

VERİ MADENCİLİĞİNDE NESNE YÖNELİMLİ BİRLEŞTİRİCİ HİYERARŞİK KÜMELEME MODELİ

Mehmet YEŞİLBUDAK*, **Hamdi Tolga KAHRAMAN**** ve **Hacer KARACAN*****

*Nevşehir Üniversitesi, Hacı Bektaş Veli Meslek Yüksek Okulu, Elektronik ve Otomasyon Bölümü, Nevşehir

**Karadeniz Teknik Üniversitesi, Of Teknoloji Fakültesi, Bilgisayar Eğitimi Bölümü, Of, Trabzon

***Gazi Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Maltepe, Ankara

myesilbudak@nevsehir.edu.tr, hamditolga@gazi.edu.tr, hkaracan@gazi.edu.tr

(Geliş/Received: 19.02.2010; Kabul/Accepted: 14.12.2010)

ÖZET

Veri madenciliği alanında farklı kullanım amacına sahip çok sayıda teknik mevcuttur. Son yıllarda sıklıkla kullanılan tekniklerden biri de birleştirici hiyerarşik kümeleme analizidir. Bu çalışmada nesne yönelimli birleştirici hiyerarşik kümeleme modeli geliştirilmiştir. Geliştirilen modelin interaktifliği sayesinde kullanıcıların sistem arayüzünü rahatlıkla kullanabilmesi ve nesne yönelimli programlama sayesinde de kullanıcı taleplerinin hızlı bir şekilde karşılanması sağlanmıştır. Geliştirilen model bu sayede optimum performansa sahip olmuştur. Modelin test aşamasında, Türkiye'deki 44 ile ait aylık ortalama rüzgâr hızı verileri hiyerarşik olarak kümelendirilmiştir. Kümeleme sonuçları Microsoft Visual Studio.NET platformunda oluşturulan arayüzde metinsel formatta sunulurken; Matlab'da ise dendrogramlar aracılığıyla görselleştirilmiştir. Kümeleme sonuçları sayesinde, önceden gizli olan fakat model sayesinde ortaya çıkarılan pek çok çıkarım elde edilmiş ve model analizi etkin bir şekilde yapılmıştır.

Anahtar Kelimeler: Veri madenciliği, nesne yönelimli programlama, kümeleme analizi.

OBJECT ORIENTED AGGLOMERATIVE HIERARCHICAL CLUSTERING MODEL IN DATA MINING

ABSTRACT

Many techniques that have different intended uses are available in data mining. One of the techniques used frequently in recent years is the agglomerative hierarchical clustering analysis. In this study, an object oriented agglomerative hierarchical clustering model has been developed. Users have used the system interface easily through the interactivity of the model developed and the user requests have been accomplished quickly through the object oriented programming. Thus, the model developed has had optimum performance. In the test process of the model, monthly average wind speed data belonging to 44 cities in Turkey are clustered hierarchically. The clustering results have not only been presented in textual format in the interface created using Microsoft Visual Studio.NET platform but also have been visualized using dendrograms in Matlab. Many inferences which remained hidden previously but uncovered by the model are achieved and the model analysis has been done efficiently.

Key Words: Data mining, object oriented programming, clustering analysis.

1. GİRİŞ (INTRODUCTION)

Teknoloji devrimi ile birlikte verilerin dijital ortamlarda saklanmaya başlanması nedeniyle, veri depolama ünitelerinin hacimlerinde ve veri tabanı sistemlerinin kullanımında olağanüstü artış meydana gelmiştir. Bu durum geleneksel sorgulama ve raporlama araçlarının dev veri yığınları karşısında

etkisiz kalmasına yol açmıştır [1]. Bunun sonucunda veri tabanlarında özbilgi keşfi adı altında yeni arayışlar ortaya çıkmıştır. Veri tabanlarında özbilgi keşif sürecinin en önemli adımı ise gizli ve yararlı örüntülerin ortaya çıkarıldığı veri madenciliği aşamasıdır [2]. Veri madenciliği alanında farklı kullanım amacına sahip çok sayıda teknik mevcuttur. Bu tekniklerden biri olan birleştirici hiyerarşik

kümeleme analizi ise ayrı ayrı ele alınan kümelerin aşamalı olarak birleştirilmesi esasına dayanmaktadır.

Günümüzde, birçok farklı problem alanı için hiyerarşik kümeleme algoritması yazılım araçları geliştirilmiştir. Bu araçlar kullanılarak, durağan olarak hiçbir anlam taşımayan fakat içerisinde önemli verileri ve bilgileri saklayan büyük veri havuzlarından anlamlı bilgiler elde edilmeye çalışılmıştır. Bu çalışmalara örnek olarak, müzik aletlerinin ses sinyallerine göre hiyerarşik olarak kümelmesiyle birçok müzik aletinin gerçek ailesinin tespit edilmesi[3], videoların özetini temsil eden fotoğrafların hiyerarşik olarak gruplandırılması[4], dokümanların otomatik olarak kümelmesi ve yönetimi[5], şirketlerin finansal olarak kümelmesi[6], havanın kalitesini etkileyen faktörleri ortaya çıkarmak amacıyla saatte bir alınan rüzgâr verilerinin hiyerarşik olarak kümelmesi[7], Türkiye'deki illerin sosyo-ekonomik benzerliklerine göre hiyerarşik olarak kümelmesi[8], yine sosyoekonomik kriterler bakımından Türkiye, Doğu Avrupa ve Kafkas ülkelerinin karşılaştırılması[9] ve Türkiye'nin de aralarında bulunduğu 30 ülkenin inovasyon göstergelerine göre hiyerarşik olarak kümelmesi [10] gösterilebilir.

Günümüzdeki hiyerarşik kümeleme yazılımlarının kısıtlarından biri uygulama-bağımlı (problem-odaklı) olup, sadece geliştirildikleri alandaki probleme cevap üretmeleridir. Uygulamaların sınırlılıklarından bir diğeri de gözlem setinin veri yapısına bağlı olarak çalışmaları ve bu durumun sonucu olarak gözlem setindeki küçük değişikliklerin bile yazılımda büyük çaplı düzenlemeler yapılmasını ve uyumluluk sorununun çözülmesini gerektirmesidir. Sorunların özeti ise; mevcut hiyerarşik kümeleme yazılım araçlarının uygulandıkları problemde ve algoritmada ortaya çıkabilecek yeniliklere ve değişikliklere karşı uyum sağlayamamalarıdır. Yazılım mühendisliğinde bu sorunlar nesne yönelimli programlama teknikleri uygulanarak çözülebilir. Nesne yönelimli programlama tekniği özetle, problemi tanımlayan doğal sınıfların tanımlanmasını ve bu sınıflar arasındaki etkileşimlerin katmanlı bir mimari yapı ile modellenmesini gerektirir.

Bu makale çalışmasında öncelikle mevcut hiyerarşik kümeleme yazılım araçlarının kısıtlarını ortadan kaldırmak amacıyla nesne yönelimli programlama teknikleri kullanılarak hiyerarşik kümeleme modelinin

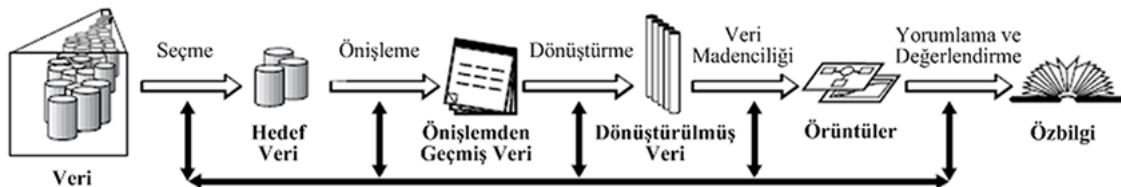
doğal sınıfları belirlenmiş ve geliştirilmiştir. Geliştirilen modelin uygulandığı yazılım aracının tasarımı, Microsoft Visual Studio.NET platformu ve bu platformun %100 nesne yönelimli kodlama yeteneğini sunan C# programlama dili kullanılarak gerçekleştirilmiştir. Geliştirilen model, interaktif bir arayüze sahip olduğu için kullanımı kolay bir yapıya; nesne yönelimli programlama tekniklerini taşıdığı için de optimum performansa ve esnekliğe sahiptir. Geliştirilen modelin test aşamasında ise, Türkiye'deki 44 il, "aylık ortalama rüzgâr hızları" açısından hiyerarşik olarak kümelmiş ve faydalı çıkarımlar elde edilmiştir. Bu makale çalışmasıyla birlikte literatüre farklı uygulama alanlarında uygulanabilecek bir hiyerarşik kümeleme modeli ve bu modeli etkili ve anlaşılabilir bir şekilde uygulayan yazılım aracı sunulmaktadır.

