KABUK TİPİ SENTETİK VERİ SETLERİNDE
ÖZGÜN BİR ANALİZ ÇALIŞMASI

YÜKSEK LİSANS TEZİ

Feyza Nur ÖZDEN

Danışman
Dr. Öğr. Üyesi Güray SONUGÜR

MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Şubat 2024

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

KABUK TİPİ SENTETİK VERİ SETLERİNDE ÖZGÜN BİR ANALİZ ÇALIŞMASI

Feyza Nur ÖZDEN

Danışman
Dr. Öğr. Üyesi Güray SONUGÜR

MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Şubat 2024

TEZ ONAY SAYFASI

Feyza Nur ÖZDEN tarafından hazırlanan "Tez Onay Sayfası" adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 09 / 02 / 2024 tarihinde aşağıdaki jüri tarafından **oy birliği** ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **Mekatronik Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman **:** Dr.Öğr.Üyesi Güray SONUGÜR

Başkan **:** Doç.Dr. Barış GÖKÇE
Necmettin Erbakan Üniversitesi, Mühendislik Fakültesi

Üye **:** Dr.Öğr.Üyesi Güray SONUGÜR
Afyon Kocatepe Üniversitesi, Teknoloji Fakültesi

Üye **:** Dr.Öğr.Üyesi Celal Onur GÖKÇE
Afyon Kocatepe Üniversitesi, Teknoloji Fakültesi

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
..... /..... /..... tarih ve
……. sayılı kararıyla onaylanmıştır.

…….
Prof. Dr. Bekir YALÇIN
Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİM SAYFASI

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- 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.

09 / 02 / 2024

İmza

Feyza Nur ÖZDEN

ÖZET

Yüksek Lisans Tezi

KABUK TİPİ SENTETİK VERİ SETLERİNDE ÖZGÜN BİR ANALİZ ÇALIŞMASI

Feyza Nur ÖZDEN

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

Mekatronik Mühendisliği Anabilim Dalı

Danışman: Dr.Öğr. Üyesi Güray SONUGÜR

Bu çalışmada, kabuk tipi sentetik veriler (KTSV) üzerinde belirlenen parametrelere göre K-means yöntemi kullanılarak özgün bir kümeleme analizi yapılmıştır. Gerçek veri setlerinin genellikle çok boyutlu olması ve bu nedenle görselleştirilmeleri mümkün olmaması nedeniyle iki boyutlu sentetik veri setleri kullanılmıştır. Çok boyutlu veri setlerinde veri noktalarının çoğunlukla dış kabuk kısmında yoğunlaşması nedeniyle veri setleri kabuk tipinde oluşturulmuştur. Görüntü işleme çalışmalarında sıklıkla kullanılan ve çoğunlukla nesne kümelerinin en dış katmanlarında veri bulunduran kenarları çıkarılmış imgeler kabuk tipi verilere örnek olarak verilebilir. Gerçekleştirilen analiz çalışması ile iki boyutlu uzayda oluşturulan farklı topolojilerdeki kabuk tipi veri setlerinin; kümeler arası boşluk, küme içi boşluk, küme sayısı, kabuk kalınlığı, veri sayısı ve iç içe küme sayısı parametrelerine göre en başarılı kümeleme performansları işlem zamanı ve doğruluk ölçütleri ile gözlenerek optimum veri seti yapıları bulunmaya çalışılmıştır. Ayrıca, kullanılan KTSV'lerin çarpıklık ve basıklık gibi istatistiksel özellikleri ile kümeleme performansı arasındaki bağlantı araştırılmıştır. Gerçekleştirilen deneylerin sonuçları tablolar ve grafikler halinde sunulmuştur.

2024, xi + 75 sayfa

Anahtar Kelimeler: Kümeleme, K-means, Kabuk tipi veri seti, Sentetik veri seti.

ABSTRACT

M.Sc. Thesis

A NOVEL ANALYSİS ON SHELL TYPE SYNTHETİC DATA SETS

Feyza Nur ÖZDEN

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Mechatronic Engineering

Supervisor: Asst. Prof. Güray SONUGÜR

In this study, a novel clustering analysis was performed on shell-type synthetic data (KTSV) using the K-means method according to specified parameters. Two-dimensional synthetic data sets were used because real data sets are usually multidimensional and therefore cannot be visualized. Since the data points in multidimensional datasets are mostly concentrated in the outer shell, the datasets were created in shell type. Edge-extracted images are an example of shell-type data since they are commonly employed in image processing research and typically contain data in the outside layers of object sets. With the analysis study, the most successful clustering performances of shell type data sets with different topologies created in two-dimensional space according to the parameters of inter-cluster spacing, intra-cluster spacing, number of clusters, shell thickness, number of data and number of nested clusters were observed with computing time and accuracy criteria and the optimum data set structures were tried to be found. In addition, the relationship between the statistical properties of the used KTSVs such as skewness and kurtosis and clustering performance is investigated. The results of the experiments are presented in tables and graphs.

2024, xi + 75 pages

Keywords: Clustering, K-means, Shell type dataset, Synthetic dataset.

TEŞEKKÜR

Bu araştırmanın konusu, deneysel çalışmaların yönlendirilmesi, sonuçların değerlendirilmesi ve yazımı aşamasında yapmış olduğu büyük katkılarından dolayı tez danışmanım Sayın Dr.Öğr.Üyesi Güray SONUGÜR, literatürde bu çalışmanın yer almasını tavsiye eden ve desteklerini esirgemeyen Sayın Dr.Öğr.Üyesi Celal Onur GÖKÇE'ye her konuda öneri ve eleştirileriyle yardımlarını gördüğüm hocalarıma ve arkadaşlarıma teşekkür ederim.

Hayatım boyunca maddi ve manevi her konuda beni destekleyen annem Leman ÖZDEN'e, babam Selami ÖZDEN'e, abim Osman Bahadır ÖZDEN'e ve teyzem Cemile SELEN'e teşekkür ederim.

Feyza Nur ÖZDEN
Afyonkarahisar 2024

İÇİNDEKİLER DİZİNİ

Sayfa

SİMGELER ve KISALTMALAR DİZİNİ

Kısaltmalar

AAA	Ailevi Akdeniz Ateşi
AEFS	AutoEncoder Feature Selector
CLARANS	Clustering Large Applications based on Randomized Search
COVID-19	Coronavirus Disease 2019
ÇKT	Çok Katmanlı Topoloji
DBSCAN	Density Based Spatial Clustering of Applications with Noise
GUI	Graphical User Interface
İHA	İnsansız Hava Aracı
ID3	Iterative Dichotomiser 3
IDE	Integrated Development Environment
KTSV	Kabuk Tipi Sentetik Veri Seti
OECD	Ekonomik İş birliği ve Kalkınma Teşkilatı
PSO	Parçacık Sürü Optimizasyonu
SPYDER	Scientific Python Development Environment
TİT	Tam İdrar Tahlili
TKT	Tek Katmanlı Topoloji
WEKA	Waikato Environment for Knowledge Analysis

ŞEKİLLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

1. GİRİŞ

Günümüz teknolojik gelişmelerin hızla ilerlemesi her alanda büyük miktarda veri oluşumunu beraberinde getirmektedir. Günümüzde artan veri miktarı, sağlık hizmetlerinden finans sektörüne, eğitimden üretim endüstrisine kadar çeşitli sektörlerde oluşmaktadır. Bu durum, veri bilimi ve analizinin, bu verileri gruplamak, anlamlı bilgiler elde etmek ve karar alma süreçlerini güçlendirmek adına önemli bir rol oynar.

Veri analizi, bu büyük veri setlerindeki gizli desenleri, eğilimleri ve ilişkileri ortaya çıkarmayı amaçlar. Kümeleme analizi, bu süreçte önemli bir araç olarak öne çıkar. Kümeleme analizi, benzer özelliklere sahip veri noktalarını aynı grupta birleştirmek amacıyla kullanılan bir tekniktir. Bu analiz yöntemi, benzer veri noktalarını belirli kümeler içinde gruplandırarak farklı veri gruplarını tanımlar. Pazarlama, müşteri segmentasyonu, sağlık ve birçok diğer alan, kümeleme analiziyle değerli bilgiler elde etmek için kullanılabilir. En yaygın kullanılan kümeleme algoritmaları arasında K-means (ortalamalar), DBSCAN (Density Based Spatial Clustering of Applications with Noise) ve Gaussian Mixture Model bulunmaktadır. Her bir algoritma, farklı veri yapılarına uygun şekilde kullanılabilmektedir.

Gelişmiş bir kümeleme algoritması olarak bilinen K-means, veri bilimi disiplini içinde önemli bir yere sahiptir ve çeşitli uygulama alanlarında kullanılmaktadır. Bu algoritma, veri setindeki benzer özelliklere sahip gözlemleri belirli sayıda küme içinde gruplayarak, her kümenin merkezini temsil eden noktaları optimize etmeye çalışır. Kullanıcıların, algoritmanın veri setlerine uygun parametreleri seçmeleri ve elde edilen sonuçları doğru bir şekilde yorumlamaları önemlidir. Bu sayede, K-means algoritması, veri setlerindeki önemli yapıları ortaya çıkarmak ve verilerden anlamlı bilgiler elde etmek için güçlü bir araç olarak kullanılabilir.

Sentetik veri setleri, gerçek dünya verilerini taklit eden ve genellikle belirli bir problemi çözmek veya bir algoritmayı eğitmek amacıyla oluşturulan yapay veri kümeleridir. Bu veri setleri, çeşitli alanlarda, özellikle de veri bilimi, makine öğrenimi ve yapay zekâ gibi disiplinlerde yaygın olarak kullanılır.

Gerçek veri setleri genellikle yüksek boyutludur, bu nedenle bu verilerin görselleştirilmesi zordur. Ancak, sentetik veri setleri kullanılarak üretilen veriler, üç boyuta kadar kolayca görselleştirilebilir ve analiz edilebilir. Bu, geliştirilen algoritmaların etkili bir şekilde test edilmesini sağlar.

Bu çalışma, veri bilimi, makine öğrenimi ve yapay zekâ alanlarında önemli bir konumda bulunan K-means kümeleme algoritmasının, çalışma kapsamında geliştirilen kabuk tipi sentetik veriler üzerinde gerçekleştirilen kapsamlı ve özgün bir kümeleme analizini ele almaktadır. K-means algoritması, veri noktalarını belirli sayıda küme içinde düzenleyerek her bir kümenin merkezini temsil eden noktaları optimize etmeye yönelik güçlü bir algoritmadır. Kabuk tipi sentetik veriler, özel olarak oluşturulmuş ve belirli özellikleri simüle etmek amacıyla tasarlanmış veri setleridir. Görüntü işleme çalışmalarında, bu tür verilerin kullanımı, görsellerin kenarlarının belirlenmesi gibi önemli görevlerde veri yoğunluğunu etkili bir şekilde yansıtabilir. Bu çalışma, kabuk tipi verilerin görüntü uygulamaları ile birlikte K-means algoritmasının kabuk tipi sentetik verilerle gerçekleştirdiği kümeleme analizini detaylı bir şekilde inceleyerek, algoritmanın performansını ve etkinliğini bu özel veri türü bağlamında değerlendirmeyi amaçlamaktadır.

Önerilen çalışma ile geliştirilen kabuk tipi sentetik veriler üzerinde, K-means kümeleme algoritmasının başarılı bir şekilde uygulanmasının yanı sıra, algoritmanın bu veri setlerine uygun parametrelerin belirlenmesi ile kapsamlı bir değerlendirme sunmaktadır. Bu bağlamda oluşturulan kabuk tipi sentetik verilerin, diğer mevcut algoritmalar veya geliştirilen özel algoritmaların deneyleri ve performans analizlerine de katkı sağlayacağı düşünülmektedir.

Bu çalışmanın ilk bölümü, çalışma kapsamında genel bir tanıtım içermektedir. İkinci bölümde, çalışma alanındaki diğer araştırmalar incelenmekte ve güncel bir literatür taraması sunulmaktadır. Bu bağlamda, ilgili referanslara atıf yapılarak, mevcut bilgi birikimi anlatılmaktadır. Üçüncü bölümde, kullanılan materyal ve metot detaylı bir şekilde anlatılmaktadır. Özellikle, çalışmanın odak noktasını oluşturan kabuk tipi sentetik

verileri geliştirmek için kullanılan formüller, özgün kümeleme analizini belirleyen parametreler, kümeleme performansı için kullanılan istatiksel ölçütler ve kabuk tipi veri setlerinin görüntü uygulamaları bu başlık altında ayrıntılı olarak ele alınmaktadır. Dördüncü bölüm, çalışma kapsamında elde edilen bulgulara ayrılmıştır. İki topolojide oluşturulan kabuk tipi veri seti ile çalışmanın sonuçları ve bu sonuçlar doğrultusundaki belirlenen uygun parametrelere göre özgün kümeleme analizleri şekiller ve çizelgeler halinde sunulmuştur. Beşinci bölümde ise çalışmanın etkinliği değerlendirilerek, elde edilen sonuçlara ilişkin derinlemesine bir analiz sunulmaktadır. Altıncı bölüm, çalışma kapsamında başvurulan kaynakları içermektedir.

2. LİTERATÜR BİLGİLERİ

2.1 Kümeleme

Kümeleme, eğiticisiz öğrenme (unsupervised learning) alanında kullanılan bir veri analizi tekniğidir. Bu teknik, veri noktalarını benzerliklerine göre gruplara veya "kümelere" ayırarak veriyi anlamamıza yardımcı olan bir süreci ifade eder. Kümeleme, verilerin etiketlenmemiş olduğu veri setlerinde kullanılır. Temel amacı, veri içindeki benzerlikleri ve farklılıkları keşfetmek ve veriyi yapılandırmaktır. Kümeleme, makine öğrenimi, veri madenciliği, örüntü tanıma, görüntü analizi ve biyoinformatik dahil olmak üzere birçok alanda kullanılan bir tekniktir. Benzer nesnelerin tanımlanmış bazı mesafe ölçülerine göre farklı gruplar veya alt kümeler halinde gruplandırılmasına çabalar. Kümeleme için hiyerarşik, parçalı, ızgara, yoğunluk tabanlı ve model tabanlı olmak üzere farklı yöntemler vardır. Bu yöntemlerde kullanılan yaklaşımlar Saxena (2017)' da ele alınmıştır. Kümeleme birçok disiplinle ilgilidir ve Rajabi (2017)'de tartışıldığı gibi elektrik güç sistemleri, biyolojik veriler, tıbbi teşhisler, görüntü bölütleme, müşteri bölümleme ve yük profili oluşturma da dahil olmak üzere geniş bir uygulama yelpazesinde önemli bir rol oynamaktadır. Kümeleme, eğiticisiz öğrenme yöntemleri içindeki temel bir tekniktir. Etiketlenmemiş veri ile çalışma yeteneği, bu yöntemi çok çeşitli uygulama alanlarında değerli kılar. Verilerin içerdiği yapıları ve ilişkileri anlama yeteneği, veri madenciliği projelerinin başarısını artırır ve karar verme süreçlerindeki bilgi eksikliklerini giderir. Sonuç olarak, kümeleme, veri bilimcileri ve araştırmacılar için vazgeçilmez bir araçtır ve veri analizi süreçlerinin temelini oluşturur (Berkhin 2006).

2.2 Kümeleme Yöntemleri

Kümeleme yöntemleri matematiksel model temelli ve öğrenme temelli yöntemler olarak başlıca iki sınıfta tartışılabilir. Model tabanlı kümeleme, kümeler ve karışım modeli bileşenleri arasında bire bir uyum olduğunu varsayan ve yaygın olarak kullanılan bir gruplama tekniğidir (Zhu ve Melnykov, 2015). Yapılan çalışmalar incelendiğinde, Gormley vd. (2023) model tabanlı kümeleme teorisine ve çıkarımsal yaklaşımlara genel bir bakış sunmakta ve çok çeşitli veri türleri için model tabanlı kümelemenin kullanımını

kolaylaştıran son metodolojik gelişmeleri vurgulamaktadır. Bouveyron ve Brunet (2014), çok boyutlu verilerin model tabanlı kümelenmesi için boyut azaltma yaklaşımlarını, düzenlemeye dayalı teknikleri, ayrıştırıcı modellemeyi, alt uzay kümeleme yöntemlerini ve değişken seçimine dayalı kümeleme yöntemleri üzerinde çalışmıştır. Jajuga (2005), hibrit model tabanlı kümeleme algoritmalarının sınırlamalarına ve bazı güncel sorunlara odaklanarak kümeleme yöntemlerini ve bunların tipik algoritmalarını analiz etmektedir. Öğrenme tabanlı yöntemleri kullanan çalışmalar incelendiğinde, Khaleel (2022) yapay zekâ kullanarak farklı kümeleme yöntemlerinin bir incelemesini sunmakta ve her veri seti için en iyi performans gösteren kümeleme algoritmasını vurgulamaktadır. Hançer vd. (2012), Yapay Arı Kolonisi tabanlı bir görüntü kümeleme yöntemi önermekte ve performansını K-ortalamalar ve Parçacık Sürü Optimizasyonu (PSO) algoritmaları ile karşılaştırmaktadır. Genel olarak, bu makaleler veri madenciliği, görüntü analizi ve nesnelerin interneti sistemleri de dahil olmak üzere çeşitli uygulamalarda yapay zekâ tabanlı kümeleme yöntemlerinin kullanışlılığını göstermektedir.

2.3 Sentetik Veri Setlerinin Kümeleme Açısından Önemi

Sentetik veri setleri, veri madenciliği ve makine öğrenimi alanlarında önemli bir rol oynamaktadır. Bu veri setleri, gerçek dünya verilerinin bazı özelliklerini taklit eden, yapay olarak oluşturulan verilerdir. Sentetik veri setleri, kümeleme yöntemlerinin performansını değerlendirmek ve yeni kümeleme yöntemleri geliştirmek için kullanılır (Han vd. 2022). Bir kısım avantajları aşağıda verilmiştir:

- Gerçek veri kümelerinin sınırlamalarını ortadan kaldırır. Gerçek veri kümeleri, genellikle eksik, gürültülü veya dengesiz olabilir. Sentetik veri setleri, bu sınırlamaları ortadan kaldırarak kümeleme yöntemlerinin daha doğru ve tutarlı bir şekilde değerlendirilmesini sağlar. Kümeleme yöntemlerinin anlaşılmasına yardımcı olur. Sentetik veri setleri, kümeleme yöntemlerinin nasıl çalıştığına dair daha derin bir anlayış kazanmak için kullanılabilir. Bu, kümeleme yöntemlerinin daha iyi kullanılmasına yardımcı olur (Han vd. 2022).
- Yeni kümeleme yöntemlerinin geliştirilmesine yardımcı olur. Sentetik veri setleri, kümeleme yöntemlerinin farklı parametre ve konfigürasyonlarının performansını karşılaştırmak için kullanılabilir. Bu, yeni kümeleme yöntemlerinin

geliştirilmesine yardımcı olur (Jain vd. 1999).

Sentetik veriler çeşitli yöntemlerle oluşturulabilmektedir. Gerçek verilerin, bir kısım özellikleri değiştirilerek veya farklı kaynaklardan alınan verilerin birleştirilmesiyle de sentetik veri setleri oluşturulabilir. Gerçek veriler olmaksızın, gerçek verilerin belirli bir parametresine göre tamamen sentetik veriler de oluşturulabilir (Dandekar vd. 2018).

Literatürde sentetik veri setlerinin önemini vurgulayan çalışmalar incelendiğinde; Pour vd. (2013) ve Szilágyi vd. (2014), kümeleme algoritmalarını test etmek için sentetik veri kümelerinin kullanımını göstermişlerdir. Pour vd. (2013) sentetik sağlık veri kümelerini ve Szilágyi vd. (2014) sentetik protein dizisi verilerini kullanmıştır. Çok boyutlu veri setlerinin kümelenmesi de kümeleme yöntemleri bakımından zorlu bir süreçtir. Armano ve Javerone (2013), çok boyutlu veri kümelerini kümelemek için karmaşık ağ analizine dayalı bir yöntem önermektedir. Genel olarak, makaleler sentetik veri kümelerinin kümeleme algoritmalarını test etmek ve geliştirmek ve kümeleme görevleri için belirli özelliklere sahip veri kümeleri oluşturmak için yararlı olabileceğini göstermektedir.

2.4 Çok Boyutlu Verilerin Kümelenmesi

Çok boyutlu veriler, birden fazla özelliği olan verilerdir. Bu özellikler, sayısal, kategorik veya karma olabilir. Çok boyutlu veriler, veri madenciliği, makine öğrenimi ve doğal dil işleme gibi birçok alanda yaygın olarak kullanılmaktadır. Çok boyutlu verilerin boyut sayısı, özelliklerin sayısıdır. Örneğin, bir veri kümesi iki özellikten (boy ve kilo) oluşuyorsa, bu veri kümesi iki boyutlu olarak kabul edilir. Bazı veri setlerinde bu şekilde yüzlerce boyuttan söz edilebilir. Veri analistleri ancak 3 boyuta kadar olan verileri görselleştirebilmektedir. Sonrasında farklı istatistiksel teknikler kullanılmalıdır.

Çok boyutlu verilerin denetimsiz öğrenmede zorlu bir sorun olarak ortaya çıkar. Çok sayıda özellik veya boyuta sahip verileri ifade eden çok boyutlu veriler, makine öğrenimi gibi alanlarda hesaplama ve analitik zorluklar ortaya çıkarmaktadır (Han vd. 2018). Denetimsiz öğrenmede amaç, etiketli örnekler kullanmadan verinin altında yatan yapıyı veya örüntüleri keşfetmektir.

Denetimsiz öğrenmede bu sorunu ele almak için geliştirilen yaklaşımlardan birisi, boyut azaltma tekniklerinin kullanılmasıdır. Bu teknikler, verilerdeki önemli bilgileri korurken özellik sayısını azaltmayı amaçlar. Teğet uzay hizalama gibi doğrusal olmayan boyutluluk azaltma yöntemleri, çok boyutlu verilerin düşük boyutlu yapısını keşfetmek için önerilmiştir (Zhang ve Zha 2004). Bu yöntemler, verilerin üzerinde bulunduğu içsel yapıyı yakalayan verilerin daha düşük boyutlu bir temsilini bulmayı amaçlamaktadır.
Denetimsiz öğrenmede çok boyutlu verileri ele almak için bir başka yaklaşım da öznitelik seçimidir. Öznitelik seçimi, orijinal çok boyutlu verilerden ilgili özelliklerin bir alt kümesinin seçilmesini içerir. Bu, girdi uzayının boyut sayısını azaltarak makine öğrenimi modellerinin performansını ve etkinliğini artırmaya yardımcı olabilir. AutoEncoder Feature Selector (AEFS) gibi autoencoder'dan ilham alan denetimsiz özellik seçme yöntemleri, çok boyutlu verilerden bilgilendirici özellikleri seçmek için autoencoder regresyonu görevlerini birleştirmek için önerilmiştir (Han vd. 2018).

Bu tip veri setlerinin avantaj ve dezavantajları aşağıda sunulmuştur:
Avantajlar:

- Daha fazla bilgi içerir: Çok boyutlu veriler, tek boyutlu verilere göre daha fazla bilgi içerir. Bu, veri madenciliği ve makine öğrenimi uygulamalarında daha iyi sonuçlar elde edilmesini sağlar.
- Daha karmaşık yapıları temsil edebilir: Çok boyutlu veriler, tek boyutlu verilerden daha karmaşık yapıları temsil edebilir. Bu, doğal dil işleme gibi alanlarda daha iyi sonuçlar elde edilmesini sağlar (Witten vd. 2016).

Dezavantajlar:

- Veri analizi zordur: Çok boyutlu veriler, tek boyutlu verilere göre daha karmaşıktır. Bu, veri analizini zorlaştırabilir.
- Veri işleme maliyeti yüksektir: Çok boyutlu veriler, tek boyutlu verilere göre daha fazla işlem gücü gerektirir. Bu, veri işleme maliyetini artırabilir (Witten vd. 2016).

Makine öğrenimi genellikle bu verileri analiz etmek ve yorumlamak için kullanılır, ancak verilerin çok boyutluluğu ve karmaşıklığı zorluklara yol açabilir. Bu zorlukların üstesinden gelmek için araştırmacılar özellik dönüşümü, özellik seçimi ve özellik kodlaması gibi çeşitli yaklaşımlar geliştirmiştir (Aggarwal 2015). Ayrıca, derin öğrenme

ve çok modlu makine öğrenimi gibi çok boyutlu verileri işleyebilen makine öğrenimi modelleri geliştirilmiştir.

Çok boyutlu verilerin bir başka özelliği de aslında verilerin hacminin büyük bir kısmının veri setinin (Gauss dağılımına sahip veriler) dış kabuğunda bulunduğu gerçeğidir (Domingos 2012). Bu durum ışığında çok boyutlu verilerin kümelenmesinde verilerin dış kabuklarının hedeflenmesi kümeleme performansını artıran bir yaklaşım olacağı düşünülmüştür.

2.5 K-means Yöntemi

K-means yöntemi literatürde oldukça fazla incelenmiş ve incelenmeye devam edilen bir konudur. Yönteme çeşitli yönlerden bakılan derleme çalışmaları araştırmacılara katkı sağlamaktadır. Steinley (2006) K-means algoritması üzerine, önceki elli yılda algoritma üzerine yapılan araştırmaları, metodolojiyi ve sonuçları sentezleyen bir derleme çalışması sunmuştur. Ahmed vd. (2020) K-means algoritması varyantlarının basit bir taksonomisini üç başlık altında sunmuştur: başlatma, veri türleri ve uygulamalar. Bock (2008) K-means algoritmasının hem sürekli hem de ayrık varyantları için kümeleme işleminde kullanılan uzaklık belirleme yöntemleriyle ilgili bir derleme çalışması sunmuştur. Ayrıca, bir kısım özel alanlar için gerçekleştirilen derleme çalışmaları da bulunmaktadır. Örneğin, K-means kümelemesinin modern güç sistemlerindeki alternatif kümeleme yöntemleriyle birlikte uygulanmasını vurgulayan pek çok makalenin incelendiği bir çalışma Miraftabzadeh vd. (2023) tarafından gerçekleştirilmiştir. Bu yöntemin, nesnelerin interneti çalışmalarındaki uygulamalarını inceleyen derleme çalışması Gupta ve Chandra (2022), hava kirliliği hakkındaki çalışmaları inceleyen derleme çalışması Govender ve Sivakumar (2020) ve rüzgâr hızı ve yönüne dayalı rüzgâr kümeleme süreçlerindeki uygulamalarını inceleyen derleme çalışması Azhar ve Hashim (2023) tarafından gerçekleştirilmiştir.

Bu konudaki bir kısım araştırma makaleleri hakkında yapılan detaylı inceleme çalışması bir sonraki başlıkta sunulmuştur.

2.5.1 Tıp Alanında Yapılan Çalışmalar

Yıldız ve Ceyhan (2023) tarafından yapılan çalışmada K-means ve Kruskal Wallis testi yöntemleri kullanılarak Ailevi Akdeniz Ateşi (AAA) hastalığına ait bir veri seti üzerinde özellik seçimi ile detaylı bir analiz gerçekleştirmişlerdir. Önerilen yöntemde "atak süresi", "ateş" ve "TİT protein" adlı belirlenen üç özellik seçimi üzerinde K-means kümeleme yöntemi ile %70,14 doğruluk oranı elde etmişlerdir.

Boboyarov ve Mixliyev (2023) tarafından yapılan çalışmada K-means ve global eşikleme yöntemleri kullanılarak mikroskop altında alınan ince kan görüntülerinden kan hücrelerini ayırt etme amacıyla bir yöntem geliştirmişlerdir. Önerilen yöntemde segmentasyon algoritmaları ile %95,5 doğruluğa, K-means algoritması ile %95 doğruluğa erişilmiştir.

Mane (2017) tarafından yapılan çalışmada K-means ve sınıflandırma için karar ağacı algoritması ID3 yöntemleri kullanılarak kalp hastalığının tahmini için bir sistem geliştirilmiştir. Önerilen yöntemde Hadoop MapReduce platformundan elde edilen büyük veriler ile en yüksek doğruluk değerlerine 5000 ve 15000 boyutlarındaki veriler için K-means algoritmasıyla % 95,18'e, geliştirilen K-means algoritması ile %98,28 doğruluk oranı elde edilmiştir.

