, mesaj alışverişinin hızlı ve gönderme maliyetlerinin düşüklüğü nedeniyle yazışma veya dosya paylaşımı için kullanılmaktadır. E-posta hizmetinin faydalı işlevlerinin yanı sıra, kimlik avı, dolandırıcılık, vb. alanlarda kötü niyetli kullanımlarda mevcuttur. E-posta hizmet sağlayıcıları bu tür şüpheli mesajları filtreleyip istenmeyen elektronik posta (spam) olarak sınıflandırmaktadır [1]. Böylece kullanıcıya ilgili e-postanın ya önemsiz olduğunu ya da zararlı içeriğe sahip olduğunu bildirmektedir. Zararlı içeriğe sahip e-postalarda yer alan bağlantılara tıklanması, alıcıyı kimlik avı sitelerine yönlendirebilir veya kötü niyetli yazılımların bilgisayara indirilmesine neden olabilir [2-3]. Kullanıcıların zararlı içeriklerden korunabilmesi için e-postaların sınıflandırması, spam tanımlama veya filtreleme işlemleri yapılmalıdır.

Literatürde, e-posta sınıflandırması üzerine birçok çalışma bulunmaktadır. Bu çalışmalarda sınıflandırma performansını artırmak amacıyla veri setleri ön işleme tabi tutulmaktadır. Ön işleme adımlarının performansa etkisi, [4] numaralı çalışmada yapılan gözlemlerle ortaya konmuştur. Bu ön işlem adımları arasında gereksiz veya boş verilerin kaldırılması gibi düzenlemeler yer almaktadır. Ayrıca, veri seti üzerinde ön işlem adımları gerçekleştirilmeden özellik çıkarımı yapılarak sık kullanılan kelimelerin çalışmalara etkisi de tartışılmaktadır. Özellik çıkarımı, e-posta içerisinde geçen kelimelerin sıklığına dayanarak veri setlerinin yeniden düzenlenmesidir. Eğer bulunan kelime veya kelimeler e-posta içerisinde sıkça kullanılıyorsa, e-posta spam olarak sınıflandırılır; aksi takdirde ham olarak sınıflandırılır [5-6].

Özellik çıkarımı sırasında karşılaşılan önemli bir zorluk, özellik sayısının veri setindeki örnek sayısından daha fazla olmasıdır. Bu durum, aşırı boyutlu veri problemi olarak adlandırılmaktadır. Aşırı boyutlu veri setleri, sınıflandırma ve hesaplama süreçlerini karmaşıklaştırabilir. Bu zorluğun üstesinden gelmek için kullanılan bir boyut azaltma yaklaşımı, özellik seçimi yöntemidir. Özellik seçimi yöntemi, sınıflandırma doğruluğunu artırmaya ve hesaplama karmaşıklığını azaltmaya yöneliktir. Bu yöntem, veri setindeki önemsiz veya gereksiz özellikleri belirleyerek çıkarır ve sadece en bilgilendirici özellikleri kullanmayı hedefler. Böylece, sınıflandırma performansı geliştirilir ve gereksiz hesaplama maliyetleri azaltılır.

Spam e-posta sınıflandırma çalışmalarında kullanılan bazı veri setlerinde, ham veri yerine özellik çıkarımı yapılmış ve kullanıma hazır veri setleri bulunmaktadır [7]. Bu veri setleri üzerinde özellik filtreleme veya özellik seçimi gibi yöntemler uygulanabilmektedir. Bazı çalışmalarda, ön işlem adımları yapılmadan özellik seçimi yönteminin sınıflandırma performansına etkisini gözlemlemek amacıyla çeşitli araştırmalar gerçekleştirilmiştir [8-9]. Bu çalışmalar, özellik seçiminin sınıflandırma doğruluğu üzerindeki etkisini anlamak ve en etkili yöntemleri belirlemek için önemli bir katkı sağlamaktadır.

Özellik seçimi yöntemi, büyük veri analizlerinde yaygın olarak kullanılarak hedef sınıf ile en düşük benzerliğe ve en yüksek ilişkiye sahip özellik alt kümesinin elde edilmesini sağlar. Büyük veri setlerindeki gereksiz, ilgisiz veya gürültü içeren verileri ayıklamak suretiyle sınıflandırma performansını olumsuz etkileyen özellikler ayrıştırılır [10].

Özellik seçim yöntemlerinden filtreleme teknikleri arasında birliktelik kuralı madenciliği (association rule mining- ARM), bilgi kazanımı (information gain- InfoGain), ki-kare (chi-square), karşılıklı bilgi özellik seçimi (mutual information feature selection- MIFS) şeklinde sıralanabilir. Sarmal yöntem, makine öğrenmesi (machine learning- ML) algoritmalarının eğitim aşamasına dahil edilmesidir. Sarmal yöntemlerde metasezgisel ve makine öğrenimi algoritmaları birlikte kullanılmaktadır.

Spam e-posta sınıflandırma çalışmaları genellikle yaygın olarak kullanılan Spambase veri seti üzerinde gerçekleştirilmiştir. [11] numaralı referans çalışmada, UCI [7] üzerinden erişilebilen Spambase, Sonar, Arrhythmia, Madelon, Isolet ve Colon veri setleri üzerinde yarasa sürüsü, çiçek tozlaşma, karınca koloni optimizasyonu (Ant Colony Optimization- ACO), arı koloni optimizasyonu, parçacık sürüsü optimizasyonu, ateş böceği ve karıncalanma gibi sürü zekâsı temelli özellik seçimi yöntemleri incelenmiştir.

Tez çalışması kapsamında, birliktelik kuralı madenciliği, karşılıklı bilgi özellik seçimi, varyans eşiği (variance threshold- VT) ve korelasyon özellik seçimi (correlation feature selection- CFS) yöntemleri kullanılarak özellik filtreleme çalışmaları gerçekleştirilmektedir. Bu yöntemler, ön işlem adımlarını takiben Spambase veri seti üzerinde uygulanmıştır. Karşılıklı bilgi özellik seçimi, standart ölçüm fonksiyonu ön işlem tekniğiyle birlikte kullanılmaktadır. Varyans eşiği yönteminde, giriş verileri üzerinde tek seferde sığdırma ve dönüştürme işlemi yapılmaktadır. Korelasyon özellik seçimi yönteminde ise verilerin

korelasyonu hesaplanarak ön işleme tabi tutulmaktadır. Birliktelik kuralı madenciliği yöntemi ise veri setine ön işlem adımı uygulamadan gruplara bölünerek filtreleme tekniğiyle çalışmaktadır.

Sarmal yöntemlerde kullanılan genetik algoritma (genetic algorithm-GA) ve karınca koloni optimizasyonu (ant colony optimization-ACO) algoritmaları özellik seçimi tekniği olarak kullanılması sırasında ön işlem adımları uygulanmayarak çalışmalar ham veri üzerinde gerçekleştirilmektedir. Algoritmaların ürettiği sonuçların tutarlılığını ölçmek amacıyla belirli sayıda tekrar yaptırılarak standart bir çalışma gerçekleştirilebilmesine bakılmıştır[1]. Genetik algoritmada çeşitli mutasyon oranları kullanılarak bu değerlerin[2] çalışma performansına etkisi araştırılmaktadır.

Özellik seçim çalışmalarından sonra algoritmalar tarafından belirlenen özellikler ile sınıflandırıcıların performansı incelenmektedir. Adaboost (adaptive boost classifier-adaboost), doğrusal destek vektör makineleri (linear support vector machine- LSVM), derin sinir ağları (deep neural networks- DNN), karar ağacı (decision tree- DT), k-en yakın komşu (k-nearest neighbors- KNN), lojistik regresyon (logistic regression- LR), naive bayes (naive bayes- NB) ve rastgele orman (random forest- RF) algoritmaları sınıflandırma için kullanılmaktadır. Yazılmış olan makale çalışmasında karar ağacı, k-en yakın komşu, lojistik regresyon ve naive bayes algoritmaları kullanılarak Enron, SpamAssassion, CS440/ECE448 veri setleri üzerinde ön işlem adımları anlatılmıştır. Ön işlem adımlarından sonra özellik çıkarımı yapılarak sınıflandırma çalışması gerçekleştirilmiş. Performans değerlendirmesi metriklerle tablolar halinde sunulmuş [6]. Bu çalışma kapsamında performans değerlendirmesi için doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1-skor (F1-score) metrikleri kullanılmaktadır. Genetik algoritma ve karınca koloni optimizasyonu ile yapılan tekrarlı sınıflandırma çalışmaları sonucunda çeşitli performans değerlendirme sonuçları elde edilmektedir. Bu sonuçların en iyi, en kötü, ortalama, medyan ve standart sapma değerleri hesaplanarak raporlanmaktadır. Tablolar aracılığıyla en iyi sonuçlara sahip olanlar genetik algoritma ve karınca koloni optimizasyonu algoritmalarıyla gösterilmektedir.

Tez çalışmasının sunumu ile ilgili organizasyon burada verilmiştir. Sonraki bölümde tez çalışması kapsamında kullanılmış olan algoritmalar, özellik seçim yöntemleri ve veri setleri

[1] GA ve ACO için tanımlanan çalışma tekrar sayısı. Çalışma sayısı= 10
[2] ‰1, %1, %20 mutasyon oranları

ile gerçekleştirilen çalışmaların araştırılması sunulmuştur. Yapılan araştırma sonucunda incelenen çalışmaların özeti literatür taraması başlığı altında verilmiştir. Üçüncü bölümde çalışma içerisinde kullanılan metot ve materyallerden bahsedilmiştir. Kullanılan algoritmaların çalışma mantığı ilgili başlık altında sunulmuştur. Dördüncü başlık altında deneysel çalışmaların işlem adımları anlatılmış ve sonuçlar tablolar halinde sunulmuştur. Beşinci bölümde elde edilen çıktılara göre değerlendirmeler ve öneriler sunulmuştur.

2. LİTERATÜR TARAMASI

İnternetin günlük hayatın içinde giderek yaygınlaşması, bir takım güvenlik risklerini de beraberinde getirmektedir. Çevrimiçi platformlarda dolandırıcılık, kimlik avı hırsızlığı vb. suçlarda hızlı bir artış olduğu gözlemlenmektedir. Bu ve buna benzer zafiyetleri tespit edebilmek için farklı makine öğrenimi yaklaşımları kullanılmaktadır. Bu zafiyetlerin tespitinde; dolandırıcılık tespiti, izinsiz giriş tespiti, kötü amaçlı yazılım ve spam tespiti şeklinde mekanizmalar kullanılmaktadır [12–14].

Literatürde bulunan zafiyet tespit yöntemlerinden spam yapıların, filtrelenmesi ve sınıflandırılması araştırma yapılan ve geliştirilen bir alandır. Siber zafiyetlerin azaltılması için yapılan tespit ve önleme çalışmalarının yanında internet erişim hızını etkileyen faktörler arasında bant genişliği ve depolama alanının boyutu sayılabilir. Sistemin verimli bir şekilde çalışabilmesi için bu kaynakların kullanılmasında da tasarruflu davranılması gerekmektedir. Verimliliği arttırabilmek için zafiyet oluşturanların doğru sınıflandırılması gerekmektedir [15–17].

Emmanuel ve diğerleri Gmail, Yahoo ve Outlook servis sağlayıcılarının elektronik postaları spam olarak sınıflandırmada nasıl çalıştığından bahsetmişlerdir. Klasik sınıflandırma algoritmaları ile yapılan çalışmalar incelenmiş. Spambase dahil çeşitli veri setleri üzerinde yapılan çalışmaların özeti tablo halinde sunulmuştur [18].

Spam e-posta, güvenlik zafiyeti oluşmasına yol açacak başlıca açıklarından kimlik avı yapılmasına imkân veren ve engellenmesi için üzerinde yoğun olarak çalışılan bir oltalama tekniğidir. Bu bölümde Spambase veri seti kullanılarak yapılan, spam e-postaların sınıflandırılması için kullanılan yöntemler incelenmiştir. Özellik seçim yöntemlerinin ve sınıflandırma algoritmalarının literatürde hangi amaçlarla kullanıldıkları incelenmiştir. Klasik özellik seçim yöntemlerinin yanında metasezgisel algoritmaların özellik seçimi tekniğinde performansları araştırılmıştır. Spambase veri seti ile özellik seçimi kullanılarak ve kullanılmadan gerçekleştirilen sınıflandırma çalışmaları karşılaştırılmıştır. Son beş yılı içeren güncel çalışmaların araştırılmasına öncelik verilmiştir. Araştırmalar Google Scholar, IEEE Xplore, ScienceDirect ve Web of Science veri tabanları üzerinde yapılmıştır.

2.1. Özellik Seçimi Gerçekleştirilmeden Yapılan Çalışmalar

E-postaların, yüksek başarımlı bir şekilde sınıflandırılması güvenlik zafiyetleri ortadan kaldırmak için büyük önem arz etmektedir. Bunun üzerine duran Maryam Shuaib ve çalışma arkadaşları, birden fazla makine öğrenimi algoritması kullanarak sınıflandırma çalışması gerçekleştirmişlerdir. Spambase veri seti üzerinde yapılan çalışma sonucunda doğruluk ve f1-skor metriklerinde LR ve NB algoritmaları kullanılarak farklı sonuçlar elde edilmiştir. LR algoritmasında %93,1 ile başarılı bir çalışma gerçekleştirdiğini göstermişlerdir [19]. SVM tabanlı spam dedektörü çalışmasında %94,06'lık doğruluk değeriyle farklı çalışmalardan iyi olduğu sonucuna varılmıştır [20-21]. NB ile yapılan, spam e-posta sınıflandırma çalışması gerçekleştiren Ghosh ve diğerleri, yapılmış olan benzer sınıflandırma çalışmalarıyla karşılaştırmalar yapmışlardır. NB yönteminin tek başına kullandığında %79,57 oranında doğruluk değerine ulaşılmıştır. Algoritmanın modifikasyonu yapıldıktan sonra doğruluk oranı %89,12' ye yükseltilebilmiştir [22].

KNN algoritmasında bulanık, ağırlıklı ve değişken komşu sayısına sahip versiyonları bulabilmek amacıyla yapılan çalışmada balina optimizasyon algoritmasına dayalı bir yöntem önerilmiştir. Bunun yanında GA ile de çalışma tekrarlanarak karşılaştırma sonucunda balina optimizasyon algoritması (whale optimization algorithm- WOA) ile gerçekleştirilen çalışmanın, daha yüksek başarıma sahip olduğu gözlemlenmiştir. K=3 için klasik KNN algoritması sonucunda %69,13 doğruluk değeri elde edilmiştir. GA-KNN ve WOA-KNN hibrit çalışmaları sonucunda sırasıyla %86,41 ve %88,04 değerlerine ulaşılmış [23].

NB, DT, KNN ve LR sınıflandırıcılarının da arasında bulunduğu 10 adet algoritmanın Spambase veri seti üzerinde karşılaştırmalı çalışması yer almaktadır. Sistemin doğruluğunu sağlamak amacıyla 10 kat çapraz doğrulama kullanılmış. RF algoritması kullanılarak %95,45 doğruluk oranı ile en yüksek başarım değeri elde edilmiştir [24]. NB, RT, RF ve J48 algoritmaları kullanılarak yapılan farklı bir sınıflandırma çalışmasında RF %95,50 ile en başarılı sonucu vermektedir [25]. UCI web sayfasından farklı veri setleri kullanarak yapılan çalışmada Spambase veri seti üzerinde SVM ile %90'luk doğruluk oranı elde edilmiştir [26-27]. E-posta sınıflandırması için LR, KNN, NB, DT, Adaboost, ANN ve SVM algoritmalarını kullanılan çalışmada topluluk yöntemlerinin, diğer yöntemlerden daha başarılı olduğu gözlemlenmektedir [28]. Nandhiniand ve Marseline güvenlik zafiyetine sebep olan spam e-postları sınıflandırabilmek amacıyla yaygın olarak kullanılan

algoritmaların etkili yöntemlerini sunmaktadır. Bunun için NB, LR, RT, KNN ve SVM algoritmaları kullanılmaktadır. Spambase üzerinde yapılan sınıflandırma çalışması sonucunda doğruluk, f1-skor, kesinlik ve duyarlılık metriklerinde sonuçlar alınmış. LR ile %93.1, NB ile %79.52, KNN ile %99.9, SVM ile %90.76 doğruluk oranlarına sahip başarım değerleri kaydedilmiştir [29].

Tek fazlı sınıflandırma çalışmasından faklı olarak yapılan hibrit sınıflandırma çalışmasında topluluk sınıflandırıcı (ensemble calssification- EC), DT, RF ve SVM kullanılarak literatüre yeni bir çalışma eklenmiş. İlk olarak DT ile e-posta sınıflandırılmış. Sonrasında bahsedilen algoritmaların tamamı kullanılarak üretilen çıktının doğaçlama sınıflandırılması gerçekleştirilmiştir. Çalışmanın sonunda doğruluk metriğine bağlı olarak elde edilen başarım sonuçları sırasıyla %97,8-99,77-98,62-99,88 şeklindedir [30].

Büyük veri her alanda günümüzün kaçınılmaz bir gerçeği haline gelmiştir. Büyük verilerden anlamlı sonuçlar çıkarılarak farklı alanlarda kullanılabilmektedir. Kişiselleştirilmiş reklamlar bunun güzel bir örneğidir. Makine öğrenimi algoritmaları, bu tür analizlerin yapılmasında sıklıkla başvurulan tekniklerdendir. Bunların yanında yapay zekâ algoritmaları da sınıflandırıcı olarak kullanılmaktadır. Spambase veri seti dahil farklı veri setleri üzerinde yapılan çalışmada NB, DT, RF, gradyan arttırılmış karar ağaçları ve evrişimli sinir ağı kullanılarak sınıflandırma algoritmalarının performansları tartışılmıştır. RF algoritmasının diğerlerinden daha iyi bir doğruluk değerine sahip olduğu gözlemlenmiştir [31].

Derin sinir ağları, katmanlardan oluşan ve bu katmanlar arasında tam bağlantıya sahip yapay sinir ağlarıdır. DNN ile yapılan e-posta sınıflandırma çalışmasının performansını ölçebilmek amacıyla, 2 gizli katmanlı yapay sinir ağı oluşturularak yapılan çalışmada her katmandaki düğüm sayısı 30 olarak belirlenmiştir. Eğitim için kullanılan veri setleri farklı büyüklüklerde olmasının performans üzerindeki etkileri incelenmiştir. Çalışmanın sonucunda NB ile yapılan sınıflandırmanın daha iyi sonuçlara sahip olduğu gözlemlenmiştir [32]. NB, DT, SVM, KNN, RF ve ANN yöntemlerinin, PU, Spambase, Enron, SpamAssasin, TREC, CCERT, LingSpam veri setleri üzerinde başarım durumlarının karşılaştırması özetlenmiştir. Diğer algoritmalara göre RF daha başarılı sonuç vermiştir [33]. Yapay Sinir Ağlarında kullanılan parametreler performansı önemli oranda etkilemektedir. Parametreleri belirleyebilmek adına yapılan [34] çalışmasında DNN parametreleri evrimsel algoritma olan

su dalgası optimizasyonundan faydalanılarak belirlenmiştir. Sınıflandırma doğruluğu Spambase veri setini üzerinde test etmişlerdir.

Etiket çevirme, sınıflandırma performansını azaltmak için eğitim örneklerinin etiketlerini değiştiren yeni bir zehirleme saldırısı türüdür. Bunun için Spambase veri seti kullanılarak test çalışması yapılmıştır. Beş klasik sınıflandırma algoritması kullanılarak çalışma gerçekleştirilmiştir [35-36]. Dolandırıcılık faaliyetlerini tespit etmek amacıyla e-posta sınıflandırma modeli geliştirilmiştir [37]. Botnet spam saldırıları için derin öğrenme ile SVM hibrit çalışması kullanılarak yapılan sınıflandırma çalışmasında %98,7'lik doğruluğa ulaşılmıştır [38].

Spam e-postaların sınıflandırılması yapılırken veri setinde bulunan özelliklerin sınıflandırmaya etkisi araştırılmıştır. Özelikler filtrelenerek sınıflandırmaya etkisi olmayan veya etkisinin az olduğu düşünülen özelliklerin veri setinden ayrıştırılarak sınıflandırma performansı artırılmak istenmektedir. Sınıflandırmanın, yüksek başarımda yapılmasının yanında kısa sürede yapılması da önem arz etmektedir. Çok sayıda özelliğe sahip elektronik postaların, özellik seçimi ile özelliklerin azaltılmasının performansa etkisi tartışılmıştır. Al-Rawashdeh, Mamat ve Abd Rahim tarafından yapılan çalışmada özellik seçiminde su döngüsü ve simüle tavlama metasezgisel optimizasyon yöntemi kullanmışlardır. KNN, NB ve SVM ile sınıflandırma çalışması yapılmış. Sonuçta en başarılı algoritmanın SVM ile %94,3 olduğu gözlemlenmiştir [39].

Spam tespit çalışmasında özellik seçimi yapılırken, belirli sayıda özellikler kullanılarak yapılan çalışmada dağınık arama algoritması ve KNN hibrit yöntemini sunmuşlardır. Özellik seçimi yapılmadan, KNN ile yapılan çalışma sonuncunda doğruluk metriğinde %71,94 başarım elde edilmiştir. 15 adet özellik seçildiğinde %98,92 değeri, 57 özelliğin hepsi seçildiğinde %94,54 doğruluk değeri elde edildiği gösterilmiştir [40]. Gereksiz özellikleri filtreleyerek yapılan çalışmada birbirinden farklı sonuçlarla karşılaştırmalar yapılmıştır. Spambase dahil üç farklı veri seti ile önerilen yöntemler tekrarlanmıştır. Filtreleme yöntemi ve sonrasında NB ve RF kombinasyonu ile %97,79 doğruluk değerine ulaşılmıştır [41].

Çizelge 2.1. Özellik seçimi yapılmadan performans karşılaştırması

Çalışma	Sınıflandırıcı	Doğruluk	Duyarlılık	Kesinlik	F1-skor
2018 [24]	RF	95,45	95,5	95,5	95,4
	ANN	92,41	92,4	92,4	92,4
	LR	92,41	92,4	92,4	92,4
	SVM	91,89	91,9	91,9	91,9
	KNN	90,78	90,8	90,8	90,8
	NB	89,85	89,9	89,9	89,8
2019 [30]	DT	99,88	-	-	-
	SVM	98,62			
	RF	99,77			
	EC	97,8			
2020 [29]	LR	93,13	93,1	93,1	93,1
	RF	99,93	99,9	99,9	99,9
	NB	79,52	84,5	79,5	79,7
	KNN	99,93	99,9	99,9	99,9
	SVM	90,76	11,7	90,8	90,7
2022 [42]	NB	94,47	94,82	91,16	92,95
	RF	95	92,97	95,02	93,98
	DT	95,21	95,21	83,61	89,03
	LR	97,71	98,3	96,13	97,2
	SVM	98	98,31	96,68	97,49
	KNN	90,87	91,12	85,55	88,25
2023 [43]	C4.5	91,28	-	-	-
	ID3	88,8			
	C-SVC	93,13			
	NB	86,76			
2023 [44]	ABC	90,84	90,8	90,8	90,8
	ANN	94	94	94	94
	DT	92,9	92,9	92,9	92,9
	KNN	90,78	90,8	90,8	90,8
	LR	93,13	93,1	93,1	93,1
	NB	79,53	84,5	79,5	79,7
	RF	99,93	99,9	99,9	99,9
	SVM	94	94	94	94
2023[Tez önerisi]	LR	91,75	91,75	88,46	90,08
	RF	95,55	97,55	91,8	94,58
	NB	88,05	90,7	80	85,01
	KNN	78,83	78,1	69,49	73,54
	Adaboost	93,54	92,54	92,3	92,42
	LSVM	84,25	78,16	87,18	82,42
	DT	91,2	92,56	86,15	89,24
	DNN	93,81	92,36	93,07	92,72

E-postalarda var olan, özellikler arasında bulunan gereksiz özelliklerin sınıflandırmayı olumsuz etkilediğini söyleyen [45] çalışmada, temel bileşen analizi ile özellik seçimi yapılarak GA ve DT algoritmasının hibrit çalışması sunulmaktadır. Yapılan çalışmada özellik seçimi yapılmadan Spambase veri seti üzerinde NB, SVM, KNN ve J-48 algoritmaları çalıştırılmıştır. Doğruluk ve f1-skor metrikleri sonucunda NB ile %79,2 ve

%80,1, SVM ile %89,8 ve %89,7, KNN %89,8 ve %89,8 başarım değerlerine ulaşılmıştır. Gerçekleştirilen hibrit çalışma sonucunda %93,4 doğruluk oranı ve f1-skor değerleri kaydedilmiştir. DNN kullanılarak yapılan, sınıflandırma çalışmasında özellik seçimi uygulanarak diğer sınıflandırma algoritmalarıyla karşılaştırılmıştır. Hata oranları verilen makalede DNN sonucu %98,12 doğruluk oranına sahip olduğu gözlemlenmiştir [46].

Çizelge 2.1 ile gösterilen tabloda Spambase veri seti ile özellik seçimi yapılmadan kullanılmış olan algoritmaların performansları gösterilmiştir. Çalışma içerisinde 4 başarım değerlendirme kriteri ele alındığından dolayı karşılaştırmalar bu şekilde yapılmaktadır. İncelenen çalışmaların her birinde farklı sonuçlar gösterilmiştir. En başarılı algoritma dağılımı çalışmalara göre değişmektedir. Tez çalışmasında Adaboost ile yapılan sınıflandırma çalışmasında %93,54 doğruluk değerinin diğer çalışmadan daha iyi sonuç verdiği gösterilmiştir. Diğer çalışmalar incelendiğinde DNN algoritmasının kullanılmadığı gözlemlenmiştir. Bunun yerine ANN algoritması ile karşılaşılmıştır. ANN algoritması ile klasik sınıflandırıcı algoritmaları aynı çalışma içerinde kullanarak literatüre yenilik katılmıştır. Tüm çalışmalarda dört farklı değerlendirme içerisinde en iyi başarım değerine sahip sonuçlar koyu renk ile gösterilmiştir.

Baktır ve Atay tarafından yazılan makale çalışmasında, özelliğe sahip olmayan ham veri seti üzerinde gerçekleştirilen ön işlem adımlarıyla klasik sınıflandırma algoritmalarının performansını değerlendirmiştir. Sınıflandırma algoritması olarak karar ağacı, k-en yakın komşu, lojistik regresyon ve naive bayes kullanılmıştır. Birbirinden farklı niteliklere SpamAssassin[3], Enron 4, 5, 6 [47] ve CS440/ECE448[4] sahip veri setleri üzerinde değerlendirme yapılmıştır. Bu veri setleri farklı sayıda veri ve farklı oranda ham/spam e-posta içermektedir. Veri setlerinde yapılan ön işlem adımları, karakter işlemleri, etkisiz elemanların kaldırılması, köklendirme ve kelime frekanslarının hesaplanması gibi işlemleri içermiştir. Bu adımlar, veri setinin daha temiz ve daha uygun hale getirilerek spam sınıflandırma çalışmalarında daha iyi sonuçlar elde edilmesine yardımcı olmuştur.

Başarım tablosu, Çizelge 2.2' de gösterilmektedir. Bu tablo, [6] çalışmasından alınmıştır. Kullanılmış olan veri setleri ile hangi algoritmaların daha başarılı bir sonuç verdiği gösterilmektedir. SpamAssassin veri seti ile en başarılı sonucun %98,32 kesinlik değeri ile

[3] https://spamassassin.apache.org/ Son Erişim Tarihi:04.11.2021
[4] https://courses.grainger.illinois.edu/cs440/fa2018/MPs/mp4/assignment4.html Son Erişim Tarihi:02.12.2021

KNN algoritması olduğu gözlemlenmektedir. Enron 4 veri setinde en başarılı sonuçlar LR algoritmasına aittir. Veri setine ait en başarılı sonucun %99,77 duyarlılık metriğine ait olduğu gözlemlenmektedir. Enron 5 ve 6 veri setlerinde metrik sıralamasında benzer algoritmaların başarılı sonuçları tabloda gösterilmektedir. Kesinlik metriği ile NB algoritmasının Enron 5 veri setinde başarılı olduğu, duyarlılık metriğinde LR algoritmasının Enron 6 veri setinde başarılı olduğu gözlemlenmektedir. CS440/ECE448 veri setinde çalışmanın en başarılı sonucu olan %100 ile KNN ve LR algoritmalarının iyi çalıştığı gözlemlenmektedir.

Deney çalışması, farklı veri setleri ve algoritmaların kullanıldığı denemelerle gerçekleştirilmiştir. Sonuçlar, 4 farklı metrik kullanılarak elde edilmiştir. Veri seti bazlı bir değerlendirme yapıldığında, bu rakamların ve ifadelerin farklılık göstereceği öngörülmektedir. Bu durum, farklı veri setleri ve algoritmaların performansının çeşitlilik gösterebileceğini vurgulamaktadır. Her veri seti farklı özelliklere, dağılımlara ve karmaşıklıklara sahip olabilir, bu da algoritmaların başarılarını etkileyebilir. Ayrıca, farklı algoritmaların farklı öğrenme yaklaşımları ve hiper parametre ayarlarına sahip olmaları, sonuçların değişmesine neden olabilir. Dolayısıyla, bir veri seti üzerinde elde edilen sonuçların başka bir veri setinde aynı olmayabileceği göz önünde bulundurulmalıdır.

Çizelge 2.2. Makale çalışması sonuçları

Veri seti / Metrik	SpamAssassin		Enron 4		Enron 5		Enron 6		CS440/ECE448	
Doğruluk	LR	94,54	LR	99,08	LR	99,42	LR	97,83	LR	98,92
Kesinlik	KNN	98,32	LR	98,97	NB	99,8	NB	99,26	KNN	100
Duyarlılık	LR	93,78	LR	99,77	LR	99,72	LR	99,65	LR	100
F1-skor	LR	91,95	LR	99,37	LR	99,58	LR	98,52	LR	98,97

2.2. Filtreleme Yöntemleri ile Yapılan Çalışmalar

Özellik seçimi yöntemlerinden olan filtreleme, süreç boyunca herhangi bir sınıflandırma algoritması kullanılmadan, eğitim verilerinin genelleştirilmesine dayanmaktadır. Bu yaklaşımın avantajları, nispeten düşük hesaplama karmaşıklığı, yeterince genelleme yeteneği ve sınıflandırıcıdan bağımsızlığı olarak sıralanabilir. Başlıca dezavantajı, özelliklerin birbirine bağlantısı olmadan seçilmesidir.

Temel özellik seçim yöntemlerinden olan filtreleme yöntemlerini kullanarak yapılan çalışmada çeşitli veri setleri kullanılarak karşılaştırmalar yapılmıştır. Belirlenen eşik değerine göre yüksek niteliklere sahip özellikler seçilmiştir. Bu özelliklerle KNN ve SVM algoritmaları kullanılarak sınıflandırma performansları ölçülmüş. Klasik filtreleme yöntemlerinin yanında metasezgisel algoritmalardan parçacık sürü optimizasyonu (Particle Swarm Optimization- PSO) ve ACO algoritmalarla yapılan hibrit özellik seçim yöntemleri de sunulmuş. Çalışma sonucunda farklı kıyaslamalar elde edilmiş. MIFS ile yapılan SVM ve KNN sınıflandırma uygulaması sonucunda %85,4 ve %83,9 doğruluk oranlarına ulaşılmıştır. Graf kümeleme karınca koloni optimizasyon (graf clustering ant colony optimization- GCACO) özellik seçiminin sonucunda SVM ile 91,6, KNN algoritması ile %87,2 doğruluk oranları elde edilmiştir [48]. Karşılıklı bilgi özellik seçim yönteminde, özelliklerin uygun ayrıştırılması, önemli bir kriterdir. Bu probleme çözüm olabilmek amacıyla ön işlem adımı olarak ortak önyargı düzeltilmiş karşılıklı bilgi (joint bias corrected mutual ınformation- JBMI) kullanılarak kare tabanlı arama yöntemi ve JBMI önerilmiş. Klasik MIFS kullanılarak yapılan çalışma sonucunda %74,5 doğruluk oranı elde edilirken, önerilen çalışma sonucunda SVM %85,5 ve KNN %85,4 değerlerine ulaşılmıştır [49].

Filtreleme özellik seçim yöntemleri ile yapılan çalışmada LR algoritması kullanılarak özellik seçimi yapılmış ve yapılmamış deneyler karşılaştırılmıştır. Korelasyon özellik seçimi yöntemi sonucunda, doğruluk metriğinde %89 başarım göstermiştir [50]. Filtreleme yöntemleri kullanılarak özellikleri filtreleyen bir çalışmada NB, SVM ve KNN algoritmalarıyla sınıflandırma çalışması gerçekleştirilmiş [51]. Özellikler arasındaki korelasyon sayısını ölçüt olarak alan çalışmada MIFS ile hibrit özelik seçim çalışması gerçekleştirilmiş. Özellikler arasındaki fazlalık dalgalanması arttığında genelleme yeteneğinin arttığı gözlemlenmiş. Spambase veri setinin de içinde olduğu 10 adet veri seti üzerinde çalışma tekrarlanmış. MI-CFS ile SVM, KNN ve NB algoritmaları çalıştırılmış. Sırasıyla %87,18 %86,64 %89,3 doğruluk değerlerine ulaşılmış [52].

Adaboost algoritmasının sınıflandırma performansını, arttırabilmek için ağırlıklı özellik seçim tabanlı adaboost yöntemi önerilmiş. Geleneksel filtreleme yöntemlerinin özellik seçimlerine dayanarak yapılan yanlış sınıflandırmalarda etkili bir şekilde özellik seçimi yapabilmek amacıyla yeni bir yaklaşım önerilmiştir. Önerilen yöntemin klasik yönteme göre %87,9 doğruluk oranı ile daha başarılı olduğu gösterilmiştir [53].

Çizelge 2.3. Filtreleme yöntemleri ile özellik seçim çalışmaları

Çalışma	Özellik Seçimi	Sınıflandırma	Seçilen Özellik	Doğruluk	F1-skor
2018 [48]	MIFS	SVM	-	85,4	-
		KNN		83,9	
2019 [52]	MI-CFS	SVM	-	87,18	-
		KNN		86,64	
		NB		89,3	
2019 [54]	CFS	KNN	16	80,5	-
		DT		74,3	
		NB		80	
		XGBoost		71,4	
2022 [55]	CFS	J48	-	92,21	92,2
		RF		93,45	93,5
		NN		91,98	92
		NB		78,8	78,9
		KNN		91,63	91,6
		LSTM		93,52	93,5
2022 [55]	ARM	J48	-	83,67	83,3
		RF		85,56	85,4
		NN		80,63	80,3
		NB		79,11	78,9
		KNN		83,09	82,9
		LSTM		83,3	89,9
2023 [Tez önerisi]	ARM	LR	33	88,71	86,10
		RF		92,73	91,22
		NB		85,23	82,06
		KNN		89,14	87,18
		Adaboost		90,88	89,20
		LSVM		88,71	86,02
		DT		87,62	85,31
		DNN		92,73	91,49
	MIFS	LR	40	91,21	89,19
		RF		95,44	94,49
		NB		88,82	85,79
		KNN		88,17	85,41
		Adaboost		93,59	92,45
		LSVM		91,21	89,19
		DT		90,99	88,86
		DNN		91,97	90,93
	VT	LR	41	91,31	89,72
		RF		95,44	94,47
		NB		89,03	86,48
		KNN		79,26	74,29
		Adaboost		93,49	92,23
		LSVM		82,63	81,22
		DT		89,14	87,08
		DNN		94,79	93,75
	CFS	LR	43	91,75	90,03
		RF		95,01	93,96
		NB		89,25	86,57
		KNN		78,61	73,63
		Adaboost		93,38	92,15
		LSVM		83,93	77,44
		DT		90,01	87,80
		DNN		93,70	92,35

Klasik özellik seçim yöntemlerinden InfoGain kullanılarak yapılan çalışmada sınıflandırma algoritması olarak topluluk yöntemleri karşılaştırılmıştır. Hibrit topluluk sınıflandırıcı, NB ve DT algoritmalarına göre daha başarılı sonuç vermiş [56]. Farklı bir çalışmada Gini algoritması kullanılarak özellikler ölçeklendirilir ve RF algoritması kullanılarak özellikler derecelendirilir. DNN sınıflandırıcı ile eğitim için ayrılmış veriler eğitilir. En üst değerdeki, beş özellik dikkate alındığında %88,6 doğruluk oranı ile DNN sınıflandırıcının KNN ve SVM algoritmalarından daha iyi sonuç verdiği gözlemlenmiş [57].

Varyans eşik değeri ile özellik seçimi çalışması literatürde çok rastlanılmamıştır. Yapılan bir çalışmada filtreleme ve sarmal yöntemler karşılaştırılarak bazı sınıflandırma algoritmalarıyla amaçlanan özellik ortalamaları hesaplanmış. Varyans eşiği yönteminin de içinde olduğu filtreleme yöntemlerinde özellik seçim gerçekleştirilmiş. Eşik değeri 0,4 seçilerek 10 adet özellik seçilmiş [58].

Çizelge 2.3 de son beş yıl içerisinde gerçekleştirilmiş olan filtreleme yöntemleri ile özellik seçim çalışmaları gösterilmiştir. Temel filtreleme yöntemlerinin bir arada bulunduğu tek bir çalışmanın olmadığı görülmektedir. Her çalışmada farklı bir yaklaşım sunulmuş. ARM yöntemi ile özellik seçim çalışmasında tez önerisinde sunulan yöntemin daha başarılı sonuçlar verdiği gösterilmiştir. MIFS yöntemi ile yapılmış olan [52] çalışmada NB ile elde edilen sonucun tez önerisinden fazla olduğu görüntülenmektedir. CFS yönteminde [55] çalışmasına ait sonuçta J48 ile KNN algoritmalarının tez önerisinden daha iyi olduğu görüntülenmektedir. Diğer algoritmalar tez önerisinde daha başarılı sonuçlanmıştır. Karşılaştırma tablosunda iyi değere sahip sonuçlar koyu renk ile vurgulanmıştır.

2.3. Sarmal Yöntemler ile Yapılan Çalışmalar

Sarmal yöntemler, özellik seçim sürecinde sınıflandırıcı oluşturma prosedürünü içermektedir ve seçilen özellik alt kümesini oluşturabilmek için makine öğreniminin tahmin etme doğruluğunu kullanmaktadır. Sınıflandırıcı ile etkileşiminden dolayı genellikle filtreleme yönteminden daha iyi sonuçlar vermektedir. Ancak, yöntemin hesaplama karmaşıklığını arttırmaktadır ve fazla uydurma riski oluşmaktadır [59].

Sarmal yöntem içerisinde genetik algoritma- rastgele orman regresyon ve karınca koloni optimizasyonu- rastgele orman sınıflandırıcı hibrit çalışmaları gerçekleştirilir. Literatür

araştırması içerisinde Spambase veri seti ile yapılmış genetik algoritma ve karınca koloni optimizasyonu özellik seçim çalışmaları yer almaktadır.

2.3.1. Genetik algoritma ile yapılan çalışmalar

Evrimsel algoritmalardan GA, çalışma içerisinde kullanılan parametrelerin belirlenmesi için Spambase dahil çeşitli veri setleri üzerinde kullanılmıştır. Popülasyon sayısı, jenerasyon sayısı, çaprazlama oranı, mutasyon oranı ve turnuva sayısını belirleyen deneyler yapılmıştır. Spambase veri setinde hangi parametrelerin iyi çalışacağını gösteren doğruluk oranı verilmiştir [60].

Elektronik postaların sınıflandırılması işlemi, gelişmekte olan evrimsel ve sürü tabanlı yöntemler için kullanılmaktadır. Yapılan çalışmada akıllı su damlaları algoritması (ıntelligent water drops algorithm- IWDA) kullanılarak özellik seçim çalışması yapılmıştır. Evrimsel algoritmalardan GA, sürü tabanlı yöntem olan ACO kullanılarak karşılaştırmalar yapılmıştır. Sınıflandırıcı olarak NB algoritmasında faydalanılmıştır. IWDA kullanılarak yapılan çalışma sonucunda %94 oranında doğruluk oranı elde edilmiş [61].

Sınıflandırmanın etkililiği ortaya koymak için yapılan çalışmada genetik algoritmanın yanında, HS, PSO ve SA algoritmaları özellik seçim yöntemi olarak kullanmışlardır. KNN, NB ve SVM algoritmaları ile özellik seçimi yapılarak ve yapılmadan performans etkilerini tartışılmıştır. Spambase veri seti içerisinde GA kullanılarak gereksiz özellikler kaldırılmış, KNN ile %87,7 NB ile %54,8 SVM ile %90,61 doğruluk oranlarına ulaşılmıştır [62].

Farklı bir çalışmada özellik seçimi yapılmadan NB ile sınıflandırma işlemi gerçekleştirilmiş. GA kullanılarak özellik seçimi işlemi yapılmış, NB algoritması ile sınıflandırma işlemi tekrarlanınca daha başarılı sonuçlar alındığı gözlemlenmiştir. Farklı adım sayısı ile denenen çalışma, turnuva ve rastgele seçim yöntemleri kullanılarak genetik algoritmanın işlem adımları gerçekleştirilmiştir. Adım sayısı değiştikçe seçilen özellik sayısında da değişim olduğu gözlemlenmiştir. Seçilen özelliklere göre başarım oranları değişiklik göstermiştir. 20 adım sayısı kullanılarak seçilen 33 özelliğin en iyi başarım oranına sahip olduğu gözlemlenmiştir [63].

Spam tespit çalışmasını hızlandırdığını iddiasında bulunan Sharaff ve Gupta tarafından sınıflandırma işleminde ikili parçacık sürü kullanılarak özellik seçimin yapılmasını önermişlerdir. Bu çalışmada PSO ve GA ile karşılaştırma yapılmış. RF, ekstra ağaç (extra tree- ET) ve DT yöntemleriyle sınıflandırma yapılmış. GA ve RF ile çalışmanın sonucunda %91,5 doğruluk oranı elde edilmiş [64]. Genetik algoritmanın geliştirilmesiyle yapılan çalışmada çaprazlama oranı olarak 0,8 ile mutasyon oranı olarak 0,02 kullanılmıştır. SVM ile gerçekleştirilen sınıflandırma çalışmasında %91,20 doğruluk oranı elde edilmiştir [65].

Özellik seçim yaklaşımlarının, çok uzun hesaplama süresine ve büyük popülasyon boyutlarına sahip olması deney sonuçlarını olumsuz etkilemektedir. Sınıflandırma doğruluğunu iyileştirmeyi amaçlayan çalışmada sinir ağları ve genetik algoritma tabanlı özellik seçim yaklaşımı önerilmiş. Genetik algoritma için rastgele oluşturulan başlangıç popülasyonundan sonra uygunluk fonksiyonu sinir ağları ile hesaplanmıştır. Sınıflandırma doğrulukları elde edilmiş, düşük değere sahip olanlar elenmiş, geriye kalanlar GA içerisinde kullanılmıştır. Spambase veri seti için doğruluk oranı %91,92 şeklinde kaydedilmiştir [66].