2. VERİ MADENCİLİĞİ (DATA MINING)

Veri madenciliği; önceden bilinmeyen, gizli, anlamlı ve yararlı örüntülerin büyük ölçekli veri tabanlarından otomatik biçimde elde edilmesini sağlayan veri tabanlarındaki özbilgi keşif ve analiz süreci içerisinde yer alan bir adımdır [11]. Veri tabanlarındaki özbilgi keşif ve analiz sürecinde yer alan adımlar Şekil 1'de görülmektedir [12].

Veri seçimi aşaması birkaç veri kümesini birleştirerek sorguya uygun örneklem kümesini elde etmeyi sağlar. Önişleme, seçilen örnekleme yer alan hatalı verilerin çıkarıldığı; eksik verilerin tamamlandığı aşamadır. Dönüştürme adımı ise verinin, veri madenciliği çözümlerine uygun forma getirilmesini sağlar. Veri çözümlenmeye hazır hale getirildikten sonra ilgili veri madenciliği tekniği ve algoritması uygulanarak örüntüler elde edilir. Yorumlama ve değerlendirme aşamasında ise, keşfedilen örüntülerin geçerlilik, yenilik ve yararlılık kriterlerine göre değerlendirmesi yapılır. Veri tabanlarındaki özbilgi keşif ve analiz süreci defalarca tekrar, aşamalar arası atlamalar ve ileri geri hareketler içerebilmektedir [11, 13].

Veri madenciliği; makine öğrenimi, örüntü tanıma, istatistik, veritabanı yönetim sistemleri, zeki sistemler, veri görselleştirme alanlarının kesişimi olarak doğmuş ve bu bağlamda gelişimini sürdürmektedir [14]. Pazarlama, banka ve sigortacılık, biyoloji, tıp ve genetik, kimya, görüntü tanıma ve robot görüş sistemleri, uzay bilimleri, yüzey analizi ve coğrafi bilgi sistemleri, meteoroloji, sosyal bilimler ve



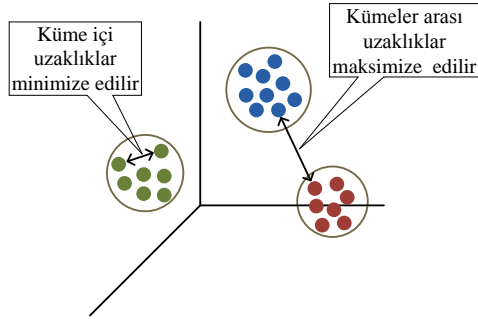
Şekil 1. Veri tabanlarında özbilgi keşif süreci (The process of knowledge discovery in databases)

davranış bilimleri, metin madenciliği, internet madenciliği vb. birçok alanda yaygın bir şekilde kullanılmaktadır [15].

Veri madenciliği alanında çok sayıda yöntem ve algoritma geliştirilmiştir. Veri madenciliği teknikleri genel olarak tanımlama ve ayırlama, birliktelik analizi, sınıflandırma ve tahmin etme, kümeleme analizi, sıradışılık analizi ve evrimsel analiz olarak gruplandırılmaktadır [13].

3. KÜMELEME ANALİZİ (CLUSTERING ANALYSIS)

Kümeleme analizi, birbirlerine benzeyen veri parçalarını kümelere ayırma işlemidir ve iki temel prensip içerir. Bunlar küme içi benzerliği maksimum; kümeler arası benzerliği ise minimum yapma prensipleridir (Bkz. Şekil 2) [16, 17].



Şekil 2. Kümeleme analizindeki temel prensipler
(The basic principles in clustering analysis)

Kümeleme yöntemlerinin birçoğu gözlem değerleri arasındaki uzaklıkların hesaplanması esasına dayanmaktadır. Bu nedenle iki nokta arasındaki uzaklığı hesaplayan bağıntılara gereksinim vardır. Uygulamada en çok kullanılan uzaklık bağıntısı Öklid uzaklık bağıntısıdır. p değişken sayısı olmak üzere, $i, j=1, 2, \dots, n$ ve $k=1, 2, \dots, p$ için Öklid uzaklığı şu şekilde olacaktır [18]:

$$d(i, j) = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2} \quad (1)$$

Diğer bir uzaklık bağıntısı ise Manhattan uzaklığıdır. Bu uzaklık, gözlemler arasındaki mutlak uzaklıkların toplamı alınarak hesaplanır. Söz konusu uzaklık şu şekilde ifade edilir [19]:

$$d(i, j) = \sum_{k=1}^p (|x_{ik} - x_{jk}|) \quad (2)$$

Gözlemler arasındaki uzaklıkların hesaplanmasında Minkowski uzaklık bağıntısı da kullanılabilir. Minkowski uzaklık bağıntısında $m = 2$ alınırsa Öklid uzaklığı; $m = 1$ alınırsa Manhattan uzaklığı elde edilir [20].

$$d(i, j) = \left[\sum_{k=1}^p (|x_{ik} - x_{jk}|^m) \right]^{\frac{1}{m}} \quad (3)$$

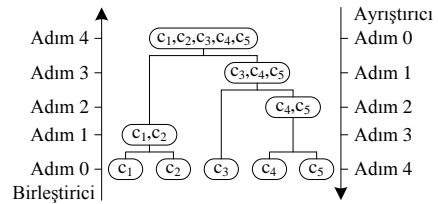
4. KÜMELEME YÖNTEMLERİ (CLUSTERING METHODS)

Veri madenciliği alanında çok sayıda kümeleme yöntemi bulunmaktadır. Kümeleme yöntemleri genel olarak hiyerarşik yöntemler, parçalı yöntemler, yoğunluk tabanlı yöntemler, şebeke tabanlı yöntemler ve sezgisel yöntemler olarak gruplandırılmaktadır [20].

4.1. Hiyerarşik Kümeleme Yöntemleri (Hierarchical Clustering Methods)

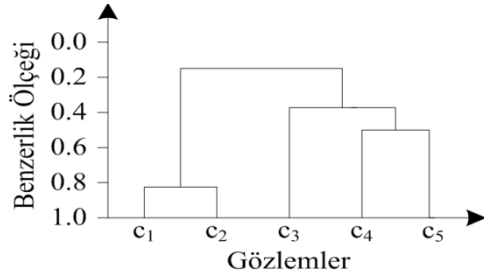
Hiyerarşik kümeleme yöntemleri, kümelerin bir ana küme olarak ele alınması ve sonra aşamalı olarak içerdiği alt kümelere ayrılması veya ayrı ayrı ele alınan kümelerin aşamalı olarak bir küme biçiminde birleştirilmesi esasına dayanır [16]. Hiyerarşik kümeleme yöntemleri temel olarak birleştirici hiyerarşik kümeleme yöntemleri ve ayrıştırıcı hiyerarşik kümeleme yöntemleri olmak üzere başlıca iki türe sahiptir.

Birleştirici hiyerarşik kümeleme yöntemlerinde başlangıçta her bir gözlem, bağımsız bir küme olarak değerlendirilir ve daha sonra tekrarlı bir biçimde, bütün gözlemleri içeren tek bir küme elde edilene kadar, her bir gözlem veya gözlem kümesinin, kendisine en yakın olan gözlem ya da gözlem kümesi ile bir küme oluşturması sağlanır. Ayrıştırıcı hiyerarşik kümeleme yöntemlerinde ise, başlangıçta bütün gözlemler tek bir küme olarak değerlendirilir ve daha sonra tekrarlı bir biçimde, bütün gözlemler birbirlerinden bağımsız tek bir küme oluncaya kadar, her bir gözlem ya da gözlem kümesi, kendisine en yakın olan gözlem ya da gözlem kümesinden ayrılıp, yeni bir küme oluşturacak şekilde ayrıştırılır. Birleştirici ve ayrıştırıcı hiyerarşik kümeleme yöntemlerinin aşamalı olarak gösterimi Şekil 3'de yer almaktadır [13].



Şekil 3. Birleştirici ve ayrıştırıcı hiyerarşik kümeleme yöntemleri (Agglomerative and divisive hierarchical clustering methods)

Hiyerarşik kümeleme yöntemlerinin çıktıları dendrogramlar ile sunulmaktadır. Örneğin Şekil 3'de yer alan birleştirici hiyerarşik kümeleme yönteminin "dendrogram" ile gösterimi Şekil 4'de görülmektedir [13].



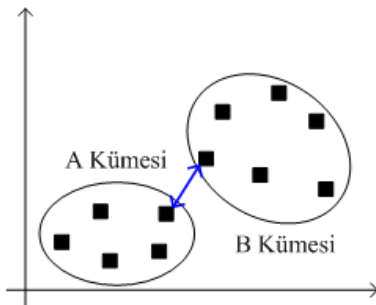
Şekil 4. Birleştirici hiyerarşik kümeleme yönteminin dendrogram ile gösterimi (Dendrogram representation of the agglomerative hierarchical clustering method)

Birleştirici ve ayrıştırıcı hiyerarşik kümeleme yöntemleri en yakın komşu algoritmasını (tek bağlantı yöntemini) ve en uzak komşu algoritmasını (tam bağlantı yöntemini) kullanmaktadır [21]. Bu çalışmada nesne yönelimli birleştirici hiyerarşik kümeleme modeli geliştirildiği için en yakın ve en uzak komşu algoritmaları kullanılmaktadır.

