

**KARINCA KOLONİ ALGORİTMASI İLE ÖZNEİELİK  
SEÇİMİ**

**UMUT AKCAN**



T.C.  
BURSA ULUDAĞ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

## KARINCA KOLONİ ALGORİTMASI İLE ÖZİNİTELİK SEÇİMİ

UMUT AKCAN

0000-0002-7533-9397

Doç. Dr. Duygu YILMAZ EROĞLU  
(Danışman)

YÜKSEK LİSANS  
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2022  
Her Hakkı Saklıdır

**B.U.Ü. Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;**

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

**beyan ederim.**

**10/01/2022**

**Umut AKCAN**

## **TEZ YAYINLANMA FİKRİ MÜLKİYET HAKLARI BEYANI**

Enstitü tarafından onaylanan lisansüstü tezin/raporun tamamını veya herhangi bir kısmını, basılı (kâğıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma izni Bursa Uludağ Üniversitesi'ne aittir. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet hakları ile tezin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları tarafımıza ait olacaktır. Tezde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alınarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederiz.

Yükseköğretim Kurulu tarafından yayınlanan “**Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge**” kapsamında, yönerge tarafından belirtilen kısıtlamalar olmadığı takdirde tezin YÖK Ulusal Tez Merkezi / B.U.Ü. Kütüphanesi Açık Erişim Sistemi ve üye olunan diğer veri tabanlarının (Proquest veri tabanı gibi) erişimine açılması uygundur.

## ÖZET

Yüksek Lisans Tezi

### KARINCA KOLONİ ALGORİTMASI İLE ÖZİNİTELİK SEÇİMİ

UMUT AKCAN

Bursa Uludağ Üniversitesi  
Fen Bilimleri Enstitüsü  
Endüstri Mühendisliği Anabilim Dalı

**Danışman: Doç. Dr. Duygu YILMAZ EROĞLU**

Gelişen bilgi teknolojileri ile günümüzde veri miktarı hızla büyümektedir. Veri madenciliğinin amacı, bu verilerden anlamlı bilgi çıkarmaktır. Veri miktarının büyük ve çok boyutlu olması, hesaplama maliyetlerini artırmakla beraber verilerden anlamlı bilgi çıkartılmasını zorlaştırmaktadır. Öznitelik seçiminin amacı bilgi kaybının asgari düzeyde tutarak verilerin çok boyutluluğunu azaltmaktır. Literatürde, öznitelik seçimi için filtre, sarmalayıcı, gömülü ve hibrit yöntemler başlıkları altında farklı yaklaşımlar önerilmiştir. Bu tez çalışmasında, karınca koloni algoritması kullanılarak hibrit bir yaklaşım önerilmiştir. Hibrit yöntemler, iki adımdan oluşmaktadır. Önerilen yöntemin ilk adımında karınca koloni algoritması ile denetimsiz öğrenme şeklinde öznitelik seçimi yapılmıştır. İkinci adımında k en yakın komşuluk ve destek vektör makineleri sınıflandırma yöntemleri kullanılarak sınıflandırma modelleri oluşturulmuştur. Elde edilen sonuçlar, literatürde karınca koloni algoritması kullanan bir çalışma ile karşılaştırılmış olup, ortak kullanılan veri setlerinin yarısında daha iyi sonuçlara ulaşılmıştır. Bu sonuçlar, önerilen yöntemin etkinliğini doğrulanmış sonrasında daha yüksek doğruluk oranları elde etmek için hangi özniteliklerin kullanılması gerektiğine karar verebilmek amacıyla 10 kat çapraz doğrulama ile farklı sınıflandırıcılar kullanılmıştır. İlâveten, bir adet özniteliğinin bile varlığının ve yokluğunun sonuçları nasıl etkilediğini göstermek için analizler yapılmış, öznitelik seçiminin önemi vurgulanmıştır. Son olarak da farklı sınıflandırıcı ve eğitim/test yapılarında veri setlerinin doğruluk oranı dışında hassasiyet ve gerçek pozitif değerler oranından hesaplanan F-puanının nasıl değiştiğine dair analizler yapılarak sonuçlar yorumlanmıştır.

**Anahtar Kelimeler:** Karınca koloni algoritması, öznitelik seçimi, sınıflandırma, veri ön işleme, hibrit algoritmalar

**2022, vii + 45 sayfa**

## ABSTRACT

MCs Thesis

### FEATURE SELECTION WITH ANT COLONY ALGORITHM

UMUT AKCAN

Bursa Uludağ University  
Graduate School of Natural and Applied Sciences  
Department of Industrial Engineering

**Supervisor: Assoc. Dr. Duygu YILMAZ EROĞLU**

Nowadays with the developing information technologies, the amount of data is growing rapidly. The purpose of data mining is to extract meaningful information from these data. The fact that the amount of data is large and multidimensional increases the computational costs and makes it difficult to extract meaningful information from the data. The purpose of feature selection is to reduce the multidimensionality of the data by keeping information loss to a minimum. In the literature, different approaches have been proposed for feature selection under the headings of filter, wrapper, embedded and hybrid methods. In this thesis, a hybrid approach is proposed using the ant colony algorithm. Hybrid methods consist of two steps. In the first step of the proposed method, feature selection was made in unsupervised learning with the ant colony algorithm. In the second step, classification models are created by using k nearest neighbor and support vector machine classification methods. The results obtained were compared with a study using the ant colony algorithm in the literature. Better results were achieved in half of the commonly used datasets. According to these results, the effectiveness of the proposed method was verified, and then different classifiers were used with 10-fold cross validation in order to decide which features should be used to achieve higher accuracy rates. In addition, analyzes were made to show how the presence and absence of even one feature affected the results, and the importance of feature selection was emphasized. Finally, analyzes were made on how the F-score calculated from precision and true positive values, apart from the accuracy rate of the data sets, changed in different classifier and training/test structures, and the results were interpreted.

**Key words:** Ant colony algorithm, feature selection, classification, data pre-processing, hybrid algorithm

**2022, vii + 45 pages**

## TEŞEKKÜRLER

Bu tez çalışmasının hazırlanmasında, bilgi ve deneyimleri ile her daim yol gösteren, bilimsel eleştiri ve yönlendirmesi katkı sağlayan ve desteğini hiçbir zaman esirgemeyen değerli danışman hocam Doç. Dr. Duygu YILMAZ EROĞLU sonsuz teşekkür ederim.

Tüm hayatım boyunca olduğu gibi yüksek lisans ders süreci ve tez hazırlama sürecinde maddi manevi desteklerini hiç esirgemeyen aileme özellikle eşim Elif AKCAN'A teşekkürlerimi iletirim

Umut AKCAN  
10/01/2022

## İÇİNDEKİLER DİZİNİ

	<b>Sayfa</b>
ÖZET .....	i
ABSTRACT .....	ii
TEŞEKKÜRLER .....	iii
İÇİNDEKİLER DİZİNİ .....	iv
KISALTMALAR DİZİNİ .....	v
ŞEKİLLER DİZİNİ .....	vi
ÇİZELGELER DİZİNİ .....	vii
1. GİRİŞ .....	1
2. KAYNAK ÖZETLERİ ve KURAMSAL TEMELLER .....	4
2.1 Kaynak Özetleri .....	4
2.2 Kuramsal Temeller .....	7
2.2.1. Öznitelik seçim teknikleri .....	7
2.2.2. Temel makine öğrenme yöntemleri .....	9
2.2.3. Sınıflandırma metrikleri .....	10
3. MATERYAL ve YÖNTEM .....	13
3.1 Materyal .....	13
3.1.1. Veri setleri .....	13
3.1.2. Karınca koloni algoritması .....	15
3.1.3. K en yakın komşuluk algoritması .....	17
3.1.4. Destek vektör makineleri .....	19
3.2 Yöntem .....	22
4. BULGULAR ve TARTIŞMA .....	26
4.1. Karınca Koloni Algoritması Parametre Ayarlaması .....	26
4.2. Sonuçların Literatürdeki Çalışmalar ile Karşılaştırılması .....	28
4.3. K En Yakın Komşuluk Sınıflandırması Sonuçları .....	30
4.4. Destek Vektör Makineleri Sınıflandırması Sonuçları .....	33
4.5 K En Yakın Komşuluk ve Destek Vektör Makineleri Karşılaştırması .....	35
5. SONUÇ .....	38
KAYNAKLAR .....	39
EKLER .....	43
EK 1 Rassal Veri Seti .....	44
ÖZGEÇMİŞ .....	45



## KISALTMALAR DİZİNİ

### Kısaltmalar Açıklama

ACO	Karınca Koloni Algoritması (Ant Colony Optimization)
DT	Karar Ağacı (Decision Tree)
FS	Öznitelik Seçimi (Feature Selection)
KKN	K En Yakın Komşuluk (K Nearest Neighbors)
MC	Karşılıklı Korelasyon (Mutual Correlation)
RSM	Rastgele Alt Uzay Yöntemi (Random Subspace Method)
SI	Sürü Zekası (Swarm Intelligent)
SVM	Destek Vektör Makineleri (Support Vector Machine)

## ŞEKİLLER DİZİNİ

	<b>Sayfa</b>
Şekil 2.1. Öznitelik seçimi yöntemlerinin bilgi akışı .....	9
Şekil 2.2. Karmaşıklık matrisi .....	10
Şekil 3.1. %80 eğitim %20 test veri seti sembolik gösterimi .....	14
Şekil 3.2. 5 katlı çapraz doğrulama sembolik gösterimi .....	14
Şekil 3.3. Gerçek karınca davranışlarının sembolik gösterimi .....	15
Şekil 3.4. K en yakın komşuluk algoritması gösterimi .....	18
Şekil 3.5. Destek vektör makineleri gösterimi .....	19
Şekil 3.6. $\ w\ $ vektörünün doğruları dis kesme grafiği .....	20
Şekil 3.7. Doğrular ve $\ w\ $ vektörü arası $\theta$ açısının gösterim grafiği .....	21
Şekil 3.8. Önerilen yaklaşımın iş akış diyagramı .....	25
Şekil 4.1. Karınca koloni algoritması parametreleri Minitab ANOVA sonucu .....	26
Şekil 4.2. Karınca koloni algoritması parametreleri ana etki grafikleri .....	27
Şekil 4.3. K en yakın komşuluk algoritması ve destek vektör makineleri rassal veri seti kümesi sınıflandırma sonucu karar sınırları .....	37

## ÇİZELGELER DİZİNİ

	<b>Sayfa</b>
Çizelge 2.1. Bazı Makine öğrenmesi algoritmaları .....	10
Çizelge 3.1. Veri seti bilgileri .....	13
Çizelge 4.1. Karınca koloni algoritması seçilen parameter değerleri .....	28
Çizelge 4.2. Önerilen yöntemin, UFSACO algoritması sonuçları ile karşılaştırılması ....	28
Çizelge 4.3. Önerilen yöntemin, RSM algoritması ile karşılaştırılması .....	29
Çizelge 4.4. Önerilen yöntemin, MC algoritması ile karşılaştırılması .....	30
Çizelge 4.5. K en yakın komşuluk algoritması doğruluk oranı sonuçları .....	31
Çizelge 4.6. K en yakın komşuluk algoritması F-puanı sonuçları .....	31
Çizelge 4.7. K en yakın komşuluk algoritması çapraz doğrulama sonuçları .....	32
Çizelge 4.8. Destek vektör makineleri algoritması doğruluk oranı sonuçları .....	33
Çizelge 4.9. Destek vektör makineleri algoritması F-puanı sonuçları .....	34
Çizelge 4.10. Destek vektör makineleri algoritması çapraz doğrulama sonuçları .....	35
Çizelge 4.11. K en yakın komşuluk ve destek vektöre makineleri F-puanı karşılaştırması .....	36
Çizelge 4.12. K en yakın komşuluk ve destek vektör makineleri çapraz doğrulama doğruluk oranları karşılaştırması .....	36

## 1. GİRİŞ

Veri miktarı son yıllarda hızla artmaktadır. Makine öğrenmesi, istatistik ve veri tabanlarının kesişimindeki yöntemleri içeren bir hesaplama süreci olarak veri madenciliği, bu devasa veri hacmi ile ilgilenir, işler ve onu analiz eder (Liu ve Yu 2005). Veriler ölçek olarak çok büyük ve yüksek boyutluluğa sahip olabilirler. Yüksek boyutlu veriler daha karmaşık ve çeşitli bilgiler içermesine rağmen, hesaplama maliyetlerini ve depolama genel giderlerini de önemli ölçüde artırır. Ek olarak, yüksek boyutluluk, veri madenciliğinin karmaşıklığını artıran önemli özniteliklerin daha az görülmesine neden olur. Ayrıca, oluşturulan modellerin performansını ve verimliliğini doğrudan etkileyen bu yüksek boyutlu verilere kaçınılmaz olarak gereksiz öznitelikler, gürültü veya aykırı değerler eklenir (Yan ve Yang 2015).

Öznitelik seçimi (FS), özellikle yüksek boyutlu bir öznitelik alanıyla uğraşırken, makine öğrenmesinde yaygın olarak kullanılan bir ön işleme adımudur. Öznitelik seçiminin amacı, bir veri kümesinin boyutsallığını azaltarak ve tahmin doğruluğundan en az kayıp ile temel öznitelikleri bularak basitleştirmektir. Bunu yaparak, seçilmeyen öznitelikler tarafından sağlanan bilgilerdeki fazlalığı da azaltır. Gerçek dünya problemlerinde, gürültülü, alakasız veya yanıltıcı öznitelik bolluğu nedeniyle öznitelik seçimi bir zorunluluktur. Öznitelik seçimi kapsamlıdır ve metin kategorizasyonu, veri madenciliği, örüntü tanıma ve sinyal işleme dahil olmak üzere birçok alana yayılır (Jensen 2005). Öznitelik seçimi esasen bir kombinatoriyal optimizasyon problemidir. Özellikle NP'nin global optimal bir çözüm bulmak polinom zamanda zordur. Bu sorunu çözmek için, geleneksel öznitelik seçim yöntemleri, veri noktalarının belirli istatistiksel veya geometrik özelliklerini göre her bir öznitelik ağırlığını ayrı ayrı değerlendirir, bunları uygun şekilde sıralar ve ardından öznitelikleri tek tek seçer. Ancak, küresel optimal değer için herhangi bir garanti sağlayamazlar. Ayrıca, farklı öznitelikler arasındaki etkileşimi ve bağımlılığı ihmal etmeleri oldukça olasıdır (Yan ve Yang 2015). Zaman karmaşıklığı probleminin üstesinden gelmek için, polinom zamanında bir optimale yakın öznitelik alt kümesini bulmak için önerilen yaklaşım algoritmaları vardır. Bu algoritmalar filtre, sarmalayıcı, gömülü ve hibrit yaklaşımlar olmak üzere dört ana kategoride sınıflandırılabilir (Rostami, Berahmand, Nasiri ve Forouzande 2021). Filtre yöntemleri, yalnızca verilerin istatistiksel özelliklerine dayanır ve herhangi bir öğrenme modelinden bağımsız olduklarından,

yüksek bir hesaplama maliyetinin oluşmasını engeller ve diğer iki kategoriden daha fazla genellik sağlarlar. Sarmalayıcı yöntemlerinde, son alt küme seçimi, verilerle tekrar tekrar eğitilen bir öğrenme algoritmasına dayanır. Sarmalayıcılar, öğrenme planının nihai doğruluğunu artırma eğiliminde olsalar da, genellikle diğer iki yaklaşımdan hesaplama açısından daha yüksek maliyete sahiptir. Gömülü yöntemlerde, sınıflandırma sürecinin bir parçasıdır (Palma-Mendoza, de Marcos, Rodriguez, ve Alonso-Betanzos 2019). Bu yöntemlerde, öznitelik seçimi ve sınıflandırma eş zamanlı olarak ilerler. Hibrit yaklaşımlar, öznitelikleri iki aşamada seçmeye çalışırlar: ilk aşamada, filtre yaklaşımını kullanarak orijinal öznitelik setini azaltmaya çalışırlar. Daha sonra ikinci aşamada, indirgenmiş öznitelik kümesindeki en iyi öznitelik alt kümesini seçmek için sarmalayıcı yaklaşımı uygulanır. Başka bir deyişle, hibrit yaklaşımın amacı, hem filtre hem de sarmalayıcı yaklaşımların avantajlarını kullanmaktır. Sonuç olarak, hibrit yaklaşımda iyi özniteliklerin ortadan kaldırılması riski, filtre yaklaşımındakinden daha azdır (Unler, Murat, ve Chinnam 2011). Hibrit yaklaşımlarda ilk filtre görevi meta sezgisel algoritmalar aracılığı ile yapılabilir.

