

## Makine Öğrenmesi Yöntemleri Kullanılarak Mahkeme Kararlarının Kümelenmesi

### Clustering Court Orders Using Machine Learning Methods

Muhammed Burak GÖRENTAŞ<sup>1</sup> , Taner Uçkan<sup>2</sup> 

<sup>1</sup>Başkale Meslek Yüksekokulu, Van Yüzüncü Yıl Üniversitesi, Van, Türkiye

<sup>2</sup>Bilgisayar Mühendisliği Bölümü, Van Yüzüncü Yıl Üniversitesi, Van, Türkiye

(burakgorentas@yyu.edu.tr, taneruckan@yyu.edu.tr)

Received:Jun.22, 2023

Accepted:Oct.6, 2023

Published:Dec.20, 2023

**Özetçe**— Yapay zekâ son yıllarda hızlı bir şekilde gelişen bir teknolojidir ve hayatın hemen her alanında uygulanma olanağı bulmuştur. Sağlık, otomotiv, eğitim, müzik, finans, tarım ve daha birçok alanda yapay zekâ kullanılmaya başlanmıştır. Bu alanlardan biri de hukuktur. Hukuk alanında yapay zekanın birçok uygulama ortamı bulunmaktadır. Hukuk araştırmaları, dava yönetimi, hukuk danışmanlığı, hukuki dil analizi, içtihat taramaları, hukuki risk analizi gibi yardımcı araç olarak kullanımının yanında yargısal kararların analizi gibi kullanımları da mevcuttur. Yapay zekâ hukuk alanında doğal dil işleme teknolojisi kullanılarak birçok uygulama geliştirilmiştir. Metin kümeleme bu uygulama alanlarından biridir. Metin kümeleme, doğal dil işleme ve makine öğrenmesinde kullanılan bir tekniktir ve içerik veya dilbilimsel özelliklerine göre benzer metinleri gruplandırmaya yardımcı olmaktadır. Özellikle hukuk alanında karmaşık ve geniş bir metin kümesi olduğundan kümeleme yöntemleri değerli bir katkı sunmaktadır. Bu yöntemler, belirli bir konuda benzer niteliklere sahip davaları gruplandırarak, hukuki prensipleri ve yargısal eğilimleri daha iyi anlamamıza yardımcı olmaktadır. Kümeleme yöntemleri, hukuki araştırmacıların geniş bir dava yelpazesine hızlı bir şekilde erişmelerini sağlaması ve hukuki analiz sürecini iyileştirmesi gibi avantajlar sunmaktadır. Ayrıca, kümeleme sonuçları, hukuki stratejilerin geliştirilmesi, dava öncesi hazırlık ve hukuki kararların temellendirilmesi gibi birçok farklı alanda kullanılmaktadır. Bu çalışmada Uyuşmazlık Mahkemesi kararları TF-IDF yöntemi ile doğal dil işleme sürecinden geçirilmiş ve ardından CURE, K-MEANS, DBSCAN, AGNES, AFFINITY ve BIRCH gibi yapay zekâ yöntemleri ile kümelendi. Değerlendirme metriklerine göre en iyi sonucu BIRCH algoritmasının verdiği görülmüştür.

**Anahtar Kelimeler** : Hukuk, Metin Kümeleme, Yapay Zekâ.

**Abstract**— Artificial intelligence (AI) is a rapidly evolving technology that has found applications in various fields of life. It has also made its way into the legal domain. AI has numerous applications in the legal realm, including legal research, case management, legal consultancy, legal language analysis, case precedents analysis, and legal risk assessment. Natural language processing (NLP) techniques have been employed to develop various AI applications in the legal field. Text clustering is one such application. Text clustering is a technique used in NLP and machine learning to group similar texts based on their content or linguistic features. Given the complexity and vastness of legal texts, clustering methods provide valuable contributions. These methods aid in grouping cases with similar attributes in a specific subject area, thus helping us better understand legal principles and judicial trends. Clustering techniques offer advantages such as facilitating quick access to a wide range of cases for legal researchers and improving the legal analysis process. Furthermore, the outcomes of clustering can be utilized in diverse areas, including the development of legal strategies, pre-trial preparations, and substantiating legal decisions. In this study, decisions of the Dispute Resolution Court were subjected to natural language processing using the TF-IDF method, followed by clustering using AI techniques such as CURE, K-MEANS, DBSCAN, AGNES, AFFINITY, and BIRCH. Based on evaluation metrics, the BIRCH algorithm yielded the best results.

**Keywords** : Artificial Intelligence, Law, Text Clustering.

## 1. Giriş

Yapay zekâ, bilgisayarların insana benzer bir şekilde öğrenmesini ve veri analizi yapmasını sağlayan bir teknolojidir. Yapay zekâ son yıllarda hızlı bir şekilde gelişen bir teknoloji olmuştur ve birçok alanda uygulanabilir potansiyele sahiptir. Yapay zekanın özellikle veri yoğun alanlarda kullanımı artmaktadır. Yapay zekâ, hukuk alanında birçok farklı uygulama alanı bulmaktadır. Hukukun tarihini ve gelişimini incelemek için büyük miktarda veriyi veya hukuki makaleleri ve dava dosyalarını analiz ederek hukuki trendleri tespit etmek, müşterilerin hukuki durumlarını analiz, belgeleri kontrol ve müşterileri belirli yasal işlemler hakkında bilgilendirmek ayrıca belgeleri analiz ederek dava sürecini takip etmek ve hukuki belgelerin doğruluğunu kontrol etmek gibi birçok alanda yapay zekâ kullanılmaktadır. Hukuki belgelerin dilini analiz ederek, anlaşılması daha kolay hale getirilmesinde yapay zekadan yararlanılmaktadır. Bunların yanında yapay zekâ, benzer davalardan elde edilen verileri kullanarak, bir hâkim veya avukatın belirli bir durumda hangi kararı vereceğini tahmin ederek, mahkeme kararları için bir öngörüleyici analiz yapmaktadır.

Bu tip hukuka yardımcı yapay zekâ uygulamalarında analiz yapılabilmesi için hukuki metinlerin doğal dil işleme yöntemleriyle hazır hale getirilmesi gerekmektedir. Doğal dil işleme teknolojisi, hukuk alanında yapay zekâ uygulamaları için oldukça önemli bir bileşendir. Bu teknolojinin kullanımı, avukatların daha hızlı ve daha doğru hukuk metinlerini anlamalarına yardımcı olmakta ve hukuk belgelerinin çevirisi veya kaydedilen duruşmaların analizi gibi birçok farklı alanda da faydalı olmaktadır.