4.1.1. En yakın komşu algoritması (The nearest neighbor algorithm)

En yakın komşu algoritmasına “*tek bağlantı kümeleme yöntemi*” adı da verilmektedir. En yakın komşu algoritmasında öncelikle gözlemler arasındaki uzaklıklar belirlenir. i ve j gözlemleri arasındaki uzaklıkların belirlenmesinde herhangi bir uzaklık bağıntısı kullanılabilir. Hesaplanan uzaklıklar göz önüne alınarak $Min[d(i, j)]$ değeri tespit edilir. Söz konusu uzaklıkla ilgili satırlar birleştirilerek yeni bir küme elde edilir. Bu duruma göre uzaklıklar yeniden hesaplanır [16].

Tek bir gözlemden oluşan kümeler arasındaki uzaklıklar herhangi bir uzaklık bağıntısı ile hesaplanırken; birden fazla gözlem değerine sahip olan iki küme arasındaki uzaklığın hesaplanmasında ise farklı bir yol izlenir. İki kümenin içerdiği gözlemler arasında “birbirine en yakın olanların uzaklığı” iki kümenin birbirine uzaklığı olarak kabul edilir [16]. Bu durum Şekil 5’de görülmektedir.

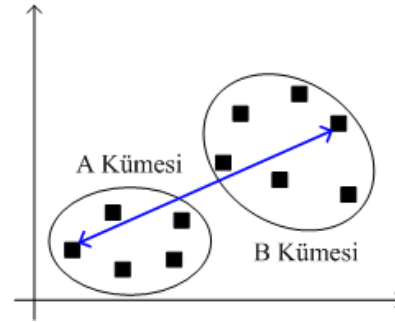


Şekil 5. İki kümenin birbirine en yakın gözlemleri arasındaki uzaklık (The distance between the nearest observations each other of two clusters)

4.1.2. En uzak komşu algoritması (The farthest neighbor algorithm)

En uzak komşu algoritmasına “*tam bağlantı kümeleme yöntemi*” adı da verilmektedir. En uzak komşu algoritmasında öncelikle gözlemler arasındaki uzaklıklar belirlenir. i ve j gözlemleri arasındaki uzaklıkların belirlenmesinde herhangi bir uzaklık bağıntısı kullanılabilir. Hesaplanan uzaklıklar göz önüne alınarak $Max[d(i, j)]$ değeri tespit edilir. Söz konusu uzaklıkla ilgili satırlar birleştirilerek yeni bir küme elde edilir. Bu duruma göre uzaklıklar yeniden hesaplanır [16].

Tek bir gözlemden oluşan kümeler arasındaki uzaklıklar herhangi bir uzaklık bağıntısı ile hesaplanırken; birden fazla gözlem değerine sahip olan iki küme arasındaki uzaklığın hesaplanmasında ise farklı bir yol izlenir. İki kümenin içerdiği gözlemler arasında “birbirine en uzak olanların uzaklığı” iki kümenin birbirine uzaklığı olarak kabul edilir [16]. Bu durum Şekil 6’da görülmektedir.



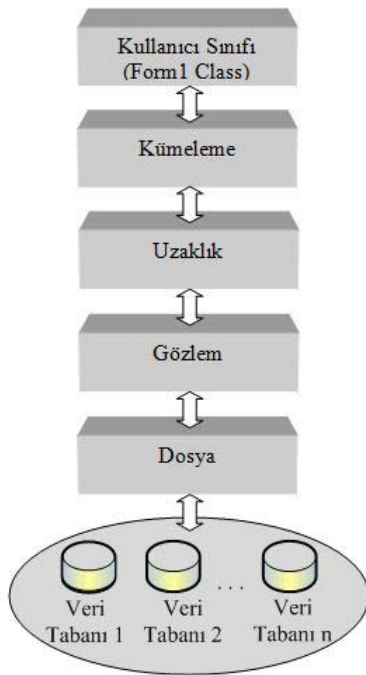
Şekil 6. İki kümenin birbirine en uzak gözlemleri arasındaki uzaklık (The distance between the farthest observations each other of two clusters)

5. NESNE YÖNELİMLİ BİRLEŞTİRİCİ HİYERARŞİK KÜMELEME MODELİ (OBJECT ORIENTED AGGLOMERATIVE HIERARCHICAL CLUSTERING MODEL)

5.1. Modelin Kavramsal Altyapısı (Conceptual Substructure of the Model)

Hiyerarşik kümelemenin temel öğeleri gözlem seti, hiyerarşik kümeleme yöntemleri ve uzaklık bağıntılarıdır [22]. Herhangi bir probleme hiyerarşik kümeleme yönteminin uygulanabilmesi için gözlem setinin hazırlanması ve hiyerarşik kümeleme algoritması yazılım aracının geliştirilmesi gerekmektedir. Hiyerarşik kümeleme yazılım aracı, nesne yönelimli programlama teknikleri kullanılarak geliştirildiğinde algoritmanın doğal sınıfları olan gözlem, kümeleme ve uzaklık sınıfları tasarlanmalıdır. Hiyerarşik kümeleme sınıflarının dışında veri tabanı ve dosyaları okuma-yazma görevlerini yerine getirecek bir dosya işlemleri sınıfı da tasarlanmalıdır.

Geliştirilen hiyerarşik kümeleme modelinde her katman bir sınıf ile temsil edilmekte ve katmanlar arası iletişim dinamik veri yapısına sahip nesnelere ve metodlar ile sağlanmaktadır. Bu sayede algoritmaya eklenebilecek yeni kümeleme yöntemlerinin ve uzaklık bağıntılarının da uygulamaya uyumlu olması sağlanmaktadır. Şekil 7’de geliştirilen modelin çok katmanlı mimari yapısı verilmektedir. Gözlem sınıfı, problem alanından bağımsız olarak gözlem nesnelere temsil edebilecek dinamik bir veri yapısına sahiptir. Bu sayede veri tabanı katmanından okunan herhangi bir gözlem nesnesinin sahip olduğu özellik sayısındaki farklılıklar ve bu özelliklerin aldıkları alt ve üst sınır değerlerindeki değişiklikler gözlem sınıfından yaratılan bir nesne ile esnek bir şekilde temsil edilebilmektedir.



Şekil 7. Geliştirilen modelin çok katmanlı mimari yapısı (Multilayer architectural structure of the model developed)

Tablo 1’de, farklı uygulama alanlarından gözlem nesnelere temsil edilebilen genelleştirilmiş gözlem tablosu verilmektedir. Gözlem tablosunun dinamik özellikleri “gözlem sayısı”, “özellik sayısı” ve özelliklerin alabileceği “alt ve üst sınır değerleri” dir. Genelleştirilmiş gözlem tablosunu temsil edecek bir gözlem sınıfı tasarlandığında veri yapısı <gözlem-no><ad-1, ad-2, ..., ad-n><özellik-1,özellik

2,...,özellik-m>, özelliklerin alabileceği azami ve asgari değerler <özellik-1 max, özellik-1 min> <özellik-2 max, özellik-2 min> <özellik-m max, özellik-m min> şeklinde düzenlenmektedir. Bu düzenleme gözlem sınıfındaki ilgili özelliklerin dinamik ve düzensiz dizilerle tanımlanmasıyla sağlanmaktadır. Herhangi bir veri tabanında genelleştirilmiş gözlem tablosu biçiminde tutulan gözlem seti Q ile ve bu gözlem setindeki bir alt gözlem kümesi ise T ile temsil edilsin. $T \subset Q$ olmak üzere, Q setindeki gözlem nesnelere $\langle q_1, q_2, \dots, q_k \rangle$ ile ve T setindeki gözlem nesnelere ise $\langle t_1, t_2, \dots, t_c \rangle$ ile temsil edilsin. Tablo 1’deki özelliklerin değerlerine bağlı olarak T setindeki c-adet gözlem nesnesinin her birinin Q setindeki k-adet gözlem noktalarına olan uzaklıkları Eşitlik 1, Eşitlik 2 ve Eşitlik 3 de verilen uzaklık bağıntıları kullanılarak hesaplanır. Hesaplama sonucunda aşağıdaki uzaklık matrisi elde edilir:

$$d(T, Q) \equiv \begin{pmatrix} d_{t_1 q_1} & \dots & d_{t_1 q_k} \\ \vdots & \ddots & \vdots \\ d_{t_c q_1} & \dots & d_{t_c q_k} \end{pmatrix}_{c \times k} \quad (4)$$

Her bir gözlem nesnesi için uzaklık matrisindeki değerler kullanılarak hiyerarşik kümeleme sınıfının metodları işletilir. Kümeleme sonucunda elde edilen dendrogramlar kullanıcı arayüzü katmanında sunulurken veri setinin test ve analizi gerçekleştirilebilmektedir.

