

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**WEB GÜNLÜĞÜNDEN ZAMAN SINIRLI BULANIK BAĞINTI
KURALLARI VE SIRALI ÖRÜNTÜLERİN ÇIKARILMASI**

İsmail İŞERİ

Tez Yöneticisi

Yrd.Doç.Dr. Mehmet KAYA

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ELAZIĞ,2005

T.C. .
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**WEB GÜNLÜĞÜNDEN ZAMAN SINIRLI BULANIK BAĞINTI
KURALLARI VE SIRALI ÖRÜNTÜLERİN ÇIKARILMASI**

İsmail İŞERİ

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Bu tez, 28.07.2025 tarihinde, aşağıda belirtilen jüri tarafından oybirliği / ~~oyçokluğu~~
ile başarılı / ~~başarısız~~ olarak değerlendirilmiştir.

Danışman : Yrd.Doç.Dr. Mehmet KAYA

Üye : Prof.Dr. Yakup DEMİR

Üye : Yrd.Doç.Dr. Ali KARCI

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 29.07.2025 tarih ve
2005.24/1.2... sayılı kararıyla onaylanmıştır.

TEŞEKKÜR

Bu tez çalışması boyunca, bilgi ve yardımlarını esirgemeyen danışmanım Sayın Yrd. Doç. Dr. Mehmet KAYA'ya, teşekkürlerimi ve şükranlarımı sunarım.



İÇİNDEKİLER

Sayfa

TEŞEKKÜR

İÇİNDEKİLER	I
ŞEKİLLER LİSTESİ	III
TABLolar LİSTESİ	IV
SİMGELER VE KISALTMALAR LİSTESİ.....	V
ÖZET	VI
1.GİRİŞ.....	1
1.1 Tezin Amacı.....	1
1.2. Tezde Yapılanlar.....	2
2. VERİ MADENCİLİĞİ.....	3
2.1. Veriden Maden Elde Edilme Süreci	5
2.2. Bağintı Kuralları ve Sıralı Örüntüler	6
2.3. Bağintı Kurallarında Temel Kavramlar ve Apriori Algoritması.....	7
2.4. Örnek Uygulama	8
2.5. Minimum Destek ve Minimum Güven Değerleri.....	9
2.6. Bulanık Küme Kavramı	9
3. WEB MADENCİLİĞİ.....	10
3.1. Giriş	10
3.2. Web Madenciliği ve Bilgiyi Yeniden Elde Etme	11
3.3. Web Madenciliği ve Bilgi Çıkarımı	11
3.4. Web Madenciliği Kategorileri	11
3.4.1 Web İçerik Madenciliği	13
3.4.2 Web Yapı Madenciliği	14
3.4.3 Web Kullanım Madenciliği	14
3.5 Veri İşleme	15
3.5.1 İçerik Önişleme	15
3.5.2 Yapı Önişleme	16
3.5.3 Kullanım Önişleme	16
3.6 Örüü Keşfi	17
3.7 İstatistiksel Analiz	17
3.8 İlişki Kuralları	17
3.9 Kümeleme	17
3.10 Sınıflandırma	17

3.11 Örüntü Analizi	17
4. ÇOKLU SEVİYELİ ZAMAN SINIRLI BULANIK BAĞINTI KURALLARI	
VE SIRALI ÖRÜNTÜLER.....	19
4.1. Algoritmada Kullanılan Notasyonlar.....	20
4.2. Algoritmanın Adımları	20
5. ÖNERİLEN YÖNTEMLE ÇOKLU SEVİYELİ BULANIK BAĞINTI KURALI	
VE SIRALI ÖRÜNTÜLER ÇIKARMA.....	25
5.1. Önişlemler.....	25
5.2. Algoritmanın Düzenlenmiş Web Verisi Üzerine Uygulanması.....	26
6. GELİŞTİRİLEN YAZILIM VE UYGULAMA	35
6.1. Giriş	35
6.2. Web Günlük Verisinin Düzenlenmesi ve Uygulama Sonuçları.....	35
6. SONUÇLAR.....	48
7. KAYNAKLAR	49
8. ÖZGEÇMİŞ	51