Topluluk tespiti tabanlı genetik algoritma (community detection-based genetic algorithm for feature selection- CDGA) uygulanarak özellik seçimi yapılmıştır. Önerilen yöntemin, metasezgisel algoritmalardan daha başarılı olduğunu gösterebilmek amacıyla PSO, ACO ile özellik seçim çalışması tekrarlanmıştır. Sınıflandırma algoritması olarak KNN, SVM ve adaboost kullanılmış, sırasıyla %93,99 %93,68 ve %93,27 doğruluk oranlarıyla diğerlerinden daha başarılı olduğu gösterilmiştir. ACO ile yapılan çalışmanın başarım oranları, sırasıyla %91,81, %89,51 ve %90,89 şeklinde olduğu gözlemlenmiştir [67]. Öğrenmeye dayalı özellik seçim yöntemi öneren çalışmada GA kullanarak yaptığı karşılaştırma sonucunda %91,68 doğruluk oranına ulaşılmıştır. Çalışma kendi amaçladığı özellik seçim yöntemi ile %93,11 oranında doğruluk oranı elde etmiştir [68].

Özellik seçiminin yanında sınıflandırmanın da önemli olduğunu vurgulamak isteyen Sokhangoee ve Rezapour tarafından, özellik seçimi olarak GA ile ARM hibrit yöntemini kullanılarak yapılmıştır. Bu çalışmada filtreleme yöntemlerinden CFS ve bilgi kazanımı yöntemleri ile yapılan özellik seçiminin başarım oranları karşılaştırılmıştır. Çeşitli sınıflandırma algoritmaları kullanılarak başarım oranları elde edilmiştir. Spambase veri seti üzerinde yapılan çalışmada GA ile özellik seçip RF ile sınıflandırma işlemi gerçekleştirilmiş, sonuç olarak %94,82 doğruluk oranı, %94,7 f1-skor oranı elde edilmiştir. Çok az özellik

seçerek yapılan yüksek başarımlar alındığı görülmektedir. Özellik sayısındaki değişimler başarım oranlarını etkilemektedir [55].

Çizelge 2.4. Genetik algoritma ile özellik seçim çalışmaları

Çalışma	Özellik Seçimi	Sınıflandırma	Seçilen Özellik	Doğruluk	F1-skor
2018 [61]	GA	NB	-	77	75
2019 [64]	GA	RF		91.5	93
		ET		91	92,5
		DT		90,3	92
2019 [62]	GA	KNN	32	87,7	84,3
		NB	48	54,8	63,5
		SVM	32	90,6	88
2019 [63]	GA	NB	30	95,2	-
			33	96,7	
2021 [67]	CDGA	KNN	8.17(ort)	93,99	-
		SVM		93,68	
		Adaboost		93,27	
2022 [55]	GA	J48	54	96,56	96,4
		RF		99,04	99
		NN		94,5	94,3
		NB		94,78	94,6
		KNN		96,45	96,4
		LSTM		90,65	90,6
2023 [Tez önerisi]	GA‰1	LR	32,4(ort)	90,99	89,18
		RF		95,98	95,2
		NB		90,01	87,5
		KNN		88,27	85,79
		Adaboost		93,7	92,41
		LSVM		87,62	85,35
		DT		93,59	92,31
		DNN		93,7	92,31
	GA%1	LR	33,13(ort)	90,23	88,19
		RF		95,77	94,9
		NB		90,45	87,95
		KNN		89,03	86,83
		Adaboost		93,92	92,69
		LSVM		87,95	85,53
		DT		92,83	91,41
		DNN		94,68	93,63
	GA%20	LR	32,4(ort)	91,53	89,71
		RF		96,09	95,29
		NB		91,1	88,95
		KNN		90,12	88,32
		Adaboost		94,57	93,57
		LSVM		89,47	87,86
		DT		94,57	93,47
		DNN		95,01	94,06

Çizelge 2.4 ile GA özellik seçim çalışmalarının performansları gösterilmiştir. GA özellik seçim çalışmasında sıklıkla kullanılmaktadır. ‘Spambase’ veri seti ile ayrı olarak araştırma

yapıldığında fazla dokümana ulaşılamamıştır. Tez önerisinde kullanılmış olan veri setinin kullanımının sık olmadığı anlaşılmıştır. Eksik olan bu durumu ortadan kaldırabilmek adına tek bir veri seti üzerinde çeşitli özellik seçim çalışmaları yapılmıştır. GA çalışmasında mutasyon oranları değiştirilerek mutasyonun etkisi araştırılmak istenmiştir. Mutasyon oranı arttırılınca kullanılan algoritmalardaki başarım değerlerinin arttığı gözlemlenmiştir. Tez önerisi ile diğer çalışmaların karşılaştırması sonucunda yüksek başarım gösterenler koyu renk ile vurgulanmıştır. GA ile DNN hibrit çalışması ile karşılaşılmamıştır.

2.3.2. Karınca koloni optimizasyonu ile yapılan çalışmalar

Metasezgisel algoritmalardan olan karınca koloni optimizasyonu, sürü zekâsı temeline dayanmaktadır. Karıncaların besin ararken, en kısa yolu bulma üzerine uyarlanmış bir yaklaşımdır. Gerçek hayatta, optimizasyon problemlerin çözümlerinde sıklıkla kullanılmaktadır. Bu çalışma içerisinde, spam e-postaların optimum bir şekilde sınıflandırılabilmesi amacıyla özellik seçim yöntemi olarak kullanılmaktadır. ACO algoritmasının çalışmalardaki kullanım şekli, özellik seçimindeki başarımları ve özellik seçiminde kullanıldıktan sonra sınıflandırma algoritmalarının hangisinden faydalanıldığı araştırılmıştır.

Özellik seçiminde ACO kullanılarak boyut azaltma üzerine yapılan çalışmada, çizge yöntemiyle tüm özellikler bir grafik halinde temsil edilmektedir. ACO, özellik kümesinden en iyi özellik alt kümesine ulaşmaya çalışmaktadır. ACO kullanılarak yapılan çalışmada, sınıflandırma doğruluğunun oranını en yüksekte tutarak başarılı sonuçlar elde edilmiştir. ACO ile modifiye edilmiş graf karınca koloni optimizasyonu (modified graph clustering based ant colony optimization- MGCACO) algoritmasıyla karşılaştırılmalar yapılmıştır. Doğruluk metriğine dayalı olarak SVM ile %90,98, RF ile %94,17, KNN ile %89,6 ve DT ile %90,88 başarım oranları gösterilmiştir [69].

ACO, yinelemeli olarak çalışmaktadır ve optimum çözüme yakınsama eğilimindedir. Karıncalar yollarını tamamladıktan sonra feromon güncellemesi yapmaktadır. Feromon izinin belirli sınırlarda olup çabuk buharlaşmasını engellemek amacıyla maksimum-mininmum karınca sistemi (max-min ant system- MMAS) önerilmiştir. K-en yakın komşu algoritmasında, k değeri olarak 1 ve 5 değerleri seçilerek sınıflandırma yapılmış, k=5 ortalama ile 38,85 özellik seçilerek %89,3 başarım oranı edilmiştir [70].

ACO, özellik seçim çalışmalarının yanında farklı alanlarda da kullanılmıştır. SVM algoritmasının hızını arttırabilmek amacıyla, karınca koloni optimizasyonu örnek seçim algoritması kullanılmıştır. ACO algoritmasındaki kenar seçiminden esinlenilmiş, sınır tespitinde ACO, örnek seçiminde KNN kullanılmıştır. Tüm deneyler e-dolandırıcılık türü olan kredi kartı dolandırıcılığı, e-posta spam ve kimlik avı e-postası veri setlerine uygulanmıştır. Çalışmanın, SVM performansını olumlu etkilediği gözlemlenmiştir [71].

Literatür taraması yapılarak metasezgisel algoritmalarla ilgili olan çalışmaların, geniş bir araştırmasını yapan [11] makalenin sonucunda, incelenen algoritmalar, Spambase de dahil olan 6 veri seti üzerinde çalıştırılmış. Sürü zekâsı temelli özellik seçim çalışmalarından sonra SVM, NB ve adaboost kullanılarak sınıflandırmalar yapılmıştır. En yüksek başarım, %92,84 doğruluk oranına sahip olan PSO-SVM ile elde edilmiştir. ACO ile yapılan çalışmanın sonucunda, sırasıyla %91,87 %89,76 %90,66 doğruluk oranları elde edilmiştir.

Çizelge 2.5. Karınca koloni optimizasyonu ile özellik seçim çalışmaları

Çalışma	Özellik Seçimi	Sınıflandırma	Seçilen Özellik	Doğruluk	F1-skor
2018 [61]	ACO	NB	-	84	87
2018 [69]	MGCACO	SVM RF KNN DT	24	90,98 94,17 89,60 90,88	-
2018 [48]	GCACO	SVM KNN	-	91,60 87,20	-
2019 [70]	MMAS-ACO	KNN	38,85	89,3	
2021 [11]	ACO	SVM NB Adaboost	8,18(ort)	91,87 89,76 90,66	
2022 [72]	ACO	KNN SVM	54 34	99,8 99,9	99,6 99,9
2023[Tez önerisi]	ACO	LR RF NB KNN Adaboost LSVM DT DNN	38,6(ort)	91,31 95,11 90,66 89,47 94,14 89,03 92,18 94,46	89,36 94,12 88,32 87,25 93,02 86,93 90,7 93,45

ACO ile yapılmış olan çalışmaların özet tablosu Çizelge 2.5 ile verilmiştir. ACO algoritması, kullanılarak yapılan özellik seçim çalışmalarına nadir olarak rastlanmıştır. Tez çalışmasında kullanılan yöntemlerin, literatürde hepsinin birlikte kullanılmadığı gözlemlenmiştir. Yapılan çalışmalar arasında RF, NB, adaboost ve DT algoritmaları başarım oranlarının

karşılaştırmaları değerlendirildiğinde tez çalışması kapsamında, başarı oranının daha yüksek olduğu gözlemlenmiştir. ACO ile LR ve DNN hibrit çalışmaları bulunamamıştır. %90 üzerinde başarım oranı ile ACO algoritmasının başarılı çalıştığı gözlemlenmiştir. 2022 yılında sunulmuş olan tez çalışmasında [72] KNN ve LSVM algoritmalarının, yüksek başarımlı olduğu görüntülenmektedir.

Tez önerisinde kullanılmış olan algoritmalar dışında, özellik seçim yöntemlerini sunan tablo Çizelge 2.6 ile verilmiştir. Tez önerisinde, ortalama olarak en iyi sonuçları veren, %20 mutasyon oranı kullanılarak koşturulan GA sonuçları ile karşılaştırmalar yapılmıştır. 2020 yılında, özellik seçiminde yapay arı kolonisi ve sınıflandırma kısmında LR kullanılarak gerçekleştirilen çalışmanın sonucunda, başarılı sonuçlar elde edilmiş. Başarım durumları koyu renk ile belirtilmiştir. En başarılı sonuca RF algoritması ile ulaşılmıştır.

Çizelge 2.6. Farklı algoritmaların özellik seçim çalışmaları

Çalışma	Özellik Seçimi	Sınıflandırma	Seçilen Özellik	Doğruluk	F1-skor
2018 [73]	FA	NB	21	95,14	-
2019 [74]	GA	NB	38	94,5	-
2019 [39]	WCFS	SVM NB KNN	38 19 24	94,3 85 87	91,9 82 85
2020 [75]	CSA-LR ABC-CSA PSO DE	LR	-	94,20 94,49 92,46 93,84	92,66 92,94 91,19 92,42
2020 [76]	BFA	NB	21	95,14	-
2021 [77]	ABC-PSO	NB	33	90,81	-
2022 [78]	HHO	KNN	57	94,3	-
2023[Tez önerisi]	GA%20	LR RF NB KNN Adaboost LSVM DT DNN	-	91,53 96,09 91,10 90,12 94,57 89,47 94,57 95,01	89,71 95,29 88,95 88,32 93,57 87,86 93,47 94,06

3. METOT VE MATERYALLER

Bu bölümde tez çalışmasında kullanılan makine öğrenmesi algoritmalarından, özellik seçimi metotlardan ve veri seti gibi kullanılan materyallerden bahsedilmektedir. Algoritmaların temel olarak anlatımlarının yanında tez çalışmasında kullanılan özellik seçimi ve sınıflandırma uygulamalarından bahsedilmektedir. Veri seti ile ilgili gerekli açıklamalar bu alanda yer almaktadır.

3.1. Önerilen Metot

Bu bölümde araştırma metodolojisi ve gerçekleştirilen işlem adımları sunulmuştur. Bu çalışma ile e-postaların doğru ve hızlı bir şekilde sınıflandırılabilmesi amaçlanmaktadır. Sınıflandırmanın, doğru bir şekilde yapılabilmesi için içerikte bulunan özelliklerin, sınıflandırmaya etkisi tartışılmaktadır. Bunun için öncelikli olarak Spambase veri seti ile özellik seçimi yapılmadan gerçekleştirilen uygulamalar incelenmiştir. Çeşitli algoritmalar kullanılarak yapılan çalışmalarda farklı değerlendirme sonuçları sunulmuş [24],[44]. Çalışma kapsamında veri seti üzerinde özellik seçimi yapılarak, sınıflandırma performansındaki değişimler incelenmiştir. Bunun için genetik algoritma (GA) ve karınca koloni optimizasyonu (ACO) sarmal yöntemlerde özellik seçim algoritmaları olarak kullanılmıştır. Bu algoritmaların yanında, sınıflandırmaya etkisi az olan özellikler birliktelik kuralı madenciliği (ARM), karşılıklı bilgi özellik seçimi (MIFS), korelasyon özellik seçimi (CFS) ve varyans eşiği (VT) filtreleme yöntemleriyle ortadan kaldırılır. Bu özellik seçimlerinin sınıflandırmaya etkisini araştırabilmek amacıyla, birçok algoritmadan faydalanılmıştır. Lojistik regresyon (LR), rastgele orman (RF), naive bayes (NB), k-en yakın komşu (KNN), adaboost, doğrusal destek vektör makineleri (LSVM), karar ağacı (DT) ve derin sinir ağları (DNN) algoritmaları sınıflandırma çalışmasında uygulanmıştır.

GA ve ACO algoritmalarının performans değerini ölçebilmek amacıyla, uygunluk fonksiyonunu çalıştırma sayısı olarak N[5] değeri seçilmiştir. N kez çalıştırmanın sonucunda N adet doğruluk, kesinlik, duyarlılık ve f1-skor metrikleri elde edilmiştir. Bu değerlerden en iyi, en kötü, medyan, ortalama ve standart sapma değerleri hesaplanmıştır. Hesaplamalar sonucunda, en iyi değeri veren sonuçlar kayıt altına alınmıştır. GA ve ACO algoritmalarının

[5] N=10

en iyi değerleriyle, diğer özelik seçim yöntemleri ve özellik seçimi yapılmadan elde edilen doğruluk, kesinlik, duyarlılık ve f1-skor değerleri karşılaştırılmıştır.



Şekil 3.1. Çalışma metodu

Bahsedilen çalışma metodu, Şekil 3.1 ile gösterilmektedir. Öncelikli olarak özellik seçimi işlemi gerçekleştirilmiştir. Seçilen özeliklerle düzenlenen veri seti sınıflandırma işlemine tabi tutulur. Eğitim ve test verisi olarak ayrılan veri seti üzerinde, sınıflandırma çalışması yapılır. Elde edilen tahmini sınıflandırma sonucu ile test verisi üzerinde bulunan sınıflandırma sonuçları karşılaştırılarak başarım metrikleri hesaplanır. GA ve ACO algoritmalarının çalışmaları sonucunda, N adet başarım metriği elde edilir. Başarım metrikleri performans değerlendirme fonksiyonuna gönderilir, en iyi sonucu veren sınıflandırma algoritmaları kayıt altına alınır. En iyi sonucu veren değerler alındıktan sonra, bir kez çalıştırılmış olan filtreleme yöntemleri ve özellik seçimi yapılmadan geçekleştirilen sınıflandırma çalışması sonuçları karşılaştırılmıştır.

3.2. Tez Çalışmasında Kullanılan Yaklaşımlar

Tez çalışmasında kullanılmış olan algoritmaların temel sınıflandırılması bu başlık altında anlatılmıştır.

Makine öğrenmesi

Makine öğrenmesi insan gücüne ihtiyaç duyulmadan bilgisayar ortamında verinin anlamlı bilgiye dönüştürülmesini sağlayan yapay zekâ tekniğidir. Makine öğrenimi günümüzde farklı sektör ve uygulamalarda kullanılmaktadır ve alanı da gittikçe genişlemektedir [79-80]. Makine öğrenimi ile verilerden anlamlı sonuçlar çıkarılabilir, otomatik bilgi üretebilir, algoritmaları eğitebilir, ilişkileri tanımlayabilir, bilinmeyen kalıpları tanıyabilir ve öğrenim süreçlerini optimize edilebilir. Örneğin e-alışverişte kullanıcılara satın aldıkları ürünler ile ilgili reklamları önermek amacıyla çok sayıda çalışma yapılmaktadır. Çevrimiçi reklam dağıtımını gerçek zamanlı olarak özelleştirebilmek için makine öğrenimi algoritmalarının kullanılmasıdır. Özelleştirilmiş pazarlamanın yanı sıra, bazı makine öğrenimi algoritmaları dolandırıcılığı tespit etmek, spam filtrelemek, tehditleri tespit etmek, kestirimci bakım ve haber beslemeleri geliştirmek amaçlarıyla da kullanılmaktadır [81-82].

Derin öğrenme

Derin öğrenme, yapay sinir ağları ve simülasyon öğrenimi kullanarak gerçek dünyadaki karmaşık sorunları çözmede insan beyninin düzeyine uyan bir model geliştirmeyi amaçlayan bir makine öğrenimi dalıdır [83-84]. Bir girdi katmanı, bir çıktı katmanı ve her biri birbirine bağlı ağırlıklı düğümler veya nöronlar koleksiyonunu içeren bir veya daha fazla ekstra gizli katmandan oluşmaktadır. Derin öğrenme, insanların başa çıkamayacağı büyük miktardaki verilerden öğrenme yapabildiği için öğrenme hızı ile karakterize edilir. Bu nedenle, derin öğrenme, insanların ulaşamayacağı fikir ve sonuçlara varabilir [85]. Derin öğrenme, görüntü tanıma, hava tahmini veya tarım alanlarında faydalanılabilir. Derin öğrenme süreci hangi özelliklerin hangi seviyeye en uygun şekilde yerleştirileceğini kendi başına öğrenebilmesidir. Bu özellik el yordamıyla ayarlama ihtiyacını ortadan kaldırmaktadır. Örneğin, değişen katman sayıları ve katman boyutları farklı soyutlama dereceleri sağlayabilmektedir.

Metasezgisel algoritmalar

Metasezgisel terimi, çok çeşitli optimizasyon problemlerinin çözümü için önerilen daha yüksek seviyeli buluşsal yöntemleri tanımlamaktadır. Pek çok metasezgisel algoritma zorlu problemlerin çözümü için başarıyla uygulanmaktadır. Karmaşık problemleri çözmek için bu algoritmalarının tercih edilme sebebi, en iyi sonucu vermeleridir. Son yıllarda, metasezgisel algoritmaları, mühendislik tasarımı, dijital görüntü işleme ve bilgisayarla görme, ağlar ve iletişim, güç ve enerji yönetimi dahil olmak üzere birçok uygulama alanlarında çeşitli optimizasyon problemlerini çözme araçları olarak popülerlik kazanmıştır [86].

Metasezgisel yöntemler, bir arama problemi için belirli kurallar dizisini uygulayan bir arama motoru veya bireylerin, uygun bir çözüm alanında farklı aday çözümleri keşfetmek için kullandığı popülasyona dayalı arama şemalarını modelleyen optimizasyon çerçeveleridir. Bu yöntemler, bireyler arasındaki etkileşimi ve popülasyonun çeşitliliğini artırarak avantajlar sunar [87].

Evrimsel algoritmalar

Evrimsel algoritmalar, polinom zamanda çözülemeyen problemleri çözen metasezgisel yaklaşımdır. Biyolojik evrimden ilham alan başlatma, seçme, genetik operatörler ve sonlandırma olmak üzere dört ana adımı içeren doğal seçilim sürecine dayanmaktadır. En sık kullanılan evrimsel algoritmalar genetik algoritma, evrim stratejisi, diferansiyel evrim ve genetik programlamadır şeklinde sıralanabilir.

Genetik algoritma, belirli bir problem için çözümler üreten buluşsal tabanlı bir optimizasyon tekniğidir ve Darwin'in evrim teorisi ilkesine dayanır. John ve diğerleri (2006), GA' nın beş ana işlevselliği olan kromozom, uygunluk, doğrulama, seçim ve rekombinasyon aşamalarından oluştuğunu açıklamıştır. İlk olarak, rastgele oluşturulmuş bir kromozom popülasyonuyla başlar ve uygunluk fonksiyonuna göre doğrulama yapılır. Daha güçlü ebeveynlerin kromozomları birleştirilir ve yeni çocuk kromozomu üretilir. Ardından, bir sonraki popülasyonla devam edilir. Yeni neslin evrim süreci, belirlenen durma kriterine kadar aktif bir şekilde devam eder [88].

Sürü tabanlı algoritmalar

Sürü tabanlı algoritmalar, doğal ve yapay davranışlara sahip birçok bireyin iş birliği yaparak kendi kendini yöneten bir organizasyon içinde çalışan bir hesaplama metodolojisidir. Bu bireyler, kuş sürüleri, hayvan grupları, karınca kolonileri ve balık sürüleri gibi örneklerde olduğu gibi, merkezi bir birey tarafından kontrol edilmez. En sık uygulanan sürü tabanlı algoritmalar yapay arı kolonisi, parçacık sürü optimizasyonu, karınca kolonisi optimizasyonu ve balık sürü algoritması şeklinde sayılabilir.

ACO, Dorigo ve diğerleri tarafından tanıtılan ve geniş bir uygulama yelpazesine sahip olan metasezgisel bir algoritmadır [89]. ACO' nun üç temel adımı vardır; ilk çözümü oluşturmak, yerel aramayı uygulamak ve son olarak feromon değerini güncellemektir. Sıfır küme çözümü ile başlar ve tüm çözümden iyi olarak seçilenleri içermektedir. Ardından ACO, yerel arama süreci boyunca belirlenen çözümleri geliştirmektedir ve son olarak feromon değerini artırarak veya azaltarak güncelleme yapmaktadır. İyi çözüme ulaşanlarla ilişkili olanların feromon değerini arttırıp ve geriye kalanlarının feromon değeri düşürülmektedir.

3.3. Özellik Seçimi

Özellik seçimi, problemi çözmeye yönelik ilgili özelliklerin saptanması, çalışma performansı üzerinde minimum bozulmaya sebep olan özellik alt kümelerin bulunması amacıyla ilgisiz veya gereksiz özelliklerin listeden çıkarılması işlemidir. Özellik seçiminin çalışmaya kattığı çeşitli avantajları bulunmaktadır. Bunlar;

- Makine öğrenimi algoritmalarının performansını iyileştirme
- Veriyi anlama, süreç hakkında bilgi edinme ve gerekli olduğu yerde görselleştirme imkânı sağlama
- Genel veri azaltma, depolama gereksinimlerini sınırlama ve maliyetleri düşürmeye yardımcı olma
- Özellik seti azaltma, bir sonraki veri toplama turunda veya kullanım sırasında kaynakları koruma
- Sadelik, daha basit modeller kullanma ve hız kazanma imkânı sağlama

Özellik seçme yöntemleri, bireysel değerlendirme ve altküme değerlendirmesi olmak üzere iki farklı yaklaşıma ayrılabilir. Bireysel değerlendirme, aynı zamanda özellik sıralaması

olarak da bilinir ve her bir özelliği alaka derecelerine göre ağırlıklandırarak değerlendirir. Alt küme değerlendirmesi ise belirli bir arama stratejisine dayanarak aday özellik alt kümeleri üretir. Her bir aday özellik altkümesi, belirli bir değerlendirme ölçütü kullanılır ve bu ölçüye göre önceki en iyi adayla karşılaştırılır. Fazlalık özelliklerin benzer sıralamalara sahip olması muhtemel olduğundan, bireysel değerlendirme yaklaşımı gereksiz özellikleri kaldıramazken, altküme değerlendirme yaklaşımı özellik alaka düzeyine göre fazlalık özellikleri ortadan kaldırabilir. Özellik indirgeme, benzerlik, tutarlılık, uzaklık ve bazı istatistiksel ölçümlere dayalı olarak yapılabilmektedir [90].

Özellik seçimi filtreleme (filter) ve sarmal (wrapper) yöntemlerle geçekleştirilebilir. Filtreleme, korelasyon özellik seçimi [91], karşılıklı Bilgi [92] gibi özellik alt kümelerinin uygunluğunu, tahmin etmek için kullanılır. Sarmal yöntemler, özellik alt kümelerini tümevarım algoritması performansına dayanarak değerlendirir ve genellikle daha iyi bir performans elde etme eğilimindedir [93]. Bommert ve diğerleri 22 farklı filtreleme yöntemini gözden geçirerek [94], herhangi bir filtreleme yönteminin diğer tüm yöntemlerden tutarlı bir şekilde daha iyi performans göstermediği sonucuna varmışlardır.

Filtreleme yöntemleri

Filtreleme, özellik seçimi sürecinin bir ön işleme adımı olarak gerçekleştirilen ve eğitim verilerinin genel özelliklerine dayanan bir yöntemdir. Bu yöntem, düşük hesaplama maliyeti ve iyi genelleme yeteneği gibi avantajlara sahiptir. Filtreleme yöntemi, azaltılmış sayıda özelliğe sahip veriler üzerinde tahmin edicileri seçerken doğrudan geri bildirim almadan, verilerden hesaplanan performans değerlendirme metriklerine dayanır.

Temelde yapılan özellik seçim yöntemleri filtreleme yöntemlerine girmektedir [50]. Tek değişkenli ve çok değişkenli yöntemler şeklinde sınıflandırılabilir. Tek değişkenli yöntemlerde özellik seçim sürecinde özellikler arasındaki olası bağımlılık göz ardı edilirken, çok değişkenli yöntemlerde özelliklerin uygunluğunun değerlendirilmesinde özellikler arasındaki olası bağımlılıklar dikkate alınır. Bu nedenle, çok değişkenli yöntemler hesaplama açısından tek değişkenli yöntemlerden daha pahalı olabilirken performansları tek değişkenli yöntemlerden daha iyi olmaktadır [95]. Çalışma içerisinde kullanılan filtreleme yöntemleri listelenmiştir. Çizelge 3.1’de filtreleme yöntemlerinin değişken türleri gösterilmektedir.

Çizelge 3.1. Filtreleme yöntemleri

Filtreleme Yöntemleri	Tek/Çok Değişken
Karşılıklı Bilgi Özellik Seçimi (MIFS)	Tek değişkenli
Varyans Eşiği (VT)	Tek değişkenli
Birliktelik Kuralı Madenciliği (ARM)	Çok değişkenli
Korelasyon Özellik Seçimi (CFS)	Çok değişkenli

MIFS ve VT yöntemlerinde tek değişken ele alınarak kullanılan özellikler, birbirinden bağımsız olarak işleme dahil edilmektedir. Tanımlanan değişkene göre özelliklerin seçilmesi yapılmaktadır. Belirli bir eşiğin altında bulunan özellikler seçilmektedir. Seçim yapılırken de özellikler birbirini etkilememektedir.

ARM ve CFS yöntemleri ile özellik alt kümeleri korelasyona dayalı buluşsal değerlendirme işlevine göre sıralayan çok değişkenli filtreleme algoritmalarıdır. Özellikler belirli bir korelasyon ile ilişkilendirilerek ilişiği iyi olan özellikler seçilmektedir.

Sarmal yöntemler

Sarmal yaklaşım, seçilen özelliklerin belirli bir alt kümesinin değerlendirilmesi için öğrenme algoritması kullanmaktadır [59]. Sarmal yaklaşımlar, arama stratejisine dayalı olarak genel olarak iki kategoriye ayrılır: sıralı arama ve rastgele arama [96]. Sıralı arama yöntemleri ardışık olarak özellik ekler veya kaldırır, ancak yerel bir optimumda sıkışıp kalma eğilimindedir. Örneğin, sıralı geriye doğru seçim, sıralı ileriye doğru seçim ve kayan arama yöntemidir [97]. Öte yandan, rastgele arama yöntemleri, yerel optimum çözümlerden kaçmak için rastgeleliği, arama prosedürlerine dahil etmektedir. Bu yöntemlerin örnekleri arasında rastgele mutasyon tepe tırmanma, benzetilmiş tavlama, genetik algoritma [74] ve karınca kolonisi optimizasyonu [98] sayılabilir. Sarmal yaklaşım genellikle belirli bir sınıflandırıcı için daha yüksek doğruluk sağladığı halde, seçilen özelliklerin diğer sınıflandırıcılar üzerinde daha az genelleştirilebildiği bir yaklaşımdır.

Bu teknik, olası özellik alt kümeleri uzayında araştırma yapmak için bir arama stratejisi kullanılmaktadır. Arama stratejisi ile bir makine öğrenimi algoritmasının kombinasyonu şeklinde çalışmaktadır. Kombinasyon sonucunda özellik alt kümeleri elde edilmektedir. Her bir alt küme, makine öğrenimi algoritmasının başarısına bağlı olarak değerlendirilir. Tez

çalışmasında verilen e-posta veri kümesi için uygun özellikleri tespit eder ve bunları kullanarak bir model eğitir. Özellikler arasındaki ilişkilere bağlı olarak, özellikler alt kümeye eklenip çıkarılır.

Genetik algoritma ile yapılan özellik seçiminde uygunluk fonksiyonu olarak rastgele orman regresyon (random forest regression- RFR) kullanılarak özellik alt kümesi seçilmektedir. Karınca koloni optimizasyonu algoritmasında uygunluk fonksiyonu olarak rastgele orman sınıflandırma (random forest classifier- RFC) algoritması kullanılmaktadır. Her aday özellik alt kümesi, değerlendirme işlevi için öğrenme algoritması tarafından başlatılan modelin kalitesine ilişkin bir tahmin döndürmektedir. Her bir aday özellik alt kümesinin arama sürecinde değerlendirilmesi ve hedef öğrenme algoritması için en iyi sonucun elde edilmesini sağlamak amacıyla N kez uygulanmıştır. Tekrarlı çalışmanın zaman açısından oldukça maliyetli bir işlem olduğu gözlemlenmiştir. Buna rağmen, sarmal yöntemler, filtreleme yöntemlerinden daha iyi performans gösterdiği sonuçlarda izlenmiştir. Sarmal yöntemlerin filtreleme yöntemlerinden daha fazla hesaplama kaynağı gerektirdiği için küçük eğitim veri kümelerinde kullanılması daha uygun olmaktadır [99]. Sarmal yaklaşımlar hesaplama açısından daha verimli bir şekilde kullanılmasına izin veren [100] örnek çalışmada, GA kullanılarak özellikle büyük veri kümelerinde elde edilebilecek özellik seçim çözümlerinin kalitesini önemli ölçüde artırdığı sonucuna varmışlardır.

Özellik seçim metotlarının avantaj ve dezavantajları Çizelge 3.2'de sunulmaktadır. Filtreleme yöntemlerinin avantajlarından bir tanesi sınıflandırma algoritmalarına bağımlı olmamasıdır. Hızlı bir şekilde sonuca ulaşması maliyet açısından diğer yöntemlere göre daha iyi olduğunu göstermektedir. Çalışma içerisinde kullanılmış olan yöntemler tek değişkenli ve çok değişkenli olarak gruplandırılmıştır. Sarmal yöntemlerin sınıflandırıcıyla etkileşimi söz konusudur.

Çizelge 3.2. Özellik seçim yöntemleri

Metot	Değişken Sayısı	Avantaj	Dezavantaj	Algoritma
Filtreleme	Tek değişkenli	*Sınıflandırıcılardan bağımsız *Paketleyicilere göre daha düşük hesaplama maliyeti *Hızlı *İyi genelleme yeteneği	*Sınıflandırıcı ile etkileşim yok *Özelliklere bağımlılık yok	MIFS VT
	Çok değişkenli	*Özelliklerin bağımlılığı *Sınıflandırıcıların bağımsızlığı *Sarmal yaklaşımından daha iyi zaman karmaşıklığı	*Tek değişkenli yöntemlerden daha yavaş *Tek değişkenli yöntemlerden daha az derecelendirilebilir *Sınıflandırıcı etkileşim yok	ARM CFS
Sarmal	Çok değişkenli	*Sınıflandırıcı ile etkileşim *Özellik bağımlılıklarını yakalar	*Sınıflandırıcıya bağlı seçim *Deterministik yönteme göre daha fazla uyum riski	GA-RFR ACO-RFC

Çok sayıda özellik seçim yöntemlerinin olduğu araştırmalarla gözlemlenmiştir. Çalışma performansları, algoritmalara ve probleme göre değişmektedir. Bundan dolayı ‘En iyi yöntem budur’ şeklinde ifade kullanılamamaktadır. Problemin %100‘e yakın sonucunun alınmasının yanında kullandığı bellek ve eğitim süresi de önemli kriter olmaktadır. En iyi sonucu almaya çalışırken bu kriterlerinden minimum maliyete indirilmesi önemli hale gelmektedir. Bu yöntemlerin kendi içerisinde tercih sebepleri bulunmaktadır. Probleme, kullanılan belleğe ve süreye bağlı olarak değişmektedir. Bu sebepten dolayı, temel filtreleme yöntemlerinin yanında karmaşık algoritmalarla kullanılan veri seti eğitilerek özellik seçim işlemleri gerçekleştirilir. Filtreleme yöntemleri algoritmalardan bağımsız olarak gerçekleştirilir. Bu özellik, filtreleri hesaplama açısından basit ve hızlı hale getirir ve son derece büyük ölçekli veri kümelerini işleyebilir. Bununla birlikte, çoğu filtreleme tek değişkenlidir, yani her bir özellik diğer özelliklerden bağımsız olarak ele alınır; bu, genellikle daha fazla hesaplama kaynağı gerektiren bir dezavantajdır. Kullanılan diğer özellik seçim yönteminde kullanılan algoritmalarda özellikler birbiriyle ilişkili olarak seçim gerçekleştirilir.

3.3.1. Birliktelik kuralı madenciliği

Birliktelik kuralı madenciliği, veri madenciliğinin en önemli ve iyi araştırılmış tekniklerinden biridir. Başlangıçta, işlem veri tabanları veya diğer veri havuzlarında korelasyonlar, sık örüntüler, ilişkilendirmeler veya geçici yapılar bulmak amacıyla kullanılmıştır. Çalışma içerisinde özellik seçim yöntemi olarak kullanılarak çok değişkenli filtreleme yöntemi gerçekleştirilmiştir. Birliktelik kuralı madenciliğinde kullanılan temel kavramlar [101];

- Kural (rule): Öncül ile sonuç arasındaki ilişkiyi gösteren koşullu yapıdır.

$X \rightarrow Y$ X öncül Y sonuçtur. X' in Y' yi temsil ettiği anlamına gelmektedir.

- Destek (support-s): Bir veri tabanındaki bir öğe kümesinin sıklığıdır.

- Güven (confidence-c): Bir kuralın doğru olduğu durumların, veri tabanında uygulanabilir olduğu sürelere göre yüzdesidir. Eş. 3.1 ile gösterilmektedir.

$$C = S(X \cup Y)/S(X) \tag{3.1}$$

- Uygunluk fonksiyonu: Oluşturulan kuralların kalitesini değerlendirmek için kullanılır. Her birliktelik kuralı madenciliğinin farklı uygunluk işlevi olabilir.

Birliktelik kuralı madenciliği için iki temel ölçü vardır; destek ve güven. Büyük veri setlerinde kullanıcılar sık kullanılan öğelerle ilgilendikleri için yararlı olmayan özellikleri ortadan kaldırabilmek amacıyla destek ve güven eşik değerlerini önceden belirler. Bu çalışma içerisinde sık kullanılan özellikler şeklinde tanımlama yapılmıştır.

Bir birliktelik kuralının desteği, $X \cup Y$ içeren kayıtların veri tabanındaki toplam kayıt sayısına oranı olarak tanımlanır. Örneğin bir işlem bir özellik içeriyorsa, bu özelliğin destek sayısı bir artar. Destek Eş. 3.2 ile hesaplanır:

$$XY\ destek = \frac{XY\ destek\ sayısı}{veri\ tabanındaki\ toplam\ işlem\ sayısı} \tag{3.2}$$

Bir öğenin desteği, birliktelik kuralının istatistiksel önemidir. Bir özelliğin desteğinin %0,1

olduğu varsayılırsa, bu işlemin yalnızca yüzde *0,1*'inin e-posta içerisinde kullanıldığı anlamına gelmektedir. Birliktelik kuralında daha yüksek bir destek gerektiği için, bu kadar düşük bir orana sahip özellikler dikkate alınmayacaktır. Madencilik sürecinden önce, kullanıcılar minimum destek eşiği olarak belirleyebilecekleri bir değer kullanır. Bu, eşik değerini aşan öğe kümelerinden türetilen belirli ilişkilendirme kurallarıyla ilgilendikleri anlamına gelir. Ancak, zaman zaman öğe kümeleri eşik değeri tarafından belirlendiği kadar sık olmasa da bu kümelerden türetilen birliktelik kuralları hala önemli olabilir. Tek başına e-posta içerisinde kullanılma sıklığı önemli olmazken birliktelik kuralıyla farklı özelliklerle kullanılması önemli hale gelebilmektedir [102].

Bir birliktelik kuralının güvenirliği, $X \cup Y$ içeren işlem sayısının X içeren toplam kayıt sayısına yüzdesi olarak tanımlanır; burada yüzde güven eşiğini aşarsa, ilişkilendirme kuralı $X \rightarrow Y$ olabilir. Güven hesaplaması Eş. 3.3 ile gösterilmektedir.

$$Güven\ (X|Y) = \frac{XY\ destek\ sayısı}{X\ in\ destek\ sayısı} \tag{3.3}$$

Güven, birliktelik kurallarının gücünün bir ölçüsüdür, $X \rightarrow Y$ birliktelik kuralının güveninin %70 olduğu varsayılırsa, bu *X*' i içeren işlemlerin *%70*'inin *Y*' yi içerdiği anlamına gelir, benzer şekilde belirtilen kuralların farklılığını sağlamak için minimum güven kullanıcılar tarafından önceden tanımlanır.

Çalışma içerisinde öncül $X \rightarrow sonuç\ (Y)$ $[destek = \%30,\ güven = \%70]$ oranları ile kullanılmıştır. Toplam işlemde öncül ve sonuç birlikte sık olduğu *%30*'luk bir oran anlamına gelmektedir. Öncül ve sonuç birlikte oluştuğuna dair *%70* güven düzeyi olduğunu göstermektedir. Verilen güven ve destek değerlerine göre hesaplanan apriori algoritmasına göre özellik alt kümeleri belirlenir. Belirlenen özellikler sınıflandırma algoritmalarına gönderilerek performans değerlendirmesi yapılır.

Apriori algoritması

Apriori algoritması, sık öğe kümesi madenciliği için önerilen ilk algoritmadır. Agarwal ve Srikant tarafından geliştirilmiş ve sonrasında Apriori algoritması olarak kullanılmaya başlanmıştır [103]. $I = i_1, i_2, \ldots, i_n$ özellikler olarak adlandırılan hazır değerler kümesidir.

D işlemler kümesi olsun, burada her T işlemi $T \subseteq I$ öğeler kümesidir. $X \subseteq T$ ise bir T işleminin I dizinindeki bazı öğelerinin bir kümesi olarak X'i içerdiğini göstermektedir. Birliktelik kuralı $X \rightarrow Y$ formunun bir uygulamasıdır. Burada $X \subset I, Y \subset I$ ve $X \cap Y = 0; X \rightarrow Y$ kuralı, D'deki X' i içeren işlemlerin $\%c$ değerinin aynı zaman da Y' yi de içeriyorsa, c güven değeriyle D işlem kümesinde geçerlidir. D'deki işlemlerin $\%s$ değeri $X \cup Y$ içeriyorsa $X \rightarrow Y$ kuralı D işlem kümesinde s desteğine sahiptir.

Apriori, *budama (pruning) ve birleştirme (combining)* adı verilen bir teknik kullanır. Aday sık öğe kümelerinin bir listesini üretir. Ardından, gereken minimum destekten daha düşük desteğe sahip olan öğe kümelerini kaldırmak için *minimum destek* değerini kullanır.

Apriori algoritmasının iki aşaması vardır: Birincisi, sık kullanılan öğe kümelerini, yani işlemler kümesinde sıklıkla meydana gelen öğe kümelerini bulurlar ve sonra bunlardan kurallar üretirler. Daha sonra, 2-uzunluklu öğeler oluşturmak için aday listesindeki her öğe kümesini diğer öğe kümeleriyle birleştirir. Bu adımda Apriori, minimum desteğe kıyasla daha düşük desteğe sahip öğeleri kaldırır. En sık kullanılan öğe kümelerini keşfetmek için tekrarlamalı bir yaklaşım sunar. Bu noktada, algoritma 3-uzunluk, 4-uzunluk, ... ürün kümelerini bulmak için aynı işlemi yinelemeli olarak yürütür. Sonuç olarak, bu algoritma, sık kullanılan öğe kümelerini temin edebilmek için kullanıcı tanımlı bir parametreye, minimum desteğe ihtiyaç duyar. Daha sonra algoritma, aday listeleri üreten her düzey için *budama* adlı bir işleme dayalı olarak her öğe kümesini kontrol eder. Bu sürece göre, aday listesindeki tüm öğelerin boş olmayan her bir alt kümesi önceki düzeyin aday listesine ait olmalıdır. Bu koşulu sağlamayan öğe kümesi varsa, algoritma onu aday listesinden kaldırır. Son aday listesi sık kullanılanlar listesidir.

Apriori daha sonra her öğenin boş olmayan tüm alt kümelerini çıkarır ve kurallar oluşturur. Bu adımda algoritma, zayıf kuralları kaldırmak için *minimum güven* adı verilen başka bir kullanıcı tanımlı parametre uygular [104].