2.5.2 Mühendislik Alanında Yapılan Çalışmalar

Ikotun vd. (2023) tarafından yapılan çalışmada büyük veri çağındaki gelişmeler ile K-means kümeleme algoritmaları için kapsamlı bir inceleme ve değişken analizi yapmışlardır. Çeşitli K-means algoritma varyantları implementasyonlarının doğrulanması için kullanılan 60 adet standart veri setinin detaylarını tablo halinde sunmuşlardır. Önerilen çalışma ile K-means algoritmalarında, başlatma problemlerinin çözümüne daha fazla yoğunlaşıldığı ve karışık veri tipi problemi çok az incelendiği sonucuna ulaşılmıştır.

Aslanyürek ve Mesut (2021) ‘un “Kümeleme Performansını Ölçmek için Yeni Bir Yöntem ve Metin Kümeleme için Değerlendirmesi” başlığı ile yaptıkları çalışmada K-means, K-medoids ve CLARANS yöntemleri kullanılarak kümeleme performansını ölçmek amacıyla yeni bir yöntem önermiştir. Önerilen yöntem İngilizce ve 6 dil veri setlerinde test edilerek, 6 dil veri setinde K-means yöntemi ile 0,9973 doğruluğa, K-medoids yöntemi ile 0,9914 doğruluğa, CLARANS (Clustering Large Applications based on Randomized Search) yöntemi ile 0,9898 doğruluğa erişilmiştir.

Ghazal vd. (2021) tarafından yapılan çalışmada K-Means kümeleme algoritmasının farklı mesafe metrikleriyle performans analizleri yapılmıştır. Önerilen çalışmada farklı veri setleri ve farklı küme sayıları açısından üç farklı matematiksel mesafe metriği (Minkowski, Manhattan ve Öklid) kullanılarak yürütme süresi bakımından değerlendirilmiştir. Bu çalışmadaki deneylere göre Manhattan mesafesi 4, 8, 12 ve 14 adet küme için daha iyi performans elde edilmiştir. Öklid mesafesi 6 ve 16 adet küme için iyi sonuç elde edilirken, Minkowski mesafesi ile 10 adet küme için daha iyi performans elde edilmiştir. Önerilen çalışmada, K-means kümeleme algoritmasında Manhattan mesafe ölçüm metriğinin çoğu durumda en iyi sonuçları elde ettiği sonucuna erişilmiştir.

2.5.3 Sosyal Bilimler Alanında Yapılan Çalışmalar

Selcan (2021) tarafından yapılan çalışmada K-means kümeleme algoritması yöntemi ile OECD (Ekonomik İş birliği ve Kalkınma Teşkilatı) ülkelerinin vergi yükü ve kayıtdışı ekonomi analizi yapılmıştır. Önerilen çalışmada vergi yükü ve kayıtdışı ekonomi verileri üzerinde Meksika, Kolombiya, Kore, Kosta Rika ve Türkiye’nin diğer ülkelere göre ayrı olduğu elde edilmiştir. Analizler sonucu bu ülkelerin sübjektif vergi yükünün fazla olduğu yorumu yapılabileceği belirtilmiştir.

Demircioğlu ve Eşiyok (2020) tarafından yapılan çalışmada K-means yöntemi kullanılarak, COVID-19 (Coronavirus Disease-2019) salgını ile mücadele için sağlık verileri doğrultusunda ülkeler kümelenmiştir. WEKA (Waikato Environment for Knowledge Analysis) programı üzerinde 36 ülke 2,3 ve 4 adet kümeye ayrılmıştır. 36

ülkenin birbirine göre sağlık göstergeleri açısından benzerlikleri değerlendirilerek sağlık hizmet değerlerinin performans ölçümü çalışmalarına yön vereceğini belirtmişlerdir.

Özari ve Özge (2018) tarafından yapılan çalışmada K-means ve çok boyutlu ölçekleme yöntemi ile illerin yaşam endeksi göstergelerinin kümeleme analizi yapılmıştır. Çok boyutlu ölçekleme analizi ile İstanbul ilinin konum olarak farklılık gösterdiği elde edilmiştir. K-means kümeleme analizinde küme sayısı 6 olarak belirlenmiştir. Birbirine benzer iller olarak İstanbul tek başına olmak üzere, Ankara, Antalya, İzmir ve Muğla illeri bir kümede yer almıştır.

3. MATERYAL ve METOT

Önerilen çalışmanın aşamaları;

1) Kabuk tipi sentetik verinin (KTSV) oluşturulması
2) K-means kümeleme yöntemi ile verilerin uygun kümelere eklenmesi
3) Parametrelerin değiştirilmesi ile özgün kümeleme analizinin gerçekleştirilmesi
4) Kabuk tipi verilerin görüntüler üzerinde uygulanması

olarak belirlenmiştir.

Bu bölüm altında gerçekleştirilen tüm aşamalar açıklanmıştır.

3.1 Python Programlama Dili

Python programlama dili, geniş bir kullanıcı kitlesine hitap eden ve çeşitli uygulama alanlarında kullanılabilen, yüksek seviyeli, etkileşimli ve yorumlanabilir bir dildir. Python'un okunabilir ve basit sözdizimi, kodun hızlı bir şekilde yazılmasını sağlar. Bu özellik, özellikle yeni başlayanlar ve karmaşık projelerde çalışanlar için büyük bir avantajdır. Python, nesne yönelimli, prosedürel ve fonksiyonel programlama paradigmalarını destekler (İnt.Kyn.2). Bu, farklı programlama stillerini kullanarak çeşitli projelere uygun esneklik sağlar. Python'un zengin standart kütüphanesi, birçok işlevi yerine getirmenizi sağlar. Bu kütüphane, dosya işlemlerinden ağ programlamaya, veri tabanı işlemlerinden GUI (Graphical User Interface) uygulamalarına kadar birçok konuda hazır çözümler sunar. Python, geniş bir ve aktif bir topluluğa sahiptir. Bu topluluk, belgeler, forumlar, kütüphaneler ve çeşitli kaynaklar sağlayarak öğrenmeyi ve projeler üzerinde çalışmayı kolaylaştırır. Python, modüler programlamayı destekler. Modülerlik, kodun parçalara bölünmesini, bakımını ve tekrar kullanılmasını kolaylaştırır. Python, yapay zekâ, veri bilimi ve makine öğrenimi gibi alanlarda yaygın olarak kullanılmaktadır. Birçok popüler kütüphane (örneğin, TensorFlow, PyTorch) Python dilinde geliştirilmiştir.

Python, geniş bir standart kütüphane içerir. Bu kütüphane, dosya işleme, ağ programlaması, veritabanı etkileşimi, GUI oluşturma ve daha birçok alanda işlevselliği

içerir. Harici kütüphaneler ve modüller eklenilerek, problemlere uygun çözümler oluşturabilmektedir.

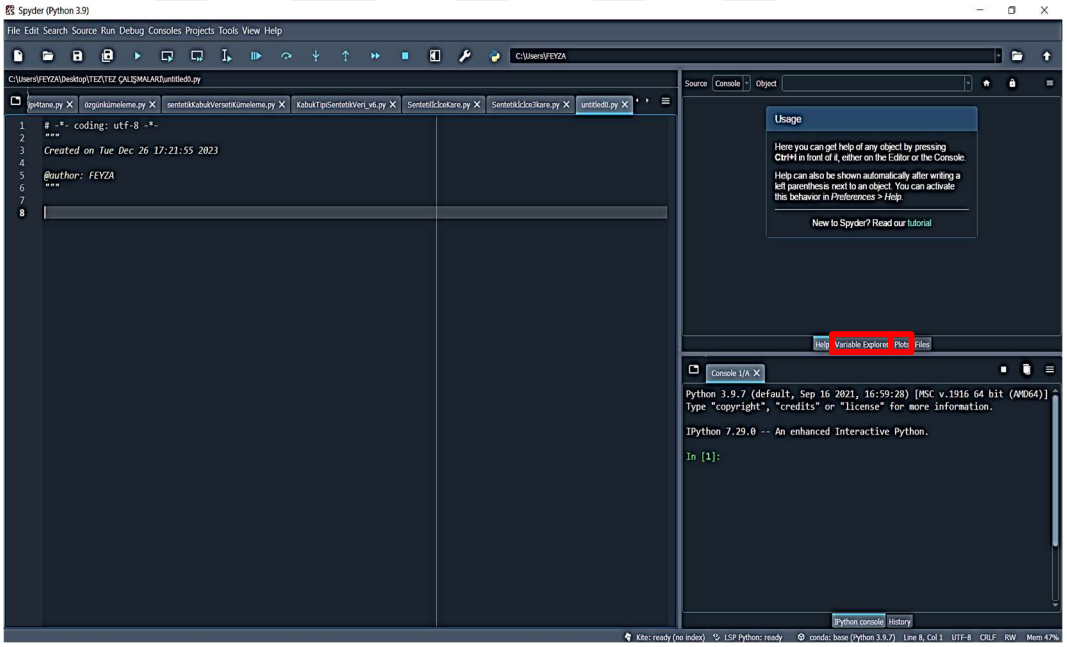
Önerilen çalışmada kullanılan Python kütüphaneleri aşağıda sunulmuştur.

- **Numpy ;** çok boyutlu dizilerle çalışma imkanı tanıyan ve bilimsel hesaplamalarda kullanılan bir kütüphanedir. Lineer cebir, istatiksel işlemler için gerekli fonksiyonları sağlar. Matris çarpımı, determinant hesaplama, matematiksel denklemler gibi işlemleri destekler.
- **Matplotlib;** Python'da veri görselleştirmesi yapmak için kullanılan bir kütüphanedir. Bu kütüphane, grafikler, çizimler ve plotlar oluşturmak için çeşitli fonksiyonlar içerir. Veri analizi, bilimsel hesaplamalar ve raporlama gibi birçok alanda yaygın olarak kullanılır.
- **Random**; rastgele sayılar üretmek ve rastgele olayları simüle etmek için kullanılan bir kütüphanedir. Rastgele sayılar kullanılan karmaşık çalışmalarda, farklı kombinasyonlarda kullanılabilir.
- **Math**; matematiksel işlemler yapmak için kullanılan bir kütüphanedir. Bu kütüphane, birçok matematiksel fonksiyonu içerir. Sayılar üzerinde çeşitli işlemleri gerçekleştirmek için kullanılır.
- **Time**; süre ölçümü, zaman aralıkları ve saat hesaplamaları gibi işlemleri yapmak için bir dizi fonksiyon içerir.
- **OpenCV**; bilgisayarlı görü ve makine öğrenimi uygulamaları geliştirmek için kullanılan bir kütüphanedir. OpenCV, geniş bir uygulama alanıyla görüntü işleme ve bilgisayarlı görü görevlerini destekler ve açık kaynaklıdır.

Anaconda ise Python tabanlı bir veri bilimi ve analitik platformudur. Spyder, Anaconda'nın içinde yer alan bir Python entegre geliştirme ortamıdır (Integrated Development Environment - IDE). Anaconda, sanal ortamları kolayca yönetmenizi sağlar. Bu, farklı projeler için izole çalışma ortamları oluşturarak çatışmaları önler ve projeler arasında geçiş yapmayı kolaylaştırır. Anaconda, Jupyter Notebook, Spyder ve diğer birçok aracı içerir. Bu araçlar, veri analizi, görselleştirme ve model oluşturma gibi veri bilimi görevlerini kolaylaştırır. Spyder (Scientific Python Development Environment), Python diline özgü bir IDE'dir. Anaconda'nın içinde yer alır ve bilimsel

hesaplamalar, veri analizi ve diğer veri bilimi görevleri için özel olarak tasarlanmıştır. Spyder, interaktif çalışma alanları, değişken gözlemleme, hızlı kaynak kodu denemeleri gibi veri bilimi çalışmalarını kolaylaştıran özelliklere sahiptir. Bu, interaktif çalışma ortamları ve hızlı prototip oluşturmayı kolaylaştırır. Spyder, gelişmiş bir kod düzenleyici içerir. Otomatik tamamlama, satır numaralandırma, hata vurgulama ve çoklu seviyeli sekme desteği gibi özellikleri barındırır.

Anaconda ve Spyder, özellikle veri bilimi, bilimsel hesaplamalar ve makine öğrenimi gibi alanlarda çalışan geliştiriciler ve veri bilimcileri için güçlü bir kombinasyon sunar. Bu araçlar, çeşitli kütüphanelere hızlı erişim, sanal ortamları yönetme yeteneği ve gelişmiş bir IDE ile veri bilimi projelerini kolaylaştırır. Şekil 3.1'de Spyder platformunun ara yüzü verilmiştir.



Şekil 3.1 Spyder platformunun ara yüzü.

Spyder'ın Variable Explorer özelliği, çalışma alanındaki değişkenlerin gözlemlenmesini, incelenmesini ve etkileşimli olarak yönetmeyi sağlar. Bu panelde, çalışma sürecinde tanımlanan değişkenlerin adları, türleri ve değerleri gibi bilgiler görüntülenir. Variable Explorer, kodu anlamak ve hata ayıklamaya yardımcı bir rol oynar. Bir Python kodu çalıştırıldığında, Variable Explorer, tanımlanan değişkenleri tablo halinde gösterir. Bu, her bir değişkenin güncel değerini görmeyi, türünü anlamayı ve verilerin daha iyi

anlaşılmasını sağlar. Ayrıca, bu kısımda değişkenler üzerinde bazı temel işlemler yapılabilir, değerler düzenlenebilir ve üzerinde analizler yapılabilmektedir.

Plots özelliği, veri görselleştirmesi ve grafik oluşturma amacı taşır. Spyder ile çalışırken, bu panel ile birlikte oluşturulan grafikler ve çizimler görülebilmektedir. Veri analizinde kullanılan matplotlib, seaborn gibi kütüphanelerle oluşturulan grafikler bu panelde gözükmektedir. Bu, özellikle veri analizi sonuçlarının görselleştirilmesinde oldukça kullanışlıdır. Çeşitli grafik seçenekleri, renk düzenlemeleri ve çeşitli özellikleri sayesinde, verileri daha iyi anlamak ve sonuçları görselleştirmek için bu panel kullanılmaktadır.

Bu iki özellik, Spyder ile kodların anlaşılırlığı, değişkenlerin izlenmesi, veri analizi yapma ve sonuçları görselleştirmek için yardımcı olmaktadır. Böylece geliştirme sürecini daha verimli hale getirmektedir. Şekil 3.1'de Spyder ara yüzünde bu iki panel de gösterilmiştir.

Bu çalışmada, Python programlama dili kullanılarak veri analizi ve görselleştirme işlemleri gerçekleştirilmiştir. Çalışma sürecinde Anaconda platformunun bir parçası olan Spyder entegre geliştirme ortamı kullanılmıştır.

3.2 Kümeleme Analizi

Kümeleme, veri madenciliği, istatiksel analiz ve makine öğrenmesinde kullanılan verileri gruplama yöntemidir. Veri noktalarını benzer özelliklerine göre gruplamak ve farklı grupları ayırt etmek amacıyla kullanılır. Böylece veriler daha kolay anlaşılır duruma gelir. Özellikle yüksek boyuttaki verilerin daha kolay analiz edilmesi sağlanır. Kümeleme işlemi, veri noktaları arasındaki benzerlik veya mesafe ölçümlerine göre gerçekleştirilir. Etiketler veya sınıflar olmadan, verilerin benzerlikleri temel alınarak gruplama işlemi yapılır. Önceden bilinmeyen veya etiketlenmemiş veri setleri üzerinde kümeleme yöntemi kullanılabilir.

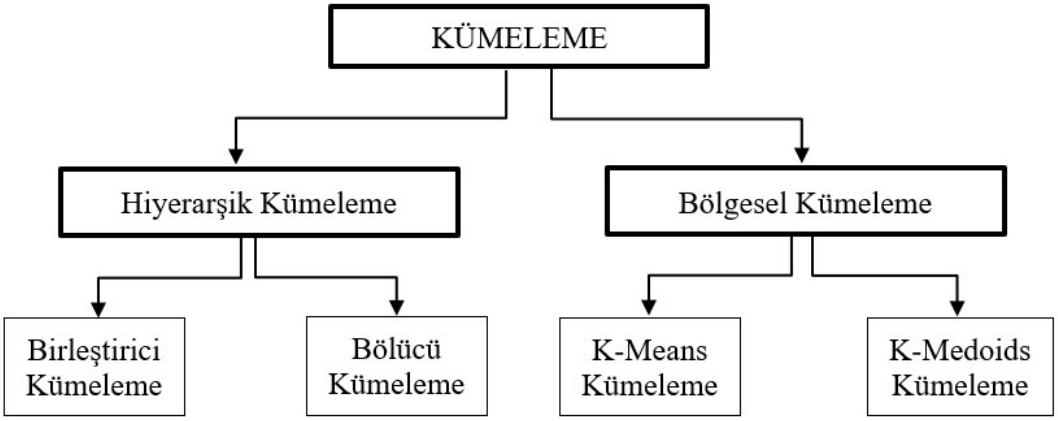
Kümeleme, bir denetimsiz öğrenme yöntemidir. Denetimsiz öğrenme, makine öğrenmesi alanında kullanılan bir öğrenme türüdür. Bu yaklaşım, modelin eğitildiği veri setinde belirli bir çıkışa (etiket veya hedef) dayalı rehberlik olmaksızın, veri setindeki desenleri ve yapıları keşfetmeye odaklanır. Denetimsiz öğrenme, özellikle veri setlerinde gizli bilgilerin ortaya çıkarılması ve anlaşılırlığının geliştirilmesi açısından önemlidir. Denetimsiz öğrenme, veri bilimi, keşifsel veri analizi ve örüntü tanıma gibi birçok uygulama alanında önemli bir rol oynamaktadır. Modelin kendi içinde desenleri keşfetmesi, veri setlerinde gizli yapıları anlamak ve bu yapıları kullanarak daha etkili kararlar almak açısından büyük öneme sahiptir.

Kümeleme sürecindeki ilk adım verilerin toplanması ve gereksiz verilerden ayırt edilmesidir. Böylece veri seti işlenmeye hazır duruma getirilir. Kümeleme analizi için veri setindeki öznitelik seçimi belirlenir. Veri setinin işlenmeye hazır hale getirilmesiyle birlikte küme yapısı, küme sayısı tahmini ve diğer faktörlere göre uygun bir kümeleme algoritması seçilir. En yaygın kullanılan kümeleme algoritmaları olarak K-means, Hiyerarşik kümeleme ve DBSCAN bulunur. Seçilen kümeleme algoritmasının yöntemine göre veri noktaları gruplanır. Her veri noktası benzerlik veya mesafe ölçüleri kullanılarak bir kümeye tahsis edilir. Kümeleme sonuçları görselleştirilerek kümelerin ve verilerin özellikleri incelenir.

Kümeleme, hiyerarşik ve bölgesel olmak üzere başlıca iki tiptir. Her iki tip kümeleme yöntemi de veri setindeki benzerliklere dayalı olarak gruplamalar yapmaktadır. Seçilecek yöntem, veri setinin yapısına ve boyutuna bağlı olarak değişebilir.

Hiyerarşik kümeleme, veri noktalarını ağaç benzeri bir yapıda düzenler. Kümeleme sonuçlarına ağaç benzeri bir diyagram aracılığıyla görsel olarak bakma olanağına sahiptir. Ancak, büyük veri setlerinde hesaplama maliyeti artar. Birleştirici ve bölücü olmak üzere iki alt türe ayrılır. Birleştirici Kümeleme, her veri noktasını tek bir küme olarak başlatır ve benzer olan küme veya veri noktalarını birleştirerek devam eder. Bölücü Kümeleme, tüm veri noktalarını bir küme olarak başlatır ve daha sonra bu küme içindeki benzer olmayan grupları ayırarak devam eder.

Bölgesel kümeleme, belirli sayıda küme oluşturmayı hedefleyen bir yöntemdir. Bölgesel kümeleme, büyük veri setlerinde daha hızlı çalışabilir ancak başlangıçta belirlenen küme sayısı önemli bir etken olabilir. Bu tip kümeleme K-means ve K-medoids gibi algoritmalar ile temsil edilir. K-means Kümeleme, veri noktalarını belirli sayıda kümeye ayırmaya çalışır. Her küme, bir merkeze sahiptir ve veri noktaları bu merkezlere olan uzaklıkları temel alınarak gruplandırılır. K-medoids, K-means algoritmasına benzer fakat küme merkezi olarak veri noktaları arasından seçilir. Yani, küme merkezi, gerçek veri noktaları arasından bir tanesi olarak belirlenir. Şekil 3.2'de kümeleme tipleri gösterilmiştir.



Şekil 3.2 Kümeleme tipleri.

3.2.1 K-means Algoritması

K-means algoritması, verileri benzer özelliklerine göre ya da mesafe ölçülerine göre gruplar oluşturmak amacıyla kullanılan bir denetimsiz öğrenme algoritmasıdır. Kullanıcının belirlediği küme (K) sayısı kadar verileri küme içinde gruplar. Veriler, her kümenin kendi merkezi etrafında toplanır. Her kümenin kendi içinde benzerlikleri en fazla, kümeler arası benzerliklerin en az düzeyde olması hedeflenir. En çok uygulanan kümeleme yöntemlerinden biri olan K-means algoritması, çok boyutlu verilerde de hızlı ve etkili bir şekilde kümeleme yapabilir (Dinçer 2006).

Tüm veriler kümelere ayrılırken mesafe ölçüm sonuçlarına göre gruplanır. Mesafe ölçüm tekniği olarak K-means algoritmasında en çok Öklid mesafesi, Manhattan mesafesi ve Minkowski mesafe yöntemi kullanılır. Öklid mesafesi iki boyutlu düzlemdeki iki veri

noktası arasındaki uzaklık Denklem 3.1'de verilmiştir.

$$D(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.1}$$

Yukarıdaki denklemde;

$D(x,y)$: Koordinat düzlemindeki x ve y noktaları arasındaki mesafe

ifade eder.
Manhattan mesafesinde ise;

$$\sum_{i=1}^{n}|x_i - y_i| \tag{3.2}$$

formülü ile hesaplanır.
Minkowski mesafe ölçüm yönteminde aşağıdaki formül kullanılır;

$$(\sum_{i=1}^{n}|x_i - y_i|^p)^{1/p} \tag{3.3}$$

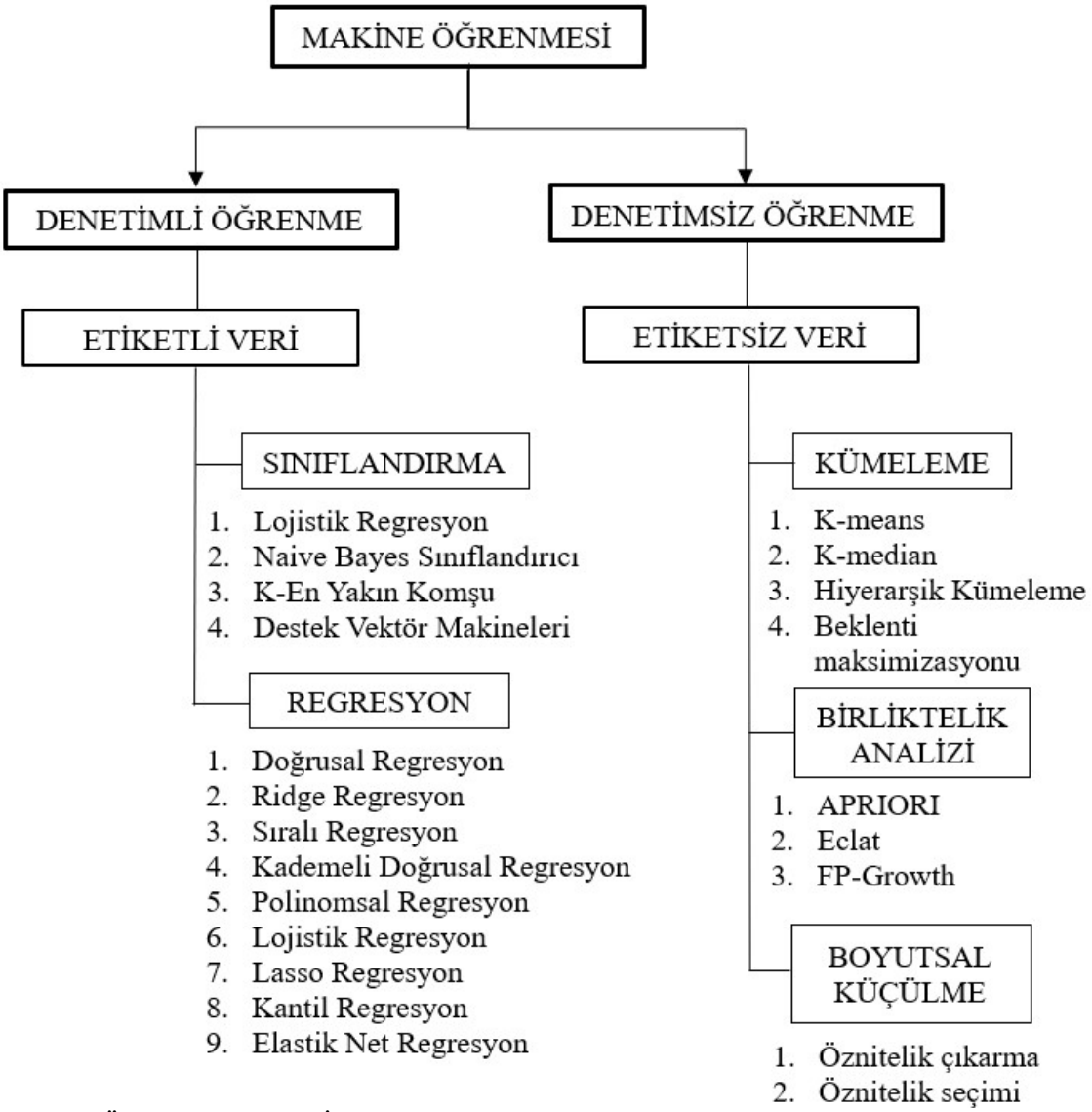
Yukarıdaki denklemde x ve y iki veri noktası olmak üzere p parametresi farklı mesafe ölçüm yöntemleri için uyarlanabilir. Minkowski uzaklığı, $p = 1$ alındığında Manhattan mesafe ölçümü, $p = 2$ alındığında Öklid mesafe ölçümüne karşılık gelir.

Öklid mesafe ölçüm yöntemi K-means algoritmasında en çok kullanılan yöntemdir. Öklid mesafe ölçümü küme merkezi hesaplamalarında kolayca kullanılmaktadır. Fakat veri setleriyle uyumluluğu açısından diğer mesafe ölçümü yöntemleri de kullanılmaktadır. Veri noktalarının özelliklerine uygun mesafe ölçüm yöntemi kullanılır.

Bu çalışmada K-means algoritması ile mesafe ölçümü için Öklid mesafe yöntemi kullanılmıştır.