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1. Veri madenciliği işlemleri	3
Şekil 2.2. Verinin hazırlanması	5
Şekil 2.3. Veri çıkarım evreleri	6
Şekil 3.1. Web madenciliği sınıflandırması.....	12
Şekil 3.2. Yüksek Seviyeli Web Kullanım Madenciliği Süreci.....	16
Şekil 4.1. Örnek Site Sınıflandırma Ağacı.....	19
Şekil 5.1. Üçgen üyelik fonksiyonları.....	25
Şekil 6.1. Web günlük verisinin işlenmemiş hali	36
Şekil 6.2. Gaziosmanpaşa Üniversitesi site ağacından bir kesit.	37
Şekil 6.3. Sayfa kodlarının tutulduğu tablo	38
Şekil 6.4. Günlük verisinin sayfa kodlama işleminden geçmiş hali.	39
Şekil 6.5. Sayfa kodlarının '*' karakteri ile kodlanmış hali	40
Şekil 6.6. Bulanıklaştırılmış veriden bir görünüm	41
Şekil 6.7. Web madenciliği programının ana sayfa ekran görüntüsü	42
Şekil 6.8. Program tarafından elde edilen ikili yoğun nesne kümesi.....	43
Şekil 6.9. Program tarafından elde edilen bulanık bağıntı kuralları	44
Şekil 6.10. Program tarafından elde edilen sıralı örüntüler	45
Şekil 6.11. Minimum Destek – Yoğun Nesne kümesi ilişkisi	46
Şekil 6.12. Minimum Güven – Kural Sayısı ilişkisi	47

TABLÖLAR LİSTESİ

	<u>Sayfa</u>
Tablo 2.1. Örnek bir veritabanı.....	8
Tablo 2.2. Örnek veri tabanı yoğun nesne kümesi.....	8
Tablo 5.1. Örnekte kullanılan web sunucu günlük verisi örneği	24
Tablo 5.2. Dosya isimlerinin site ağacına göre kodlanması	26
Tablo 5.3. Kullanıcıların ziyaret ettikleri sayfa bekleme süreleri ile zaman damgası değerleri	27
Tablo 5.4. Sayfaların rakamlar yerine “*” karakteri gelecek şekilde yeniden kodlanması	27
Tablo 5.5. Bekleme sürelerinin bulanık kümelere dönüştürülmesi	28
Tablo 5.6. Bekleme sürelerinin bulanık kümelere dönüştürülmesi	29
Tablo 5.7. Her bölgenin sayısal büyüklük değerleri... ..	30
Tablo 5.8. L_1 geniş nesne kümesi.....	31
Tablo 5.9. C_2 aday nesne kümesi.....	31
Tablo 5.10. (2**,Short,3**,Middle) ikili nesne kümesi için sonuçlar	32
Tablo 5.11. C_2 içerisindeki her nesne için sayısal büyüklük değerleri	32

SİMGELER VE KISALTMALAR LİSTESİ

- n : Toplam günlük kayıt veri sayısı ;
- m : Günlük verisindeki toplam dosya sayısı ;
- c : Günlük verisi içerisindeki toplam istemci sayısı ;
- n_i : i'ninci kullanıcının toplam günlük verisi içerisindeki kayıt sayısı ($1 \leq i \leq c$);
- D_i : i' ninci kullanıcının sitedeki sayfalar arasındaki ziyaret sırası. ($1 \leq i \leq c$);
- D_{id} : D_i içerisinde d'ninci kayıt hareketi, ($1 \leq d \leq n_i$);
- I^g : Günlükteki toplam dosyalar içerisinde g.ninci sayfa , $1 \leq g \leq m$;
- R^{g_k} : I^g sayfasının k.nıncı bulanık bölgesi , $1 \leq k \leq |I^g|$;
- $|I^g|$: I^g sayfasının bulanık bölgelerdeki sayısı ;
- v_{id}^g : I^g sayfasının D_{id} ziyareti içerisindeki geçen süre;
- f_{id}^g : v_{id}^g 'den dönüştürülen bulanık küme;
- $f_{id}^{g_k}$: v_{id}^g Bulanık kümesinin, R^{g_k} bulanık bölgesindeki üyelik değeri;
- $f_i^{g_k}$: i.ninci ziyaretçinin D_i sıralı ziyaretindeki R^{g_k} bulanık bölgesinin üyelik değeri;
- $count^{g_k}$: R^{g_k} bulanık bölgesinin sayısal büyüklüğü;
- $\max count^{g_k}$: $count^{g_k}$ değerleri içerisinde maksimum değer ;
- $\max - R^g$: I^g sayfasının maksimum count değerine sahip olduğu fuzz bölgesi ;
- α : Önceden tanımlanmış minimum destek değeri ;
- λ : Önceden tanımlanmış minimum güven değeri ;
- C_r : Her birisi r adet dosya içeren aday nesne kümeleri ;
- L_r : Her birisi r adet dosya içeren geniş nesne kümeleri ;