3.3.2. Karşılıklı bilgi özellik seçimi

Karşılıklı bilgi özellik seçimi, klasik bilgi teorisinde ilk olarak 1948' de Shannon tarafından tanıtılmıştır [105]. İki değişken arasındaki hem doğrusal hem de doğrusal olmayan karşılıklı bağımlılığı ölçen ve bu bağımlılığı doğrusal olarak ifade eden seçim yöntemidir. Bu nedenle,

bir değişken seçildiğinde diğerinde ortadan kalkan belirsizlik miktarına bakmaktadır. $I(Y:X)$ olarak temsil edilen karşılıklı bilgi, gerçekte Y'nin bilinmesinden kaynaklanan X' teki belirsizlikteki azalmayı ölçer, yani Y'nin X hakkında ne kadar bilgi aktardığını gösterir. Karşılıklı bilgi özellik seçimi aşağıdaki özelliklere sahiptir;

- Özellikler simetriktir. Yer değiştirildiklerinde birbirleri hakkında bilgi verir. Eş. 3.4' te gösterilmektedir.

$$I(X:Y) = I(Y:X) \tag{3.4}$$

- X ve Y arasındaki bağlam negatif değildir; X' in belirsizliği, Y'nin öğrenilmesiyle artırılamaz. Eş. 3.5 ile gösterilmektedir.

$$I(X:Y) \geq 0 \tag{3.5}$$

X' in kendisi hakkında içerdiği bilgi, X' in entropisidir. Eş. 3.6 ile gösterilmektedir.

$$I(X:X) = H(X) \tag{3.6}$$

Değişkenlerin birbirleri hakkında içerdikleri bilgiler asla değişkenlerin kendilerindeki bilgilerden daha büyük olamaz. Eş. 3.7'de X' teki bilgi hiçbir şekilde Y ile ilgili değildir, yani Y verildiğinde X hakkında hiçbir bilgi edinilmez ve bunun tersi de geçerlidir. Bu nedenle, X ve Y kesinlikle bağımsızdır. Karşılıklı bilgi özellik seçimi, en iyi yol olarak kabul edilen entropi kullanılarak, olasılık yoğunluğu kullanılarak hesaplanabilir [92].

$$\begin{aligned} I(X:Y) &\leq H(X) \\ I(X:Y) &\leq H(Y) \end{aligned} \tag{3.7}$$

Özellik seçim yöntemi ve optimizasyon kriterinin tasarımı için etkili bir bilgi ölçüm tekniğidir. Anahtar kavramlar arasında bilgi entropisi, ortak entropi, koşullu entropi vb. yer almaktadır. Bu klasik ölçütlerin yardımıyla, temel özellikleri seçmek ve ilgisiz özellikleri ortadan kaldırmak amacıyla, önceki belirsizlik ve beklenen sonraki belirsizlik ile istatistiksel korelasyonun gücü arasındaki farkı almaya çalışan karşılıklı bilgi, bilgi kazancı ve geliştirilmiş bilgi ölçüm stratejisi türetilebilir. Aday özellikler ile seçilen özellikler arasında gereksiz bilgilerin varlığını dikkate almaktadır [106].

3.3.3. Varyans eşiği

Varyans eşiği yöntemi, özellik seçimine yönelik temel bir yaklaşımdır. Varyansı bir eşik değeri karşılayamayan tüm özellikleri kaldırır. Varsayılan olarak, tüm sıfır varyans özelliklerini, yani tüm örneklerde aynı değere sahip olan özellikleri kaldırır. Bu özellikler kaldırıldıktan sonra yeni veri setinin doğruluğu hesaplanır ve orijinal veri seti ile karşılaştırılır [107]. Örneğin, mantıksal özelliklere sahip bir veri kümesinde örneklerin %80'inden fazlasında bir veya sıfır olan özelliklerin kaldırılması istendiğinde varyans hesaplaması yapılır. Mantıksal özellikler, bernoulli rastgele değişkenleridir ve bu tür değişkenlerin varyansı Eş. 3.8 ile gösterildiği şekilde hesaplanır:

$$T_h = \alpha(1-\alpha) \tag{3.8}$$

İstatistikte varyans, bir özelliğin varyans değerini elde etmek için bir veri kümesinin yayılmasının ne kadar uzağa uygulandığını ölçmek için kullanılır. Eş. 3.9 da verilmiştir.

$$\sigma^2 = \frac{\sum(x-\mu)^2}{N} \tag{3.9}$$

Burada μ ortalamadır ve N veri satırlarının sayısıdır. Varyans değeri, varyans eşiğini karşılamıyorsa özellik düşürülür. Eşiğin kendisi serbestçe tanımlanabilir. Ancak referans olarak Eş. 3.8’ deki denklem hesaplamak için uygulanır.

Burada α sözde veri yayılmasının yüzdesidir. Sıfır sapma eşiği değeri, tüm örneklerde yalnızca bir değer olduğu anlamına gelir. Eğitim veri setinde varyansı bu eşikten daha düşük olan özellikler kaldırılacaktır. Varsayılan, tüm özellikleri sıfır olmayan varyansta tutmaktır, yani tüm örneklerde aynı değere sahip olan özellikleri kaldırmaktır [108].

3.3.4. Korelasyon özellik seçimi

Korelasyon özellik seçimi, özellik alt kümelerini korelasyona dayalı buluşsal değerlendirme işlevine göre sıralayan basit ve çok değişkenli filtreleme algoritmasıdır. Sınıfla yüksek derecede ilişkili olan ve birbiriyle ilişkisi olmayan özellikleri içeren alt kümelerle ilgili olarak, değerlendirme işlevinin yanlılığı mevcuttur. Alakasız özellikler, sınıfla düşük

korelasyona sahip olacağından göz ardı edilir. Fazlalık özellikler, kalan özelliklerden bir veya daha fazlasıyla yüksek oranda ilişkili olacağından elenir. Bir özelliğin kabulü, örnek uzayının diğer özellikler tarafından önceden tahmin edilmemiş alanlarındaki sınıfları ne ölçüde tahmin ettiğine bağlı olmaktadır [91]. Korelasyon özellik seçiminin özellik altkümesi değerlendirme fonksiyonu Eş. 3.10;

$$F_s = \frac{k\overline{c_{mn}}}{\sqrt{k+k(k-1)\overline{c_{nn}}}} \tag{3.10}$$

Burada F_s, bireysel özellikler ile çıktı sınıfı arasındaki korelasyondur. k, özelliklerin sayısıdır. $\overline{c_{mn}}$ bireysel özellikler ile çıktı değişkeni arasındaki korelasyonun ölçüsüdür. $\overline{c_{nn}}$ özellikler arasındaki ortalama karşılıklı korelasyondur. Bu nedenle, kullanılan ölçü, çıktı ile yüksek düzeyde ilişkili olan alt kümelere yüksek değerler verirken, birbiriyle zayıf bir şekilde ilişkilidir.

Eş. 3.10'daki ifade korelasyon özellik seçiminin çekirdeğini oluşturur ve tüm olası özellik alt kümelerinin arama uzayındaki özellik alt kümelerine bir sıralama uygular. Tüm olası özellik alt kümelerinin kapsamlı bir şekilde sıralanması çoğu durumda engelleyici olduğundan, aramanın organizasyonu, başlangıç noktası ve durdurma kriteri ile ilgili ele alınır.

3.3.5. Genetik algoritma

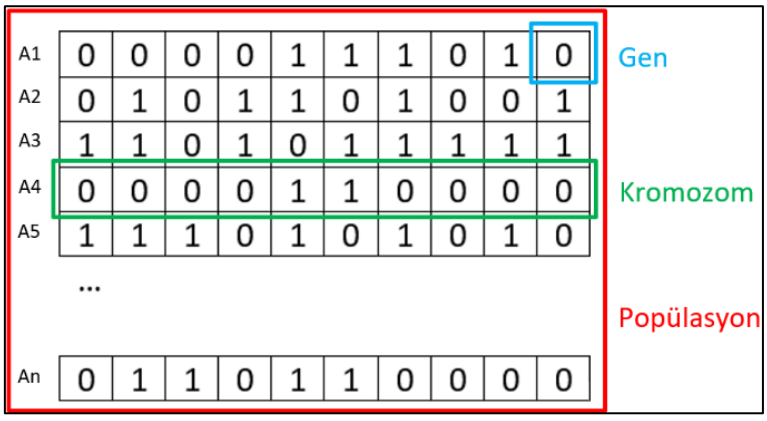
Genetik algoritma, en uygun olanın hayatta kalması ve doğal seçilim teorisinden ilham alan bir yaklaşımdır. Bu evrimsel yaklaşım sistematik olarak ilk kez John H. Holland tarafından önerilmiştir [109]. Bu yaklaşım en temel metasezgisel optimizasyon algoritmasıdır ve birçok uygulamada kullanılmaktadır. Sayısal optimizasyon, sinir ağlarının optimizasyonu, sınıflandırma, özellik seçimi gibi farklı görevlere uygulanmaktadır. Probleme özgü uygunluk fonksiyonuna bağlı olarak kodlanmış çözüm popülasyonunu geliştiren evrimsel bir algoritmadır [110].

Bir problem için olası her çözümün bir kromozom olarak kodlandığı popülasyon tabanlı bir yöntemdir. Popülasyon bireylerin oluşturduğu kümeye verilen addır. İlk popülasyonu, rastgele seçim olmak üzere çeşitli şekillerde seçilebilir. Bir başlangıç popülasyonu

oluşturulduktan sonra, bu popülasyonun bireyleri, optimizasyon görevinin çözümünde bir kromozomun iyiliğini bildiren bir uygunluk fonksiyonu aracılığıyla değerlendirilir. Ek olarak uygunluk, genetik algoritmada optimize edilecek (minimizasyon veya maksimizasyon) işlevidir. Bu uygunluk fonksiyonuna bağlı olarak, kromozomlar seçilir ve seçilen kromozomlara bazı genetik operatörler (mutasyon ve çaprazlama) uygulanarak yeni bireyler oluşturulur. Buradaki fikir, bu kromozomların evrim geçirmesi ve küresel optimuma ulaşana kadar daha iyi bireyler oluşturma eğiliminde olmasıdır [22].

Popülasyon

Popülasyon, stokastik arama sürecinin başlaması için bir dizi olası çözüm içermektedir. Şekil 3.2 ile gösterilen popülasyon örneğinde, genetik algoritma kabul edilebilir ve optimize edilmiş bir çözüm bulana kadar birden çok nesil boyunca tekrarlanır. Birinci nesil rastgele oluşturulur. Her birey, çözmek istenilen soruna bir çözümdür. *Kromozom*, nesilde veya popülasyonda bulunan bir aday çözümü temsil etmektedir. Bir kromozom aynı zamanda bir *genotip* olarak da adlandırılır. Bir kromozom, optimal değişkenler için değer içeren *gen*lerden oluşur. Genler "0" veya "1" biti ile gösterilir. Bir birey, bu genlerle bir dizi parametre (değişken-özellik) ile karakterize edilir. "1" biti özelliğin seçildiğini, "0" biti ise o özelliğin seçilmediğini temsil etmektedir. Kromozom seçilmiş olan her bir adayın gen özelliğini gösterir.



Şekil 3.2. Popülasyon

Uygunluk fonksiyonu

Uygunluk fonksiyonu, bir bireyin ne kadar uygun olduğunu belirleyen fonksiyondur. Bir

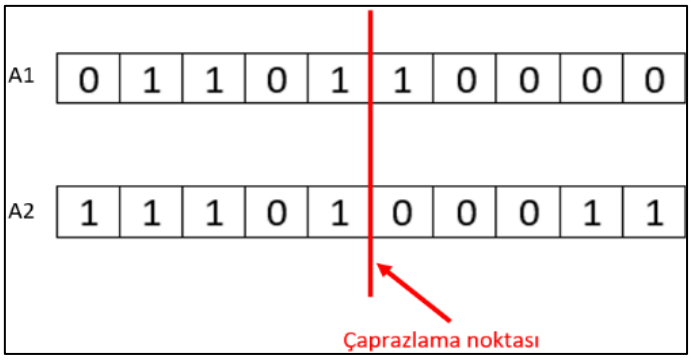
bireyin diğer bireylerle rekabet etme yeteneğini ölçmektedir. Her bireye bir uygunluk puanı verir. Bir bireyin üreme için seçilme olasılığı, uygunluk puanına bağlıdır. İlk iki birey popülasyon içerisinden rastgele seçilir. Sonraki bireyler seçim operasyonuna gönderilir. Uygunluk fonksiyonu hesaplanırken rastgele orman regresyon algoritması kullanılmıştır.

Seçim

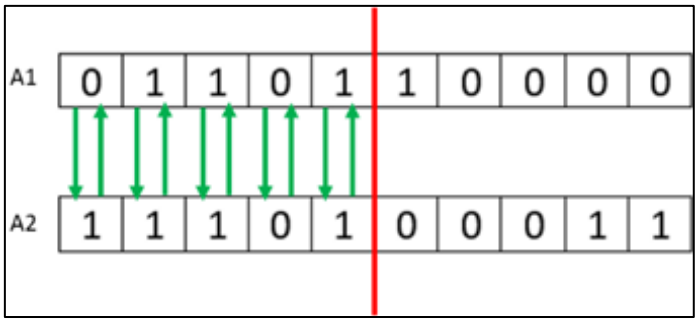
Seçim, bir popülasyondan en uygun çözümü seçme sürecidir. En uygun çözümler, yeni nesil çözümlerin ebeveynleri olarak işlem görmektedir. Bu, gelecek neslin güçlü özellikleri doğal olarak miras almasına izin verir. Seçim, uygunluk değerine bağlı olarak ortalama kare hata yöntemi kullanılarak yapılmaktadır.

Çaprazlama

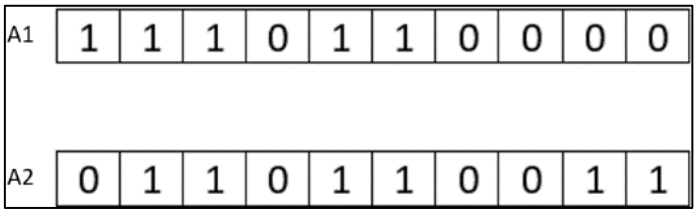
Çaprazlama veya rekombinasyon, en uygun iki ebeveynden gelen genlerin yeni bir genotip veya çözelti oluşturmak için rastgele değiştirilmesiyle gerçekleşir. Çaprazlama, ebeveynin değiştirilen gen segmentlerine dayalı olarak tek noktalı çaprazlama veya çok noktalı çaprazlama olabilir. Bu çalışma içerisinde tek noktalı çaprazlama kullanılmıştır. Çiftleştirilecek her bir ebeveyn çifti için genler içinden rastgele bir çaprazlama noktası seçilir. Örneğin, çaprazlama noktasının Şekil 3.3'teki gibi 5 olduğu varsayılırsa; Şekil 3.4'te gösterildiği gibi çaprazlama noktasına ulaşılana kadar ebeveynlerin genleri kendi aralarında değiş tokuş edilerek yavrular oluşturulur. Yeni bireyler ebeveynlerinden genleri miras alırlar. Şekil 3.5 ile gösterilen yeni bireyler popülasyona eklenir.



Şekil 3.3. Çaprazlama noktası



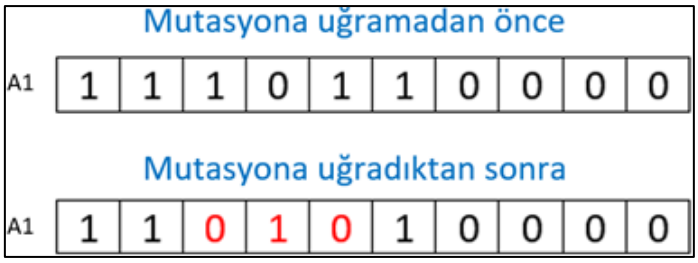
Şekil 3.4. Gen değişimi



Şekil 3.5. Çaprazlama sonrası yeni bireylerin belirlenmesi

Mutasyon

Bu mekanizma, popülasyondaki çeşitliliği teşvik etmek, daha iyi veya optimize edilmiş çözümlere ulaşmak amacıyla genotipi değiştiren rastgele bir süreci temsil eder. Yeni nesillerde, rastgele seçilen bireyler mutasyona uğratılır. Seçilen bireyin genlerinden bazıları, düşük bir rastgele olasılıkla mutasyona uğratılır. Örnek gösterim Şekil 3.6'da verilmektedir.



Şekil 3.6. Genlerin mutasyona uğratılması

Mutasyon, popülasyon içerisindeki çeşitliliği korumak ve erken yakınsamayı önlemek için uygulanır. Yeni popülasyon oluşturulduğunda, her bireyin mutasyona uğrama şansı ‰1' dir. Başlangıç olarak bu oranla başlanılmıştır. Sonra %1 ve %20 oranları da ayrı ayrı olarak denenmiştir.

- Sonlandırma: Algoritma, popülasyon yakınsadığında sona erer. Yani önceki nesilden önemli ölçüde farklı olan yavrular üretilmeyince sonlandırma gerçekleşir.
- Elitizm: Bir sonraki nesle devam edecek olan en uygun adayı seçer.
- Rastgelelik: Bir adayı tamamen rastgele seçer.
- Turnuvalar: Rastgele seçilen 5 adaydan oluşan 9 turnuva gerçekleştirilir. Her turnuvada, çaprazlama için iki kişi (uygunluklarına orantılı olarak seçilme olasılığı ile) seçilir. Çaprazlama: Bireyleri rastgele bir noktada böler ve birleştirir.

Çizelge 3.3 içerisinde, genetik algoritma yönteminin sözde kodu yer almaktadır. Temel olarak genetik algoritma metodunun çalışma mantığı sunulmuştur. Kullanılan ifadeler çalışma içerisinde kullanılan parametre ve değerlerdir. Parametrelerin açıklaması dipnot içerisinde sunulmaktadır.

Çizelge 3.3. Genetik algoritmanın sözde kodu

Genetik algoritma[6]

P
Rastgele orman regresyon algoritması ile uygunluk fonksiyonunu belirlenir
For m
Do While g
Popülasyondan ebeveyn olacak iki kişi seçilir. Yeni neslin ilk elemanları atanır
Diğer elemanlar için turnuvaya gönderilir
While t turnuvaya gönder
Çaprazlama yap
Yeni bireyler popülasyona eklenir.
Oluşturulan popülasyonu kontrol et. Mutasyona uğrayacak mı? Mutasyon uygulanacaksa mutasyon fonksiyonuna gönder
End do
Return N
Elde edilen *N* adet çözüm sınıflandırma algoritmalarına gönderilir
Return

3.3.6. Karınca koloni optimizasyonu

Karınca koloni optimizasyonu fikri ilk olarak Dorigo ve diğerleri tarafından uygulanmış ve buna karınca sistemi adını vermişlerdir [111]. 1996 yılında gezgin satıcı probleminde

[6] P: Başlangıç popülasyonu belirlenir. Özellik sayısı adedince gen içeren 20 kişilik bir popülasyon oluşturulur.
m: Mutasyon sayısıdır. 3 adet mutasyon uygulanacaktır. Çalışma içerisinde ‰1, %1 ve %20 oranları verilir.
g: Nesil sayısı= 100
t: turnuva sayısı= 9
N: Deney sayısı= 10

(travelling salesman problem- TSP) uygulanmak üzere karınca kolonisi dönüştürülmüştür [112]. Karınca kolonisi optimizasyonu, gerçek hayattaki karıncaların yiyecek arama tekniğinden yararlanılan popüler bir optimizasyon algoritmasıdır. Karıncalar, yuvalarından yiyecek ararken belirli bir düzeni izleyerek en iyi yolu bulurlar. İzledikleri iz feromon olarak bilinen özel bir kimyasaldır.

Karıncalar doğada herhangi bir görsel bilgi kullanmadan yuvaları ile besin kaynağı arasındaki en kısa yolu hedeflerler. Bir karınca herhangi bir yoldan geçtiğinde, o yola bir miktar feromon bırakır. Kademeli olarak feromon buharlaşır, böylece tüm yollardaki feromon yoğunluğu azalır. Feromon seviyesi karıncaların kat ettiği popüler yolları net bir şekilde belirleyebilir ve bu nedenle karıncalar daha fazla feromon içeriği olan yolu takip etme eğiliminde olur. Böylece, en kısa yol karıncalar tarafından seçilme eğilimi göstermektedir [98].

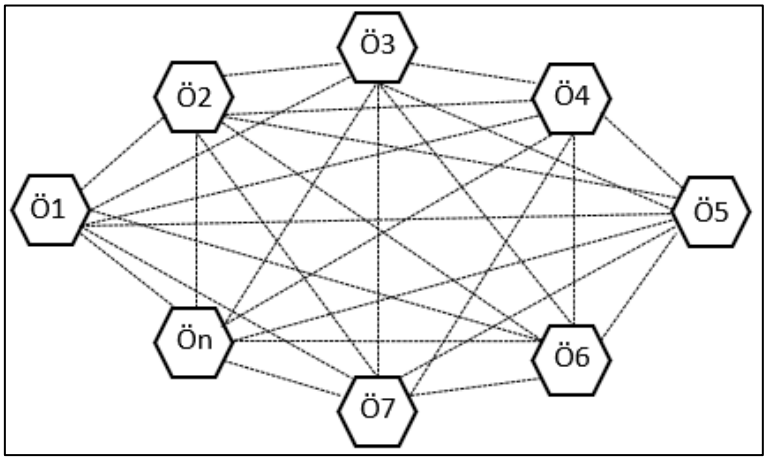
Karınca koloni optimizasyonunda kullanılan yapay karıncalar, problemleri çözerken sezgisel bilgileri kullanır. Karıncaların yaymakta olduğu feromon izleri, kısmi çözümlere yinelemeli olarak yeni çözüm bileşenleri ekleyerek olasılıksal en iyi çözümleri oluşturan süreçtir. Karıncaların edindiği feromon izi, deneyimlerini yansıtmak amacıyla çalışma zamanında dinamik olarak değişir. Özellikleri düğüm kabul ederek karıncaların ziyaret edebileceği farklı noktaları temsil etmektedir. Bağlantılar tek yönlüdür ve döngü yoktur. Daha önce ziyaret edilen bir noktaya geri dönüş yoktur [113]. Algoritmanın başlangıcında her karınca bir noktada bulunur ve durma koşulu sağlanana kadar en iyi yolu bularak bir çözüm oluşturur. Sonunda her karınca, probleme aday bir çözüm bulmaktadır. Karınca koloni optimizasyonu çalışma adımları:

- Çizge (graph) gösterimi
- Sezgisel arzu edilirlik (heuristic)
- Feromon güncellemesi
- Çözüm oluşturma

Çizge oluşturma

Başlangıçta e-posta spam tespit etme problemi karınca problemi olarak tanımlanır. Şekil 3.7’de gösterildiği gibi her bir özellik karıncaların ziyaret edeceği durak noktası olarak belirtilir. Kenarların bir sonraki özelliklerin seçimini temsil ettiği grafik biçimidir. Tüm

düğümler, herhangi bir özelliğin seçilebilmesine izin vermek için birbirine bağlanır. Her bir karınca belirli bir kural dahilinde bütün durak noktalarına uğramak zorundadır. Bu uğradığı durak noktaları olası çözümü sunmaktadır. Performans değerlendirme puanı ile en iyi çözümü sunan karıncanın seçmiş olduğu güzergâh özellik azaltma yapılacak olan aday listesini sunmaktadır. Yeniden formüle edilmiş grafik kullanılarak, geleneksel karınca koloni optimizasyonunun geçiş kuralları ve feromon kuralı güncellemesi kullanılır. Bu durum için her özelliğin kendine özgü feromonu ve buluşsal değeri bulunmaktadır [89].



Şekil 3.7. Karınca koloni optimizasyonu yapısal gösterimi

Sezgisel arzu edilirlik

Özellik "*i"* buluşsal fonksiyonunun η_i' sinin hesaplanmasında kullanılacak bir başlangıç yerel optimum çözümünü belirlemek için *r* çözümden oluşan bir başlangıç popülasyonu hesaplanır. Başlangıç çözüm popülasyonu, basit birçok başlangıçlı yerel araştırma süreciyle belirlenebilir. Genel olarak algoritmada η_i, yapılacak geçişleri sabitlemek için feromon değeri ile kullanılır. Bu değer, tahmin performansını artırma kabiliyeti temelinde tüm özelliklerin kalitesine ilişkin bir tahmin sağlamaktadır. η_i, başlangıç popülasyonunun r_1 en iyi çözümlerinden ($r_1 < r$, burada r_1 bir parametredir) hesaplanır. En önemli özelliklerin ilk tahmini mevcuttur. Bu nedenle, r_1 en iyi çözümlerinde bulunan özellikler kabul edilir ve her özellik, r_1 en iyi çözümlerinde ortaya çıkma sayısına göre ağırlıklandırılır. En iyi çözümler kümesinde daha sık görülen özelliklere daha büyük bir η_i değeri atanır. Tüm r_1 en iyi çözümlerinde bir *i* özelliği etkinleştirilirse, o zaman $\eta_i = r_1$ olur. Öte yandan, bir özellik hiç etkinleştirilmemişse, o zaman $\eta_i = 0$'dır. j özelliğinde bulunan bir karınca, Eş. 3.11 ile hesaplanan $P_i^k(t)$ olasılığı ile *i* özelliğinin ne zaman seçileceğine karar verir [114].

$$P_i^k(t) = \begin{cases} \frac{[\tau_i(t)^\alpha].[\eta_i]^\beta}{\sum_{u \in j_k}[\tau_u(t)^\alpha].[\eta_u]^\beta} \; eğer\; u \in \; j_k \\ diğer\; 0 \end{cases} \quad (3.11)$$

Burada J_k, kısmi çözüme dahil edilebilecek özelliklerin sayısıdır; τ_i ve η_i, özellik *i* ile ilişkili feromon değeri ve buluşsal arzudur; α ve β feromon değeri ve sezgisel bilginin göreceli önemini belirleyen parametrelerdir. Sömürü-keşif değiş tokuşunu verimli bir şekilde dengelemektedir. j_k gidilebilecek komşuluklar kümesini temsil eder. Keşif ve kullanım arasındaki en iyi denge, α ve β parametrelerinin uygun şekilde seçilmesiyle elde edilmektedir. *α=0* olduğunda, başlangıç çözümlerinde seçilen özellikler daha çok seçilmeye çalışır. Benzer şekilde, *β=0*, buluşsal bilgi olmadan yalnızca feromonun kullanıldığını gösterir. Daha sonra, her karıncanın uygunluğu belirlenir ve Eş. 3.11 formülü kullanılarak her karınca en yüksek olasılıkla $P_i^k(t)$ özelliği temelinde ziyaret edecek bir sonraki özelliği seçmektedir.

Feromon güncellemesi

Tüm karıncalar ilk çözümlerini bulduktan sonra, tüm düğümlerin feromon buharlaşması başlar ve her *k* karınca Eş. 3.12'de verilen miktarda feromon biriktirir [114].

$$\Delta\tau_i^k(t) = \begin{cases} \emptyset.\gamma(S^k(t)) \; + \; \frac{\emptyset(n-|S^k(t)|)}{n}, eğer\; i \in \; S^k(t) \\ diğer\; 0 \end{cases} \quad (3.12)$$

Burada $S^k(t)$, karınca *k* tarafından *t*' inci yinelemesinde üretilen özellik alt kümesidir ve $|S^k(t)|$ uzunluğu, $\emptyset$ özellik altkümesi uzunluğunun bağıl ağırlığını kontrol eden parametredir ve *0* ile *1* arasında değişmektedir. Önerilen bu algoritmada sadece en iyi karınca feromonu kendi özelliklerinde bırakmaktadır. Pratik olarak, feromon buharlaşması, Eş. 3.13'te verilen kurala göre tanımlanır [115].

$$\tau_i(t+1) \; = \; (1-\rho)\tau_i(t) \; + \; \sum_{k=1}^{m} \Delta_i^k(t) \; + \; \Delta\tau_i^g(t) \quad (3.13)$$

Burada *m*, her yinelemedeki karıncaların sayısı ve *(0, 1)* feromon izi buharlaşma katsayısıdır. Feromon buharlaştırmanın başlıca kullanımı durgunluğu önlemektir, her karınca feromon güncellemesinden geçer ve en iyi karınca en iyi çözümün düğümlerine daha fazla feromon

bırakır. Sonraki yinelemelerde karıncaların optimal çözüm etrafında keşif yapmasıyla sonuçlanır.

Çözüm oluşturma

Süreç, grafiğe yerleştirilecek karıncaların sayısı belirlenmesiyle başlar. Ardından karıncalar rastgele yerleştirilir. Her karınca, benzersiz bir özellikten yol oluşturmaya başlar. Bu başlangıç konumlarından, durma koşulu sağlanana kadar düğümleri olasılıksal bir şekilde çaprazlarlar. Ortaya çıkan özellik alt kümesi toplanır ve optimal alt küme için Rastgele Orman Sınıflandırıcı ile hesaplama yapılır. En iyi değere sahip alt küme tanımlandığında, prosedür sonlandırılır ve en iyi özellik alt kümesi not edilir. Koşullar karşılanmadığında feromon güncellemesi gerçekleşir, yeni bir karınca grubu üretilir ve süreç tekrarlanır.

Çizelge 3.4. Karınca koloni optimizasyonu sözde kod

Karınca koloni optimizasyonu sözde kod[7]

```
K
Feromon matrisi F ile oluşturulur
Sezgisel fonksiyonunu çağır
Başlangıç feromon tanımlanır
While N
    While epoch
        Her karınca tüm özellikleri rastgele gezerek kendisine güzergâh oluşturur
        Durum listesi alınır //Durumu 1 olanlar seçilen özelliklerdir
        Karınca sayısınca performans puanı hesaplanır //En iyi performansa sahip karınca hafızaya alınır
        Feromon güncelleme fonksiyonu çağrılır
        Return hafızada bulunan en iyi karıncanın güzergahı döndürülür
    End
Return deney sayısınca özellik listesi döndürülür
End
```

Karınca koloni optimizasyonunun çalışma sözde kodu Çizelge 3.4 ile verilmiştir. Düğüm olarak tanımlanan veri setindeki özelliklere giden karıncalar rastgele her bir düğüme giderler. Her bir düğüm için başlangıç feromonu hesaplanır. Gidilen her düğüme hesaplama

[7] K: Karınca sayısı= 100

F: Özellik sayısı= 57

N: Deney sayısı= 10

epoch: bir deney içerisindeki öğrenme sırası= 50

yapılarak *0/1* değeri atanır. *0* seçilemeyen özellikleri, *1* ise seçilen özellikleri temsil etmektedir. Her bir karınca güzergahını tamamladıktan sonra öğrenme performansı hesaplanır. En iyi performansa sahip karınca hafızaya alınır. Sonrasında her bir karınca feromon güncelleme fonksiyonuna gönderilir. İşlemler tekrar sayısınca yinelenir. *N* adet deney çalışması yapıldığı için başlangıç feromon hesaplanmasından itibaren deney sayısınca tekrarlanır. Sonuçta elde edilen *N* adet özellik kümesi sınıflandırma çalışmasına gönderilir. Dipnot ile gösterilen değerler tez çalışmasında kullanılmış olan değerlerdir.

3.4. Sınıflandırma Algoritmaları

Özellik seçim çalışmaları tamamlandıktan sonra, seçilmiş olan özelliklerin sınıflandırmaya etkisini ölçebilmek amacıyla çeşitli algoritmalar kullanılmıştır. Bu başlık altında tez çalışmasında kullanılmış olan algoritmaların tanıtılması yapılmaktadır. Lojistik regresyon, rastgele orman, naive bayes, k-en yakın komşu, adaboost, doğrusal destek vektör makineleri, karar ağacı ve derin sinir ağları algoritmaları kullanılmaktadır. Kullanılan bu algoritmaların nasıl kullanıldığı, grafikleri, formülleri, işlem adımları, avantaj ve dezavantajları verilmek istenmektedir.

Her bir algoritmanın hesaplama yöntemi, kullandığı parametreleri farklı olduğu için sonuçlar çeşitlilik göstermektedir. Parametre farklılıklarından dolayı algoritmaların kütüphanelerde bulunan varsayılan değerleri kullanılmıştır. Çeşitliliği fazla tutarak yüksek doğruluğa sahip olan algoritmaların sonraki çalışmalarda kullanılması amaçlanmaktadır.

3.4.1. Lojistik regresyon algoritması

Lojistik regresyon, bağımlı tek değişken ile bir veya daha fazla değişkenler arasındaki ilişkileri karakterize edebilmek amacıyla ikili *(0,1)* olarak sonucu tahmin eden modellemedir. Tahmin edicileri tarafından ayarlanan cevaba dayalı olarak (ham/spam) modeli sınıflandırılır. Dolayısıyla bağımlı değişkeni *Y*, açıklayıcı değişkenlere $X_1, X_2, \ldots, X_n$ bağlamayı mümkün kılan bir modeldir. Açıklanacak değişken nitel (ham/spam) olduğundan Lojistik Regresyon uygulanır. Açıklayıcı değişkenler X_i niteliksel veya niceliksel olabilir. Bu bağımsız değişkenlerin, elektronik postanın ham ya da spam olup olmamasını etkileyen değişkenlerdir [116].

Lojistik regresyon uygulaması, doğrusal regresyona çok yakındır. Doğrusal regresyon, açıklanacak bir nicel değişken *Y*, açıklayıcı değişkenler $X_1, X_2, \ldots, X_n$ şeklinde tanımlanır. *X* ve *Y* arasındaki bağlantıları karakterize etmeyi mümkün kılmaktadır.

$$Y = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \ldots + \beta_n * X_n + \varepsilon \quad (3.14)$$

Eş. 3.14 çoklu doğrusal regresyon modeli: açıklanacak değişken *Y*, β_i katsayılarına ve bir ε gürültü terimine bağlı açıklayıcı değişkenlerin (X_i) β_0 bir kesme noktasının (veya orijindeki ordinatın) bir fonksiyonu olarak ifade edilir.

$$\ln(\frac{p}{1-p}) = \log p = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \ldots + \beta_n * X_n + \varepsilon \quad (3.15)$$

Eş. 3.15 çoklu lojistik regresyon model, açıklanacak değişkenin *Y* ortaya çıkma olasılığının $\log p$'yi, β_i katsayılarına ve bir ε gürültü terimine bağlı açıklayıcı değişkenlerin X_i bir kesme β_0 'ın fonksiyonu olarak ifade edilir.

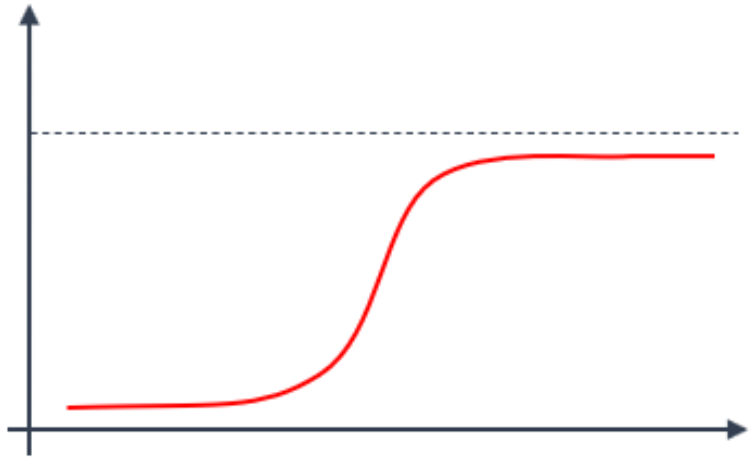
Açıklayıcı değişkenleri tahmin edebilecek niteliksel değişkene *Y*'yi bağlamak için uygun bir model kullanmak gereklidir. Lojistik regresyon yönteminin püf noktası, nitel değişken *Y*'yi modellemek değil, bunun gerçekleşme olasılığını modellemektir. Lojistik model (Eş. 3.15'teki formül), açıklayıcı değişkenlerin X_i bir fonksiyonu olarak bu olasılığın *0* ile *1* arasında monoton olarak değişen doğrusal olmayan bir ifadesine izin verir.

$$p(M + |X) = f(X) = \frac{exp(\beta_0 + \beta X + \varepsilon)}{1 + exp(\beta_0 + \beta X + \varepsilon)} \quad (3.16)$$

Eş. 3.16 ile $(M + |X)$: X değişkeni dikkate alındığında e-postanın spam olma olasılığı $(M+)$; $f(X)$ lojistik fonksiyondur.

Açıklanacak değişken Y'nin bir e-postanın spam olup olmadığını örnek olarak ele alınırsa, değişken ikilidir: spam e-posta $(M+)$ veya ham e-posta $(M-)$ şeklinde ifade edilir. Tek bir açıklayıcı değişken *X* düşünülürse (basit lojistik regresyon), model Eş. 3.16'daki formülle gösterilmektedir. Bu formülde, maruz kalma değişkenine *X*'e bağlı OR (odds ratio- olma olasılığı), β katsayısının üssüne karşılık geldiği görülmektedir. Bu nedenle β katsayısı, açıklayıcı değişken *X* ile açıklanan değişken *Y*, yani spam e-postalar arasındaki ilişkiyi ölçen

olasılık oranının $(\beta = log\ (OR))$ logaritmasına karşılık gelir. Böyle basit bir lojistik regresyon durumunda, lojistik fonksiyon sigmoid bir şekle sahiptir (Şekil 3.8).



Şekil 3.8. Lojistik regresyon sigmoid fonksiyonu

Lojistik fonksiyona göre (-100 ile 100) arasında değişen bir değişkenin olayın meydana gelme olasılığı *p*, *X*'in bir fonksiyonu olarak *p*'yi temsil eden eğri sigmoid bir şekle sahiptir ve monoton olarak *0* ile *1* arasında değişir. Eş. 3.17'de $p(M + |X_1 \ldots X_n) : X_i$ değişkenleri dikkate alındığında spam e-posta olma olasılığı $(M+)$ şeklinde ifade edilir.

$$p(M + |X_1 \ldots X_n) = \frac{exp\{\beta_0 + \sum_i^n \beta_i X_i + \varepsilon\}}{1 + exp\{\beta_0 + \sum_i^n \beta_i X_i + \varepsilon\}} \tag{3.17}$$

Şimdi birkaç açıklayıcı değişkeni X_i (çoklu Lojistik regresyon) ele alınırsa, Sunulan modelin yazımı Eş. 3.17'deki formüle benzemektedir. Her bir açıklayıcı değişken için X_i, bir β_i katsayısı ile ilişkilendirilir ve bu nedenle yine bu katsayının üsteline karşılık gelen ve X_i değişkeni ile söz konusu sonuç *Y* arasındaki ilişkiyi ölçen bir olma olasılığıdır (OR). Her bir açıklayıcı değişken için bu katsayı, modelin diğer açıklayıcı değişkenlerine göre ayarlanmaktadır [117].

3.4.2. Rastgele orman algoritması

Rastgele orman, birbirinden tamamen bağımsız birçok temel sınıflandırıcıdan (karar ağacı) oluşur. Rastgele işlem, her bir karar ağacının bağımsızlığını sağlamak, sınıflandırma doğruluğunu artırmak ve daha iyi genelleme yeteneği elde etmek için örnek alt kümesi ve

özellik alt kümesi seçimini içeren bir süreçtir [118]. Sınıflandırıcıya bir test örneği girilir ve bu örneğin sınıf etiketi, her bir sınıflandırmadan alınan oylama sonuçlarına göre belirlenebilir. $\{h(x,\theta_k), k = 1,\dots\}$ ağaç yapılı sınıflandırıcılardan oluşan bir sınıflandırıcıdır; burada $\{\theta_k\}$ aynı şekilde dağıtılmış rastgele vektörler bağımsızdır ve her ağaç, *x* girişindeki en popüler sınıf için bir birim oy kullanır.

Bu tanım, rastgele ormanın birçok ağaç yapısı sınıflandırıcısının bir kombinasyonu olduğunu göstermektedir. Şekil 3.9 ile temsil edilen görselde birden fazla karar ağacının birleşimi ile oluştuğu gösterilmektedir. Her bir karar ağacı sonuçları birleştirilerek rastgele orman algoritmasının sonucunu vermektedir. Breiman' ın rastgele orman modelinde, her ağaç bir eğitim örneklem seti ve bir rastgele değişken temelinde ekilir, *k.* ağaca karşılık gelen rastgele değişken θ_k olarak gösterilir, bu rastgele değişkenlerin aralarında bağımsızdır. Aynı şekilde sonuç olarak *x*'in giriş vektörü olduğu bir $h(x,\theta_k)$ sınıflandırıcısı paylaşılır. *k* kez çalıştıktan sonra, $\{ h_1(\mathrm{x}), \mathrm{h}_2(\mathrm{x}), \dots, \mathrm{h}_k(\mathrm{x})\}$ sınıflandırıcı dizisi elde edilir ve bunları birden fazla sınıflandırma modeli oluşturmak için kullanılır. Karar fonksiyonu Eş. 3.18;

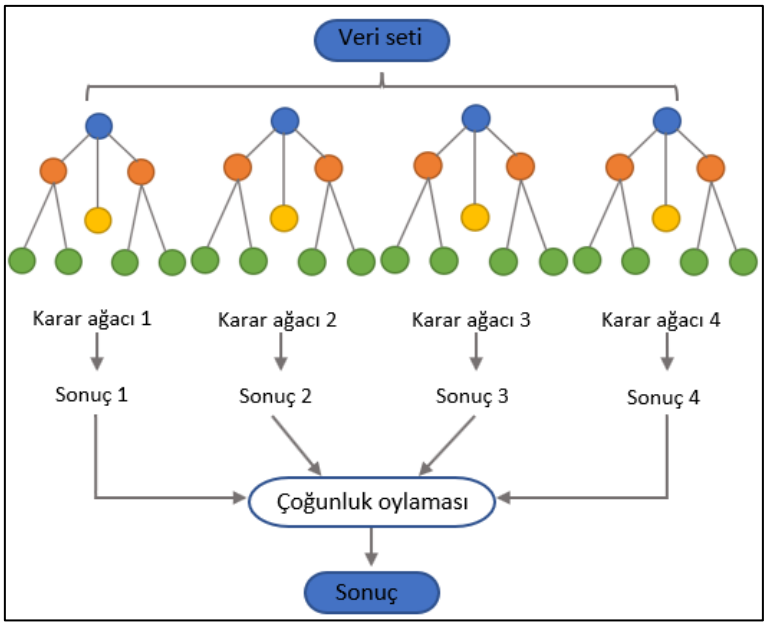
$$H(x) = arg \max_{y} \sum_{i=1}^{k} I(h_i(x) = Y) \tag{3.18}$$

Burada $H(x)$ sınıflandırma modelinin birleşimidir, h_i tek bir karar ağacı modelidir, *Y* çıktı değişkenidir, *I(.)* gösterge işlevidir. Belirli bir girdi değişkeni için, her ağacın en iyi sınıflandırma sonucunu seçmek için oy hakkı bulunmaktadır.

Rastgele orman sınıflandırma algoritmasında, *X* ve *Y* deki doğru sınıf ortalama oy sayısının, yanlış sınıf ortalama oy sayısının ne ölçüde aştığını hesaplayabilmek için marj fonksiyonu kullanılır.

$$mg(X,Y) = av_k I(h_k(X) = Y) - max_{j \neq Y} av_k I(h_k(X) = j) \tag{3.19}$$

Eş. 3.19'da hesaplanan marj değeri ne kadar büyük olursa, sınıflandırma tahmininin doğruluğu o kadar yüksek olur ve sınıflandırmaya olan güven o kadar artmaktadır [119].