K-means algoritması etiketsiz verilerle çalışır ve denetimsiz öğrenmedir. Şekil 3.3'te öğrenme çeşitleri detaylı şekilde gösterilmiştir. Bu çalışmada etiketi olmayan KTSV'ler

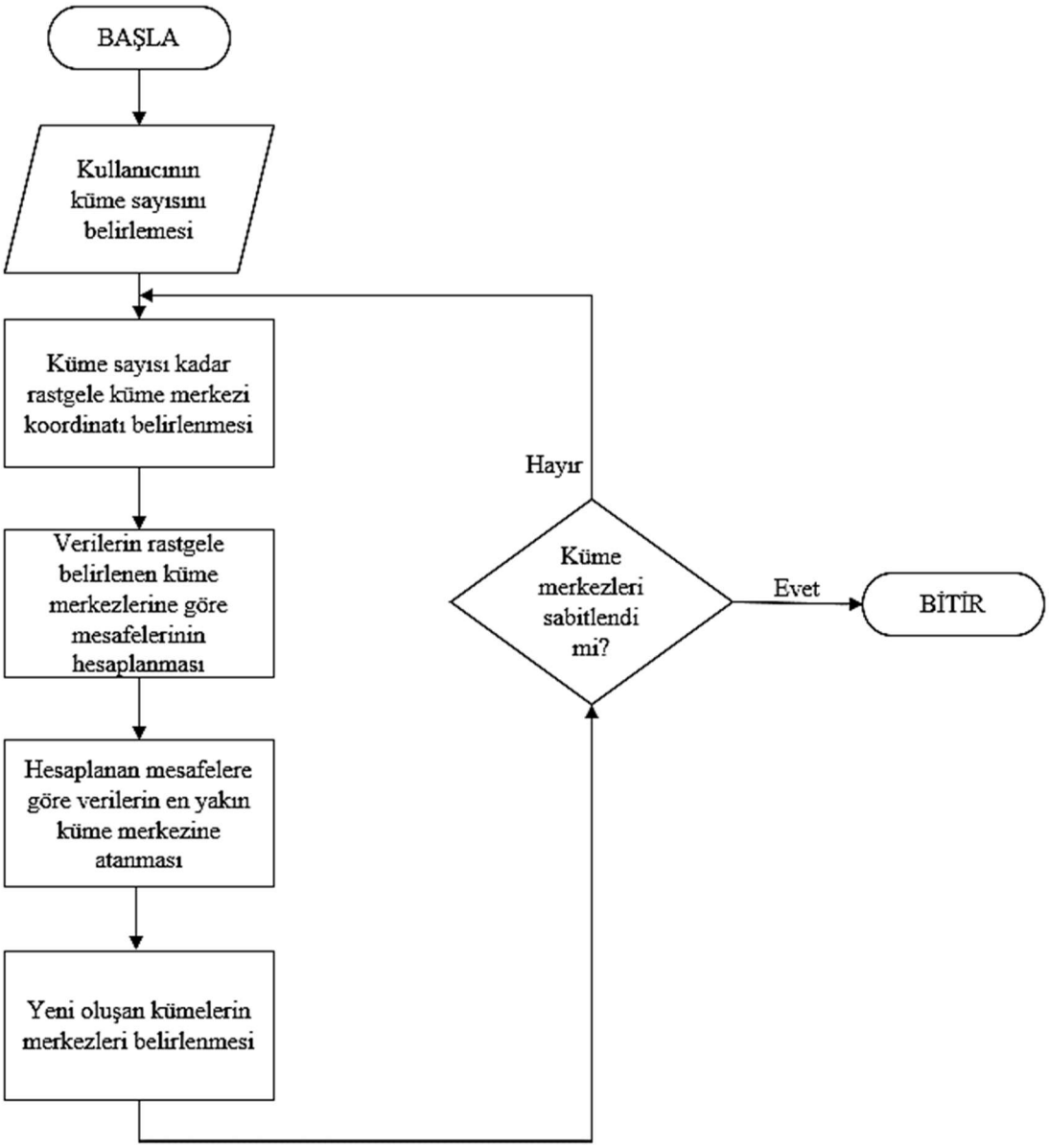
matematiksel yöntemlerle iki boyutlu uzayda oluşturulup K-means algoritmasının bu yapay verileri kümeleme performansı gözlenmiştir.



Şekil 3.3 Öğrenme çeşitleri (İnt.Kyn.1).

K-means algoritmasında kullanıcının belirlediği küme sayısı kadar rastgele küme merkezleri saptanır. Veri noktaları, Öklid mesafe yöntemi ile ölçülen en yakın geçici küme merkezine atanır. Her küme için, merkez kümeye ait veri noktalarının ortalamasına göre yeni bir küme merkezi hesaplanır. Küme merkezlerinin optimize edildiği duruma kadar bu döngü devam eder. Döngü her seferinde yeni küme merkezlerinin hesaplanması ve veri noktalarının yeniden kümelenmesini sağlar. Küme merkezleri optimize edilip

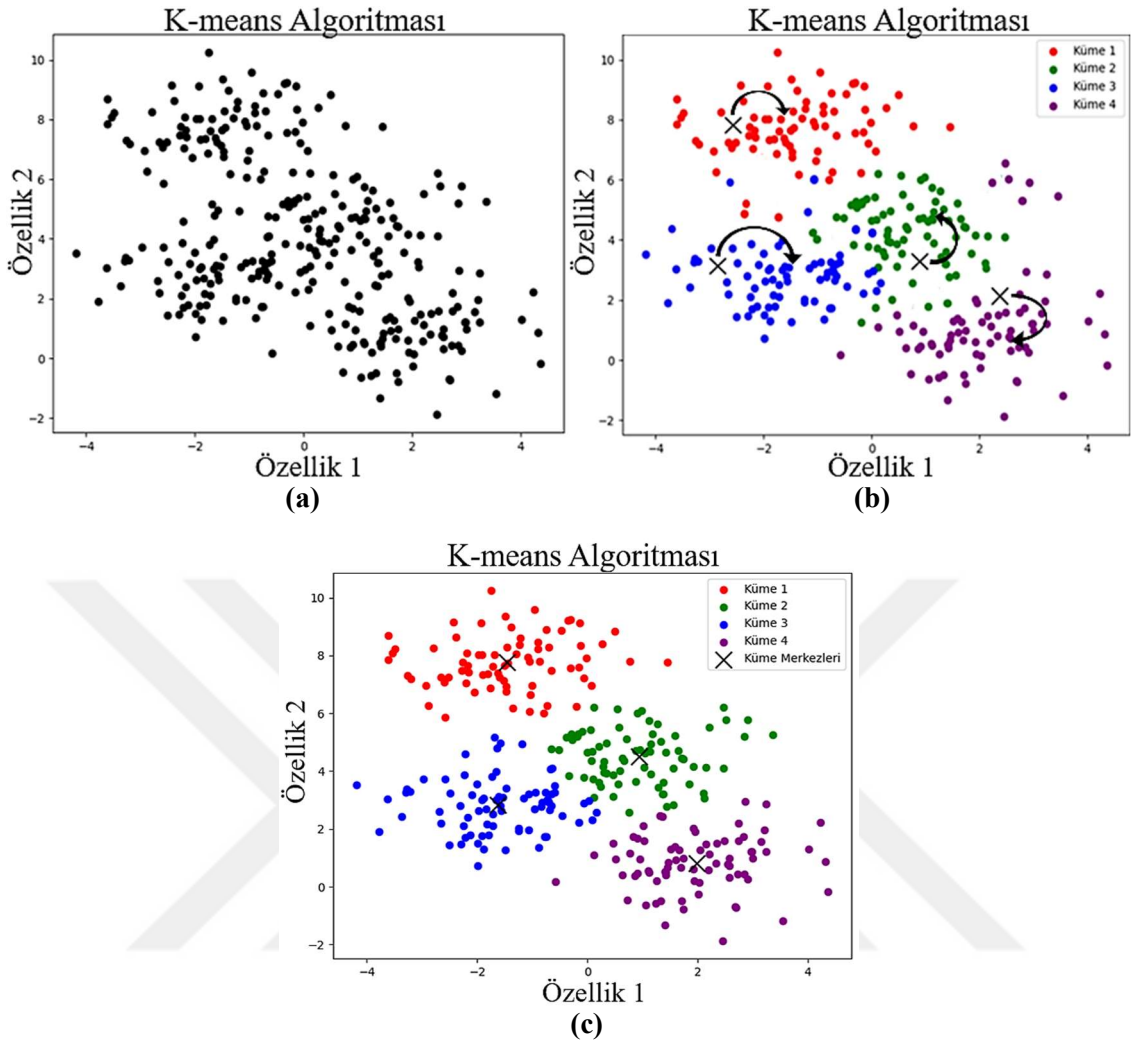
döngü sonlandırıldığında veri noktaları K sayısı kadar kümeye ayrılır ve küme merkezleri belirlenir. Şekil 3.4' te bu durum akış şeması şeklinde anlatılmıştır.



Şekil 3.4 K-means algoritması akış şeması.

K-means algoritmasının aşamaları Şekil 3.4'te akış şeması halinde sunulmuştur. Küme merkezlerinin sabitlenmesi ile birlikte K-means algoritması sonuçlanır.

Şekil 3.5'te K-means algoritması akış şeması koordinat düzlemindeki rastgele oluşturulan verilerin dağılımları ve K-means algoritmasının aşamaları görselleştirilmiştir.



Şekil 3.5 **(a)**İşlenmemiş veriler **(b)**K-means algoritmasında geçici küme merkezlerinin belirlenmesi **(c)**K-means algoritması ile kümeleme sonuçları.

Şekil 3.5 (a) 'da rastgele oluşturulan işlenmemiş veriler görülmektedir. Şekil 3.5 (b) 'de kullanıcı tarafından belirlenen 4 adet geçici küme merkezleri görülmektedir. K-means algoritmasının başlamasıyla verilerin mesafe ölçümlerine göre en yakın küme merkezine tahsis edilmiştir. Şekil 3.5 (c)' de küme merkezlerinin sabitlenmesine bağlı olarak optimize edildiği ve her veri noktasının kendi kümesine atandığı görülmektedir. K-means algoritmasının sonuçlandığı aşama görülmektedir.

K-means kümeleme algoritmasının temel adımlarını açıklayan Python dili ile sözde kodu ifade edilecek olursa;

1. Veri Setini ve Küme Sayısını Belirle:
   - Input: Veri seti (X), Küme sayısı (k)

2. Başlangıç Merkezlerini Rastgele Seç:
   - centroids = X[random.choice(m, k, replace=False)
3. İterasyon Başlat:
   - for iterasyon in range(max_iters):
     a. Her Veriyi En Yakın Küme ile Eşleştir:
       - labels = argmin(norm(X[:, np.newaxis] - centroids, axis=2), axis=1)
     b. Yeni Küme Merkezlerini Hesapla:
       - new_centroids = [X[labels == j].mean(axis=0) for j in range(k)]
     c. Küme Merkezi Değişikliğini Kontrol Et:
       - Eğer norm(new_centroids - centroids) < tol, döngüyü sonlandır
       - Değilse, centroids = new_centroids ve iterasyona devam et
4. Sonuçları Döndür:
   - return centroids, labels.

3.3 Kabuk Tipi Sentetik Veri Setleri

Sentetik veri setleri, teknolojinin hızla ilerlemesine bağlı olarak günümüzün veri odaklı dünyasında önemli bir fonksiyona sahip olan yapay verilerdir. Bu veri setleri, gerçek veri setlerindeki benzer özelliklerini ve dağılımlarını yansıtmak amacıyla üretilirler. Sentetik veriler, veri bilimi, yapay zekâ, veri analizi, makine öğrenmesi ve benzeri alanlarda kullanılarak çeşitli amaçları destekler. Geliştirilen algoritmaların eğitimi ve test aşaması için kullanılmaktadır. Veri analizi ve model eğitiminde pratiğe dayalı çalışmalar ve detaylı deneyler için de idealdir. Gerçek veri setlerinin çoğunlukla yüksek boyutlarda olması nedeniyle, erişimin sınırlı ve maliyetli olduğu durumlarda sentetik veriler önemli role sahiptir. Sentetik veri setleri, gerçek verilerin benzer özelliklerine, dağılımına ve yapısına sahiptir. Bu durum, makine öğrenimi ve yapay zekâ modellerinin gerçek verilerle uyumluluğunu artırır. Böylece daha yüksek doğrulukta sonuçlar elde edilebilir. Sentetik veriler, veri tabanlı uygulamaların sürdürülebilirliğini sağlamasına önemli bir alternatif sunmaktadır.

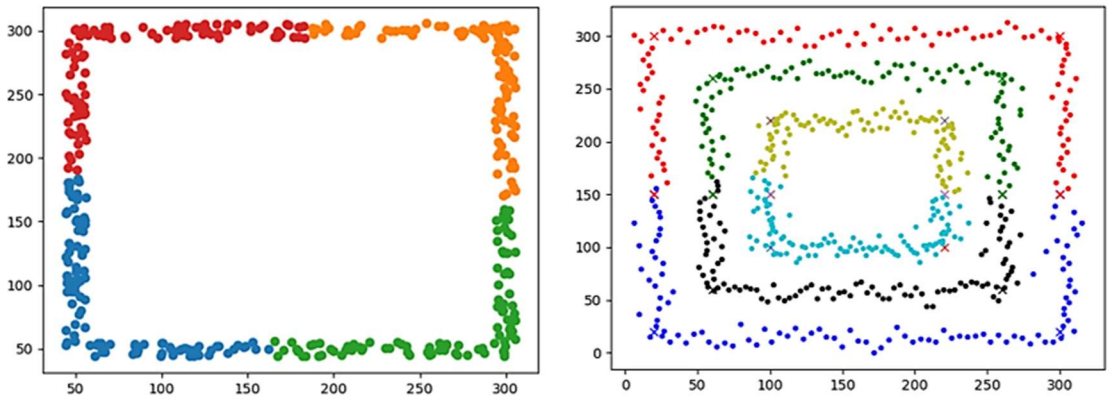
Bu veriler, görselleştirebilmek amacı ile iki boyutlu uzayda üretilebilir. İki boyutlu uzayda sentetik veri üretimi, koordinat düzlemi içinde veri noktalarının gösterildiği bir yöntemdir. Bu iki boyutlu uzay veri özelliklerinin temsil edildiği bir çerçeveyi içerir. Her

bir veri noktası, belirli bir özelliğin değerlerini ifade eder. Sentetik veri üretiminin iki boyutlu uzayda yapılması birçok konuda avantaj sağlar. İki boyutlu uzay, verilerin grafiksel olarak görselleştirilmesini sağlar ve görsel olarak da anlaşılabilirliğini artırmaktadır. Verilerdeki farklı özellikler arasındaki ilişkileri daha kolay ve hızlı bir şekilde gözlemlenmesini sağlar.

Gauss dağılımı temel alınarak oluşturulan çok boyutlu veri setlerinde elemanların çoğu merkezde değil, verinin etrafına doğru giderek uzaklaşan bir "kabuk" içinde yer alır (Domingos 2012). Bu nedenle, bu çalışmada veri yoğunluğunun kabuk kısmında olduğu iki boyutlu sentetik veri setleri üretilmiştir. Geliştirilen yöntem aslında çok boyutlu verilerde kullanılacaktır ancak verilerin görselleştirilebilmesi amacıyla bu çalışmada iki boyutlu KTSV'ler kullanılmıştır. Bu kapsamda iki farklı topolojide KTSV'ler oluşturulmuştur. Bunlar, Tek Katmanlı Topoloji (TKT) ve Çok Katmanlı Topoloji (ÇKT)'dir.

1. TKT, tek katmanlı dörtgen şeklinde olup küme sayısı parametrik olarak ayarlanabilmektedir,
2. ÇKT, birden fazla katman içeren iç içe dörtgenler şeklinde tasarlanmıştır.

TKT ve ÇKT yapılarından birer örnek KTSV Şekil 3.6' da gösterilmiştir.



Şekil 3.6 (a) TKT yapısında oluşturulan bir KTSV **(b)** ÇKT yapısında oluşturulan bir KTSV.

Şekil 3.6 (a)'da TKT yapısında oluşturulan bir KTSV, (b)'de ÇKT yapısında oluşturulan bir KTSV görülmektedir.

3.3.1 Kabuk Tipi Sentetik Veri Seti Parametreleri

Bu bölümde, önerilen çalışmada geliştirilen KTSV'nin genel yapısı açıklanmıştır. Kullanılan parametreler, oluşturulan KTSV'ler ile birlikte detaylı bir şekilde açıklanmıştır. Önerilen çalışma kapsamında, özgün kümeleme analizini belirleyen parametreler aşağıda listelenmiştir.

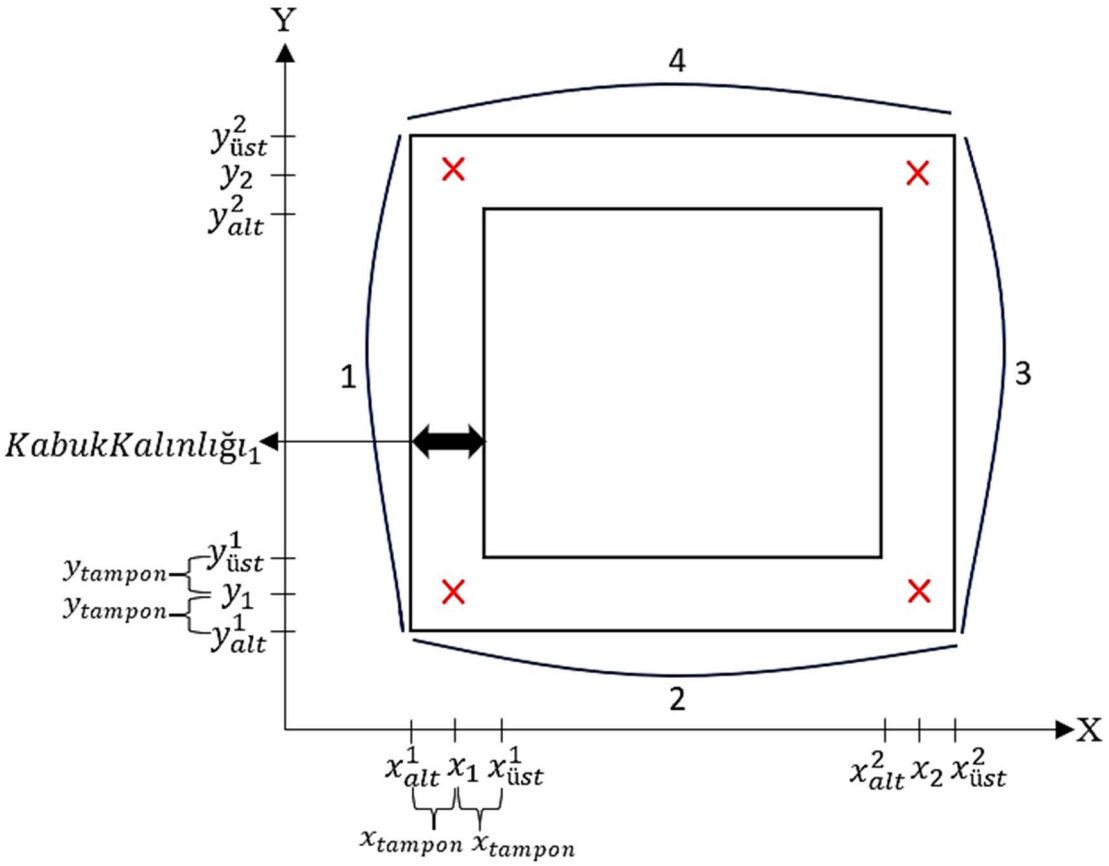
1. Kümeler arası boşluk
2. Küme sayısı
3. Küme içi boşluk
4. Kabuk kalınlığı
5. Toplam veri sayısı
6. İç içe küme (Katman) sayısı

3.3.1.1 KTSV Genel Yapısı

Küme sayısının değişimi ile analizi için oluşturulan kare KTSV'de ilk olarak köşe noktaları belirlenmiştir. Koordinat düzleminde rastgele belirlenen X ve Y değerlerinin kesişimi köşe noktaları olarak belirlenmiştir. Şekil 3.7' de bu köşe noktaları temsili olarak kırmızı renk görülmektedir. Her iki köşe nokta arası, kare topolojisindeki KTSV'nin bir kenarının uzunluğunu ifade eder. Sentetik veriler bu kare içerisinde rastgele olarak oluşturulmuştur. Koordinat düzlemindeki X ve Y noktalarının alt ve üst değerleri, oluşturulan kare kenarının kalınlığını belirler. Karenin 4 kenarının oluşturulma sırası Şekil 3.7' de belirtilmiştir.

Kare KTSV oluşturmak için geliştirilen bu yöntemde, her bir küme için belirli bir tampon bölgesi oluşturuluyor. Tampon, oluşturulan rastgele noktaların koordinatlarını belirlerken kullanılır. Örneğin, x_{tampon} ve y_{tampon} değerleri, X ve Y koordinatlarını oluştururken her bir noktanın belirli bir uzaklık içinde bir bölgede yer almasını sağlar. Burada, x_{alt}^{1} ve $x_{\text{üst}}^{2}$ arasında rastgele X koordinatları oluşturulurken, y_{alt}^{1} ve $y_{\text{üst}}^{2}$ arasında rastgele Y koordinatları oluşturuluyor. Ancak, her bir noktanın bu koordinatlardan x_{tampon} ve y_{tampon} değerleri kadar uzaklıkta olması sağlanır. Yani, her bir noktanın X ve Y koordinatlarına, x_{tampon} ve y_{tampon} değerleri eklenip çıkarılarak tampon bölgesi içinde

rastgele bir yerde oluşturulur. Bu tampon değerleri, noktalar arasında belirli bir uzaklık bırakmak ve daha düzenli bir dağılım elde etmek için kullanılır. Tamponlar, oluşturulan noktaların birbirine çok yakın olmasını önleyerek daha okunaklı ve düzenli bir veri seti elde etmeye yardımcı olmaktadır. Şekil 3.7'de oluşturulan tampon bölgeleri belirtilmiştir.



Şekil 3.7 KTSV'lerin temsili gösterimi.

Şekil 3.7'de görülen kare şekli, oluşturulan KTSV'lerin temsili olarak bir gösterimidir.

Kare şeklinde oluşturulan bu KTSV'nin koordinat düzlemindeki X ekseninde oluşturulan iç ve dış köşe noktaları;

$$x^1_{alt} = x_1 - x_{tampon} \quad (3.4)$$

$$x^1_{üst} = x_1 + x_{tampon} \quad (3.5)$$

$$x^2_{alt} = x_2 - x_{tampon} \quad (3.6)$$

$$x^2_{üst} = x_2 + x_{tampon} \quad (3.7)$$

Aynı şekilde Y eksenindeki iç ve dış köşe noktaları:

$$y_{alt}^{1} = y_1 - y_{tampon} \tag{3.8}$$

$$y_{\ddot{u}st}^{1} = y_1 + y_{tampon} \tag{3.9}$$

$$y_{alt}^{2} = y_2 - y_{tampon} \tag{3.10}$$

$$y_{\ddot{u}st}^{2} = y_2 - y_{tampon} \tag{3.11}$$

olarak hesaplanabilir.

Yukarıdaki denklemler ile birlikte kabuk kalınlığı elde edilir. Kabuk kalınlığı için;

$$KabukKalınlığı_1 = x_{\ddot{u}st}^{1} - x_{alt}^{1} \tag{3.12}$$

$$KabukKalınlığı_2 = y_{\ddot{u}st}^{1} - y_{alt}^{1} \tag{3.13}$$

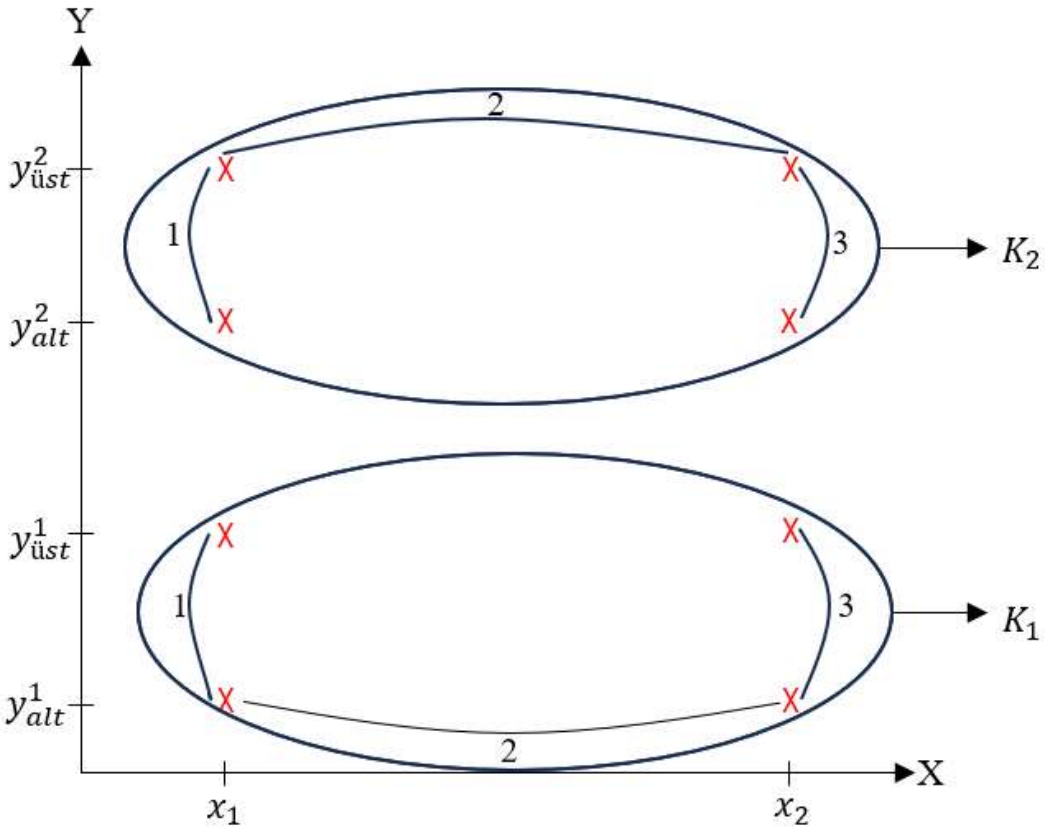
$$KabukKalınlığı_3 = x_{\ddot{u}st}^{2} - x_{alt}^{2} \tag{3.14}$$

$$KabukKalınlığı_4 = y_{\ddot{u}st}^{2} - y_{alt}^{2} \tag{3.15}$$

denklemleri uygulanır.

Yapılan çalışmada, Şekil 3.7'de belirtildiği gibi herhangi bir kümelere ayırma işlemi gerekmeksizin kullanılan parametrelerle birlikte kümeleme işlemi ve sonrası için de bir kısım parametreler kullanılmaktadır. Şekil 3.8'de bu parametreler görsel olarak açıklanmıştır.

Kümelenmiş KTSV'nin iki boyutlu uzayda oluşturulması için öncelikle koordinat düzleminde veri noktalarının oluşturmak istenilen şeklin köşe noktaları oluşturulur. Şekil 3.8'de bu köşe noktaları temsili olarak kırmızı renk görülmektedir. Her iki köşe nokta arası bir küme kolunun uzunluğunu ifade eder. Sentetik veriler bu köşe noktaları içerisinde rastgele olarak oluşturulmuştur. Kabuk şeklinin oluşması için verilerde belirli bir bant aralığında rastgele uzaklıklarda, koordinatların x_1 ve x_2 değerleri etrafında belirlenmiştir (Özden vd. 2022). Çalışma kapsamında oluşturulan dikdörtgen ve iç içe yapılandırılmış dikdörtgen KTSV'ler bahsedilen yöntemle oluşturulmuştur. Şekil 3.8 'de bu yöntemle oluşturulan iki kümeden oluşan dikdörtgen KTSV köşe noktaları belirtilmiştir.



Şekil 3.8 İki kümeden oluşan dikdörtgen KTSV köşe noktaları.

Şekil 3.8'de iki kümeden oluşan dikdörtgen KTSV köşe noktaları görülmektedir.

İki boyutlu uzayda oluşturulan dörtgen şeklinde KTSV'lerin geometrileri belirlenirken kullanılan önemli parametreleriden biri de köşe mesafe parametresidir. X ve Y eksenlerinde farklı mesafeler oluşabilir. Bu köşe mesafeleri Denklem 3.16-21'de ifade edilmiştir.

$$K_1Mesafe_1 = |y^1_{üst} - y^1_{alt}| \quad (3.16)$$

$$K_1Mesafe_2 = |x_2 - x_1| \quad (3.17)$$

$$K_1Mesafe_3 = |y^1_{üst} - y^1_{alt}| \quad (3.18)$$

Yukarıdaki denklemlerde;

$K_1Mesafe_1$: K_1 kümesinin birinci köşe mesafesi

$K_1 Mesafe_2$: K_1 kümesinin ikinci köşe mesafesi

$K_1 Mesafe_3$: K_1 kümesinin üçüncü köşe mesafesi

ifade eder. Aynı şekilde ikinci küme içinde uygulanmıştır.

$$K_2 Mesafe_1 = |y^2_{\text{üst}} - y^2_{alt}| \tag{3.19}$$

$$K_2 Mesafe_2 = |x_2 - x_1| \tag{3.20}$$

$$K_2 Mesafe_3 = |y^2_{\text{üst}} - y^2_{alt}| \tag{3.21}$$

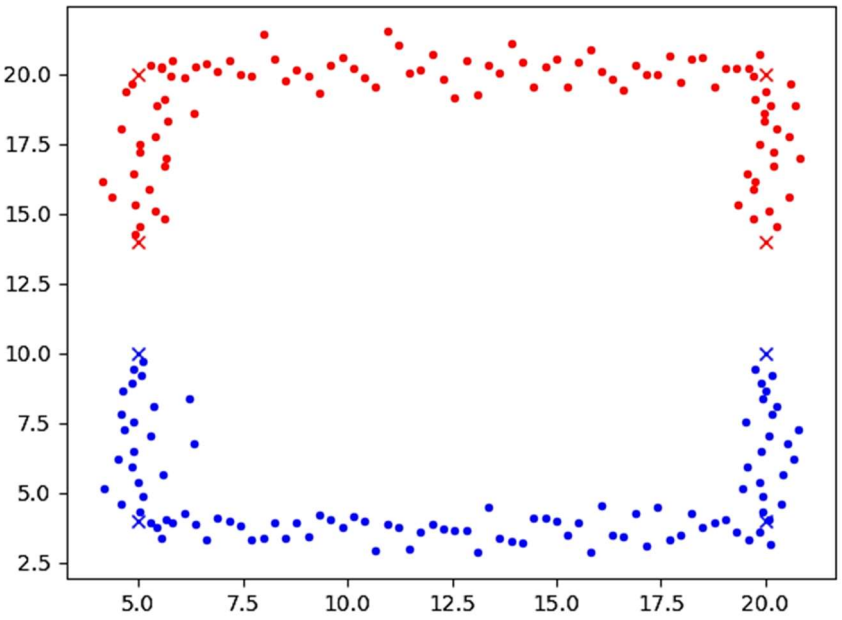
Yukarıdaki denklemlerde;

$K_2 Mesafe_1$: K_2 kümesinin birinci köşe mesafesi

$K_2 Mesafe_2$: K_2 kümesinin ikinci köşe mesafesi

$K_2 Mesafe_3$: K_2 kümesinin üçüncü köşe mesafesi

ifade eder. Şekil 3.9'da yukarıdaki denklemlerle oluşturulan kümelenmiş bir KTSV gösterilmiştir.