Meta-sezgisel yaklaşımlar, minimum sürede en uygun öznitelik alt kümesini bulmada başarılı yaklaşımlardan biridir. Son yıllarda, meta-sezgisel yaklaşımların bir kategorisi olan sürü zekası algoritmaları (SI), öznitelik seçimi ile verimli bir şekilde uğraşmaktadır. SI algoritmaları, karıncalar, arılar, kuş sürüsü vb. gibi doğadaki canlıların sosyal davranışlarından ilham alır. Yiyecek arama gibi ortak bir hedefe ulaşmak için birlikte çalışan, kendi kendine organize olmuş bir popülasyonu modellerken kolektif zekayı kullanırlar (Bindu ve Sabu 2020).

Bu çalışmada, öznitelik seçimi için bir hibrit yaklaşımı sunulmuştur. Tez çalışmasında, sunulan hibrit yaklaşımın ilk adımı olan filtre yöntemi için önerilen yöntem meta sezgisel algoritmalarından sürü zekası algoritması olan karınca koloni algoritmasıdır. Karınca koloni algoritmasında sezgisel uzaklıkların hesaplanması için kosinüs benzerlik yöntemi kullanılmıştır. ACO ile tespit edilen rotalar ikinci adımda kullanılmıştır. İkinci adım için önerilen sınıflandırma yöntemleri K en yakın komşuluk ve destek vektör makineleri sınıflandırma yöntemleridir.

Bu çalışmada, önerilen algoritma python üzerinde kodlanmıştır. Kod, hibrit yaklaşımda olduğu gibi iki parçadan oluşmuştur. İlk parçada karınca koloni algoritması çalıştırılarak öznelik rotaları oluşturulmuştur. İkinci parçada rotalar kullanılarak sınıflandırma yapılmıştır. Kodun iki parça olmasının avantajı sınıflandırma yöntemlerinin bazı veri setlerinde, weka programı ile elde edilen sonuçların karşılaştırılmasına olanak sağlamasıdır.

## 2. KAYNAK ÖZETLERİ ve KURAMSAL TEMELLER

Bu bölümde tanımlardan, genel bilgilerden ve araştırılan kaynaklardan bahsedilmiştir.

### 2.1 Kaynak Özetleri

Çalışmalarında karınca kolonisi algoritması kullanarak öznitelik seçimi sınıflandırma yapan ve literatürde öne çıkan bazı makaleler ile, diğer meta-sezgisel yöntemler kullanarak öznitelik seçimi yapan çalışmalar, yayınlanma yılına göre aşağıdaki gibi derlenmiştir.

Al-Ani (2005) çalışmasında karınca koloni algoritması ile öznitelik seçimi yöntemi önermiştir. Sezgisel uzaklıkların hesaplanması için karşılıklı bilgiye dayalı ölçü adlı yaklaşım kullanmıştır. Feromon güncellemelerini ortalama kare hatasından oluşan bir formülle yapmışlardır.

Sivagaminathan ve Ramakrishnan (2007) çalışmalarında öznitelik seçimi için karınca koloni algoritması ve yapay sinir ağlarından oluşan bir yöntem önermişlerdir. Karınca kolonisi feromon güncellemeleri yapay sinir ağlarından gelen geri bildirimle yapılır.

Kanan ve Faez (2008) çalışmalarında karınca koloni algoritmasını yüz tanıma sistemlerinde öznitelik seçimi için kullanmışlardır. Karıncaların buldukları rotaların değerlendirme kriteri olarak seçilen ortalama hata karesini azaltmasına göre öznitelik seçim yöntemi uygulanmıştır.

Deriche (2009) çalışmasında karınca koloni algoritması ile öznitelik seçiminde farklı yerel ölçütlerinin etkisini değerlendirmiştir. Fisher kriteri ve karşılıklı bilgiye dayalı ölçü ölçütlerini ortalama kare hatasına göre karşılaştırmıştır.

Aghdam, Ghasem-Aghaee ve Basiri (2009) çalışmalarında karınca koloni algoritmasını metin madenciliğinde öznitelik seçimi için kullanmıştır. Metin sınıflandırmasının önemli bir sorunu, metnin yüksek boyutluluğudur. Öznitelik seçimi ile yüksek boyutluluk azaltılması hedeflenmiştir. Karınca koloni algoritmasında seçilen değerlendirme kriteri ortalama hata karesi kullanılmıştır.

Min ve Fangfang (2010) çalışmalarında filtre ve sarmalayıcı yöntemlerden oluşan bir hibrit yöntemi önermişlerdir. Filtre yöntemi için Korelatif Aile Seçimi, Relief Sınıf Ayrılabilirliği, Mahalanobis Uzaklığı, Çok Değişkenli Korelasyon Katsayısı ve Karşılıklı Bilgi yöntemlerini kullanmışlardır. Öznitelikler belirtilen altı kritere göre ağırlıklandırma değeri verilerek seçilmiştir. Sarmalayıcı yöntemler için Uyarlamalı Genetik Algoritma, Kaotik İkili Parçacık Sürü Optimizasyon ve Klonal Seçim Algoritması olmak üzere üç algoritma denenmiştir. Global optimizasyonda daha başarılı olduğu için klonal seçim algoritması seçilmiştir.

Ali ve Shahzad (2012) çalışmalarında karınca koloni algoritması ile öznitelik alt kümesi seçimi yapmışlardır. Simetrik belirsizlik isimli bilgi kazancını ve entropiden oluşan bir formül ile öznitelikler arası sezgisel uzaklık hesaplamışlardır.

Wald, Khoshgoftaar ve Napolitano (2013) çalışmalarında sarmalayıcı yöntemler ile öznitelik seçimi problemlerinde öznitelik seçimi ve modelde kullanılmak üzere iki farklı sınıflandırma yöntemi gerektiğini önermişlerdir.

Saraç ve Özel (2014) çalışmalarında web sayfaları sınıflandırma problemlerinde öznitelik seçimi için karınca koloni algoritması kullanılmasını önermişlerdir. Sezgisel uzaklıklar için eğitim setindeki özniteliliğin doküman frekansını kullanmışlardır.

Hamed, Dara ve Kremer (2014) çalışmalarında öznitelik seçimi problemi için gömülü öznitelik seçimi yöntemi önermişlerdir. Çalışmada destek vektör makineleri kullanılmıştır ve yöntem, boş bir öznitelik kümesinden ileriye doğru öznitelik ekleyerek çalışır.

Tabakhi, Moradi, ve Akhlaghian (2014) çalışmalarında öznitelik seçimi problemleri için karınca koloni algoritması ile hibrit bir yaklaşım önermiştir. İlk adımda karınca koloni algoritması öznitelikler arası kosinüs benzerlik yöntemi ile bulunan uzaklıklara göre denetimsiz öğrenme olarak çalıştırılmıştır. İkinci adımda sınıflandırma algoritmalarında karınca koloni algoritmasından gelen feromon miktarları kullanılmıştır. Öznitelikler için feromon miktarları ağırlıklandırma ölçütü yerine geçmektedir.



Kashef ve Nezamabadi-pour (2015) çalışmalarında öznitelik seçimi yöntemi için karınca koloni algoritması ile ikili seçim yöntemi önermişlerdir. İkili seçim yöntemi bir öznitelikliğin varlığını veya yokluğunu temsil eder. Karıncalar tüm öznitelikleri ziyaret etmesi gerekirken düğümleri seçmek için karınca kolonisi algoritması kullanmışlardır. İkili seçim yolları için farklı istatistiksel ölçüleri kullanmışlardır.

Mohammed, Deeba, Bui, ve Wahid (2016), çalışmalarında endoskopi görüntülerini kanayan, kanamayan ve bilgi vermeyen bölgeler olarak sınıflandırma problemlerinin boyut karmaşıklığını azaltmak için karınca koloni algoritmasına dayanan bir öznitelik seçimi yöntemi önermiştir. Değerlendirme işlevinin hatalı sınıflandırma sonucu yüksek maliyet eklemek istedikleri için duyarlılık ve doğruluk oranlarını kullanarak yeni bir değerlendirme işlevi önermişlerdir.

Fahrudin, Syarif, ve Barakbah (2016) çalışmalarında mikro dizilerde öznitelik seçimi için karınca koloni algoritması önermişlerdir. Mikro diziler belirli bir yerde bir nükleik asit dizisini (genetik bilgi içeren nükleotid zincirleri) bulmak için kullanılan ve aynı anda binlerce numuneyi analiz etmek için kullanılabilen bir teknoloji olup, nükleotid asitler DNA ve RNA olmak üzere her canlıda bulunmaktadır. Mikro dizi veri seti ayrıca algoritmadaki hız ve doğruluğu etkileyen birçok alakasız ve gereksiz öznitelik içermektedir.

Shunmugapriya ve Kanmani (2017) çalışmalarında karınca koloni ve yapay arı koloni algoritmaların özelliklerini birleştirerek melez bir hibrit algoritma önermişlerdir. Arılar, karıncaların bulunduğu öznitelik alt kümelerini kullanırlar.

Manosij, Ritam, Sarkar, ve Abraham (2020) çalışmalarında öznitelik seçimi problemleri için karınca koloni algoritması kullanılarak sarmalayıcı yaklaşım önermişlerdir. Karınca koloni algoritması için kosinüs benzerlik yöntemi seçilmiştir. Karınca kolonisinin ürettiği alt kümeleri, doğruluk oranı ve seçilmemiş öznitelik sayısı ile hesaplanmış bir değerlendirme fonksiyonuna göre değerlendirilmiştir..

Paniri, Dowlatsahi, ve Nezamabadi-pour (2020) çalışmalarında çoklu etiketli sınıflandırma problemleri için karınca koloni algoritması ile öznitelik seçim yöntemi önermişlerdir. Algoritmada karınca koloni algoritmasında pearson korelasyon katsayısı

ve kosinüs benzerlik yöntemi kullanmışlardır. Çoklu sınıflandırma problemi olduğu için öznitelikler arası ve sınıf etiketleri arasında ilişkiler dikkate alınmıştır.

## 2.2 Kuramsal Temeller

Bu bölümde tanımlardan ve genel bilgilerden bahsedilmiştir.

### 2.2.1. Öznitelik seçim teknikleri

Filtre yöntemi, tek değişkenli istatistiklerle öznitelik seçimi yapar ve diğer yöntemlerden az hesaplama maliyetine sahiptir. Bu yüzden daha hızlı çalışır (Gheyas ve Smith 2010). Bazı filtre teknikleri ve kısa açıklamaları aşağıdaki gibi derlenmiştir;

Bilgi entropisi, bir veri setinin içerdiği bilgi miktarıdır. Bilgi entropisi formülü denklem 2.1'de gösterilmiştir.

$$H = - \sum_i^C p(x) \log_2 p(x) \quad (2.1)$$

$$p(x) = \frac{x \text{ sınıfına ait veri sayısı}}{\text{toplam veri sayısı}} \quad (2.2)$$

- C = Sınıflar kümesi.

Denklem 2.1'de bir olayın sonucu kesin (0 veya 1) olduğu takdirde bilgi miktarı sıfır olacaktır (Cover ve Thomas 2006).

Bilgi kazanımı (IG), veri setini rastgele bölerek entropi değerlerine bakar. Bütün veri setinin entropisinden çıkartılarak bilgi kazanımı bulur (Tabakhi, Moradi, ve Akhlaghian 2014).

Kazanç oranı, öznitelik örüntülerine göre bölünmesinin kazanç oranını gösterir. Bölünen kısmın entropisini veri setinin entropisine böler (Tabakhi ve diğerleri, 2014).

Terim varyansı (TV), en basit tek değişkenli değerlendirmedir. Veri setindeki değerlerin ortalamadan ne kadar saptığını açıklar. Ortalama ve değerler arasındaki uzaklıkların karelerinin ortalamasıdır (Tabakhi ve diğerleri, 2014).

Rastgele alt uzay yöntemi (RSM), bu yöntem, özgün öznitelik uzayının rastgele seçilmiş bir alt uzayına belirlenmiş sınıflandırma yöntemi uygular. Özniteliklerin büyük bir bölümünü değerlendirmek için seçim işlemi birkaç kez tekrarlanır ve son olarak sonuçlara göre seçilen öznitelikler olarak birleştirilir (Tabakhi ve diğerleri, 2014).

Uygunluk-yedeklilik öznitelik seçimi (RRFS), hem denetimli hem de denetimsiz modlarda çalışabilen, uygunluk ve artıklık analizlerine dayalı verimli bir öznitelik seçme tekniğidir. Bu yöntemde, verilen bir kritere göre ilk öznitelik seçilecek ve daha sonra her yinelemede, son seçilen öznitelik benzerliği önceden tanımlanmış bir eşikten küçükse bir öznitelik seçilecektir (Tabakhi ve diğerleri, 2014).

Karşılıklı korelasyon (MC), iki öznitelik arasındaki bağımlılığı hesaplayan çok değişkenli öznitelik seçim yöntemidir. Bu yöntemde, her iterasyonda en yüksek ortalama korelasyon değerine sahip öznitelik kaldırılır (Tabakhi ve diğerleri, 2014).