Şekil 3.9. Rastgele orman algoritması

3.4.3. Naive bayes algoritması

Naive bayes algoritmasının uygulanabilirliği kolay ve sonuca erişiminin hızlı olması açısından kullanışlı yaklaşımlardandır. Bu algoritma Bayes teoremine dayanmaktadır. Bayes, iki koşullu olasılık arasındaki ilişkiyi açıklamak için bir Eş. 3.20'de bulunan formülü önermiştir;

$$P(A|B) = P(A)\frac{P(B|A)}{P(B)} \tag{3.20}$$

Burada;

- *P(A | B)* = *B*'ye dayalı *A* hipotezinin olma olasılığı
- *P(B | A)* = *A*'ya dayalı *B* hipotezinin olma olasılığı
- *P(B)* = *B*'nin olma olasılığı
- *P(A)* = *A*'nın oluşma olasılığı

Naive bayes algoritması, e-postaların sınıflandırmasında yaygın olarak kullanılmaktadır. Toplanan ham e-posta örnek seti ve spam örnek setinin özellikleri öğrenilip eğitildikten sonra bir model oluşturulur. Ardından, e-posta örnek kümesindeki postanın metin özelliklerini test etmek için bu model kullanılır ve posta metninin ham e-posta olma olasılığını ve spam e-posta olma olasılığını hesaplamak için Eş. 3.20'deki formül kullanılır.

E-postanın ham olma olasılığı spam olma olasılığından yüksekse, o elektronik posta olarak değerlendirilir. Aksi takdirde, spam olarak değerlendirilir [120].

Naive bayes algoritmasının temelde 3 yöntemi bulunmaktadır. Gerçek değerli özellikler içeren veri setlerinde, *gauss dağılımı* kullanılır. İkili özellikler durumunda, $x_i \in \{0,1\}$ *bernoulli dağılımı* kullanılır. Kategorik özellikler durumunda, $x_i \in \{1,\ldots,K\}$ *multinoulli dağılımı* kullanılarak modelleme yapılır [121].

Bu çalışma kapsamında bernoulli naive bayes sınıflandırma algoritması kullanılmaktadır. Algoritma ayrık değişkenler için uygundur. Her karakteristik değişkenin x_i'nin her bir y_i kategorisi altında n-katlı bir bernoulli dağılımına uyduğu varsayılır ve sadece ham/spam şeklinde iki sonuç vardır. Formülü Eş. 3.21'deki gösterilmektedir.

$$P(x_i|y) = P(x_i = 1|y)x_i + (1 - P(x_i = 1|y))(1 - x_i) \quad (3.21)$$

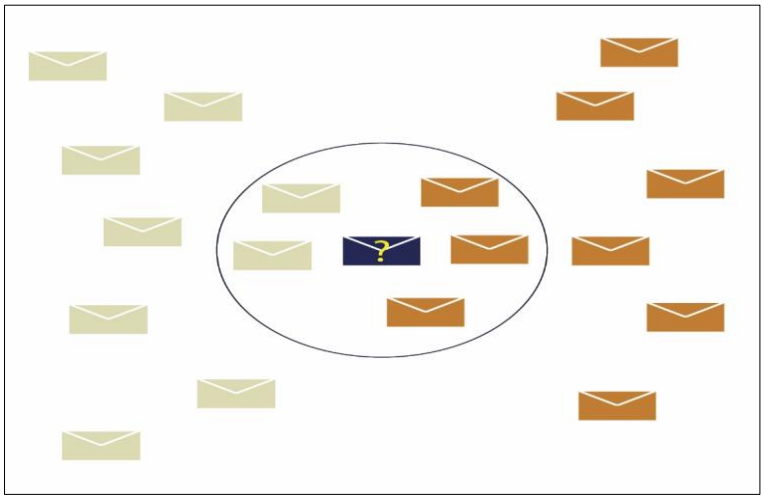
İstenmeyen e-posta sınıflandırması metin sınıflandırmasına ait olduğundan, çoğu zaman metnin özelliklerini ayıklamak ve daha fazla sınıflandırmadan önce bu özellikleri vektörleştirmek gerekir. Bu nedenle bernoulli naive bayes algoritması sıklıkla kullanılmaktadır. Bu algoritma metin analizi yaparken bir kelime görünmüyorsa *0* olarak görüntülenmektedir. Kelime görünüyorsa, kaç kez geçerse geçsin *1* olarak numaralandırılır. Bu nedenle küçük metin kümeleri için uygun bir algoritma modelidir [122].

3.4.4. K-en yakın komşu algoritması

K-en yakın komşu algoritması kullanımının basit ve sınıflandırma performansının etkili olmasından dolayı sıklıkla tercih edilen sınıflandırma yöntemidir. En yakın komşu yöntemi gelen test verinin belirtilen sayıda komşu sınıfından en uygun yere konumlandırması şeklinde çalışmaktadır. Geleneksel k-en yakın komşu algoritmasında komşu sayının (*k*) belirlenmesi kolay değildir. *k* değerini belirlemeye yönelik çalışmalar bulunmaktadır [123].

K-en yakın komşu algoritmasında öncelikli olarak k değeri, pozitif tam sayı olarak küçük ve tek sayı şeklinde belirlenir. Test örneğine en yakın komşu olan k adet eğitim örneğine göre kategorisini tahmin etmeye yardımcı olur. Test verisini en büyük olasılık kategorisine sahip gruba seçer. Algoritmanın basit gösterimi Şekil 3.10 ile verilmiştir. Sınıflandırmadaki k

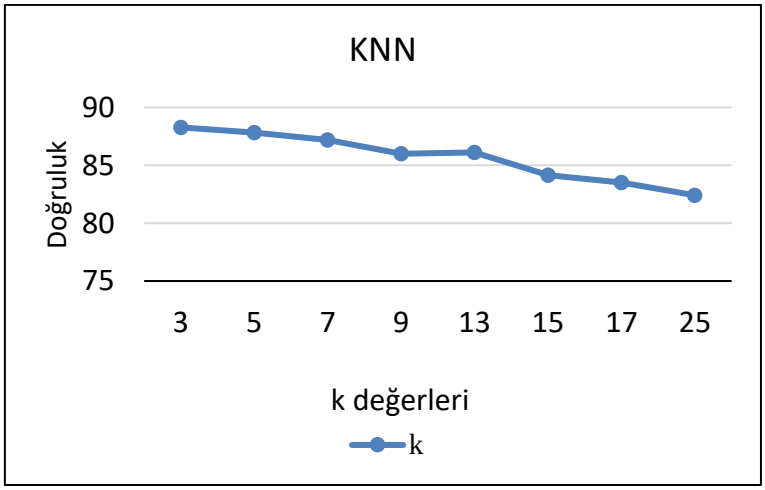
değeri 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21 ve 23 olabilmektedir. Eğer *k=1* ise test örneği en yakın komşunun sınıfına atanır. Bu yöntem, ayrık değerlerle de ilgilenebilmesine rağmen, sayısal veriler için daha uygundur [124]. Çalışma içerisinde *k = 3,5,7,9,13,15,17,25* değerleri denenmiştir. Bu yazılan değerlere göre doğruluk başarım sonuçları Çizelge 3.5' de gösterilmektedir. Verilen k değeri arttıkça performansın azaldığı gözlemlenmektedir. Bundan dolayı tüm çalışma boyunca *k = 3* değeri kullanılmaktadır. Şekil 3.11'de verilen k değerine karşılık gelen doğruluk sonuçlarının grafik hali gösterilmektedir.



Şekil 3.10. K-en yakın komşu algoritması

Çizelge 3.5. Prova edilen k değerleri

k değeri	3	5	7	9	13	15	17	25
Doğruluk	88,27	87,83	87,18	86	86,1	84,14	83,5	82,41



Şekil 3.11. Prova edilen k değerlerine karşılık doğruluk değerleri

k değeri kararlaştırıldıktan sonra belirlenmesi gereken diğer bir parametre veriler arasındaki mesafenin hangi yöntemle hesaplanacak olmasıdır. Algoritma içerisinde mesafeyi hesaplayabilmek için üç yöntem bulunur. Öklid, manhattan ve minkowski yöntemleridir. Bu çalışma içerisinde iki veri noktası arasındaki mesafeyi (benzerliği) hesaplamak için minkowski yöntemi kullanılmaktadır. Minkowski yöntemi, sıklıkla kullanılan mesafe hesaplama yöntemlerinin genelleştirilmiş bir versiyonudur. Eş. 3.22'deki formül ile tanımlanır.

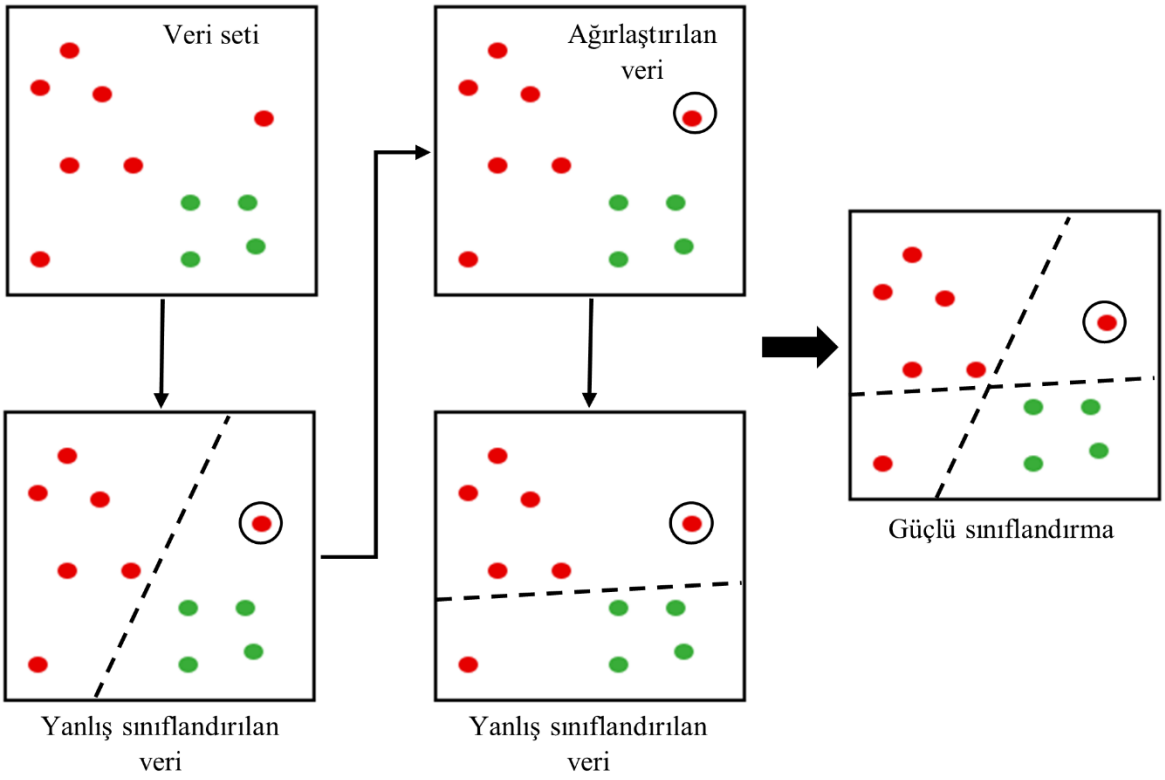
$$(\sum_{i=1}^{n}|x_i - y_i|^p)^{\frac{1}{p}} \tag{3.22}$$

Algoritmanın avantajları arasında basitlik, şeffaflık, güvenilebilirlik ve gürültülü eğitim verilerinde uygulanması kolay şeklinde sayılabilir. Dezavantajları arasında hesaplama karmaşıklığı, bellek sınırlaması, büyük eğitim seti için zayıf çalışma zamanıdır.

3.4.5. Adaboost sınıflandırma algoritması

Adaboost, eğitim veri setinde farklı sınıflandırıcılar kullanan ve bu sınıflandırıcıları entegre ederek en yüksek doğrulukla nihai sonuca ulaşmayı hedefleyen artırıcı bir tekniktir. Adaboost sınıflandırıcı, temel olmayan eğitim verisi özelliklerini filtrelemeye ve en önemli eğitim verisi özelliklerine odaklanmaya olanak tanır. Zayıf sınıflandırıcıların doğrusal kombinasyonu yoluyla güçlü sınıflandırıcılar oluşturur [125-126].

Artırma (boosting) karar ağacı algoritması, sınıflandırma ve regresyon ağaçlarını kullanan artırımlı yöntemdir. Bu durumdaki adaboost yöntemi karar ağacı algoritmasının özel bir durumudur. Eğitim sürecinde her bir zayıf sınıflandırıcı sırayla eğitilerek ağırlıkları elde edilir ve daha önce hatalı sınıflandırılan örnekler dinamik olarak düzeltilir. Önceki zayıf sınıflandırıcılar tarafından yanlış sınıflandırılan örneklerin ağırlığı arttıracak ve tersi ağırlığı azaltılacaktır. Ağırlığın güncellenmesinin amacı, birbirini izleyen her yinelemede en yararlı örnekleri geliştirmektir. Sınıflandırma yapan en son algoritmanın tahmin sonuçları, zayıf sınıflandırıcılar tarafından üretilen tahmin sonuçlarının ağırlıkları toplamıdır [127].



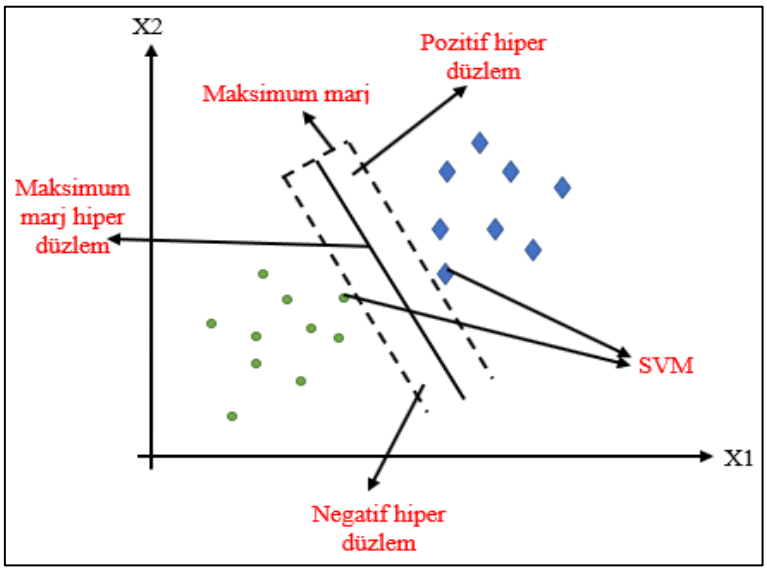
Şekil 3.12 Adaboost algoritması

Şekil 3.12 ile gösterilen resimde bir önceki ağaç tarafından yanlış sınıflandırılan bir verinin ağırlığı artırılır, böylece sonraki ağaç daha önce yanlış sınıflandırılmış veriyi doğru bir şekilde sınıflandırmaya odaklanır. Modele seri olarak daha zayıf sınıflandırıcılar eklendiğinde sınıflandırma doğruluğu artar; ancak bu, ciddi şekilde fazla uydurmaya ve genelleme yeteneğinde düşüşe yol açabilir. Bundan dolayı adaboost' un eğitilmesi daha yavaştır. Adaboost, dengesiz veri kümeleri için uygun bir algoritmadır [128].

3.4.6. Doğrusal destek vektör makineleri

Destek vektör makineleri, makine öğreniminin üstün bir biçimidir. İki gruplu sınıflandırma problemleri için kullanılmaktadır. Ayrı kategoriler için herhangi bir etiketli eğitim verisi verildiğinde, yeni metinleri kategorilere ayırabilir. Sınırlı veri ile sağlandığında optimum kapasitesini gerçekleştirebilir. Doğrusal olarak ayrılabilir veri, kolayca sınıflandırabildiği için destek vektör makinelerinin birincil işlevidir. Destek vektör makineleri modelleri, n-boyutlu uzayları gruplara ayırmak için en iyi çizgiyi oluşturmayı amaçlar, böylece yeni veriler ilerleyen zamanlarda kolayca doğru kategoriye ayrılabilir. En iyi hiper düzlem, eğitim verilerinin marjını en üst düzeye çıkarmasıdır [129]. En büyük dezavantajlarından

biri, mükemmel sınıflandırma sonuçları elde etmek için doğru şekilde ayarlama yapılması şartıyla çok sayıda anahtar parametre gerektirmesidir. Diğer dezavantajı ise, daha fazla eğitim süresi nedeniyle büyük miktarda veri işlendiğinde yavaş çalışmasıdır [130].



Şekil 3.13. Destek vektör makineleri algoritması

Şekil 3.13'te gösterildiği gibi hiper bir düzlem oluşturularak yeni gelen test verilerinin doğru bir şekilde sınıflandırılması hedeflenir.

Destek vektör makineleri genel olarak iki alt bölüme ayrılır:

- Doğrusal destek vektör makineleri (linear SVM), lineer olarak ayrılabilir veridir (tek bir düz çizgi ile iki gruba bölünebilen istatistikler). Doğrusal veriler, doğrusal destek vektör makineleri ile sınıflandırılır. Bu çalışma kapsamında doğrusal destek vektör makineleri kullanılmaktadır.
- Doğrusal olmayan destek vektör makineleri (non-linear SVM), doğrusal olarak ayrılamayan veriler üzerinde çalışır (tek bir satır kullanılarak gruplara ayrılamayan istatistikler). Doğrusal olmayan destek vektör makineleri kullanılarak sınıflandırılır.

Destek vektör makinelerinin genel yapısı, örüntü tanıma ve tahmin gibi problemleri çözme hem doğrusal hem de doğrusal olmayan fonksiyonları analiz etme ve haritalama yapmak amacıyla kullanılabilecek öğrenme yöntemidir. Yüksek boyutlu bir uzay, daha sonra sınıflandırma için kullanılabilecek bir hiper düzlem veya hiper düzlem grubu (sınıflar) oluşturur. Temel olarak matematiksel programlama teknikleri ve çekirdek işlevi üzerinde çalışır. Verilerin regresyon ve sınıflandırması amacıyla da kullanılmaktadır.

Doğrusal destek vektör makineleri üzerinde n eğitim veri seti (x_1, y_1 ile x_n, y_n) sağlanır. İki veri sınıfı arasında büyük bir marj sınıflandırıcısı elde edilecektir. Herhangi bir hiper düzlem, $\vec{x}$ tatmin edici noktaları kümesi olarak belirtilebilir. Eş. 3.23;

$$\vec{w}.\vec{x} - b = 0 \tag{3.23}$$

Burada $\vec{w}$, hiper-düzlemin normal vektörü olarak hizmet eder. 2 tip marj olabilir, sert marj ve yumuşak marj. Eğitim verileri doğrusal olarak ve tamamen hatasız (aykırı değerler ve gürültü) ayrılabilirse, sert kenar boşluğu kullanılır. Hata olması durumunda, kenar boşluğu daha küçüktür veya sert kenar boşluğu başarısız olur. Sert kenar boşlukları aşağıdaki adımlarda oluşturulur:

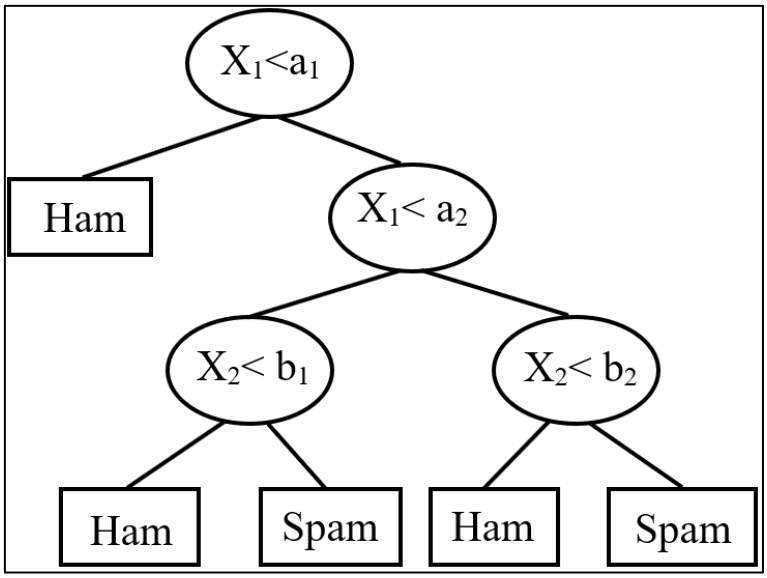
Doğrusal destek vektör makinelerinde, iki veri sınıfı arasında ayrım yapan iki paralel hiper düzlem seçilir, böylece aralarındaki mesafe maksimize edilebilir. Burada yeniden ölçeklendirme yoluyla, veri kümeleri kullanılarak sınırla ayrılmış sınıfların her birini etiketlemek için iki denklemle temsil edilebilir. Sınıflandırıcılar, Eş. 3.24'teki sınırın üzerindeki her şey bir sınıfa, Eş. 3.25'teki belirtilen kısıtlamanın altındaki herhangi bir şey diğer sınıf olarak kabul edilecek şekilde tasarlanmıştır [131].

$$\vec{w}.\vec{x} - b = 1 \tag{3.24}$$
$$\vec{w}.\vec{x} - b = 0 \tag{3.25}$$

3.4.7. Karar ağacı algoritması

Karar ağacı, sıralı bir karar sürecini grafiksel olarak düzenlemenin yoludur. Karar ağacı, her biri alternatif kararların dalları olan karar düğümlerini içerir. Her dalın faydası o dalın yaprağında hesaplanır. Herhangi bir kararın beklenen faydası, girişten başlayıp o daldan tüm yapraklara kadar olan bütün dalların ağırlıkları toplamına göre hesaplanır.

Karar ağaçları, sınıflandırma görevlerinde çok popüler olan hiyerarşik modellerdendir. Şekil 3.14'te baş aşağı gösterilen resim, ağaç yapılarının temel gösterimidir. Kök en üste yerleştirilir ve dallar aşağı doğru büyür. Ağaç düğümleri, tek özellikler kullanılarak mantıksal koşullarla tanımlanır [132].



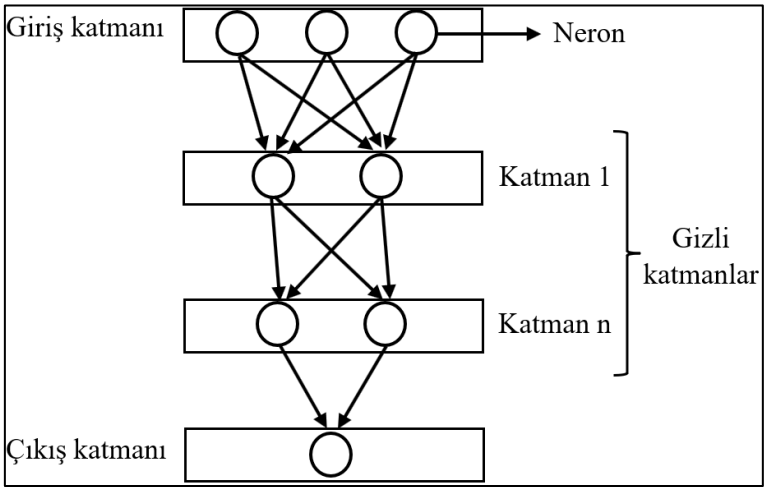
Şekil 3.14. Karar ağacı algoritması

Ağaç büyütme algoritmaları, kök düğüm olarak belirlenen belirli bir eğitim verisiyle başlar ve farklı sınıflara atanan vektörleri ayırmak için düğümleri yinelemeli olarak birkaç ayrık parçaya böler. Her bölme, aslında tüm özellik uzayını (sadece eğitim setini değil) ayrı bölgelere bölen genel bir mantıksal kuralla tanımlanır. Bu nedenle, karar ağaçları genellikle mantıksal kurallar biçiminde sunulabilir. Bu tür anlaşılır modeller birçok alanda avantajlıdır ve belirli özelliklerin uygunluğu hakkında bilgi sağlar. Bununla birlikte, düğümler, alt düğümler, alt ağaçlar, yapraklar, dallar, dal uzunluğu, ağaç derinliği, düğümlere atanan sınıf etiketleri ve bir düğüme düşen veri vektörleri dahil olmak üzere tüm ağaçlar aynı terminoloji ile tanımlanabilir. Genel olarak, bir karar ağacı algoritması üç bileşen tarafından tanımlanır: düğümleri alt düğümlere bölme yöntemi, çoğu durumda sadece bir arama süreci olan ağaç oluşturma stratejisi ve genellemeyi gerçekleştirme yöntemi şeklindedir [133].

3.4.8. Derin sinir ağları algoritması

Derin sinir ağları, giriş ve çıkış katmanları arasında birden fazla gizli katman bulunan bir yapay sinir ağı algoritmasıdır. Bir sinir ağının temel amacı, bir dizi girdi almak, bunlar üzerinde aşamalı olarak hesaplamalar yapmak ve sınıflandırma gibi gerçek dünya problemlerini çözmektir. Tipik olarak derin sinir ağları, girdinin giriş katmanından çıkış katmanına ve gizli katmanlara (iki veya daha fazla) aktığı ve katmanlar arasındaki dayanışmaların bir olduğu ileri beslemeli ağlardır. Çıktılar, geri yayılmaya dayanan etiketli bilgilerin veri kümeleriyle öğrenilerek elde edilir (Şekil 3.15). Derin sinir ağlarının

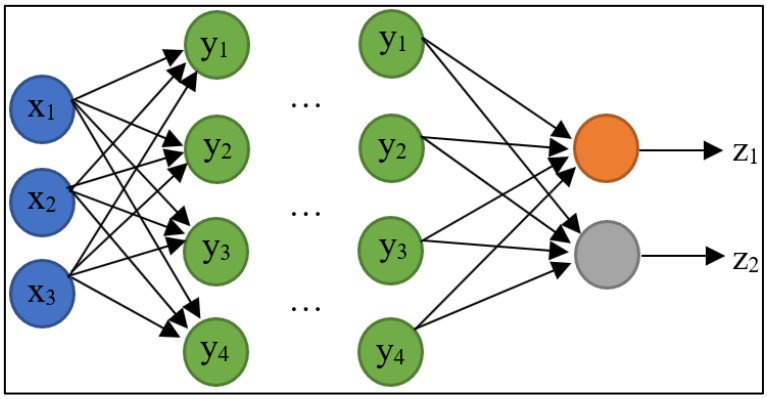
sınırlaması, herhangi bir hafıza birimine sahip olmamasıdır. Karmaşık doğrusal olmayan ilişkileri modelleyebilir. Gizli katman 1 tane olabildiği gibi n adette olabilmektedir.



Şekil 3.15. Derin sinir ağları algoritması

Yapay sinir ağları, denetimli öğrenme ve pekiştirmeli öğrenme problemlerinde yaygın olarak kullanılmaktadır. Bu ağlar, birbirine bağlı bir dizi katmana dayanmaktadır. Derin öğrenmede çoğunlukla doğrusal olmayan gizli katmanların sayısı fazla olabilir. Derin öğrenme modelleri, normal makine öğrenimi ağlarından daha iyi sonuçlar üretmektedir. Ağı optimize etmek ve kayıp fonksiyonunu en aza indirmek için çoğunlukla gradyan iniş yöntemi kullanılmaktadır.

Temel yapay sinir ağları, derinliklerine göre ayrılır; yani, çok adımlı bir örüntü tanıma sürecinde verilerin içinden geçtiği düğüm katmanlarının sayısıdır. Burada bir ağın her katmanının, önceki katmandan alınan belirli özellikleri öğrenmesi beklenmektedir. Katman ne kadar uzaksa, düğümler daha karmaşık özellikleri anlayabilir, çünkü önceki katmanlardaki bu özellikleri toplar ve yeniden birleştirir. Derin öğrenme ağları, çoğu geleneksel makine öğrenimi algoritmasının aksine, insan müdahalesi olmadan otomatik özellik çıkarma işlemi gerçekleştirebilmektedir [134].



Şekil 3.16. Derin sinir ağları katmanlı yapısı

$$y_1 = \sigma(w_{1,1}x_1 + w_{1,2}x_2 + w_{1,3}x_3) \quad (3.26)$$

$$z_1 = \sigma(v_{1,1}y_1 + v_{1,2}y_2 + v_{1,3}y_3 + v_{1,4}y_4) \quad (3.27)$$

Şekil 3.16 da gösterilen yapının matematiksel ifadeleri Eş. 3.26 ve 3.27 ile verilmiştir. (x_1, x_2, x_3) gibi girdilere sahip olan gizli katmalara gerekli işlemler yapıldıktan sonra (z_1, z_2) gibi çıktılara dönüştürülür. Ağırlıklar ve önyargılar katmandan katmana değişmektedir. *w* ve *v*, sinir ağlarının katmanlarının ağırlıkları veya sinapslarıdır. Y'nin bulunduğu gizli katman birden fazla olabilir. Derin öğrenmenin en iyi kullanım durumu, denetimli öğrenme problemidir. Sınıflandırma çalışmalarında fonksiyonel yapıyı tahmin edebilmek için çok katmanlı sinir ağı kullanılmaktadır. Gizli nöron ve gizli katman sayısının tespit edilmesinde belirli bir kural bulunmamaktadır. Bunun için farklı yaklaşımlar söz konusudur. Nöron sayısı arttırılarak yapılan çalışmada eğitim ve test sınıflandırması sabit değere ulaşılması optimum kullanılacak olan nöron sayısını ortaya çıkarmıştır [135].

Aktivasyon fonksiyonu

Yapay sinir ağları, birden fazla yapay nörondan oluşur. Her yapay nöronun, ateşleme hızına karar verebilmek için aktivasyon fonksiyonuna ihtiyaç vardır. Aktivasyon fonksiyonu fikri, merkezi sinir sistemimizde bulunan biyolojik nöron tarafından esinlenilmiştir.

Sigmoid fonksiyonu

Sigmoid fonksiyonu, günümüzde en yaygın kullanılan aktivasyon fonksiyonlarından biridir. Denklemi Eş. 3.28 ile verilir. Sigmoid' in *0* ile *1* arasında bir aralığı vardır.

$$S(t) = \frac{1}{1+e^{-t}} = \frac{e^t}{e^t+1} \quad (3.28)$$

t değişip sigmoid fonksiyonu ile dönüşüme uğrayacağı için *sig(t)* de yeterli değişim fark edilecektir. Bu nedenle, bu öncelikle çıktı katmanında kullanılır.

ReLU fonksiyonu

ReLU (doğrultulmuş lineer birim- rectified linear unit), düzeltilmiş bir doğrusal birim işlevidir. Bu işlev, diğer herhangi bir işleve kıyasla çok daha sık kullanılmaktadır. İstatistiksel olarak Eş. 3.29 da gösterilmektedir.

$$\mathrm{ReLU(x)} = \max(0, \mathrm{x}) \quad (3.29)$$

x, bir nörona giriş verisidir. ReLU, bir nöron tarafından hesaplanan pozitif değeri döndürür ve negatif değerleri yok sayar. Yani herhangi bir düğümde hesaplanan değer negatif ise; *0* döndürür, aksi takdirde hesaplanan değeri iletir. Fonksiyonunun dezavantajı, girdi değeri pozitif ise hiçbir şeyi değiştirmeyecek ve değeri bir sonraki katmanın düğümüne iletecektir. Fonksiyonunun negatif bir değer alması durumunda, nöronu *0* yaparak uyaranları durduracak ve dolayısıyla nöronun ateşlenmesini önleyecektir. Diyelim ki binlerce nöronun olduğu bir modelde; ReLU işlevini kullanarak, bunların birçoğu gereksiz kesme ile sınırlandırılacak ve dolayısıyla daha az ek yük ile hızlı yürütme sağlayacaktır [136].

Softmax fonksiyonu

Sigmoid fonksiyonu gibi ikili sınıflandırma için kullanışlıdır. Softmax, etkili sonuçlar gösterdiğini ve çok sınıflı sınıflandırma problemi için kullanılabileceğini iddia edilmiş. Softmax, *0* ile *1* arasında bir çıkış aralığına sahiptir.

Optimize Ediciler

Optimizasyon algoritmaları, model içinde erişilen tahmin ediciler kümesinden objektif değerlerin hesaplanmasında kullanılan, modelin anayasal sorumlu parametreleri üzerindeki sayısal bir fonksiyonu olan hata fonksiyonunu azaltmaya yardımcı olmaktadır. Dahili öğrenilebilir parametreler, çıktı değerlerinin hesaplanmasında kullanılan sinir ağının önyargı

değerleri ve ağırlıklarıdır. Optimize ediciler, sinir ağlarının eğitim sürecinde ve modelin eğitim sırasında oluşan kayıpların azaltılmasında önemli bir rol oynamaktadır.

Adam

Adam, Jimmy Lei Ba ve Diederik P.Kingma tarafından sunulmuştur [137]. Kompakt bellek özelliği ile birinci dereceden gradyanlara ihtiyaç duyan etkili stokastik gelişim yaklaşımıdır. Optimize edici, gradyanların birinci ve ikinci anlarının değerlendirilmesinden farklı parametreler için ayrık çok yönlü öğrenme oranlarını hesaplar. Adam kelimesi uyarlanabilir moment tahmini (adaptive moment estimation) ifadesinden gelmektedir.

Adamax

Adamax, adam' ın sonsuzluk standardına bağlı bir versiyonudur.

RMSProp

RMSProp, kök ortalama kare yayılımın (root mean square propagation) kısaltmasıdır. Momentumlu gradyan inişine benzer gradyanların üstel olarak ağırlıklı ortalaması kavramını kullanır, ancak parametre güncellemelerinde bir değişiklik sunmaktadır [138].

Nadam

Nadam [139], nesterov hızlandırmalı uyarlanabilir moment tahminin (nesterov accelerated adaptive moment estimation) kısaltmasıdır. μ için zaman adımı μ1…… μT ile indekslenmiş bir değerler listesi varsayılır. Eğimi işlemek için basitçe kuvvet adımıyla parametreleri yenilemek, ardından orijinal parametre durumuna dönmek için bu adımı geri almak, ardından gerçek güncelleme sırasında momentum adımını tekrar almak yerine, zaman adımı t+1'in momentum adımı uygulanır. Adımının güncellenmesi sırasında yalnızca bir kez t+1 yerine önceki t zamanı kullanılır.

3.5. Veri Seti

Çalışmanın amacı özellik seçiminde algoritmaların performanslarını incelemektir. Bu

sebepten dolayı aynı özelliklere sahip tek veri setinin çalışmaya katkısı üzerinde durulmuş ve tek veri setinden yararlanılmıştır. UCI Machine Learning Repository sitesinden erişilen Spambase veri seti, Mark Hopkinsand, Erik Reeber, George Forman, Jaap Suermondt yazarları tarafından 01.07.1999 tarihinde oluşturulmuş. Spambase veri kümesi, tek bir elektronik posta hesabından alınan verileri içeren çok değişkenli bir veri kümesidir. Önceden tanımlanmış 4601 kayıt içermektedir. 1813 kayıt ham (%39,4) ve 2788 kayıt spam (%60,6) olarak sınıflandırılmıştır.

E-posta türünü belirlemek için 57 veri özniteliği ve bir sınıflandırma özniteliği içerir. Ham için 0, spam için 1 şeklinde sınıflandırma yapılır. 1-54 aralığındaki nitelikler belirli sözcükleri veya karakterleri ifade eder. 55-57 arağındaki nitelikler ardışık büyük harf dizilerinin uzunluğunu ölçer. 58 numaralı nitelik, e-postanın spam mı ham mı olarak kabul edilip edilmediğini, yani istenmeyen e-posta olup olmadığını gösterir. "Spam" kavramı çeşitlidir: ürün/web sitesi reklamları, hızlı para kazanma planları, zincirleme mektuplar gibi tanımlanabilir. Niteliklerin çoğu, belirli bir kelime veya karakterin e-postada sıklıkla bulunup bulunmadığını gösterir. Çizelge 3.6'da verilen tabloda Spambase veri setinin içeriği görüntülenmektedir [140].

Çizelge 3.6. Spambase veri setinin özellikleri

Özellik	Veri Tipi	Aralık	Tanım
48	Real(gerçel)	[0,100]	=*word_freq_WORD* türünde sürekli gerçek [0,100] özellik = e-postadaki WORD ile eşleşen kelimelerin yüzdesi, yani 100 * (WORD'ün e-postada görünme sayısı) / e-postadaki toplam kelime sayısı. Bu durumda bir "kelime", alfa sayısal olmayan karakterlerle veya dize sonuyla sınırlanan herhangi bir alfa sayısal karakter dizisidir.
6	Real(gerçel)	[0,100]	=*char_freq_CHAR]* türünde sürekli gerçek [0,100] özellik = e-postadaki CHAR ile eşleşen karakterlerin yüzdesi, yani 100 * (CHAR oluşum sayısı) / e-postadaki toplam karakter
1	Real(gerçel)	-	=sürekli gerçek [1,...] tür *capital_run_length_average* = kesintisiz büyük harf dizilerinin ortalama uzunluğu
1	Integer (tam sayı)	-	=*Capital_run_length_longest* türünde sürekli tamsayı [1,...] özniteliği = en uzun kesintisiz büyük harf dizisinin uzunluğu
1	Integer (tam sayı)	-	= *Capital_run_length_total* türünde sürekli tamsayı [1,...] özniteliği = kesintisiz büyük harf dizilerinin uzunluklarının toplamı = e-postadaki toplam büyük harf sayısı
1	Sayısal	{0,1}	= Spam türünde nominal {0,1} sınıf özelliği = e-postanın (1) spam olarak kabul edildiğini, (0) ham olarak kabul edildiğini

4. DENEYSEL ÇALIŞMALAR

Bu başlık altında sunulan, özellik seçimi uygulamaları, elde edilen özelliklerle yapılan sınıflandırma uygulamaları ve performans sonuçları yer almaktadır. Özellik seçimi çalışmaları çeşitli başlıklar altında incelenmiştir. Filtreleme ve sarmal özellik seçim yöntemleri kullanılarak çeşitli yöntemler denenmiştir. Seçilen özelliklerin sınıflandırmaya katkısını değerlendirebilmek için çok sayıda algoritma seçeneği kullanılmıştır. Filtreleme yöntemleriyle tek bir sonuç elde edilerek doğruluk, duyarlılık, kesinlik ve f1-skor gibi başarı metrikleri kullanılarak performans değerlendirmesi yapılmıştır. Sarmal yöntemlerde ise N kez çalıştırma sonucunda N adet başarı metriği elde edilmiştir. Bu sonuçlar arasından en iyi, en kötü, medyan, ortalama ve standart sapma değerleri hesaplanmıştır. En iyi performansa sahip olan sonuçlar kaydedilmiştir.

Özellik seçimi, özellik seçimi sonrası sınıflandırma uygulamaları, özellik seçimi yapılmadan sınıflandırma ve performans değerlendirmeleri başlıkları altında tez çalışmasında yapılan işlemler anlatılmaktadır.

4.1.Özellik Seçim Çalışması

Özellik, bir veri setindeki hedeflenen sınıflandırmanın sağlanması amacıyla modelin tasarımına yardımcı olan sütun veya niteliklerdir. Her bir özellik, veri setindeki bir öğenin veya birimin belirli bir özelliğini temsil eder. Özellikler, modele giriş olarak sağlanır ve sınıflandırma algoritmasının verileri analiz etmesi ve desenleri tanımlaması için kullanılır.

Özellikler, farklı veri türleri ve uygulama alanlarına bağlı olarak çeşitli şekillerde temsil edilebilir. Örneğin, müşteri verileri içeren bir veri setinde özellikler, müşterinin yaşı, cinsiyeti, gelir düzeyi gibi bilgileri içerebilir. Bir e-posta veri setinde ise özellikler, e-posta gönderenin bilgisi, e-posta içinde geçen kelimelerin frekansı veya karakterlerin yüzdesi gibi özellikler olabilir.

Özellik seçimi için farklı yöntemler kullanılmıştır. Bu yöntemler arasında filtreleme, sarmal ve özellik önem sıralaması gibi teknikler yer almaktadır. Seçilen özellikler, sınıflandırma algoritmalarının eğitim ve test süreçlerinde kullanılmıştır. Özellik seçimi sonucunda elde edilen veri seti, daha az boyuta sahip olmasıyla birlikte gereksiz veya gürültülü özelliklerin

elemesini sağlamıştır. Bu da sınıflandırma algoritmalarının performansını artırmıştır. Özellik seçim çalışması, sınıflandırma algoritmalarının performansını artırmak amacıyla veri setindeki özelliklerin seçilmesi ve kullanılması üzerine odaklanan bir çalışmadır. Yapılan özellik seçimi, sınıflandırma sürecinde gereksiz bilgi girişini azaltarak daha etkili ve verimli bir sınıflandırma sağlamaktadır.

4.1.1. Birliktelik kuralı madenciliği ile özellik seçimi

Birliktelik kuralı madenciliği, veri setlerinde bulunan özellikler arasında belirli ilişkileri tespit ederek gereksiz özellikleri ortadan kaldırmaya yardımcı olan bir yöntemdir. Bu yöntem, geçmişte olduğu gibi günümüzde de araştırmacıların dikkatini çekmektedir. Örneğin, katalog tasarımı, mağaza düzeni veya müşteri satın alma profili oluşturma gibi alanlarda bu yöntem kullanılmıştır [103]. Çalışma içerisinde ise spam tespiti yaparken özellikler arasında belirli ilişkileri kurarak, gereksiz özellikleri tespit edip ortadan kaldırmaya yardımcı olmaktadır.

Sık örüntü madenciliği, bir öğenin toplam işlemde ne sıklıkla kullanıldığını belirlemeye yardımcı olan bir yöntemdir. Burada, sık öğe, belirlenen bir minimum eşik değerini karşılayan öğelerin bir kümesidir. Bu yöntemde kullanılan eşik değeri, destek ve güven metrikleri tarafından belirlenir.

Birliktelik kuralı madenciliği algoritmasının çalışma adımları:

1. Veri seti 10’ a bölünerek 460 e-posta içeren veri kümeleri oluşturulur.

Veri seti uygulamaya alındıktan sonra belirli bir oranda bölme işlemi yapılır. Bölme sayısı artırıldıkça, bölümlerin sayısı da artar. Bu durumda işlem yapılacak olan veri seti kümesi de artar ve işlem süresi uzar. Bu nedenle, bölme işlemi ve veri seti büyüklüğü arasında bir denge sağlanması önemlidir. Deneme amaçlı olarak 10, 20, 50, 100 ve 200 bölme sayıları kullanılmıştır. Her bölmede çalışılan özellik sayısının ve özelliklerin neredeyse aynı olduğu gözlemlenmiştir. Bunun sebebi, veri kümesinin az olmasıdır. İstenilen özellik sayısına ulaşmak için veri seti 10'a bölünmüştür. Bu işlem sonucunda 10 adet veri seti kümesi grupları oluşturulmuş ve her bir veri seti yaklaşık olarak aynı sayıda içeriğe sahip olacak şekilde 460 veri içermiştir. Özellik olarak sayısal veriler kullanıldığı için üzerinde herhangi

bir işleme gerek duyulmamıştır.