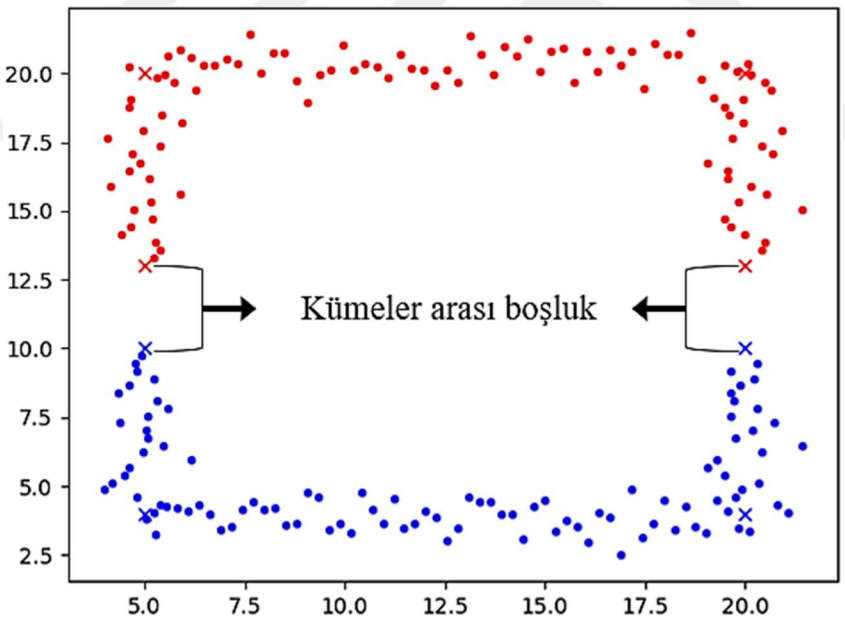
Şekil 3.9 İki kümeli dikdörtgen KTSV.

Şekil 3.9'da iki kümeli dikdörtgen KTSV gösterilmiştir.

3.3.1.2 Kümeler Arası Boşluk

Kümeler arası boşluk, kümeleme performansını değerlendirmek için kullanılan bir ölçüdür. Bu değer, küme içi benzerliği küme dışındaki benzerlikten ayırarak bir kümenin ne kadar homojen olduğunu ölçmeye çalışır. Kümeler arası boşluk parametresi, bir kümenin diğer kümelerden ne kadar uzak olduğunu ölçer. Bu parametre, kümeleme algoritmasının doğruluğunu ve veri setinin gerçek yapısına olan uyumunu değerlendirmeye yardımcı olabilir.

Görselleştirmeler, kümeleme sonuçlarını anlamak ve yorumlamak için önemli bir araçtır. Kümeleme algoritmalarının veri setindeki yapıları ortaya çıkarmadaki başarısını göstermek için çeşitli grafikler ve görsel analiz yöntemleri kullanılabilir. Bu, kümeleme sonuçlarının daha iyi anlaşılmasına ve algoritma parametrelerini ayarlamalarına olanak sağlar. Şekil 3.10' da kümeler arası boşluk parametresi ile veri dağılımı verilmiştir.



Şekil 3.10 Kümeler arası boşluk parametresi.

Kümeler arası boşluk, sadece bir ölçüt olarak değil, daha geniş bir perspektif içinde diğer parametreler ve görselleştirmelerle birleştirilerek değerlendirildiğinde, kümeleme algoritmalarının başarısını daha verimli bir şekilde değerlendirilmesini sağlar. Bu

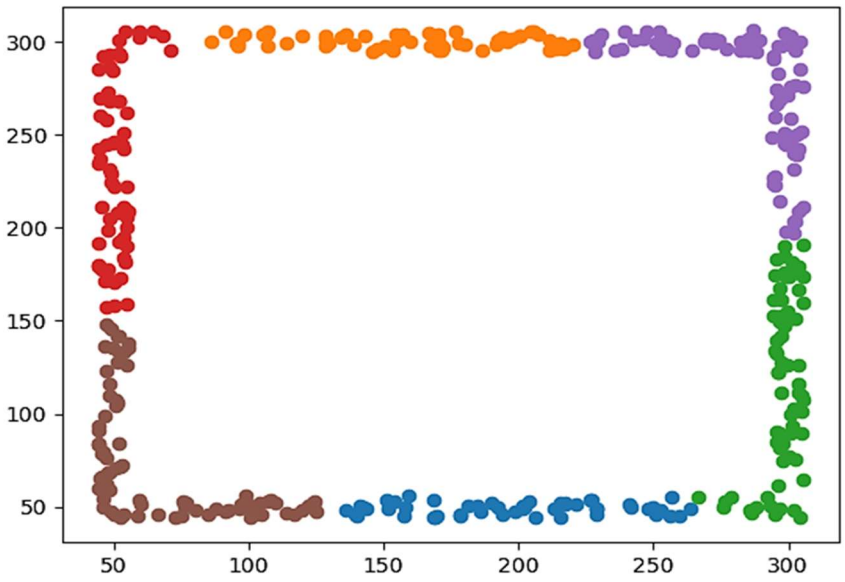
yaklaşım, veri setinin karmaşıklığına ve yapısına daha iyi uyum sağlamak için daha kapsamlı bir değerlendirme sunabilir.

Daha yüksek bir kümeler arası boşluk, genellikle daha etkili bir kümeleme performansını yansıtabilir. Ancak, bu tek parametreyi kullanmak yerine, kümeleme algoritmalarının gerçek veri setine uygunluğunu daha kapsamlı bir şekilde değerlendirmek için diğer parametreler ile birlikte ele almak önemlidir.

3.3.1.3 Küme Sayısı

Küme sayısı, kümeleme analizinde oldukça kritik bir parametredir ve bu parametrenin seçimi, analizin başarısı ve etkinliği üzerinde belirleyici bir etkiye sahiptir. Küme sayısının değişimi, analiz sonuçlarını önemli ölçüde etkileyebilir ve doğru küme sayısını belirlemek, anlamlı ve yararlı sonuçlar elde etmek için önemlidir. Küme sayısı, veri setindeki doğru yapıyı anlamak ve ortaya çıkarmak için önemlidir.

Küme sayısının artırılması, analizin daha hassas ve detaylı olmasını sağlayabilir. Daha fazla küme, daha küçük ve özelleştirilmiş gruplamalara olanak tanır, bu da veri setindeki ince yapıları açığa çıkarabilir. Şekil 3.11'de 6 adet küme sayısına sahip KTSV görülmektedir.



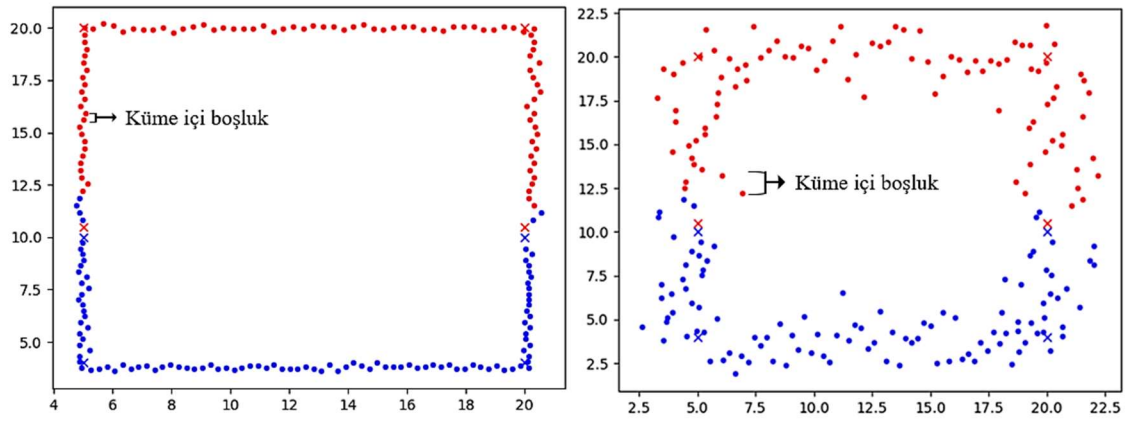
Şekil 3.11 6 kümeli bir KTSV.

Küme sayısının değişimi ile kümeleme analizi, geniş bir perspektif sunabilir ve bu durum, analizin anlamlılığını ve kapsamını artırabilir. Bu yöntem, çeşitli küme sayıları üzerinde yapılan analizler aracılığıyla veri setinin farklı yapılarını anlama ve değerlendirme imkânı sağlar. Farklı küme sayılarına sahip kümeleme analizleri, veri setindeki potansiyel yapıları keşfetme olanağı tanır. Veri setindeki farklı gruplamaların anlaşılmasını ve ayırt edilmesini sağlar. Farklı küme sayıları için yapılan analizler, optimal küme sayısının belirlenmesine yardımcı olabilir. Elde edilen performans metrikleri ve görsel analiz, hangi küme sayısının veri setine en iyi uyan yapıyı sağladığını belirlemede önemli bir role sahip olabilir.

3.3.1.4 Küme İçi Boşluk

Küme içi boşluk parametresi, kümeleme analizinde kümelerin içindeki veri noktalarının birbirleriyle olan mesafesini değerlendiren önemli bir ölçüdür. Bu parametre, her bir kümenin içindeki verilerin birbirine ne kadar yakın olduğunu ölçerek kümenin homojenliğini değerlendirir. Küme içindeki verilerin mesafesinin değişimi, bir küme içindeki veriler arasındaki benzerliği yansıtır. Artan homojenlik, veri noktalarının birbirine daha yakın olduğu ve kümelerin daha tutarlı olduğu anlamına gelir. Kapsamlı değerlendirilen kümeleme analizlerinde küme içindeki verilerin mesafesi, kümeleme performansını artırabilir. Homojen kümeler, analizin daha sağlam ve anlamlı sonuçlar üretmesini sağlar. Küme içindeki verilerin mesafesi, farklı küme sayıları için değerlendirildiğinde, optimal küme sayısının belirlenmesine yardımcı olabilir. Mesafe değişimi, belirli bir küme sayısının veri setine en iyi şekilde uyup uymadığını değerlendirmeye yardımcı olabilir. Mesafe değişiklikleri, potansiyel anormal veri noktalarını tespit etme ve eleme konusunda bilgi sağlar. Bu durum, veri setindeki aykırı değerleri tanımlama imkânı sunar. Küme içi mesafe değişikliği, kümeleme sonuçlarının görsel inceleme sürecini zenginleştirebilir. Veri setindeki küme yapılarının ve örüntülerin daha iyi anlaşılmasına yardımcı olabilir.

Şekil 3.12 (a)'da küme içindeki verilerin birbirine göre mesafesinin az olduğu (b)'de küme içindeki verilerin birbirine göre mesafesinin çok olduğu küme içi boşluk parametresi görülmektedir.



Şekil 3.12 (a) Küme içi boşluk parametresinin az olduğu veri dağılımı **(b)** Küme içi boşluk parametresinin fazla olduğu veri dağılımı.

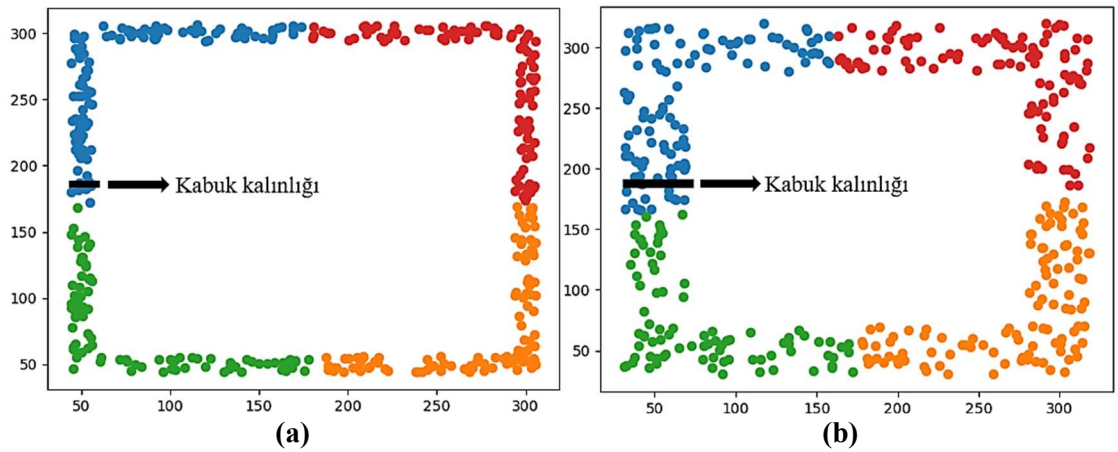
3.3.1.5 Kabuk Kalınlığı

Kabuk kalınlığı, kabukların genişliği veya yoğunluğu gibi özellikleri temsil eden bir parametredir. Kabuk kalınlığının uygun bir şekilde seçilmesi, veri noktalarının gerçek dünya özelliklerine daha iyi uyum sağlayan daha anlamlı kümelemeler elde etmeye yardımcı olabilir. İyi seçilmiş bir kabuk kalınlığı parametresi, kümeleme sonuçlarını optimize edebilir ve benzer özelliklere sahip veri noktalarının daha doğru bir şekilde gruplandığından emin olabilir. Bu, analizin anlamlılığını artırabilir ve veri setindeki desenleri daha etkili bir şekilde ortaya çıkarabilir.

Bu çalışmada, veri seti geometrisine bağlı olarak kabuk kalınlığı parametrik olarak değiştirilebilmektedir. Bulgular bölümünde bu parametrenin kümeleme sonuçlarına etkisi ayrıntılı olarak incelenmiştir.

Kabuk kalınlığının değiştirilmesi, kümeleme analizinde bir denge kurmaya çalışmak anlamına gelir. Uygun bir kabuk kalınlığı seçimi, analizin hem detaylı hem de genel bir perspektife sahip olmasını sağlar. Bu şekilde, veri setindeki önemli desenler ve ilişkiler daha iyi anlaşılabilir.

Şekil 3.13 (a)'da kabuk kalınlığının az olduğu veri dağılımı (b)'de kabuk kalınlığının çok olduğu veri dağılımı görülmektedir.



Şekil 3.13 (a) Kabuk kalınlığının az olduğu KTSV **(b)** Kabuk kalınlığının çok olduğu KTSV.

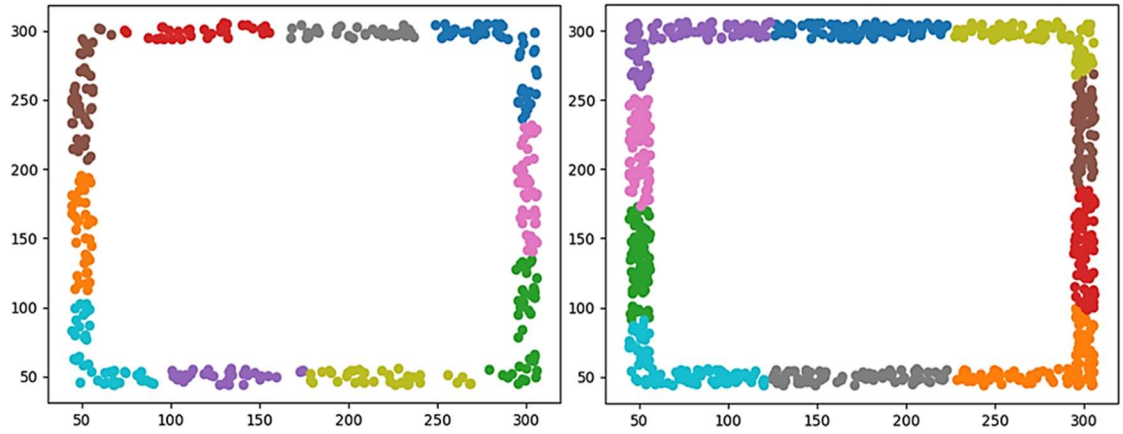
3.3.1.6 Toplam Veri Sayısı

Kabuk tipinde oluşturulan sentetik veri setinde toplam veri sayısı parametresini değiştirmek, kümeleme analizi üzerinde önemli bir etkiye sahiptir. Toplam veri sayısı, kümeleme algoritmalarının veri setindeki desenleri ve ilişkileri tanımlamak için kullanabileceği örnek sayısını belirler. Bu parametrenin değiştirilmesi, analizin güvenilirliği, doğruluğu ve genellikle analizin genel performansı üzerinde etkili olabilir.

Daha fazla veri eklemek, genellikle daha genel ve temsilci kümelemelere yol açabilir. Büyük veri setleri, algoritmaların veri setindeki desenleri daha iyi anlamasına ve daha genel kümeler oluşturmasına olanak tanır. Ancak, aşırı miktarda veri eklemek, işlem sürelerini artırabilir ve bilgi aşırı yüklenmesine neden olabilir. Daha az veri kullanmak, daha spesifik ve detaylı kümelemelere yol açabilir. Ancak, bu durumda, analizin genel doğruluğu ve güvenilirliği azalabilir.

Toplam veri sayısının değiştirilmesi, kümeleme analizi için bir denge kurma sürecidir. Uygun bir veri seti boyutu seçimi, analizin hem yeterli örnekleri içermesini hem de işleme uygun olmasını sağlar. Bu durum, hem analizin doğruluğunu artırabilir hem de hesaplama kaynaklarını daha etkili kullanabilir.

Şekil 3.14 (a)'da toplam veri sayısının az olduğu 10 kümeli KTSV (b)'de toplam veri sayısının çok olduğu 10 kümeli KTSV görülmektedir.



Şekil 3.14 (a) Toplam veri sayısının az olduğu KTSV **(b)** Toplam veri sayısının çok olduğu KTSV.

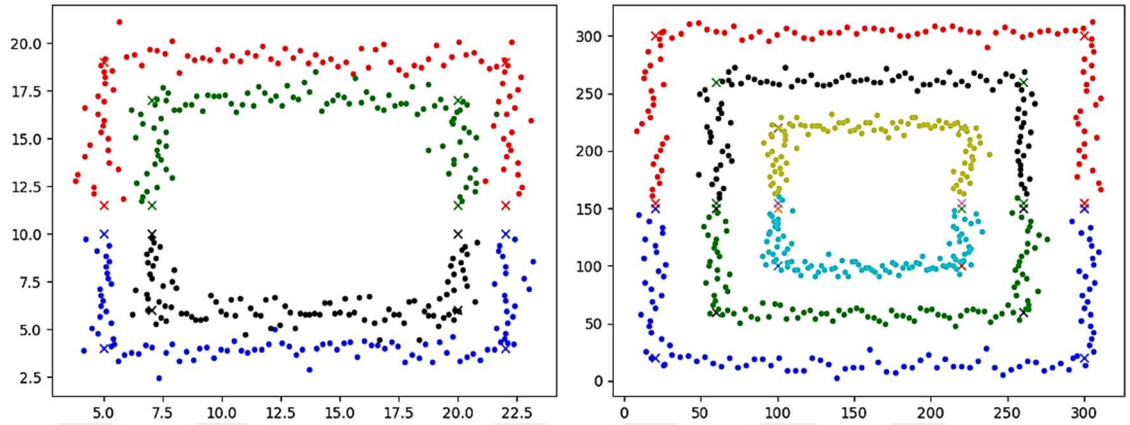
3.3.1.7 İç İçe Küme (Katman) Sayısı

Kabuk tipinde oluşturulan sentetik veri setine iç içe küme eklemek ve bu küme sayısını değiştirmek, kümeleme analizi üzerinde önemli bir etkiye neden olabilir. İç içe kümeleme, veri setinde daha fazla detay ve hiyerarşik yapıyı tanımlama olanağı sağlar. Bu durum, kümeleme analizini daha kapsamlı ve ayrıntılı hale getirir.

Küme sayısını değiştirmek, iç içe kümeleme analizindeki esnekliği artırabilir. Daha fazla küme sayısı, veri setindeki farklı alt grupları daha iyi anlamamıza olanak tanır. Ancak, çok sayıda küme kullanmak, analizin karmaşıklığını artırabilir ve anlamsız kümelemelere yol açabilir. Daha az küme sayısı seçmek ise genel desenleri daha vurgulu hale getirebilir, ancak bu durumda detaylardan feragat edilmiş olabilir. Küme sayısını belirlerken, veri setinin doğası, analizin amacı ve sonuçlardan elde edilmek istenen bilgiler dikkate alınmalıdır.

İç içe kümeleme ve küme sayısını değiştirmek, kümeleme analizinin hassasiyetini artırabilir. Ancak, bu değişiklikleri dikkatlice yapmak, aynı zamanda analizin doğruluğunu ve anlamlılığını sürdürmek için önemlidir. Ayrıca, bu tür değişikliklerin sonuçlarını anlamak ve yorumlamak için analizin amacını ve bağlamını anlamak önemlidir.

Şekil 3.15 (a)' da iç içe küme sayısının 4 olduğu KTSV (b)'de iç içe küme sayısının 6 olduğu KTSV görülmektedir.



(a) **(b)**

Şekil 3.15 (a) İç içe küme sayısının 4 olduğu 2 katmanlı KTSV **(b)** İç içe küme sayısının 6 olduğu 3 katmanlı KTSV.

3.4 Kümeleme Performans Ölçütleri

Kümeleme performans ölçümü olarak doğruluk, kesinlik, duyarlılık ve F1 skoru yöntemi sıklıkla kullanılmaktadır (Aslanyürek vd. 2021).

Doğruluk

Doğruluk, doğru tahmin edilen gözlemlerin toplam gözlem sayısına oranını ifade eder. Matematiksel olarak şu şekilde hesaplanır:

$$Doğruluk = \frac{TP+TN}{TP+TN+FP+F} \tag{3.22}$$

Kesinlik

Kesinlik, pozitif olarak tahmin edilen gözlemlerin gerçekten pozitif olma olasılığını ifade eder. Matematiksel olarak şu şekilde hesaplanır:

$$Kesinlik = \frac{TP}{TP+FP} \tag{3.23}$$

Duyarlılık

Duyarlılık, gerçek pozitif gözlemlerin ne kadarının doğru bir şekilde tespit edildiğini ifade eder. Matematiksel olarak şu şekilde hesaplanır:

$$Duyarlılık = \frac{TP}{TP+FN} \tag{3.24}$$

F1 Skoru

F1 skoru, kesinlik ve duyarlılığın harmonik ortalamasını ifade eder. Bu skor, bir modelin hem kesinlikle hem de duyarlılıkla ne kadar iyi performans gösterdiğini değerlendirmek için kullanılır. Matematiksel olarak şu şekilde hesaplanır:

$$F1\ Skoru = 2 \times \frac{Kesinlik \times Duyarlılık}{Kesinlik+Duyarlılık} \tag{3.25}$$

Formüllerde kullanılan değişkenler;
TP: Gerçek pozitif (doğru pozitif) sayısı
TN: Gerçek negatif (doğru negatif) sayısı
FP: Yanlış pozitif sayısı
FN: Yanlış negatif sayısı

Her biri farklı yönleri temsil eder: doğruluk, genel doğru tahminleri; kesinlik, pozitif tahminlerin doğruluğunu; duyarlılık, gerçek pozitiflerin kapsayıcılığını; F1 skoru ise kesinlik ve duyarlılığın birleşimini gösterir.

Önerilen çalışmada TKT yapısında oluşturulmuş KTSV'de, parametrelerin değişimine bağlı olarak kümeleme performans ölçütleri hesaplanmıştır.

3.5 Çarpıklık Ölçütü

Çarpıklık, bir veri setinin dağılımının simetrik olup olmadığını ifade eden bir ölçüdür. Pozitif çarpıklık, dağılımın sağa doğru çekik olduğu anlamına gelir. Yani, veri seti sağa

doğru uzanır ve sağ kuyruk daha uzundur. Negatif çarpıklık ise dağılımın sola doğru çekik olduğunu gösterir. Bu durumda, veri seti sola doğru uzanır ve sol kuyruk daha uzundur. Çarpıklık genellikle üçüncü merkezi moment kullanılarak hesaplanır. Bu, veri noktalarının ortalamadan olan sapmalarının küpünü ifade eder.

Çarpıklık katsayısı;

$$\gamma_1 = \frac{\mu_3}{\sigma^3} \tag{3.26}$$

Formülde kullanılan değişkenler;
μ_3: Üçüncü ortalama etrafındakı moment
σ: Standart Sapma

Bu denklemin katsayı sonucu ile;

- γ_1= 0 ise veri dağılımı simetrik,
- γ_1> 0 ise veri dağılımı sağa çarpık,
- γ_1< 0 ise veri dağılımı sola çarpık,

sonucuna ulaşılmaktadır.

Önerilen çalışmada kabuk tipi sentetik verilerin dağılımı rastgele uzaklıklarda olduğundan dolayı çarpıklık ölçütü ile kümeleme ilişkisi hesaplanmıştır. Çarpıklık ölçütü [X Y] koordinatları şeklinde hesaplanmıştır.

3.6 Basıklık Ölçütü

Basıklık, bir veri setinin dağılımının tepesinin ne kadar yüksek veya düşük olduğunu ifade eden bir ölçüdür. Normal dağılıma göre, pozitif basıklık daha yüksek ve dar bir zirveye, negatif basıklık ise daha düşük ve geniş bir zirveyi ifade eder. Basıklık genellikle dördüncü merkezi moment kullanılarak hesaplanır.

Basıklık katsayısı;

$$\gamma_2 = \frac{\mu_4}{\sigma^4} - 3 \quad (3.27)$$

Formülde kullanılan değişkenler;

μ_4: Dördüncü ortalama etrafındakı moment

σ: Standart Sapma

Bu denklemin katsayı sonucu ile;

- γ_2= 0 ise veri dağılımı normaldir,
- γ_2> 0 ise veri dağılımı sivri(dik)dir,
- γ_2< 0 ise veri dağılımı basıktır,

sonucuna ulaşılmaktadır.

Önerilen çalışmada kabuk tipi sentetik verilerin dağılımı rastgele uzaklıklarda oluşturulduğu için basıklık ölçütü ile kümeleme ilişkisi hesaplanmıştır. Basıklık ölçütü [X Y] koordinatları şeklinde hesaplanmıştır.

3.7 Kabuk Tipi Veri Setlerinin Görüntü Uygulamaları

Kabuk tipi sentetik veriler, verilerin dış kısımlarına doğru iki boyutlu uzayda oluşturulması ile elde edilir. Çok boyutlu verilerin, hacminin büyük bir kısmı veri setinin (Gauss dağılımına sahip veriler) dış kabuğunda bulunur (Domingos 2012). Bu bağlamda görüntü işleme uygulamalarında da çeşitli ön işlemler uygulanarak görüntü verilerinde bulunan nesnelerin kenar kısımları kabuk tipi yapısında oluşturulmuştur. Görüntü verilerinin ön işleme aşamasında, kabuk tipi verileri belirlemek amacı ile kenar algılama yöntemi olan Canny filtresi kullanılmıştır. Görüntü verisindeki nesnelerin kabuk tipi yapısında oluşturulması ile birlikte K-means algoritması ile kümeleme yapılmıştır.