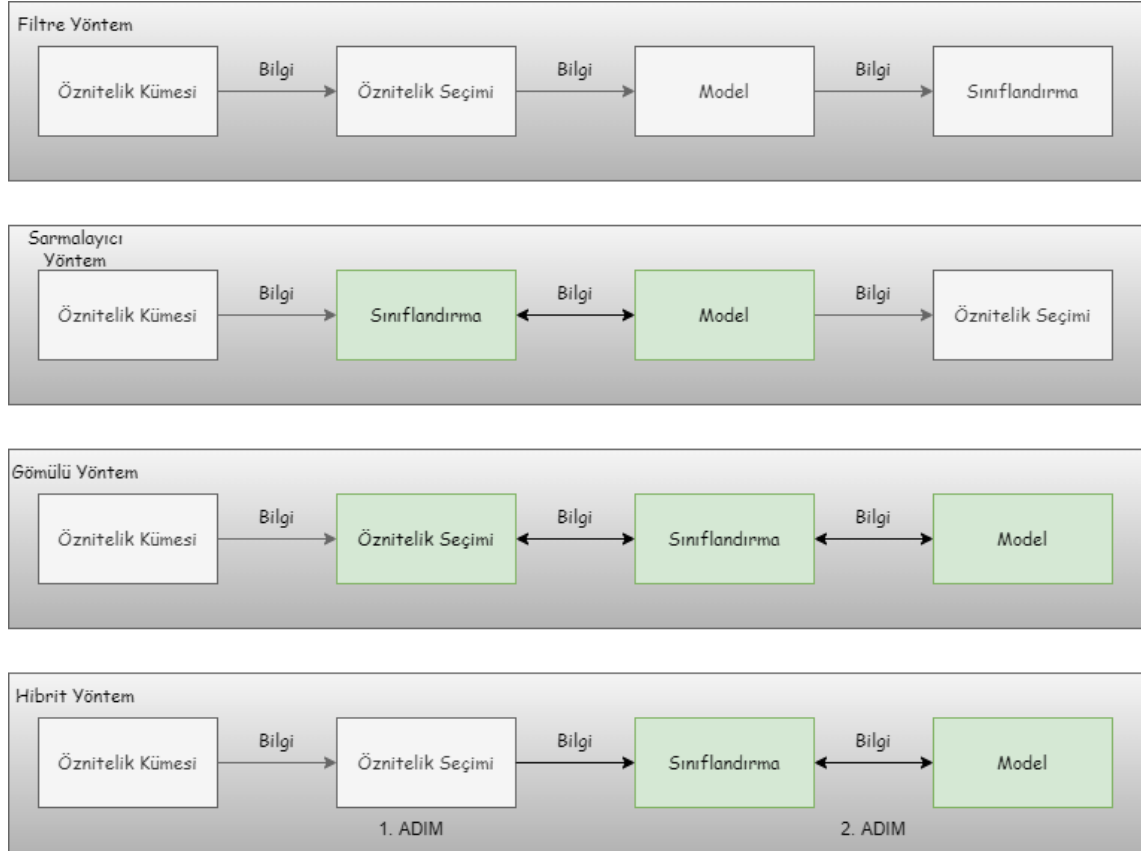
Laplacian Skoru (LS), her öznitelik için yerelliğini koruma gücünü yansıtacak şekilde hesaplanır. Birbirine yakın olan iki veri noktasının muhtemelen aynı konu ile ilgili olduğu gözlemine dayanmaktadır. Yerel geometrik yapıyı modellemek için en yakın komşu grafiği oluşturuyoruz. Bu grafik yapısına uyan özellikleri arar (He, Cai, D ve Niyogi 2005).

Sarmalayıcı yöntemler, özniteliklerin alt kümelerini değerlendirmek için bir sınıflandırıcıdan gelen geri bildirimini kullanır. Bu sınıflandırıcı genellikle nihai modeli oluşturmak için kullanılacak olanla aynıdır. Destek vektör makineleri, karar ağacı ve Naive bayes algoritmaları örnek olarak verilebilir (Shanab, Khoshgoftaar, ve Wald 2014).

Gömülü yöntemler, aynı anda modeli eğitir ve öznitelik seçimini gerçekleştirir. Yani öğrenme kısmı ve öznitelik seçme kısmı aynı anda gerçekleştirilir. Bu nedenle, tahmin modeli ile etkileşime girdiklerinden filtre stratejilerinden daha “akıllı” yöntemler ve bir öznitelik alt kümesi her seçildiğinde öğrenme modelinin eğitilmesi gerekmediğinden sarmalayıcı yöntemlerden daha hızlı olarak kabul edilebilirler (Jiménez-Cordero, Morales ve Pineda 2021). You, Liu, Li ve Chen (2012), Yılmaz Eroglu ve Kilic (2017), Maldonado ve López (2018), çalışmaları örnek olarak verilebilir.

Hibrit yöntemler, filtreleme yöntemleri ile başlangıçta öznitelik seçimi kullanıldığı, ardından sarmalayıcı veya gömülü yaklaşımlarının sınıflandırma için kullanıldığı yöntemlerdir (Nagarajan ve Dhinesh Babu 2021).

Öznitelik seçimi yöntemlerin bilgi akışı yönleri Şekil 2.1’de gösterilmiştir.



**Şekil 2.1.** Öznitelik seçimi yöntemlerin bilgi akışı

### 2.2.2 Temel makine öğrenme yöntemleri

Takviyeli öğrenmede, temsilci bir dizi takviyeden öğrenir. Takviye doğru sınıflandırmayı veya yanlış sınıflandırmaya göre iyi veya kötü ödül değeri verilir.

Denetimli öğrenmede, veri setindeki girdi ve çıktı çiftlerini gözlemler ve girdiden çıktıya eşleşen bir işlevi öğrenir. Çıktının olduğu veriler etiketli verilerdir.

Denetimsiz öğrenmede, girdi değerleri vardır fakat çıktı değerleri yoktur. Veriler arasındaki örüntüleri öğrenir. Çıktının olmadığı veriler etiketsiz verilerdir.

Yarı denetimli öğrenme de, veri setinde hem etiketli hem etiketsiz veriler bulunur (Russell, Norvig ve Davis 2010).

En çok bilinen öğrenme algoritmaları Çizelge 2.1’de gösterilmiştir.

**Çizelge 2.1.** Bazı makine öğrenmesi algoritmaları

	<b>DENETİMSİZ ÖĞRENME</b>	<b>DENETİMLİ ÖĞRENME</b>
<b>SÜREKLİ VERİ</b>	K-ortalama	Regresyon
	Temel bileşen analizi(PCA)	Karar ağaçları
	Tekil değer ayrıştırması(SVD)	Rastgele orman
<b>KATEGORİK VERİ</b>	Apriori	K en yakın komşuluk
		Destek vektör makineleri
	Sık örüntü keşfi(FP-growth)	Lojistik regresyon
		Naive bayes

Tez çalışmasında kullanılacak olan algoritmalar sonraki bölümlerde detaylandırılmıştır.

### 2.2.3 Sınıflandırma metrikleri

Bu bölümdeki bilgilerin önemli bir kısmı Naser ve Alavi (2021) tarafından hazırlanan makaleden derlenmiştir. Sınıflandırma modellerinin performans değerlendirmesinde kullanılan ölçütler karmaşıklık matrisinden hesaplanmıştır. Karmaşıklık matrisi Şekil 2.2’de gösterilmiştir.

	Tahmin Edilen Değer	
Gerçek Değer	Gerçek Pozitif	Gerçek Negatif
	Yanlış Pozitif	Yanlış Negatif

**Şekil 2.2.** Karmaşıklık matrisi

- Gerçek Pozitif (TP): Gerçek değeri 1 ve tahmin edilen değer de 1 olduğu örneklerdir.
- Gerçek Negatifler (TN): Gerçek değeri 0 ve tahmin edilen değer de 0 olduğu örneklerdir.

- Yanlış Pozitifler (FP): Gerçek değeri 0 ancak tahmin edilen değerin 1 olduğu örneklerdir.
- Yanlış Negatifler (FN): Gerçek değeri 1 ancak tahmin edilen değerin 0 olduğu örneklerdir.

Karmaşıklık matrisinin sütun ve satır sayısı verinin sınıf sayısına göre değişir. Şekil 2.2’te iki sınıf vardır. Sınıf sayısı değişirse de karmaşıklık matrisinden hesaplanan metrik değerler aşağıda anlatılan hesaplama yöntemleri ile hesaplanabilir.

Doğruluk Oranı (Accuracy Rate), sınıflandırma sonucunun hangi oranda doğru tahmin ettiğinin bir ölçüsüdür. Hesaplanma yöntemi denklem 2.3’de gösterilmiştir.

$$\text{Doğruluk Oranı} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.3)$$

Yanlış Sınıflandırma Oranı (Misclassification Rate), sınıflandırma sonucunun hangi oranda yanlış tahmin ettiğinin bir ölçüsüdür. Hata Oranı olarak da bilinir (Error Rate). Hesaplanma yöntemi denklem 2.4’de gösterilmiştir. Doğruluk oranı ile arasındaki ilişki denklem 2.5’de gösterilmiştir.

$$\text{Yanlış Sınıflandırma Oranı} = \frac{FP+FN}{TP+TN+FP+FN} \quad (2.4)$$

$$\text{Yanlış Sınıflandırma Oranı} = 1 - \text{Doğruluk Oranı} \quad (2.5)$$

Gerçek Pozitif Değerlerin Oranı (True Positive Rate), sınıflandırma sonucunun hangi oranda gerçek pozitif değeri doğru tahmin ettiğinin bir ölçüsüdür. Hassasiyet, İsbet Oranı veya Hatırlama olarak da bilinir. (Sensitivity, Hit Rate or Recall) Mümkün olduğu kadar yüksek olmalıdır. Hesaplanma yöntemi denklem 2.6’te gösterilmiştir.

$$\text{Gerçek Pozitif Değerler Oranı} = \frac{TP}{TP+FN} \quad (2.6)$$

Gerçek Negatif Değerlerin Oranı (True Negative Rate), sınıflandırma sonucunun hangi oranda gerçek negatif değeri doğru tahmin ettiğinin bir ölçüsüdür. Özgüllük veya Seçicilik olarak da bilinir. (Specificity or Selectivity) Hesaplanma yöntemi denklem 2.7’a gösterilmiştir.

$$\text{Gerçek Negatif Değerlerin Oranı} = \frac{TN}{TN+FP} \quad (2.7)$$

Yanlış Pozitif Değerlerin Oranı (False Positive Rate), sınıflandırma sonucunun hangi oranda gerçek değeri 0 olmasına karşın 1 olarak tahmin edilenlerin oranıdır. Yan Ürün olarak da bilinir. (Fall-out) Hesaplanma yöntemi denklem 2.8’de gösterilmiştir.

$$\text{Yanlış Pozitif Değerlerin Oranı} = \frac{FP}{TN+FP} \quad (2.8)$$

Yanlış Negatif Değerlerin Oranı (False Negative Rate), sınıflandırma sonucunun hangi oranda gerçek değeri 1 olmasına karşın 0 olarak tahmin edilenlerin oranıdır. Kayıp oranı olarak da bilinir. (Miss Rate) Hesaplanma yöntemi denklem 2.9’de gösterilmiştir.

$$\text{Yanlış Negatif Değerlerin Oranı} = \frac{FN}{TP+FN} \quad (2.9)$$

Hassasiyet (Precision), tüm sınıflardan, doğru olarak ne kadar tahmin edildiğinin bir ölçüsüdür. Mümkün olduğu kadar yüksek olmalıdır. Pozitif Tahmin Edici Değer olarak da bilinir. (Positive Predictive Value) Hesaplanma yöntemi denklem 2.10’da gösterilmiştir.

$$\text{Hassasiyet} = \frac{TP}{TP+FP} \quad (2.10)$$

F Puanı (F Score), sınıflandırma sonucunun gerçek pozitif değerlerin oranının (recall) ve hassasiyetin (precision) harmonik ortalamasıdır. Sınıflandırıcının ne kadar iyi performans gösterdiğinin bir ölçüsüdür ve sınıflandırıcıları karşılaştırmakta sıklıkla kullanılır. Hesaplanma yöntemi denklem 2.11’de gösterilmiştir.

$$F \text{ Puanı} = \frac{2 \times \text{Hassasiyet} \times \text{Gerçek Pozitif Değerlerin Oranı}}{\text{Hassasiyet} + \text{Gerçek Pozitif Değerlerin Oranı}} \quad (2.11)$$

ROC Eğrisi (ROC Curve), sınıflandırma sonucunun tüm olası değerler üzerinde performansını özetlemek için kullanılan bir grafiktir. Belirli bir sınıfa gözlem atanması eşliğini değiştirdiğinizde Gerçek Pozitif Değerlerin Oranına (Hassasiyet) (x eksen) karşı Yanlış Pozitif Değerlerin Oranını (Özgüllük) (y eksen) çizerek oluşturulur. ROC Eğrisi, Hassasiyet / Özgüllük (Sensitivity / Specificity) raporu oluşturmaya yarar. ROC eğrisinin altındaki alan (Area Under Curve (AUC)), bir parametrenin iki sınıf arasında ne kadar iyi ayırt edilebileceğinin bir ölçüsüdür.

### 3. MATERYAL ve YÖNTEM

Bu bölümde kullanılan veri setleri, algoritmalar ve geliştirilen yöntem hakkında bilgi verilmiştir.

#### 3.1 Materyal

Bu bölümde kullanılan veri setleri, algoritmalar hakkında bilgi verilmiştir.

##### 3.1.1 Veri setleri

Bu bölümde, çalışmada kullanılan veri setleri açıklanmıştır. Veri setlerine ait bilgiler Çizelge 3.1.'de gösterilmiştir. Veri setleri açık kaynak kodlu olup, UCI makine öğrenme deposundan alınmıştır (Dua ve Graff 2019).

**Çizelge 3.1.** Veri seti bilgileri

Veriseti	Öznitelik Sayısı	Sınıflar	Veri Sayısı
Wine	13	3	178
Hepatitis	19	2	155
WDBC	30	2	569
Ionosphere	34	2	351
Dermatology	34	6	366
SpamBase	57	2	4601
Arrhythmia	279	16	452
Madelon	500	2	4400

Wine veri setinde, şarapların kimyasal analiz sonuçları vardır. Üç farklı sınıf vardır. Analize göre şarapların hangi sınıfta olduğu tahmin edilir. Hepatitis veri setinde ölür ve yaşar şeklinde iki farklı sınıf vardır. WDBC veri seti, bir meme kitlesinin ince iğne aspirasyonun sayısallaştırılmış görüntüsünden hesaplanır. Görüntüde bulunan hücre çekirdeklerinin özelliklerini tanımlarlar. Veri setinde kötü huylu veya iyi huylu şeklinde iki farklı sınıf vardır. Ionosphere veri seti, Goose Bay tarafından Labrador'daki bir sistem tarafından toplanmıştır. İki farklı sınıfı vardır; iyi türü iyonosferden yansır, kötü türü iyonosferden geçer. Dermatology veri setinin amacı Erihemato-Skuamöz Hastalığın



tipini belirlemektir. Sedef hastalığı, seboreik dermatit, liken planus, pityriasis rosea, kronik dermatit ve pityriasis rubra pilaris olmak üzere 6 sınıfı vardır. Spambase veri seti, istenmeyen e-postaları ayıran 2 sınıflı bir veritabanıdır. Arrhythmia veri seti, Amaç kardiyak aritmi varlığını ve yokluğunu ayırt etmek ve 16 gruptan birinde sınıflandırmaktadır. Madelon veri seti, ikili sınıflandırma problemleri için oluşturulmuş yapay bir veri setidir.