Grup halinde çalışmanın sebebi, her bir grupta en fazla kullanılan özelliklerin tespit edilmesidir. Veri seti bölünmeden çalışıldığında ise ortak kullanılan özellik sayısının sınırlı olduğu gözlemlenmiştir, genellikle 2 özellik ile sınırlı kalmaktadır. Bu nedenle gruplama yöntemi kullanılarak veri seti daha etkili bir şekilde analiz edilebilir ve özelliklerin kullanımı daha iyi bir şekilde optimize edilebilir.

2. Her veri seti kümesinde bulunan her özelliğin modu hesaplanır. En sık değeri 0 ise, özelliklere 1 değeri atanır. İçeriğinde değer bulunan özelliklere ise 0 değeri atanır.

Mod, özelliğin en sık görülen değeridir. Her veri seti kümesinde bulunan özelliklerin sık değerleri tanımlanır. Bu sık değerler doğru olarak işaretlenirken, geriye kalan değerler yanlış olarak tanımlanır. Modu bulunan özellikler için, eğer mod değeri 0 ise 1 olarak, içeriğinde değer bulunan özellikler için ise 0 olarak işaretlenir. Tam tersi bir deneme yapıldığında ise sınıflandırma algoritmalarından önemli bir başarı elde edilmemiştir. Bu nedenle, mod değerlerinin kullanılması ve belirli bir işaretlemenin yapılması sınıflandırma performansını artırabilir.

3. Minimum destek değeri 0.3 olarak belirlenerek Apriori algoritmasına veri gönderilir.

Bu değer üzerinde değişiklik yapılarak 0.4 denendiğinde, sınıflandırma algoritması sonuçlarında çok küçük değişiklikler olduğu gözlemlenmiştir. Minimum destek değerinin artırılmasıyla, yani daha sık geçen özelliklerin kullanılmasıyla sınıflandırma performansında hafif bir iyileşme elde edilebilir. Ancak, bu değişikliğin etkisi göreceli olarak küçüktür ve daha yüksek minimum destek değerleriyle sınıflandırma sonuçları üzerinde önemli bir fark yaratmamaktadır.

4. Apriori algoritması için en fazla yapılacak gruplama sınırı olarak 2 değeri seçilir.

Minimum destek değerine eşit veya daha büyük olan gruplamalar listelenmiş ve minimum destek değerinden daha düşük sıklığa sahip özellikler filtrelenerek yalnızca minimum destek değerini karşılayan veya daha yüksek sıklığa sahip özellikler kullanılmıştır.

5. Minimum destek değerine eşit ya da daha büyük değere sahip olan gruplamalar listelenir.
6. Apriori algoritması sonucunda elde edilen değerler birliktelik kuralına gönderilir.
7. Güven metriği ve minimum eşik değeri olarak 1 seçilir.

Apriori algoritmasının çıktıları birliktelik kuralına gönderilmiştir. Bu işlemde güven metriği ve minimum eşik değeri olarak 1 seçilmiştir. Eşik değeri, minimum destek değerine kıyasla daha yüksek bir oluşum sıklığına sahip olan öğeler koleksiyonunu belirler. Minimum destek değerinden büyük bir eşik değeri (0.5 gibi) verildiğinde neredeyse tüm özelliklerin seçildiği, ancak 1.5 ve daha yüksek bir değer verildiğinde hiçbir özelliğin seçilmediği gözlenmiştir.

8. En yüksek güven ve ayrılma değerine sahip olanlardan en düşük değere doğru sıralanır.
9. Her bir veri seti kümesi için bulunan özellikler, bir liste içerisine eklenir.
10. Özelliklerin bir kez kullanıldığı tespit edildiğinde, bu özellikler liste içerisine yazılır. Böylece, kullanılacak olan özelliklerin listesi oluşturulur.
11. Geriye kalan bölünmüş veri setinin 9 tanesine de bu işlem adımları uygulanır.
12. Liste içerisinde bulunan seçilmiş özellikler, bir döngü içerisinde çalıştırılarak kullanılmış olan tüm özellikler belirlenir.
13. Bu döngü, tüm seçilmiş özellikler için tekrarlanır ve en iyi performans sağlayan özelliklerin belirlendiği bir sonuç listesi oluşturulur.

Her bir veri seti kümesinde kullanılan özellikler listelenir. Bu listede yer alan özellikler, sonuçta kullanılacak olan özellik listesini oluşturur. Yani, her bir bölme için kullanılan özelliklerin birleşimi sonucunda nihai liste oluşturulur.

14. Seçilmiş özellikler kullanılarak, sınıflandırma algoritmaları çalıştırılır.

4.1.2. Karşılıklı bilgi özellik seçimi

Karşılıklı bilgi fonksiyonu, bir özellik ile hedef arasındaki bağı ölçerek özellik seçimi işleminde kullanılır. Bu fonksiyon, özelliğin hedefle ne kadar güçlü bir ilişkisi olduğunu ölçer ve güçlü ilişkiye sahip olanları listeleyerek seçer. Karşılıklı bilgi fonksiyonu tarafından hesaplanan değerler ne kadar yüksekse, o özelliklerle hedef arasındaki ilişki o kadar güçlüdür. Bu nedenle, öncelikli olarak kullanılan özellikler standart ölçekleyici (*StandartScaler*) kullanılarak standart hale getirilir. Standart hale getirilen eğitim verisi karşılıklı bilgi fonksiyonundan geçirilir ve ilişki gücü değerlendirilir. Özelliklere atanmış olan değerler büyükten küçüğe doğru sıralanarak listelenir. Karşılıklı bilgi yaklaşımı ise iki

rastgele değişken arasındaki bilgi miktarını ölçer ve giriş ve çıkış arasındaki ilişkiyi ifade eder. Bu yaklaşım, özellik seçimi sürecinde kullanılarak, özelliklerin hedefle olan ilişkisini değerlendirmeye yardımcı olur. Özelliklerin sıralanması, karşılıklı bilgi değerlerine dayanarak yapılır, böylece daha yüksek karşılıklı bilgi değerine sahip olan özellikler öncelikli olarak seçilir.

Karşılıklı bilgi özellik seçimi yönteminde özelliklerin standart hale getirilmesi sonucunda elde edilen grafik EK-1 olarak EKLER bölümünde sunulmuştur. *SelectKBest* fonksiyonu kullanılarak *mutual_info_classif* yöntemiyle istenilen değişken sayısı kadar anlamlı çıktı değişkeni seçilir. En yüksek değere sahip olan 40 adet özellik seçilir. Seçilemeyen özellikler, ilk veri setinden çıkarılarak yeni bir veri seti oluşturulur ve bu veri seti sınıflandırma çalışmalarında kullanılır.

4.1.3. Varyans eşik değeri ile özellik seçimi

Varyans eşiği, veri setindeki düşük varyanslı özellikleri filtreleyerek modelleme sürecinde kullanılmayan özellikleri kaldırmak için bir kullanılan tekniktir. Bu yöntem, çıktılar (y) yerine sadece özellikler (x) üzerinde durur ve bu nedenle denetimsiz öğrenme için kullanılabilir.

- Varyans eşiği = 0 ise (sabit özellikleri kaldırır)
- Varyans eşiği> 0 ise (yarı sabit özellikleri kaldırır)

Farklı eşik değerleri olan 0.1, 0.2, 0.3, 0.25 ve 0.05 uygulanır. Bu eşik değerleri, seçilecek özellik sayısını etkiler. Seçilen özellik sayıları, Çizelge 4.1'de gösterilmektedir. Uygulamalarda, ortalama bir özellik sayısı seçileceği için 0.1 eşik değeri kullanılmıştır.

Çizelge 4.1. Varyans eşik değerine göre seçilen özellikler

Eşik Değeri	Seçilen Özellik Sayısı	Çıkarılan Özellik Sayısı
0.3	23	34
0.25	27	30
0.2	30	27
0.1	41	16
0.075	48	9
0.05	54	3

Eşik değeri küçüldükçe, çalışılmak istenen özellik sayısının arttığı gözlemlenir. Özellik sayısı üzerinde neredeyse aynı etkiye sahip olan 0.075 ve 0.05 değerleri arasında benzer özellik sayısı görülür. Bu elde edilen özelliklerle yapılan denemelerde LSVM ve DT algoritmalarında yüksek dalgalanmaların olduğu gözlemlenmiştir. 0.05 değeri verildiğinde neredeyse tüm özellikler seçilmiştir.

Eşik değeri artırıldığında çalışılacak özellik sayısı azalır. Her bir değer için elde edilen özellikler ayrı ayrı sınıflandırma algoritmalarına gönderildiğinde, sonuçların neredeyse aynı olduğu görülmüştür. Farkın çok büyük olmamasına rağmen, en iyi sonuçların 0.1 değeriyle elde edildiği gözlenmiştir.

4.1.4. Korelasyon özellik seçimi

Korelasyon özellik seçimi, spam e-postaları tespit etmek için kullanılan bir yöntemdir. Bu yöntem, spam e-postaları belirlemek için önemli olan ve birbiriyle güçlü bir ilişkiye sahip olan özellikleri seçmeyi amaçlar. Korelasyon özellik seçimi, spam e-posta sınıflandırmasında kullanılan özellikler arasındaki ilişkiyi değerlendirerek, en etkili özellikleri seçmeye yardımcı olur. Bu sayede, spam e-postaları doğru bir şekilde tespit etmek için en uygun özelliklerin seçilmesine olanak sağlar.

Düşük oranda ilişkili olan özelliklerin veri setinden çıkarılarak yapılan özellik seçimi, yüksek oranda birbirleriyle ilişkili olan özelliklerin seçilmesini sağlar. Geriye kalan özelliklerle sınıflandırma çalışmaları yapılır. İlişkiyi belirlemek için eşik değeri bu süreçte de kullanılır. Çizelge 4.2'de, verilen korelasyon eşik değeriyle seçilen özelliklerin sayısal değerleri listelenir.

Eşik değerinin büyüklüğüne bağlı olarak seçilen özellik sayısında değişiklikler gözlenir. Elde edilen özellikler, sınıflandırma algoritmalarına gönderilerek performans karşılaştırmaları yapılmıştır. 0.1 eşik değeriyle seçilen 43 özelliğin olduğu çalışmanın başarılı olduğu görülmüştür. LSVM algoritmasıyla diğer eşik değerleriyle elde edilen özelliklerle yapılan sınıflandırmada daha fazla dalgalanma olduğu gözlemlenmiştir. Ancak 0.1 eşik değerinde dalgalanmanın az olduğu ve sonuçların daha iyi olduğu tespit edilmiştir.

Çizelge 4.2. Korelasyon eşik değerine göre seçilen özellikler

Eşik Değeri	Seçilen Özellik Sayısı	Çıkarılan Özellik Sayısı
0.5	9	48
0.3	19	38
0.25	21	36
0.2	26	31
0.15	34	23
0.1	43	14
0.05	52	5
0.01	56	1

Her bir özelliğin birbiriyle olan ilişkisini gösteren grafik EKLER bölümünde EK-2 olarak sunulmuştur. Yüksek korelasyona sahip özellikler, genellikle birbirleriyle doğrusal bir şekilde ilişkilidir. Bu durumda, bu özelliklerden biri, diğerlerinin üzerinde benzer bir etkiye sahip olacaktır. Örneğin, iki özelliğin arasında yüksek bir pozitif korelasyon varsa, bir özelliğin değeri arttığında diğer özelliğin değeri de genellikle artar. Bu durumda, bu iki özellik arasında yüksek bir ilişki olduğu söylenebilir.

Yüksek korelasyona sahip özelliklerin birbirlerine benzer bilgi sağladığı düşünülebilir. Özellikler arasında bu yüksek ilişki nedeniyle, bu özelliklerin hepsini kullanmak, gereksiz tekrarlamalara veya fazla bilgiye neden olabilir. Bu durumda, bilgi kaybını minimize etmek ve gereksiz karmaşıklığı azaltmak amacıyla, bu yüksek korelasyona sahip özelliklerden sadece bir tanesi seçilebilir veya çıkarılabilir. Böylece, modelin daha az sayıda özellikle daha iyi bir performans elde etmesi mümkün olabilir. Özellik seçimi bu nedenle, yüksek korelasyona sahip özelliklerin tespiti ve gereksiz olanların elenmesi açısından önemli bir adımdır.

4.1.5. Genetik algoritma ile özellik seçimi

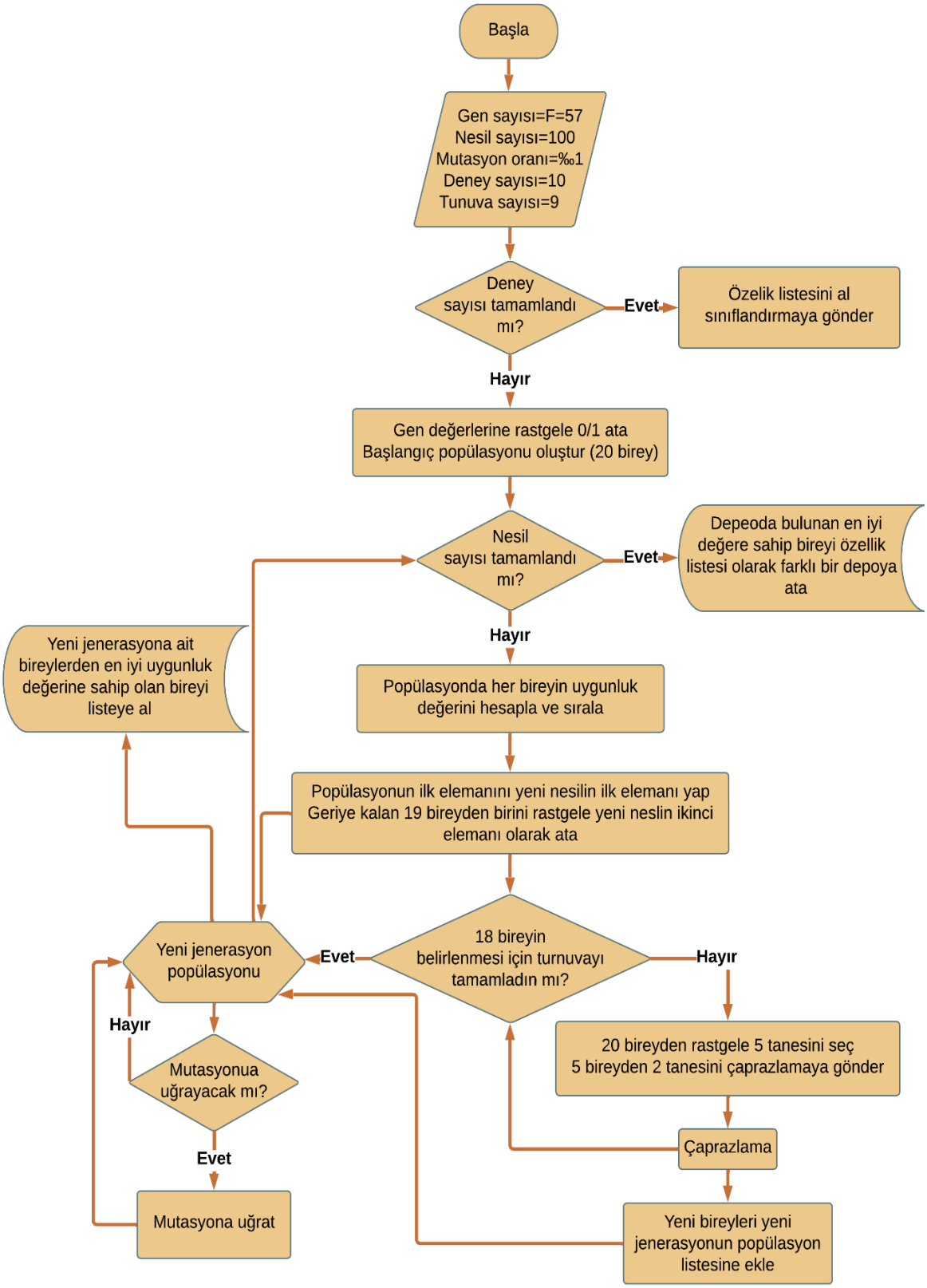
Optimum çözüme ulaşmak ve gereksiz zaman kaybını önlemek amacıyla, günümüzde problemlerin çözümünde makine öğrenimi algoritmalarının yanı sıra metasezgisel algoritmaların kullanımı giderek yaygınlaşmaktadır. Özellikle, karmaşık veya çok boyutlu problemlerle karşılaşıldığında veya bir problemin optimum çözümüne ulaşmak için gereğinden fazla zaman harcanması durumunda, metasezgisel yaklaşımların kullanılması önem kazanmaktadır.

Genetik algoritma, doğal seçilim sürecinden esinlenerek geliştirilen evrimsel yaklaşımlardan biridir. Biyolojiden ilham alan mutasyon, çaprazlama ve seçilim adımlarından oluşur. Bu adımlar, optimum sonuca ulaşmak için en iyi bireyleri bulmayı hedefleyen bir işlem dizinini temsil eder. Genetik algoritma, Spambase veri setine uygulanarak üç farklı mutasyon oranıyla çalışmalar gerçekleştirilmiştir. Bu şekilde, en uygun sonuçları elde etmek için farklı mutasyon stratejileri denenebilir.

Genetik algoritma, özellik seçimi için kullanıldığında üç adımda gerçekleştirilen işlemleri içerir: Seçim, çaprazlama ve mutasyon. Spambase veri setinde, her iterasyonda gelecek neslin genetik operatörler aracılığıyla oluşturulacağı varsayılır. Bu bağlamda, 57 özellik gen olarak kabul edilir ve bu genlerle 20 kişilik bir popülasyon oluşturulur. Genetik algoritma kullanılarak özellik seçimi çalışması, 100 adet nesil üzerinde gerçekleştirilir. En uygun özellik seçimi için N tekrarlama yapılarak en iyi birey listesi oluşturulur, bu liste N adet seçilmiş özelliği içerir. Seçilen özellikler, döngüye aktarılarak her bir birey sınıflandırma algoritmalarına gönderilir. Böylece N adet doğruluk, kesinlik, duyarlılık ve f1-skor başarım değeri elde edilir. Sonuçları görselleştirmek ve analiz etmek için en iyi, en kötü, medyan, ortalama ve standart sapma karşılaştırma fonksiyonları kullanılarak sonuç tablosu oluşturulur. Bu tabloda en iyi değerler sunulmaktadır. Algoritma içerisinde kullanılan parametreler Çizelge 4.3'te verilmiştir.

Çizelge 4.3. Genetik algoritmada kullanılan parametreler

Parametre Adı	Parametre Değeri
Gen sayısı	F=57 (özellik sayısı)
Nesil (Jenerasyon)	100
Popülasyon	20
Seçim türü	Çaprazlama
Turnuva ölçüsü	9
Algoritma öğrenme sırası (epoch)	10
Mutasyon oranı	‰1, %1, %20
Çaprazlama	%25
Uygunluk fonksiyonu	Rastgele orman regresyon
Onaylama (elitizm)	Doğru



Şekil 4.1. Genetik algoritmasının akış diyagramı

Şekil 4.1 de verilen akış diyagramının çalışma adımları aşağıda sunulmaktadır.

Başlangıç:

1. Özellikler gen olarak işlenir, Spambase veri setindeki özellik sayısı (F) kullanılır.
2. Rastgele 0/1 değerleriyle özellik sayısı kadar birey oluşturularak başlangıç popülasyonu oluşturulur (20 birey).
3. Başlangıç popülasyonu, 100 nesil sayısı ve ‰1 mutasyon oranıyla genetik algoritma fonksiyonuna gönderilir.
4. Uygunluk fonksiyonu kullanılarak popülasyon uygunluk değerleri hesaplanır ve başarılıdan başarısıza doğru sıralanır.
5. Uygunluk fonksiyonunda Rastgele Orman Regresyon algoritması kullanılarak tahminler yapılır. Bu tahmini değerleri ile test sınıflandırıcı kullanılarak ortalama kare hata yöntemi ile uygunluk değeri döndürülür.
6. Mevcut popülasyon ve mutasyon oranı yeni popülasyonu belirleyecek olan fonksiyona gönderilir.
7. Mevcut popülasyonun en iyi bireyi yeni popülasyonun ilk elemanı olarak eklenir (elitizm).
8. Geriye kalan 19 bireyden bir tanesi rastgele seçilerek yeni jenerasyona eklenir.

Seçim:

9. Mevcut popülasyon 9 turnuva için çaprazlamaya gönderilir.
10. Turnuva fonksiyonunda mevcut popülasyondan rastgele olarak 5 birey seçilir ve sıralanır.
11. Seçilen bireylerin genleri ve uygunluk fonksiyonları turnuva üye listesine eklenir.
12. Seçilen bireylerin uygunluk değerleri toplanır.
13. Her bireyin uygunluk değeri, toplanan uygunluk değerine bölünerek olasılıkları hesaplanır.
14. Rastgele olarak 2 adet değer seçilir ve bu sıradaki turnuva üyeleri çaprazlamaya gönderilir.

Çaprazlama:

15. Çaprazlama noktası belirlenir ve tek noktalı çaprazlama yapılır.
16. Çaprazlama noktasından önce genler değiştirilmez, noktadan sonra gen değişimi yapılır.
17. Yeni bireyler yeni popülasyon listesine eklenir.
18. 9 turnuva yapılır çünkü toplamda 20 kişilik popülasyon oluşturulmak istenir.

Mutasyon:

19. Yeni popülasyonda mutasyon yapılıp yapılmayacağı kontrol edilir.
20. Mutasyona uğramamış bireyler döngüye devam eder, mutasyon uğramayanlar üzerinde değişiklik yapılmaz.
21. ‰1 oranında mutasyona uğrayacaksa, rastgele bir mutasyon noktası belirlenir.
22. Belirlenen noktadaki gen değeri 0 ise 1'e, 1 ise 0'a dönüştürülerek tek noktalı mutasyon uygulanır.

Yeni nesil:

23. Birinci nesil için işlemler tamamlanır ve yeni mevcut popülasyon bulunur.
24. İkinci nesil başlar ve uygunluk fonksiyonuna gönderilerek yeni mevcut liste oluşturulur. İlk eleman ikinci jenerasyonun başlangıç noktası olarak seçilir. Bu işlemler 100 kez tekrarlanır (100 jenerasyon).
25. Jenerasyon sayısı 100 olduğunda, en iyi uygunluk değerine sahip jenerasyon sayısı kadar birey ve uygunluk değeri bir listede toplanır. Sonuçlar genetik algoritma fonksiyonundan döndürülür.
26. Uygunluk değeri artan jenerasyonlardan maksimum puana sahip olan seçilir ve ‰1 mutasyon yapılarak elde edilen sonuçlar listesine eklenir. Maksimum uygunluk değere sahip bireyin genleri (özellikleri), sınıflandırma algoritmalarında kullanılmak üzere Excel'e kaydedilir.
27. N tekrarlamada maksimum uygunluk değere sahip bireyler listelenir.
28. Çalışmada mutasyon oranının etkisi incelenmektedir. Bu işlemler %1 ve %20 mutasyon oranları için tekrarlanır.

Çizelge 4.4. ‰1 mutasyon ile elde edilen değerler

No	Uygunluk Değerleri	Seçilen Özellik Sayısı	Çıkarılan Özellik Sayısı
1.	475,5288	32	25
2.	474,2168	34	23
3.	481,7323	34	23
4.	458,3839	31	26
5.	493,3355	32	25
6.	472,8064	33	24
7.	477,9603	33	24
8.	457,8911	32	25
9.	466,6951	31	26
10.	472,0603	32	25
Ort.	473,0611	32,4	24,6

Çizelge 4.4'te verilen durumda, mutasyon oranı ‰1 olduğunda uygunluk değeri ve seçilen/çıkarılan özellik sayıları incelenmiştir. Bu durumda, seçilen özellik sayılarıyla seçilmeyen özellik sayıları birbirine yakın olmuştur, neredeyse yarı yarıya ayrılmışlardır. Uygunluk değerleri dalgalanmalara sahiptir. Bunun sebebi, genetik algoritmanın farklı jenerasyonlarda farklı performans sergilediğini gösterir. Bu dalgalanmalar, her jenerasyonda genetik çaprazlama ve mutasyon işlemlerinin rastgelelik içermesinden kaynaklanır. Bazı jenerasyonlarda uygunluk değerleri artarken, diğerlerinde azalabilir. Bu dalgalanmalar, genetik algoritmanın çözüm alanında keşif ve iyileştirme arasında denge kurmasını sağlar.

Çizelge 4.5. %1 mutasyon ile elde edilen değerler

No	Uygunluk Değerleri	Seçilen Özellik Sayısı	Çıkarılan Özellik Sayısı
1.	480,5824	33	24
2.	464,8776	29	28
3.	474,1999	34	23
4.	473,5878	28	29
5.	492,5529	27	30
6.	476,5249	32	25
7.	468,8077	34	23
8.	468,8844	35	22
9.	490,3589	35	22
10.	481,532	30	27
Ort.	477,1909	31,7	25,3

Mutasyon oranının %1 olduğu durumda uygunluk değeri, seçilen özellik sayısı ve çıkarılan özellik sayısı Çizelge 4.5'te verilmiştir. %1 mutasyon uygulandığında seçilen özellik sayısında daha az dalgalanma olduğu görülmektedir. Ancak, uygunluk değerinde hala dalgalanma mevcuttur. Ortalama uygunluk değeri, ‰1 mutasyona göre daha iyi sonuçlar göstermiştir.

Çizelge 4.6. %20 mutasyon ile elde edilen değerler

No	Uygunluk Değerleri	Seçilen Özellik Sayısı	Çıkarılan Özellik Sayısı
1.	494,4605	32	25
2.	493,9062	35	22
3.	482,1204	28	29
4.	492,2366	33	24
5.	520,0735	28	29
6.	484,3884	34	23
7.	485,3187	35	22
8.	491,2737	35	22
9.	493,4381	28	29
10.	482,0974	36	21
Ort.	491,9314	32,4	24,6

Çizelge 4.6'da, mutasyon oranının %20 olduğu durumda uygunluk değeri, seçilen özellik sayısı ve çıkarılan özellik sayısı verilmiştir. %20 mutasyon ile seçilen özellik sayısında daha fazla dalgalanma olduğu görülmektedir. Ayrıca, seçilen özellik sayısının arttığı da gözlemlenmektedir. Uygunluk değerleri artmış ve diğer iki senaryoya göre daha iyi sonuçlar elde edilmiştir. Ortalama değerin de arttığı gözlemlenmektedir.

Çizelge 4.4, 4.5 ve 4.6'de sunulan tablolarda, farklı senaryolara göre hesaplanan uygunluk

değerleri görülmektedir. Bu tablolardan anlaşıldığı üzere, mutasyon olasılığı arttıkça uygunluk değerlerinde bir artış gözlemlenmektedir. Özellikle %20 mutasyon oranının en yüksek ortalama uygunluk değerine sahip olduğu belirlenmiştir. Bununla birlikte, seçilen özelliklerin sayısı her çalışmada farklılık göstermektedir.

Özellik seçim çalışmasında, mutasyon oranının değişimiyle sonuçlar etkilenmektedir. Farklı mutasyon oranları kullanıldığında, seçilen özellik sayısı dalgalanmalar göstermiştir. Bu da mutasyon oranının özellik seçim çalışmasının sonucunu etkilediğini göstermektedir.

4.1.6. Karınca koloni optimizasyonu ile özellik seçimi

Hayvan davranışlarından esinlenerek geliştirilen çalışmalar, problemlere çözüm arayışında sıkça kullanılmaktadır. Bu çalışmalardan biri sürü zekâsı temelli yöntemlerdir, bu yöntemler biyolojiden ilham alarak geliştirilmektedir [141]. Bu yöntemlerin problemleri çözme potansiyeli, bazı çalışmalarda gösterilmiştir [142].

Karınca koloni optimizasyonu, özellik seçimi yöntemlerinde yaygın olarak kullanılan bir yaklaşımdır. ACO' nun küresel arama yetenekleri, bu yöntemin özellik seçimi problemlerinde etkili olmasını sağlar. Ayrıca, ACO genellikle düşük boyutlu veri setlerine sahip problemler üzerinde kolaylıkla uygulanabilir. Bu durumun sebebi, özellik seçimi probleminde arama uzayının büyüklüğü nedeniyle uygun özellik listesini bulmanın zorlaşmasıdır.

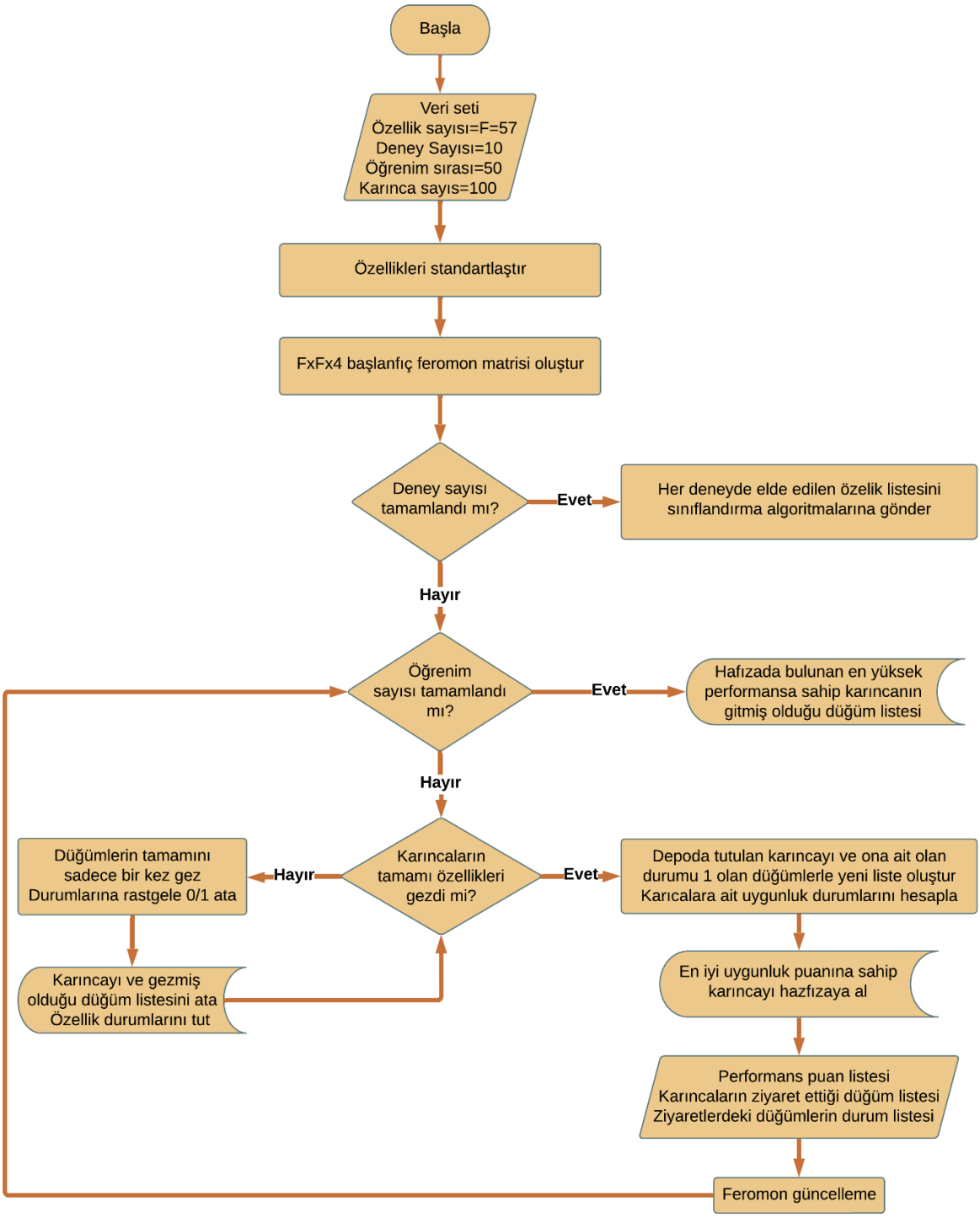
Karınca kolonilerindeki yiyecek arama davranışı esas alınarak ACO, problemlere uyarlanır. Bu uyarlamada, yapay karıncalar kullanılır ve feromon güncelleme formülüyle bir sonraki karıncanın seçeceği yol belirlenir. Her bir karınca, gideceği yolu rastgele seçer ve her bir özellik, ulaşılacak yiyecek hedefi olarak kabul edilir. Karıncaların uygunluk değerleri hesaplanır ve en iyi uygunluk değerine sahip olan karıncanın seçtiği özellikler, kullanılacak özellik listesini oluşturur. Karıncaların feromonları güncellenerek tekrar gidecekleri güzergahlar belirlenir ve bu şekilde ACO iteratif bir süreçle çalışır.

ACO da kullanılan parametreler, Çizelge 4.7 ile gösterilmektedir. Deneylerin bir parçası olarak, uygunluk değerlerine göre en iyi özellik listesi belirlenir ve görüntülenir. Deneyde belirlenen tekrarlama sayısı 50 olarak seçilmiştir ve bu sayıya ulaşmak için denemeler

gerçekleştirilmiştir. Karınca sayısının artırılmasının çözümde iyileşme sağladığı gözlemlendiğinden, karınca sayısı 100 olarak belirlenmiştir. Buharlaşma oranı ise 0,05 olarak kullanılmıştır. Bu parametrelerin seçimi, ACO' nun performansı ve sonuçları üzerinde etkili olmuştur.

Çizelge 4.7. Karınca koloni optimizasyonunda kullanılan parametreler

Parametre Adı	Parametre Değeri	Açıklama
Deney sayısı (iterasyon-N)	10	Algoritmanın döngü sayısı
Epoch	50	Bir deney içerisindeki öğrenme sırası
Başlangıç feromon matrisi	F*F*4	F=57 (özellik sayısı)
Karınca sayısı (K)	100	Algoritma içerisinde kullanılacak olan karınca sayısıdır.
Maksimum ölçek	0.9	scaleMax
Minimum ölçek	0.1	scaleMin
Alfa (α)	1	Feromon izinin etkisi (Sömürü dengesi faktörü)
Beta (β)	0.2	Sezgisel bilgi değeri (keşif dengesi faktörü)
Lambda (λ)	0.005	Doğruluğu en üst düzeye çıkarırken seçilen özellik sayısını en aza indirgemek için amaç işlevini değiştiren parametredir.
P1	0.75	Sezgisel yöntemin seçildiği olasılık
P2	0.1	Sezgisel yöntemin seçildiği olasılık
Uygunluk fonksiyonu		Rastgele orman sınıflandırıcı
Buharlaşma	0.05	Feromon buharlaşma oranı



Şekil 4.2. Karınca koloni optimizasyonu akış diyagramı

Algoritmaya ait akış diyagramı Şekil 4.2 ile verilmiştir. Çalışmanın adımları aşağıda sunulmuştur.

Başlangıç:

1. Veri seti çağrılır.

2. Özellikler standartlaştırılır: Ortalaması kaldırılır ve birim varyansa göre ölçeklendirilir.
3. Özelliklerin sayısı (F) belirlenir.
4. Verilen özelliklerle başlangıç feromon matrisi oluşturmak için ilgili fonksiyona gönderilir. FxF ölçülü 4 adet matris oluşturmak için döngüye gönderilir.
5. Sezgisel önemin kimden kime olacağını belirten 4 adet matris, buluşsal (heuristic) fonksiyonuna gönderilir.
6. Sonuç olarak dönen feromon matrisi, minimum ölçek (scaleMin) ve maksimum ölçek (scaleMax) değerleriyle skala fonksiyonuna gönderilir.
7. Ölçekli matris geriye döndürülür ve başlangıç feromon tanımlanmış olur.

Seçim:

8. Belirlenen döngü sayısınca çalışmak üzere ACO fonksiyonu çağrılır.
9. Özellik sayısı, karınca sayısı ve feromon matrisi girdileri alınır.
10. Her karıncanın bir özelliğe gitmesi sağlanır ve gitmiş olduğu özellik listeden çıkarılır. Sonraki özellik seçimi için yeni liste oluşturulur. Her bir karınca, tüm özelliklere gidene kadar bu döngü devam eder. Her karıncanın gitmiş olduğu özellik sıra listesi kaydedilir.
11. ACO fonksiyonuna, karıncaların hangi sıralamayla özellikleri gezdiğini gösteren liste ve bu sıralama durumunu içeren bilgiler gönderilir.
12. Her bir karıncanın seçtiği özellik listesini öğrenebilmek için karınca sayısı kadar bir döngü başlatılır.
13. Bu karıncaların gitmiş oldukları özellik listesinden, durumu 1 olan özellikler seçilerek yeni bir özellik listesi oluşturulur.
14. Rastgele orman sınıflandırıcı ile doğruluk değeri hesaplanır onun ortalaması performans puanını verir. Karınca sayısı kadar performans puanı elde edilir ve en iyi performansa sahip olan karınca hafızaya alınır.
15. Performans puan listesi, karıncalar tarafından ziyaret edilen düğümlerin (özelliklerin) ve durumlarının listesi feromon güncelleme fonksiyonuna gönderilir.

Feromon güncelleme ve sonuç:

16. Alınan değerlerle feromon matrisi güncellenir.
17. Güncellenen feromon matrisi, scaleMin ve scaleMax değerleriyle skala fonksiyonuna gönderilir. Adım 5'e geçilir ve deney sayısında belirlenen tekrarlama sayısı kadar çalışmaya devam edilir.
18. Hafızada tutulan en iyi performans değeri ile yeni tekrarlanan performans değerleri karşılaştırılır. Yeni en iyi değer bulunduğunda hafızada tutulan karınca bilgisi güncellenir.
19. Öğrenim sırası olarak 50 belirlenmiştir. 50 tekrarlamanın sonunda en iyi performans puanına sahip karıncanın bilgiler hafızada tutulur.
20. Deney sayısı olarak 10 belirlenmiştir. En iyi performansa sahip 10 adet karıncanın seçtiği özellik listesi elde edilir. Sınıflandırma algoritmalarına gönderilerek çalışma tamamlanır.

Çizelge 4.8. Karınca koloni optimizasyonu ile elde edilen değerleri

No	Uygunluk Değerleri	Seçilen Özellik Sayısı	Çıkarılan Özellik Sayısı
1.	0.924794	36	21
2.	0.925014	35	22
3.	0.928056	41	16
4.	0.924146	40	17
5.	0.927187	38	19
6.	0.925882	39	18
7.	0.932184	40	17
8.	0.923490	39	18
9.	0.927186	41	16
10.	0.916319	37	20
Ort	-	38,6	18,4

Karınca koloni optimizasyonu çalıştırılır ve en iyi değerlere sahip olan karıncaların ziyaret ettiği özellikler hafızaya alınır. Ardından, N adet seçilen özellik ve bu özelliklerle elde edilen uygunluk değerleri belirlenir. Bu uygunluk değerleri ile seçilen özellik sayısını belirten tablo, Çizelge 4.8'de sunulmuştur. Karınca koloni optimizasyonu algoritmasının özellik seçimindeki önemi, elde edilen sonuçlarda görülmektedir. Algoritmanın kullanılmasıyla elde edilen uygunluk değerleri daha yüksek gelmiş ve daha fazla sayıda özellik seçilmiştir. Bu durum, karınca koloni algoritmasının etkili bir şekilde özellikleri keşfederek daha iyi performans sağladığını göstermektedir.

4.2. Özellik Seçimi ile Sınıflandırma Çalışması

Seçilen özellikler, filtreleme ve sarmal yöntemlerin performanslarını analiz etmek amacıyla sınıflandırma algoritmalarına gönderilir. Bu özellikler, makine öğrenimi algoritmaları tarafından kolayca temsil edilebilecek bir formatta sunulmaktadır. Literatürde farklı makine öğrenimi algoritmaları sınıflandırma çalışmalarında kullanılmaktadır. Mevcut çalışmalar, makine öğrenimi yaklaşımlarının elde edilen sonuçları hala iyileştirmesi gerektiğini göstermektedir [69] [74].

Veri sınıflandırması, iki aşamalı bir süreç olan öğrenme ve sınıflandırma adımlarını içerir. Bu süreçte, makine öğrenimi algoritmaları kullanılarak veri kümelerinin hangi sınıfa ait olduğu tahmin edilir [143]. Birçok farklı makine öğrenimi algoritması mevcuttur ve sınıflandırma işlemlerinde yardımcı olurlar. Farklı algoritmaların kullanılmasının nedeni, hibrit çalışmalarla yüksek performansa ulaşmanın hangi algoritmalarla mümkün olduğunu belirleyebilmektir. Bu şekilde, her algoritmanın farklı özellikleri ve yaklaşımları

birleştirilerek daha geniş bir çözüm alanı keşfedilebilir ve daha yüksek performans elde edilebilir.

Makine öğrenimi sınıflandırıcılarında genellikle kullanıcı tarafından belirlenen parametreler kullanılır. KNN ve DNN algoritmalarında farklı parametreler denenerek sonuca ulaşılmıştır. Diğer algoritmalarda Python'un Jupyter platformunda bulunan sınıflandırıcı kütüphaneleri kullanılmaktadır. Tez çalışması kapsamında özellik seçimi performans değerlendirilebilmesi amacıyla varsayılan parametreler kullanılmıştır.

DNN sınıflandırıcısı, çoklu parametreler içerir ve çalışma sırasında çeşitli parametreler tanımlanır. En iyi sonucu veren parametreler hafızaya alınır ve elde edilen değerler kullanılarak sınıflandırma çalışmaları gerçekleştirilir. Bu şekilde, DNN sınıflandırıcısı için en etkili parametre kombinasyonu belirlenir ve daha iyi performans elde edilir.