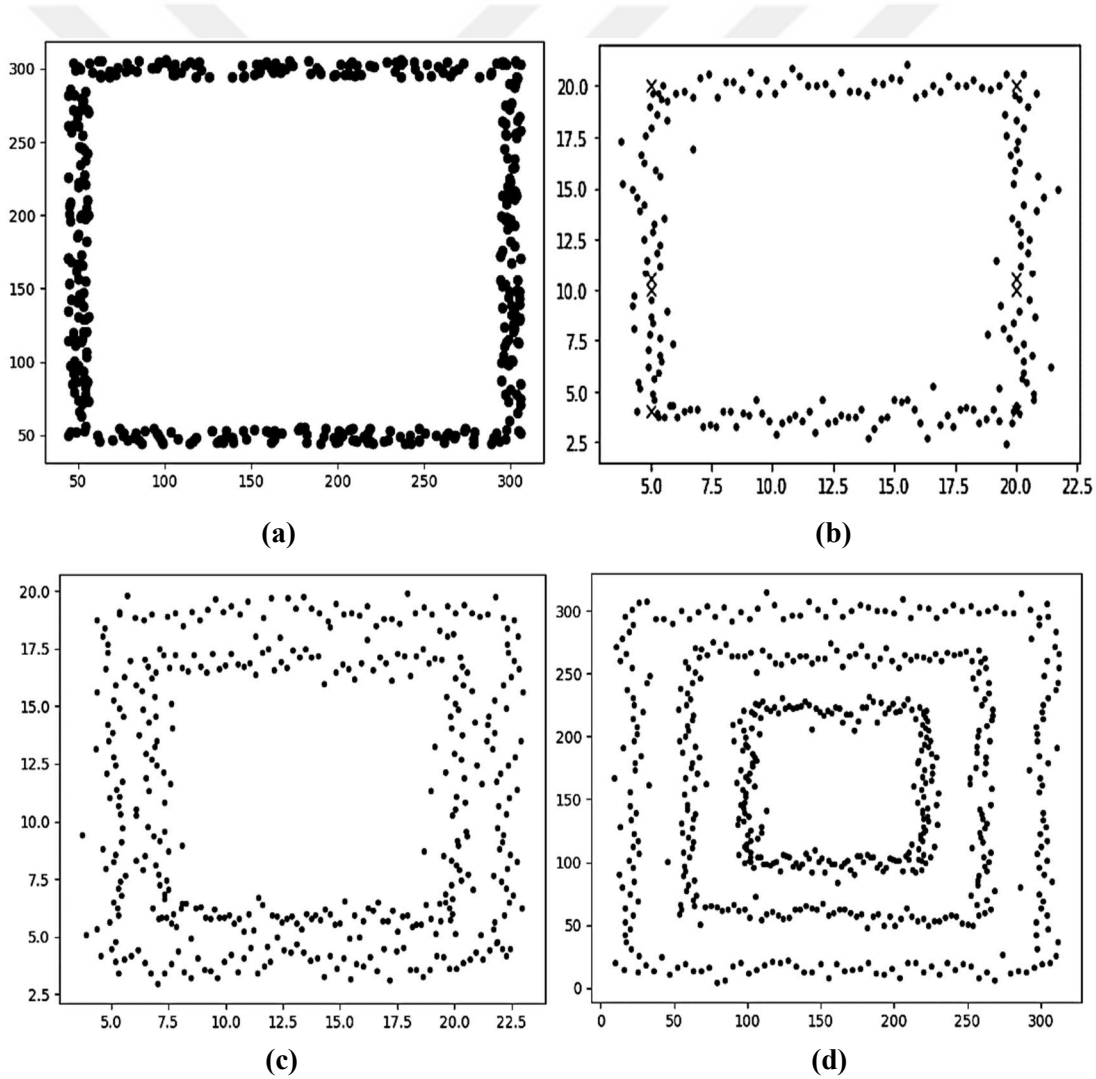
Önerilen bu yöntem ile görüntü işleme uygulamalarında kişi yeniden tanımlama, tıbbi görüntüleme ile hastalık teşhisi, nesne tanıma, güvenlik sistemleri ile plaka tanıma gibi çeşitli alanlarda katkı sağlayacağı düşünülmektedir.

4. BULGULAR

Bu bölümde, oluşturulan TKT ve ÇKT yapıdaki KTSV'ler üzerinde gerçekleştirilen deneysel kümeleme çalışmalarının özgün analizleri çizelgeler ve şekiller halinde sunulmuştur.

4.1 Deneysel Çalışmalar

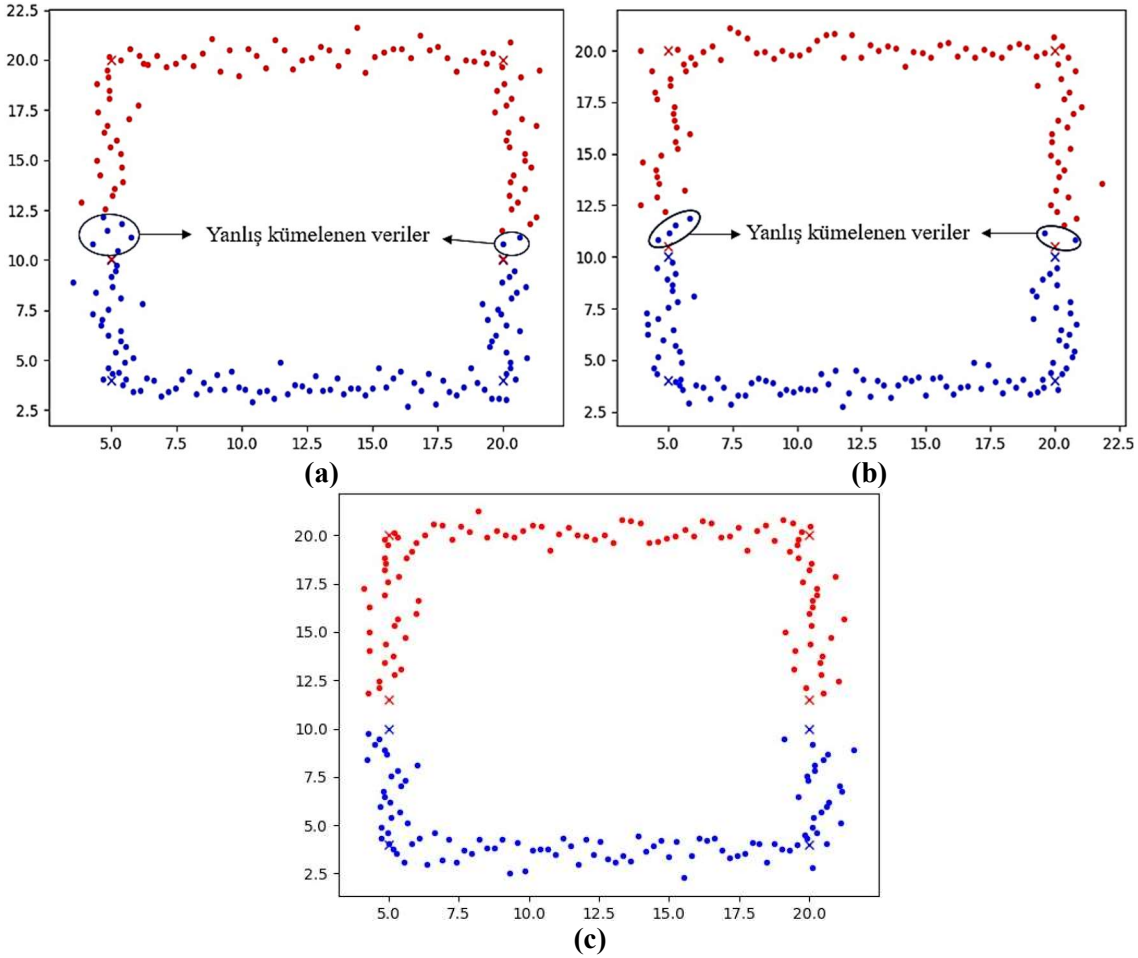
Önerilen çalışmada TKT ve ÇKT yapısında oluşturulan 4 farklı KTSV'nin işlenmemiş veri dağılımı Şekil 4.1 verilmiştir.



Şekil 4.1 (a) TKT yapısında oluşturulan KTSV'nin işlenmemiş veri dağılımı **(b)** TKT yapısında oluşturulan iki kümeli KTSV'nin işlenmemiş veri dağılımı **(c)** ÇKT yapısında oluşturulan 2 katmanlı KTSV'nin işlenmemiş veri dağılımı **(d)** ÇKT yapısında oluşturulan 3 katmanlı KTSV'nin işlenmemiş veri dağılımı.

4.1.1 Kümeler Arası Boşluk Parametresinin Değişimi

a) TKT; Bu deneysel çalışmada TKT yapısında oluşturulan iki kümeli KTSV 200 adet veri içermektedir. Her kümede 100 adet veri bulunmaktadır. Bu deneyler sırasında küme içi boşluk parametresi 0.5 birim olarak sabit tutulmuştur. Kümeler arası boşluk parametresi değiştirilerek 20 adet deney yapılmış ve hatalı kümelenen veri noktalarının kümülatif toplamları gözlenmiştir. Yapılan deneylerde kullanılan KTSV'lerden bir örnek dağılım Şekil 4.2'de verilmiştir.



Şekil 4.2 (a) Kümeler arası boşluk parametresinin 0.1 birim olduğu veri dağılımı **(b)** Kümeler arası boşluk parametresinin 0.5 birim olduğu veri dağılımı **(c)** Kümeler arası boşluk parametresinin 1.5 birim olduğu veri dağılımı.

Şekil 4.2 (a)'da kümeler arası boşluk parametresinin 0.1 birim olduğu veri dağılımı, (b)'de kümeler arası boşluk parametresinin 0.5 birim olduğu veri dağılımı, (c)'de kümeler arası boşluk parametresinin 1.5 birim olduğu veri dağılımı görülmektedir. Şekilde görülen

"x" işaretleri küme sınırları olup bu işaretin dışında kalan veri noktaları hatalı kümelelenmiş veriler olarak nitelendirilmiştir.

Çizelge 4.1'de 2 küme içeren TKT yapısında oluşturulan KTSV'nin kümeler arası boşluk parametresinin değişimi ile deneylerin sonuçları görülmektedir.

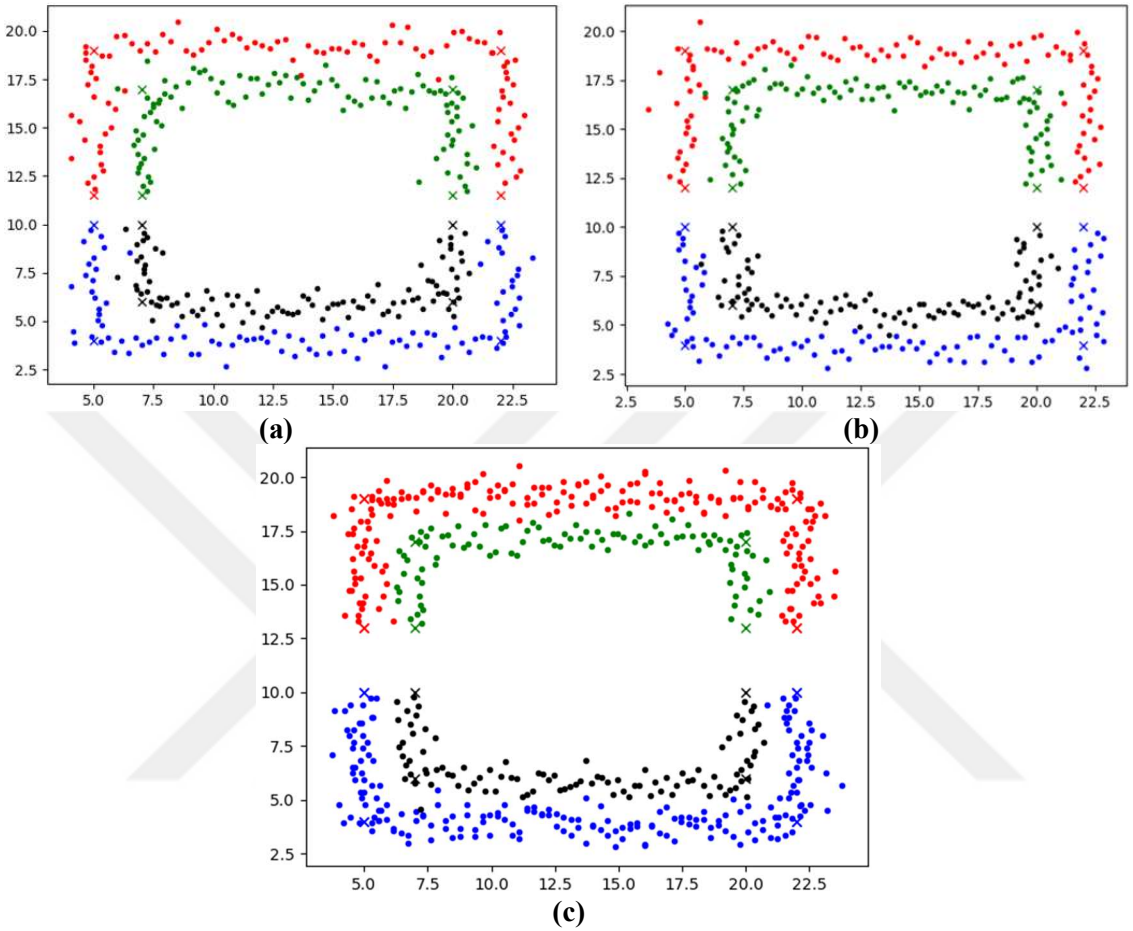
Çizelge 4.1 İki kümeli TKT yapısında oluşturulan KTSV'nin kümeler arası boşluk ile deney sonuçları.

Kümeler Arası Boşluk	Hata Oranı (%)	İşlem Süresi (sn)	Çarpıklık		Basıklık	
0.1	4	0.031	0.017	0.182	-1.633	-1.623
0.2	3.5	0.031	0.015	0.178	-1.627	-1.626
0.3	3.5	0.031	0.011	0.170	-1.629	-1.627
0.4	3	0.031	0.017	0.167	-1.635	-1.632
0.5	3	0.031	0.018	0.149	-1.626	-1.643
0.6	2.5	0.031	0.017	0.155	-1.624	-1.639
0.7	2.5	0.031	0.009	0.166	-1.631	-1.641
0.8	2	0.030	0.006	0.141	-1.628	-1.656
0.9	1.5	0.020	0.024	0.139	-1.620	-1.657
1.0	1	0.020	0.002	0.132	-1.621	-1.666
1.1	0.5	0.015	0.022	0.116	-1.623	-1.679
1.2	0.5	0.015	0.013	0.118	-1.614	-1.681
1.3	0.5	0.019	0.006	0.121	-1.618	-1.681
1.4	0.5	0.022	0.005	0.102	-1.618	-1.693
1.5	0	0.015	0.003	0.112	-1.608	-1.688
1.6	0	0.015	0.023	0.098	-1.606	-1.700
1.7	0	0.015	0.018	0.086	-1.606	-1.704
1.8	0	0.015	0.023	0.092	-1.614	-1.709
1.9	0	0.015	0.012	0.090	-1.613	-1.711
2.0	0	0.018	0.001	0.079	-1.604	-1.713

Çizelge 4.1'de kümeler arası boşluk parametresinin 1.5 birim veya daha büyük olduğunda hata gözlenmemiştir. Çizelge 4.1'de içeren deneysel sonuçlara göre, kümeler arası boşluk parametre biriminin artması ile daha kısa sürede kümeleme yaptığı ve çarpıklık ölçütünün azalması ile birlikte veri dağılımının daha simetrik olduğu gözlenmiştir. Basıklık ölçütü ile kümeler arası boşluk parametresinin değişimi ile arasında bir ilişki gözlenmemiştir. Basıklık ölçütü sonuçları 0'dan küçük olduğu için, bu deneysel çalışmada kullanılan TKT yapısındaki KTSV'nin veri dağılımının basık olduğu gözlenmiştir.

b) ÇKT; Bu deneyde kullanılan ÇKT yapısındaki 2 katmanlı KTSV 4 küme içermektedir. Bu veri setinde her küme 100 adet veri içermek üzere toplam 400 adet veri bulunmaktadır. Küme içi boşluk parametresi 0.5 birim olarak sabit tutulmuştur. Kümeler arası boşluk

parametresi değiştirilerek gerçekleştirilen 20 adet deney neticesinde oluşan dağılımlardan bir kısım örnek Şekil 4.3'de verilmiştir.



Şekil 4.3 (a) Kümeler arası boşluk parametresinin 1.5 birim olduğu veri dağılımı **(b)** Kümeler arası boşluk parametresinin 2 birim olduğu veri dağılımı **(c)** Kümeler arası boşluk parametresinin 3 birim olduğu veri dağılımı.

Şekil 4.3 (a)'da kümeler arası boşluk parametresinin 1.5 birim olduğu veri dağılımı, (b)'de kümeler arası boşluk parametresinin 2 birim olduğu veri dağılımı, (c)'de kümeler arası boşluk parametresinin 3 birim olduğu veri dağılımı görülmektedir.

Bu deney ile dört kümeli ÇKT yapısında oluşturulmuş KTSV'de, kümeler arası boşluk parametresinin değişimi ile hata oranı, işlem süresi, çarpıklık ve basıklık değerleri elde edilmiştir. Elde edilen sonuçlara göre K-means algoritmasının kümeleme performans analizi yapılmıştır. Bu deneylerde elde edilen dört kümeli ÇKT yapısında oluşturulan KTSV'nin kümeler arası boşluk ile deney sonuçları Çizelge 4.2'de verilmiştir.

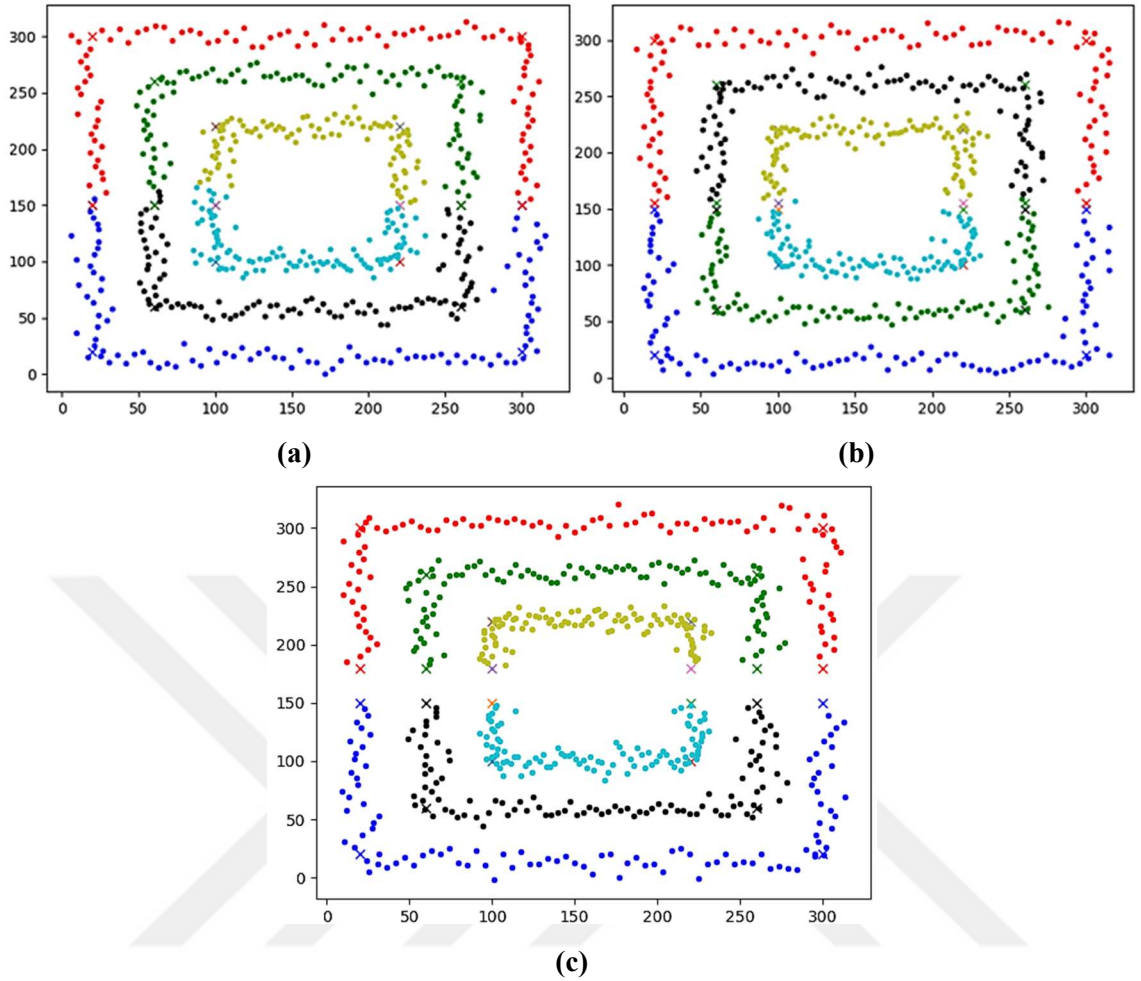
Çizelge 4.2 Dört kümeli ÇKT yapısında oluşturulan KTSV'nin kümeler arası boşluk ile deney sonuçları.

Kümeler Arası Boşluk	Hata Oranı (%)	İşlem Süresi (sn)	Çarpıklık		Basıklık	
0.1	4	0.046	0.020	0.176	-1.572	-1.639
0.2	3	0.031	0.001	0.171	-1.575	-1.651
0.3	3	0.046	0.027	0.157	-1.572	-1.666
0.4	3	0.046	0.014	0.152	-1.569	-1.667
0.5	3	0.031	0.026	0.146	-1.562	-1.670
0.6	2.5	0.050	0.018	0.141	-1.562	-1.681
0.7	2.5	0.031	0.004	0.131	-1.566	-1.687
0.8	2.5	0.046	0.017	0.125	-1.556	-1.699
0.9	2.5	0.032	0.006	0.114	-1.554	-1.707
1.0	2.5	0.039	0.032	0.100	-1.537	-1.720
1.1	2	0.046	0.010	0.082	-1.558	-1.722
1.2	2	0.046	0.008	0.094	-1.548	-1.718
1.3	2	0.046	0.024	0.077	-1.544	-1.729
1.4	2	0.046	0.018	0.092	-1.551	-1.748
1.5	2	0.038	0.010	0.075	-1.542	-1.741
1.6	1	0.032	0.013	0.070	-1.544	-1.749
1.7	0	0.031	0.030	0.059	-1.548	-1.760
1.8	0	0.031	0.025	0.056	-1.540	-1.759
1.9	0	0.031	0.003	0.050	-1.545	-1.759
2.0	0	0.046	0.160	0.041	-1.535	-1.768

Çizelge incelendiğinde kümeler arası boşluk 1.7 birim veya daha büyük olduğunda hata gözlenmemiştir. Kümeler arası boşluk parametresinin artması ile hata oranının azaldığı gözlenmiştir. En yüksek hata oranı, kümeler arası boşluk parametresinin en az olmasıyla elde edilmiştir. Bu deneysel çalışmada işlem süresi, çarpıklık ve basıklık değerleri ile bir ilişki gözlenmemiştir. Basıklık değerleri 0'dan düşük olduğu için veri dağılımının basık olduğu sonucuna ulaşılmıştır. İşlem süresi [0.031 – 0.050] sn değerleri arasında çıkmıştır. İşlem süresinin düşük değerlerde olması ile birlikte K-means algoritmasının kümeleme performansının kısa sürede gerçekleştiği sonucuna ulaşılmıştır.

ÇKT yapısında oluşturulan 6 kümeli 3 katman içeren KTSV'de 600 adet veriden oluşturulmuştur. Koordinat düzlemindeki veri dağılımı 20 ile 300 bant aralığındadır. Kümeler arası boşluk parametreleri değiştirilerek gerçekleştirilen 20 adet deney neticesinde hata oranı, işlem süresi, çarpıklık, basıklık değerleri elde edilmiştir.

Şekil 4.4'te altı kümeli 3 katman içeren ÇKT yapısında oluşturulan KTSV'de kümelenmiş verilerin koordinat düzlemindeki veri dağılımı görülmektedir.



Şekil 4.4 (a) Kümeler arası boşluk parametresinin 0.1 birim olduğu veri dağılımı **(b)** Kümeler arası boşluk parametresinin 5 birim olduğu veri dağılımı **(c)** Kümeler arası boşluk parametresinin 30 birim olduğu veri dağılımı.

Şekil 4.4 (a)'da kümeler arası boşluk parametresinin 0.1 birim olduğu, (b)'de kümeler arası boşluk parametresinin 5 birim olduğu, (c)'de kümeler arası boşluk parametresinin 30 birim olduğu veri dağılımı görülmektedir.

ÇKT yapısında oluşturulan 6 kümeli 3 katman içeren KTSV'de, kümeler arası boşluk parametresinin değişimi ile deney sonuçları Çizelge 4.3'te verilmiştir.

Çizelge 4.3 İç içe küme sayısının 6 olduğu ÇKT yapısında oluşturulan KTSV'nin kümeler arası boşluk parametresi ile deney sonuçları.

Kümeler Arası Boşluk	Hata Oranı (%)	İşlem Süresi (sn)	Çarpıklık		Basıklık	
0.1	3	0.093	0.002	0.123	-1.640	-1.654

Çizelge 4.3 (Devam) İç içe küme sayısının 6 olduğu ÇKT yapısında oluşturulan KTSV'nin kümeler arası boşluk parametresi ile deney sonuçları.

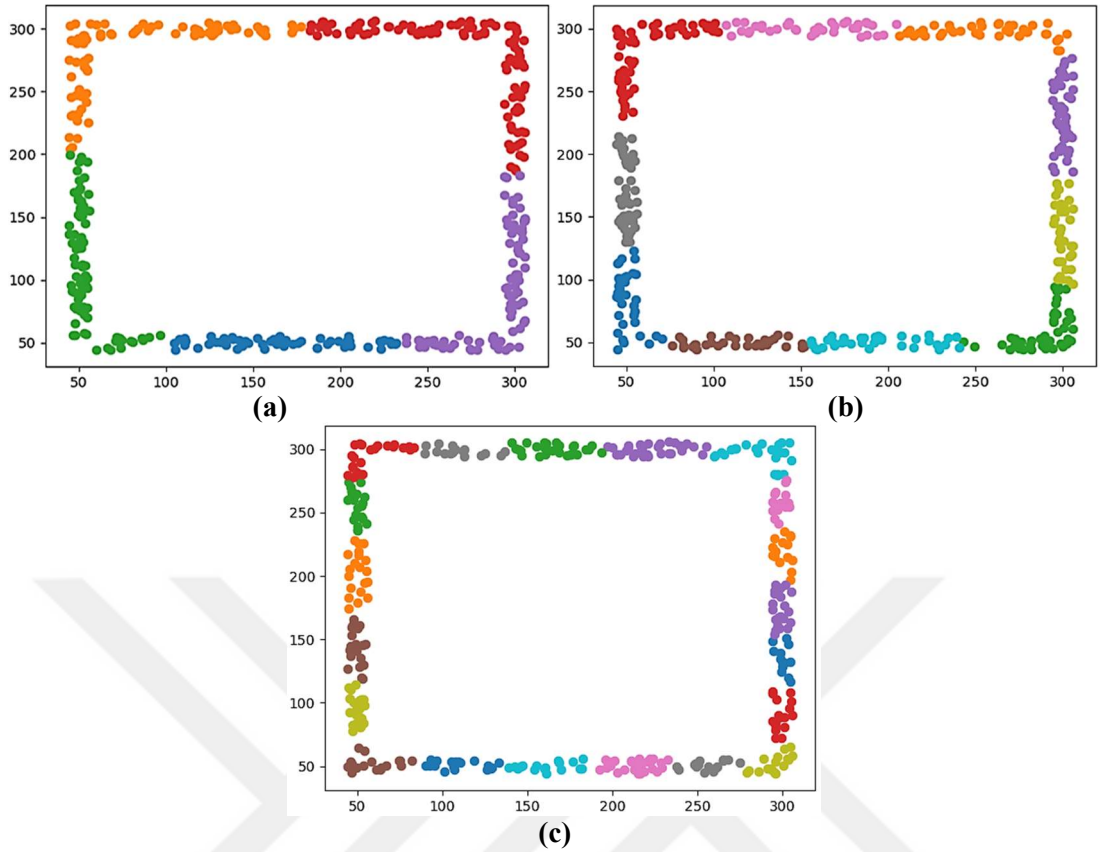
Kümeler Arası Boşluk	Hata Oranı (%)	İşlem Süresi (sn)	Çarpıklık		Basıklık	
0.2	2.5	0.093	0.002	0.123	-1.640	-1.654
0.3	2.5	0.109	0.002	0.123	-1.640	-1.655
0.4	2.5	0.125	0.002	0.122	-1.639	-1.654
0.5	2.5	0.109	0.005	0.122	-1.639	-1.656
0.6	2.5	0.109	0.001	0.122	-1.640	-1.655
0.7	2	0.125	0.001	0.119	-1.640	-1.657
0.8	2	0.109	0.011	0.120	-1.640	-1.657
0.9	2	0.109	0.002	0.123	-1.640	-1.657
1	2	0.113	0.005	0.125	-1.640	-1.657
2	1.5	0.093	0.006	0.166	-1.640	-1.661
3	1.5	0.124	0.009	0.101	-1.635	-1.677
4	1	0.130	0.009	0.099	-1.636	-1.680
5	0.5	0.139	0.020	0.083	-1.629	-1.695
6	0.5	0.109	0.002	0.083	-1.630	-1.690
7	0	0.109	0.001	0.079	-1.631	-1.694
8	0	0.127	0.009	0.066	-1.626	-1.708
9	0	0.138	0.008	0.064	-1.626	-1.712
10	0	0.124	-0.008	0.062	-1.625	-1.707

Çizelge incelendiğinde kümeler arası boşluk 7 birim veya daha büyük olduğunda hata gözlenmemiştir.