Metin kümeleme, kullanıcıların internet, haber kaynakları ve dijital kütüphanelerde bulunan büyük miktarda metinsel belgeleri gezinmelerine, özetlemelerine ve düzenlemelerine yardımcı olan kullanışlı bir araçtır. Çeşitli kümeleme yöntemleri, farklı derecelerde başarıyla hukuk alanına uygulanmaktadır. Hukuki içeriğin benzersiz özellikleri ve hukuk alanının doğası, bir dizi zorluk sunmaktadır. Hukuki belgeler genellikle hukuki konuların geniş ve düzensiz bir kapsamına sahiptir ayrıca çok konulu olup, özenle hazırlanmış, mesleki ve alana ait özel dil içermektedir. Bununla beraber, geniş çapta erişilebilen genellikle arama ve kategorizasyon hizmetlerinin ücretsiz olduğu belgelerin aksine, hukuk alanı hala büyük ölçüde ücretli hizmetlerin sunulduğu bir alan olmaktadır.

Hukuk araştırmacılarının belirli konularla ilgili önemli bilgilere erişimini kolaylaştırmak için bazı araçlar bulunmaktadır. Bu araçlar, hukuki belgeleri belirli konulara göre sınıflandırmakta ve bu sayede araştırmacılar, ilgili konulara odaklanarak hızlı ve etkili bir şekilde bilgiye ulaşabilmektedir. Bu tür konu bazlı sınıflandırmalar, hukuk alanında araştırma yaparken, kaynaklarda gezinirken ve ilgili bilgilere erişirken araştırmacılara rehberlik etmek için tasarlanmaktadır. Bu araçlar, hukukçuların, hukuki metinleri daha verimli bir şekilde keşfetmelerini ve önemli bilgilere odaklanmalarını sağlamaktadır.

Hukuk alanında, metin kümeleme yöntemleri, benzer niteliklere sahip hukuki davaları gruplandırarak, hukuki prensipleri ve yargısal eğilimleri daha iyi anlaşılmasına yardımcı olmaktadır. Özellikle karmaşık ve geniş bir metin kümesi olduğunda, bu yöntemler önemli bir katkı sağlamaktadır.

Bu çalışmada metin kümelemenin hukuk alanındaki avantajları gözetilerek Uyuşmazlık Mahkemesi kararları üzerinde çalışılmıştır. Mahkeme kararları öncelikle TF-IDF yöntemi ile doğal dil işleme sürecinden geçirilmiştir. Sonrasında kümeleme için CURE, K-MEANS, DBSCAN, AGNES, AFFINITY ve BIRCH algoritmaları kullanılmıştır. Değerlendirme metriklerine göre en iyi sonucu BIRCH algoritmasının verdiği görülmüştür.

## 2. Literatür

Türk mahkemelerinin hukuki metinlerinin analizine yönelik çalışmalar bulunmaktadır. (Mumcuoğlu et al., 2021) Anayasa Mahkemesi kararları, (Kılıç & Öner, 2021) Yargıtay mahkemesi kararları üzerinde yapay zekâ yöntemleriyle metin analizi yapmış (Aydın, 2020) ise mobbing içeren yargı kararları üzerinde makine öğrenmesi yöntemlerini kullanmıştır. Türk yargı kararlarına yönelik kümeleme çalışması olarak Türkiye’deki ceza davaları üzerine bir istatistiksel analiz çalışması (Ay, 2018) bulunmaktadır. Ancak aynı veri seti üzerine herhangi bir çalışma yapılmamıştır.

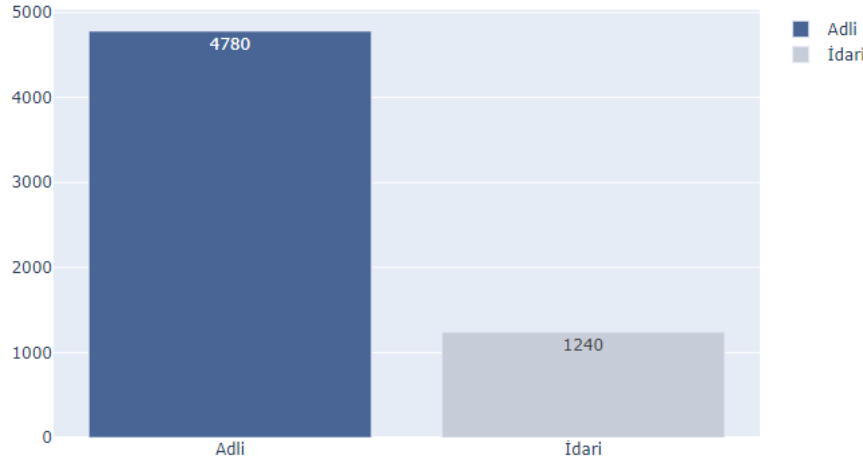
Yabancı ülkelerin mahkeme kararları ve hukuki metinleri üzerine yapılan çalışmalara bakıldığında; (Venkatesh & Raghuvver, 2013), yapmış oldukları çalışmada Hindistan mahkeme kararlarını hierarchial Latent Dirichlet Allocation (hLDA) kullanarak, konularına göre kümelendirmiş, konular ve belgeler arasındaki benzerlik ölçüsünü kullanarak her belgenin özetini bulmak için bir yaklaşım önermişlerdir. (Lu vd., 2011), yapmış oldukları çalışmada dahili konu bölümlenmesiyle bir sınıflandırma tabanlı rekürsif yumuşak kümeleme algoritması adı verilen algoritmayı tanıtmışlar ve algoritmanın, yargı kararları, kanunlar, düzenlemeler, idari materyaller ve analitik belgeleri içeren çok büyük hukuki belge veritabanlarına başarıyla uygulandığını belirtmişlerdir. (Vaughn & Boley, 2001), yapmış oldukları çalışmada Amerika Birleşik Devletleri’nin 50 eyaletindeki alkol yasalarını k-means algoritması ile kümelememişlerdir. (Merkel & Schweighofer, 1997), yapmış oldukları çalışmada, kümeleme ve yapay sinir ağları tekniklerinin kullanımıyla uluslararası anlaşma metinlerindeki bilgiyi keşfetme ve analiz etme amaçlamış ve hukuki metinlerde veri madenciliği ve analitik yöntemlerin kullanımının, uluslararası hukuk alanında bilgi keşfi ve anlayışını geliştirmede önemli bir potansiyele sahip olduğu sonucuna varmışlardır.