Geliştirilen birleştirici hiyerarşik kümeleme modeli, Microsoft Visual Studio.NET platformunda yer alan C# programlama dili kullanılarak nesne yönelimli olarak programlanmış, bir adet arayüz katmanı ve beş adet sınıftan oluşmaktadır. Geliştirilen model, giriş parametreleri olarak algoritma türü, uzaklık bağıntısı ve hedef dosya bilgilerini kullanmaktadır. Çıkış parametreleri olarak da hedef dosyada yer alan verilerin, seçilen algoritma türüne ve uzaklık bağıntısına göre hiyerarşik kümeleme sonuçlarını metinsel olarak sunmaktadır. Kümeleme sonuçlarının dendrogramlar aracılığıyla görselleştirilmesinde ise Matlab programı kullanılmıştır.

5.2. Modelin UML Diyagramları (UML Diagrams of the Model)

UML (Unified Modeling Language), bir birleşik modelleme dilidir. UML diyagramları genel olarak yapı, eylem ve etkileşim diyagramları olmak üzere üç

Tablo 1. Genelleştirilmiş gözlem tablosu (Generalized observation table)

Gözlem Numarası	Gözlem Adı	Özellik 1	Özellik 2	.	Özellik m
1	Ad-1	Değer-1-1	Değer-2-1	.	Değer-m-1
2	Ad-2	Değer-1-2	Değer-2-2	.	Değer-m-2
.
n	Ad-n	Değer-1-n	Değer-2-n	.	Değer-m-n

türe sahiptir [23, 24]. Burada, geliştirilen modelin sunumunda yapı diyagramları arasında yer alan sınıf diyagramı ve eylem diyagramları arasında yer alan aktivite diyagramı kullanılmıştır.

5.2.1. Sınıf diyagramı (Class diagram)

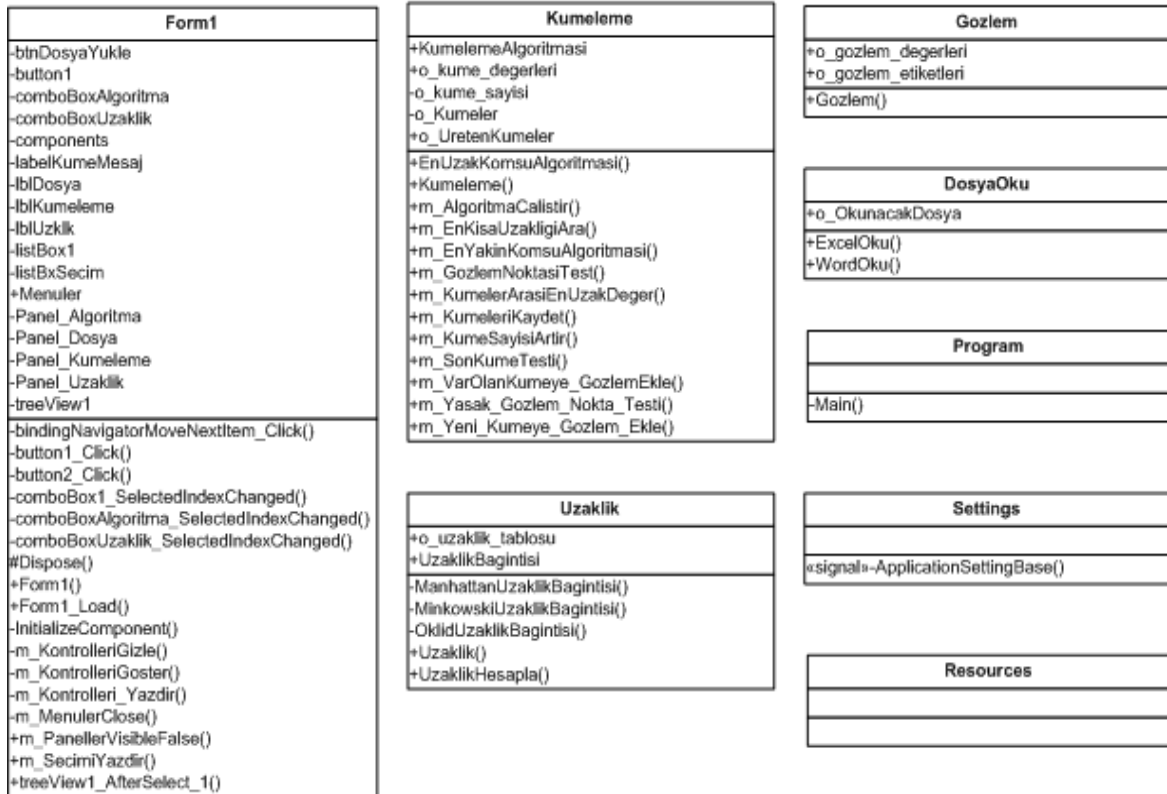
Geliştirilen model Form1, Kümeleme, Uzaklık, Gözlem ve DosyaOku olmak üzere toplam beş adet sınıftan oluşmaktadır. Geliştirilen modelin sınıf diyagramı Şekil 8'de görülmektedir.

5.2.2. Aktivite diyagramı (Activity diagram)

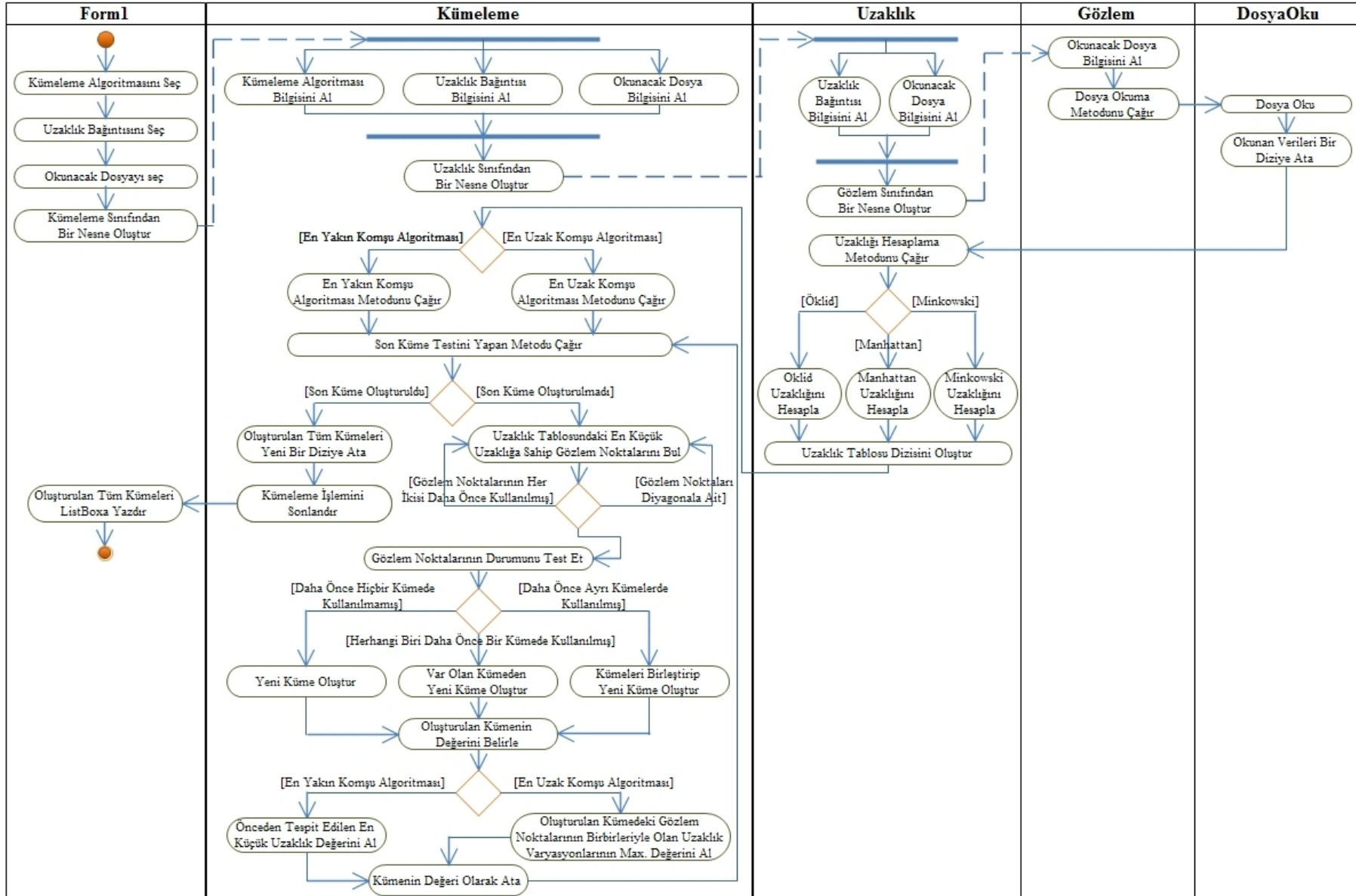
Geliştirilen modelin UML aktivite diyagramı Şekil 9'da görülmektedir. Aktive diyagramından da anlaşılabilir gibi; kullanıcı öncelikle arayüz aracılığıyla kullanmak istediği kümeleme algoritmasını (en yakın komşu algoritması veya en uzak komşu algoritması), algoritmada kullanmak istediği uzaklık bağıntısını (Öklid, Manhattan veya Minkowski) ve daha sonra hiyerarşik olarak kümelemek istediği verilerin bulunduğu hedef dosyayı göstermektedir. Burada dosya formatı olarak Excel kullanılmıştır. Bu işlemler yapıldıktan sonra kullanıcı arayüzde yer alan “kümele” butonuna bastığında, Form1 sınıfında Kümeleme sınıfından bir nesne