Kümeler arası boşluk parametresinin değişimi ile, 200 adet veri içeren TKT'ye göre kümeler arasındaki boşluğun az olduğu durumlarda da hata sayısının az olmasıyla birlikte istenilen sonuca ulaşılmıştır. Kümeler arası boşluk parametresinin 400 adet veri içeren 2 katmanlı ve 600 adet veri içeren 3 katmanlı ÇKT'de kümeler arası boşluğun az olması durumunda da hata sayısının az olduğu görülmektedir. Bu durum önerilen yöntemin, K-means algoritması ile kümeleme analizinde yüksek performans elde edildiği gözlenmiştir.

4.1.2 Küme Sayısı Parametresinin Değişimi

Bu deneysel çalışmada TKT yapısında oluşturulan KTSV'de 100 adet veri içermektedir. Küme sayısı parametresinin değişimi ile 28 adet deney yapılmıştır. Şekil 4.5 (a)'da küme sayısının 5 adet olduğu, (b)'de küme sayısının 10 adet olduğu, (c)'de küme sayısının 20 adet olduğu veri dağılımı görülmektedir.



Şekil 4.5 (a) Küme sayısının 5 adet olduğu veri dağılımı **(b)** Küme sayısının 10 adet olduğu veri dağılımı **(c)** Küme sayısının 20 adet olduğu veri dağılımı.

TKT'deki 100 adet veri içeren KTSV, küme sayısının değişimi ile K-means algoritmasının kümeleme analizi yapılmıştır. TKT yapısında oluşturulan KTSV'de K-means algoritmasının küme sayısına göre deneysel sonuçlar Çizelge 4.4'de verilmiştir.

Çizelge 4.4 TKT yapısında oluşturulan KTSV'nin küme sayısı değişimi ile deneysel sonuçları.

Küme Sayısı	İşlem Süresi(sn)	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
3	0.024	0.045	0.136	0.18	0.155
4	0.031	0.03	0.339	0.33	0.334
5	0.031	0.14	0.055	0.042	0.048
6	0.031	0.207	0.275	0.232	0.252
7	0.041	0.285	0.25	0.097	0.140
8	0.047	0.292	0.366	0.2	0.258
9	0.053	0.355	0.127	0.055	0.076
10	0.047	0.46	0.480	0.195	0.276
11	0.046	0.492	0.25	0.11	0.152
12	0.050	0.58	0.398	0.157	0.224
13	0.046	0.59	0.25	0.072	0.112
14	0.046	0.65	0.25	0.075	0.115

Çizelge 4.4 (Devam)TKT yapısında oluşturulan KTSV'nin küme sayısı değişimi ile deneysel sonuçları.

Küme Sayısı	İşlem Süresi(sn)	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
15	0.0624	0.692	0.25	0.072	0.112
16	0.062	0.73	0.140	0.045	0.068
17	0.065	0.785	0.25	0.055	0.090
18	0.062	0.802	0.062	0.015	0.024
19	0.078	0.865	0.25	0.04	0.068
20	0.078	0.86	0.75	0.117	0.202
21	0.063	0.897	0.25	0.03	0.053
22	0.080	0.92	0.25	0.047	0.079
23	0.087	0.937	0.25	0.045	0.076
24	0.093	0.955	0.25	0.035	0.061
25	0.082	0.967	0.25	0.035	0.061
26	0.078	0.975	0.5	0.077	0.134
27	0.111	0.98	0.5	0.062	0.110
28	0.116	0.987	0.25	0.045	0.07
29	0.094	0.99	0.5	0.08	0.137
30	0.118	1.0	0.25	0.035	0.061

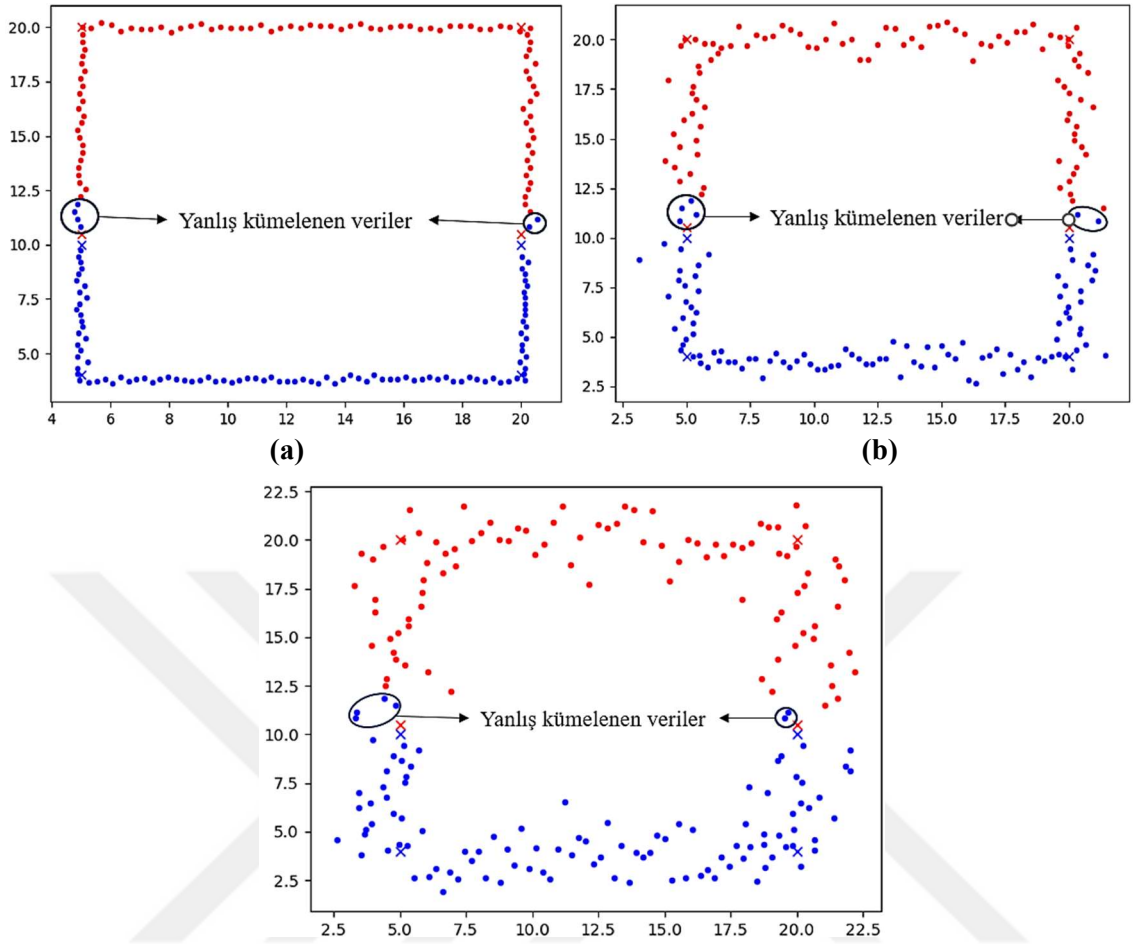
Çizelge 4.4'de küme sayısının artmasına bağlı olarak K-means kümeleme performansının doğruluk oranı ve işlem süresinin arttığı gözlenmiştir.

4.1.3 Küme İçi Boşluk Parametresinin Değişimi

a) TKT; Bu deneysel çalışmada TKT yapısında oluşturulan iki kümeli KTSV ile küme içi boşluk parametresi değiştirilerek 14 adet deney yapılmıştır. Her kümede 100 adet veri olmak üzere 200 adet veri içermektedir. Küme içi boşluk parametresi değişimi analizi yapılırken kümeler arası boşluk 1.5 birim olarak sabit değer belirlenmiştir. Deneyler sonucunda hata oranı, işlem süresi, çarpıklık, basıklık değerleri elde edilmiştir.

Şekil 4.6 (a)'da küme içi boşluk parametresinin 0.1 birim olduğu, (b)'de parametrenin 0.5 birim olduğu, (c)'de parametrenin 1 birim olduğu veri dağılımı görülmektedir.

Şekilde görülen "x" işaretleri küme sınırları olup bu işaretin dışında kalan veri noktaları hatalı kümelenen veriler olarak nitelendirilmiştir.Hatalı kümelenen veriler, hata oranı başlığı altında Çizelge 4.5'te sunulmuştur.



Şekil 4.6 (a) Küme içi boşluk parametresinin 0.1 birim olduğu veri dağılımı **(b)** Küme içi boşluk parametresinin 0.5 birim olduğu veri dağılımı **(c)** Küme içi boşluk parametresinin 1 birim olduğu veri dağılımı.

Şekil 4.6 'da iki kümeli TKT'de kümelenmiş verilerin küme içi boşluk parametresine göre koordinat düzlemindeki veri dağılımı görülmektedir.

Çizelge 4.5 'de 2 küme içeren TKT yapısında oluşturulan KTSV'nin küme içi boşluk parametresinin değişimi ile deneysel sonuçları görülmektedir.

Çizelge 4.5 İki kümeli TKT yapısında oluşturulan KTSV'nin küme içi boşluk parametresinin değişimi ile sonuçları.

Küme İçi Boşluk	**Hata Oranı (%)**	**İşlem Süresi (sn)**	**Çarpıklık**		**Basıklık**	
0.1	1.5	0.007	0.008	0.109	-1.635	-1.703
0.2	1.5	0.011	0.009	0.106	-1.632	-1.701

Çizelge 4.5 (Devam) İki kümeli TKT yapısında oluşturulan KTSV'nin küme içi boşluk parametresinin değişimi ile sonuçları.

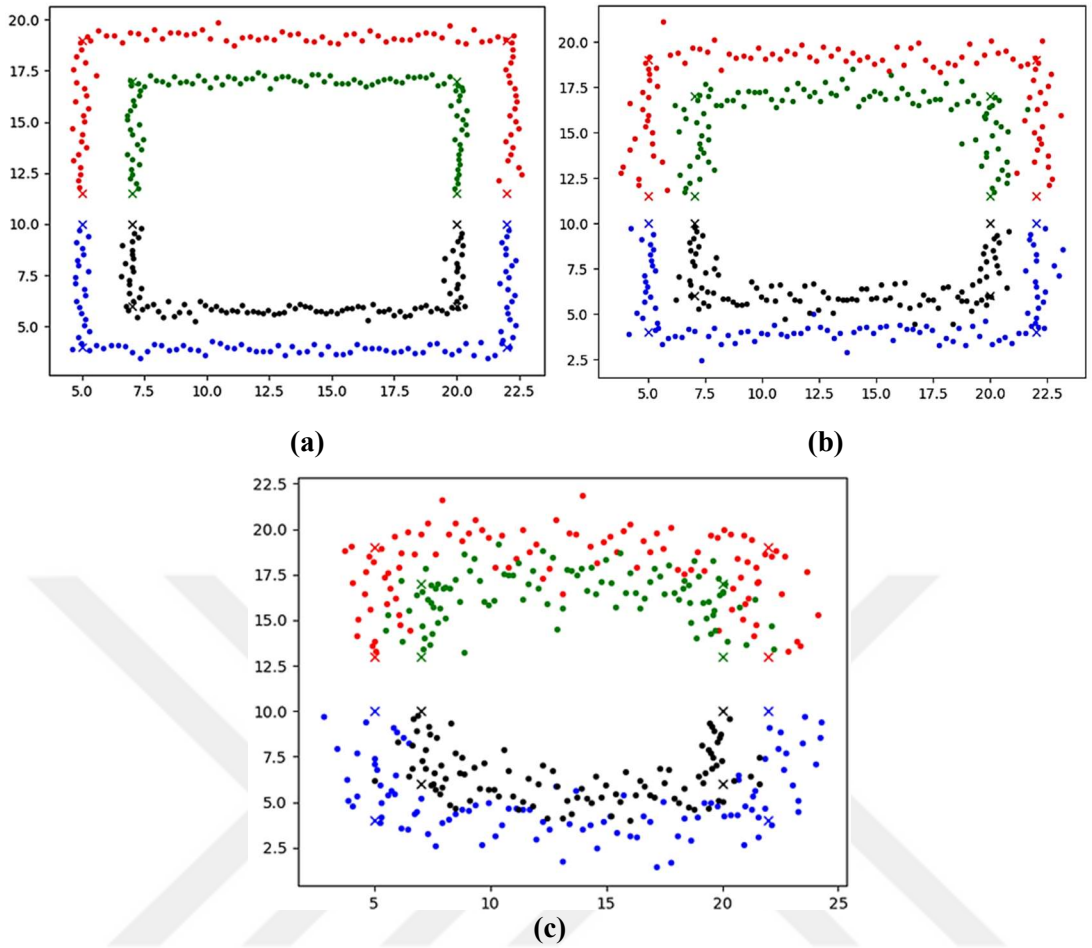
Küme İçi Boşluk	Hata Oranı (%)	İşlem Süresi (sn)	Çarpıklık		Basıklık	
0.3	2	0.015	0.011	0.102	-1.623	-1.697
0.4	2	0.015	0.002	0.105	-1.623	-1.693
0.5	2.5	0.015	0.022	0.108	-1.614	-1.689
0.6	2.5	0.015	0.001	0.120	-1.603	-1.668
0.7	2.5	0.015	0.019	0.095	-1.591	-1.674
0.8	3	0.015	0.016	0.103	-1.587	-1.666
0.9	3	0.015	0.016	0.116	-1.560	-1.653
1	3	0.031	0.032	0.109	-1.566	-1.642
2	3	0.024	0.080	0.145	-1.314	-1.443
3	3	0.015	0.006	0.086	-1.050	-1.171
4	3	0.015	0.035	0.177	-0.659	-1.015
5	6	0.031	0.138	0.090	-0.155	-0.605

Çizelge 4.5'e göre küme içi boşluk parametresinin çarpıklık ve basıklık değerleri ile K-means algoritması arasında bir ilişki gözlenmemiştir. Küme içi boşluk parametre değerinin artmasına bağlı olarak hata oranının da arttığı gözlenmiştir.

b) ÇKT; Bu deneysel çalışmada iki katmanlı ÇKT yapısında oluşturulmuş KTSV 400 adet veri içermektedir. Her küme 100 adet veriden oluşmaktadır. Küme içi boşluk parametresinin değişimi ile 16 adet deney yapılmıştır. Deneyler sonucunda hata oranı, işlem süresi, çarpıklık, basıklık değerleri elde edilmiştir. Küme içi boşluk parametresi değişimi analizi yapılırken kümeler arası boşluk 1.5 birim olarak sabit tutulmuştur.

Veri noktaları belirlenen aralıklarda rastgele dağılıma sahip olması nedeniyle, bazı veri dağılımları daha yoğunken, diğerlerinde daha seyrektir. Dolayısıyla, küme içi aralıklarının ayarlanması, veri kümesinin yapısal özelliklerine ve dağılımına daha uygun getirilir. K-means algoritması, veri noktalarını merkeze olan mesafesine göre kümeleme yaptığından dolayı küme içi aralıkların artması veri yapılarının düzenini bozmaktadır. Bu kapsamda küme içi boşluk parametresinin artması kümeleme performansını düşürmektedir.

Şekil 4.7 (a)'da küme içi boşluk parametresinin 0.2 birim olduğu (b)'de parametrenin 0.5 birim olduğu, (c)'de parametrenin 1 birim olduğu veri dağılımı görülmektedir.



Şekil 4.7 (a) Küme içi boşluk parametresinin 0.2 birim olduğu veri dağılımı **(b)** Küme içi boşluk parametresinin 0.5 birim olduğu veri dağılımı **(c)** Küme içi boşluk parametresinin 1 birim olduğu veri dağılımı.

Çizelge 4.6 'da 4 küme içeren ÇKT yapısında oluşturulan KTSV'nin küme içi boşluk parametresinin değişimi ile hata sayısı görülmektedir.

Çizelge 4.6 Dört kümeli ÇKT yapısında oluşturulan KTSV küme içi boşluk parametresinin değişimi ile sonuçları.

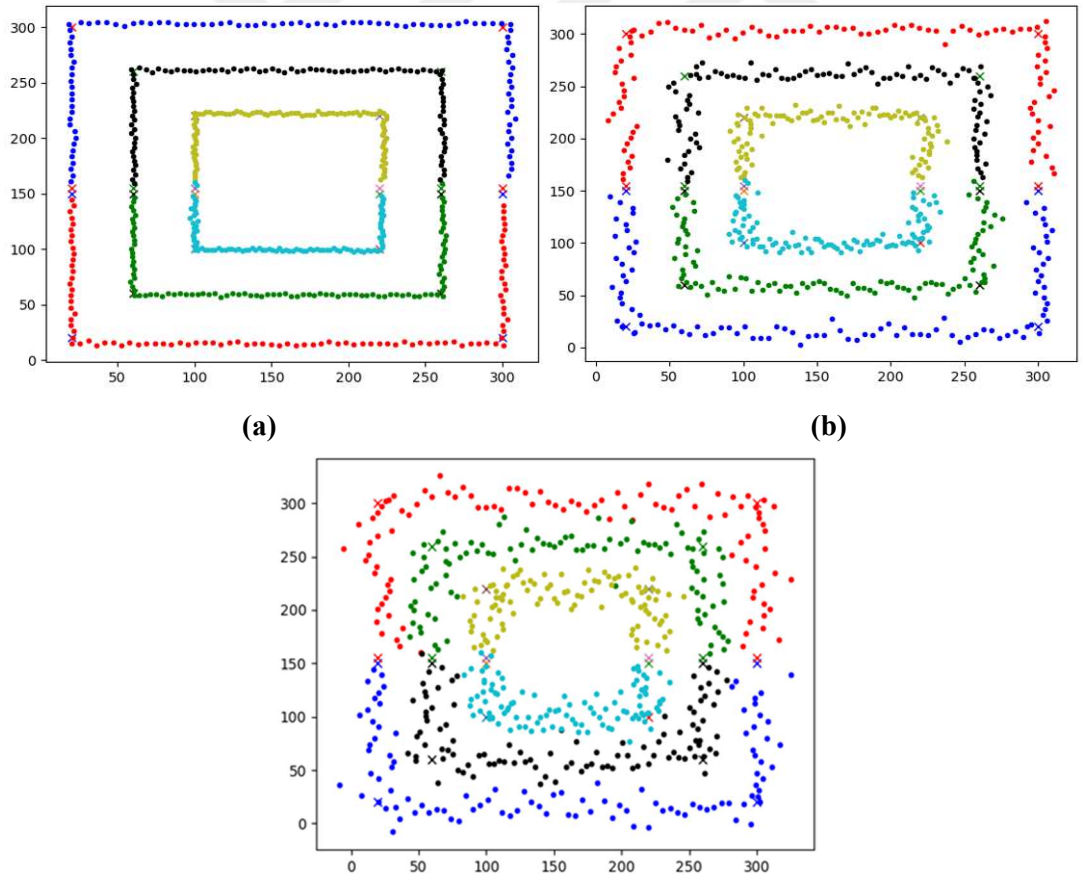
Küme İçi Boşluk	Hata Oranı (%)	İşlem Süresi (sn)	Çarpıklık	Basıklık
0.5	0	0.046	0.010 0.079	-1.541 -1.736
0.6	0.5	0.046	0.022 0.073	-1.525 -1.733
0.7	0.5	0.046	0.007 0.073	-1.518 -1.723
0.8	0.5	0.046	0.023 0.059	-1.483 -1.693
0.9	0.5	0.046	0.015 0.066	-1.477 -1.677
1.0	1	0.062	0.028 0.086	-1.443 -1.606
1.1	1	0.051	0.001 0.026	-1.454 -1.631
1.2	1	0.048	0.028 0.078	-1.395 -1.587
1.3	1	0.046	0.061 0.035	-1.342 -1.526

Çizelge 4.6 (Devam) Dört kümeli ÇKT yapısında oluşturulan KTSV küme içi boşluk parametresinin değişimi ile sonuçları.

Küme İçi Boşluk	Hata Oranı (%)	İşlem Süresi (sn)	Çarpıklık	Basıklık
1.4	1.5	0.046	0.030 0.078	-1.371 -1.551
1.5	2.5	0.044	0.033 0.043	-1.394 -1.505
1.6	2.5	0.046	0.053 0.047	-1.343 -1.428
1.7	3	0.046	0.035 0.116	-1.331 -1.483
1.8	3	0.046	0.013 0.138	-1.285 -1.331
1.9	3.5	0.046	0.022 0.071	-1.173 -1.346
2.0	4	0.031	0.056 0.059	-1.209 -1.226

Çizelge 4.6'daki sonuçlar ile dört kümeli ÇKT yapısında oluşturulan KTSV'de küme içi boşluk parametresi en iyi 0.5 birim ile kümeleme yaptığı gözlenmiştir.

Şekil 4.8'de üç katmanlı ÇKT'nin veri dağılımı görülmektedir.



(a) **(b)**

(c)

Şekil 4.8 (a) Küme içi boşluk parametresinin 1 birim olduğu veri dağılımı **(b)** Küme içi boşluk parametresinin 5 birim olduğu veri dağılımı **(c)** Küme içi boşluk parametresinin 10 birim olduğu veri dağılımı.

ÇKT yapısında oluşturulan 3 katmanlı KTSV'de küme içi boşluk değişimi analizi yapılırken kümeler arası boşluk 5 birim olarak sabit tutulmuştur. Şekil 4.8 (a)'da küme içi boşluk parametresinin 1 birim olduğu (b)'de küme içi boşluk parametresinin 5 birim olduğu, (c)'de küme içi boşluk parametresinin 10 birim olduğu veri dağılımı görülmektedir.

Çizelge 4.7 'de 6 küme içeren 3 katmanlı ÇKT yapısında oluşturulan KTSV'nin küme içi boşluk parametresinin değişimi ile deneysel sonuçları görülmektedir.

Çizelge 4.7 Üç katman içeren ÇKT yapısında oluşturulan KTSV'nin küme içi boşluk değişimi ile sonuçları.

Küme İçi Boşluk	Hata Oranı (%)	İşlem Süresi (sn)	Çarpıklık		Basıklık	
0.1	0.5	0.096	0.001	0.117	-1.641	-1.660
0.2	0.5	0.126	0.001	0.118	-1.640	-1.660
0.3	0.5	0.093	0.006	0.117	-1.640	-1.659
0.4	0.5	0.109	0.004	0.118	-1.640	-1.660
0.5	0.5	0.109	0.009	0.117	-1.641	-1.660
0.6	0.5	0.109	0.002	0.117	-1.640	-1.660
0.7	0.5	0.109	0.003	0.116	-1.640	-1.660
0.8	0.5	0.110	0.003	0.117	-1.640	-1.660
0.9	0.5	0.093	0.001	0.120	-1.640	-1.657
1	0.5	0.109	0.003	0.118	-1.639	-1.658
2	0.5	0.109	0.001	0.122	-1.637	-1.653
3	0.5	0.109	0.006	0.124	-1.630	-1.643
4	1	0.125	0.006	0.118	-1.620	-1.635
5	1	0.109	0.004	0.111	-1.619	-1.632

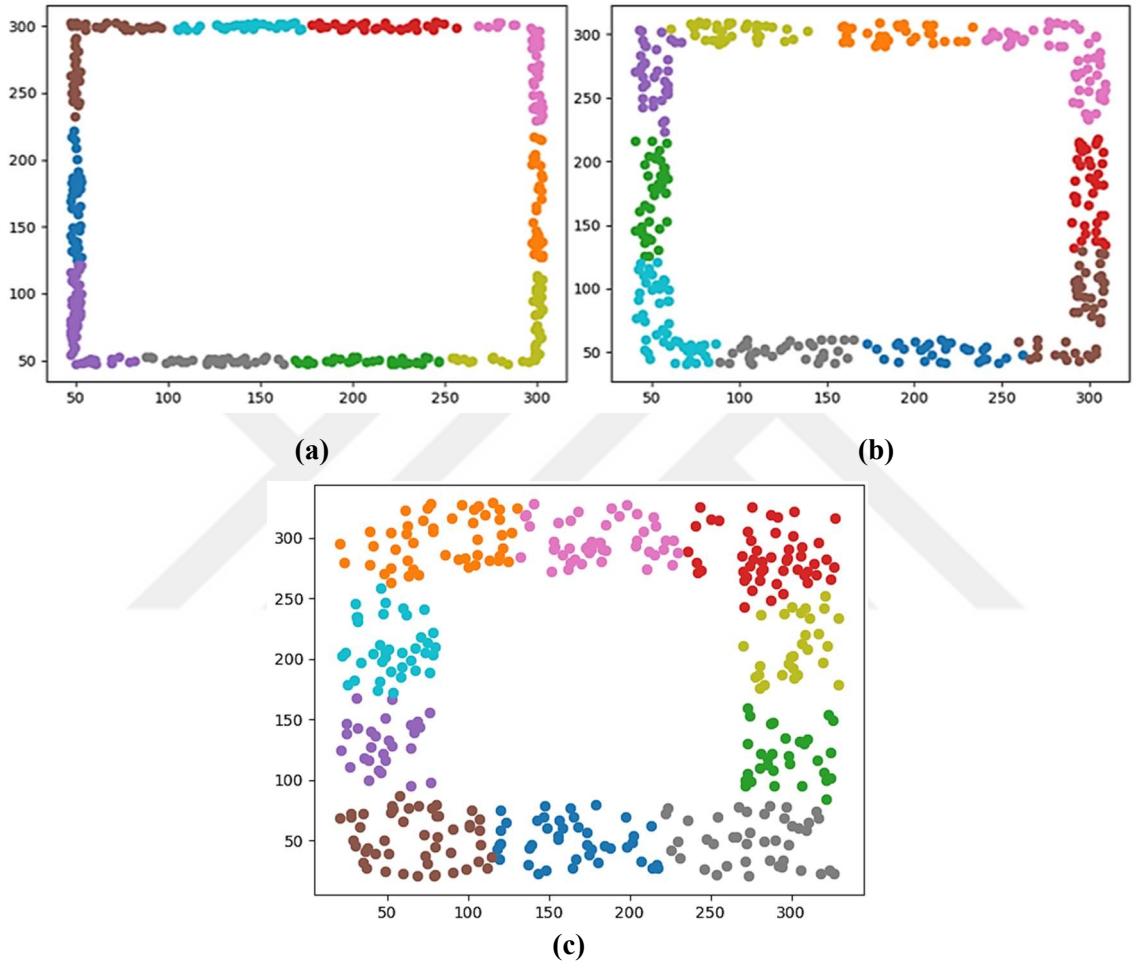
Oluşturulan KTSV'lerin 2 kümeli 200 adet veri içeren TKT'de, 4 kümeli 400 veri içeren 2 katmanlı ve 6 kümeli 600 veri içeren 3 katmanlı ÇKT'de küme içi mesafenin değişimi ile hata sayısı görülmektedir. Küme içi boşluğun değişimi ile boşluğun arttığı durumlarda da az olduğu belirlenen hata sayısı ile kümeleme performansının başarılı olduğu gözlenmiştir.

4.1.4 Kabuk Kalınlığı Parametresinin Değişimi

Bu deneysel çalışmada kabuk kalınlığı parametresinin değişimi ile K-means algoritmasının kümeleme performansı analiz edilmiştir. 100 adet veriden oluşan TKT

yapısında oluşturulan KTSV ile yapılan deneyde küme sayısı 10 adet olmak üzere sabit değer belirlenmiştir. Kabuk kalınlığı parametresinin artması, veri aralıklarının da artması ile birlikte kümeleme performansına etki eder.

Şekil 4.9'da kabuk kalınlığı parametresi değişimi ile veri dağılımları görülmektedir.