Türkçe metin kümeleme çalışmalarına bakıldığında; (Aslanyürek & Mesut, 2021), yapmış oldukları çalışmada kümeleme performansını ölçmek için kullanılacak değerlendirme metriklerine alternatif bir yöntem önermişlerdir. (Tunalı, 2011), yapmış olduğu çalışmada metin veri setlerinin kümelenmesi için geliştirilmiş algoritma ve yaklaşımları inceleyerek çok boyutlu ve çok büyük metin veri setlerini kümelenmesinde karşılaşılan sorunlar ve zorlukları irdelemiş ve bunlar için çözüm yöntemleri getirmiştir. (Çelik, 2020), yapmış olduğu çalışmada Shakespeare eserlerini k-means kümeleme yöntemi ile analiz etmiştir. (Ay, 2021), yapmış olduğu çalışmada benzer müşteri yorumlarını kümeleme yöntemleriyle bulmaya çalışmıştır.

### 3. Materyal ve Yöntem

#### 3.1. Veri Seti

Veri seti olarak Uyuşmazlık mahkemesi olumsuz görev uyuşmazlıkları kararları kullanılmıştır. Uyuşmazlık mahkemesinin kendi web sitesinde yayınlamış olduğu kararlar çekilmiştir. Bu kararlar olumsuz görev uyuşmazlığına dair olduğu için sonucu adli veya idari olarak iki sonucu bulunan kararlardır.



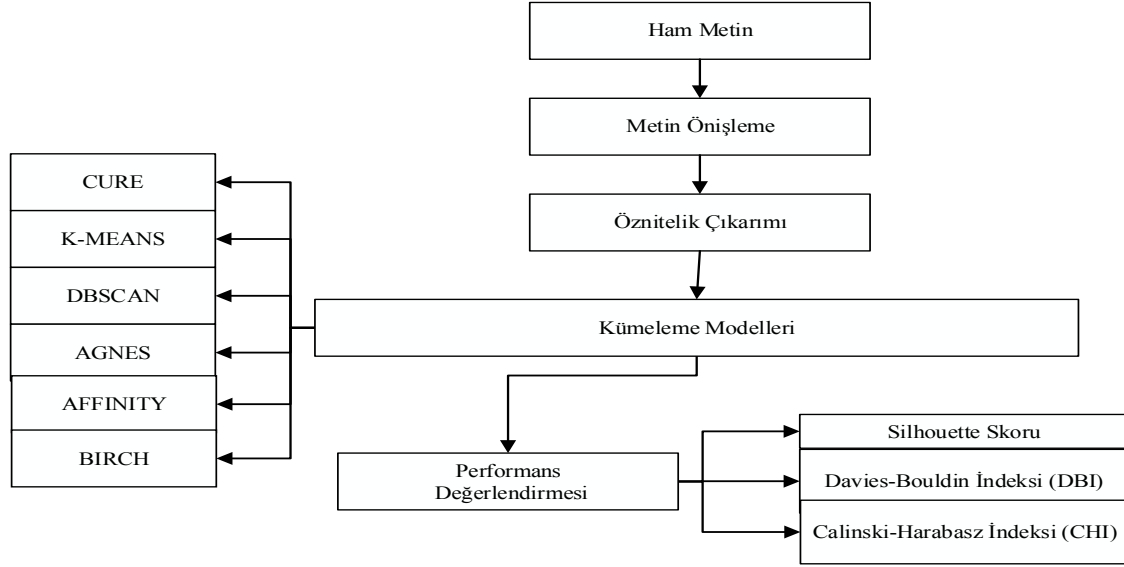
Şekil 1. Veri Setinin Dağılımı

#### 3.2. Yöntem

Çekilen bu kararlar doğal dil işleme yöntemleriyle analize hazır hale getirilmiştir. Bunun için TF-IDF yöntemi kullanılmıştır. Ardından makine öğrenmesi algoritmalarından CURE, K-MEANS, DBSCAN, AGNES, AFFINITY ve BIRCH algoritmaları kullanılarak analiz yapılmış ve sonrasında değerlendirme metrikleri kullanılarak algoritmaların sonuçları karşılaştırılmıştır.

##### 3.2.1. Cure

CURE (Clustering Using Representatives) algoritması, kümeleme (clustering) yöntemlerinden biridir. Bu algoritma, verileri önce örneklem olarak küçük alt kümeler halinde böler ve her alt kümeden bir temsilci nokta seçer. Daha sonra, bu temsilciler arasındaki uzaklıkları hesaplayarak birbirine en yakın olanları birleştirerek yeni temsilci noktalar oluşturur. Bu işlem, birbirine en yakın temsilci noktalar kalmayana kadar tekrarlanır ve sonuçta kümeleme işlemi tamamlanır (Guha vd., 2001).



Şekil 2. Akış Şeması

### 3.2.2. K-Means

K-Means, iyi bilinen kümeleme problemini çözen en basit denetimsiz öğrenme algoritmalarından biridir. Bu yöntem, önceden belirlenmiş k sayıda belirli bir veri kümesini sınıflandırmak için basit ve kolay bir yol izler. Temel fikir, her bir küme için bir merkez noktası tanımlamaktır. Bu merkez noktaları, farklı konumlar farklı sonuçlar doğurduğu için akıllıca bir şekilde yerleştirilmelidir. Bu nedenle, mümkün olduğunca birbirlerinden uzak bir şekilde yerleştirilmeleri daha iyi bir seçenektir. Bir sonraki adım, veri kümesine ait her bir noktayı en yakın merkez noktasına bağlamaktır. Hiçbir nokta kalmayana kadar bu işlem devam eder ve erken bir gruplama gerçekleştirilir. Bu noktada, bir önceki adımdan elde edilen küme merkezlerinin merkez olarak yeniden hesaplanması gerekmektedir. Yeni bu k merkez noktalarıyla aynı veri noktaları arasında ve en yakın yeni merkez noktası arasında bir bağlantı yapılması gerekmektedir. Bu döngünün sonucunda k merkez noktasının adım adım yer değiştirdiği ve daha fazla değişiklik yapılmadığı görülebilir. Yani merkez noktaları artık hareket etmez. Bu algoritma, bu durumda bir kare hatası işlevini en aza indirmeyi amaçlar (Kodinariya & Makwana, 2013).

$$W(S, C) = \sum_{k=1}^K \sum_{i \in S_k} \|y_i - c_k\|^2 \quad (1)$$

### 3.2.3. DBSCAN

DBSCAN (Density Based Spatial Clustering of Applications with Noise-Yoğunluk Tabanlı Gürültülü Uygulamaların Mekansal Kümelemesi), veri madenciliğinde kullanılan bir kümeleme algoritmasıdır. DBSCAN, veri noktalarını yoğunluk tabanlı olarak kümelere ayırır. Temel amacı, veri kümesindeki yoğun bölgeleri tanımlamak ve bu bölgeler arasındaki seyrek alanları belirlemektir. DBSCAN algoritması, kullanıcının belirtmesi gereken iki önemli parametreye dayanır: epsilon ( $\epsilon$ ) ve minPts. Epsilon, bir veri noktasının çevresindeki bir bölgedeki diğer noktaların ne kadar yakınında olması gerektiğini belirler. MinPts ise bir veri noktasının etrafındaki  $\epsilon$ -çevresinde kaç tane komşu nokta olması gerektiğini belirler (Rehman vd., 2014).