oluşturulmaktadır. Bu durumda Kümeleme sınıfının yapıcı metodu aracılığıyla arayüzden alınan “Kümeleme Algoritması”, “Uzaklık Bağıntısı” ve “Okunacak Dosya” bilgileri Kümeleme sınıfına aktarılmaktadır. Daha sonra Kümeleme sınıfında Uzaklık sınıfından bir nesne oluşturulup, Uzaklık sınıfının yapıcı metodu aracılığıyla “Uzaklık Bağıntısı” ve “Okunacak Dosya” bilgileri Uzaklık sınıfına aktarılmaktadır. Uzaklık sınıfında Gözlem sınıfından bir nesne oluşturulup, Gözlem sınıfının yapıcı metodu aracılığıyla “Okunacak Dosya” bilgisi Gözlem sınıfına aktarılmaktadır. Böylece arayüzden alınan bilgiler ilgili sınıflara sunulmuştur.

Gözlem sınıfı “Okunacak Dosya” bilgisini aldıktan sonra DosyaOku sınıfına ait dosya okuma metodunu çağırarak hedef dosyadaki verileri bir diziye atar. Gözlem sınıfı görevini yerine getirdiği için Uzaklık sınıfına döndürülür. Bu sınıfta seçilen uzaklık bağıntısına göre ilgili metod çağırılır ve hedef dosyadaki verileri temsil eden dizi kullanılarak uzaklık tablosu dizisi oluşturulur. Bu aşamada Uzaklık sınıfı da görevini yerine getirdiği için Kümeleme sınıfına döndürülür. Bu sınıfta seçilen algoritma türüne göre ilgili metod çağırılır. Hangi algoritma türü seçilirse seçilsin, öncelikle son kümenin oluşturulup oluşturulmadığı test edilir



Şekil 8. Modelin UML sınıf diyagramı (UML class diagram of the model)



Şekil 9. Geliştirilen modelin UML aktivite diyagramı (UML activity diagram of the model developed)

Eğer son küme oluşturulduysa, oluşturulan tüm kümeler bir diziyeye atanır, kümeleme işlemi sonlandırılır ve oluşturulan tüm kümeler indeks değerleriyle birlikte arayüzde yer alan listBoxa yazdırılır. Eğer son küme oluşturulmadıysa, oluşturulan uzaklık tablosundaki en küçük uzaklığa sahip gözlem noktaları bulunur. Bulunan gözlem noktaları uzaklık matrisinin diyagonalini temsil ediyorsa veya gözlem noktalarının her ikisi daha önce birlikte kullanılmışsa, bu gözlem noktaları dikkate alınmaz. Bu koşullar sağlanmıyorsa, gözlem noktalarının durumu test edilir. Bulunan gözlem noktaları daha önce hiçbir kümede kullanılmamışsa, bu gözlem noktalarından yeni bir küme oluşturulur. Bulunan gözlem noktalarından herhangi biri daha önce bir kümede kullanılmışsa, var olan kümeden yeni bir küme oluşturulur. Bulunan gözlem noktalarının her ikisi de daha önce ayrı ayrı kümelerde kullanılmışsa, bu kümeler birleştirilip yeni bir küme oluşturulur. Bulunan gözlem noktalarının durumuna göre bu şartlar aracılığıyla küme oluşturulduktan sonra oluşturulan kümenin değeri belirlenir.

Eğer seçilen algoritma türü “En Yakın Komşu Algoritması” ise, önceden tespit edilen en küçük

uzaklık değeri oluşturulan kümenin değeri olarak atanır. Eğer seçilen algoritma türü “En Uzak Komşu Algoritması” ise, oluşturulan kümedeki gözlem noktalarının birbirleriyle olan uzaklık varyasyonlarının maksimum değeri oluşturulan kümenin değeri olarak atanır. Bu işlemlerden sonra tekrar son kümenin oluşturulup oluşturulmadığı test edilerek, bu duruma göre model, akışını tamamlamakta ya da sürdürmektedir.

5.3. Geliştirilen Modelin Arayüz Katmanı (Interface Layer of the Model Developed)

Herhangi bir kullanıcı hiyerarşik kümeleme modelini kullanarak veri setini kümelemek istediğinde yapması gereken ilk işlem, veri setini Tablo 1’de verilen geliştirilmiş gözlem tablosu biçiminde hazırlamaktır. Bu işlemden sonra kullanıcı, Şekil 10’da verilen ve geliştirilen modelin uygulamasına ait kullanıcı arayüzünü gösteren yazılım aracını dört basit adımda kullanarak veri setini hiyerarşik olarak kümeleyebilmektedir. Kullanıcı, arayüzde yer alan ağaç yapısındaki menüleri kullanarak, sırasıyla, “Kümeleme Algoritması”, “Uzaklık Bağıntısı” ve “Hedef Dosya” bilgilerini ağacın ilgili dalına tıklayarak, sistemin sunduğu paneller aracılığıyla

Şekil 10. Modelin arayüz katmanı (Interface layer of the model)

seçimlerini yapar. Daha sonra menüde yer alan “Hesaplama” butonunun tıklanması ile, kullanıcının yaptığı tercihleri gösteren ve kümeleme butonuna sahip olan panel kullanıcıya sunulur. Kullanıcı “Kümele” butonuna bastığında, hedef dosyada yer alan veriler hiyerarşik olarak kümelenir ve kümeleme sonuçları arayüzdeki listede (listBox1 kutusu) kullanıcıya metinsel formatta sunulur. Kümeleme sonuçlarının dendrogramlar aracılığıyla görselleştirilmesinde ise Matlab programı kullanılır.

6. GELİŞTİRİLEN NESNE YÖNELİMLİ BİRLEŞTİRİCİ HİYERARŞİK KÜMELEME MODELİNİN TEST EDİLMESİ (TESTING OBJECT ORIENTED AGGLOMERATIVE HIERARCHICAL CLUSTERING MODEL)

Geliştirilen nesne yönelimli birleştirici hiyerarşik kümeleme modelini test etmek amacıyla, T.C. Çevre ve Orman Bakanlığı Devlet Meteoroloji İşleri Genel Müdürlüğünden 44 ile ait “günlük ortalama rüzgâr hızları” alınmıştır. Veri madenciliği çözümlemesinin sağlıklı bir şekilde yapılması amacıyla, öncelikle alınan veri setindeki eksik olan veriler, ilgili değişkenin ortalaması alınarak hesaplanmıştır. Daha sonra her bir ile ait günlük veriler aylık verilere dönüştürülmüştür. Bu ön işlemlerden sonra elde edilen “aylık ortalama rüzgâr hızı” verileri, modele hedef dosya olarak sunulmuş ve böylece Türkiye’deki 44 ilin, “aylık ortalama rüzgâr hızları” açısından hiyerarşik olarak kümelenmesi sağlanmıştır. Kümeleme sonuçları ve bu sonuçlar aracılığıyla elde edilen dendrogramlar kullanıcılara sunulmuş, etkin ve optimum sonuçlar elde edilmiştir.