ÖZET

YÜKSEK LİSANS TEZİ

WEB GÜNLÜĞÜNDEN ZAMAN SINIRLI BULANIK BAĞINTI KURALLARI VE SIRALI ÖRÜNTÜLERİN ÇIKARILMASI

İsmail İŞERİ

Fırat Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

2004, Sayfa: 51

Bu tezde, web günlük verilerinden çoklu seviyeli, zaman sınırlı bulanık bağıntı kuralları ve sıralı örüntüler elde etmeye yarayan bir web madenciliği çalışması yapılmıştır. Son zamanlarda yapılan çalışmalarda hep basit bağıntı kuralları ve sıralı örüntüler üzerine odaklanılmıştır. Ayrıca bu çalışmaların birçoğu web sayfalarına yapılan ziyaretler dikkate alınarak doğrudan sıralı örüntüler ve ilişkiler elde etmek üzerine olmuştur. Dahası bu kuralların hemen hepsi tek seviyelidir.

Bu tezde, web sayfalarına göz atma zamanı ele alınmış ve dilsel terimlerle biçimlendirmek için bulanık küme kavramı tanımlanmıştır. Böylece önerilen algoritma, dilsel terimlerle zaman sınırlı çok seviyeli göz atma davranışını, etkin bağıntı kuralları ve sıralı örüntülerin elde edilmesini sağlar.

Gerçek bir veri kümesi üzerinde gerçekleştirilen uygulama sonuçları, önerilen yöntemin uygulanabilirliğini ve etkinliğini gösterir.

Anahtar Kelimeler: Veri madenciliği, Web Kullanım Madenciliği, Bulanık Mantık, Sıralı Örüntüler, Bağıntı Kuralları

ABSTRACT

MS Thesis

MINING TIME CONSTRAINED FUZZY ASSOCIATION RULES AND SEQUENTIAL PATTERNS FROM WEB LOGS

İsmail İŞERİ

Firat University

Graduate School of Natural and Applied Sciences

Department of Computer Engineering

2005, Page: 51

In this thesis, a novel web mining algorithm has been proposed to discover time constrained multiple-level browsing patterns from log data in web servers. Web mining of browsing including simple sequential patterns and association rules with browsing times has been studied recently. However, most of these works focus on mining browsing patterns of web pages directly. Moreover, almost all of them deal with single level rule and patterns. In this thesis, we have introduced the problem of mining browsing patterns on multiple levels of a taxonomy comprised of web pages.

In addition, browsing time was considered and processed using fuzzy set concepts to form linguistic terms. The proposed algorithm thus discovers time constrained multiple-level relevant browsing behavior from linguistic data and promotes the discovery of coarsen granularity of web browsing patterns.

The experimental results conducted on a real data set show the applicability and efficiency of the proposed method.

Keywords: Data Mining, Web Usage Mining, Fuzzy Sets, Sequential Patterns, Association Rules

1. GİRİŞ

Günümüzde www (world wide web) uygulamaları çok hızlı gelişmiş ve bilgisayar sistemleri üzerinde önemli etkiler yapmıştır. Bunlardan kullanışlı bilgi için web verisinin taranması en yaygın kullanılan uygulama alanlarından biridir. Bu nedenle, etkili ve verimli web erişimi bu alanda çok önemli bir araştırma konusu haline gelmiştir. Web madenciliği alanında son yıllarda geliştirilen teknikler bu amaca ulaşmak içindir [1].