$$N_{Eps} = \{q \in D / \text{dist}(p, q)\} < Eps \quad (2)$$

Burada, D nesnelere veritabanını temsil eder. Eğer bir nokta P'nin  $\epsilon$ -komşulukları en azından bir minimum nokta sayısını içeriyorsa, bu noktaya çekirdek nokta denir. Çekirdek nokta şu şekilde tanımlanır:

$$N_{\epsilon}(P) > minPts \quad (3)$$

### 3.2.4. AGNES (Agglomerative Nesting Clustering -Agnostic Hierarchical Clustering)

AGNES, veri noktalarını hiyerarşik bir yapıya göre gruplandırır. Bu algoritma, alt kümelere ayırma yerine veri noktalarını birleştirerek gruplar oluşturur. İlk başta her veri noktası ayrı bir küme olarak kabul edilir ve ardından benzerlik ölçütlerine göre en yakın iki küme birleştirilir. Bu birleştirme işlemi, veri noktaları tüm veri seti bir küme oluşana kadar tekrarlanır. AGNES algoritması, aglomeratif bir yaklaşım kullanır, yani alt kümeleme yerine grupların birleştirilmesiyle ilerler. Benzer veri noktaları, mesafe veya benzerlik ölçütleri kullanılarak bir araya getirilir. Bu benzerlik ölçütleri, genellikle Öklidyen mesafesi, Manhattan mesafesi veya korelasyon gibi metrikler kullanılarak hesaplanır. AGNES algoritması, bir veri setindeki kümeleri hiyerarşik bir ağaç yapısı olarak temsil eden bir dendrogram üretir. Bu dendrogram, veri noktalarının nasıl gruplandığını ve birbirleriyle ilişkisini gösterir. Dendrogram, kümeleme sonuçlarını farklı düzeylerde inceleme ve farklı kümeleme seviyelerini seçme olanağı sağlar (Anders, 2003; Han vd., 2011).

### 3.2.5. AFFINITY

Affinity Clustering, veri kümesindeki örnekleri gruplara ayıran bir kümeleme algoritmasıdır. Bu yöntem, örneklerin birbirlerine olan benzerliklerine dayanarak gruplar oluşturur. Her örnek, diğer örneklerle olan benzerliklerine dayanarak bir "affinity" (bağlılık) değerine sahiptir. Affinity Clustering algoritması, veri kümesindeki her bir örneği birbirleriyle olan benzerliklerine göre bir bağlılık matrisiyle temsil eder. Bu matris, her örneğin diğer örneklerle olan benzerliklerini ifade eder. Daha sonra bu bağlılık matrisi kullanılarak gruplara ayrılma işlemi gerçekleştirilir. Affinity Clustering, veri kümesindeki örnekleri "yumuşak" kümeleme (soft clustering) yaparak gruplara ayırır. Yani, bir örnek birden fazla gruba ait olabilir ve her bir grup üyeliği için bir olasılık değeri hesaplanır (Bateni vd., 2017).

### 3.2.6. BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

BIRCH, büyük veri kümelerini hızlı ve etkili bir şekilde kümelere ayırmak için kullanılan bir hiyerarşik kümeleme algoritmasıdır. BIRCH algoritması, kümeleme işlemi sırasında bellek ve hesaplama gücü açısından verimli olmayı hedefler. Büyük veri kümesinin üzerinde döngüsel bir yaklaşım kullanarak iteratif olarak çalışır. BIRCH, veriyi özetleyen bir yapının oluşturulmasına dayanır ve bu özet yapısı üzerinde kümeleme işlemleri gerçekleştirir. BIRCH algoritması, CF (Clustering Feature) ağacı olarak adlandırılan bir veri özetleme yapısı kullanır. Bu yapı, veri noktalarını yoğunluk ve dağılım bilgileriyle birlikte özetler. Veri noktaları, CF ağacına eklenerek birleştirilir ve ağaç üzerinde yükseltme ve birleştirme işlemleri gerçekleştirilir. Bu sayede, birleştirme işlemi sırasında veri tekrar tekrar taranmadığından ve hesaplama gücü verimli bir şekilde kullanıldığından yüksek performans elde edilir. BIRCH algoritması, özellikle büyük ölçekli veri kümeleme problemleri için uygundur. Yüksek boyutlu verileri işlerken ve hızlı ve bellek açısından verimli bir çözüm gerektiğinde kullanılabilir. BIRCH, genel olarak veri madenciliği, desen tanıma ve veri analizi gibi alanlarda kullanılan bir algoritmadır (Zhang vd., 1997).

### 3.2.7. TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF (Terim Frekansı-Ters Belge Frekansı) yaklaşımı, belge kümeleme, doğal dil işleme ve bilgi erişimi alanında yaygın olarak kullanılan bir tekniktir. Bu yaklaşım, belgeleri içeriklerine ve terim benzerliklerine dayalı olarak bir araya getirilmesine olanak sağlar. Önişleme adımlarından geçirilerek sayısal forma çevrilen metinler için TF-IDF hesaplanır (Aizawa, 2003).

TF (Terim Frekansı): Bu, bir terimin belgedeki görülme sayısını belgedeki toplam terim sayısına bölünmesiyle elde edilir.

$$TF = \frac{n_{ij}}{|b_j|} \quad (3)$$

Bu formülde herhangi bir  $b_j$  belgesinde (metninde)  $t_i$  teriminin kaç defa geçtiği  $n_{ij}$  bilgisidir.  $|b_j|$  ise metin içerisindeki toplam (tekrarlı) terim sayısını göstermektedir.

IDF (Ters Belge Frekansı): Bu ölçüm, bir terimin tüm belge koleksiyonu içinde ne kadar önemli olduğunu belirtir. IDF, terimi içeren belge sayısını toplam belge sayısına bölerek hesaplanır.

$$IDF = \log \frac{n}{n_i} \quad (4)$$

Bu formülde  $n_i$  sayısı,  $t_i$  teriminin geçtiği belge sayısını;  $n$  ise kümedeki toplam belge sayısını ifade etmektedir.