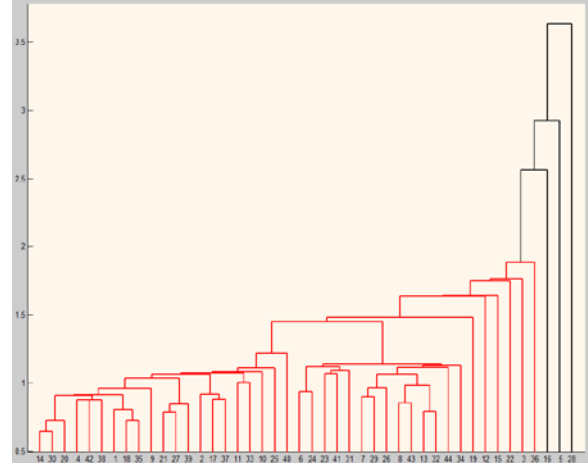
Modele sunulan hedef dosyada yer alan veriler Ek-1’de tablo şeklinde görülmektedir. Bu tabloda her bir gözlem bir ili; her bir gözleme ait değişkenler ise, o ilde her aya ait ortalama rüzgâr hızlarını temsil etmektedir. Tabloda yer alan iller sütunu, anlaşılabilirliğin artırılması amacıyla modele sunulan hedef dosyada da yer almaktadır fakat gözlemler, gözlem numaraları ile temsil edilmektedir. Kullanıcının arayüzden yaptığı seçimler ve elde edilen dendrogramlar aşağıdaki testlerde ayrı ayrı sunulmuştur.

6.1. Test I

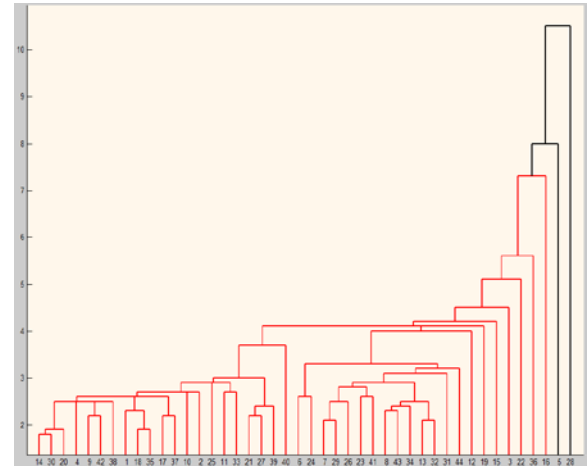
Kümeleme Algoritması=En Yakın Komşu Algoritması ve Uzaklık Bağıntısı=Öklid Uzaklık Bağıntısı olarak tercih edildiğinde elde edilen dendrogram Şekil 11’de görülmektedir.

6.2. Test II

Kümeleme Algoritması=En Yakın Komşu Algoritması ve Uzaklık Bağıntısı=Manhattan Uzaklık Bağıntısı olarak tercih edildiğinde elde edilen dendrogram Şekil 12’de görülmektedir.



Şekil 11. Test I şartlarında elde edilen dendrogram (Obtained dendrogram under Test I circumstances)



Şekil 12. Test II Şartlarında Elde Edilen Dendrogram (Obtained dendrogram under Test II circumstances)

6.3. Test III

Kümeleme Algoritması=En Yakın Komşu Algoritması ve Uzaklık Bağıntısı=Minkowski Uzaklık Bağıntısı olarak tercih edildiğinde elde edilen dendrogram Şekil 13’de görülmektedir.

6.4. Test IV

Kümeleme Algoritması=En Uzak Komşu Algoritması ve Uzaklık Bağıntısı=Öklid Uzaklık Bağıntısı olarak tercih edildiğinde elde edilen dendrogram Şekil 14’de görülmektedir.

6.5. Test V

Kümeleme Algoritması=En Uzak Komşu Algoritması ve Uzaklık Bağıntısı=Manhattan Uzaklık Bağıntısı olarak tercih edildiğinde elde edilen dendrogram Şekil 15’de görülmektedir.

6.6. Test VI

Kümeleme Algoritması=En Uzak Komşu Algoritması ve Uzaklık Bağıntısı=Minkowski Uzaklık Bağıntısı olarak tercih edildiğinde elde edilen dendrogram Şekil 16'de görülmektedir.

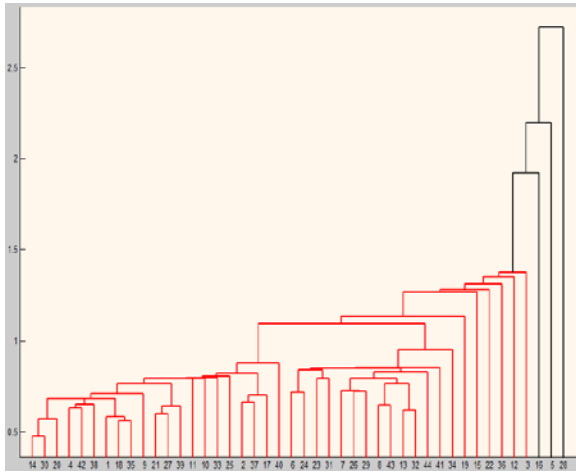
6.7. Test Sonuçlarının Değerlendirilmesi (Evaluation of the Test Results)

Test sonuçları incelendiğinde; hem en yakın komşu algoritması hem de en uzak komşu algoritmasında kullanılan tüm uzaklık bağıntılarında “aylık ortalama rüzgâr hızları” açısından” birbirine en çok benzeyen iller Erzincan ve Muş olmuştur. Başka bir ifadeyle, birleştirici hiyerarşik kümeleme analizlerinin tümünde Erzincan ve Muş “aylık ortalama rüzgâr hızları” açısından birbirine en çok benzeyen iller olarak tespit edilmiştir.

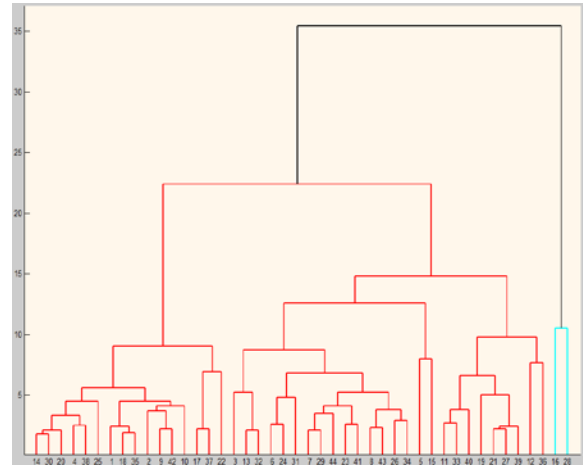
Genel itibariyle, hem en yakın hem de en uzak komşu algoritmasında, tercih edilen uzaklık bağıntısına göre aynı iller için farklı benzerlik seviyesinde farklı

kümeler elde edilmiştir. Bu durum tercih edilen uzaklık bağıntısından kaynaklanmaktadır. Bilgi alanı uzmanı test sonuçları ile hedef dosyada yer alan verileri karşılaştırdığında, hangi uzaklık bağıntısını tercih etmesi gerektiğini rahatlıkla belirleyebilmektedir. Burada yapılan çalışmada hem en yakın komşu algoritması hem de en uzak komşu algoritması için optimum hiyerarşik kümeleme sonuçlarını Öklid uzaklık bağıntısı vermektedir. Literatürdeki çalışmalarda da yaygın olarak kullanılan Öklid uzaklık bağıntısıdır. İki algoritmadan birinin tercih edilmesi gerektiğinde ise en uzak komşu algoritması daha etkin sonuçlar vermiştir.

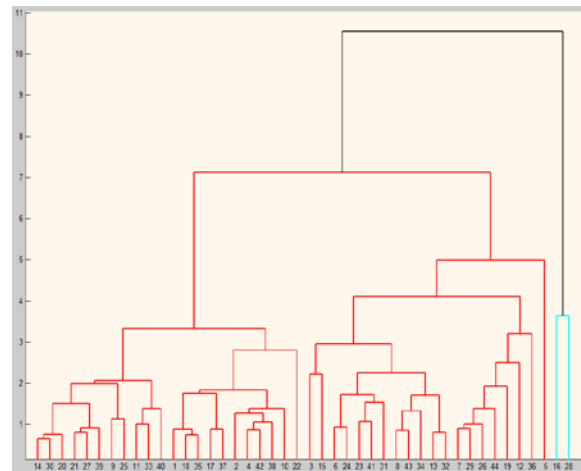
En yakın komşu algoritmasında kullanılan tüm uzaklık bağıntılarında, Şırnak kendinden önceki illere en az benzeyen ildir çünkü Şırnak'tan önceki tüm iller bir küme oluştururken; Şırnak en son olarak bu kümeye katılmıştır. En uzak komşu algoritmasında kullanılan tüm uzaklık bağıntılarında ise, Kahramanmaraş kendinden önceki illere en az benzeyen ildir.