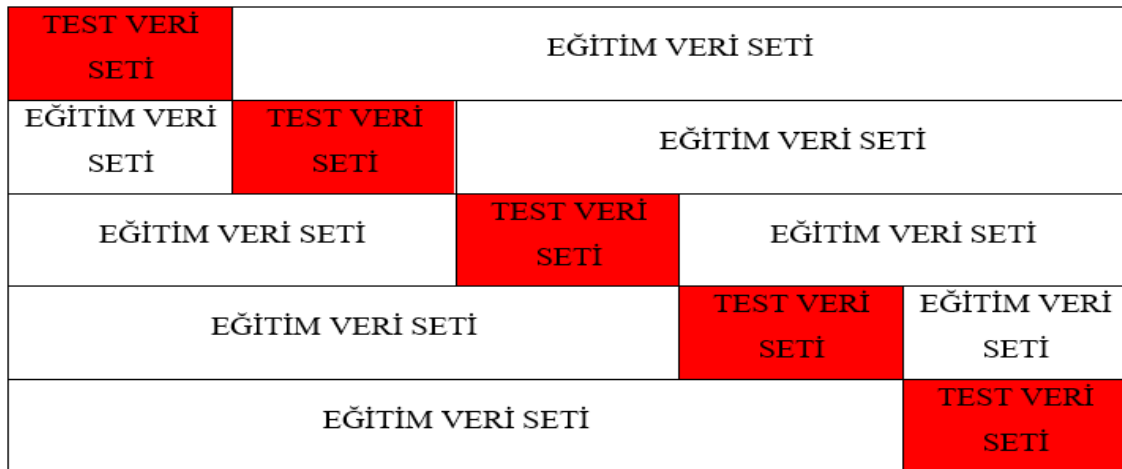
Veri setlerinde seçimindeki kriterler öznitelik sayısı, boyut sayısı ve veri sayısı olmuştur. Öznitelik sayısı ile farklı boyutlardaki, veri sayısı ile farklı büyüklükteki ve sınıf sayısı ile farklı sınıf büyüklüklerindeki önerilen yaklaşımın performansını gözlemleyebilmek için küçük orta büyük veri setleri seçilmiştir.

Veri setlerinin sınıflandırma algoritma için eğitim ve test veri seti olarak bölünmesi gerekir. Veri setleri istenilen oranlarda bölünebilir. Örnek olarak %80 eğitim veri seti %20 test veri seti şeklinde bölünebilir (bkz. Şekil 3.1).



**Şekil 3.1.** %80 eğitim %20 test veri seti sembolik gösterimi

İkince ve sıklıkla kullanılan yöntem, k katlı çapraz doğrulama yöntemidir. Veri seti k eşit parçaya bölünür. Sırası ile her parça test veri seti olarak seçilir ve geri kalan parçalar eğitim seti olarak atanır. Bu yöntem ile verinin tamamı sınıflandırma için kullanılabilir. Şekil 3.2’de 5 katlı çapraz doğrulama sembolik olarak gösterilmiştir.



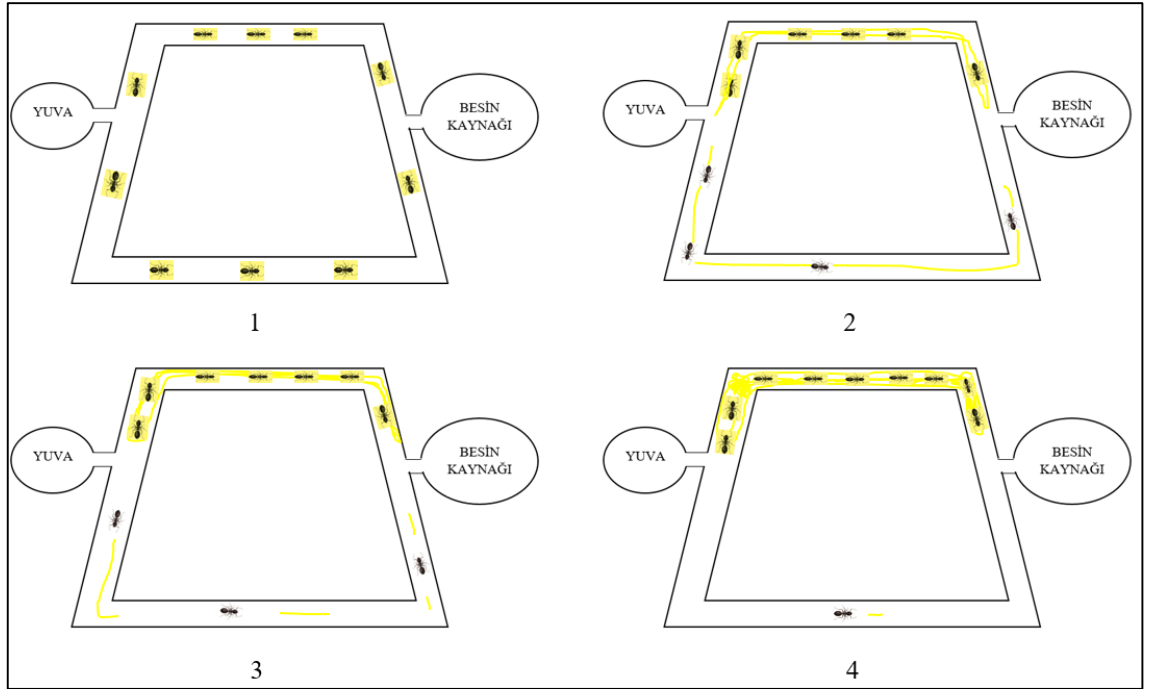
**Şekil 3.2.** 5 katlı çapraz doğrulama sembolik gösterimi

### 3.1.2. Karınca koloni algoritması

Karınca koloni algoritması, Dorigo (1992) tarafından önerilmiştir. Algoritma, gerçek karıncaların yemek bulma davranışları üzerine oluşturulmuştur.

Karıncalar, sürü halinde yaşarlar. Sürü olarak bireyler kendi aralarında bilgi alışverişinde bulunurlar. Birçok karınca türünde, bir besin kaynağına gidip gelen karıncalar geçtikleri yere feromon adı verilen bir madde bırakır. Diğer karıncalar, feromon varlığını algılar ve feromon konsantrasyonunun daha yüksek olduğu yolları takip etme eğilimindedir. Bu mekanizma sayesinde karıncalar, yiyecekleri yuvalarına oldukça etkili bir şekilde taşıyabilirler (Deneubourg, Aron, Goss ve Pasteels 1990).

Gerçek karıncaların yuva besin kaynağı arasında davranışlarının sembolik olarak gösterimi Şekil 3.3'te gösterilmiştir.



**Şekil 3.3.** Gerçek karınca davranışlarının sembolik gösterimi

Algoritma bu davranışı taklit eder. Optimizasyon için çözüm yollarını çözümün uygunluğuna göre feromon ile işaretler. İterasyonlar devam ettikçe daha iyi çözüme giden yollarda feromon miktarı artar. Buna bağlı olarak yolların seçim şansı artar (Dorigo, Birattari ve Stutzle 2006).

Karınca koloni algoritması parametreleri:

- Karınca Sayısı: Algoritmada kullanılan karınca sayısı.
- İterasyon Sayısı: Algoritma döngü sayısı.
- Alfa( $\alpha$ ): Feromon oranının önemi.
- Beta( $\beta$ ): Sezgisel uzaklık oranının önemi.
- Buharlaşma Katsayısı( $\rho$ ): Her iterasyonda azalan feromon oranı.

Karınca Koloni Algoritması Seçim Formülü:

$$P_{ij} = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{u \in J_k} \tau_{iu}^\alpha \eta_{iu}^\beta} & \text{eğer } j \in J_k \\ 0 & \text{diğer durumlarda} \end{cases} \quad (3.1)$$

- $\tau_{ij}$ : i ve j arasında bulunan feromon miktarı.
- $\eta_{ij}$ : i ve j arası sezgisel uzaklık. ( $1/d_{ij}$ ,  $d_{ij}$ : i ve j arası uzaklık)
- $J_k$ : Gidilebilecek komşuluklar kümesi.

Feromon güncelleme formülü:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3.2)$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{1}{L_k} & k \text{ karıncası bu rotada } i-j \text{ yolunu kullandı ise} \\ 0 & \text{diğer durumlarda} \end{cases} \quad (3.3)$$

- $L_k$ : k karıncasının kullandığı rota uzunluğu.

Karıncalar gidecekleri noktaya denklem 3.1'e göre karar verirler. Karıncalar rotalarını tamamlayınca denklem 3.2 ve denklem 3.3'e göre feromon güncellemeleri yapılır.

Karınca koloni algoritması sözde kodu (pseudo code):

Input: Karınca koloni algoritması parametreleri.

Output: Feromon miktarının en fazla olduğu rota.

1: begin algorithm

2: Karıncalar rastgele noktalara yerleştirilir.

3: while durdurma koşulları sağlanmadığında do

- 4: Her karınca için rotayı hesapla
- 5: Bulunan en iyi rota için feromon miktarlarını artır
- 6: Bütün yollarda feromon buharlaştır.
- 7: end while

### 3.1.3 K en yakın komşuluk algoritması

K en yakın komşuluk algoritmaları Fix ve Hodges (1951) tarafından geliştirilmiştir. Cover ve Hart (1967) tarafından genişletilmiştir. K en yakın komşuluk algoritması sınıflandırma ve regresyon problemlerini çözmek için kullanılabilir. Denetimli makine öğrenme algoritmasıdır. Denetimli öğrenme, sınıfların etiketlerinin belli olduğu durumlardır.

K en yakın komşuluk sınıflandırıcı kullanarak, eğitim modellerinden gelen bilgiyi kullanır. Test verileri, en yakın k komşu arasında çoğunluk oylaması kullanılarak belirli bir sınıfa sınıflandırılır. Ayrıca, eşit olmayan dağılım kusurlarının da üstesinden gelir (Malini Devi, Seetha, ve Sunitha 2016).

K en yakın komşuluk algoritması sözde kodu:

- Sınıflandırma aşamasında, k komşu sayısı kullanıcı tanımlı bir sabittir ve etiketlenmemiş bir veri ile veri kümesindeki tüm noktalar arasındaki mesafe hesaplanarak sınıflandırılır.
- Döngü, veri kümesindeki tüm noktalar tamamlanana kadar tekrarlanır.
- Elde edilen uzaklıklar sıralanır ve k değerine göre bu komşular grafiğe alınır.
- Komşu belirli bir sınıfta yer alıyorsa, o zaman gerçek pozitif olarak kabul edilir, aksi halde gerçek negatif olarak kabul edilir.
- Gerçek pozitif ve gerçek negatif değere bağlı olarak doğruluk oranı hesaplanır.

Tembel öğrenen algoritması olarak da adlandırılır. Çünkü eğitim kümesinden hemen öğrenmez, bunun yerine veri kümesini depolar ve sınıflandırma anında veri kümesi üzerinde bir işlem gerçekleştirir.

K en yakın komşuluk algoritması parametreleri:

- Komşuluk sayısı(k): En yakın kaç komşunun dikkate alınacağını belirleyen parametredir.
- Uzaklık formülü: Komşuluklar arası uzaklıkların hesaplanmasında kullanılan formüldür.

K en yakın komşuluk algoritmasında genelde kullanılan uzaklık formülleri:

- Minkowski Uzaklığı: Reel sayılar için tasarlanmış bir vektördür. Mesafelerin uzunluğu bir vektör olarak temsil edilmeli ve pozitif olmalıdır. Hesaplanma formülü denklem 3.4'te gösterilmiştir. Denklemdaki x ve y vektörlerin kartezyen değerleridir.

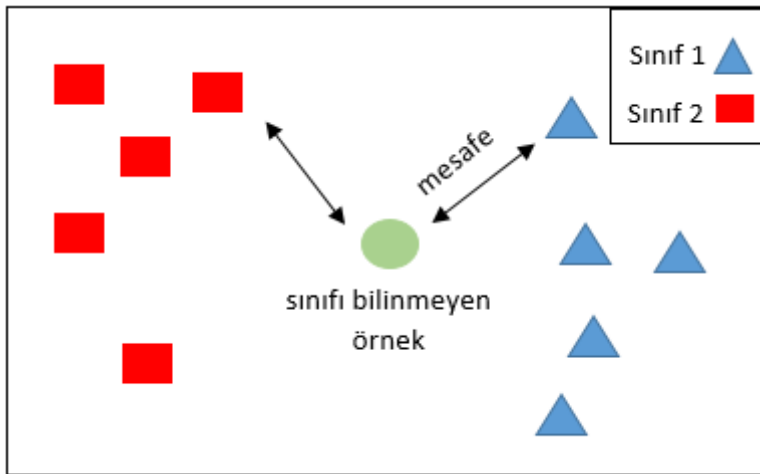
$$d = (\sum_{i=1}^n |x_i - y_i|^p)^{1/p} \quad (3.4)$$

- Manhattan Uzaklığı: Minkowski uzaklığının p=1 değeri için özel halidir. Kartezyen koordinatlarının mutlak farklarının toplamıdır. Hesaplanma formülü denklem 3.5'te gösterilmiştir.

$$d = \sum_{i=1}^n |x_i - y_i| \quad (3.5)$$

- Öklid Uzaklığı: Minkowski uzaklığının p=2 değeri için özel halidir. Düzlem üzerinde iki nokta arasındaki düz çizgi mesafesinin bir ölçüsüdür. Hesaplanma formülü denklem 3.6'te gösterilmiştir.

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.6)$$



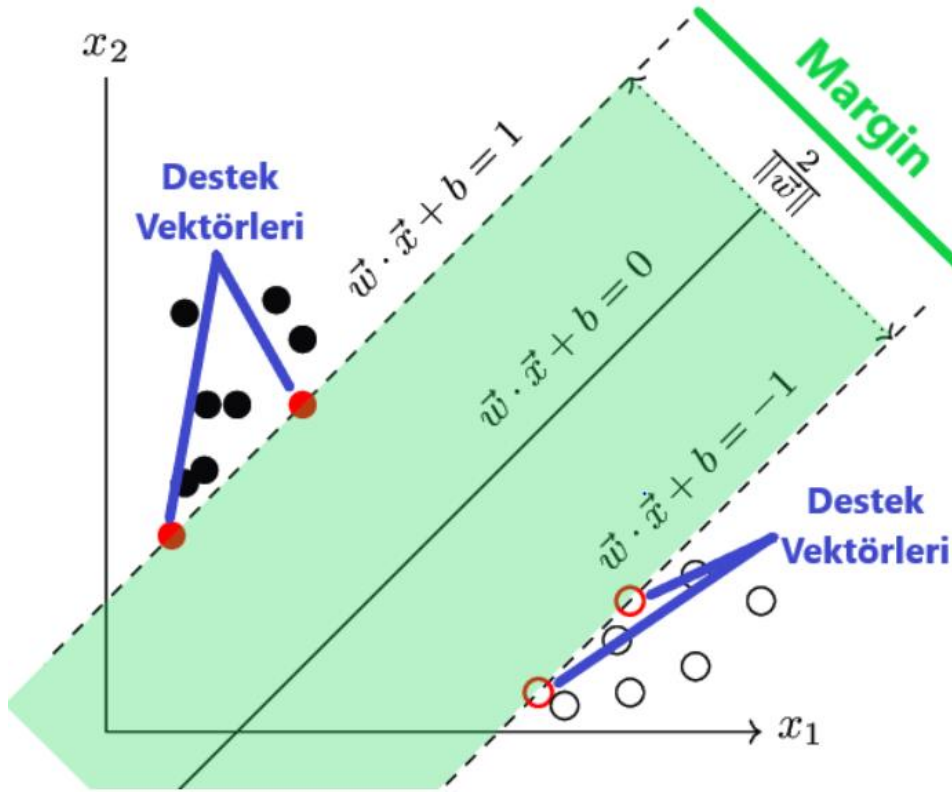
Şekil 3.4. K en yakın komşuluk algoritmasının gösterimi

Şekil 3.4'teki örnekte,  $k=1$  ise sınıfı bilinmeyen örnek en yakın mesafedeki sınıf etiketi ile etiketlenir.