TF-IDF: Her belgedeki her terimin TF değerinin IDF değeriyle çarpılmasıyla bulunur. Bu, her satırın bir belgeyi temsil ettiği ve her sütunun bir terimi temsil ettiği bir TF-IDF matrisi oluşturur. Değerler, terimin belgedeki önemini gösterir.

$$a_{ij} = (TS - BST)_{ij} = TS_{ij} \times BST_i \quad (5)$$

### 3.3. Değerlendirme Metrikleri

Kümeleme sonuçlarının performansını değerlendiren birçok metrik bulunmaktadır. Bu çalışma kapsamında özellikle metin analizi konusunda yaygın olarak kullanılan ve kümeleme için en doğru sonuçları vermede yardımcı olduğu düşünülen Silhouette Skoru (Altszyler et al., 2017; Hosseini & Varzaneh, 2022; Lovmar et al., 2005; Ogbuabor & F. N, 2018; Shahapure & Nicholas, 2020; Suchacka et al., 2021), Davies-Bouldin İndeksi (DBI) (Petrovic, 2006; Prameswari et al., 2017; Ramadhani et al., 2022; Singh et al., 2020) ve Calinski-Harabasz İndeksi (CHI) (Lukasik et al., 2016; Rashid et al., 2020; Wang & Xu, 2019; Zhao & Fränti, 2014) kullanılacaktır.

**Silhouette Skoru** kümeleme sonuçlarının kalitesini ölçmek için yaygın olarak kullanılan bir metriktir. Her bir veri noktasının kendi kümesi içindeki benzerliği ile diğer kümelere olan uzaklığına dayanır. Yüksek bir silüet skoru, iyi bir kümeleme sonucunu gösterir (Shahapure & Nicholas, 2020). Silhouette Skoru, -1 ile 1 arasında bir değer alır. 1'e yakın bir skor, kümeleme sonuçlarının oldukça iyi olduğunu ve noktaların doğru kümelere olduğunu gösterir. 0'a yakın bir skor, noktaların çakıştığını veya yanlış kümelere olduğunu gösterir. -1'e yakın bir skor, noktaların yanlış kümelere olduğunu ve kümeleme sonuçlarının oldukça kötü olduğunu gösterir. Dolayısıyla, Silhouette Skoru ne kadar yüksekse o kadar iyi bir kümeleme sonucunun elde edildiği söylenebilir. Ancak, her veri kümesi ve problemi farklıdır. Bu nedenle, Silhouette Skoru'nun kabul edilebilir bir değerini belirlemek için bir eşik veya referans değer vermek zordur. Skorlar, genellikle farklı model veya parametre ayarlarıyla elde edilen sonuçları karşılaştırmak için kullanılır. Diğer bir ifade ile skorun mutlak değeri yerine farklı model veya parametrelerle elde edilen sonuçların karşılaştırılması daha anlamlı olabilir.

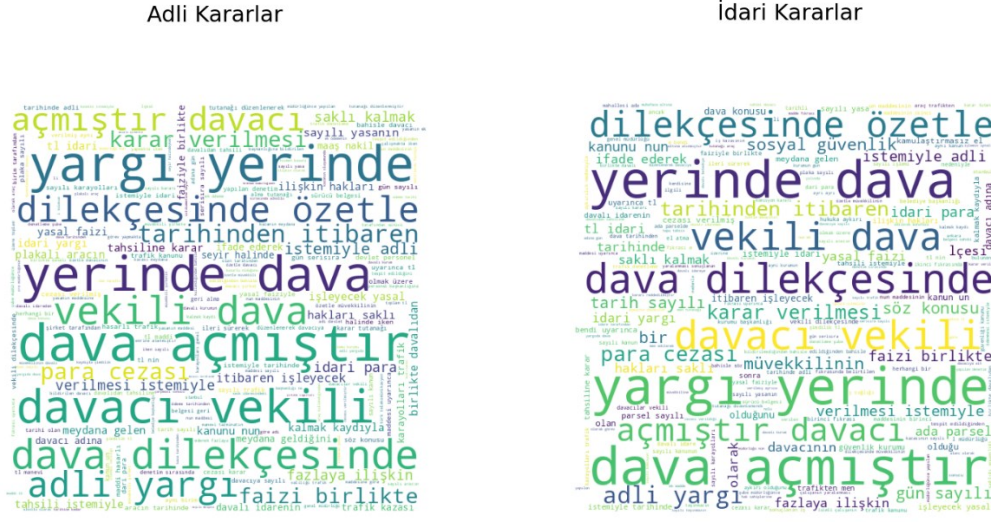
Bu skorun yanında sıklıkla kullanılan ve (Petrovic, 2006), tarafından önerilen *Davies-Bouldin İndeksi (Davies-Bouldin Index, DBI)* ile (Wang & Xu, 2019) tarafından sunulan *Calinski-Harabasz İndeksi (CHI)* endeksleri de vardır.

**Davies-Bouldin İndeksi (DBI)** kümeleme sonuçlarının kalitesini ölçen bir metrik olarak kullanılır. Bu indeksin değeri küçüldükçe, kümeleme sonuçlarının daha iyi olduğu kabul edilir. DBI, küme içi benzerliği ve kümeler arası benzerliği dikkate alır. İdeal bir DBI değeri sıfırdır ve bu durumda kümeleme sonuçları mükemmel olarak kabul edilir. Ancak, pratikte sıfıra yakın bir değer elde edilmesi zordur. Dolayısıyla, kümeleme algoritması ne kadar düşük bir DBI değeri üretirse, o kadar iyi bir sonucun elde edildiği söylenebilir.

**Calinski-Harabasz İndeksi (CHI)** kümeleme sonuçlarının kalitesini ölçmek için kullanılan diğer bir metriktir. CHI değeri, küme içi benzerlik ve kümeler arası farklılık arasındaki oranı ifade eder. Yüksek CHI değeri daha iyi bir kümeleme sonucunu gösterir. CHI değerinin kesin bir kabul edilebilir aralığı yoktur. Ancak, genellikle yüksek değerlerin daha iyi kümeleme sonuçlarını gösterdiği kabul edilir. CHI değerini yorumlamak için genellikle diğer kümeleme sonuçlarıyla veya karşılaştırmalarla birlikte kullanmak önemlidir.

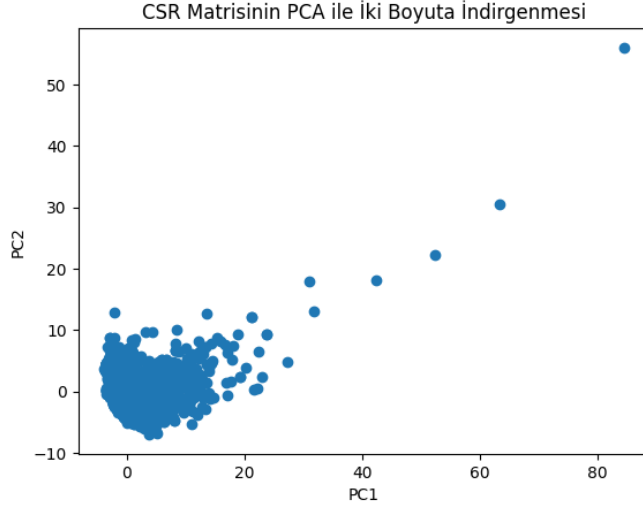
#### 4. Bulgular

Uyuşmazlık mahkemesi kararları doğal dil işleme süreçlerinden geçirildikten sonra kümeleme algoritmaları ile analiz edilmiştir. Doğal dil işleme sürecinden önce adli ve idari yargıda çokça geçen kelimeler bulunmuştur. Şekil 3'te kararlarda geçen kelimeler gösterilmiştir.