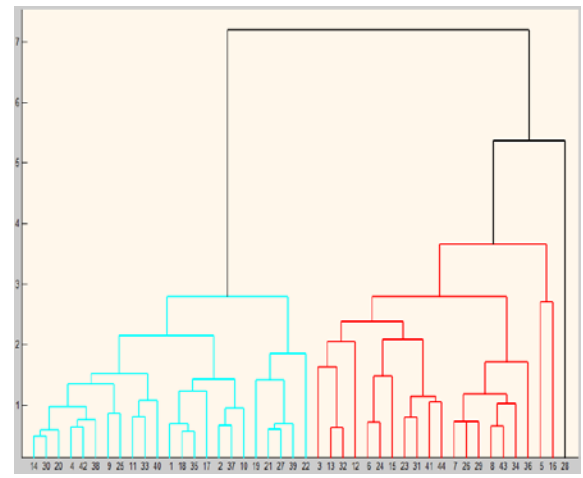
Şekil 13. Test III Şartlarında Elde Edilen Dendrogram (Obtained dendrogram under Test III circumstances)



Şekil 15. Test V Şartlarında Elde Edilen Dendrogram (Obtained dendrogram under Test V circumstances)



Şekil 14. Test IV Şartlarında Elde Edilen Dendrogram (Obtained dendrogram under Test IV circumstances)



Şekil 16. Test VI Şartlarında Elde Edilen Dendrogram (Obtained dendrogram under Test VI circumstances)

En yakın komşu algoritmasında kullanılan tüm uzaklık bağıntılarında, Mardin'e en çok benzerliği Antakya göstermiştir. En uzak komşu algoritmasında ise, Öklid ve Manhattan Uzaklık bağıntılarında Mardin'e en çok benzerliği Eskişehir gösterirken; Minkowski uzaklık bağıntısında Mardin'e en çok benzerliği, hem Eskişehir hem de Kahramanmaraş'ın bulunduğu son küme göstermiştir. Yapılan bu analizlere benzer değerlendirmeler diğer tüm iller için de rahatlıkla yapılabilmektedir.

7. SONUÇLAR (CONCLUSIONS)

Bu makale çalışmasında günümüzdeki hiyerarşik kümeleme sistemlerinin temel kısıtları olan uygulamaya bağımlılık ve nesne modeli eksikliği konularında çözümler üretilmiştir. Bu amaçla çalışmada hiyerarşik kümeleme için genelleştirilmiş bir gözlem nesnesi modeli geliştirilmiştir. Geliştirilen gözlem nesnesi modeli sayesinde farklı alanlardaki veri setlerini temsil edebilecek esnek bir veri yapısı elde edilmiştir. Geliştirilen hiyerarşik kümeleme modelinin katmanlı mimarisi sayesinde her katmanın görevini üstlenen sınıflar tasarlanmış ve sınıflar arasındaki veri yapısı bütünlüğü “gözlem nesnesi” ile sağlanmıştır. Çok katmanlı mimari yapı aynı zamanda sınıflar arası bağımsızlığı da sağlamış, uygulamada ve algoritmada meydana gelebilecek değişikliklere de esnek bir şekilde uyum sağlama yeteneğini kazandırmıştır. Hiyerarşik kümeleme algoritmasında uzaklık sınıfına yeni bir uzaklık metodu eklenmek istendiğinde yapılması gereken tek işlem gözlem nesnesinden yaratılan bir uzaklık bağıntısını uzaklık sınıfında tanımlamaktır. Bu esneklik diğer sınıflar içinde aynı şekilde geçerlidir. Makale çalışmasının devamında geliştirilen modelin uygulandığı form-tabanlı bir yazılım aracı sunulmuştur. Geliştirilen yazılım aracının tek girdisi genelleştirilmiş gözlem tablosu formatında hazırlanmış bir veri setidir. Sisteme yüklenen veri seti hangi uygulama alanından olursa olsun, gözlem sınıfının dinamik ve esnek yapısı sayesinde gözlem nesnelere tarafından temsil edilebilmekte ve uygulamanın diğer katmanlarındaki uzaklık ve kümeleme bağıntılarına sorunsuz bir şekilde aktarılabilirler. Gerek geliştirilen genelleştirilmiş nesne modelinin esnekliği gerekse de geliştirilen yazılımın kullanım kolaylığı hiyerarşik kümeleme yönteminin çeşitli sebeplerle uygulanmadığı birçok alanda da uygulanmasına imkân yaratmıştır.

Geliştirilen modelin interaktifliği sayesinde kullanıcılar sistem arayüzünü rahatlıkla kullanabilmekte; nesne yönelimli programlama sayesinde de kullanıcı talepleri hızlı bir şekilde karşılanabilmektedir.

Geliştirilen modelin test aşamasında, T.C. Çevre ve Orman Bakanlığı Devlet Meteoroloji İşleri Genel Müdürlüğünden 44 ile ait “günlük ortalama rüzgâr

hızları” alınmış ve birtakım önışlemlerden sonra elde edilen 44 ile ait “aylık ortalama rüzgâr hızı” verileri hiyerarşik olarak kümelendi. Kümeleme sonuçları arayüzde metinsel formatta sunulurken; Matlab'da ise dendrogramlar aracılığıyla görselleştirilmiştir. Kümeleme sonuçları incelendiğinde, önceden gizli olan fakat model sayesinde ortaya çıkarılan pek çok çıkarım elde edilmiştir. Bununla birlikte, modelde yer alan algoritmaların ve uzaklık bağıntılarının karşılaştırmalı olarak analizi yapılmıştır. Kullanıcılara önerilerde bulunulmuştur. Burada yapılan çalışmada hem en yakın komşu algoritması hem de en uzak komşu algoritması için optimum hiyerarşik kümeleme sonuçlarını Öklid uzaklık bağıntısı vermektedir. İki algoritmadan birinin tercih edilmesi gerektiğinde ise en uzak komşu algoritması daha etkin sonuçlar vermiştir. Bilgi alanı uzmanı bu değerlendirmeleri rahatlıkla yapabilmektedir.

Gelecekte yapılacak çalışmalarda, geliştirilen bu platforma ayrıştırıcı hiyerarşik kümeleme modeli de adapte edilerek, uygulamanın web tabanlı ve gerçek zamanlı olarak tasarlanması sistemin kullanılabilirliğini arttıracaktır.

8. KAYNAKLAR (REFERENCES)

1. Akpınar, H., “Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği”, **İstanbul Üniversitesi İşletme Fakültesi Dergisi**, Cilt 29, No 1, 1-22, 2000.
2. Luo, Q., “Advancing Knowledge Discovery and Data Mining”, **The First International Workshop on Knowledge Discovery and Data Mining (WKDD'08)**, Adelaide, SA, 3-5, 23-24 Ocak 2008.
3. Essid, S., Richard, G. ve David, B., “Inferring Efficient Hierarchical Taxonomies for MIR Tasks: Application to Musical Instruments”, **The 6th International Conference on Music Information Retrieval (ISMIR'05)**, London, UK, 324-328, 11-15 Eylül 2005.
4. Choi, Y. S., Kim, S. J. ve Lee, S., “Hierarchical Shot Clustering for Video Summarization”, **The Seventh European Conference on European Interactive Television (EuroITV'09)**, New York, USA, 1100-1107, 3-5 Haziran 2009.
5. Fung, B. K., Wang, K. ve Ester, M., “Hierarchical Document Clustering”, **Encyclopedia of Data Warehousing and Mining**, Idea Group Reference, Cilt 1, A-H, 555-559, 2004.
6. Doherty, K. J., Adams, R. G., Davey, N. ve Pensuwon, W., “Hierarchical Topological Clustering Learns Stock Market Sectors”, **Congress on Computational Intelligence Methods and Applications**, İstanbul, Türkiye, 11-17 Aralık 2005.
7. Beaver, S. ve Palazoglu, A., “Cluster Analysis of Hourly Wind Measurements to Reveal Synoptic Regimes Affecting Air Quality”, **Journal of**