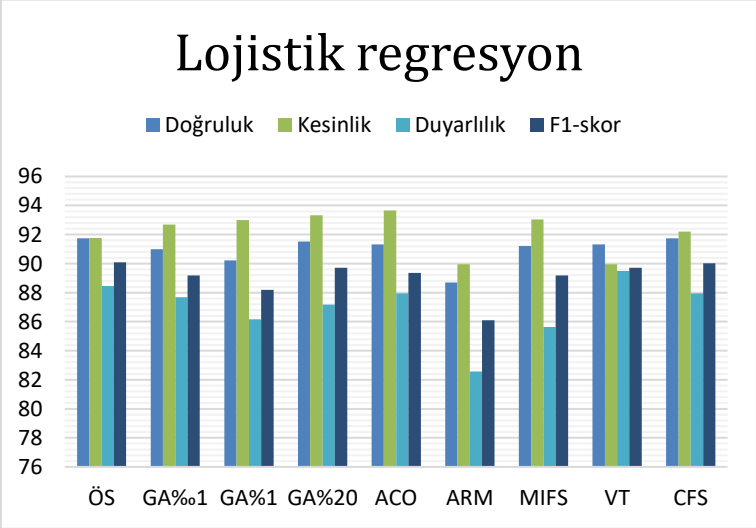
Veri seti, eğitim ve test veri setleri olarak ayrılmıştır. Bu ayrılma işlemi %20 test verisi ve %80 eğitim verisi oranında gerçekleştirilmiştir. Python, her çalıştırmada farklı noktalardan bölme işlemini yapmaktadır. Bu nedenle rastgele sayılar farklı sıralarda üretilir. Ancak, her çalıştırmada aynı eğitim ve test veri setlerini kullanmak için random_state=42 sabit değeri belirlenmiştir. Bu yaklaşım sayesinde, her test işlemi sırasında aynı eğitim ve test veri setleri kullanılarak sonuçların karşılaştırılması sağlanmıştır. Bu tutarlılık, farklı parametrelerin veya algoritmaların performansını doğru bir şekilde değerlendirmemize olanak tanır. Ayrıca, aynı veri seti üzerinde tekrarlanan deneyler yaparak sonuçların güvenilirliğini artırabiliriz ve elde ettiğimiz sonuçları daha iyi yorumlayabiliriz.

4.2.1. Lojistik regresyon ile sınıflandırma çalışması

Makine öğrenimi algoritmaları arasından ilk olarak gerçekleştirilen, lojistik regresyon algoritmasıdır. LR, bağımlı yanıt değişkenini veya daha fazla öngörücü değişkeni tanımlayan verileri analiz etmek için kullanılan bir veri madenciliği yöntemidir. Bu algoritma, kategorik veya numerik verilerden oluşan ve genellikle iki veya daha fazla sınıfa ayrılan verilerin sınıflandırılmasında kullanılır [144]. Çalışma kapsamında, LR algoritması ham/spam sınıflandırması amacıyla kullanılmıştır. Bu sayede, metin verilerinin spam veya ham kategorilerine doğru bir şekilde sınıflandırılması hedeflenmiştir.

Lojistik regresyon algoritması kullanılarak gerçekleştirilen spam sınıflandırması çalışmasında elde edilen sonuçlar genel olarak başarılıdır. Bu algoritma, ham/spam sınıflandırması için kullanılmış ve en iyi sonuçların kesinlik metriği üzerinde elde edildiği gözlenmiştir.

Şekil 4.3'e göre, elde edilen en başarılı sonuçların neredeyse tamamında kesinlik metriğinin öne çıktığı gözlemlenmektedir. ACO algoritmasıyla seçilen özelliklerin kullanıldığı durumda, %93,33 kesinlik metriğiyle en iyi sonucun elde edildiği görülmektedir. Öte yandan, doğruluk ve f1-skor metriklerinde özellik seçimi yapılmadan ve korelasyon kullanılarak yapılan özellik seçiminin başarılı sonuçlar verdiği gözlenmektedir. GA ile denenen farklı mutasyon oranlarında dalgalanmalar görülmektedir, ancak %20 mutasyon oranıyla sonuçların iyileştiği gözlenmiştir.



Şekil 4.3. Lojistik regresyon başarım değerlendirme

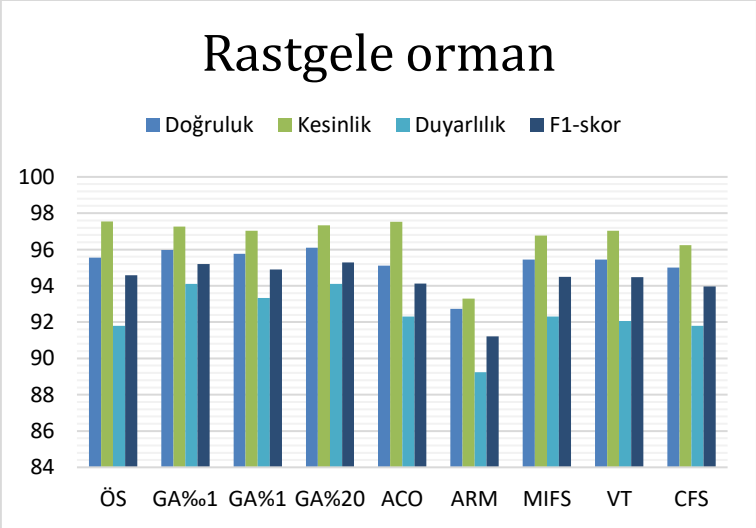
4.2.2. Rastgele orman ile sınıflandırma çalışması

Rastgele orman, birden fazla karar ağacı algoritması kullanarak çalışan bir topluluk sınıflandırıcıdır. Tüm ağaçların çoğunluk oyu kullanılarak bilinmeyen değerlere sınıf ataması yapılır. Bu yöntemde herhangi bir ağacın tek başına optimal olmadığı, ancak birçok ağacın bir araya gelerek küresel optimuma ulaşmada yardımcı olduğu gözlenir [145].

Rastgele orman algoritması kullanılarak gerçekleştirilen spam sınıflandırması çalışmasında, genel olarak yüksek bir kesinlik metriği elde edilmiştir. Bu algoritma, birden fazla karar

ağacını kullanarak spam ve ham e-postaları sınıflandırmada etkili olmuştur. Topluluk sınıflandırıcı olarak çalışan rastgele orman, her bir ağacın tahminlerini çoğunluk oyuyla birleştirerek son bir sınıf atamaktadır. Bu sayede, herhangi bir ağacın tek başına optimal olmasa da birçok ağacın bir araya gelmesiyle daha güvenilir bir sınıflandırma sonucu elde edilmiştir.

Şekil 4.4'teki grafikten genel olarak kesinlik metriğinin iyi sonuçlar elde edildiği görülmektedir. ACO algoritması ile seçilen özelliklerin kullanıldığı rastgele orman algoritmasının performansında %92,52 kesinlik değeri elde edildiği gözlemlenmektedir. Doğruluk metriği açısından en başarılı sonuç, genetik algoritma çalışmasıyla elde edilen %20 mutasyon sonrası seçilen özelliklerin kullanıldığı sınıflandırmada görülmüştür. Bu durumda %96,09 doğruluk değeri en yüksek başarıyı temsil etmektedir. F1-skor metriği incelendiğinde ise GA %20 mutasyon uygulanarak elde edilen özelliklerle %95,28 başarı değeri göstermiştir.



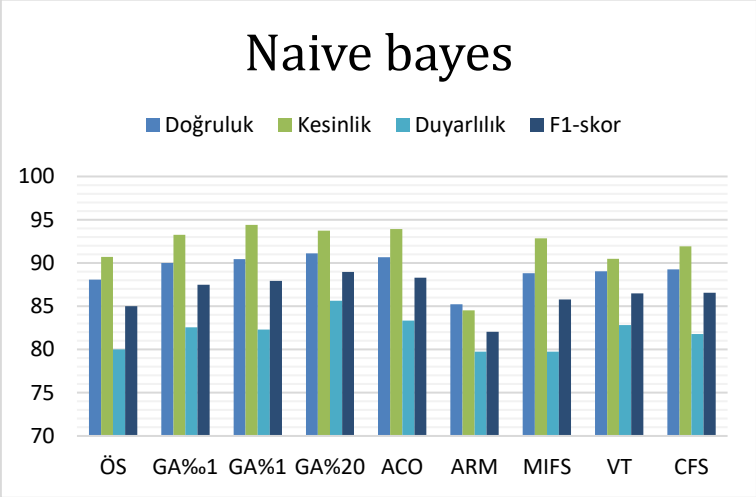
Şekil 4.4. Rastgele orman başarım değerlendirme

4.2.3. Naive bayes ile sınıflandırma çalışması

Naive bayes algoritması kolay uygulanması nedeniyle sıkça tercih edilen bir yöntemdir. Bayes yöntemi, olasılık ve istatistiklerin temel bir sonucu olup, kararları modellemek için kullanılan bir çerçeve şeklinde tanımlanabilir. Çalışmada bernoulli naive bayes algoritması kullanılmıştır. Bu algoritma ile değişkenler koşullu olarak bağımsızdır. Bu yöntem, birbirini doğrudan etkileyen veriler üzerinde bir model belirlemek için kullanılabilmektedir.

Naive bayes algoritması kullanılarak yapılan spam sınıflandırması sonuçları incelendiğinde, bu basit ve etkili algoritmanın yüksek performans gösterdiği görülmektedir. Bernoulli naive bayes algoritmasıyla yapılan sınıflandırmada, değişkenlerin koşullu olarak bağımsız olduğu ve birbirini etkilemediği varsayımıyla modelleme yapılmıştır. Elde edilen sonuçlar, kesinlik metriği üzerinde başarılı bir performans sergilemektedir. Bu nedenle, naive bayes algoritması spam sınıflandırması için sıklıkla tercih edilen bir seçenektir.

Şekil 4.5'te, diğer iki sınıflandırma algoritmasından farklı olarak, genetik algoritmanın %1 mutasyon oranıyla elde edilen özelliklerle yapılan sınıflandırma çalışmasında kesinlik metriğinde %94,41'lik en yüksek sonucun elde edildiği görülmektedir. Ayrıca, genetik algoritmada mutasyon oranı arttıkça doğruluk ve F1-skor değerlerinde bir artış olduğu gözlemlenmektedir.



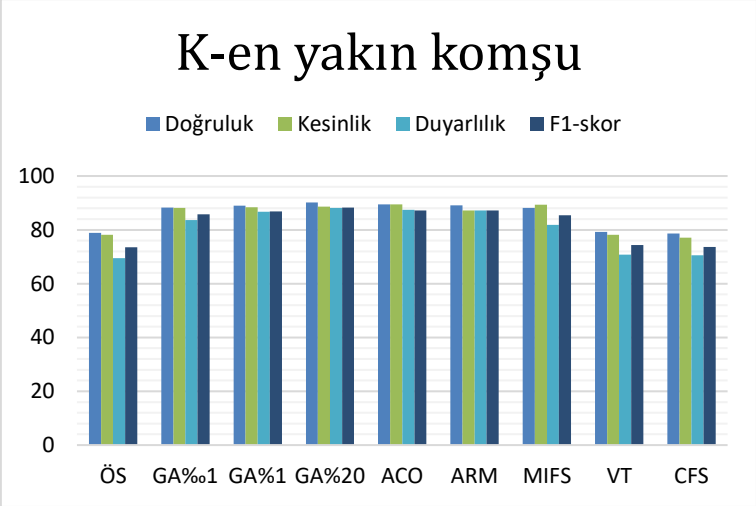
Şekil 4.5. Naive bayes başarım değerlendirme

4.2.4. K-en yakın komşu ile sınıflandırma çalışması

K-en yakın komşu tekniğinde, örnek veri noktalarının sınıflandırılması için en yakın komşuların sayısını belirleyen k değeri kullanılır. KNN sınıflandırıcısı diğer modellerden farklı olarak bir model oluşturmak için eğitilmez. Bunun yerine, her bir bilinmeyen örnek doğrudan orijinal eğitim verileriyle karşılaştırılır. Bilinmeyen örneğe, özellik uzayında en yakın komşularına bakılarak en yaygın sınıf atanır. KNN, bu yaklaşımıyla örnekler arasındaki benzerlik ölçüsünü kullanarak sınıflandırma yapar [146].

K-en yakın komşu algoritmasının performans sonuçları Şekil 4.6'da gösterilmiştir. GA %20,

ACO ve ARM özellik seçim yöntemlerinde elde edilen metrik sonuçları birbirine yakın değerler verdiği gözlemlenmiştir. KNN çalışmasında diğer sınıflandırma algoritmalarından farklı olarak yüksek dalgalanmaların olmadığı görülmüştür ve sayısal olarak diğer algoritmalardan daha düşük sonuçlar verdiği gözlemlenmiştir. En yüksek doğruluk metriği sonucu %90,11 oranında GA %20 mutasyon sonucu yapılan çalışmada elde edilmiştir.

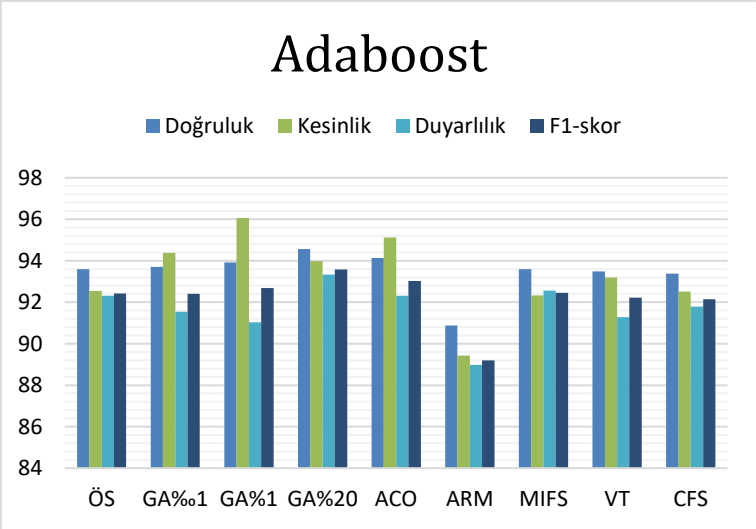


Şekil 4.6. K-en yakın komşu başarım değerlendirme

4.2.5. Adaboost ile sınıflandırma çalışması

Adaboost, güçlendirilmiş karar ağaçlarını kullanarak bir topluluk yöntemi olarak adlandırılmaktadır. Bu yöntem, önceki ağaçların hatalarını en aza indirgemeyi hedefleyerek model oluşturur. Adaboost, uyarlanabilir destek sınıflandırıcı karar ağacının güçlendirilmiş bir türü olarak da tanımlanabilir. Bu sayede Adaboost, etkili bir sınıflandırma yöntemi olarak kullanılabilmektedir [147].

Adaboost sınıflandırıcısının sonuçları Şekil 4.7'de görüntülenmiştir. Sınıflandırma çalışması sonucunda, GA yönteminde kullanılan mutasyon oranının artırılması doğruluk ve f1-skor metriklerinde artış sağlamıştır. En başarılı sonuca GA %1 mutasyon kullanılarak elde edilen çalışmada kesinlik metriği %96'nın üzerinde bir değer vermiştir. KNN sınıflandırmaya göre sayısal değerlendirmeler daha iyidir ve %94,57 doğruluk metriği GA %20 mutasyon kullanılarak elde edilmiştir.

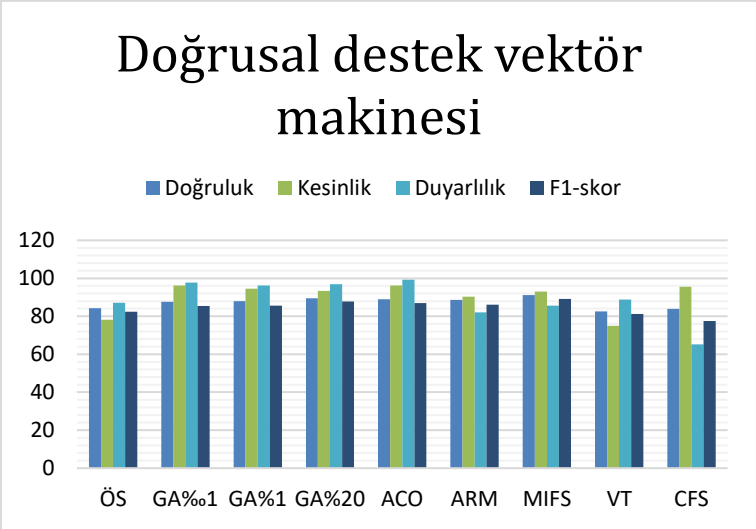


Şekil 4.7. Adaboost başarım değerlendirme

4.2.6. Doğrusal destek vektör makinesi ile sınıflandırma çalışması

Doğrusal destek vektör makinesi, uzun süredir yapılan araştırmalarla desteklenen bir yöntemdir ve elektronik postaların sınıflandırılmasında başarılı sonuçlar vermektedir [148]. LSVM, iki veri sınıfını ayırmak için hiper düzlemin oluşturulduğu bir "marj" kavramına dayanmaktadır. Ayrıştırıcı hiper düzlemi en iyi olanı tahmin etmek için kenar boşluğunu maksimize etmeye çalışır. Doğrusal olarak ayrılabilen verilerde, optimum ayırıcı hiper düzlem bulunduğunda, destek vektör noktaları olarak bilinen veri noktaları kenar boşluğunda yer alır ve çözüm sadece bu noktaların doğrusal bir kombinasyonu olarak temsil edilir. Diğer veri noktaları göz ardı edilir. Spam ve ham verilerin doğru bir şekilde sınıflandırılabilmesi için en uygun marj değerinin bulunması işlemi gerçekleştirilir. Test verilerinin doğru sınıflandırılabilmesi için optimal bir ayrıştırıcı düzleme ulaşmaya çalışılır. LSVM, spam sınıflandırmasında etkili bir yöntem olarak kullanılabilir.

Şekil 4.8'de, GA ve ACO tarafından seçilen özelliklere dayalı olarak yapılan LSVM sınıflandırıcı sonuçlarında benzer bir artış grafiği gözlemlenmektedir. Diğer sınıflandırıcılardan farklı olarak, duyarlılık metriğinde başarılı sonuçlar elde edilmiştir. Özellikle ACO ile elde edilen sonuçlar, %99,23 duyarlılık metriği ile çalışmadaki en iyi sonucu göstermektedir. Ayrıca, doğruluk metriği açısından diğer algoritmalardan farklı olarak MIFS yöntemi ile başarılı sonuçlar elde etmiştir. Bu bulgular, sınıflandırıcının GA ve ACO ile özellik seçimi yapıldığında etkileyici bir performans sergilediğini göstermektedir.

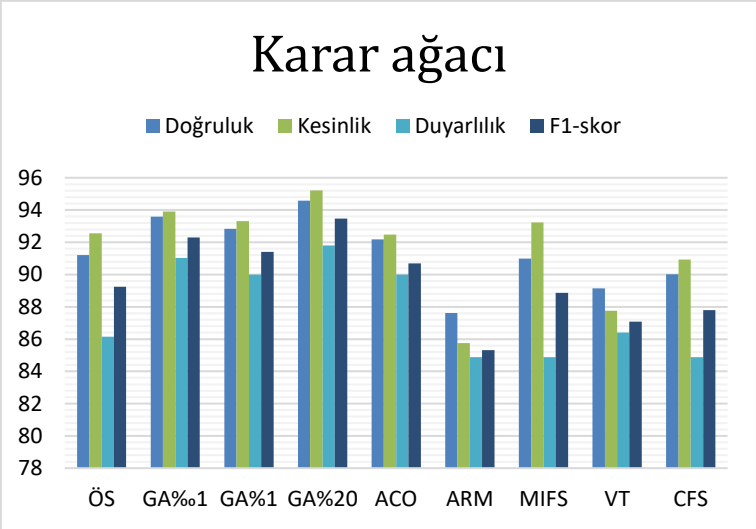


Şekil 4.8. Doğrusal destek vektör makinesi başarım değerlendirme

4.2.7. Karar ağacı ile sınıflandırma çalışması

Karar ağaçları, özellik değerlerine göre örnekleri sınıflandıran ve her düğümde bir özellik ve değer temsil eden ağaçlardır. Örnekler, kök düğümden başlayarak sınıflandırılır ve kolay anlaşılırlığı sayesinde sınıflandırma işlemlerini basitleştirir. Karar ağaçları, kategorik verilerin sınıflandırılmasında avantaj sağlar, örneğin ham/spam gibi sınıflandırma kriterleriyle başarılı bir şekilde çalışır. Model oluşturulduktan sonra farklı çalışmalara kolayca uyarlanabilir. Bununla birlikte, bir dezavantajı, optimum olmayan çözümler üretebilmesi ve aşırı uyum sorunuyla karşılaşma olasılığıdır [149].

Şekil 4.9'da ARM ve VT ile yapılan özellik seçimi çalışması sonucunda elde edilen karar ağacı sınıflandırıcısının başarım grafiği, diğer özellik seçim yöntemlerinden farklılık göstermektedir. Diğer algoritmalarda farklı oranlarda artış gösteren f1-skor, doğruluk ve kesinlik metriklerine sahip sonuçlar elde edilmiştir. GA ile %20 mutasyon çalışması sonuçları en iyi performansı göstermektedir. %94,57 doğruluk değeri, %93,47 f1-skor metriği ve %95,21 kesinlik metriği ile en üstün sonuçlar elde edilmiştir.



Şekil 4.9. Karar ağacı başarım değerlendirme

4.2.8. Derin sinir ağları ile sınıflandırma çalışması

Sınıflandırma algoritmaları, yalnızca doğrusal olarak ayrılabilen örnek kümelerini sınıflandırmak için kullanılır. Bu algoritmalar, girdi örneklerini doğrusal bir çizgi veya düzlemle sınıflandırarak doğru kategorilere ayırır. Ancak, eğer örnekler doğrusal olarak ayrılamazsa, sınıflandırıcı uygun bir çözüm bulamaz ve tüm örnekleri doğru şekilde sınıflandırmak mümkün olmayabilir. Bu problemi çözmek amacıyla yapay sinir ağları geliştirilmiştir. Zhang, yapay sinir ağları alanındaki çalışmalara genel bir bakış sunmak amacıyla bir araştırma makalesi yayınlamıştır [150].

Spambase veri kümesindeki e-postaların ham veya spam olarak sınıflandırmak için Keras kullanılarak derin bir konvolüsyonel sinir ağı eğitilmiştir. Konvolüsyonel sinir ağları sayesinde kendi kendine özellik seçimi yapılıp sınıflandırma yapılabilir fakat burada özellik seçimi filtreleme ve sarmal yöntemlerle yapıldıktan sonra derin sinir ağı sınıflandırıcı şeklinde kullanılmaktadır. Sinir ağı sınıflandırıcısı, bir skor üretmek için ateşlenmiş veya etkinleştirilmiştir. Her düğümün kendi sınıflandırıcısı vardır. Derin sinir ağları, girdi belgelerini ve çıktı özelliklerini kullanarak, girdi, gizli ve çıktı katmanları arasındaki ağırlıkları ayarlayarak eğitir. İleri yayılım adı verilen süreç, giriş katmanından başlayarak puanları bir sonraki gizli katmana iletir ve bu işlem çıktıya ulaşılıncaya kadar devam eder. Çıktı katmanı, verilen belgenin spam veya ham olduğu sınıf etiketiyle ilişkilidir.

DNN, Keras kütüphanesinde tanımlanan bir katman dizinidir. Yapay sinir ağlarında, gizli

katmanın boyutu veya içinde kullanılacak nöron sayısı kesin olarak belirlenemez. Seçilen değerler çalışmaya göre değişebilir. Bu nedenle, çoklu parametre fonksiyonu tanımlanmıştır. Tez çalışmasında kullanılan parametreler Çizelge 4.9'da verilmiştir. Çizelgede gösterilen katman ve nöron değerlerinden en iyi sonucu verenleri kullanılmıştır.

Her bir katmandaki nöronlardan gelen toplam sinyali dönüştüren aktivasyon fonksiyonu çağrılır. Aktivasyon, bir katmandaki nöronların çıktı değerini sonraki katmana iletmek için kullanılan bir fonksiyondur. Bir katmanın çıktısının diğer katmana iletilip iletilmeyeceği, bir eşik değerine göre belirlenir [151]. Bu çalışmada sigmoid, relu ve softmax gibi aktivasyon fonksiyonları kullanılmıştır. Bu fonksiyonlar, eşik değerini belirleyerek çıktıların uygun şekilde iletilmesini sağlar.

Çizelge 4.9. Derin sinir ağları çoklu parametre listesi

Derin Sinir Ağları Parametreleri		
Katman	[80, 60, 45, 30, 15] [120, 80, 60, 45, 30] [30, 20, 15, 10]	Keras kütüphanesinde sinir ağlarının temel yapı taşlarıdır.
Aktivasyon	Sigmoid ReLu Softmax	Doğrusal olmayan gerçek dünya özelliklerini tanıtmak amacıyla kullanılır.
Optimize ediciler	RMSprop Adam [137] Adamax Nadam	Keras modelini derlemek için gereken değişkenlerdendir.
Kayıp fonksiyonu	binary_crossentropy	Her eğitimden sonra elde edilen değerler ile gerçek değerler arasındaki hata oranını hesaplamaya yardımcı olur.
Öğrenme sırası (epoch)	[100,150,200]	Veri setinin modele kaç kez gireceğini belirtir.
Toplu ölçüt (batch_size)	[32,64,128]	Eğitim veya test sırasında girdi bilgilerinin tek tek değil grup halinde eğitime alınmasına yardımcı olur.
Metrik	Doğruluk	Eğitim aşamasında her tur sonrası sonuçları değerlendirebilmek amacıyla sınama işlemi yapar.

Derin sinir ağlarının uygulamalarında başarılı sonuçlar elde edebilmek için hata fonksiyonunun en küçük değeri bulunması gerekmektedir. Bu amaçla optimize ediciler kullanılır. Optimize ediciler, sinir ağının ürettiği çıkış değeri ile gerçek değer arasındaki farkı en küçük yapmaya çalışan yöntemlerdir [139]. Bu çalışmada, en iyi optimize ediciyi bulabilmek için RMSprop, adam, adamax ve nadam yöntemleri kullanılmaktadır.

Tanımlanan katman dizisi, Keras kütüphanesinin yapılandırılmasına bağlı olarak GPU veya

CPU'da çalışacak şekilde bir matris dizisine dönüştürülür. Bu dönüşüm işlemi, belirli bir optimizasyon ve kayıp (loss) fonksiyonu kullanarak gerçekleştirilir. Bu çalışmada ikili sınıflandırma yapıldığı için kayıp fonksiyonu olarak binary_crossentropy kullanılmaktadır. Bu şekilde, ağın eğitim süreci için uygun bir yapı oluşturulur.

Öğrenme sırası (epoch) ve toplu ölçüt (batch_size) sayısı belirlenir. Toplu ölçüt, DNN' nin çalıştırılması sırasında modelin gruplara bölünmesini sağlar. Her bir turda, güncelleme yapılmadan önce ağın maruz kaldığı veri örneklerinin sayısı belirlenir. Bu aynı zamanda bellek kullanımını optimize ederek tek seferde çok fazla girdi verisinin yüklenmesini engeller. Öğrenme sırası ise eğitim verilerinin ağa kaç kez gösterileceğini ifade eder. Eğitim doğruluğunu artırabilmek amacıyla artan üç farklı değer kullanılmaktadır. En iyi uygunluk değerini sağlayan öğrenme sırası değeri kullanılarak sınıflandırma çalışması gerçekleştirilir.

Her özellik seçimi çalışmasından sonra, farklı çoklu parametre değerleri hesaplanır ve uygulamada kullanılır. GA özellik seçiminde üç farklı mutasyon değeri uygulanarak, 3*N adet farklı DNN parametresi elde edilir ve en iyi değerleriyle sınıflandırma çalışması yapılır. ACO çalışmasında ise N adet farklı DNN çoklu parametresi hesaplanır ve sınıflandırma çalışmaları gerçekleştirilir. Diğer özellik seçim yöntemlerinde ise birer adet özellik seçimi yapıldığı için 5 adet en iyi çoklu parametre değeri elde edilir. DNN sınıflandırma algoritmasında kullanılan en iyi çoklu parametre değerleri Çizelge 4.10, 4.11 ve 4.12'de gösterilmektedir.

Çizelge 4.10. Filtreleme yöntemleri ile DNN hibrit çalışmasının parametreleri

	DNN Kullanılan En İyi Değerler				
	Aktivasyon	Öğrenme Sırası	Tur	Katman-Nöron	Optimizasyon
ARM	Sigmoid	32	100	[120, 80, 60, 45, 30]	adam
MIFS	Sigmoid	64	100	[30, 20, 15, 10]	adam
VT	Sigmoid	64	150	[30, 20, 15, 10]	nadam
CFS	Sigmoid	32	100	[30, 20, 15, 10]	nadam
ÖSY	Relu	128	100	[80, 60, 45, 30, 15]	adam

Çizelge 4.10'da filtreleme özellik seçim yöntemlerine ait kullanılan parametreler yer almaktadır. Filtreleme yöntemleriyle beraber özellik seçimi yapılmadan (ÖSY) gerçekleştirilen DNN algoritmasında seçilen parametreler de gösterilmiştir. Optimizasyon fonksiyonu dışındaki parametreler, diğer filtreleme yöntemlerinden farklı olarak seçilmiştir.

Çizelge 4.11. GA ile DNN hibrit çalışmasının parametreleri

DNN Kullanılan En İyi Değerler						
		Aktivasyon	Öğrenme Sırası	Tur	Katman-Nöron	Optimizasyon
GA‰1	1	relu	128	100	[120, 80, 60, 45, 30]	nadam
	2	sigmoid	64	200	[80, 60, 45, 30, 15]	nadam
	3	relu	32	200	[80, 60, 45, 30, 15]	adam
	4	relu	64	100	[30, 20, 15, 10]	nadam
	5	sigmoid	32	200	[120, 80, 60, 45, 30]	adam
	6	sigmoid	32	200	[120, 80, 60, 45, 30]	adam
	7	relu	32	150	[80, 60, 45, 30, 15]	adam
	8	sigmoid	32	200	[120, 80, 60, 45, 30]	rmsprop
	9	relu	32	200	[120, 80, 60, 45, 30]	nadam
	10	relu	32	150	[80, 60, 45, 30, 15]	nadam
GA%1	1	sigmoid	64	100	[80, 60, 45, 30, 15]	rmsprop
	2	sigmoid	32	200	[80, 60, 45, 30, 15]	adam
	3	sigmoid	32	200	[30, 20, 15, 10]	adam
	4	sigmoid	64	200	[30, 20, 15, 10]	nadam
	5	relu	64	200	[120, 80, 60, 45, 30]	adamax
	6	sigmoid	64	150	[120, 80, 60, 45, 30]	nadam
	7	relu	128	150	[80, 60, 45, 30, 15]	adamax
	8	softmax	32	100	[30, 20, 15, 10]	nadam
	9	relu	64	200	[80, 60, 45, 30, 15]	nadam
	10	relu	128	150	[30, 20, 15, 10]	nadam
GA%20	1	relu	32	200	[120, 80, 60, 45, 30]	adam
	2	sigmoid	64	200	[30, 20, 15, 10]	rmsprop
	3	relu	128	200	[120, 80, 60, 45, 30]	adam
	4	sigmoid	64	150	[80, 60, 45, 30, 15]	rmsprop
	5	relu	64	200	[30, 20, 15, 10]	adamax
	6	relu	32	150	[120, 80, 60, 45, 30]	nadam
	7	relu	32	150	[30, 20, 15, 10]	adamax
	8	sigmoid	64	200	[120, 80, 60, 45, 30]	nadam
	9	sigmoid	64	150	[30, 20, 15, 10]	adam
	10	sigmoid	64	200	[120, 80, 60, 45, 30]	adam

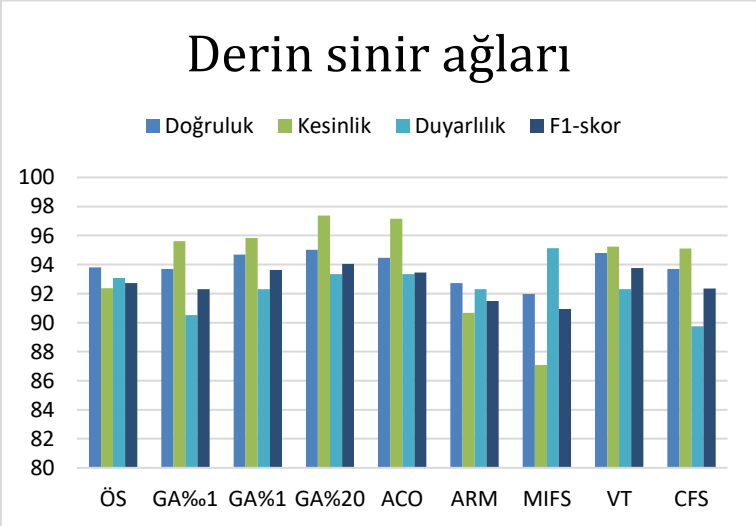
Çizelge 4.11'de GA üzerinde uygulanan mutasyon sıralamasına göre sunulmuştur. Aktivasyon fonksiyonu olarak *relu* veya *sigmoid* fonksiyonları ağırlıklı olarak kullanılmaktadır. Toplu ölçüt olarak kullanılan birimlerin küçük parçalara bölünerek çalışması tercih edilmektedir.

Çizelge 4.12'de ise ACO yönteminde kullanılan parametreleri göstermektedir. Aktivasyon fonksiyonu olarak *sigmoid* ve *relu* arasında değişim mevcuttur ve genellikle *adam* optimizasyon yöntemi tercih edilmiştir.

Çizelge 4.12. ACO ile DNN hibrit çalışmasının parametreleri

		DNN Kullanılan En İyi Değerler				
		Aktivasyon	Öğrenme Sırası	Tur	Katman-Nöron	Optimizasyon
ACO	1	sigmoid	128	150	[30, 20, 15, 10]	adam
	2	relu	128	200	[120, 80, 60, 45, 30]	adam
	3	sigmoid	128	100	[80, 60, 45, 30, 15]	adam
	4	sigmoid	64	200	[30, 20, 15, 10]	rmsprop
	5	sigmoid	32	200	[30, 20, 15, 10]	nadam
	6	relu	32	200	[80, 60, 45, 30, 15]	adam
	7	sigmoid	64	200	[30, 20, 15, 10]	adam
	8	relu	32	100	[80, 60, 45, 30, 15]	adam
	9	relu	64	200	[80, 60, 45, 30, 15]	adam
	10	relu	32	150	[120, 80, 60, 45, 30]	adam

DNN diğer sınıflandırma algoritmalarından farklı olarak çoklu parametre kullanılmıştır. En uygun parametreler seçilerek çalışmalar gerçekleştirilmiştir. Şekil 4.10'da sunulan sonuçlarda, GA üzerinde uygulanan mutasyon oranının artmasıyla metriklerde bir artış olduğu gözlemlenmiştir. En başarılı sonuçlar GA %20 mutasyon oranıyla elde edilmiştir. Bu durumda, %95 doğruluk değeri ve %94,05 f1-skoru elde edilmiştir. ACO yöntemi ise doğruluk ve f1-skor metriklerinde genetik algoritmaya göre daha ortalama sonuçlar vermiştir.



Şekil 4.10. Derin sinir ağları başarım değerlendirme

4.2.9. Özellik seçimi yapılmadan sınıflandırma çalışması

Tez kapsamında, özellik seçim çalışmalarının sonucunda elde edilen sınıflandırma başarı değerlerini karşılaştırabilmek amacıyla tüm özelliklerin kullanıldığı bir sınıflandırma

yürütülmüştür. Bu adım, özellik seçimi sonucunda belirlenen özelliklerin performansını değerlendirebilmek ve seçim sürecinin etkisini anlayabilmek için önemlidir. Bütün özellikleri içeren veri seti, sınıflandırma algoritmalarına herhangi bir işlem uygulanmadan adapte edilmiştir.

Tez çalışması sonucunda ulaşılan başarım metrikleri Çizelge 4.13 ile gösterilmektedir. Doğruluk, kesinlik ve f1-skor metriklerine göre en başarılı sonuçların RF algoritmasıyla elde edildiği gözlemlenmektedir. Duyarlılık metriğine göre ise en başarılı sonucun DNN algoritması tarafından verildiği görülmektedir. Genel olarak, RF algoritmasının iyi sonuçlar elde edildiği gözlemlenmektedir.

Çizelge 4.13. Özellik seçimi olmadan algoritmaların başarım değerleri

	Doğruluk	Kesinlik	Duyarlılık	F1-Skor
LR	91,75	91,75	88,46	90,08
RF	95,55	97,55	91,8	94,58
NB	88,05	90,70	80	85,01
KNN	78,83	78,10	69,49	73,54
Adaboost	93,54	92,54	92,30	92,42
LSVM	84,25	78,16	87,18	82,42
DT	91,20	92,56	86,15	89,24
DNN	93,81	92,36	93,07	92,72

Özellik seçimi yapılmadan sınıflandırıcı algoritmalarıyla gerçekleştirilen çalışmada, veri setindeki tüm özellikler kullanılarak sınıflandırma işlemi gerçekleştirilmiştir. Bu yöntemle elde edilen sonuçlara göre, her bir algoritmanın başarım değerleri belirlenmiştir. Algoritmaların doğruluk, kesinlik, duyarlılık ve f1-skor metrikleri üzerindeki performansları incelenmiştir. Elde edilen sonuçlar, her algoritmanın başarımının farklı olduğunu göstermektedir. Özellik seçimi yapılmadan sınıflandırıcı algoritmalarının başarım değerleri, veri setinin içerdiği tüm özelliklerin dikkate alındığı durumu yansıtmaktadır. Bu şekilde elde edilen sonuçlar, algoritmaların genel performansını değerlendirmek için önemli bir referans noktası sağlamaktadır.

4.3. Başarım Durumlarının Değerlendirilmesi

Özellik seçimi çalışması sonrasında gerçekleştirilen sınıflandırma algoritmalarının performansını değerlendirmek için çeşitli metrikler kullanılmıştır. Bu metrikler, modelinin

doğruluğunu, kesinliğini, duyarlılığını ve f1-skorunu ölçmektedir. Bu analizlerin sonuçları, özellik seçiminin sınıflandırma modellerinin performansını artırdığını ve daha iyi sonuçlar elde edildiğini göstermektedir. Bu sayede, daha etkili ve güvenilir sınıflandırma modelleri oluşturulabilir ve uygulama alanında daha iyi kararlar alınabilir.

Doğruluk metriği, modelin doğru sınıflandırma oranını gösterirken, kesinlik metriği, pozitif olarak sınıflandırılan örneklerin gerçekten pozitif olma oranını ölçer. Duyarlılık metriği ise gerçek pozitifleri ne kadar doğru bir şekilde tespit edilebildiğini gösterir. F1-skor, kesinlik ve duyarlılık metriklerinin harmonik ortalamasıdır ve modelin dengeli bir performans sergileyip sergilemediğini gösterir.

Bir sınıflandırıcı uygulandıktan sonra özellik seçme yöntemlerinin davranışını değerlendirebilmek için çeşitli değerlendirme metrikleri tanımlanması gerekmektedir. Bu çalışma kapsamında kullanılan metrikler ve açıklamaları Çizelge 4.14'te sunulmaktadır.

Çizelge 4.14. Performans değerlendirmesi için kullanılan metrikler [6]

Metrik	Metrik Özellikleri		
	Açılım	Tanım	Eşitlik
TP	True positive-Doğru pozitif	Doğru bir şekilde pozitif olarak sınıflandırılan pozitif örneklerinin sayısı	-
TN	True negative-Doğru negatif	Doğru bir şekilde negatif olarak sınıflandırılan negatif örneklerin sayısı	-
FP	False positive-Yanlış pozitif	Yanlış bir şekilde pozitif olarak sınıflandırılan negatif örneklerin sayısı	-
FN	False negative-Yanlış negatif	Yanlış bir şekilde negatif olarak sınıflandırılan pozitif örneklerin sayısı	-
ACC	Accuracy-Doğruluk	Toplam sonucun içerisinde doğru tahminlerin oranıdır.	$\frac{TP+TN}{TP+TN+FP+FN}$
PRC	Precision-Kesinlik	Pozitif olarak tahmin edilen değerlerden gerçekte kaç adedinin pozitif olduğunun oranıdır.	$\frac{TP}{TP+FP}$
REC	Recall-Duyarlılık	Pozitif olarak tahmin edilmesi gereken değerlerin ne kadarının pozitif olarak tahmin edildiğinin oranıdır.	$\frac{TP}{TP+FN}$
F1	F1-Score-F1-Skor	Kesinlik ve duyarlılık değerlerinin harmonik ortalamasıdır.	$\frac{2*TP}{2*TP+FN+FP}$

Tüm özellik seçim yöntemlerinin ardından gerçekleştirilen sınıflandırma çalışmalarının performansı, bu yöntemlerle değerlendirilmektedir. Genetik algoritma (GA) ve karınca koloni optimizasyonu (ACO) algoritmalarında birden fazla sınıflandırma çalışması yapıldığından elde edilen sonuçlar en iyi, en kötü, medyan, ortalama ve standart sapma gibi değerlendirme fonksiyonlarına tabi tutulur. Bu değerlendirmeler ve istatistiksel ölçümler,

performansın değerlendirilmesine ve sonuçların grafiksel olarak gösterilmesine olanak sağlar. Kullanılan metotların tanımlaması Çizelge 4.15'te sunulmaktadır. Bu şekilde, farklı yöntemlerin performansı karşılaştırılır ve hangi yöntemlerin daha iyi sonuçlar verdiği analiz edilir.

Çizelge 4.15. Performans değerlendirmesi fonksiyonları [6]

Ölçüt	Metot Özellikleri		
	Anlam	Tanım	Denklem
Best	En iyi değer	Elde edilen sonuçlar arasında en iyi sonucu verir.	-
Worst	En kötü değer	Elde edilen sonuçlar arasında en kötü sonucu verir.	-
Average	Ortalama değer	Elde edilen değerlerin ortalamasını alır.	$x_1 + x_2 + \cdots + x_N/N$
Median	Medyan	N adet değerin ortasındaki değeri verir.	$p = \mathrm{N}/2\,,\ \ (x_p + x_{p+1})/2$
ST	Standart sapma	Verilerin ortalamadan ne kadar farklı olduğunun bir ölçüsüdür.	$\sqrt{\sum_{i=1}^{N}(x_i - \mu)^2/N}$

5. BULGULAR

Bu tez çalışmasının temel amacı, özellik seçimi yöntemleriyle elde edilen özellikleri veri setine uyarlayarak sınıflandırma çalışmalarının gerçekleştirilmesidir. Özellik seçimi yöntemlerinin sınıflandırma performansı üzerindeki etkisi incelenmektedir.

Öncelikle, farklı özellik seçimi yöntemleri (filtreleme ve sarmal) kullanılarak veri setindeki özellikler belirlenir. Daha sonra, bu seçilen özellikler veri setinde düzenleme yapılarak sınıflandırma modelleri oluşturulur ve performans değerlendirmesi yapılır.

Elde edilen sonuçlar, özellik seçimi yöntemlerinin sınıflandırma performansını önemli ölçüde artırdığını göstermektedir. Sınıflandırma metrikleri arasında doğruluk, kesinlik, duyarlılık ve f1-skor ölçütler kullanılarak yapılan değerlendirme, özellik seçimi yöntemlerinin daha iyi sonuçlar elde ettiğini göstermektedir.