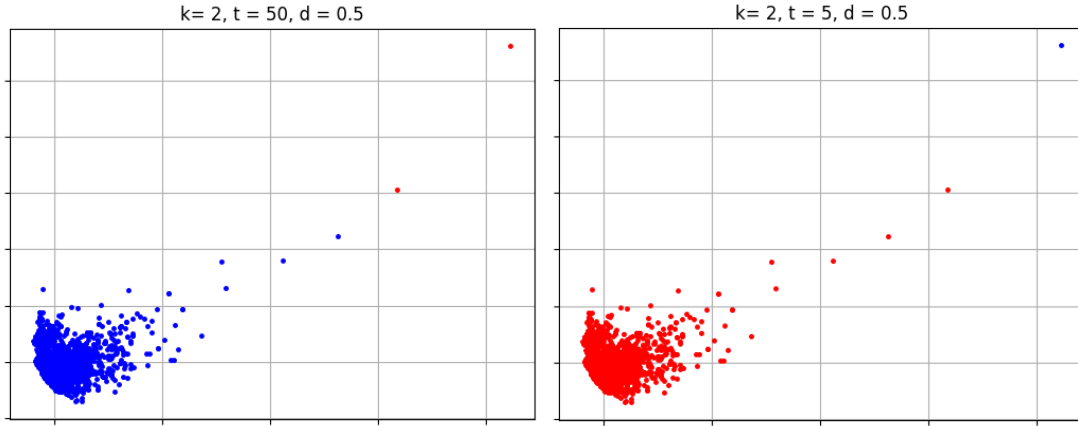
Şekil 3. Kararlarda En Çok Geçen Kelimeler

Kümeleme algoritmalarının çalışma prensiplerine uygun olarak ön işleme ve TF-IDF ile vektörel hale getirilen Uyuşmazlık Mahkemesi kararlarına temel bileşenler analizi uygulanmış ve 2 boyutlu duruma getirilmiştir. 2 boyutlu kararlar Şekil 4'te gösterilmiştir.



Şekil 4. Uyuşmazlık Mahkemesi Kararlarının 2 Boyutlu Hali

Uyuşmazlık Mahkemesinin olumsuz görev uyuşmazlıkları kararları adli ve idari olmak üzere 2 kategoriden oluştuğu için 2 boyutlu duruma getirilen bu kararlar CURE, K-MEANS, DBSCAN, AGNES, AFFINITY ve BIRCH algoritmalarıyla 2 kümeye ayrılmıştır.



**Şekil 5.** CURE Algoritması Sonuçları

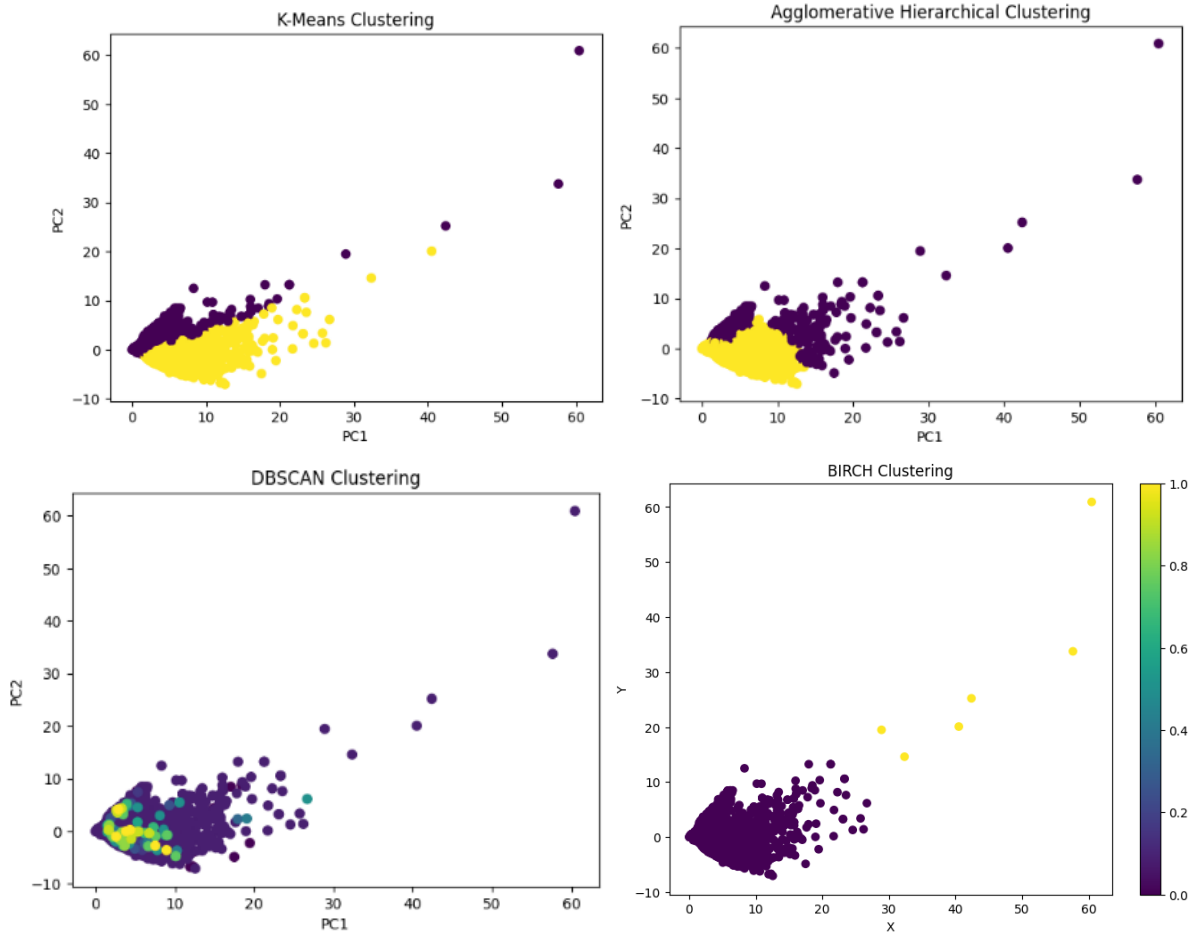
CURE algoritmasının birkaç parametresi vardır.  $k$  küme sayısı,  $t$  temsil noktası ve  $d$  daraltma katsayısını göstermektedir. Diğer algoritmaların sonuçları Şekil 6'da görselleştirilmiş olarak sunulmuştur.