### 3.1.4. Destek vektör makineleri

Boser, Guyon ve Vapnik (1992) tarafından destek vektör makineleri, örüntü tanıma ve sınıflandırma problemlerini çözümü için geliştirilmiştir.

Destek vektör makineleri denetimli öğrenme algoritmalarıdır. Bir düzlem üzerindeki noktaları ayırmak için bir doğru belirler. Bu doğru ile farklı sınıfları ayırarak aralarındaki mesafeyi maksimum uzaklıkta olmasını amaçlar (Yang, Awan ve Vall-Llosera 2019).



Şekil 3.5. Destek vektör makineleri gösterimi (Yang ve diğerleri 2019, Şekil 1)

Şekil 3.5'te beyaz noktalar ve siyah noktalar olmak üzere iki sınıf vardır. Tolerans(margin) ne kadar geniş ise sınıflar o kadar iyi ayrıştırılır.

$$\hat{y} = \begin{cases} 0 & \text{eğer } w^T \cdot x + b < 0, \\ 1 & \text{eğer } w^T \cdot x + b \geq 0 \end{cases} \quad (3.7)$$

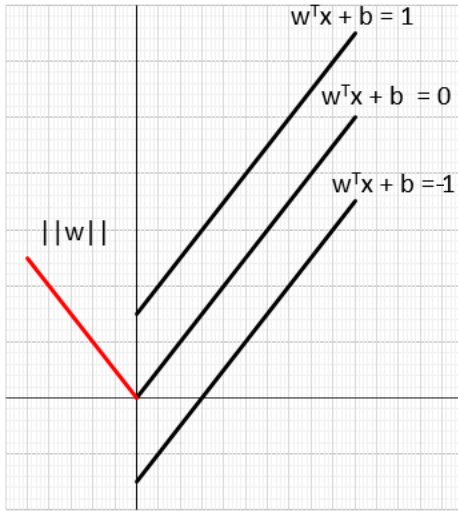
- **w**: ağırlık vektörü.
- **x**: Girdi vektörü.
- **b**: Sapma miktarı.

Denklem 3.7’de çıkan sonuç 0’dan küçük ise beyaz noktalara yakın olacaktır. 0’dan büyük ise siyah noktalara yakın olacaktır.

Denklem 3.7, denklem 3.8’te bulunan tek eşitsizlikte birleştirilebilir.

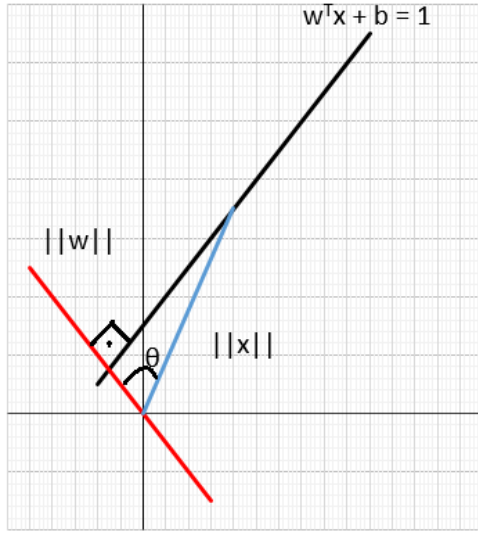
$$\forall_i \text{ için } y_i(\langle w^T, x_i \rangle + b) - 1 \geq 0 \quad (3.8)$$

Algoritmadaki temel mantık iki düzlem arasında bulunan mesafeyi en uzak tutmaktır. Mesafeyi hesaplayabilmek için problemi iki boyutta ele alınırsa düzlem yerine doğru kullanılabilir.



**Şekil 3.6.**  $\|w\|$  vektörünün doğruları dik kesme grafiği

Orijin noktasından geçen  $w^T x + b = 0$  doğrusu tanımlarırsa Şekil 3.6’daki gibi doğruya dik olarak  $\|w\|$  vektörü tanımlanabilir.  $\|w\|$  vektörü aynı zamanda  $w^T x + b = 0$  doğrusuna paralel olan  $w^T x + b = 1$  ve  $w^T x + b = -1$  doğrularına diktir.



**Şekil 3.7.** Doğrular ve  $\|w\|$  vektörü arası  $\theta$  açısının gösterim grafiği

Şekil 3.7'deki gibi  $w^T x + b = 1$  doğrusu üzerinde bir  $a$  noktası alınıp orjinden  $a$  noktasına doğru çizilirse  $\|w\|$  vektörü ve doğrular ile bir üçgen elde edilebilir. Şekil 3.7 gösterilen  $\theta$  açısının denklem 3.9'de gösterilen kosinüs değerinden iki doğru arasındaki uzaklık bağlantısı elde edilir. Aynı işlem  $w^T x + b = -1$  doğrusu içinde yapılır.

$$\cos(\theta) = \frac{w^T a}{\|w\| \|a\|} \quad (3.9)$$

Doğru denklemlerinden  $w^T a$  yerinde  $1-b$  yazılarak düzenlenir ise 1.denklem için 3.10 ve 2. Denklem için 3.11 elde edilir.

$$\|a\| \cos(\theta) = \frac{1-b}{\|w\|} \quad (3.10)$$

$$\|a\| \cos(\theta) = \frac{-1-b}{\|w\|} \quad (3.11)$$

Şekil 3.7 görüldüğü gibi aralarındaki fark bulmak için denklem 3.10 denklem 3.11 çıkartılarak denklem 3.12 elde edilir.

$$Tolerans = \frac{2}{\|w\|} \quad (3.12)$$

$$Tolerans = \frac{\|w\|}{2} \quad (3.13)$$

Denklem 3.12 amaç fonksiyonu olarak en büyüklebilir bir fonksiyon olur veya denklem 3.12 çarpmaya göre tersi alınarak denklem 3.13 elde edilip, amaç fonksiyonu en küçüklebilir bir fonksiyon olur. Denklem 3.13 türevlenebilir bir denklem olmadığı için



türevlenebilir bir denklem elde edebilmek için denklem 3.13 karesi alınarak denklem 3.14 elde edilir.

$$Tolerans = \frac{\|w\|^2}{2} \quad (3.14)$$

Tam ayrıştırılmayan verilerde amaç fonksiyonuna esneme katsayısı eklenir. Denklem 3.15 gösterilmiştir. Denklem 3.15'te  $\varepsilon$  aylak değişkendir ve  $c$  ceza ağırlık terimidir.

$$Tolerans = \frac{\|w\|^2}{2} + c \sum_{i=1}^n \varepsilon_i \quad (3.15)$$

Çekirdek dönüşümleri yapmak için amaç fonksiyonun yanına kısıtlarımızı eklenerek Lagrange fonksiyonu oluşturulur. Denklem 3.16 gösterilmiştir. Denklem 3.16'a  $\alpha$  Lagrange çarpanı,  $A$  aktif kısıtlar kümesini göstermektedir.

$$L(w, b, \alpha) = \frac{\|w\|^2}{2} - \sum_{i \in A} \alpha_i (y_i (w^T x_i + b) - 1) \quad (3.16)$$

$w$  göre türev alınır ve düzenlenirse ifade denklem 3.17'deki gibi olur.

$$L(w, b, \alpha) = \sum_{i \in A} \alpha_i - \frac{1}{2} \sum_{i \in A} \sum_{j \in A} \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (3.17)$$

Denklem 3.17'ye göre girdilerin iç çarpanlarına bağlı  $(x_i^T x_j)$  bir vektör elde edilir. Bu vektöre değiştirilerek çekirdek dönüşümleri yapılabilir. Örnek olarak polinom çekirdek için  $(1 + x_i^T x_j)^d$  dönüşümü yapılır (Metlek ve Kayaalp 2020).

Vektör destek makineleri parametreleri:

- $c$ : Ceza ağırlık terimidir. Büyüdükçe yanlış sınıflandırma için ceza miktarını büyütür. Bu nedenle tolerans aralığını daraltır.
- Kernel trick: Çekirdek dönüşümü yapar. Doğrusal olmayan sınıflandırmalar da yapılabilir.

### 3.2. Yöntem

Önerilen yöntem sınıflandırma problemlerinde işlem yükünün azaltılması amacı ile sınıflandırma performans göstergelerinin en iyilemeye çalışacak şekilde öznelik seçimini hedeflemektedir.

Karınca koloni algoritmasında sadece en iyi rotadaki yollara feromon ekler. Karınca koloni algoritmasının, öznitelikler arası sezgisel uzaklıklara ihtiyaç duymaktadır. Bu uzaklıkları bulmak için kosinüs benzerlik yöntemi kullanılmıştır. Kosinüs benzerlik yöntemi, matematiksel olarak vektörlerin iç çarpımının, normlarına bölümüdür.

Kosinüs benzerlik formülü 3.18 denklemde gösterildiği gibidir. A ve B vektörleri temsil etmektedir.

$$\text{Benzerlik}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i x B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (3.18)$$

Daha önce karınca koloni algoritmasında anlatıldığı gibi, iki nokta arası mesafe olan  $d_{ij}$  öznitelik seçimi uyarlamasında denklem 3.18’de gösterilen kosinüs benzerlik formülü ile hesaplanmıştır.

Algoritma temel olarak iki aşamadan oluşur. İlk aşamada Karınca koloni algoritması ile frekans sıraları oluşturulur. İkinci aşamada sınıflandırma algoritmaları kullanılır. Algoritmanın iş akış diyagramı Şekil 3.8’de gösterilmiştir.

1. Kosinüs benzerlik formülü ile özniteliklerin sezgisel uzaklıklarını hesapla.
2. Karınca koloni algoritmasını 5 defa çalıştır. Bulunan 5 rotayı tut.
3. Bulunan rotalarda ilk %80 kısmında gözükten öznitelikleri say.
4. Sayılan öznitelikleri artmayan sıraya göre diz. Öznitelik frekans listesi oluştur.
5. Öznitelikleri öznitelik frekans listesine göre ekleyerek sınıflandırma algoritmasını çalıştır.
6. En iyi sınıflandırma metriğini veren öznitelik sayısını, öznitelikleri ve sınıflandırma metriğini döndür.

Önerilen yöntemin sözde kodu (pseudo code):

Input: Veri seti, algoritma parametreleri.

Output: En iyi metrik değeri veren öznitelikler ve metrik değeri.

1: begin algorithm

2: Kosinüs benzerlik yöntemi ile veri setinden sezgisel uzaklıkları hesapla.

3: while iterasyon sayısı 5’ten küçük ise do

4: Karınca koloni algoritması çalıştır.

5: Bulunan rotayı rota listesine ekle.

6:end while

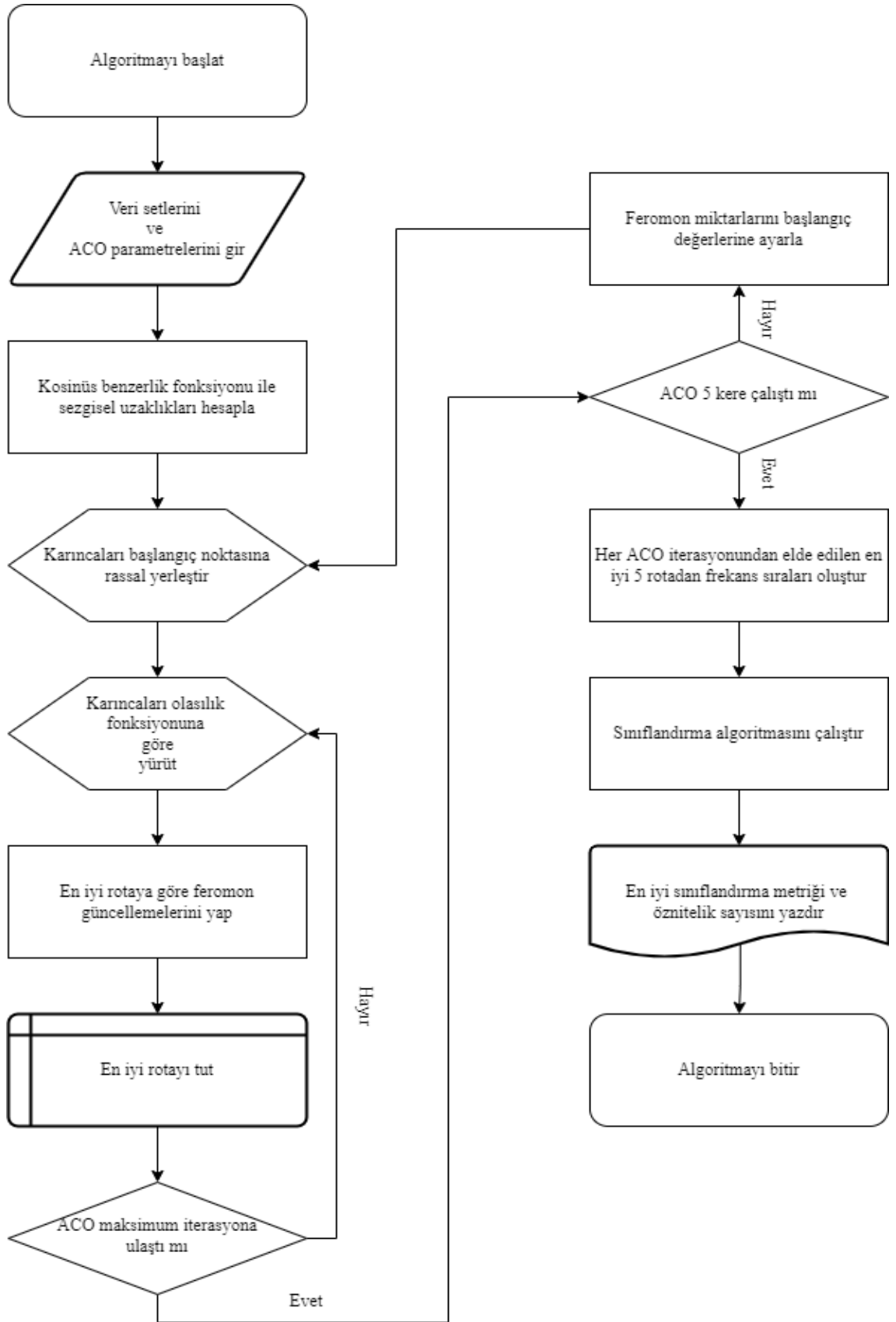
7: Rota listesinde bulunan her rota için ilk %80 kısmında bulunan öznitelikleri say. Öznitelik frekans listesine ekle.

8: Öznitelik frekans listesini artmayan sıraya diz.