Bu çalışmanın bulguları, özellik seçimi yöntemlerinin sınıflandırma performansını artırdığı ve veri setinin en uygun özellikleri kullanarak daha doğru ve güvenilir sonuçlar elde edilebileceğini ortaya koymaktadır. Ayrıca, farklı özellik seçimi yöntemleri arasında karşılaştırmalar yapılarak, en etkili yöntemlerin belirlenmesi ve uygulama alanlarına yönelik önerilerde bulunulması da mümkün olmaktadır.

Sarmal özellik seçim yöntemi olarak kullanılan GA-RFR ve ACO-RFC hibrit çalışmalar ayrı ayrı incelenmekte ve bu çalışma kapsamında GA da mutasyon oranının sınıflandırmaya etkisi tartışılmaktadır.

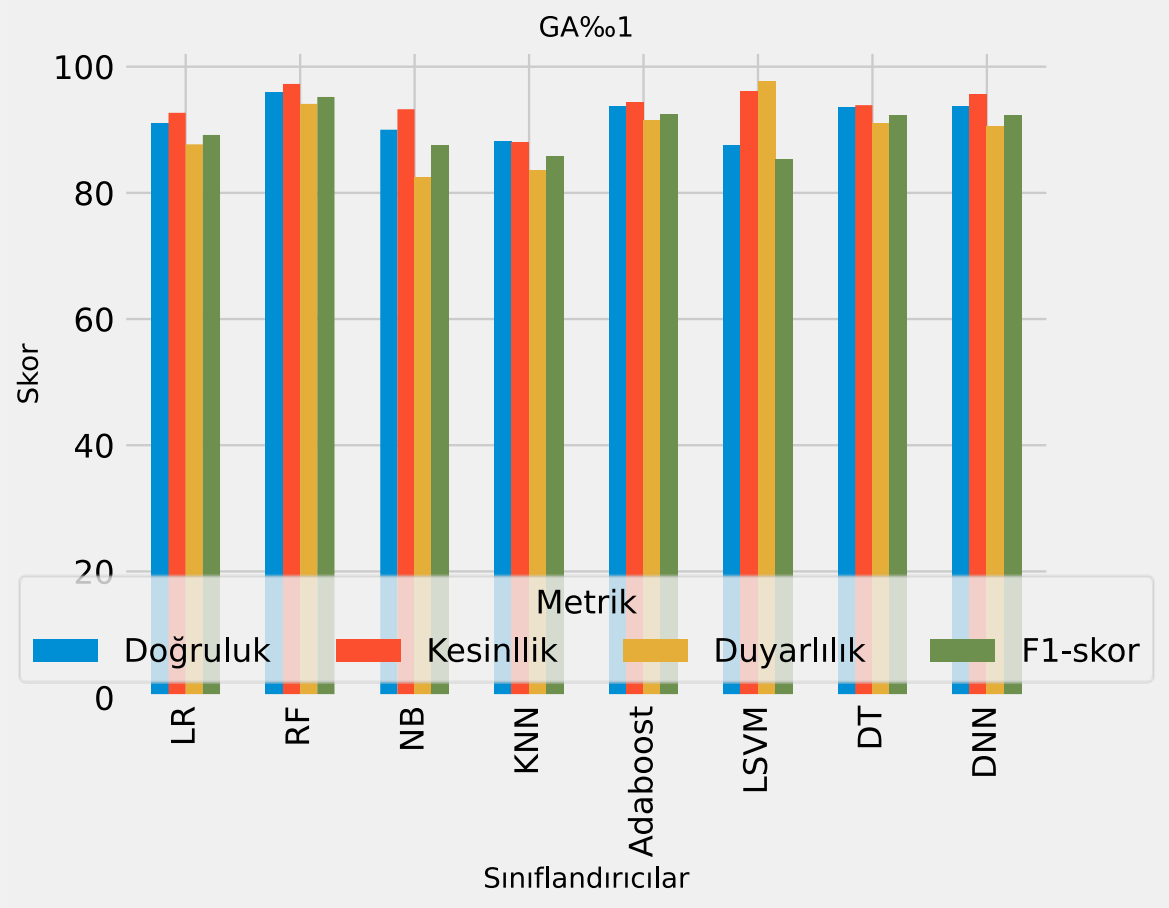
5.1. Genetik Algoritma Farklı Mutasyon Değerlerinin Sonuçları

Farklı mutasyon oranlarına sahip GA sınıflandırma algoritmalarının performansı, dört ayrı başarı metriği kullanılarak analiz edilmektedir. Bu sonuçlar Şekil 5.1-5.3'te görselleştirilmiştir.

Bu grafikler, mutasyon oranlarının sınıflandırma algoritmalarının performansı üzerindeki etkisini göstermektedir. Mutasyon oranının artmasıyla birlikte bazı metriklerde iyileşme görülebilirken, diğer metriklerde belirli bir eşiği geçtikten sonra performans düşüşü de

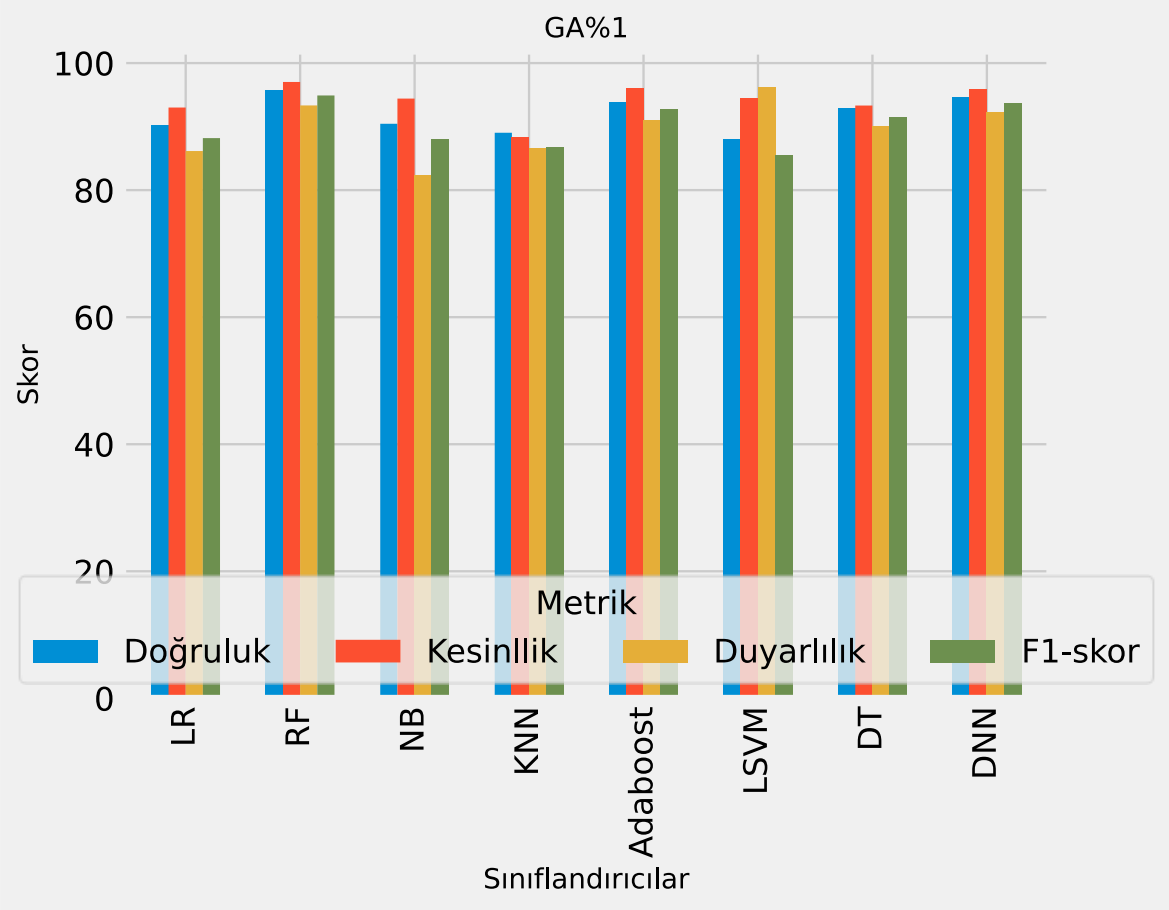
gözlenebilir. Bu bulgular, mutasyon oranının GA' nın sınıflandırma performansını etkileyen önemli bir faktör olduğunu göstermektedir.

Şekil 5.1, GA' nın ‰1 mutasyon oranıyla elde edilen başarı sonuçlarını göstermektedir. Bu grafikte, en yüksek performansı RF algoritmasının gösterdiği görülmektedir. LSVM algoritmasında ise mutasyon oranlarına bağlı olarak farklı performans değerleri elde edilmiştir, en yüksek performans duyarlılık metriğiyle elde edilmiştir.



Şekil 5.1. Genetik algoritma ‰1 mutasyon sonuçları

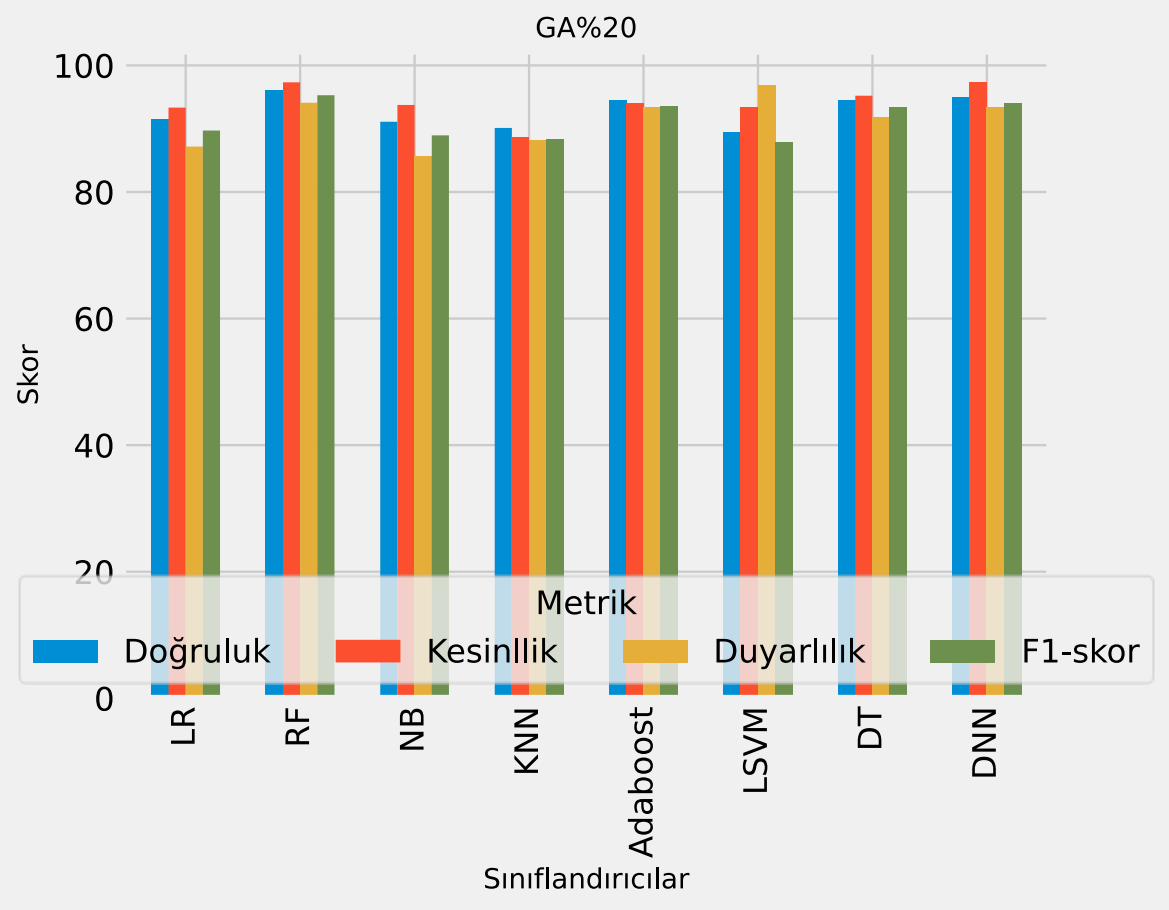
Şekil 5.2'de ise DNN ve RF algoritmaları arasında benzer performans sonuçları elde edildiği gözlemlenmektedir. Ayrıca, DNN' den sonra Adaboost algoritmasının da başarılı sonuçlar verdiği grafikte görülmektedir. LSVM algoritmasının ise duyarlılık metriğinde yüksek performans sergilediği gözlemlenmektedir.



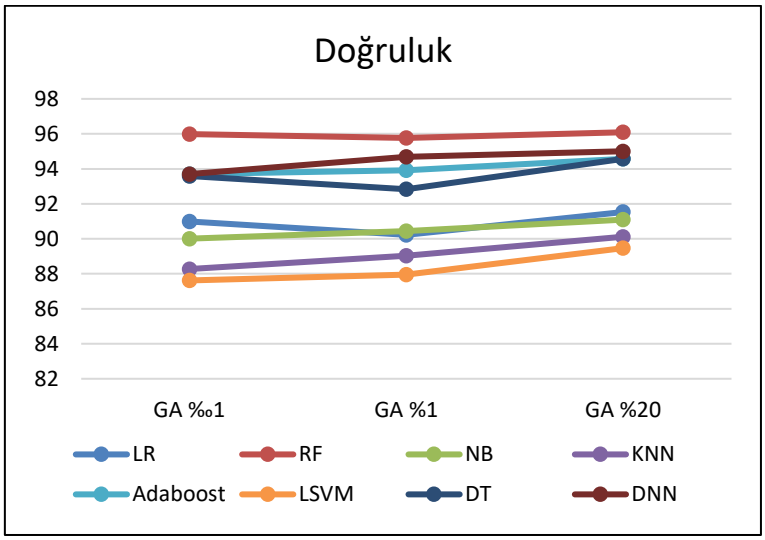
Şekil 5.2. Genetik algoritma %1 mutasyon sonuçları

Şekil 5.3'te diğer mutasyon oranlarına ait verilerle benzer dalgalanmalar gözlenmektedir. Sayısal olarak, GA %20 mutasyon oranının daha iyi sonuçlar verdiği tespit edilmiştir. DNN ve RF algoritmaları diğer algoritmalarda olduğu gibi başarılı performans göstermiştir. Tüm mutasyon sonuçlarına bakıldığında, KNN algoritmasının ortalama olarak en düşük performansı sergilediği gözlenmektedir. Fakat diğer sonuçlara göre GA %20 sonucunda yükselmeler gözlemlenmiştir.

Metrikler üzerinde mutasyon oranına göre algoritmaların oransal farklılıklarının daha iyi incelenebilmesi amacıyla ayrı grafikler oluşturulmuştur. Genetik algoritmanın mutasyon değerlerinin başarım metriklerinde elde edilen sonuçları Şekil 5.4-5.7 ile gösterilmektedir.



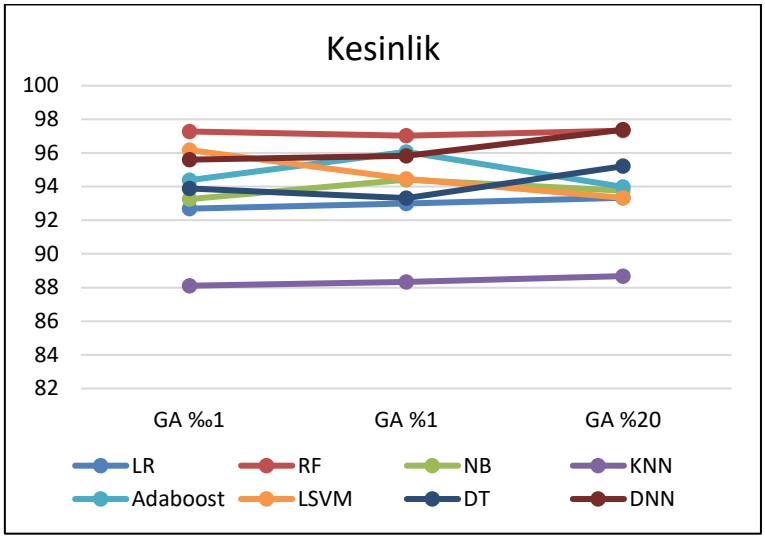
Şekil 5.3. Genetik algoritma %20 mutasyon sonuçları



Şekil 5.4. GA doğruluk değerleri

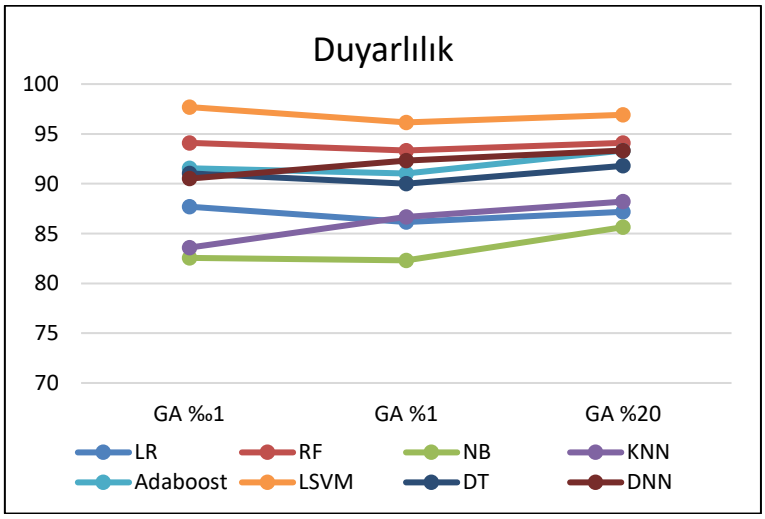
GA üzerinde uygulanan mutasyon oranlarının sınıflandırma çalışmalarına etkisi doğruluk metriği açısından Şekil 5.4'te incelenmiştir. Görselde, mutasyon oranının artmasıyla doğruluk değerlerinin iyileştiği görülmektedir. En başarılı sonuçlar RF algoritmasına aittir.

GA‰1 de sonuçlar düşükken mutasyon oranı arttırılınca ufak dalgalanmaların sonucunda başarım değerlerinin arttığı gözlemlenmektedir.



Şekil 5.5. GA kesinlik değerleri

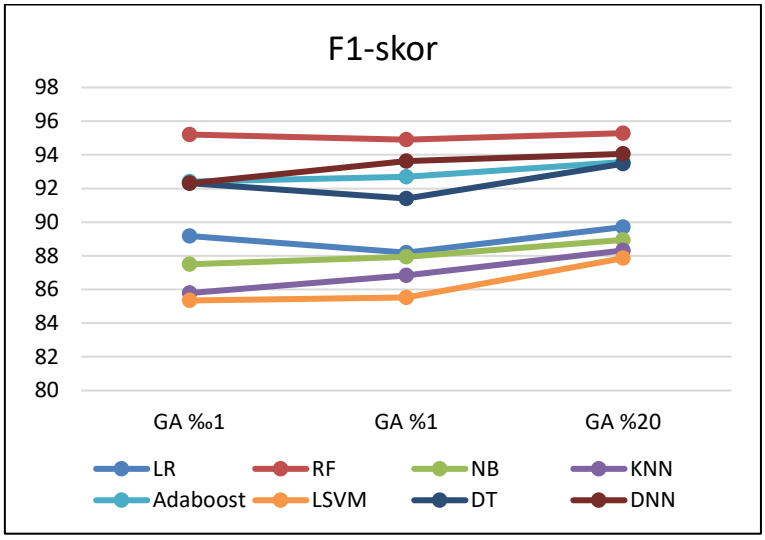
Şekil 5.5'te, doğruluk metriğinden daha farklı sonuçlarla karşılaşılmıştır. Kesinlik metriği üzerinde yapılan değerlendirmede en başarılı algoritmanın RF olduğu görülmektedir. %20 mutasyon oranında DNN algoritmasının en iyi performansı gösterdiği gözlemlenmiştir. KNN algoritmasının ise en düşük performansa sahip olduğu sonucuna varılmıştır. Adaboost algoritmasında ise büyük dalgalanmalar gözlenmiştir.



Şekil 5.6. GA duyarlılık metriği

Şekil 5.6 üzerinde yapılan değerlendirmede, diğer algoritmalardan farklı olarak en başarılı

algoritmanın LSVM olduğu gözlemlenmektedir. En iyi sonuç, %1 mutasyon oranıyla elde edilmiştir. RF ve DNN algoritmalarının da başarılı performans gösterdiği görülmektedir. NB algoritmasının ise en düşük duyarlılık sonucunu verdiği gözlemlenmiştir. Duyarlılık metriğinde mutasyon oranı arttırımıyla belirli bir düşüşün olduğu gözlemlenmektedir.

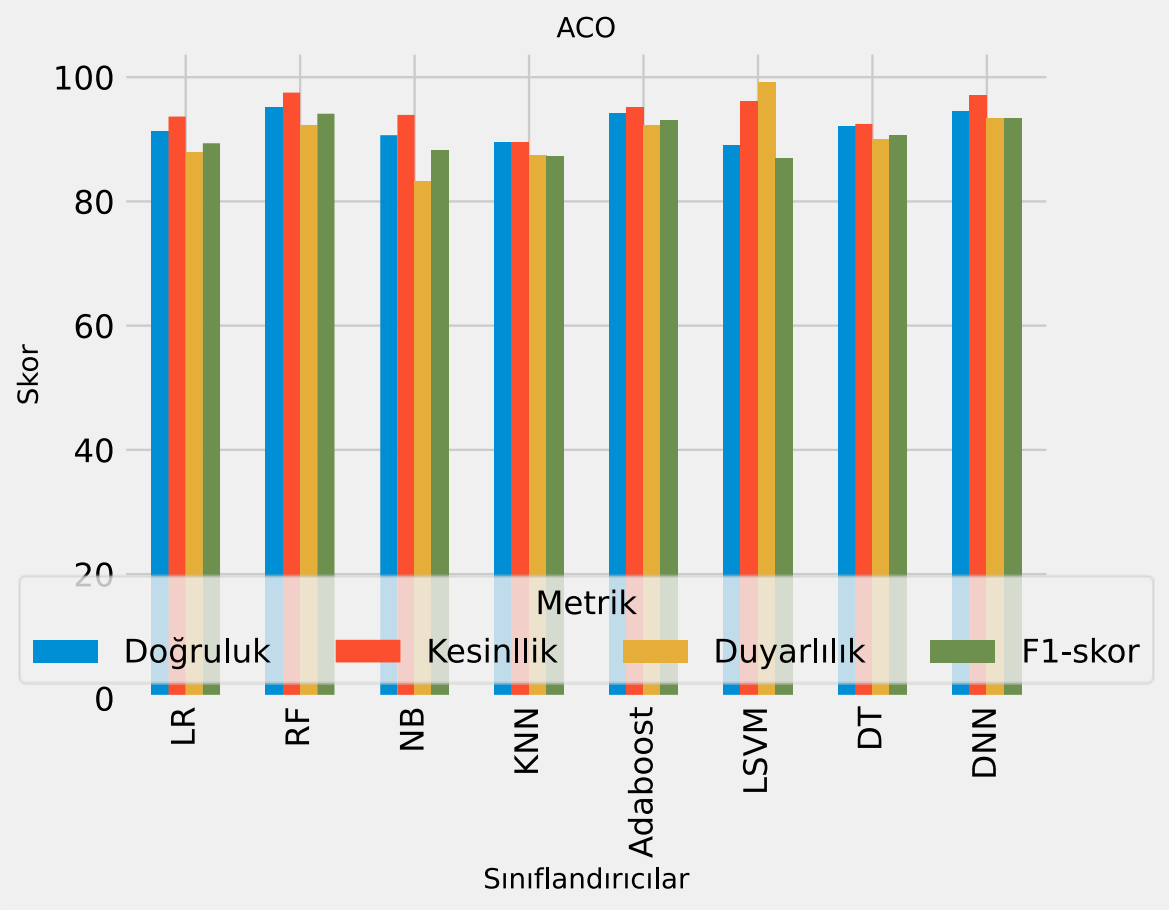


Şekil 5.7. GA f1-skor metriği

Şekil 5.7'de f1-skor metriğine göre elde edilen sonuçlar gösterilmektedir. Bu değerlendirmede, RF ve DNN algoritmalarının başarılı sonuçlar verdiği görülmektedir. En düşük performans ise LSVM algoritmasıyla elde edilmiştir. Mutasyon oranının artmasıyla birlikte sonuçların iyileştiği gözlemlenmektedir.

5.2. Karınca Koloni Optimizasyonu ile Alınan Sonuçlar

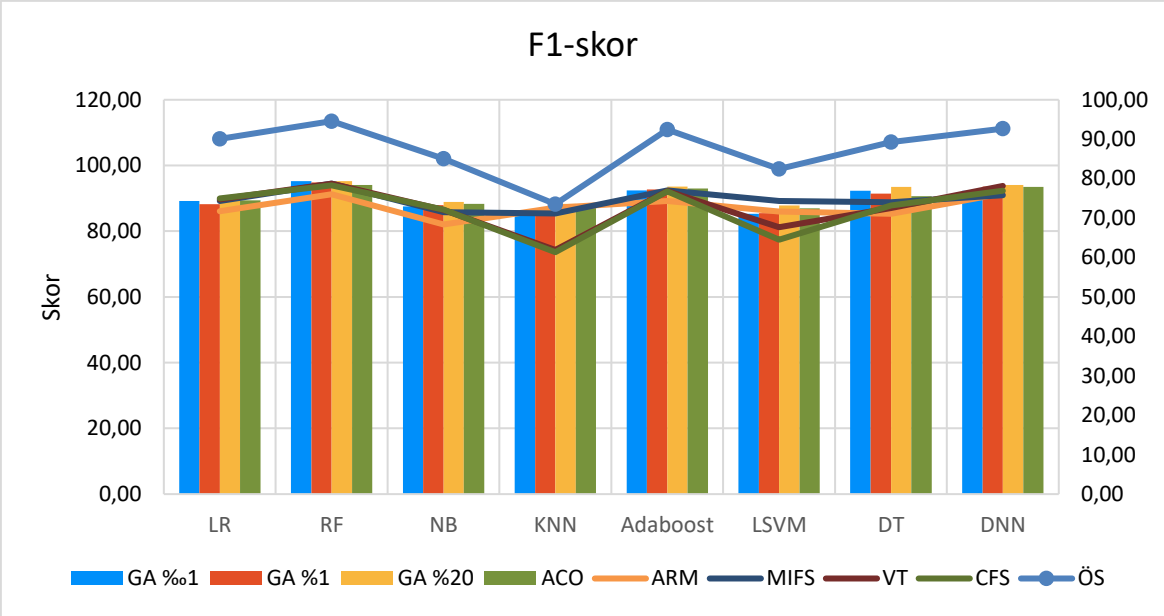
Şekil 5.8'de, karınca koloni optimizasyonu yönteminin sınıflandırma algoritmaları sonucunda elde edilen metrik değerlerini gösteren sonuçlar verilmiştir. Bu sonuçlara göre, doğrusal destek vektör makinesi algoritması duyarlılık metriğinde en yüksek performansı göstermiştir. Kesinlik metriğinde ise RF ve DNN algoritmalarında benzer sonuçlar elde etmiştir. Doğruluk ve f1-skor metriklerinde ise diğerlerine kıyasla daha iyi sonuçlar RF algoritmasıyla elde edilmiştir. Adaboost ve DNN algoritmalarının sonuçları ise RF algoritmasına yakın performans göstermektedir.



Şekil 5.8. Karınca koloni optimizasyonu sonuçları

5.3. Çalışma İçerisindeki Tüm Uygulamaların Sonuçları

Filtreleme yöntemleriyle sarmal yöntemlerin birleştirildiği özellik seçimi çalışmasında, elde edilen sonuçlar f1-skor metriği üzerinden sunulmaktadır. Bu metrik, sınıflandırma performansının bir ölçüsü olarak kullanılmaktadır. Filtreleme yöntemleri, veri setindeki özellikleri değerlendirerek en önemlilerini seçmektedir. Sarmal yöntemler, kendi içerisinde kullanılan seçim yöntemleri aracılığıyla özellikleri seçerler. Bu algoritmalar, veri setinin analizini yaparak en önemli özellikleri belirler ve bu özellikleri kullanarak sınıflandırma çalışmaları gerçekleştirilir. Şekil 5.9 üzerinde hem filtreleme yöntemlerinin etkinliği hem de sarmal yöntemlerin performansı f1-skor metriği üzerinden analiz edilmektedir.



Şekil 5.9. Tüm algoritmaların f1-skor sonuçları

Çalışmada farklı özellik seçimi yöntemleri kullanılmış ve elde edilen sonuçlar çeşitlilik göstermiştir. Sarmal yöntemler ile elde edilen özellik seçimi performansı sütun grafikleriyle gösterilirken, diğer filtreleme yöntemleri çizgi grafikleriyle gösterilmiştir. Ayrıca, özellik seçimi yapılmadan gerçekleştirilen sınıflandırma çalışmaları işaretli çizgiyle belirtilmiştir. F1-skor sonuçları Çizelge 5.1'de tablo şeklinde sunulmuştur ve başarılı sonuçlar koyu renkle vurgulanmıştır.

Çizelge 5.1. Tüm algoritmaların f1-skor sonuç değerleri

	ÖSY	GA‰1	GA%1	GA%20	ACO	ARM	MIFS	VT	CFS
LR	90,08	89,18	88,19	89,71	89,36	86,10	89,19	89,72	90,03
RF	94,58	95,20	94,90	95,29	94,12	91,22	94,49	94,47	93,96
NB	85,01	87,50	87,95	88,95	88,32	82,06	85,79	86,48	86,57
KNN	73,54	85,79	86,83	88,32	87,25	87,18	85,41	74,29	73,63
Adaboost	92,43	92,41	92,69	93,57	93,02	89,20	92,45	92,23	92,15
LSVM	82,42	85,35	85,53	87,86	86,93	86,02	89,19	81,22	77,44
DT	89,24	92,31	91,41	93,47	90,70	85,31	88,86	87,08	87,80
DNN	92,72	92,31	93,63	94,06	93,45	91,49	90,93	93,75	92,35

Özellik seçimi çalışmasının karşılaştırması için özellik seçimi yapılmadan önce ve sonra sınıflandırma çalışmaları gerçekleştirilmiştir. Bu şekilde özellik seçiminin performansa olan etkisi incelenmiştir. Elde edilen sonuçlar, özellik seçimi yapıldığında genel olarak daha iyi performans elde edildiğini göstermektedir. Ancak, LR algoritması özellik seçimi yapılmadan daha iyi sonuç vermiştir. Ayrıca, KNN algoritması en düşük performansa sahip

olan algoritma olarak gözlemlenmiştir. Bu durum, f1-skor metriği ile elde edilen sonuçlar içinde geçerlidir.

Genetik algoritma kullanılarak gerçekleştirilen özellik seçimi sonucunda, mutasyon oranı arttıkça sınıflandırma algoritmalarında f1-skor değerinde bir artış gözlemlenmiştir. GA' da ‰1 mutasyon uygulandığında en yüksek başarı %95,2 olarak RF algoritmasıyla elde edilmiştir. RF ayrıca, GA'nın %1 mutasyonuyla %94,9 ve %20 mutasyonuyla %95,28 başarı değerleriyle yüksek performans sergilemiştir. %20 mutasyon sonucunda en başarılı ikinci algoritma DNN olarak belirlenmiştir.

Karınca koloni optimizasyonu kullanıldığında, genetik algoritmaya benzer sonuçlar elde edilmiştir. En başarılı sonuç, RF algoritmasından sonra DNN algoritmasında elde edilmiştir. ACO incelendiğinde, GA'nın çalışmalarının ortalama değerlerine yakın sonuçlar elde edildiği görülmektedir. ACO ile DT sınıflandırıcısı arasında ortalama değerler karşılaştırıldığında, ACO'nun DT için daha düşük bir sonuç verdiği gözlemlenmiştir. Genetik algoritmaların DT sınıflandırıcısı için ortalama %92,39 değerini verdiği, ACO'nun ise %90,69 değerini verdiği görülmüştür.

Filtreleme yöntemlerinde dalgalanmalar olduğu gözlemlenmektedir. ARM yöntemiyle DNN sınıflandırıcısında yüksek bir başarı elde edilmiştir. Diğer filtreleme yöntemlerinde ise en yüksek başarı oranı RF algoritması tarafından gösterilmiştir. MIFS yöntemiyle elde edilen özelliklerle LSVM algoritmasının daha başarılı sonuçlar verdiği gözlemlenmektedir. %89,19 değeriyle diğer seçim yöntemlerinden daha iyi bir sonuç elde edilmiştir.

Genel olarak, özellik seçimi çalışmalarının sınıflandırma performansını iyileştirdiği gözlemlenmiştir. Sarmal ve filtreleme yöntemleri, sınıflandırma algoritmalarıyla birlikte kullanıldığında daha iyi sonuçlar elde edilmesini sağlamıştır. Bu çalışmanın sonuçları, özellik seçimi yöntemlerinin uygunluğunu ve uygulama alanlarına yönelik önerilerin belirlenmesini destekleyen önemli bulgular sunmaktadır.

SONUÇ VE ÖNERİLER

Elektronik postaların çok amaçlı olarak kullanımı, internet kullanımının artmasıyla birlikte artış göstermiştir. E-postalar, iletişim aracı olarak kullanıldığı gibi bilgi paylaşımı, dosya aktarımı, depolama ve güvenlik düzeyini artırma gibi farklı işlevleri de yerine getirmektedir. İnsanlar, e-posta aracılığıyla hızlı ve kolay bir şekilde iletişim kurabilirken aynı zamanda belgeleri paylaşabilir, dosyaları transfer edebilir ve önemli verileri güvenli bir şekilde saklayabilir. E-postaların bu çok yönlü kullanımı, kişilerin iş ve günlük yaşamlarını kolaylaştırırken iletişim ve bilgi akışını da hızlandırmaktadır.

E-postaların faydalı kullanımlarının yanında, kötü niyetli kişiler tarafından da zararlı amaçlara yönelik kullanımı mümkündür. Bu kişiler, zararlı yazılımları e-postalar aracılığıyla yayabilir ve kullanıcıları kötü niyetli web sitelerine yönlendirebilir. Özellikle gizlilik ve güvenliğin önemli olduğu teknoloji çağında, gelen ve giden elektronik postaların içerikleri büyük önem taşımaktadır. Kötü niyetli kişiler, e-postaları manipüle ederek kullanıcıların kişisel bilgilerini ele geçirebilir veya hassas verilere erişebilir. Bu nedenle, e-postaların içeriklerinin güvenliği ve gizliliği, kullanıcılar ve şirketler için büyük bir endişe haline gelmiştir. İnsanlar ve kurumlar, e-postaların kötü amaçlı kullanımını engellemek ve kişisel bilgilerini korumak için çeşitli önlemler almaktadır.

Ayrıca, kötü niyetli kişiler veya kuruluşlar çoklu posta göndererek kişilere veya kurumlara zaman kaybı yaşatabilirler. Bu tür saldırılar, bir ağdaki bant genişliğini daraltarak, internet hızını olumsuz etkileyebilir. Bu tür olumsuz durumları engellemek amacıyla, gelen e-postaların spam (istenmeyen) veya ham (istenen) olarak sınıflandırılması büyük önem taşımaktadır. Spam filtreleme sistemleri, kötü amaçlı veya gereksiz içeriklere sahip e-postaların tespit edilerek kullanıcının önemli zaman ve kaynaklarını korumasına yardımcı olmaktadır. Bu şekilde, kullanıcılar istenmeyen e-postalardan korunabilir ve iletişimlerini daha verimli bir şekilde yönetebilirler.

E-posta, SMS veya oltalama verileri üzerinde sınıflandırma çalışmaları yapılmış ve halen yapılmaya devam etmektedir. Spam sınıflandırmasının başarılı olabilmesi için çeşitli yöntemler denenmiştir. Bu yöntemlerden biri, veri setleri üzerinde ön işlem adımlarının gerçekleştirilmesiyle sınıflandırma çalışmasının yapılmasıdır. Ön işlem adımları arasında

veri temizleme, düşük öneme sahip veri özelliklerinin filtrelenmesi ve veri ölçeklendirme gibi adımlar bulunmaktadır. Bir diğer yöntem ise ham verilerden özellik çıkarımı yapılarak örnek çalışmaların gerçekleştirilmesidir. Tüm bu yöntemler, spam sınıflandırmasının daha etkili ve doğru sonuçlar vermesine yardımcı olmaktadır.

Özellik çıkarımı işlemi sırasında karşılaşılan zorluklardan biri çok boyutluluk problemidir. Bu durumda, veri setindeki özelliklerin sayısı çok fazla olabilir ve bu da analiz ve sınıflandırma sürecini karmaşık hale getirebilir. Bu nedenle, özellik çıkarımı yapılmış veri setleri üzerinde özellik seçimi çalışmaları önem kazanmıştır. Bu çalışmalarda, sınıflandırma başarısını maksimize etmek için az veri ile çok işlem yapılabilmesi amaçlanmıştır.

Özellik seçimi ve az veriyle çalışma yaklaşımları, sınıflandırma çalışmalarında verimlilik, hız ve etkinlik sağlamak amacıyla kullanılmaktadır. Bu yöntemler, çok boyutlu veri setlerinin karmaşıklığını azaltmak, gereksiz özellikleri elemek ve daha iyi sonuçlar elde etmek için değerli bir araçtır.

Birçok araştırmada, çeşitli özellik seçim yöntemlerinin olduğu gözlemlenmiştir. Bu yöntemlerin performansı, kullanılan algoritmalara ve çözülmek istenen probleme bağlı olarak değişmektedir. Dolayısıyla, "en iyi yöntem" şeklinde kesin bir ifade kullanmak mümkün değildir. Her özellik seçim yöntemi farklı bir yaklaşım ve metrikler kullanabilir ve farklı veri setleri üzerinde farklı sonuçlar verebilir. Bu nedenle, bir yöntemin başarısı diğer bir yönteme göre daha iyi olabilirken, başka bir veri seti veya farklı bir problem üzerinde aynı başarıyı göstermeyebilir.

Özellik seçiminde en etkili yöntemi belirlemek için genellikle dikkate alınması gereken faktörler algoritmanın hesaplama karmaşıklığı, özelliklerin ilişkisi, veri setinin boyutu ve dağılımı, hedeflenen metrikler ve problemin doğası gibi çeşitli faktörlerdir. Bu nedenle, en uygun yöntemi seçmek için dikkatli bir analiz yapılması ve deneylerin yapılması önemlidir.

Temel filtreleme yöntemlerinin yanında karmaşık algoritmalarla kullanılan veri seti eğitilerek özellik seçim işlemleri gerçekleştirilir. Filtreleme yöntemleri algoritmalardan bağımsız olarak gerçekleştirilir. Bu özellik, filtreleri hesaplama açısından basit ve hızlı hale getirir ve son derece büyük ölçekli veri kümelerini işleyebilir. Bununla birlikte, çoğu filtreleme tek değişkenlidir, yani her bir özellik diğer özelliklerden bağımsız olarak ele

alınır; bu, genellikle daha fazla hesaplama kaynağı gerektiren bir dezavantajdır.
Kullanılan sarmal özellik seçim yöntemlerinde kullanılan algoritmalarda özellikler birbiriyle ilişkili olarak seçim gerçekleştirilir. Özellik seçiminde karşılaştırma çalışması olarak yapılan tez önerisinde, metasezgisel algoritmaların sarmal olarak çalıştırılması klasik filtreleme yöntemlerine göre daha başarılı sonuçlar vermiştir.

Tez çalışmasında örnek veri seti olarak kullanılan Spambase veri seti üstünde çalışma gerçekleştirilmiştir. Özellik seçimi üzerinde durulduğundan dolayı tek veri seti kullanılmıştır. Klasik filtreleme yöntemleri kullanılarak veri seti üzerinde ön işlem adımları yapılmadan belirli eşik değerine göre özellikler seçilmiştir. Her alanda kullanılan metasezgisel algoritmalar bu çalışma kapsamında sarmal yöntemler içerisinde özellik seçiminde kullanılmıştır. Sıklıkla karşılaşılan genetik algoritma ve karınca koloni optimizasyonu karşılaştırılması yapılmıştır. Genetik algoritmada kullanılan mutasyon değerleri değiştirilerek kendi içerisinde mutasyon oranının sınıflandırmaya etkisi tartışılmıştır. Klasik filtreleme yöntemleri ile kıyaslandığında metasezgisel algoritmaların başarım durumlarının daha iyi olduğu gözlemlenmiştir.

Sınıflandırma algoritmalarında kullanılan parametrelerin çoklu parametre olarak kullanılması çalışma amacına yönelik olmamaktadır. İncelenen özellik seçim yöntemlerinin başarım durumlarının karşılaştırılabilmesi maksadıyla kütüphanelerde varsayılan değeler kullanılmıştır. Derin sinir ağları algoritmasında çoklu parametrelerin belirlenmesi her özelik seçimi yönteminde ayrı olarak hesaplanmıştır. Katman değeri olarak belirli bir sayı olmadığı için çoklu parametre hesaplama yöntemine gidilmiştir.

Elde edilen sonuçların tamamı “EKLER” kısmanda EK-3 ve Ek-4 ile verilmiştir. Sarmal yöntemlerde en başarılı sonuç LSVM algoritmasında elde edilmiştir. Vermiş olduğu %99,23 kesinlik değeri ile çalışmanın başarılı olduğunu göstermektedir. Diğer algoritmalarda da sonuçlar %90 üzerindedir. Filtreleme yöntemlerinde RF ve DNN algoritmasının sonuçları yüksek orandadır. Fakat sarmal yöntemler daha yüksek başarım göstermiştir.

Tez önerisinin literatür incelemesi sonucunda daha önce yapılan çoğu çalışmadan iyi performans gösterdiği “*Literatür Taraması*” başlığı altında tablolar halinde sunulmuştur. Özellik seçimi yapılmadan sınıflandırma algoritmalarında ham veri seti uygulanmıştır. Sonuçta özellik seçiminin önemli olduğu gözlemlenmiştir. Birliktelik kuralı madenciliği

yöntemi ile elde edilen sonuçlarda RF, NB KNN DT ve DNN algoritmalarının [55] çalışmasından iyi olduğu gözlemlenmektedir. Korelasyon özellik seçimi yönteminde [54-55, 152] çalışmalarından daha iyi sonuç alındığı görülmektedir. Genetik algoritma %20 mutasyon sonuçlarının diğer mutasyon oranlarından daha iyidir. LR, adaboost ve DNN algoritmalarında incelenen diğer çalışmalardan daha başarılı sonuçlar vermiştir. İncelemeler sonucunda e-posta sınıflandırma çalışmalarında karınca koloni optimizasyonun çok fazla kullanılmadığı gözlemlenmiştir. Mevcut çalışmaların başarım durumlarının da düşük olduğu çizelgelerde gösterilmiştir.