Değerlendirme metrikleri ile ilgili bulgular Tablo 1'de gösterilmiştir.

**Tablo 1.** Değerlendirme Metrikleri Karşılaştırması

	<b>Silhouette Skoru</b>	<b>DBI</b>	<b>CHI</b>
<b>CURE</b> <b>k=2, t=50, d=0.5</b>	-0.072	2.136	0.701
<b>CURE</b> <b>k=2, t=5, d=0.5</b>	-0.332	15.833	0.002
<b>K-MEANS</b>	0.466	0.881	<b>3361.272</b>
<b>DBSCAN</b>	-0.352	10.513	1.635
<b>AGNES</b>	0.482	0.984	2596.884
<b>AFFINITY</b>	0.165	<b>0.312</b>	145.284
<b>BIRCH</b>	<b>0.904</b>	0.412	918.566





Şekil 6: Kümeleme Algoritmalarının Görselleştirilmesi

## 5. Sonuç

Değerlendirme metrikleri karşılaştırıldığında en iyi sonucu BIRCH algoritmasının verdiği görülmektedir. Silhouette Skoru 1'e çok yakın, DBI değeri nispeten küçük ve CHI değeri nispeten büyüktür. BIRCH algoritmasının Silhouette skoru 0.904 olarak saptanmıştır. Bu değer 1'e çok yakın bir değerdir. 1'e yakın skor, iyi bir kümeleme sonucunu gösterir ve verilerin doğru kümelerde ayrıldığını işaret eder. Ancak, her zaman diğer değerlendirme yöntemleri ve veri anlayışıyla birlikte kullanılmalıdır. Silhouette skorunun yanında BIRCH algoritmasının DBI değeri 0.412 dir. Bu değer diğer algoritmaların DBI değerleri ile karşılaştırıldığında nispeten küçüktür. DBI değeri ne kadar küçükse, kümeleme sonuçları o kadar iyi kabul edilir. Düşük bir DBI değeri, küme içi benzerliklerin yüksek ve küme arasındaki uzaklığın düşük olduğunu gösterir. Bu durum, kümelerin birbirlerine yakın ve homojen olduğunu gösterir. BIRCH algoritmasının CHI değerine bakıldığında 918 olduğu görülmektedir. Bu değer bazı algoritmaların küçük ancak genel itibarıyla diğer algoritmaların sonuçlarıyla değerlendirildiğinde büyük bir değer olarak saptanmıştır. Yüksek bir CHI değeri, kümelerin birbirlerine göre daha belirgin ve homojen olduğunu ve küme içi benzerliklerin küme arasındaki farka göre daha büyük olduğunu gösterir. Bu, verilerin başarılı bir şekilde kümelendiğini düşündürür.

DBI değeri en küçük olan AFFINITY algoritması olurken CHI değeri en büyük olan K-MEANS algoritması olmuştur. CURE algoritması için bakıldığında her ne kadar diğer algoritmalara göre kötü sonuç verdiyse de temsil noktaları artırıldığında daha iyi sonuçlara ulaşıldığı görülmektedir. Bir kümeleme algoritmasının performansı için Silhouette Skoru, DBI ve CHI değerleri beraber değerlendirilmelidir. Bu üç ölçü bir arada değerlendirildiğinde BIRCH algoritmasının diğer algoritmalara göre daha iyi sonuç verdiği görülmektedir. BIRCH algoritması hukuki metinlerle etkili bir sinerji oluşturabilir, çünkü büyük veri kümeleri ve karmaşık belgelerle başa çıkabilme yeteneği vardır. Bu algoritma, hukuki metinlerin kümeleme, temsil ve analizi için potansiyel olarak kullanışlı bir araç olabilir.

Hukuki metinlerin kümeleme yöntemlerinin başarısı, verinin kalitesine, ilgili özelliklerin seçimine ve kümeleme algoritmasının etkinliğine de bağlıdır. Bununla beraber kümeleme sürecinden elde edilen sonuçların yorumlanması ve doğrulanması için insan uzmanlığı ve alan bilgisi önemlidir. Kümeleme yöntemleri, hukuki kararları düzenleme yoluyla hukuki uygulamaların kısa bir değerlendirmesini oluşturmak için güçlü bir araç görevi

görür. Bu yöntemler, benzer davalara kategorize ederek, hukuk aktörlerinin önemli bilgiler elde etmelerini, hukuki eğilimleri belirlemelerini ve yargı sisteminin kavrayışını artırmalarını sağlar.

Özellikle hukuk alanında karmaşık ve geniş bir metin kümesi olduğunda, kümelenme yöntemleri değerli bir katkı sunar. Bu yöntemler, belirli bir konuda benzer niteliklere sahip davaları gruplandırarak, hukuki prensipleri ve yargısal eğilimleri daha iyi anlaşılmasına yardımcı olur. Bu şekilde, benzer hukuki konuları ele alan davalar bir araya getirilerek, benzer kararlar ve önemli içtihatlar belirlenebilir.

Kümelenme yöntemlerinin avantajları arasında, hukuki uygulamanın özeti oluşturulurken zaman ve çaba tasarrufu sağlaması, hukuki araştırmacıların geniş bir dava yelpazesine hızlı bir şekilde erişmelerini sağlaması ve hukuki analiz sürecini iyileştirmesi sayılabilir. Ayrıca, kümelenme sonuçları, hukuki stratejilerin geliştirilmesi, dava öncesi hazırlık ve hukuki kararların temellendirilmesi gibi birçok farklı alanda kullanılabilir.

Ancak, kümelenme yöntemlerinin sınırlamaları da vardır. Veri kalitesi, doğru özelliklerin seçilmesi, uygun kümelenme algoritmasının kullanılması ve insan uzmanlığının katkısı gibi faktörler başarılı bir kümelenme analizi için önemlidir. Ayrıca, hukuki metinlerin karmaşıklığı ve özgünlüğü nedeniyle, kümelenme sonuçları doğru bir şekilde yorumlanmalı ve doğrulanmalıdır.

Sonuç olarak, kümelenme yöntemleri hukuki uygulamanın kısa bir değerlendirmesini oluşturmak için etkili bir araç sağlar. Bu yöntemler, benzer davaların gruplandırılması yoluyla hukuki prensiplerin, içtihatların ve eğilimlerin belirlenmesini sağlar. Hukuki uygulama üzerindeki geniş kapsamlı analiz ve anlayış sağlama potansiyeli, hukuki araştırmalar ve profesyonel uygulamalar için büyük bir avantaj sağlar.