9: Öznitelikleri öznitelik frekans listesine göre ekleyerek sınıflandırma algoritmasını çalıştır.

10: En iyi metrik değeri ve seçilen öznitelikleri çıktı olarak ver.

Önerilen yaklaşımın literatürden farklılaştığı noktalar, karınca koloni algoritmasında sezgisel uzaklıkların kullanımı ve çıkan bilginin farklı değerlendirilmesidir. Sezgisel uzaklıklar, literatürde karınca koloni algoritmasının olasılık fonksiyonunda çarpmaya göre tersi alınarak kullanılmıştır. Önerilen yöntemde çarpmaya göre tersi alınmamıştır. Literatürde karınca koloni algoritmasından gelen bilgi, rotaların alt öznitelik kümeleri oluşturması veya feromon miktarlarının yüksek olduğu rotaların kullanılması şeklindedir. Önerilen yöntemde karınca koloni algoritması birden fazla kez çalıştırılmış olup, çıkan rotalardan frekans sırası oluşturulmuştur. Sezgisel yaklaşımlar başlangıç noktası gibi değişkenlerin değişmesi ile farklı çözüm kümelerin araştırılması mümkün olur. Böylece farklı çözüm kümelerinden gelen bilgi birleştirilmiştir.



Şekil 3.8. Önerilen yaklaşımın iş akış diyagramı

#### 4. BULGULAR ve TARTIŞMA

Bu bölümde yapılan çalışma sonuçları gösterilmiştir. Çalışma Python kullanılarak kodlanmıştır. Daha önceki çalışma ile karşılaştırmak amacı ile veri kümeleri 2/3 oranı eğitim kümesine ve 1/3 oranı ile test kümesine bölünmüştür. Veri setlerindeki tüm değerlerin etkilerini görmek amacı ile 10'lu çapraz doğrulama yöntemi de kullanmıştır. Veri setlerinde eksik değerler yerine ortalama değerler seçilmiştir.

##### 4.1. Karınca Koloni Algoritması Parametre Ayarlaması

Tüm meta- sezgisel yöntemlerde olduğu gibi, karınca koloni algoritmasının da daha verimli çalışması için uygun parametre değerleri kullanılmalıdır.

Parametre ayarlamaları WDBC veri seti üzerinden yapılmıştır. Veri seti 2/3 eğitim seti, 1/3 test seti olarak bölünmüştür. Sınıflandırma algoritması olarak, K en yakın komşuluk algoritması komşuluk sayısı  $k=5$  değeri kullanılmış ve performans metriği olarak doğruluk oranı kullanılmıştır. Sınıflandırma algoritması beş kere çalıştırılmış beş çalışmanın ortalama doğruluk oranı değeri alınmıştır.

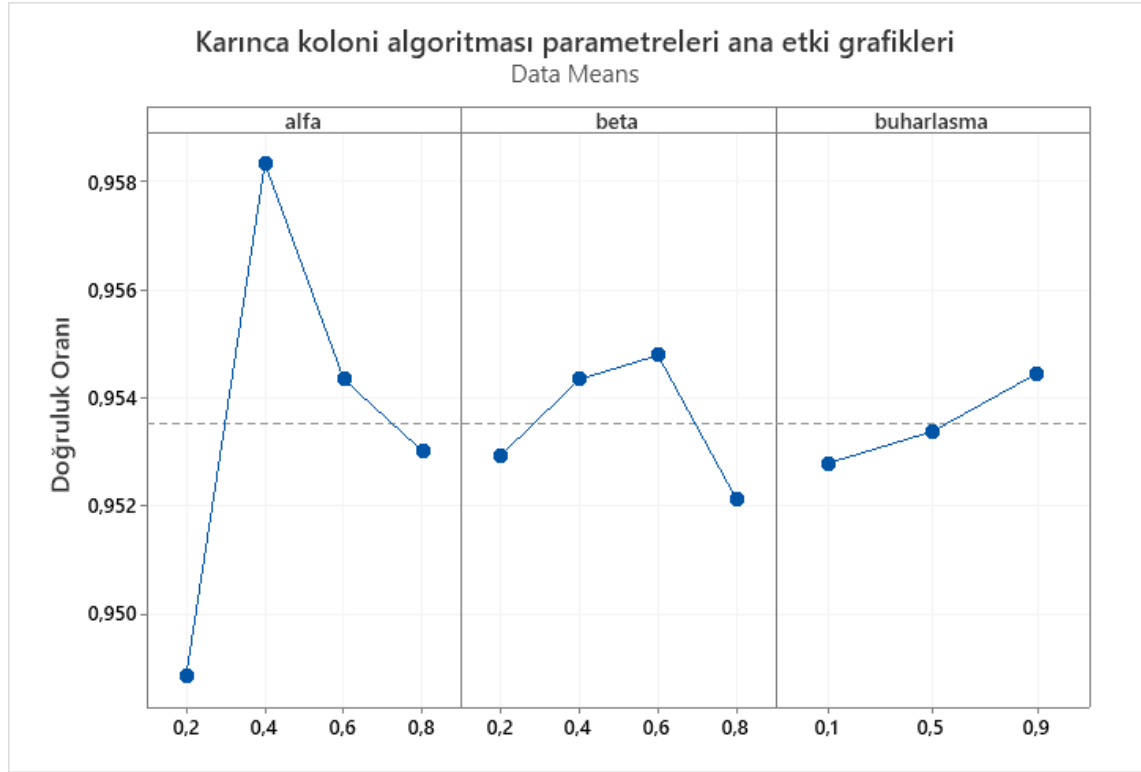
#### Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Model	8	0,000556	0,000070	1,01	0,442
Linear	8	0,000556	0,000070	1,01	0,442
alfa	3	0,000483	0,000161	2,34	0,088
beta	3	0,000049	0,000016	0,24	0,870
buharlasma	2	0,000025	0,000012	0,18	0,836
Error	39	0,002678	0,000069		
Total	47	0,003234			

##### Şekil 4.1. Karınca koloni algoritması parametreleri Minitab ANOVA sonucu

Şekil 4.1'de de görülebileceği gibi, doğruluk oranı üzerinde en etkili faktör alfa faktörü seçilmiştir. P değerleri %5'den büyüktür Ho hipotezi olan ana etkiler çıktı üzerinde etkisizdir red edilemez. İstatiksel olarak doğruluk oranı üzerinde etkili faktör yoktur. Doğruluk oranı [0,941-0,973] arasında değişmektedir.

Karınca koloni algoritması parametre değerlerini sezgisel olarak seçmemek için ana etki grafikleri oluşturulmuştur. Şekil 4.2’de gösterilmiştir.



**Şekil 4.2.** Karınca koloni algoritması parametre değerleri ana etki grafikleri

Şekil 4.2’deki grafiklerden de görülebileceği gibi, en yüksek doğruluk değerlerinin elde edildiği değerler alfa için 0,4, beta için 0,6 ve buharlaşma katsayısı için 0,9 olup, algoritma için bu parametre değerleri seçilmiştir.

Karınca koloni algoritması iterasyon sayısı artıkça hesaplama maliyeti katlanarak artmaktadır. Farklı iterasyon sayısındaki performans değerleri için WDBC veri seti üzerinde 100 ve 1000 iterasyon sayısı değerleri denenmiştir. 100 iterasyon sayısı için 14 öznitelik 0,95 doğruluk oranı elde edilmiştir. 10 iterasyona göre %2 daha iyidir. 1000 iterasyonda 5 öznitelik 0,95 doğruluk oranı elde edilmiştir. 10 iterasyona göre %2 daha iyidir aynı zamanda 100 iterasyona göre daha az öznitelik seçmiştir. Büyük veri setlerinde uzun çalışma süreleri sebebi ile 10 iterasyon değeri seçilmiştir. Karınca koloni algoritması seçilen parametre değerleri Çizelge 4.1’de gösterilmiştir.

**Çizelge 4.1** Karınca koloni algoritması seçilen parametre değerleri

Parametre	Değeri
Alfa	0,4
Beta	0,6
Buharlaştırma Katsayısı	0,9
Karınca Sayısı	10
Başlangıç Feromon Miktarları	10
İterasyon Sayısı	10

Çizelge 4.1’de gösterilen parametre değerleri ile karınca koloni algoritması çalıştırılarak her veri seti için öznelik frekans sıraları oluşturulmuştur.

#### 4.2. Sonuçların Literatürdeki Çalışmalar ile Karşılaştırılması

Bu bölümde, Tabakhi ve diğerleri (2014)’nin karınca koloni algoritması kullanarak yaptıkları yöntem (UFSACO) ile tez kapsamında önerilen yöntem karşılaştırılmıştır. UFSACO’dan farklı olarak önerilen yaklaşımda sezgisel uzaklıkların çarpmaya göre tersi alınmadan kullanılması ve öznelik frekans sıraları önerilmiştir.

UFSACO’da 2/3 eğitim, 1/3 test seti bölünmesi kullanılmıştır. Karşılaştırmak için aynı bölünme yöntemi ve UFSACO’da seçilen öznelik sayıları kullanılmıştır.

**Çizelge 4.2.** Önerilen yöntemin, UFSACO algoritması sonuçları ile karşılaştırılması (Tabakhi ve diğerleri (2014), Tablo 2 ve Tablo 3)

Veri Seti	Seçilen Öznelik Sayısı	KNN Doğruluk Oranı	SVM Doğruluk Oranı	UFSACO DT Doğruluk Oranı	UFSACO SVM Doğruluk Oranı
WDBC	5	0,931	<b>0,941</b>	0,919	0,907
Dermatology	25	0,826	<b>0,968</b>	0,918	0,953
Ionosphere	30	0,828	0,882	<b>0,886</b>	<b>0,886</b>
Arrhythmia	20	<b>0,597</b>	<b>0,597</b>	0,591	0,592
Wine	5	0,915	0,857	<b>0,951</b>	<b>0,951</b>
Hepatitis	5	0,788	0,788	0,788	<b>0,831</b>

**Çizelge 4.2.** Önerilen yöntemin, UFSACO algoritması sonuçları ile karşılaştırılması (Tabakhi ve diğerleri (2014), Tablo 2 ve Tablo 3) (devam)

<b>Spambase</b>	40	0,997	<b>0,998</b>	0,925	0,878
<b>Madellon</b>	70	0,510	0,486	<b>0,764</b>	0,611

Çizelge 4.2'ye göre, WDBC, Dermatology, Arrhythmia ve Spambase veri setlerinde önerilen yöntemin SVM sınıflandırma sonuçlarında daha iyi doğruluk oranı elde edilmiştir. 4 veri setinde, UFSACO DT sınıflandırmasına sonucundan daha iyi ortalama doğruluk oranına sahip UFSACO SVM sınıflandırmasında sonucu, ortalama %4,5 daha iyi sonuçlar elde edilmiştir.

Tabakhi ve diğerleri (2014) tarafından yapılan çalışmada RSM ve MC filtreleme algoritmaları ile karşılaştırılmıştır.

**Çizelge 4.3.** Önerilen yöntemin, RSM algoritması ile karşılaştırılması (Tabakhi ve diğerleri (2014), Tablo 2 ve Tablo 3)

<b>Veri Seti</b>	<b>Seçilen Öznitelik Sayısı</b>	<b>KNN Doğruluk Oranı</b>	<b>SVM Doğruluk Oranı</b>	<b>RSM SVM Doğruluk Oranı</b>	<b>RSM DT Doğruluk Oranı</b>
<b>WDBC</b>	5	0,931	<b>0,941</b>	0,8382	0,8634
<b>Dermatology</b>	25	0,826	<b>0,968</b>	0,9488	0,916
<b>Ionosphere</b>	30	0,828	0,882	0,8784	<b>0,8834</b>
<b>Arrhythmia</b>	20	<b>0,597</b>	<b>0,597</b>	0,5611	0,5571
<b>Wine</b>	5	<b>0,915</b>	0,857	0,8197	0,8634
<b>Hepatitis</b>	5	0,788	0,788	<b>0,8094</b>	0,7755
<b>Spambase</b>	40	0,997	<b>0,998</b>	0,8547	0,9176
<b>Madellon</b>	70	0,510	0,486	<b>0,535</b>	0,5217

Çizelge 4.3'e göre, WDBC, Dermatology, Ionosphere, Arrhythmia, Wine ve Spambase veri setlerinde RSM öznitelik seçiminden daha iyi sonuçlar elde edilmiştir. 5 veri setinde, önerilen yöntemin SVM sınıflandırma sonucu, RSM SVM sınıflandırma sonucundan daha iyi ortalama doğruluk oranına sahip RSM DT sınıflandırma sonucundan, ortalama %6 daha iyi sonuçlar elde edilmiştir.



**Çizelge 4.4.** Önerilen yöntemin, MC algoritması ile karşılaştırılması (Tabakhi ve diğerleri (2014), Tablo 2 ve Tablo 3)

Veri Seti	Seçilen Öznitelik Sayısı	KNN Doğruluk Oranı	SVM Doğruluk Oranı	MC filtresi SVM Doğruluk Oranı	MC filtresi DT Doğruluk Oranı
WDBC	5	0,931	<b>0,941</b>	0,8897	0,9108
Dermatology	25	0,826	<b>0,968</b>	0,9456	0,9112
Ionosphere	30	0,828	0,882	0,8533	<b>0,885</b>
Arrhythmia	20	<b>0,597</b>	<b>0,597</b>	0,4546	0,4544
Wine	5	0,915	0,857	0,8962	<b>0,9235</b>
Hepatitis	5	0,788	0,788	0,8273	<b>0,8359</b>
Spambase	40	0,997	<b>0,998</b>	0,8633	0,9146
Madellon	70	0,510	0,486	<b>0,515</b>	0,5

Çizelge 4.4'e göre, WDBC, Dermatology, Arrhythmia ve Spambase veri setlerinde MC öznitelik seçiminden daha iyi sonuçlar elde edilmiştir. 4 veri setinde, önerilen yöntemin SVM sınıflandırma sonucu, MC SVM sınıflandırma sonucundan daha iyi ortalama doğruluk oranına sahip MC DT sınıflandırma sonucundan, ortalama %9 daha iyi sonuçlar elde edilmiştir.

#### 4.3. K En Yakın Komşuluk Sınıflandırması Sonuçları

Veri setleri 1/3 test ve 2/3 eğitim olmak üzere bölünmüştür. Karınca koloni algoritması bulunan parametre değerleri ile her veri seti için çalıştırılmıştır. Elde edilen öznitelik sırasına göre her öznitelik değişiminde K en yakın komşuluk algoritması 5 defa çalıştırılmıştır, başarı ölçütü olarak doğruluk oranı ortalaması alınmıştır. Komşuluk sayısı k=5 seçilmiştir.