Şekil 4.9 (a) Kabuk kalınlığı parametresinin 3 birim olduğu veri dağılımı **(b)** Kabuk kalınlığı parametresinin 10 birim olduğu veri dağılımı **(c)** Kabuk kalınlığı parametresinin 30 birim olduğu veri dağılımı.

Şekil 4.9 (a)'da kabuk kalınlığı parametresinin 3 birim olduğu veri dağılımı, (b)'de kabuk kalınlığı parametresinin 10 birim olduğu veri dağılımı, (c)'de kabuk kalınlığı parametresinin 30 birim olduğu veri dağılımı görülmektedir.

Kabuk kalınlığı parametresinin değişimi ile deneysel sonuçları Çizelge 4.8'de sunulmuştur.

Çizelge 4.8 Kabuk kalınlığı parametresinin değişimi ile sonuçlar.

Kabuk Kalınlığı	İşlem Süresi(sn)	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
1	0.046	0.467	0.033	0.01	0.015
2	0.031	0.457	0.405	0.08	0.121
3	0.046	0.475	0.223	0.085	0.123
4	0.061	0.467	0.5	0.182	0.263
5	0.046	0.43	0.335	0.122	0.178
6	0.031	0.41	0.335	0.097	0.140
7	0.046	0.415	0.5	0.2	0.285
8	0.046	0.435	0.25	0.102	0.145
9	0.046	0.422	0.398	0.155	0.222
10	0.048	0.44	0.375	0.147	0.211
20	0.046	0.347	0.394	0.17	0.233
30	0.039	0.24	0.25	0.08	0.121
40	0.046	0.155	0.036	0.015	0.021
50	0.046	0.155	0.25	0.105	0.147

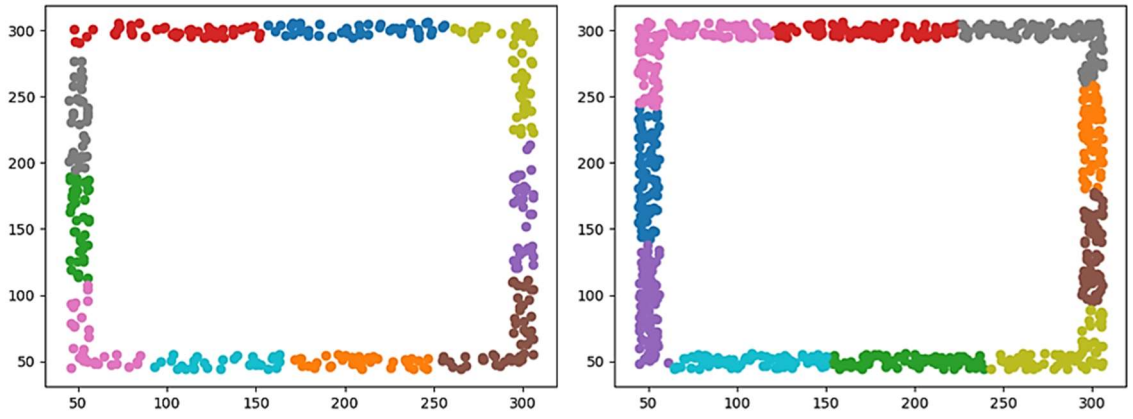
Çizelge 4.8'e göre kabuk kalınlığı parametresinin artması ile birlikte kümeleme performansının azaldığı gözlenmiştir. Kabuk kalınlığı parametresinin değişimi ile kümeleme hızı arasında bir ilişki gözlenmemiştir.

4.1.5 Toplam Veri Sayısı Parametresinin Değişimi

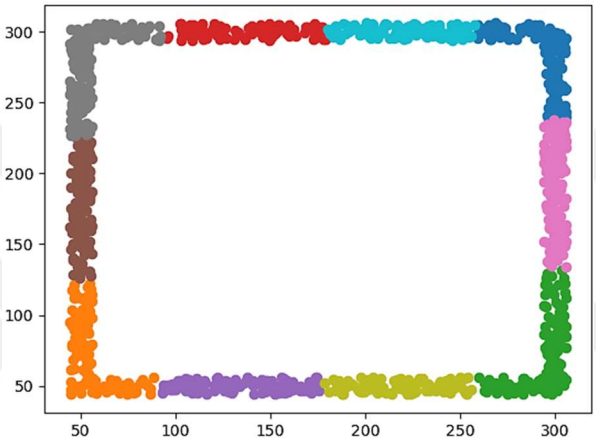
Bu deneysel çalışmada TKT yapısında oluşturulan KTSV'nin toplam veri sayısı değişimine bağlı olarak büyük veri ile kümeleme analizi yapılmıştır. Küme sayısı 10 adet, kabuk kalınlığı parametresi 6 birim olarak sabit değer belirlenmiştir.

Toplam veri sayısının değişimi, veri dağılımı düzenini de değiştirir. Bu durum kümeleme performansında önemli bir etkiye sahiptir. Toplam veri sayısı, kümeleme işleminin hızını belirler. Büyük veri kümeleri üzerinde çalışmak, daha uzun süre ve daha fazla işlemci gücü gerektirir. Veri sayısının artması, analizin tamamlanma süresinin uzamasına veya daha fazla işlemci gücüne neden olur.

Şekil 4.10 (a)'da toplam veri sayısı parametresinin 100 adet olduğu veri dağılımı, (b)'de toplam veri sayısı parametresinin 200 adet olduğu veri dağılımı, (c)'de toplam veri sayısı parametresinin 300 adet olduğu veri dağılımı görülmektedir.



(a) **(b)**



(c)

Şekil 4.10 (a) Toplam veri sayısı parametresinin 100 adet olduğu veri dağılımı **(b)** Toplam veri sayısı parametresinin 200 adet olduğu veri dağılımı **(c)** Toplam veri sayısı parametresinin 300 adet olduğu veri dağılımı.

Şekil 4.10'da toplam veri sayısı parametresinin değişimi ile veri dağılımı görülmektedir.

Deneysel sonuçlar Çizelge 4.9 ile sunulmuştur.

Çizelge 4.9 Toplam veri sayısı parametresinin değişimi ile sonuçlar.

Toplam Veri Sayısı	İşlem Süresi (sn)	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
100	0.046	0.437	0.25	0.09	0.132
200	0.059	0.428	0.586	0.25	0.351
300	0.109	0.417	0.25	0.091	0.134
400	0.140	0.421	0.421	0.045	0.060
500	0.178	0.4	0.024	0.010	0.014
600	0.189	0.396	0.128	0.061	0.082
700	0.207	0.403	0.5	0.194	0.278
800	0.203	0.402	0.430	0.200	0.273

Çizelge 4.9 (Devam) Toplam veri sayısı parametresinin değişimi ile sonuçlar.

Toplam Veri Sayısı	İşlem Süresi (sn)	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
900	0.195	0.398	0.353	0.160	0.220
1000	0.234	0.393	0.065	0.028	0.040
5000	0.531	0.392	0.178	0.082	0.113
10000	0.691	0.389	0.25	0.102	0.145

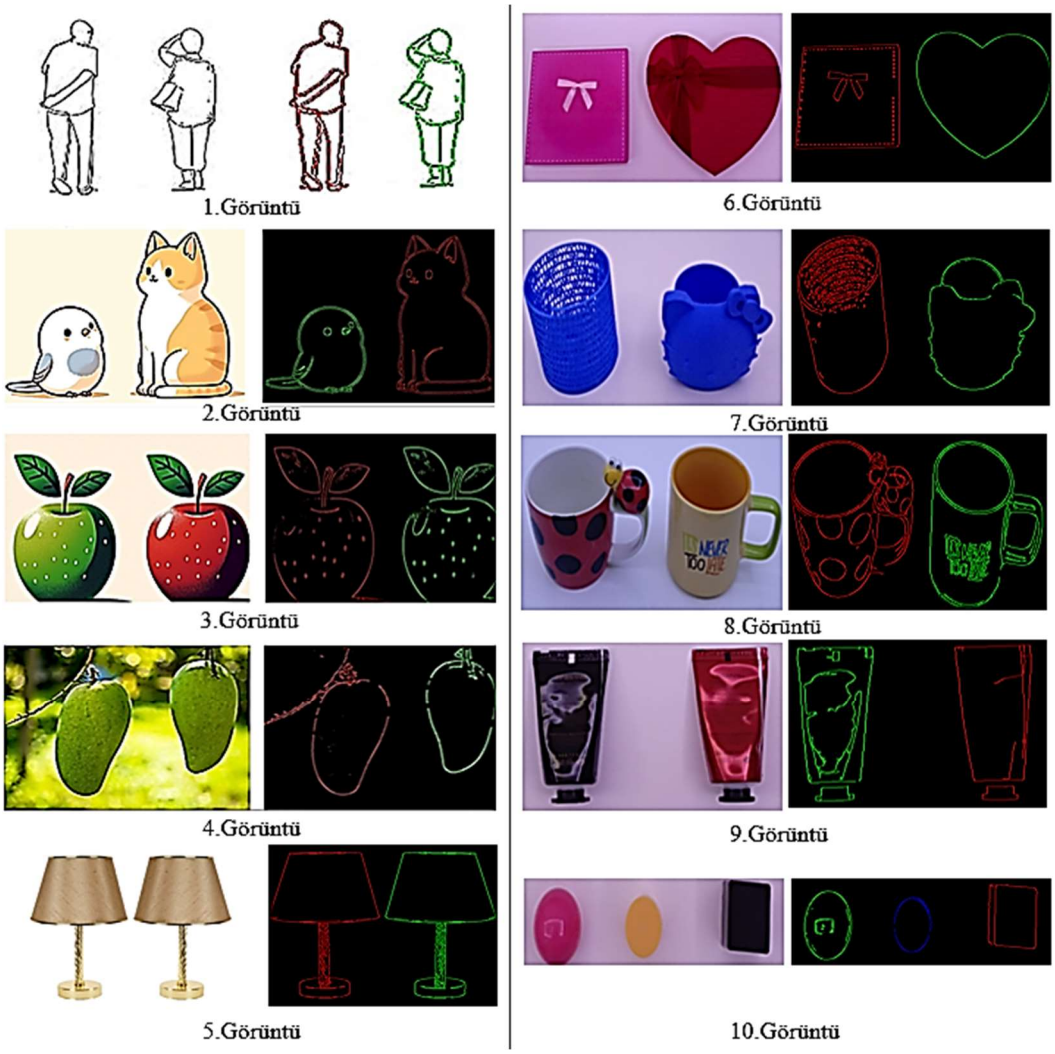
Çizelge 4.9'a göre toplam veri sayısı parametresinin değişimi ile birlikte büyük veri üzerinde kümeleme performansında ciddi oranda bir azalma görülmemiştir. Toplam veri sayısının 100 ve 10000 adet büyük veri üzerinde yapılan deneyler ile yaklaşık %5 oranında kümeleme performans değişikliği görülmektedir. Toplam veri sayısı parametresinin artması ile birlikte işlem süresinin de arttığı görülmektedir.

4.1.6 Kabuk Tipi Veri Setlerinin Görüntü Uygulamaları

Bu bölümde, oluşturulan kabuk tipi veri setlerinin görüntü uygulamaları üzerinde gerçekleştirilen deneysel kümeleme çalışmalarının özgün analizleri çizelgeler ve şekiller halinde sunulmuştur.

Kabuk tipi veri seti, nesnelerin dış sınırlarını tespit ederek nesnelerin tanınmasına ve kümelenmesine katkı sağlar. Bu durum, farklı nesnelerin birbirinden ayrılmasını ve tanınmasını mümkün kılar. Kabuklar, bir nesnenin şeklinin ve yapısının analiz edilmesinde kullanılır. Özellikle tıbbi görüntülerde organların sınırlarını belirlemek için ve insansız hava araçları (İHA), robotlar gibi navigasyon sistemlerinde çevredeki nesnelerin tespiti ve haritalanması için kabuk tipi veri setlerinin görüntü uygulamaları önemlidir.

Nesnelerin kabuk tipi verileri belirlenerek, K-means algoritmasının kümeleme analizi 10 adet görüntüde uygulanmıştır. Şekil 4.11 (a)'da orjinal ve kabuk tipi verilerin kümelenmiş görüntüsü, (b)'de oluşturulan gerçek görüntü veri setinin orjinal ve kabuk tipi verilerin kümelenmiş görüntüsü görülmektedir.



Şekil 4.11 (a) Orjinal ve kabuk tipi verilerin kümelenmiş görüntüsü **(b)** Oluşturulan gerçek görüntü veri setinin orjinal ve kabuk tipi verilerin kümelenmiş görüntüsü.

Şekil 4.11'de kabuk tipi veri setlerinin görüntüleri verilmiştir (İnt.Kyn.3-4).

Bu deneysel çalışmada 10 adet görüntü ile 10 adet deneysel çalışma yapılmıştır. Çizelge 4.10 'da kabuk tipi veri setlerinin görüntü uygulamalarında deneysel sonuçları sunulmuştur.

Çizelge 4.10 Kabuk tipi veri setlerinin görüntü uygulamalarında deneysel sonuçlar.

Görüntü	İşlem Süresi(sn)	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
1.Görüntü	0.109	0.521	0.600	0.500	0.546
2.Görüntü	0.048	0.496	0.373	0.498	0.426

Çizelge 4.10 (Devam) Kabuk tipi veri setlerinin görüntü uygulamalarında deneysel sonuçlar.

Görüntü	İşlem Süresi(sn)	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
3.Görüntü	0.062	0.510	0.547	0.508	0.527
4.Görüntü	0.093	0.495	0.535	0.494	0.513
5.Görüntü	0.062	0.498	0.501	0.487	0.494
6.Görüntü	0.100	0.498	0.238	0.478	0.318
7.Görüntü	0.058	0.499	0.188	0.504	0.274
8.Görüntü	0.046	0.501	0.512	0.501	0.506
9.Görüntü	0.031	0.507	0.618	0.511	0.511
10.Görüntü	0.033	0.544	0.540	0.574	0.557

Çizelge 4.10'a göre en yüksek doğruluk oranı 10. Görüntü ile elde edilmiştir.

5. TARTIŞMA ve SONUÇ

Bu çalışmada, K-means kümeleme yönteminin farklı topolojilere sahip iki boyutlu KTSV'ler kullanılarak özgün bir başarım analizi gerçekleştirilmiştir. Analizler kapsamında toplam 158 adet deney yapılmıştır. Kabuk tipi sentetik verilerin kullanılmasının sebebi; çok boyutlu verilerde verilerin büyük çoğunluğunun veri setinin kabuk kısmında yer almasıdır. Buna örnek olarak görüntü işleme temelli çalışmalarda kullanılan imgelerin kenarları belirlendiğinde verilerin çoğunlukla dış kabuklarda bulunması verilebilir. K-means yönteminde kümelerin başlangıç koordinatları rastgele belirlenirken geliştirilen çalışmada tüm başlangıçların kabuk üzerinde olması sağlanmıştır. Bu da gerçekleştirilen analizin özgün kısımlarından biridir. Bu katkı sayesinde kümeleme süreçlerinin işlem sürelerinin düştüğü tespit edilmiştir. Gerçekleştirilen analiz çalışmasında iki boyutlu verilerin kullanılmasının sebebi; kullanılan verilerin görselleştirilebilmesi ve grafik ortamda sunulabilmesidir. Oluşturulan sentetik verilerin kümeler arası boşluk, küme içi boşluk, küme sayısı, kabuk kalınlığı, eleman sayısı ve katman sayısı gibi özelliklerine göre en başarılı kümeleme performansları işlem zamanı ve doğruluk ölçütleri ile hesaplanarak optimum veri seti yapıları bulunmaya çalışılmıştır. Ayrıca, kullanılan KTSV'lerin çarpıklık ve basıklık gibi istatistiksel özellikleri ile kümeleme performansı arasındaki bağlantı araştırılmıştır.

Yapılan analiz çalışmalarında ulaşılan sonuçların özellikle kenarları çıkartılmış görüntülerde yapılacak bölütleme çalışmalarında kullanılabileceği düşünülmektedir. Çünkü bu tip görüntüler doğaları gereği çok sayıda kabuk tipi TKT ve ÇKT içermekte ve nesne tanıma çalışmalarında kümelenerek birbirlerinden ayrılmaktadır. Böylece literatürde sıklıkla çalışılan nesne sınıflandırma çözümlerinin bir parçası olarak kişi yeniden tanımlama, nesne tespiti, nesne tanımı, otonom sürüş engellerden kaçınma vb. uygulamalarda kullanılabileceği düşünülmektedir.

Yapılan analiz çalışmalarında, en doğru kümeleme performansı kümeler arası boşluk parametresinin 7 birim olduğu deneylerde %100 doğruluk değeriyle gerçekleşmiştir. Bunun sebebi küme merkezine olan mesafelerine göre gruplanan veriler, küme

merkezlerinin birbirinden uzaklaşmasıyla, verilerin ait olduğu kümeyi belirlemesi daha belirgin olduğu düşünülmektedir.

En hızlı kümeleme performansı küme içi boşluk parametresinin 0.1 birim olduğu deneylerde 0.007 sn işlem hızı değeriyle gerçekleşmiştir. Bu değer en düşük küme içi mesafedir. Böylelikle K-means kümelemesinde küme içi mesafenin düşük olması ve kümeler arası mesafenin yüksek olması prensibiyle uyumlu bir sonuca ulaşılmıştır. Bu deneyde, kullanılan KTSV'nin, 2 küme ile en az küme sayısı ve 200 adet veri ile en az veri sayısı içeren TKT yapısında bir veri seti olması bu sonucun elde edilmesinde önemli bir yer tutmaktadır.

Önerilen çalışma kapsamında, özgün kümeleme analizini belirleyen uygun parametrelerin seçimi K-means algoritmasının kümeleme sonuçlarını etkilemektedir. Kabuk kalınlığı parametresinin değiştirilmesi de kümeleme analizi sonuçlarını etkilediği gözlenmiştir. Daha büyük bir kabuk kalınlığı seçimi, daha geniş ve genel kümelerin oluşturulmasına yol açmıştır. Bu durum, veri setindeki genel desenleri vurgularken, detaylardan feragat edilmesine neden olabilir.

Küme sayıları sabit tutularak ve KTSV'lerin kabuk kalınlıklarının değiştirilerek gerçekleştirilen kümeleme deneylerinde en uygun kalınlığın 1 ile 4 birim arasında olduğu tespit edilmiştir. Bu aralıkta TKT için doğruluğun [0.457 - 0.475] aralığında değiştiği gözlenmiştir. Kalınlık çok arttırıldığında KTSV'nin ideal kabuk tipinden serbest dağılıma geçebileceği için bu eşiklerin tespit edilmesi önemlidir.

Küme sayısı sabit tutularak gerçekleştirilen toplam veri sayısı deneylerinde toplam veri sayısı arttıkça doğruluk ve hız değerlerinin azaldığı gözlenmiştir. Bu durumda veri sayısı ile hesapsal maliyet arasında doğrusal bir ilişki olduğu ortaya çıkmıştır.

Küme içi boşluk parametresinin değişimi deneylerinde KTSV'lerin kümeleme doğruluğunun en yüksek olduğu kısımlarda istatistiksel basıklık [-1.641 -1.660] ile [-1.541 -1.736] arasında çıkmıştır. İstatistiksel çarpıklık ise [0.001 0.117] ile [0.010 0.079] arasındadır. Genel olarak deneyler incelendiğinde her ne kadar çok belirgin

sonuçlar olmasa da KTSV'nin basıklık değeri için [-1.641 -1.660] ile [-1.535 -1.768] aralığı en ideal aralık olarak belirlenmiştir.

Yapılan analiz çalışmasında, genel olarak deney sonuçları incelendiğinde K-means algoritmasının kümeleme performansını en çok etkileyen parametrelerin küme sayısı ve kümeler arası boşluk parametresi olduğu gözlenmiştir.

Bu çalışma, oluşturulan KTSV'lerde K-means kümeleme algoritmasının başarılı bir şekilde uygulanmasının yanı sıra, algoritmanın avantajları ve dikkat edilmesi gereken noktalar hakkında kapsamlı bir değerlendirme sunmaktadır. Elde edilen sonuçlar, K-means algoritmasının çeşitli uygulama alanlarında etkili bir şekilde kullanılabileceğini ve veri analitiği süreçlerinde değerli bilgilerin ortaya çıkarılmasına katkı sağlayabileceğini göstermektedir. Ek olarak, bu çalışmanın ortaya koyduğu KTSV'ler, diğer mevcut algoritmalar veya özgün olarak oluşturulan algoritmaların deneyleri ve performans analizlerine de katkı sağlayabilir.

6. KAYNAKLAR

Aggarwal C C, 2015, Data Mining: The Textbook (Vol. 1), Springer, New York.

Ahmed M, Seraj R, Islam S M S, 2020, The K-means Algorithm: A Comprehensive Survey and Performance Evaluation, Electronics (Basel), 9(8), 1295.

Armano G, Javarone M A, 2013, Clustering Datasets by Complex Networks Analysis, Complex Adaptive Systems Modeling, 1, 1-10.

Aslanyürek M, Mesut A, 2021, Kümeleme Performansını Ölçmek İçin Yeni Bir Yöntem ve Metin Kümeleme için Değerlendirmesi, Avrupa Bilim ve Teknoloji Dergisi, (27), 53-65.

Azhar A, Hashim H, 2023, A Review of Wind Clustering Methods Based on the Wind Speed and Trend in Malaysia, Energies (Basel), 16(8), 3388.

Berkhin P, 2006, A Survey of Clustering Data Mining Techniques, ın Grouping Multidimensional Data: Recent Advances in Clustering, Springer, 25-71, Heidelberg, Berlin.

Boboyarov A, Mixliyev R R, 2023, Blood Cell Segmantatıon Images From Mıcroscopıc Blood Images, Journal of Academic Research and Trends in Educational Sciences, 2(2), 71-79.

Bock H H, 2008, Origins and Extensions of The K-means Algorithm in Cluster Analysis, Electronic Journal for History of Probability and Statistics, 4(2), 1-18.

Bouveyron C, Brunet-Saumard C, 2014, Model-Based Clustering Of High-Dimensional Data: A Review, Computational Statistics & Data Analysis, 71, 52-78.

Dandekar A, Zen R A M, Bressan S, 2018, A Comparative Study of Synthetic Dataset Generation Techniques, in: Database And Expert Systems Applications: 29th International Conference, 2018 September 3–6, Regensburg, Germany, Proceedings Part II 29. Springer, 387-395.

Demircioğlu M, Eşiyok S, 2020, Covıd-19 Salgını ile Mücadelede Kümeleme Analizi ile Ülkelerin Sınıflandırılması, İstanbul Ticaret Üniversitesi Sosyal Bilimler Dergisi, 19(37), 369-389.

Dinçer Ş E, 2006, Veri Madenciliğinde K-means Algoritması ve Tıp Alanında Uygulanması, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü.

Domingos P, 2012, A Few Useful Things To Know About Machine Learning, Communications of The ACM, 55(10), 78-87, Washington.

Ghazal T M, 2021, Performances of K-Means Clustering Algorithm with Different distance metrics, Intelligent Automation & Soft Computing, 30(2), 735-742.

Gormley I C, Murphy, T B, Raftery A E, 2023, Model-Based Clustering, Annual Review of Statistics and Its Application, 10, 573-595.

Govender P, Sivakumar V, 2020, Application of K-means And Hierarchical Clustering Techniques For Analysis of Air Pollution: A review (1980–2019), Atmos Pollut Res, 11(1), 40-56.

Gupta M K, Chandra P, 2022, Effects of Similarity/Distance Metrics on K-means Algorithm with Respect To İts Applications in IoT and Multimedia: A review, Multimed Tools Appl, 81(26), 37007-37032.

Han J, Pei J, Tong H, 2022, Data Mining: Concepts And Techniques, Morgan Kaufmann.

Han K, Wang Y, Zhang C, Li C, Xu C, 2018, Autoencoder İnspired Unsupervised Feature Selection, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2941-2945, Canada.

Hançer E, Ozturk C, Karaboga D, 2012, Artificial Bee Colony Based İmage Clustering Method, IEEE Congress on Evolutionary Computation, Australia, 1-5.

Ikotun A M, Ezugwu A E, Abualigah L, Abuhaija B, Heming J, 2023, K-Means Clustering Algorithms: A Comprehensive Review, Variants Analysis and Advances in the Era of Big Data, Inf Sci (N Y), 622, 178-210.

Jain A K, Murty M N, Flynn P J, 1999, Data Clustering: A Review, ACM Computing Surveys (CSUR), 31(3), 264-323.

Jajuga K, 2005, Model-Based Clustering Discussion on Some Approaches, in Data Analysis and Decision Support, Springer, 73-81, Berlin.

Khaleel B I, 2022, A Review of Clustering Methods Based on Artificial Intelligent Techniques, Journal of Education and Science, 31(2), 69-82, Irak.

Mane T U, 2017, Smart Heart Disease Prediction System Using Improved K-means And ID3 on Big Data, in: 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), 239-245.

Miraftabzadeh S M, Colombo C G, Longo M, Foiadelli F, 2023, K-Means and Alternative Clustering Methods in Modern Power Systems, IEEE Access. Published online 2023.

Özari Ç, Özge E, 2018, İllerin Yaşam Endeksi Göstergelerinin Çok Boyutlu Ölçekleme ve K-Ortalamalar Kümeleme Yöntemi ile Analizi, Afyon Kocatepe Üniversitesi Sosyal Bilimler Dergisi, 20(2), 303-313.

Özden F N, Gökçe C O, Sonugür G, 2022, K-Means Kümeleme Algoritmasının Sentetik Kabuk Veri Seti Üzerinde Analizi, 1st International Conference on Scientific and Academic Research (ICSAR), December 10-13, Konya.

Pour S G, Maeder A, Jorm L, 2013, Validating Synthetic Health Datasets for Longitudinal Clustering, in Proceedings of The Sixth Australasian Workshop on Health Informatics and Knowledge Management-Volume 142, Australia, 15-19.

Rajabi A, Li L, Zhang J, Zhu J, Ghavidel S, Ghadi M J, 2017, A Review on Clustering of Residential Electricity Customers and İts Applications, 20th International Conference on Electrical Machines and Systems (ICEMS), Australia, 1-6.

Saxena A, Prasad M, Gupta A, Bharill N, Patel O P, Tiwari A, Lin C T, 2017, A Review of Clustering Techniques and Developments, Neurocomputing, 267, 664-681.

Selcan Ü, 2021, K-means Kümeleme Algoritması ile OECD Ülkelerinin Vergi Yükü ve Kayıtdışı Ekonomi Analizi, Kırklareli Üniversitesi Sosyal Bilimler Meslek Yüksekokulu Dergisi, 2(2), 1-15.

Steinley D, 2006, K-means Clustering: A Half-Century Synthesis, British Journal of Mathematical and Statistical Psychology, 59(1), 1-34.

Szilágyi L, Kovács L, Szilágyi S M, 2014, Synthetic Test Data Generation for Hierarchical Graph Clustering Methods, in International Conference on Neural Information Processing, 303-310, Cham.

Witten I H, Frank E, Hall M A, Pal C J, Data M, 2017, Practical Machine Learning Tools and Techniques, Data Mining, Fourth Edition, Elsevier Publishers.

Yıldız M İ, Ceyhan S, 2023, Ailevi Akdeniz Ateşi'nde Kolşisin Tedavisinin Makine Öğrenmesi Analizi, in: Cognitive Models and Artificial Intelligence Conference, Ankara, Türkiye, 40-44.

Zhang Z, Zha H, 2004, Principal Manifolds and Nonlinear Dimensionality Reduction Via Tangent Space Alignment, Journal of Shanghai University (English Edition), 8(4), 406-424.

Zhu X, Melnykov V, 2015, Probabilistic Assessment of Model-Based Clustering, Advances in Data Analysis and Classification, 9, 395-422.

İnternet Kaynakları

1- http:// www.gatevidyalay.com/, 26.09.2023

2- https://en.wikipedia.org/, 03.01.2024

3- https://www.copilot.com/,05.02.2024

4- https://www.piqsels.com/tr/,05.02.2024

EKLER

EK 1. Tez için yapılan yayınlar.