### Kaynaklar

- Aizawa, A. (2003). An information-theoretic perspective of tf-idf measures. *Information Processing & Management*, 39(1), 45–65. [https://doi.org/10.1016/S0306-4573\(02\)00021-3](https://doi.org/10.1016/S0306-4573(02)00021-3)
- Altszyler, E., Ribeiro, S., Sigman, M., & Fernández Slezak, D. (2017). The interpretation of dream meaning: Resolving ambiguity using Latent Semantic Analysis in a small corpus of text. *Consciousness and Cognition*, 56, 178–187. <https://doi.org/10.1016/j.concog.2017.09.004>
- Anders, K.-H. (2003). Anders, K. H. (2003, April). A hierarchical graph-clustering approach to find groups of objects. *5th Workshop on Progress in Automated Map Generalization*, 1–8.
- Ay, S. (2018). Türkiye’deki Ceza Davalarının İstatistiksel Analizi. *İstanbul Ticaret Üniversitesi Sosyal Bilimler Dergisi*, 17(33), 25–36.
- Aydın, Ö. (2020). *Mobbing İçerikli Yargı Kararlarının Makine Öğrenmesi Algoritmaları ile Sınıflandırılması* [Yayımlanmış Yüksek Lisans Tezi]. Balıkesir Üniversitesi Fen Bilimleri Enstitüsü.
- Bateni, M., Behnezhad, S., Derakhshan, M., Hajiaghayi, M., Kiveris, R., Lattanzi, S., & Mirrokni, V. (2017). Affinity clustering: Hierarchical clustering at scale. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 1–11).
- Guha, S., Rastogi, R., & Shim, K. (2001). Cure: an efficient clustering algorithm for large databases. *Information Systems*, 26(1), 35–58. [https://doi.org/10.1016/S0306-4379\(01\)00008-4](https://doi.org/10.1016/S0306-4379(01)00008-4)
- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining. Concepts and Techniques, 3rd Edition (The Morgan Kaufmann Series in Data Management Systems)*.
- Hosseini, S., & Varzaneh, Z. A. (2022). Deep text clustering using stacked AutoEncoder. *Multimedia Tools and Applications*, 81(8), 10861–10881. <https://doi.org/10.1007/s11042-022-12155-0>
- Kılıç, B., & Öner, Y. (2021). *Yargıtay Kararlarının Suç Türlerine Göre Makine Öğrenmesi Yöntemleri İle Sınıflandırılması* (Vol. 4, Issue 3). <https://dergipark.org.tr/en/download/article-file/2032425>
- Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of Cluster in K-Means Clustering. *International Journal of Advance Research in Computer Science and Management Studies*, 1(6), 90–95.
- Lovmar, L., Ahlford, A., Jonsson, M., & Syvänen, A.-C. (2005). Silhouette scores for assessment of SNP genotype clusters. *BMC Genomics*, 6(1), 35. <https://doi.org/10.1186/1471-2164-6-35>
- Lukasik, S., Kowalski, P. A., Charytanowicz, M., & Kulczycki, P. (2016). Clustering using flower pollination algorithm and Calinski-Harabasz index. *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2724–2728. <https://doi.org/10.1109/CEC.2016.7744132>
- Mumcuoğlu, E., Öztürk, C. E., Ozaktas, H. M., & Koç, A. (2021). Natural language processing in law: Prediction of outcomes in the higher courts of Turkey. *Information Processing and Management*, 58(5). <https://doi.org/10.1016/j.ipm.2021.102684>
- Ogbuabor, G., & F. N, U. (2018). Clustering Algorithm for a Healthcare Dataset Using Silhouette Score Value. *International Journal of Computer Science and Information Technology*, 10(2), 27–37. <https://doi.org/10.5121/ijcsit.2018.10203>
- Petrovic, S. (2006). A Comparison Between The Silhouette Index And The Davies-Bouldin Index in Labelling Ids Clusters. *11th Nordic Workshop of Secure IT Systems*, 53–64.
- Prameswari, P., Zulkarnain, Surjandari, I., & Laoh, E. (2017). Mining online reviews in Indonesia’s priority tourist destinations using sentiment analysis and text summarization approach. *2017 IEEE 8th International*

- Conference on Awareness Science and Technology (ICAST)*, 121–126. <https://doi.org/10.1109/ICAwST.2017.8256429>
- Ramadhani, S., Azzahra, D., & Z, T. (2022). Comparison of K-Means and K-Medoids Algorithms in Text Mining based on Davies Bouldin Index Testing for Classification of Student's Thesis. *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, 13(1), 24–33. <https://doi.org/10.31849/digitalzone.v13i1.9292>
- Rashid, J., Shah, S. M. A., & Irtaza, A. (2020). An Efficient Topic Modeling Approach for Text Mining and Information Retrieval through K-means Clustering. *Mehran University Research Journal of Engineering and Technology*, 39(1), 213–222. <https://doi.org/10.22581/muet1982.2001.20>
- Rehman, S. U., Asghar, S., Fong, S., & Sarasvady, S. (2014). DBSCAN: Past, present and future. *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, 232–238. <https://doi.org/10.1109/ICADIWT.2014.6814687>
- Shahapure, K. R., & Nicholas, C. (2020). Cluster Quality Analysis Using Silhouette Score. *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, 747–748. <https://doi.org/10.1109/DSAA49011.2020.00096>
- Singh, A. K., Mittal, S., Malhotra, P., & Srivastava, Y. V. (2020). Clustering Evaluation by Davies-Bouldin Index (DBI) in Cereal data using K-Means. *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 306–310. <https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00057>
- Suchacka, G., Cabri, A., Rovetta, S., & Masulli, F. (2021). Efficient on-the-fly Web bot detection. *Knowledge-Based Systems*, 223, 107074. <https://doi.org/10.1016/j.knosys.2021.107074>
- Wang, X., & Xu, Y. (2019). An improved index for clustering validation based on Silhouette index and Calinski-Harabasz index. *IOP Conference Series: Materials Science and Engineering*, 569(5), 052024. <https://doi.org/10.1088/1757-899X/569/5/052024>
- Zhang, T., Ramakrishnan, R., & Livny, M. (1997). BIRCH: A New Data Clustering Algorithm and Its Applications. *Data Mining and Knowledge Discovery*, 1(2), 141–182. <https://doi.org/10.1023/A:1009783824328>
- Zhao, Q., & Fränti, P. (2014). WB-index: A sum-of-squares based index for cluster validity. *Data & Knowledge Engineering*, 92, 77–89. <https://doi.org/10.1016/j.datak.2014.07.008>