**Çizelge 4.5.** K en yakın komşuluk algoritması doğruluk oranı sonuçları

<b>Veri Seti</b>	<b>Toplam Öznitelik Sayısı</b>	<b>Bütün Öznitelikler Seçildiğinde Doğruluk Oranı</b>	<b>ACO ile Seçilen Öznitelik Sayısı</b>	<b>Doğruluk Oranı</b>
<b>WDBC</b>	30	0,931	25	<b>0,945</b>
<b>Dermatology</b>	34	0,832	18	<b>0,957</b>
<b>Ionosphere</b>	34	0,83	4	<b>0,857</b>
<b>Arrhythmia</b>	279	0,627	140	<b>0,69</b>
<b>Wine</b>	13	0,69	8	<b>0,963</b>
<b>Hepatitis</b>	19	0,654	3	<b>0,803</b>
<b>Spambase</b>	57	0,997	54	<b>0,998</b>
<b>Madellon</b>	500	0,718	465	<b>0,736</b>

Çizelge 4.5'e göre bütün veri setlerinde özniteliklerin seçimi yapılmadan önce ve öznitelik seçimi yapıldıktan sonra k en yakın komşuluk algoritması ile sınıflandırma sonuçları karşılaştırıldığında öznitelik seçilmesi sonrası daha iyi doğruluk oranları elde edilmiştir. 8 veri seti için ortalama %10 iyileşme görülmüştür.

Başarı ölçütü F-puanı seçilerek algoritma çalıştırılmıştır. K en yakın komşuluk algoritması 5 defa çalıştırılmış, F-puanı değerlerinin ortalaması alınmıştır.

**Çizelge 4.6.** K en yakın komşuluk algoritması F-puanı sonuçları

<b>Veri Seti</b>	<b>Toplam Öznitelik sayısı</b>	<b>Bütün Öznitelikler Seçildiğinde F-Puanı</b>	<b>ACO ile Seçilen Öznitelik Sayısı</b>	<b>F-puanı</b>
<b>WDBC</b>	30	0,941	25	<b>0,947</b>
<b>Dermatology</b>	34	0,836	23	<b>0,953</b>
<b>Ion</b>	34	0,891	4	<b>0,897</b>
<b>Arrhythmia</b>	279	0,630	233	<b>0,659</b>
<b>Wine</b>	13	0,681	8	<b>0,942</b>

**Çizelge 4.6.** K en yakın komşuluk algoritması F-puanı sonuçları (devam)

<b>Hepatitis</b>	<b>19</b>	<b>0,857</b>	<b>3</b>	0,887
<b>Spambase</b>	57	<b>0,998</b>	34	0,996
<b>Madellon</b>	500	0,706	441	<b>0,721</b>

Çizelge 4.6'ya göre, Spambase veri seti haricinde diğer veri setlerinde k en yakın komşuluk algoritması sınıflandırma sonucuna göre öznitelik seçimi sonrası öncesine göre daha iyi F-puanı değeri elde edilmiştir. 7 veri seti için %8 iyileşme görülmüştür. Spambase veri setinde 34 öznitelik seçildiğinde ise doğruluk oranı miktarı 0,002 kötüleşir. Kötüleşme oranı %0,2'dir.

Veri setleri için k-kat çapraz doğrulama yapılmıştır. k 10 seçilmiştir. Her veri seti için öznitelik değişiminde k en yakın komşuluk algoritması çalıştırılmış. Sınıflandırma metriği olarak doğruluk oranı kullanılmıştır.

**Çizelge 4.7.** K en yakın komşuluk algoritması çapraz doğrulama sonuçları

<b>Veri Seti</b>	<b>Öznitelik Sayısı Tamamı</b>	<b>Bütün Öznitelikler Seçildiğinde Doğruluk Oranı</b>	<b>ACO ile Seçilen Öznitelik Sayısı</b>	<b>Doğruluk Oranı</b>
<b>WDBC</b>	30	0,929	27	<b>0,942</b>
<b>Dermatology</b>	34	0,879	23	<b>0,967</b>
<b>Ionosphere</b>	34	0,843	3	<b>0,869</b>
<b>Arrhythmia</b>	279	0,628	129	<b>0,638</b>
<b>Wine</b>	13	0,691	8	<b>0,943</b>
<b>Hepatitis</b>	19	0,766	4	<b>0,818</b>
<b>Spambase</b>	57	<b>0,998</b>	34	<b>0,998</b>
<b>Madellon</b>	500	0,731	497	<b>0,733</b>

Çizelge 4.7'ye göre, Spambase veri seti hariç bütün veri setlerinde özniteliklerin seçimi yapılmadan önce ve öznitelik seçimi yapıldıktan sonra k en yakın komşuluk algoritması ile sınıflandırma sonuçları karşılaştırıldığında öznitelik seçilmesi sonrası daha iyi

doğruluk oranları elde edilmiştir. 7 veri seti için ortalama %8 iyileşme görülmüştür. Spambase veri setinde iyileşme veya kötüleşme olmamıştır.

Veri setleri 10 kat çapraz doğrulama ile analiz edildiği zaman Dermatology, Ionsphere ve Hepatitis veri setlerinde 2/3 eğitim seti analizine göre daha iyi doğruluk oranları vermiştir. Dermatology ve Hepatitis veri setlerinde 10 kat çapraz doğrulama, 2/3 eğitim setine göre daha fazla öznitelik seçmiştir (bkz Çizelge 4.5 ve Çizelge 4.7).

#### 4.4 Destek Vektör Makineleri Sınıflandırması Sonuçları

Veri setleri 1/3 test ve 2/3 eğitim olmak üzere bölünmüştür. Karınca koloni algoritmasından elde edilen öznitelik sırasına göre öznitelik değişiminde destek vektör makineleri 5 defa çalıştırılmıştır. Sınıflandırma sonucu olarak başarı ölçütü olarak doğruluk oranı ortalaması alınmıştır. Çekirdek fonksiyon doğrusal ve C değeri 1 seçilmiştir.

**Çizelge 4.8.** Destek vektör makineleri algoritması doğruluk oranı sonuçları

Veri Seti	Toplam Öznitelik Sayısı	Bütün Öznitelikler Seçildiğinde Doğruluk Oranı	ACO ile Seçilen Öznitelik Sayısı	Doğruluk Oranı
WDBC	30	0,951	19	<b>0,960</b>
Dermatology	34	0,960	23	<b>0,973</b>
Ionosphere	34	0,865	23	<b>0,884</b>
Arrhythmia	279	0,646	262	<b>0,693</b>
Wine	13	0,942	11	<b>0,979</b>
Hepatitis	19	0,834	9	<b>0,861</b>
Spambase	57	<b>0,999</b>	34	<b>0,999</b>
Madellon	500	<b>0,543</b>	461	0,541

Çizelge 4.8'e göre, Spambase veri seti hariç bütün veri setlerinde özniteliklerin seçimi yapılmadan önce ve öznitelik seçimi yapıldıktan sonra destek vektör makineleri algoritması ile sınıflandırma sonuçları karşılaştırıldığında öznitelik seçilmesi sonrası

daha iyi doğruluk oranları elde edilmiştir. 6 veri seti için %2,9 iyileşme görülmüştür. Spambase veri setinde iyileşme veya kötüleşme gözükmemiştir. Madellon veri setinde 461 öznitelik seçildiğinde ise doğruluk oranı miktarı 0,02 kötüleşir. Kötüleşme oranı %0,37'dir.

Başarı ölçütü F-puanı seçilerek algoritma çalıştırılmıştır. Destek vektör makineleri algoritması 5 defa çalıştırılmış, F-puanı değerlerinin ortalaması alınmıştır.

**Çizelge 4.9.** Destek vektör makineleri algoritması F-puanı sonuçları

<b>Veri Seti</b>	<b>Toplam Öznitelik Sayısı</b>	<b>Bütün Öznitelikler Seçildiğinde F puanı</b>	<b>ACO ile Seçilen Öznitelik Sayısı</b>	<b>F puanı</b>
<b>WDBC</b>	30	0,955	25	<b>0,972</b>
<b>Dermatology</b>	34	0,960	19	<b>0,975</b>
<b>Ionosphere</b>	34	0,907	31	<b>0,927</b>
<b>Arrhythmia</b>	279	0,673	241	<b>0,690</b>
<b>Wine</b>	13	0,976	12	<b>0,980</b>
<b>Hepatitis</b>	19	0,874	12	<b>0,923</b>
<b>Spambase</b>	57	<b>0,999</b>	34	<b>0,999</b>
<b>Madellon</b>	500	0,543	461	<b>0,549</b>

Çizelge 4.9'a göre, Spambase veri seti hariç bütün veri setlerinde özniteliklerin seçimi yapılmadan önce ve öznitelik seçimi yapıldıktan sonra destek vektör makineleri algoritması ile sınıflandırma sonuçları karşılaştırıldığında öznitelik seçilmesi sonrası daha iyi F-puanı elde edilmiştir. 7 veri seti için ortalama %2 iyileşme görülmüştür. Spambase veri setinde iyileşme veya kötüleşme olmamıştır. 6 veri seti için %2 iyileşme görülmüştür. Spambase veri setinde iyileşme veya kötüleşme gözükmemiştir.

Veri setleri için k-kat çapraz doğrulama yapılmıştır. k 10 seçilmiştir. Her veri seti için öznitelik değişiminde destek vektör makineleri algoritması çalıştırılmış. Sınıflandırma metriği olarak doğruluk oranı kullanılmıştır.

**Çizelge 4.10.** Destek vektör makineleri algoritması çapraz doğrulama sonuçları

<b>Veri Seti</b>	<b>Toplam Öznitelik Sayısı</b>	<b>Bütün Öznitelikler Seçildiğinde Doğruluk Oranı</b>	<b>ACO ile Seçilen Öznitelik Sayısı</b>	<b>Doğruluk Oranı</b>
<b>WDBC</b>	30	0,949	25	<b>0,960</b>
<b>Dermatology</b>	34	0,956	20	<b>0,980</b>
<b>Ionosphere</b>	34	0,871	26	<b>0,888</b>
<b>Arrhythmia</b>	279	0,640	132	<b>0,654</b>
<b>Wine</b>	13	0,982	12	<b>0,988</b>
<b>Hepatitis</b>	19	0,864	11	<b>0,857</b>
<b>Spambase</b>	57	<b>0,999</b>	34	<b>0,999</b>
<b>Madellon</b>	500	0,560	440	<b>0,573</b>

Çizelge 4.10'a göre, Spambase veri seti hariç bütün veri setlerinde özniteliklerin seçimi yapılmadan önce ve öznitelik seçimi yapıldıktan sonra destek vektör makineleri algoritması ile sınıflandırma sonuçları karşılaştırıldığında öznitelik seçilmesi sonrası daha iyi doğruluk oranları elde edilmiştir. 7 veri seti için ortalama %1 iyileşme görülmüştür. Spambase veri setinde 34 öznitelik seçildiğinde ise doğruluk oranı miktarı 0,001 kötüleşir. Kötüleşme oranı %0,1'dir.

Veri setleri 10 kat çapraz doğrulama ile analiz edildiği zaman Dermatology, Ionospher, Wine ve Madellon veri setlerinde 2/3 eğitim seti analizine göre daha iyi doğruluk oranları vermiştir. Ionospher ve Wine veri setlerinde 10 kat çapraz doğrulama, 2/3 eğitim setine göre daha fazla öznitelik seçmiştir. (bkz Çizelge 4.8 ve Çizelge 4.10)

#### **4.5 K En Yakın Komşuluk Ve Destek Vektör Makinelerin Karşılaştırması**

Bu bölümde, kullanılan sınıflandırma yöntemleri karşılaştırılmıştır. Daha önce verilen çapraz doğrulama ve F-puanı sonuçları kullanılmıştır.

**Çizelge 4.11** K en yakın komşuluk ve destek vektöre makineleri F-puanı karşılaştırması

Veri Seti	ACO ile Seçilen Öznitelik Sayısı	KNN F-puanı	ACO ile Seçilen Öznitelik Sayısı	SVM F-puanı
WDBC	25	0,947	25	<b>0,972</b>
Dermatology	23	0,953	19	<b>0,975</b>
Ionosphere	4	0,897	31	<b>0,927</b>
Arrhythmia	233	0,659	241	<b>0,690</b>
Wine	8	0,942	12	<b>0,980</b>
Hepatitis	3	0,887	12	<b>0,923</b>
Spambase	34	0,996	34	<b>0,999</b>
Madellon	441	<b>0,721</b>	461	0,549

Çizelge 4.11'e göre, destek vektör makineleri Madellon veri seti haricinde daha iyi F-puanı elde edilmiştir. 7 veri seti için ortalama %2.95 iyileşme görülmüştür. Madellon veri setinde K en yakın komşuluk algoritması kullanıldığında %31 iyileşme görülmüştür. Destek vektör makineleri, bütün veri setlerinde %0,18 daha başarılıdır. Ionosphere, Arrhythmia, Wine ve Hepatitis veri setlerinde öznitelik sayısı daha fazladır. Spambase ve WDBC veri setlerinde öznitelik sayısı aynıdır.

**Çizelge 4.12.** K en yakın komşuluk ve destek vektöre makineleri çapraz doğrulama doğruluk oranları karşılaştırması

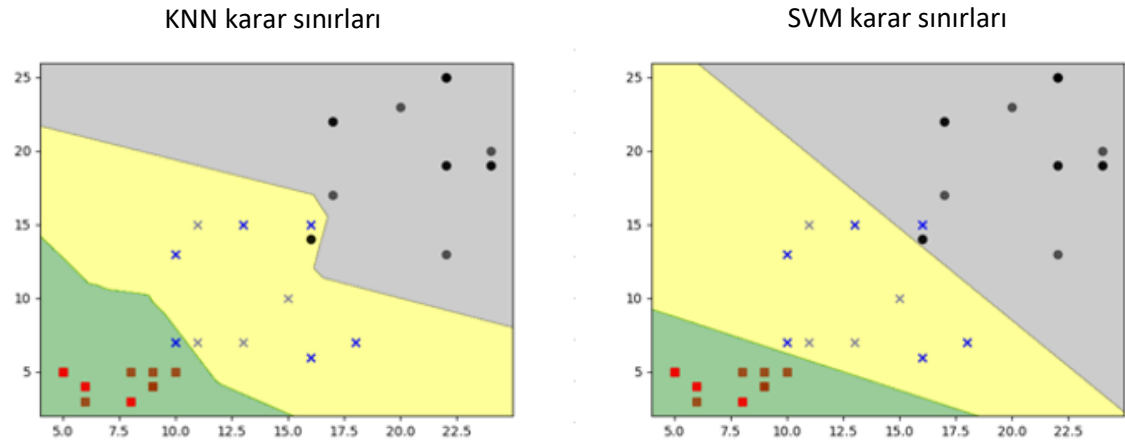
Veri Seti	ACO ile Seçilen Öznitelik Sayısı	KNN Doğruluk Oranı	ACO ile Seçilen Öznitelik Sayısı	SVM Doğruluk Oranı
WDBC	27	0,942	25	<b>0,960</b>
Dermatology	23	0,967	20	<b>0,980</b>
Ionosphere	3	0,869	26	<b>0,888</b>
Arrhythmia	129	0,638	132	<b>0,654</b>
Wine	8	0,943	12	<b>0,988</b>
Hepatitis	4	0,818	11	<b>0,857</b>

**Çizelge 4.12.** K en yakın komşuluk ve destek vektöre makineleri çapraz doğrulama doğruluk oranları karşılaştırması (devam)

<b>Spambase</b>	34	0,998	34	<b>0,999</b>
<b>Madellon</b>	497	<b>0,733</b>	440	0,573

Çizelge 4.12'ye göre, destek vektör makineleri WDBC, Dermatology, Ionosphere, Arrhythmia, Wine, Hepatitis ve Spambase veri setlerinde daha iyi doğruluk oranları elde edilmiştir. 7 veri seti için ortalama %2.44 iyileşme görülmüştür. Madellon veri setinde K en yakın komşuluk algoritması kullanıldığında %28 iyileşme görülmüştür. K en yakın komşuluk, bütün veri setlerinde %0,13 daha başarılıdır. Destek vektör makinelerinde Ionosphere, Arrhythmia, Wine ve Hepatitis veri setlerinde öznelik sayısı daha fazladır. Spambase veri setinde öznelik sayıları aynıdır.