Detailed Performance Analysis of K-Means Method on Synthetic Shell-Type Datasets with Different Topologies

Feyza Nur ÖZDEN[1]*, Celal Onur GÖKÇE[2], Güray SONUGÜR[1]
[1] *Afyon Kocatepe University, Mechatronics Engineering Department, Afyonkarahisar-TURKEY*
[2] *Afyon Kocatepe University, Software Engineering Department, Afyonkarahisar-TURKEY*
**feyzanurozden97@gmail.com*

ABSTRACT
In this study, the performance analysis of the K-means clustering algorithm has been conducted on shell-type synthetic datasets with parameters such as the number of clusters, inter-cluster gap, intra-cluster gap, and cluster topology. This type of dataset was chosen due to the fact that in high-dimensional real datasets, the majority of data distribution mass is not near the centre but rather in the surrounding shell region. Synthetic datasets are artificially generated datasets with similar characteristics to real data, produced using mathematical methods. Synthetic datasets are used when real datasets are not suitable, often due to their high dimensionality and certain constraints such as security issues. Therefore, the aim is to evaluate algorithms with synthetic datasets and facilitate the analysis of the performance of proposed methods. Through experimental studies, the relationship between the performance of K-means and parameters such as the number of clusters, inter-cluster gap, intra-cluster gap, and cluster topology has been observed, and the results are presented in tables.
KEYWORDS *- Clustering, K-means, Shell-type dataset.*

1. INTRODUCTION

Data science algorithms have a significant impact on information processing and decision-making processes in today's technology. They are used to analyse high-dimensional data, determine relationships between data, and extract meaningful information from data. These algorithms process data using principles from statistics, mathematics, and computer science to generate desired results. The outcomes find applications in various fields such as medical diagnosis, risk management, and optimizing marketing strategies. Clear understanding of analyses is crucial in different sectors, including the defence industry [1]. To obtain meaningful information from data, several processes, such as cleaning erroneous data and reducing dimensions in high-dimensional data, are applied during the data preprocessing stage. Cluster analysis is a technique used to group data points with similar features into the same cluster. It identifies different data groups by placing similar data points into the same clusters. Cluster analysis can be applied to various fields such as marketing, customer segmentation, and healthcare. The most commonly used algorithms in cluster analysis are K-means, Hierarchical Clustering, DBSCAN, and Gaussian Mixture Model. Each algorithm is suitable for different data structures. The K-means algorithm, in particular, is a clustering method used to create groups based on distance measures of data.

When reviewing the literature, it is observed that the K-means clustering algorithm has been applied in many studies, such as grouping diseases [2], clustering drug data [3], and selecting potential customers [4]. Real datasets are often high-dimensional, making it challenging to visualize the data. However, the use of synthetic datasets allows for the creation of data up to three dimensions, facilitating easy visualization and analysis. Consequently, the developed algorithms can be easily tested. In this context, Özden et al. [5] conducted a clustering analysis on two-clustered shell-type datasets as part of their study.

In the proposed method, detailed performance analysis of the K-Means algorithm is conducted by changing various parameters such as the number of clusters, inter-cluster distance, intra-cluster distance, by employing shell-type synthetic datasets with nested structured square and rectangular topologies.

EK 1.(Devam) Tez için yapılan yayınlar

2. MATERIALS AND METHODS

2.1. K-means Clustering Algorithm

The K-means algorithm is an unsupervised learning algorithm that clusters data based on distance measures. It groups data into clusters according to the number of clusters specified by the user. Different techniques can be applied for distance measurement based on the features of the data. In this study, the Euclidean distance is used in the K-means clustering method. The K-means algorithm is initiated with the user-specified number of clusters. Cluster centres are randomly generated for the specified number of clusters. Distances between the data and the created centres are calculated, and data points are assigned to the nearest cluster centre. The centres of the newly formed clusters are determined, and this process continues until the cluster centres remain constant.

2.2. The Proposed Method

In high-dimensional datasets, the majority of the Gaussian distribution is not centred but rather situated in an "envelope" towards the edges of the data [6]. In this context, in the conducted study, a synthetic dataset with data density in the envelope part was generated in a two-dimensional space. These types of datasets are referred to as Shell-Type Synthetic Datasets (STSD). In the study, STSDs were created in square and rectangular topologies. Figure 1 shows a square STSD, which consists of 200 data points.

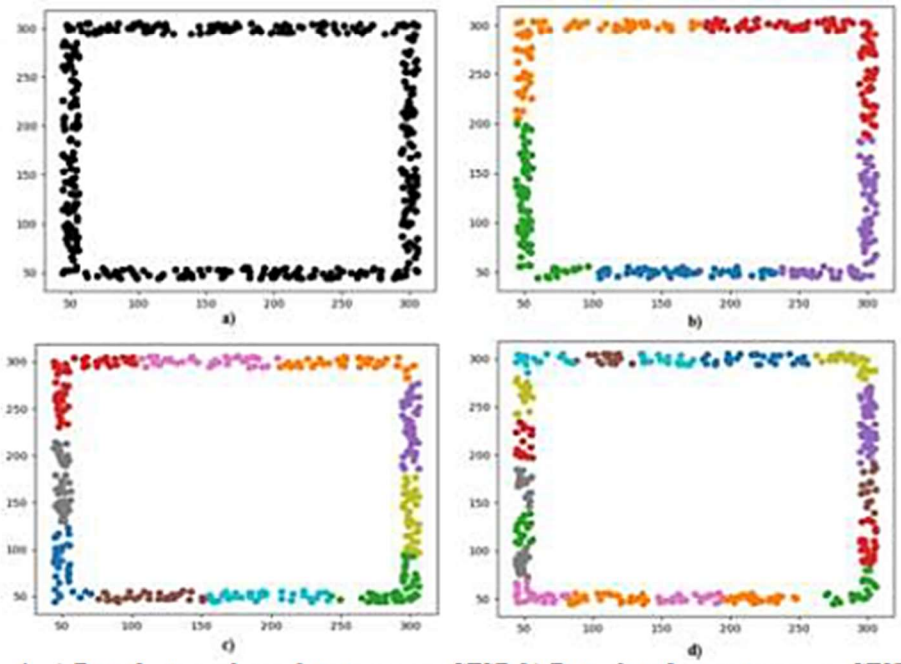


Figure 1. a) Distribution of raw data in square STSD b) Data distribution in square STSD with 5 clusters c) Data distribution in square STSD with 10 clusters d) Data distribution in square STSD with 20 clusters

In Figure 2, raw data is shown, generated in two dimensions to create four clusters with nested rectangular topologies. This STSD is formed from 200 data points.

EK 1.(Devam) Tez için yapılan yayınlar

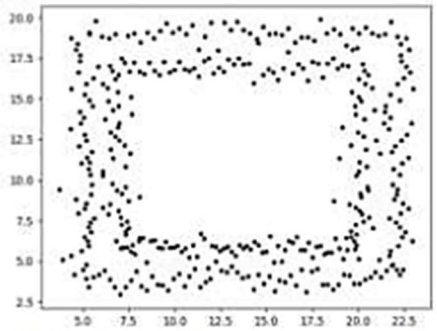


Figure 2. Distribution of raw data in rectangular STSD

In Figure 3, the change in inter-cluster distance of the rectangular STSD is shown along with the data distribution in the coordinate plane. In the performance analysis of the K-means clustering algorithm with the change in inter-cluster distance, the intra-cluster distance parameter is kept constant at 0.5.

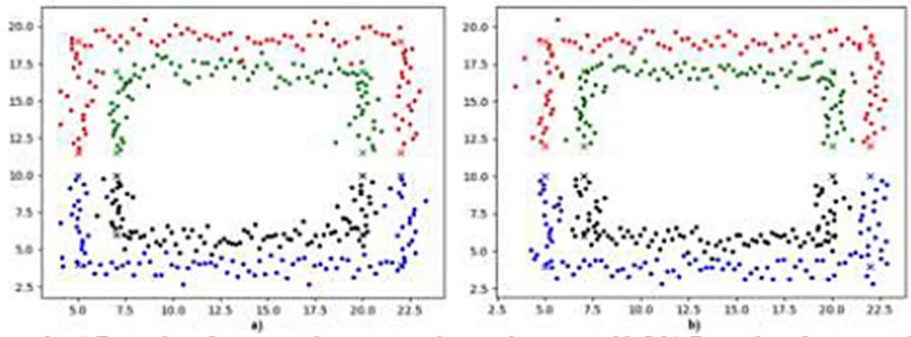


Figure 3. a) Data distribution with an inter-cluster distance of 1.5 b) Data distribution with an inter-cluster distance of 2

In Figure 4, the change in intra-cluster distance of the rectangular STSD is shown along with the data distribution in the coordinate plane. In the performance analysis of the K-means clustering algorithm with the change in intra-cluster distance, the inter-cluster distance parameter is kept constant at 1.5.

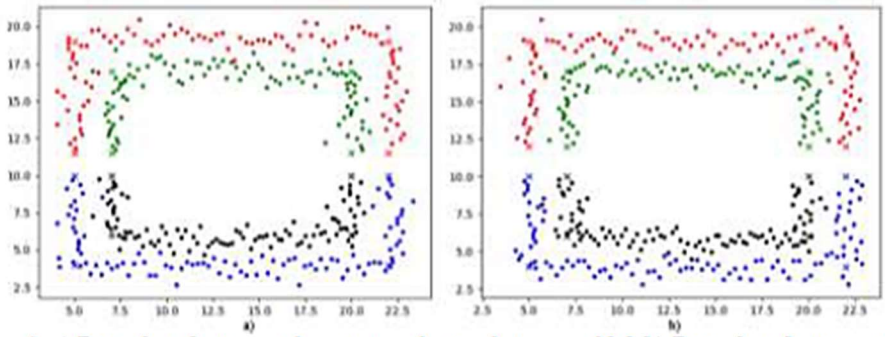


Figure 4. a) Data distribution with an intra-cluster distance of 0.2 b) Data distribution with an intra-cluster distance of 0.5

85

EK 1.(Devam) Tez için yapılan yayınlar

3. CONCLUSIONS

Table 1 provides accuracy rates for the STSD containing 200 data points with a square topology, based on the number of clusters. It is observed that as the number of clusters increases, the performance of the K-Means Clustering improves.

Table 1. Analysis of the square STSD with varying cluster numbers

Number of Clusters	Accuracy Rate (%)
5	13.5
10	44.5
15	67.5
20	87.5
25	96.5
26	97
27	98.5
28	99.25
29	99.50
30	100

Table 2 shows the change in inter-cluster distances and intra-cluster distances along with the error count for the rectangular topology with 4 clusters using the generated 200 data points.

Table 2. Inter-cluster distance, intra-cluster distance change, and error count with the rectangular STSD

Inter-cluster Distance	Error Count (%)	Intra-cluster Distance	Error Count (%)
0.1	1.5	0.1	0
0.5	1.5	0.5	0
1	1	1	2
1.5	1	1.5	2.5
1.6	0.25	1.6	2.5
1.7	0	1.7	3
1.8	0	1.8	3

In the proposed study, it has been observed that the clustering performance of the K-means algorithm is successful with the variation of various parameters in the generated STSD.

REFERENCES

[1]. O. B. Özden, B. Gökçe, Fatigue life analysis of welded joints in the frequency plane in a structure designed for the defense industry, European Mechanical Science (2023) 184-191. DOI:10.26701/ems.1309271

[2]. C. A. Sugianto, A. H. Rahayu, A. Gusman, Algoritma K-Means Untuk Pengelompokkan Penyakit Pasien Pada Puskesmas Cigugur Tengah, Journal of Information Technology (JOINT), 39–44, 2020.

[3]. G. Gustientiedina, M. H. Adiya, Y. Desnelita, Penerapan Algoritma K-Means Untuk Clustering Data Obat-Obatan, Jurnal Nasional Teknologi Dan Sistem Informasi 5.1 (2019) 17-24.

[4]. P, R. Rian, C. Wadisman, Implementasi Data Mining Pemilihan Pelanggan Potensial Menggunakan Algoritma K Means, INTECOMS: Journal of Information Technology and Computer Science 1.1 (2018) 72-77.

EK 1.(Devam) Tez için yapılan yayınlar

[5]. F. N. Özden, C. O. Gökçe, G. Sonugür, K-Means Kümeleme Algoritmasının Sentetik Kabuk Veri Seti Üzerinde Analizi, 1st International Conference on Scientific and Academic Research (ICSAR), 10-13 December, 2022 Konya-Turkey.
[6]. P. Domingos, A Few Useful Things to Know About Machine Learning, communications of the acm, October 2012.

EK 2. Tez için yapılan yayınlar

1st International Conference on Scientific and Academic Research

ICSAR

https://www.icsarconf.com/

December 10-13, 2022, Konya, Turkey

K-Means Kümeleme Algoritmasının Sentetik Kabuk Veri Seti Üzerinde Analizi

Feyza Nur Özden[1*], Celal Onur Gökçe [2] ve Güray Sonugür [3]

[1]Mekatronik Mühendisliği / Fen Bilimleri Enstitüsü, Afyon Kocatepe Üniversitesi, Türkiye
[2]Yazılım Mühendisliği / Fen Bilimleri Enstitüsü, Afyon Kocatepe Üniversitesi, Türkiye
[3]Mekatronik Mühendisliği / Fen Bilimleri Enstitüsü, Afyon Kocatepe Üniversitesi, Türkiye

*(feyzanurozden97@gmail.com)

Özet – Bu çalışmada kabuk tipi sentetik bir veri seti kullanılarak k-means algoritmasının performans analizi yapılmıştır. Sentetik veri seti, görselleştirebilmek için iki boyutlu uzayda üretilmiştir. Veri setinde iki küme kullanılmıştır. Yüksek boyutlu uzaylardaki gerçek veri setlerinin önemli bir kısmı kabuk şeklinde olduğu için bu tip seçilmiştir. Kümeler arası mesafe parametreleri değiştirilerek performans analizi yapılmış ve sonuçlar raporlanmıştır. Veri setinin performansa etki eden en önemli parametreleri, küme içi ortalama mesafe ve kümeler arası en yakın boşluk olarak gözlemlenmiştir. Bu iki parametrenin birbirlerine oranı performansı etkilemektedir. Özellikle kümeler arası en yakın boşluğun performans üzerinde ciddi etkisi olduğu gözlemlenmiştir. İleride yapılması hedeflenen çalışmalardan birisi yapılan analizin MNIST gibi gerçek veri setlerinde de geçerli olduğunu göstermektir. Gerçek veri setleri genellikle çok yüksek boyutlu uzaylarda yer aldıkları için görselleştirilmeleri mümkün değildir. Bu yüzden iki boyutlu sentetik veri seti kullanılarak kavramların daha kolay analiz edilmesi amaçlanmıştır. Üretilen iki boyutlu veri seti başka algoritmaların test edilmesi için de kullanılabilir ki bu da ileride yapılması hedeflenen çalışmalardan birisidir. İleride yapılması planlanan çalışmalardan bir diğeri sentetik veri setinin türlerini ve meta parametrelerini değiştirerek farklı veri seti türleri için farklı algoritmaların performans analizini yapmaktır. Bu çalışma ile yapılan deneylerde k-means yöntemi kullanılarak yapılan kümeleme performanslarının kümeler arası mesafe değişimi ile arasındaki ilişki başarı ile gözlenmiştir.

Anahtar Kelimeler – Kümeleme, k-means, kabuk veri seti, sentetik veri seti, performans analizi

I. GİRİŞ

Teknolojinin hızla gelişmesiyle beraber yapay zekâ alanlarında da çalışmalar artmıştır. Yapay zekâ alt dallarından birisi de makine öğrenmesidir. Makine öğrenmesi, giriş verileri kullanılarak bu verilerden öğrenme ve iyileştirme yöntemleri ile sonuçların otomatik olarak makineler tarafından tahmin edilmesidir. Makine öğrenmesi bilgisayarlı görü, nesne tanıma, nesnelerin sınıflandırılması, doğal dil işleme, otomotiv, veri güvenliği, biyomedikal ve tıp gibi pek çok alanda kullanılmaktadır.

Sonuçların makineler tarafından tahmin edilebilmesi için veri anlama aşaması oldukça önemlidir. Özellikle büyük verileri anlama aşaması oldukça zor olmaktadır. Veri ön işlemede, veri seti üzerinde yapılan veri boyutunun küçültülmesi, tekrar edilen verilerin kaldırılması, eksik verilerin uygun şekilde tamamlanması gibi birçok işlem verileri anlama aşaması için kullanılır.

Kümeleme analizi, verilerin benzerliklerine göre farklı gruplara ayıran bir tekniktir. Kümeleme işlemi tamamen verinin özelliklerine göre yapılır [1]. K-means algoritması da verilerin mesafelerine göre en yakın merkezdeki kümeye eklenmesi temeline dayanan bir kümeleme yöntemidir. K-means algoritması en çok kullanılan kümeleme yöntemlerinden biridir.

EK 2.(Devam) Tez için yapılan yayınlar

Literatürde k-means kümeleme yöntemi kullanılarak yapılan fındık meyvesinin tespit ve sınıflandırılması [2], trafik kaza desenlerinin tanımlanması [3], sediment taşınımının tespiti [4] ve tıp alanında uygulanması [1] gibi pek çok çalışmanın olduğu görülmektedir.

Bu çalışmada kabuk tipi sentetik bir veri seti kullanılarak k-means algoritmasının performans analizi yapılmıştır. Veri setinde iki küme seçilmiştir. Kabuk tipi sentetik veri setinde mesafenin değiştirilmesi ile de yüksek performans elde edildiği gözlenmiştir.

II. MATERYAL VE YÖNTEM

A. *K-MEANS KÜMELEME YÖNTEMİ*

K-means yaygın olarak kullanılan kümeleme algoritmalarından biridir. K-means kümeleme algoritması gözetimsiz öğrenmedir. Yani verilerin önceden bir etiketi, bilgisi yoktur. Bu yöntemde, N adet veri kullanıcı tarafından belirlenen K adet kümeye verilerin mesafe benzerliklerine göre ayırılır. K-means kümeleme yöntemi ile oluşan kümelerin aynı küme içinde benzerliklerin en fazla, kümeler arası benzerliklerin en az olması amaçlanmaktadır.

Mesafe ölçüm yöntemleri olarak kümeleme algoritmalarında en çok uygulanan Öklid mesafesi, Manhattan mesafesi [5] ve Minkowski [6] mesafe yöntemleridir. Bu çalışmada K-means kümeleme yönteminde Öklid mesafesi kullanılmıştır.

K-means algoritmasının aşamaları kısaca aşağıdaki gibi tanımlanabilir;

a) Kullanıcıdan küme sayısını belirlemesi istenir.

b) Belirlenen küme sayısı kadar rastgele küme merkezi koordinatı belirlenir.

c)Tüm verilerin rastgele belirlenen küme merkezlerine Öklid mesafeleri hesaplanır.

d)Hesaplanan mesafelere göre tüm veriler en yakın küme merkezine atanır.

e) Yeni oluşan kümelerin merkezleri belirlenir.

f) "c" maddesine dönülerek küme merkezleri sabitlenene kadar döngüsel olarak işlemler tekrar edilir.

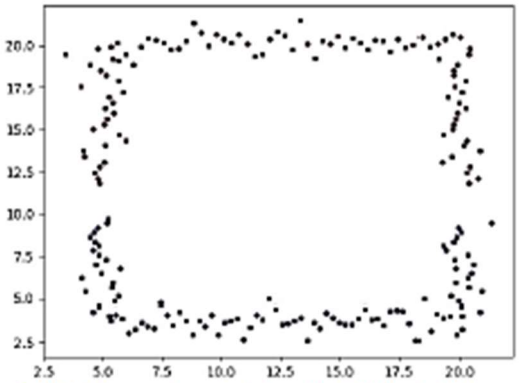
B. *ÖNERİLEN YÖNTEM*

Önerilen çalışma başlıca üç aşamadan oluşmaktadır. İlk aşamada kabuk tipi sentetik veri seti oluşturulmuştur. Yüksek boyutlarda, çok değişkenli bir Gauss dağılımının kütlesinin çoğu ortalamanın yakınında değil, etrafındaki giderek uzaklaşan bir "kabuk" içindedir. Örneğin, ince bir portakalın hacminin çoğu kabuğundadır, özünde değil [7]. Bu nedenle önerilen çalışmada kullanılmak üzere 100 adet veriden oluşan kabuk tipi sentetik veri oluşturulmuştur. K-means kümeleme yöntemi ile veriler mesafelerine göre en yakın merkezdeki kümeye eklenerek iki adet küme merkezi belirlenmiştir. Son aşama olarak kümeler arası boşluk mesafeleri değiştirilerek kümeleme performansı gözlenmiştir.

Şekil 1: Önerilen çalışmanın akış şeması

III. BULGULAR

Deneysel çalışmalarda kullanılmak üzere üretilen iki boyutlu veri setinin koordinat düzlemindeki dağılımı Şekil 2' de verilmiştir.

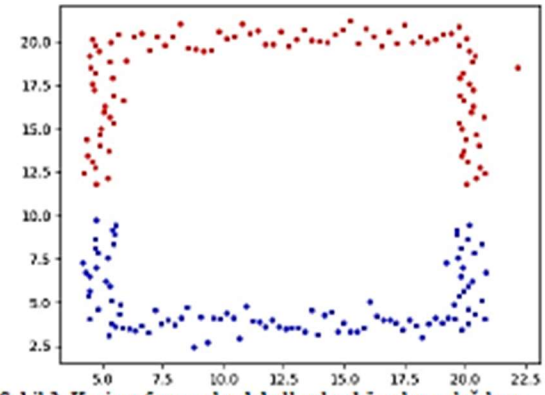


Şekil 2: Işlenmemiş verilerin koordinat düzlemindeki dağılımları

Yapılan kümeleme çalışmasının performansını belirlemek üzere kesin referans (ground truth)

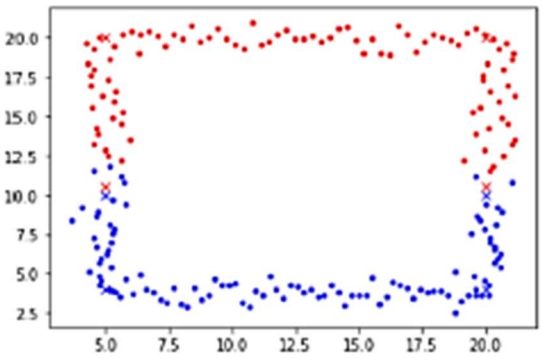
EK 2.(Devam) Tez için yapılan yayınlar

olarak kullanılan kümeleme dağılımı Şekil 3'de verilmiştir.



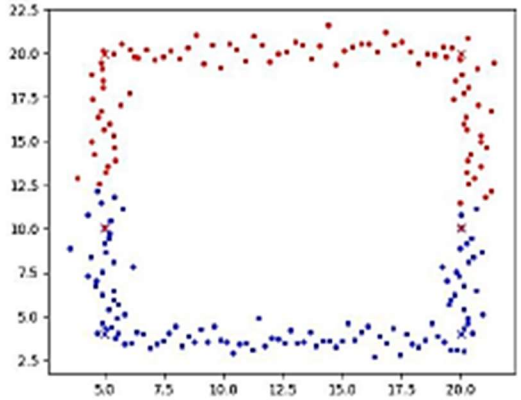
Şekil 3: Kesin referans olarak kullanılan kümeleme dağılımı

Şekil 4' de kabuk tipi sentetik verinin oluşturulması ile çalışmada kullanılan gerçek veriler görülmektedir. Verilerde kabuk şeklinin oluşması için koordinatlar maksimum ve minimum değerler etrafında belirli bir bant aralığında rastgele uzaklıklarda belirlenmiştir.

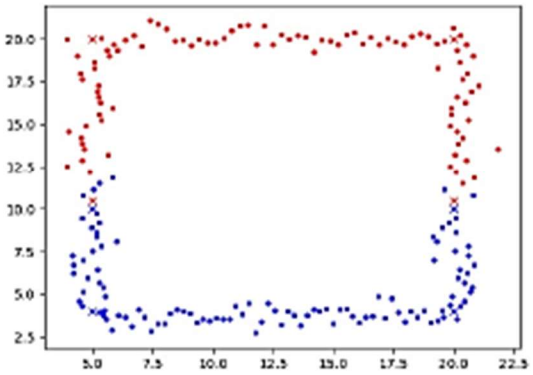


Şekil 4: Önerilen çalışmada oluşturulan sentetik veri

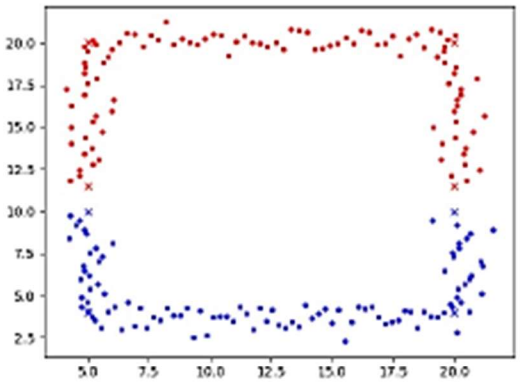
Kümeleme performansını ölçmek amacıyla 15 adet deneysel çalışma gerçekleştirilmiştir. Bu çalışmalarda kümeler arası mesafeler 0.1 ile 1.5 birim arasında değiştirilmiş ve hatalı kümelenen veri sayıları gözlenmiştir. Şekil 5, Şekil 6 ve Şekil 7'de sırasıyla 0.1, 0.5 ve 1.5 birim kümeler arası mesafelere sahip dağılımlar grafiksel olarak gösterilmiştir.



Şekil 5: Kümeler arası mesafenin 0.1 olduğu veri dağılımı



Şekil 6: Kümeler arası mesafenin 0.5 olduğu veri dağılımı



Şekil 7: Kümeler arası mesafenin 1.5 olduğu veri dağılımı

Tablo 1'de oluşturulan 100 adet veri ile kümeler arası boşluk mesafelerinin değişimi ile hata sayısı görülmektedir. Önerilen çalışma ile 100 adet veriye göre kümeler arasındaki boşluk mesafesinin az olduğu durumlarda da hata sayısının az olmasıyla birlikte istenilen sonuca ulaşılmıştır.

EK 2.(Devam) Tez için yapılan yayınlar

Tablo 1: Kümeler Arası Boşluk Mesafesi ile Hata Sayısı

Kümeler Arasındaki Boşluk Mesafesi	Hata Sayısı
1.5	0
1.4	0
1.3	0
1.2	1
1.1	1
1.0	2
0.9	2
0.8	3
0.7	5
0.6	5
0.5	5
0.4	5
0.3	6
0.2	7
0.1	7

IV. SONUÇLAR

Önerilen çalışmada kabuk tipi sentetik verinin iki boyutlu uzayda oluşturulması ile kümeleme performansının başarılı olduğu gözlenmiştir. Kabuk tipi sentetik veri arasındaki boşluk mesafesinin değişimi ile istenilen kümeleme verimi elde edilmiştir. Gelecek çalışmalarda yüksek boyutlu uzaylarda yapılacak çalışmalarda da verilerin kabuklarda toplandığı varsayılarak benzer kümeleme algoritmaları geliştirilecektir. Bu şekilde iki boyuttan daha büyük ölçekte öznitelikler içeren nesnelerin kümelemesi yapılabilecektir.

KAYNAKLAR

[1] Ş.E.Dinçer, "Veri Madenciliğinde K-Means Algoritması ve Tıp Alanında Uygulanması", Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, 101s, 2006.

[2] S.Solak, and U.Altınışık, "Görüntü işleme teknikleri ve kümeleme yöntemleri kullanılarak fındık meyvesinin tespit ve sınıflandırılması", Sakarya University Journal of Science 22.1, pp.56-65, 2018.

[3] S.Güner, K. Seçkin Codal , H. S. Geçer, and E. Coşkun, "Trafik Kaza Desenlerinin Tanımlanmasında K-means Kümeleme Algoritmasının Kullanılması: Sakarya İli Uygulaması", İşletme Bilimi Dergisi , 6 (3) , pp.89-105 , 2018.

[4] K. Saplıoğlu and R. ACAR, "K-Means Kümeleme Algoritması Kullanılarak Oluşturulan Yapay Zekâ Modelleri ile Sediment Taşınımının Tespiti", Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 9(1), 306-322, 2020.

[5] B. Gökçe, and G. Sonugür, "Recognition of dynamic objects from UGVs using Interconnected Neural network-based Computer Vision system", Automatika, 63(2), 244-258, 2022.

[6] M. Iqbal, G. M. Putra, N. Puspitasari, H. J. Setyadi, F. A. Dwiyanto, A. P. Wibawa, and R. Alfred, "A Performance Comparison of Euclidean, Manhattan and Minkowski Distances in K-Means Clustering", In 2020 6th International Conference on Science in Information Technology, ICSITech, pp. 184-188. IEEE, October 2020.

[7] P. Domingos, "A Few Useful Things to Know About Machine Learning", communications of the acm, vol. 55, No:10, October 2012.