- Applied Meteorology and Climatology**, Cilt 45, No 12, 1710-1726, 2006.
8. Karabulut, M., Gürbüz, M. ve Sandal, E. K., “Hiyerarşik Küme Tekniği Kullanılarak Türkiye’deki İllerin Sosyo-ekonomik Benzerliklerinin Analizi”, **SDÜ Sosyal Bilimler Enstitüsü Dergisi**, Cilt 3, No 5, 65-78, 2008.
 9. Sandal, E. K., “Sosyo-ekonomik Kriterler Bakımından Türkiye, Doğu Avrupa ve Kafkas Ülkelerinin Karşılaştırılması”, **Doğu Coğrafya Dergisi**, Cilt 22, No 1, 89-105, 2009.
 10. Ersöz, F., “Savunma Yeteneklerinin Geliştirilmesinde Bilgi&Arge ve Inovasyonun Rolü ile Inovasyon Göstergelerine Göre Türkiye’nin Yeri”, **TÜBAV Bilim Dergisi**, Cilt 2, No 3, 664-371, 2009.
 11. Sever, H. ve Oğuz, B., “Veri Tabanlarında Bilgi Keşfine Formel Bir Yaklaşım: Eşleştirme Sorguları ve Algoritmalar”, **Bilgi Dünyası**, Cilt 3, No 2, 173-204, 2002.
 12. Fayyad, U., Shapiro, G. P. ve Symth, P., “The KDD Process for Extracting Useful Knowledge From Volumes of Data”, **Communications of the ACM**, Cilt 39, No 11, 27-34, 1996.
 13. Han, J. ve Kamber, M., “Introduction”, **Data Mining Concepts and Techniques**, Elsevier Inc., San Francisco, USA, 21-27, 2006.
 14. Kohavi, R., “Data Mining and Visualization”, **The Sixth Annual Symposium on Frontiers of Engineering**, , Irvine, California, 14-16 Eylül 2000.
 15. Berry, M. J. A. ve Linoff, G. S., “Data Mining Applications”, **Data Mining Techniques For Marketing Sales And Customer Relationship Management**, Wiley Publishing, Canada, 87-120,2004.
 16. Özkan, Y., “Kümeleme”, **Veri Madenciliği Yöntemleri**, Papatya Yayıncılık, İstanbul, 131-148, 2008.
 17. Tan, P. N., Steinbach, M. ve Kumar, V., “Cluster Analysis: Basic Concepts and Algorithms”, **Introduction to Data Mining**, Addison-Wesley, Boston, USA, 487-495, 2006.
 18. Witten, I. H. ve Frank, E., “Implementations: Real machine learning schemes”, **Data Mining Practical Machine Learning Tools and Techniques**, Morgan Kaufmann Publishers, San Francisco, 235-242, 2005.
 19. Internet: <http://www.lrde.epita.fr/dload//20080702-Seminar/folio-dt.pdf>
 20. Merigó, J. ve Gil-Lafuente, A., “Using the OWA operator in the minkowski distance”, **International Journal of Computer Science**, Cilt 3, No 3, 149-157, 2008.
 21. Larose, D. T., “Hierarchical and k-Means Clustering”, **Discovering Knowledge in Data: An Introduction to Data Mining**, John Wiley & Sons Inc., Hoboken, NJ, 147-162, 2005.
 22. Agarwal, P., Alam, M. A. ve R. Biswas, “Analysing the Agglomerative Hierarchical Clustering Algorithm for Categorical Attributes”, **International Journal of Innovation, Management and Technology**, Cilt 1, No. 2, 186-190, 2010.
 23. Internet: http://ceng.gazi.edu.tr/~hkaracan/NYP_H6.pdf
 24. Larman, C., “More UML Notation”, **Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Unified Process**, Prentice Hall, Upper Saddle River, NJ, 607-608, 2002.

EKLER (APPENDICES)

EK-1. 44 ile ait aylık ortalama rüzgâr hızı verileri (Monthly average wind speed data belonging to 44 cities)

Gözlemler	İller	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
1	Adana	1,5	1,5	1,1	1,1	1,2	1,3	1,3	1,3	1,2	0,7	0,8	1,2
2	Ağrı	0,5	0,6	1,5	1,4	1,2	1,1	1,0	1,2	1,2	0,8	1,0	0,2
3	Aksaray	2,6	2,2	3,4	2,5	2,1	3,4	3,2	3,4	2,1	2,0	2,3	2,0
4	Amasya	1,3	0,9	1,5	1,2	1,5	1,2	1,4	1,4	1,3	0,9	1,0	0,9
5	Antakya	1,4	1,8	2,1	2,7	3,1	3,7	4,6	4,2	2,8	1,5	1,1	1,5
6	Ardahan	1,2	1,7	2,8	3,4	2,5	2,5	2,1	2,3	2,3	2,0	2,0	1,3
7	Artvin	1,7	2,0	1,9	1,9	2,2	2,3	2,7	2,4	1,9	1,3	1,3	1,5
8	Batman	2,1	2,0	2,5	2,6	2,2	2,6	2,1	2,1	2,2	1,8	1,8	1,7
9	Bayburt	0,9	1,1	1,8	1,9	1,3	1,6	1,1	1,0	1,0	0,7	1,1	0,9
10	Bingöl	0,8	0,7	1,0	0,9	1,2	1,7	1,3	1,2	1,4	1,2	1,3	0,8
11	Bitlis	2,0	1,5	1,6	1,5	1,4	1,6	1,3	1,5	1,4	1,1	1,2	1,4
12	Diyarbakır	2,2	2,3	1,9	2,4	2,5	4,0	2,8	2,1	1,8	1,3	1,0	1,7
13	Elazığ	2,6	2,3	2,9	2,8	2,3	2,6	2,3	2,0	2,2	1,9	1,8	2,3
14	Erzincan	0,9	0,8	1,4	1,8	1,6	1,8	1,8	1,5	1,3	0,9	1,1	1,1
15	Erzurum	1,3	1,8	2,8	2,8	2,8	2,7	3,0	3,5	2,4	2,6	2,6	1,0
16	Eskişehir	2,9	3,0	3,8	3,1	3,5	3,3	3,7	3,8	3,1	2,5	2,3	3,3
17	Gaziantep	0,7	0,8	0,8	0,9	0,8	1,2	1,0	0,7	0,6	0,3	0,4	0,6
18	Giresun	1,3	1,0	1,4	1,0	0,9	1,3	1,2	0,8	1,1	0,6	0,9	1,3

19	Gümüşhane	1,5	1,0	1,4	1,6	2,2	2,8	3,0	2,2	1,7	1,1	1,2	0,9
20	Hakkâri	1,2	0,8	1,5	1,6	1,6	1,7	1,7	2,0	1,5	1,1	0,9	1,1
21	İğdır	0,9	1,4	2,0	1,8	2,0	2,3	2,0	1,6	1,4	1,1	0,8	1,1
22	Kahramanmaraş	0,5	0,4	0,9	1,3	1,7	2,4	2,4	1,6	1,2	0,4	0,0	0,3
23	Karaman	1,9	1,4	3,1	2,5	2,0	2,4	2,5	2,2	2,1	1,4	1,7	1,4
24	Kars	1,6	1,7	3,0	2,8	2,7	2,6	2,4	2,6	2,4	1,8	1,8	1,3
25	Kayseri	1,0	1,1	2,5	2,3	1,3	1,6	1,4	1,4	1,4	1,1	1,1	1,1
26	Kilis	1,7	2,0	2,2	2,3	2,4	2,8	2,8	2,4	1,9	1,5	1,8	1,8
27	Malatya	1,2	1,3	2,0	2,0	1,8	2,1	1,8	1,5	1,7	1,2	1,3	1,1
28	Mardin	4,4	4,8	3,1	3,5	3,1	3,9	3,3	3,0	3,2	3,2	3,4	5,3
29	Mersin	1,7	2,0	1,9	2,5	2,0	2,3	2,6	2,3	2,2	1,8	1,5	1,4
30	Muş	0,9	1,1	1,6	1,7	1,5	1,5	1,8	1,5	1,6	1,0	0,9	0,9
31	Nevşehir	1,4	1,4	3,1	2,6	1,6	2,1	2,0	1,9	1,7	1,5	2,1	1,3
32	Niğde	2,6	2,4	3,4	3,0	1,9	2,6	2,4	2,3	2,1	1,8	2,0	2,4
33	Ordu	1,3	1,3	1,4	1,1	1,5	1,6	1,7	1,6	1,3	1,3	1,4	1,5
34	Osmaniye	1,9	2,9	2,4	2,4	2,2	2,7	2,7	2,2	2,2	1,9	1,7	1,7
35	Rize	1,3	1,3	1,4	1,4	1,1	1,3	1,1	0,9	1,2	1,0	1,1	1,2
36	Samsun	2,6	2,7	2,0	1,4	1,6	1,7	2,0	1,9	1,7	1,3	1,6	2,6
37	Siirt	0,8	0,8	1,2	1,0	1,1	1,2	1,0	0,8	1,2	0,6	0,6	0,5
38	Sivas	1,1	0,9	1,6	1,4	1,5	1,6	1,6	1,3	1,1	0,6	0,7	0,4
39	Şanlıurfa	1,2	1,3	1,6	2,0	2,1	2,5	2,1	1,6	1,5	1,2	1,1	1,6
40	Şırnak	1,5	1,7	1,2	1,7	2,0	1,2	0,8	1,8	1,4	1,4	1,4	1,3
41	Tokat	1,8	2,1	3,1	2,4	2,0	2,2	2,7	2,7	1,9	1,9	1,7	1,3
42	Tunceli	0,9	0,6	1,8	1,2	1,2	1,5	1,2	1,3	0,9	0,9	1,2	0,9
43	Van	2,1	2,1	2,3	2,3	2,2	2,2	2,1	2,3	2,4	1,9	2,1	2,2
44	Yozgat	1,8	1,8	2,2	1,9	1,6	2,2	2,2	2,5	1,6	1,6	1,8	1,3