Ek 1'de yer alan rassal veri kümesi ile sınıflandırma yöntemleri karar sınırları Şekil 4.3 gösterilmiştir.



**Şekil 4.3.** K en yakın komşuluk algoritması ve destek vektör makineleri rassal veri kümesi sınıflandırma sonucu karar sınırları

Şekil 4.3'te görülebileceği gibi, K en yakın komşuluk karar sınırları komşu değerlerden etkilendiği için girintili karar sınırları elde edilmiştir. Destek vektöre makineleri çekirdek fonksiyonu doğrusal seçildiği için doğrusal karar sınırları elde edilmiştir.



## 5. SONUÇ

Bu tez çalışmasında, öznitelik seçim problemlerinde yüksek doğruluk oranı verebilecek öznitelik kümesi bulabilmek için karınca koloni algoritmasına dayalı hibrit bir yaklaşım önerilmiştir.

Seçilen öznitelikleri değerlendirmek için destek vektör makineleri ve K en yakın komşuluk algoritmaları kullanılmıştır. Sınıflandırma metrikleri doğruluk oranı ve F-puanı seçilmiştir. Farklı eğitim/test kümeleri ile sonuçlar elde edilmiş değerlendirilmiştir.

Önerilen yaklaşımı doğrulamak için tez kapsamında önerilen yöntem ile literatürdeki UFSACO yöntemi karşılaştırılmış, kullanılan 8 veri setinin 4 adedinde daha yüksek doğruluk oranları elde edilmiştir. İlaven ve bilinen tek değişkenli filtreleme yöntemleri olan RSM ve MC algoritmaları ile de yöntem karşılaştırılmıştır.

Sonuçlar baz alınarak önerilen yöntem öznitelik seçimi problemlerinde kullanılabilir.

Zaman kısıtı olmadığı karınca koloni algoritmasında karınca ve iterasyon sayısı artırarak daha iyi rotalar elde edilebilir. Öznitelik seçimi frekans sırası yerine aynı frekans değerine sahip özniteliklerin rassal seçimi ile birden fazla öznitelik alt kümesi elde edilerek topluluk öğrenimi yapılırsa daha iyi tahmin değerleri elde edilebilir. Önerilen yöntemin başarısı sınıflandırma yöntemlerine göre değiştiği için, gelecek çalışmalarda tez kapsamında değerlendirilmemiş olan daha gelişmiş sınıflandırma teknikleri ile de sonuçlar değerlendirilebilir. Karınca koloni algoritmasında kullanılan öznitelikler arası sezgisel uzaklık hesaplama yöntemi, algoritmanın başarısını büyük ölçüde etkilediği için öznitelik arası örüntüleri daha iyi yakalayabilecek metotlar ile de çalışmalar yapılabilir. Sonraki çalışmalarda ilaven, hibrit yaklaşım yerine gömülü yöntem kullanılarak karınca koloni algoritması ile sınıflandırma algoritmaları arasında karşılıklı bilgi alışverişi yapılırsa karıncaların rota seçiminin sınıflandırma sonuçları ile değerlendirerek daha iyi rotalar elde edilebilmesi sağlanabilir.

## KAYNAKLAR

- Aghdam, M. H., Ghasem-Aghaee, N., ve Basiri, M. E. 2009. *Expert Systems with Applications*, 36(3), 6843–6853. doi: 10.1016/j.eswa.2008.08.022
- Alanazi, M. A., Siraj, M. M., ve Ghaleb, F. A. 2021. Hybrid Feature Selection by Combining Wrapper and Filter Methods for Malware Detection. In *2021 International Conference on Data Science and Its Applications (ICoDSA)* (pp. 233-239). IEEE. doi: 10.1109/icodsa53588.2021.9617521
- Al-Ani, A. 2005. Ant Colony Optimization for Feature Subset Selection. In *WEC (2)* (pp. 35-38).
- Ali, S. I., ve Shahzad, W. 2012, October. A feature subset selection method based on symmetric uncertainty and ant colony optimization. In *2012 International Conference on Emerging Technologies* (pp. 1-6). IEEE. doi: 10.1109/ICET.2012.6375420
- Bindu, M. G., ve Sabu, M. K. 2020. A Hybrid Feature Selection Approach Using Artificial Bee Colony and Genetic Algorithm. In *2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)* (pp. 211-216). IEEE. doi: 10.1109/ACCTHPA49271.2020.9213197
- Boser, B. E., Guyon, I. M., ve Vapnik, V. N. 1992, July. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144-152). doi: 10.1145/130385.130401
- Cover, T., ve Hart, P. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1), 21-27. doi: 10.1109/TIT.1967.1053964
- Cover, T. M., ve Thomas, J. A. 2006. *Elements of Information Theory*, (2nd edn, 2006). doi: 10.1002/047174882X
- Deneubourg, J. L., Aron, S., Goss, S., ve Pasteels, J. M. 1990. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, 3(2), 159–168. doi: 10.1007/BF01417909
- Deriche, M. 2009. Feature selection using ant colony optimization. In *2009 6th International Multi-Conference on Systems, Signals and Devices* (pp. 1-4). IEEE. doi: 10.1109/SSD.2009.4956825
- Dorigo, M. 1992. *Optimization, Learning and Natural Algorithms*. (Doktora Tezi). Italy.
- Dorigo, M., Birattari, M., ve Stutzle, T. 2006. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4), 28-39. doi: 10.1109/MCI.2006.329691
- Dua, D. ve Graff, C. 2019. UCI Machine Learning Repository. *University of California, School of Information and Computer Science*, Irvine, CA. Erişim adresi: <http://archive.ics.uci.edu/ml>

- Eroglu, D. Y., ve Kilic, K. 2017. A novel Hybrid Genetic Local Search Algorithm for feature selection and weighting with an application in strategic decision making in innovation management. *Information Sciences*, 405, 18-32. doi: 10.1016/j.ins.2017.04.009
- Fahrudin, T. M., Syarif, I., ve Barakbah, A. R. 2016. Ant colony algorithm for feature selection on microarray datasets. In *2016 International Electronics Symposium (IES)* (pp. 351-356). IEEE. doi: 10.1109/ELECSYM.2016.7861030
- Fix, E. and Hodges, J.L. 1951 Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties. *Technical Report 4, USAF School of Aviation Medicine, Randolph Field.*
- Gheyas, I. A., ve Smith, L. S. 2010. Feature subset selection in large dimensionality domains. *Pattern recognition*, 43(1), 5-13. doi: 10.1016/j.patcog.2009.06.009
- Hamed, T., Dara, R., ve Kremer, S. C. 2014. An accurate, fast embedded feature selection for SVMs. In *2014 13th International Conference on Machine Learning and Applications* (pp. 135-140). IEEE. doi: 10.1109/ICMLA.2014.104
- He, X., Cai, D., ve Niyogi, P. 2005. Laplacian score for feature selection. *Advances in neural information processing systems*, 18. (pp. 507–514). Erişim adresi: <https://papers.nips.cc/paper/2005/file/b5b03f06271f8917685d14cea7c6c50a-Paper.pdf>
- Jensen R. 2005. Combining rough and fuzzy sets for featurer selection (Doktora Tezi). Erişim adres: <https://users.aber.ac.uk/rkj/PhDthesis.pdf>
- Jiménez-Cordero, A., Morales, J. M., ve Pineda, S. 2021. A novel embedded min-max approach for feature selection in nonlinear Support Vector Machine classification. *European Journal of Operational Research*, 293(1), 24-35. doi: 10.1016/j.ejor.2020.12.009
- Kanan, H. R., ve Faez, K. 2008. An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system. *Applied Mathematics and Computation*, 205(2), 716-725. doi: 10.1016/j.ame.2008.05.115
- Kashef, S., ve Nezamabadi-pour, H. 2015. An advanced ACO algorithm for feature subset selection. *Neurocomputing*, 147, 271-279. doi: 10.1016/j.neucom.2014.06.067
- Liu, H., ve Yu, L. 2005. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering*, 17(4), 491-502. doi: 10.1109/TKDE.2005.66
- Maldonado, S., ve López, J. 2018. Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification. *Applied Soft Computing*, 67, 94-105. doi: 10.1016/j.asoc.2018.02.051
- Malini Devi, G., Seetha, M., ve Sunitha, K. V. 2016. A Novel K-Nearest Neighbor Technique for Data Clustering using Swarm Optimization. *International Journal of Geoinformatics*, 12(1). 75-82. doi: 10.52939/ijg.v12i1.935
- Manosij, G., Ritam, G., Sarkar, R., ve Abraham, A. 2020. A wrapper-filter feature selection technique based on ant colony optimization. *Neural Computing & Applications*, 32(12), 7839-7857. doi: 10.1007/s00521-019-04171-3

- Metlek, S., ve Kayaalp, K. 2020. Makine öğrenmesinde, teoriden örnek matlab uygulamalarına kadar destek vektör makineleri (p. 100). Erişim adresi: <https://iksadyayinevi.com/wp-content/uploads/2020/12/MAKINE-OGRENMESINDE-TEORIDEN-ORNEK-MATLAB-UYGULAMALARINA-KADAR-DESTEK-VEKTOR-MAKINELERI.pdf>
- Min, H., ve Fangfang, W. 2010. Filter-wrapper hybrid method on feature selection. In *2010 Second WRI Global Congress on Intelligent Systems* (Vol. 3, pp. 98-101). IEEE. doi: 10.1109/GCIS.2010.235
- Mohammed, S. K., Deeba, F., Bui, F. M., ve Wahid, K. A. 2016. Feature selection using modified ant colony optimization for wireless capsule endoscopy. In *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* (pp. 1-4). IEEE. doi: 10.1109/UEMCON.2016.7777925
- Nagarajan, G., ve Babu, L. D. 2021. A hybrid feature selection model based on improved squirrel search algorithm and rank aggregation using fuzzy techniques for biomedical data classification. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 10(1), 1-29. doi: 10.1007/s13721-021-00313-7
- Naser, M. Z., ve Alavi, A. H. 2021. Error Metrics and Performance Fitness Indicators for Artificial Intelligence and Machine Learning in Engineering and Sciences. *Architecture, Structures and Construction*, 1-19. doi: 10.1007/s44150-021-00015-8
- Palma-Mendoza, R. J., de-Marcos, L., Rodriguez, D., ve Alonso-Betanzos, A. 2019. Distributed correlation-based feature selection in spark. *Information Sciences*, 496, 287-299. doi: 10.1016/j.ins.2018.10.052
- Paniri, M., Dowlatshahi, M. B., ve Nezamabadi-pour, H. 2020. MLACO: A multi-label feature selection algorithm based on ant colony optimization. *Knowledge-Based Systems*, 192, 105285. doi: 10.1016/j.knsys.2019.105285
- Rostami, M., Berahmand, K., Nasiri, E., ve Forouzande, S. 2021. Review of swarm intelligence-based feature selection methods. *Engineering Applications of Artificial Intelligence*, 100, 104210. doi: 10.1016/j.engappai.2021.104210
- Russell, S. J., Norvig, P., ve Davis, E. 2010. Artificial intelligence: a modern approach. 3rd ed. Upper Saddle River, NJ: Prentice Hall. Erişim adresi: <https://cs.calvin.edu/courses/cs/344/kvlinden/resources/AIMA-3rd-edition.pdf>
- Sarac, E. ve Ozel, S. A. 2014. An ant colony optimization based feature selection for web page classification. *The Scientific World Journal*, 2014. doi: 10.1155/2014/649260
- Shanab, A. A., Khoshgoftaar, T. M., ve Wald, R. 2014. Evaluation of wrapper-based feature selection using hard, moderate, and easy bioinformatics data. In *2014 IEEE International Conference on Bioinformatics and Bioengineering* (pp. 149-155). IEEE. doi: 10.1109/BIBE.2014.62
- Shunmugapriya, P., ve Kanmani, S. 2017. A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid). *Swarm and Evolutionary Computation*, 36, 27-36. doi: 10.1016/j.swevo.2017.04.002

- Sivagaminathan, R. K., & Ramakrishnan, S. 2007. A hybrid approach for feature subset selection using neural networks and ant colony optimization. *Expert systems with applications*, 33(1), 49-60. doi: 10.1016/j.eswa.2006.04.010
- 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. doi: 10.1016/j.engappai.2014.03.007
- Unler, A., Murat, A., ve Chinnam, R. B. 2011. mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. *Information Sciences*, 181(20), 4625-4641. doi: 10.1016/j.ins.2010.05.037
- Wald, R., Khoshgoftaar, T. M., ve Napolitano, A. 2013, November. Should the Same Learners Be Used Both within Wrapper Feature Selection and for Building Classification Models?. In *2013 IEEE 25th International Conference on Tools with Artificial Intelligence* (pp. 439-445). IEEE. doi: 10.1109/ICTAL.2013.72
- Yan, H., ve Yang, J. 2015. Sparse discriminative feature selection. *Pattern Recognition*, 48(5), 1827-1835. doi: 10.1016/j.patcog.2014.10.021
- Yang, J., Awan, A. J., ve Vall-Llosera, G. 2019. Support Vector Machines on Noisy Intermediate Scale Quantum Computers. Erişim adresi: <http://arxiv.org/abs/1909.11988>
- You, M., Liu, J., Li, G. Z., & Chen, Y. 2012. Embedded feature selection for multi-label classification of music emotions. *International Journal of Computational Intelligence Systems*, 5(4), 668-678. doi: 10.1080/18756891.2012.718113

## **EKLER**

**EK 1**      Rassal Veri Seti

## EK 1 Rassal Veri Seti

Öznitelik 1	Öznitelik 2	Sınıf
8	5	1
9	4	1
8	3	1
10	5	1
9	5	1
5	5	1
9	4	1
6	3	1
6	4	1
5	5	1
10	13	2
16	15	2
11	15	2
10	7	2
18	7	2
15	10	2
13	15	2
16	6	2
13	7	2
11	7	2
22	13	3
20	23	3
22	25	3
22	25	3
17	22	3
24	20	3
16	14	3
24	19	3
22	19	3
17	17	3

## ÖZGEÇMİŞ

Adı Soyadı : Umut AKCAN  
Doğum Yeri ve Tarihi : İNEGÖL 07.09.1994  
Yabancı Dil : İngilizce

Eğitim Durumu  
Lise : İnegöl Turgutalp Anadolu Lisesi, 2012  
Lisans : Eskişehir Osmangazi Üniversitesi – Endüstri  
Mühendisliği, 2016  
Yüksek Lisans : Bursa Uludağ Üniversitesi – Endüstri Mühendisliği, 2022

Çalıştığı Kurum/Kurumlar : -

İletişim (e-posta) : umutakcan656@gmail.com

Yayımları :