Literatür incelemesi sonucunda e-posta veri setinde yapılmış olan filtreleme yöntemleri ile özellik seçim çalışmalarının çok fazla olmadığı gözlemlenmiştir. Varyans eşiği yöntemi kullanılarak yapılmış bir çalışmaya erişilmemiştir. Literatürde olan çalışmalarda da bu kadar çeşitli sınıflandırma algoritmalarının bir arada olmadığı görülmüştür. Genetik algoritma ile özellik seçim çalışması mevcuttur. Buna ek olarak mutasyon oranı değiştirilerek genetik algoritmanın özellik seçiminde farklı bir kullanımı sağlanmıştır. Karınca koloni optimizasyonu e-posta sınıflandırma çalışmalarında farklı şekillerde kullanıldığı gözlemlenmiştir. Çeşitli sınıflandırıcılarla hibrit çalışması yapılarak literatüre katkısı sağlanmıştır. Önümüzdeki süreçte, tek bir veri seti kullanılarak yapılan çalışmada farklı veri setleri kullanılarak çalışma çeşitlendirilebilir ve özellik seçiminin sınıflandırmaya etkisi tartışılabilir. Ham veri setleri kullanılarak özellik çıkarımı yapılabilir. Özellik çıkarımı ile literatürden farklı özelliklere sahip veri setleri ile ilgili çalışmalar ele alınabilir. Böylece farklı özellik seçim yaklaşımları yeni modellere uygulanabilir ve çalışmalar genişletilebilir.

Elde edilen sonuçlar, özellik seçimi yönteminin sınıflandırma performansını artırdığını göstermektedir. Yani, daha az önemli veya gürültülü özelliklerin çıkarılmasıyla, modelin daha iyi sonuçlar elde ettiği gözlenmiştir. Özellik seçimi, sınıflandırma modelinin daha kesin ve güvenilir kararlar vermesini sağlamaktadır.

Tez çalışması sonucunda yapılabilecek öneriler arasında;

Veri kümesi genişletilebilir. Elektronik postaların farklı kategorilerde daha iyi sınıflandırılabilmesi için daha geniş ve çeşitli bir veri kümesi kullanılabilir. Farklı endüstrilerden, farklı kullanıcı gruplarından ve çeşitli konulardan örnekler içeren bir veri kümesi, daha iyi bir sınıflandırma performansı sağlayabilir.

Elektronik postaların sınıflandırılmasında kullanılacak özelliklerin seçimi ve çıkarılması önemlidir. Metin madenciliği ve doğal dil işleme tekniklerini kullanılarak, e-postalardan faydalı özellikler çıkarılabilir. Örneğin, kelime frekansı, kelime vektörleri, cümle yapısı, başlık bilgileri gibi özellikler kullanılabilir.

Elektronik postaların sınıflandırılmasında kullanılmış olan algoritmalara ek olarak farklı makine öğrenimi algoritmaları da kullanılabilir. Diğer algoritmalar ile e-postaları sınıflandırmak için eğitim ve test veri setleri oluşturulabilir. Kullanılmış olan algoritmalar arasında iyileştirmeler yapılarak performans değerleri arttırılabilir.

Makine öğrenimi algoritmalarında kullanılan çoklu parametrelerin doğru bir şekilde ayarlanması önemlidir. Grid arama veya rastgele arama gibi çoklu parametre ayarlama tekniklerini kullanarak, en iyi performansı sağlayacak kombinasyonlar bulunmaya çalışılabilir.

Geliştirilen sınıflandırma modelini doğrulama ve değerlendirme için ayrı bir veri setiyle test edilebilir. Doğruluk, duyarlılık, kesinlik gibi kullanılmış olan metrikler dışında farklı metriklerle modelin performansı değerlendirilebilir. Bu adımda, modelin hata analizi yapılarak iyileştirmeler yapılabilir.

Kullanılmış olan mutasyon oranının yanında çaprazlamada kullanılan oran da değiştirilerek çaprazlamanın çalışmaya etkisi tartışılabilir.

KAYNAKLAR

1. Cranor, L. F., ve Lamacchia, B. A. (1998). Spam! *Communications of the ACM*, 41(8), 74-83.

2. Zieni, R., Massari, L., ve Calzarossa, M. C. (2023). Phishing or not phishing? a survey on the detection of phishing websites. *IEEE Access*, 11, 18499-18519.

3. Sewak, M., Sahay, S. K., ve Rathore, H. (2018, 27-29 Haziran). *Comparison of deep learning and the classical machine learning algorithm for the malware detection.* In 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Busan, Korea (South), 293-296

4. Ranganathan, G. (2021). A study to find facts behind preprocessing on deep learning algorithms. *Journal of Innovative Image Processing*, 3(1), 66-74.

5. Ma, T. M., Yamamori, K., ve Thida, A. (2020, 13-16 Ekim). *A comparative approach to naïve bayes classifier and support vector machine for email spam classification.* In IEEE 9th Global Conference on Consumer Electronics (GCCE), Kobe, Japan, 324-326.

6. Baktır, N., ve Atay, Y. (2022). Comparative analysis of machine learning approaches in the spam-mail classification problem. *Bilişim Teknolojileri Dergisi*, 15(3), 349-364.

7. İnternet: Aha, D. (1987). UC irvine machine learning repository. *UCI Machine Learning Repository*. Web: https://archive.ics.uci.edu/ml/datasets.php, Son Erişim Tarihi: 09.10.2022.

8. Shuaib, M., Abdulhamid, S. M., Adebayo, O. S., Osho, O., Idris, I., Alhassan, J. K., ve Rana, N. (2019). Whale optimization algorithm-based email spam feature selection method using rotation forest algorithm for classification. *SN Applied Sciences*, 1(5), 1-17.

9. Hassani, Z., Hajihashemi, V., Borna K, ve Sahraei Dehmajnoonie, I. (2020). A classification method for e-mail spam using a hybrid approach for feature selection optimization. *Journal of Sciences, Islamic Republic of Iran*, 31(2), 165-173.

10. Cai, J., Luo, J., Wang, S., ve Yang, S. (2018). Feature selection in machine learning: a new perspective. *Neurocomputing*, 300, 70-79.

11. Rostami, M., Berahmand, K., Nasiri, E., ve Forouzandeh, S. (2021). Review of swarm intelligence-based feature selection methods. *Engineering Applications of Artificial Intelligence*, 100, 104210.

12. Martínez Torres, J., Iglesias Comesaña, C., ve García-Nieto, P. J. (2019). Review: machine learning techniques applied to cybersecurity. *International Journal of Machine Learning and Cybernetics*, 10(10), 2823-2836.

13. Ucci, D., Aniello, L., ve Baldoni, R. (2019). Survey of machine learning techniques for malware analysis. *Computers & Security*, 81, 123-147.

14. Li, J. (2018). Cyber security meets artificial intelligence: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(12), 1462-1474.

15. Alurkar, A. A., Ranade, S. B., Joshi, S. V., Ranade, S. S., Shinde, G. R., Sonewar, P. A., ve Mahalle, P. N. (2019). A comparative analysis and discussion of email spam classification methods using machine learning techniques. *Applied Machine Learning for Smart Data Analysis*,185-206.

16. Shah, N. F., ve Kumar, P. (2017, 5 Ağustos). *A comparative analysis of various spam classifications.* Progress in Intelligent Computing Techniques: Theory, Practice, and Applications: Proceedings of ICACNI 2016, Singapore, 265-271.

17. Chandrasekar, Dr. C. (2018). Classıfıcatıon techniques using spam filtering email. *International Journal of Advanced Research in Computer Science*, 9(2), 402-410.

18. Dada, E. G., Bassi, J. S., Chiroma, H., Abdulhamid, S. M., Adetunmbi, A. O., ve Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6), e01802.

19. Shuaib, M., Osho, O., Ismaila, I., ve Alhassan, J. K. (2018). Comparative analysis of classification algorithms for email spam detection. *International Journal of Computer Network and Information Security*, 12(1), 60.

20. Olatunji, S. O. (2019). Improved email spam detection model based on support vector machines. *Neural Computing and Applications*, 31(3), 691-699.

21. Venkatraman, S., Surendiran, B., ve Arun Raj Kumar, P. (2020). Spam e-mail classification for the internet of things environment using semantic similarity approach. *The Journal of Supercomputing*, 76(2), 756-776.

22. Ghosh, A., ve Senthilrajan, A. (2022, 25-27 Mayıs). *A modified naïve bayes classifier for detecting spam e-mails based on feature selection.* In 6th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 1634-1641.

23. Anvari, S., Abdollahi Azgomi, M., Ebrahimi Dishabi, M. R., ve Maheri, M. (2023). (2205-7400) weighted k-nearest neighbors classification based on whale optimization algorithm. *Iranian Journal of Fuzzy Systems*, 20(3), 61-74.

24. Bassiouni, M., Ali, M., ve El-Dahshan, E. A. (2018). Ham and spam e-mails classification using machine learning techniques. *Journal of Applied Security Research*, 13(3), 315-331.

25. Shrivastava, A., ve Dubey, R. (2018,28-29 Aralık). *Classification of spam mail using different machine learning algorithms.* In 2018 International Conference on Advanced Computation and Telecommunication (ICACAT), Bhopal, India,1-10.

26. Alghobiri, M. (2018). A comparative analysis of classification algorithms on diverse datasets. *Engineering, Technology & Applied Science Research*, 8(2), 2790-2795.

27. Mohammed, Z., Ja, M. F., Mp, M. I., Basthikodi, M., ve Faizabadi, A. R. (2019). A comparative study for spam classifications in email using naïve bayes and svm algorithm. *Journal of Emerging Technologies and Innovative Research*, 6(5), 391-393.

28. Kumar, P. (2020). Predictive analytics for spam email classification using machine learning techniques. *International Journal of Computer Applications in Technology*, 64(3), 282-296.

29. Nandhini, S., ve Marseline K.S., J. (2020, 24-25 Şubat). *Performance evaluation of machine learning algorithms for email spam detection.* In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 1-4.

30. Verma, S., ve Gautam, A. K. (2019, Eylül). *Machine learning techniques for classification of spambase dataset.* In Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control, New York, USA, 1-6.

31. Ishfaq, U., Shabbir, D., Khan, J., Khan, H. U., Naseer, S., Irshad, A., Shafiq, M., Hamam, H. (2022). Empirical analysis of machine learning algorithms for multiclass prediction. *Wireless Communications and Mobile Computing*, 1-12.

32. Yawen, W., Fan, Y., ve Yanxi, W. (2018). Research of email classification based on deep neural network. *International Journal of Advanced Network, Monitoring and Controls*, 3(2), 17-21.

33. Vinitha, V. S., ve Renuka, D. K. (2019, 04-06 Nisan). *Performance analysis of e-mail spam classification using different machine learning techniques.* In International Conference on Advances in Computing and Communication Engineering (ICACCE), Sathyamangalam, India, 1-5.

34. Zhou, X.-H., Zhang, M.-X., Xu, Z.-G., Cai, C.-Y., Huang, Y.-J., ve Zheng, Y.-J. (2019). Shallow and deep neural network training by water wave optimization. *Swarm and Evolutionary Computation*, 50, 100561.

35. Paudice, A., Muñoz-González, L., ve Lupu, E. C. (2019). Label sanitization against label flipping poisoning attacks. *Springer Nature Switzerland AG 2019*, 5-15.

36. Zhang, H., Cheng, N., Zhang, Y., ve Li, Z. (2021). Label flipping attacks against naive bayes on spam filtering systems. *Applied Intelligence*, 51, 4503-4514.

37. Oyegoke, T. O., Akomolede, K. K., Aderounmu, A. G., ve Adagunodo, E. R. (2021). A multi-layer perceptron model for classification of e-mail fraud. *European Journal of Information Technologies and Computer Science*, 1(5), 16-22.

38. Alauthman, M. (2020). Botnet spam e-mail detection using deep recurrent neural network. *International Journal of Emerging Trends in Engineering Research*, 8(5), 1979-1986.

39. Al-Rawashdeh, G., Mamat, R., ve Abd Rahim, N. H. B. (2019). Hybrid water cycle optimization algorithm with simulated annealing for spam e-mail detection. *IEEE Access*, 7, 143721-143734.

40. Amjad, S., ve Soleimanian Gharehchopogh, F. (2019). A novel hybrid approach for email spam detection based on scatter search algorithm and k-nearest neighbors. *Journal of Advances in Computer Engineering and Technology*, 5(3), 181-194.

41. Juneja, K. (2022). Two-phase fuzzy feature-filter based hybrid model for spam classification. *Journal of King Saud University - Computer and Information Sciences*, 34(10), 10339-10355.

42. Iqbal, K., ve Khan, M. S. (2022). Email classification analysis using machine learning techniques. *Applied Computing and Informatics*, (ahead-of-print).

43. Takci, H., Nusrat, F., ve Women, B. (2023). Highly accurate spam detection with the help of feature selection and data transformation. *International Arab Journal of Information Technology*, 20(1), 29-37.

44. Ghosh, A., ve Senthilrajan, A. (2023). Comparison of machine learning techniques for spam detection. *Multimedia Tools and Applications*, 1-28.

45. Taloba, A. I., ve Ismail, S. S. I. (2019, 08-10 Aralık*). An intelligent hybrid technique of decision tree and genetic algorithm for e-mail spam detection.* In Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 99-104.

46. Deepika, M., ve Hegde, N. P. (2021). Framework for spam detection using multi-objective optimization algorithm. *Smart Computing Techniques and Applications*, 225, 345-355.

47. Metsis, V., Androutsopoulos, I., ve Paliouras, G. (2006,27-28 Temmuz). *Spam filtering with naive bayes-which naive bayes?* In Third Conference on Email and Anti-Spam (CEAS), Mountain View, California USA, 17, 28-69.

48. Wang, Y., ve Feng, L. (2018). Hybrid feature selection using component co-occurrence based feature relevance measurement. *Expert Systems with Applications*, 102, 83-99.

49. Sharmin, S., Shoyaib, M., Ali, A. A., Khan, M. A. H., ve Chae, O. (2019). Simultaneous feature selection and discretization based on mutual information. *Pattern Recognition*, 91, 162-174.

50. Wah, Y. B., Ibrahim, N., Hamid, H. A., Abdul-Rahman, S., ve Fong, S. (2018). Feature selection methods: case of filter and wrapper approaches for maximising classification accuracy. *Pertanika Journal of Science & Technology*, 26(1).

51. Ara, J. (2018). A survey of existing e-mail spam filtering methods considering machine learning techniques. *Global Journal of Computer Science and Technology*, 18(C2), 21-29.

52. Peng, X., ve Zhu, Y. (2019, 19-21 Ekim). *A novel strategy to adjust feature weights for data classification.* In 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Suzhou, China, 1-5.

53. Wang, Y., ve Feng, L. (2020). Improved adaboost algorithm for classification based on noise confidence degree and weighted feature selection. *IEEE Access*, 8, 153011-153026.

54. Zhao, Y., ve Duangsoithong, R. (2019, 10-13 Temmuz). *Empirical analysis using feature selection and bootstrap data for small sample size problems*. In 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Pattaya, Thailand, 814-817.

55. Sokhangoee, Z. F., ve Rezapour, A. (2022). A novel approach for spam detection based on association rule mining and genetic algorithm. *Computers & Electrical Engineering*, 97, 107655.

56. Ablel-Rheem, D. M., Ibrahim, A. O., Kasim, S., Almazroi, A. A., ve Ismail, M. A. (2020). Hybrid feature selection and ensemble learning method for spam email classification. *International Journal of Advanced Trends in Computer Science and Engineering*, 9, 217-223.

57. Sumathi, S., ve Pugalendhi, G. K. (2021). Cognition based spam mail text analysis using combined approach of deep neural network classifier and random forest. *Journal of Ambient Intelligence and Humanized Computing*, 12(6), 5721-5731.

58. Sisodia, D., ve Sisodia, D. S. (2022). Feature distillation and accumulated selection for automated fraudulent publisher classification from user click data of online advertising. *Data Technologies and Applications*, 56(4), 602-625.

59. Kohavi, R., ve John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2), 273-324.

60. Sipper, M., Fu, W., Ahuja, K., ve Moore, J. H. (2018). Investigating the parameter space of evolutionary algorithms. *BioData Mining*, 11(1), 2.

61. Singh, M. (2019, 10 Temmuz). *Classification of spam email using intelligent water drops algorithm with naive bayes classifier*. In Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2017, Singapore, 133-138.

62. Rawashdeh, G., Mamat, R., Bakar, Z. B. A., ve Abd Rahim, N. H. (2019). Comparative between optimization feature selection by using classifiers algorithms on spam email. *International Journal of Electrical and Computer Engineering*, 9(6), 5479.

63. Ebadati, O. M. E., ve Ahmadzadeh, F. (2019). Classification spam email with elimination of unsuitable features with hybrid of ga-naive bayes. *Journal of Information & Knowledge Management*, 18(01), 1950008.

64. Sharaff, A., ve Gupta, H. (2019, 22 Mayıs). *Extra-tree classifier with metaheuristics approach for email classification.* In Advances in Computer Communication and Computational Sciences: Proceedings of IC4S 2018, Singapore, 189-197.

65. Sabbah, T. (2020, 16-17 Aralık). *Enhanced genetic algorithm for optimized classification.* In International Conference on Promising Electronic Technologies (ICPET), Jerusalem, Palestine, 161-166.

66. Huang, Z. (2020). A feature selection approach combining neural networks with genetic algorithms. *AI Communications*, 32(5-6), 361-372.

67. Rostami, M., Berahmand, K., ve Forouzandeh, S. (2021). A novel community detection based genetic algorithm for feature selection. *Journal of Big Data*, 8(1), 2.

68. Kim, M., Bae, J., ve Lim, J. S. (2021, 20-22 Ekim). *Selecting a suitable feature subset for classification using multi-agent reinforcement learning.* In International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 501-504.

69. Ghimatgar, H., Kazemi, K., Helfroush, M. S., ve Aarabi, A. (2018). An improved feature selection algorithm based on graph clustering and ant colony optimization. *Knowledge-Based Systems*, 159, 270-285.

70. Montemayor, J. J. M., ve Crisostomo, R. V. (2019, 10-13 Haziran). *Feature selection in classification using binary max-min ant system with differential evolution.* In IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 2559-2566.

71. Akinyelu, A. A., Ezugwu, A. E., ve Adewumi, A. O. (2020). Ant colony optimization edge selection for support vector machine speed optimization. *Neural Computing and Applications*, 32(15), 11385-11417.

72. Akcan, U. (2022). *Karınca Koloni Algoritması ile Öznitelik Seçimi*, Yüksek Lisans Tezi, Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü, Bursa.

73. Al-Azzawi, F. (2018). *Wrapper Feature Selection Approach for Spam E-Mail Filtering*, Yüksek Lisans Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Kayseri.

74. Ebadati, O. M. E., ve Ahmadzadeh, F. (2019). Classification spam email with elimination of unsuitable features with hybrid of ga-naive bayes. *Journal of Information & Knowledge Management*, 18(01), 1950008.

75. Dedetürk, B. K. (2020). *Yapay arı kolonisi temelli lojistik regresyon sınıflayıcıların optimal tasarımı ve Türkçe spam maillerin filtrelenmesinde başarımlarının incelenmesi*, Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Kayseri.

76. Ahmed, B. (2020). Wrapper feature selection approach based on binary firefly algorithm for spam e-mail filtering. *Journal of Soft Computing and Data Mining*, 1(2), 44-52.

77. Saleh, H. M. (2021). An efficient feature selection algorithm for the spam email classification. *Periodicals of Engineering and Natural Sciences*, 9(3), 520-531.

78. Mashaleh, A. S., Binti Ibrahim, N. F., Al-Betar, M. A., Mustafa, H. M. J., ve Yaseen, Q. M. (2022). Detecting spam email with machine learning optimized with harris hawks optimizer (hho) algorithm. *Procedia Computer Science*, 201, 659-664.

79. Kühl, N., Goutier, M., Hirt, R., ve Satzger, G. (2020, 27 Mart). *Machine learning in artificial intelligence: towards a common understanding.* In Proceedings of the 52nd Hawaii International Conference on System Sciences, Hawaii, 5236-5245.

80. Mijwil, M. M. (2020). Malware detection in android os using machine learning techniques. *Data Science and Applications*, 3(2), 5-9.

81. Friedman, J. H. (1998). Data mining and statistics: what's the connection? *Computing science and statistics*, 29(1), 3-9.

82. Ongsulee, P. (2017, 22-24 Kasım). *Artificial intelligence, machine learning and deep learning.* In 15th International Conference on ICT and Knowledge Engineering (ICT&KE), Bangkok, Thailand, 1-6.

83. Fong, R. C., Scheirer, W. J., ve Cox, D. D. (2018). Using human brain activity to guide machine learning. *Scientific reports*, 8(1), 1-10.

84. Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Ganguli, S. (2019). A deep learning framework for neuroscience. *Nature neuroscience*, 22(11), 1761-1770.

85. Aggarwal, K., Mijwil, M. M., Al-Mistarehi, A.-H., Alomari, S., Gök, M., Alaabdin, A. M. Z., ve Abdulrhman, S. H. (2022). Has the future started? The current growth of artificial intelligence, machine learning, and deep learning. *Iraqi Journal for Computer Science and Mathematics*, 3(1), 115-123.

86. Hussain, K., Mohd Salleh, M. N., Cheng, S., ve Shi, Y. (2019). Metaheuristic research: a comprehensive survey. *Artificial intelligence review*, 52, 2191-2233.

87. Sörensen, K. (2015). Metaheuristics the metaphor exposed. *International Transactions in Operational Research*, 22(1), 3-18.

88. Kumar, R. A., Franklin, J. V., ve Koppula, N. (2022). A comprehensive survey on metaheuristic algorithm for feature selection techniques. *Materials Today: Proceedings*, 64(1), 435-441.

89. Dorigo, M., ve Blum, C. (2005). Ant colony optimization theory: a survey. *Theoretical computer science*, 344(2-3), 243-278.

90. Jović, A., Brkić, K., ve Bogunović, N. (2015, 25-29 Mayıs). *A review of feature selection methods with applications.* In 38th international convention on information and communication technology, electronics and microelectronics (MIPRO), Opatija, Croatia, 1200-1205.

91. Hall, M. A. (1999). *Correlation-based Feature Selection for Machine Learning*, Doktora Tezi, Waikato Üniversitesi, Yeni Zelenda.

92. Zhou, H., Wang, X., ve Zhu, R. (2022). Feature selection based on mutual information with correlation coefficient. *Applied Intelligence*, 1-18.

93. Altarabichi, M. G., Fan, Y., Pashami, S., Mashhadi, P. S., ve Nowaczyk, S. (2021, 06-09 Ekim). *Extracting invariant features for predicting state of health of batteries in hybrid energy buses.* In 8th international conference on data science and advanced analytics (dsaa), Porto, Portugal, 1-6.

94. Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., ve Lang, M. (2020). Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis*, 143, 106839.

95. Tabakhi, S., Moradi, P., ve Akhlaghian, F. (2014). An unsupervised feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence*, 32, 112-123.

96. Gheyas, I. A., ve Smith, L. S. (2010). Feature subset selection in large dimensionality domains. *Pattern Recognition*, 43(1), 5-13.

97. Yin, P.-Y. (2009). *Pattern Recognition*, Croatia: In-teh.

98. Ghosh, M., Guha, R., Sarkar, R., ve Abraham, A. (2020). A wrapper-filter feature selection technique based on ant colony optimization. *Neural Computing and Applications*, 32(12), 7839-7857.

99. Bolón-Canedo, V., Sánchez-Maroño, N., ve Alonso-Betanzos, A. (2015). *Feature selection for high-dimensional data*, London: Springer.

100. Altarabichi, M. G., Nowaczyk, S., Pashami, S., ve Mashhadi, P. S. (2021, 28 Haziran-01 Temmuz). *Surrogate-assisted genetic algorithm for wrapper feature selection.* In IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 776-785.

101. Ghafari, S. M., ve Tjortjis, C. (2019). A survey on association rules mining using heuristics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(4), e1307.

102. Zhao, Q., ve Bhowmick, S. S. (2003). Association rule mining: a survey. *Nanyang Technological University*, 135.

103. Agrawal, R., ve Srikant, R. (1994, Eylül). *Fast algorithms for mining association rules.* In Proceeding 20th International Conference Very Large Databases, VLDB, Santiago, Chile, 487-499.

104. Ahmed, I., Guan, D., ve Chung, T. C. (2014). Sms classification based on naive bayes classifier and apriori algorithm frequent itemset. *International Journal of machine Learning and Computing*, 4(2), 183.

105. Shannon, C. E. (1949). Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4), 656-715.

106. Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4), 537-550.

107. Unhale, T., ve Sonawani, S. (2022, 27 Nisan). *Novel approach for feature selection using genetic algorithm.* In Proceeding of International Conference on Computational Science and Applications: ICCSA, Singapore, 3-12.

108. Ferrari Dacrema, M., Moroni, F., Nembrini, R., Ferro, N., Faggioli, G., ve Cremonesi, P. (2022, 11-15 Temmuz). *Towards feature selection for ranking and classification exploiting quantum annealers.* In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 2814-2824.

109. Holland J. H. (1976), *Adaptation in natural and artificial systems*, London: MIT Press.

110. Krömer, P., Platoš, J., Nowaková, J., ve Snášel, V. (2018). Optimal column subset selection for image classification by genetic algorithms. *Annals of Operations Research*, 265(2), 205-222.

111. Dorigo, M., Maniezzo, V., ve Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1), 29-41.

112. Gambardella, L. M., ve Dorigo, M. (1996, 20-22 Mayıs). *Solving symmetric and asymmetric tsps by ant colonies.* In Proceedings of IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 622-627.

113. Leguizamon, G., ve Michalewicz, Z. (1999, 06-09 Temmuz). *A new version of ant system for subset problems.* In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 1459-1464.

114. Aghdam, M. H., ve Kabiri, P. (2016). Feature selection for intrusion detection system using ant colony optimization. *International Journal of Network Security*, 18(3), 420-432.

115. Uthayakumar, J., Metawa, N., Shankar, K., ve Lakshmanaprabu, S. K. (2020). Financial crisis prediction model using ant colony optimization. *International Journal of Information Management*, 50, 538-556.

116. Hosmer Jr, D. W., Lemeshow, S., ve Sturdivant, R. X. (2013). *Applied logistic regression*. New Jersey: John Wiley & Sons.

117. El Sanharawi, M., ve Naudet, F. (2013). Comprendre la régression logistique. *Journal français d'ophtalmologie*, 36(8), 710-715.

118. Breiman, L. (2001). Random forests. *Machine learning*, 45, 5-32.

119. Breiman, L. (1996). Bagging predictors. *Machine learning*, 24, 123-140.

120. Cuijuan, C. (2021). Improved multiple naive bayes classification algorithms and python implementation [j]. *Journal of Jingdezhen University*, 36(03), 92-95.

121. Murphy, K. P. (2018). *Machine learning: a probabilistic perspective*, London: MIT Press.

122. Ye, F., Chen, G., Liu, Q., Zhang, L., Qi, Q., Hu, B., ve Fan, X. (2022, 04-06 Mart). *A spam classification method based on naive bayes.* In IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 1856-1861.

123. Zhang, S., Li, X., Zong, M., Zhu, X., ve Wang, R. (2018). Efficient knn classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5), 1774-1785.

124. Bolón-Canedo, V., Sánchez-Maroño, N., ve Alonso-Betanzos, A. (2015). *Feature selection for high-dimensional data*, London: Springer.

125. Nagendran, G. A., Singh, H., Raj, R. J. S., ve Muthukumaran, N. (2021, 04-06 Şubat). *Input assistive keyboards for people with disabilities: a survey.* In Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 829-832.

126. Manimegalai, T., Ananthi, S., Kamatchi, S., Anitha, G., ve Muthuraman, U. (2022, 20-22 Haziran). *ASVM: adaboost with svm-based classifier implementation for aspect-based opinion mining to appraise products.* In International Conference on Inventive Computation Technologies (ICICT), Nepal, 1034-1041.

127. Lee, S.-J., Tseng, C.-H., Yang, H.-Y., Jin, X., Jiang, Q., Pu, B., Zhao, N. (2022). Random rotboost: an ensemble classification method based on rotation forest and adaboost in random subsets and its application to clinical decision support. *Entropy*, 24(5), 617.

128. Misra, S., Li, H., ve He, J. (2020). Noninvasive fracture characterization based on the classification of sonic wave travel times. *Machine Learning for Subsurface Characterization*, 243.

129. Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 121-167.

130. Soofi, A. A., ve Awan, A. (2017). Classification techniques in machine learning: applications and issues. *International Journal of Basic and Applied Science*, 13, 459-465.

131. Ghosh, S., Dasgupta, A., ve Swetapadma, A. (2019, 21-22 Şubat). *A study on support vector machine based linear and non-linear pattern classification.* In 2019 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 24-28.

132. Safavian, S. R., ve Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3), 660-674.

133. Guyon, I., Gunn, S., Nikravesh, M., ve Zadeh, L. A. (2008). *Feature extraction: foundations and applications*, Netherlands: Springer.

134. Stamp, M., Alazab, M., ve Shalaginov, A. (2021). *Malware analysis using artificial intelligence and deep learning*, Cham: Springer.

135. Marakhimov, A. R., ve Khudaybergenov, K. K. (2020). Approach to the synthesis of neural network structure during classification. *International Journal of Computing*, 20-26.

136. Sharma, O. (2019, 14-16 Şubat). *A new activation function for deep neural network.* In International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 84-86.

137. Kingma, D. P., ve Ba, J. (2017). *Adam: a method for stochastic optimization*, In International Conference for Learning Representations, San Diego, 1-13.

138. Dauphin, Y., De Vries, H., ve Bengio, Y. (2015). Equilibrated adaptive learning rates for non-convex optimization. *Advances in Neural İnformation Processing Systems*, 28.

139. Yousoff, S. N. M., Baharin, A., ve Abdullah, A. (2016, 04-08 Aralık). *A review on optimization algorithm for deep learning method in bioinformatics field.* In IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES), Kuala Lumpur, Malaysia, 707-711.

140. Brown, C. T., Bullen, H. W., Kelly, S. P., Xiao, R. K., Satterfield, S. G., ve Hagedorn, J. G. (2003). Visualization and data mining in an 3d immersive environment: summer project 2003. *US National Institute of Standards and Technology*.

141. Abualigah, L. (2021). Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. *Neural Computing and Applications*, 33(7), 2949-2972.

142. Abualigah, L., ve Diabat, A. (2021). A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments. *Cluster Computing*, 24, 205-223.

143. Mining, W. I. D. (2006). Data mining: concepts and techniques. *Morgan Kaufinann*, 10, 559-569.

144. Bimantara, A., ve Dina, T. A. (2019). Klasifikasi web berbahaya menggunakan metode logistic regression. *Annual Research Seminar (ARS)*, 4(1), 173-177.

145. Genuer, R., Poggi, J.-M., Genuer, R., ve Poggi, J.-M. (2020). *Random forests*, London: Springer.

146. Kramer, O., ve Kramer, O. (2013). K-nearest neighbors. *Dimensionality Reduction with Unsupervised Nearest Neighbors*, 13-23.

147. Schapire, R. E. (2013). Explaining adaboost. *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, 37-52.

148. Singh, M., Pamula, R., ve shekhar, S. kumar. (2018, 28-29 Eylül). *Email spam classification by support vector machine.* In International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 878-882.

149. De Ville, B. (2013). Decision trees. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(6), 448-455.

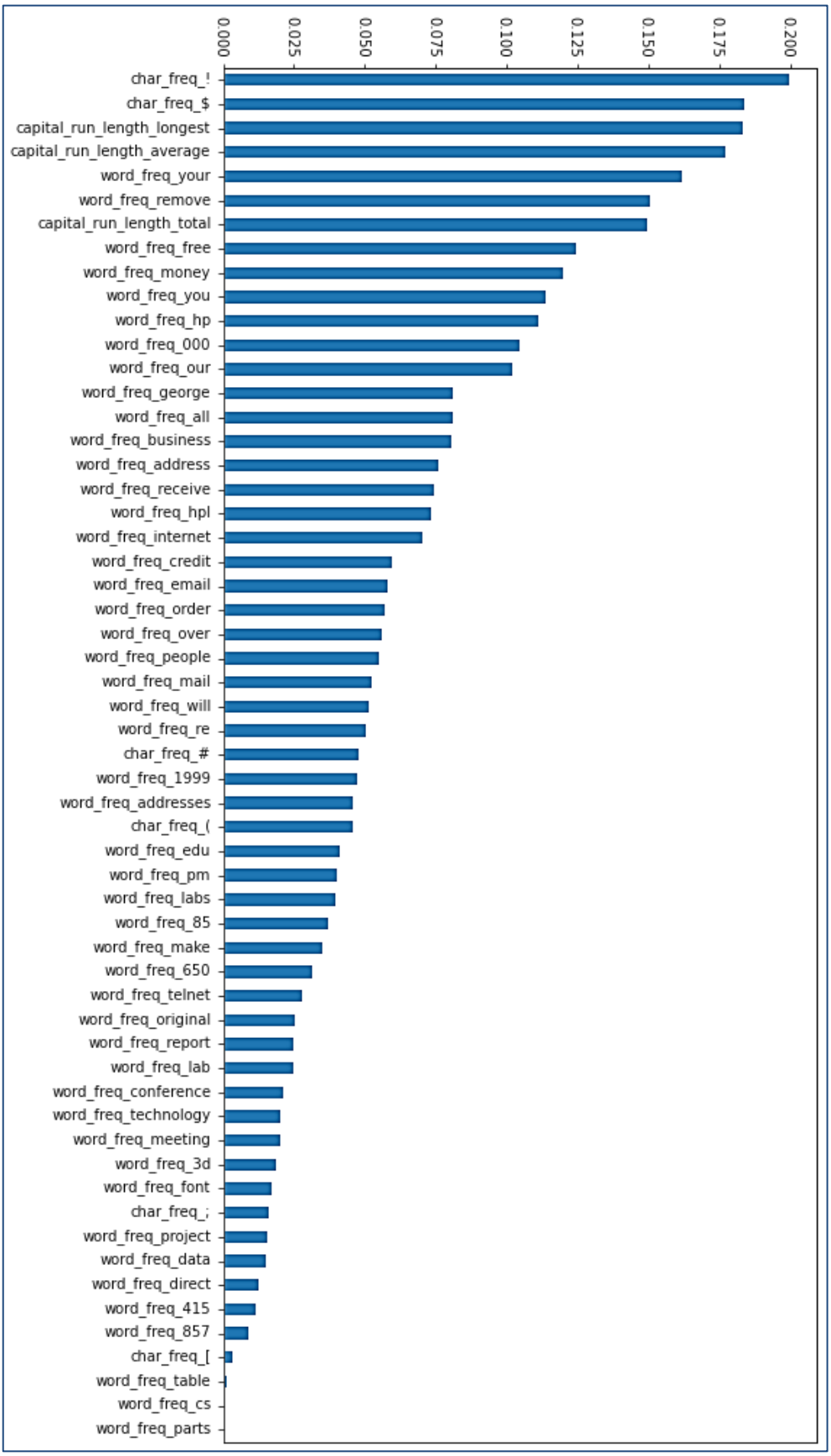
150. Zhang, G. P. (2000). Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(4), 451-462.

151. Agostinelli, F., Hoffman, M., Sadowski, P., ve Baldi, P. (2015). Learning activation functions to improve deep neural networks. *International Conference on Learning Representations (ICLR) 2015*.

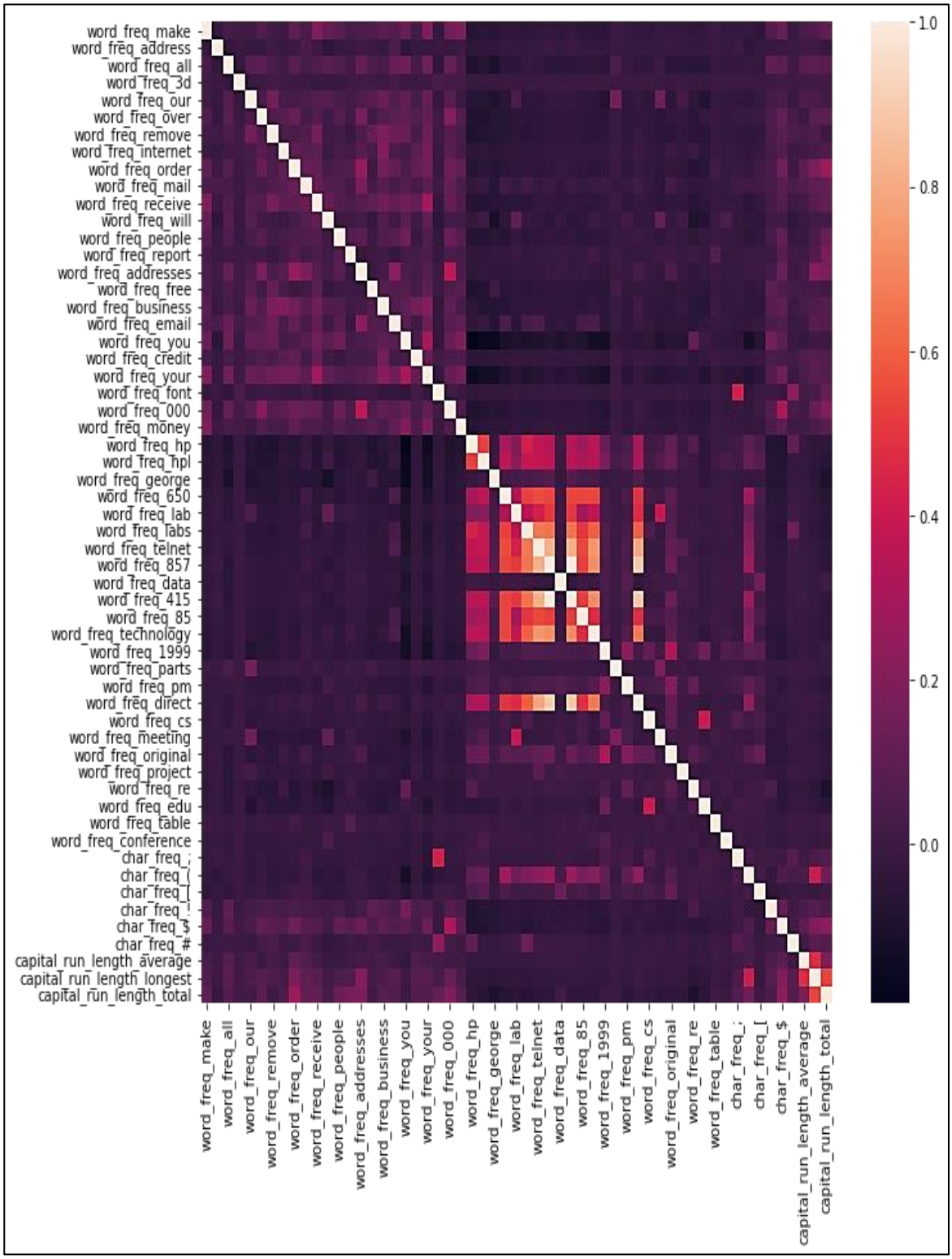
152. Sarker, S. (2019). *A Feature Group Weighting Method for Classifying High-Dimensional Big Data*, Yüksek Lisans Tezi, Birleşik Uluslararası Üniversite, Bangladeş.

EKLER

EK-1. Karşılıklı bilgi özellik seçim yöntemi ile özeliklerin sıralanışı



EK-2. Korelasyon özellik seçiminde niteliklerin birbiriyle ilişkisi



EK-3. Sarmal yöntemler ile sınıflandırma sonuçları[8]

		LR	RF	NB	KNN	Adaboost	LSVM	DT	DNN
GA ‰1	Doğruluk	90,99	95,98	90,01	88,27	93,7	87,62	93,59	93,7
	Kesinlik	92,69	97,27	93,26	88,11	94,39	96,17	93,9	95,6
	Duyarlılık	87,69	94,1	82,56	83,59	91,54	97,69	91,03	90,51
	F1-Skor	89,18	95,2	87,5	85,79	92,41	85,35	92,31	92,31
GA %1	Doğruluk	90,23	95,77	90,45	89,03	93,92	87,95	92,83	94,68
	Kesinlik	93	97,03	94,41	88,33	96,07	94,46	93,32	95,82
	Duyarlılık	86,15	93,33	82,31	86,67	91,03	96,15	90	92,31
	F1-Skor	88,19	94,9	87,95	86,83	92,69	85,53	91,41	93,63
GA %20	Doğruluk	91,53	96,09	91,1	90,12	94,57	89,47	94,57	95,01
	Kesinlik	93,33	97,33	93,75	88,68	93,98	93,33	95,21	97,38
	Duyarlılık	87,18	94,1	85,64	88,21	93,33	96,92	91,79	93,33
	F1-Skor	89,71	95,29	88,95	88,32	93,57	87,86	93,47	94,06
ACO	Doğruluk	91,31	95,11	90,66	89,47	94,14	89,03	92,18	94,46
	Kesinlik	93,66	97,52	93,93	89,49	95,12	96,17	92,47	97,16
	Duyarlılık	87,95	92,31	83,33	87,44	92,31	99,23	90	93,33
	F1-Skor	89,36	94,12	88,32	87,25	93,02	86,93	90,7	93,45

[8] Sonuçlar iki haneli olarak yuvarlama yapılarak gösterilmiştir.

EK-4. Filtreleme yöntemleri ile sınıflandırma sonuçları

		LR	RF	NB	KNN	Adaboost	LSVM	DT	DNN
ARM	Doğruluk	88,71	92,73	85,23	89,14	90,88	88,71	87,62	92,73
	Kesinlik	89,94	93,3	84,51	87,18	89,43	90,4	85,75	90,68
	Duyarlılık	82,56	89,23	79,74	87,18	88,97	82,05	84,87	92,31
	F1-Skor	86,1	91,22	82,06	87,18	89,2	86,02	85,31	91,49
MIFS	Doğruluk	91,21	95,44	88,82	88,17	93,59	91,21	90,99	91,97
	Kesinlik	93,04	96,77	92,84	89,36	92,33	93,04	93,24	87,09
	Duyarlılık	85,64	92,31	79,74	81,79	92,56	85,64	84,87	95,13
	F1-Skor	89,19	94,49	85,79	85,41	92,45	89,19	88,86	90,93
VT	Doğruluk	91,31	95,44	89,03	79,26	93,49	82,63	89,14	94,79
	Kesinlik	89,95	97,03	90,48	78,19	93,19	74,89	87,76	95,24
	Duyarlılık	89,49	92,05	82,82	70,77	91,28	88,72	86,41	92,31
	F1-Skor	89,72	94,47	86,48	74,29	92,23	81,22	87,08	93,75
CFS	Doğruluk	91,75	95,01	89,25	78,61	93,38	83,93	90,01	93,7
	Kesinlik	92,2	96,24	91,93	77,03	92,51	95,49	90,93	95,11
	Duyarlılık	87,95	91,79	81,79	70,51	91,79	65,13	84,87	89,74
	F1-Skor	90,03	93,96	86,57	73,63	92,15	77,44	87,8	92,35



Gazili olmak ayrıcalıktır