Literatürde web madenciliği; web içerik madenciliği, web yapı madenciliği ve web kullanım madenciliği olmak üzere üç ayrı kategoride ele alınmaktadır [2]. Bunlardan web içerik madenciliği, web verileri üzerinden bilgi çıkarma üzerinde yoğunlaşmıştır. Web yapı madenciliğinin amacı, web sitesi ve web sayfası hakkında bağlantı verisine bakarak bilgi üretmektir. Teknik olarak, web içerik madenciliği dokümanın içeriğine, yapı madenciliği ise dokümanlar arası bağlantılara yoğunlaşır. Web kullanım madenciliği ise kullanıcı web ile etkileşimde bulunurken kullanıcı davranışları hakkında tahmin etme tekniklerine odaklanır.

Geçmişte, web verilerinden sıralı örüntülerin bulunması ve ilginç kullanıcı bilgilerinin çıkarılması için birçok web madenciliği yaklaşımı sunulmuştur. Chen ve Sycara [3], geliştirdikleri Web Mate isimli sistemlerinde göz atılan web sayfalarının içeriğinden kullanıcı meraklarını belirlemeyi ve saklamayı amaçlamışlardır. Bu sayede kullanıcılara web üzerinde yapacakları aramalarda daha kolay bir yöntem sunmuşlardır. Chen ve diğ. [4] yaptığı çalışmada, öncelikle günlük verilerinden maksimum referansa sahip olanları elde etmiş, bunları daha sonra sık referansa sahip sıralıları elde etmede kullanmıştır. Chen ve diğ. [5], sunucu kayıt günlüklerinin sadece bir kısmını kullanarak kullanıcı erişim örüntülerini elde etmeye çalışmıştır. Bu parçalar daha sonra günlükler olarak tutulmuş ve web sunucunun verimliliği için kullanılmıştır.

Görüldüğü gibi dünyadaki internet devriminin beraberinde getirdiği world wide web kavramı üzerine yüzlerce çalışma yapılmış ve yapılmaya da devam edilmektedir. Çünkü daha elde edilemeyen çok sayıda bilgi ve keşfedilmeyi bekleyen çok sayıda yöntem vardır.

1.1 Tezin Amacı

Bu tez çalışmasının amacı web günlüklerinden çoklu seviye zaman sınırlı bulanık bağıntı kurallarının ve sıralı örüntülerin çıkarılmasıdır. Bu çalışmada bilinen web kullanım madenciliği yöntemlerinden farklı olarak zaman sınırlama kavramı kullanılarak daha etkin kurallar ve örüntüler elde edilmiştir. Bunun yanında site sınıflandırma ağaçları kullanılarak dolaylı yoldan

ilişki kuralları ve sıralı örüntüler elde edilmiştir. Bu sayede doğrudan elde edilemeyecek kural ve örüntülerin dolaylı yoldan elde edilmesi sağlanmıştır.

1.2 Tezde Yapılanlar

Bu tez çalışmasının ikinci bölümünde; veri madenciliği, veriden maden elde etme süreci, bağıntı kurallarında temel kavramlar, minimum destek ve güven değerleri anlatılacaktır, üçüncü bölümde; Web madenciliği, web madenciliğinin sınıflandırılması, web içerik madenciliği, web yapı madenciliği, web kullanım madenciliği ve web verisinin üzerine uygulanan önışlemler verilecektir, dördüncü bölümde; çoklu seviyeli zaman sınırlamalı bulanık bağıntı kuralları ve sıralı örüntüler, önerilen algoritmanın notasyonları ve adımları sunulacaktır, beşinci bölümde; önerilen algoritma ile zaman sınırlı bulanık kural çıkarımı ve sıralı örüntü elde etme süreci, altıncı bölümde; geliştirilen yazılım ve uygulama sonuçları, yedinci bölümde ise; tez çalışması boyunca elde edilen sonuçlar anlatılacaktır.

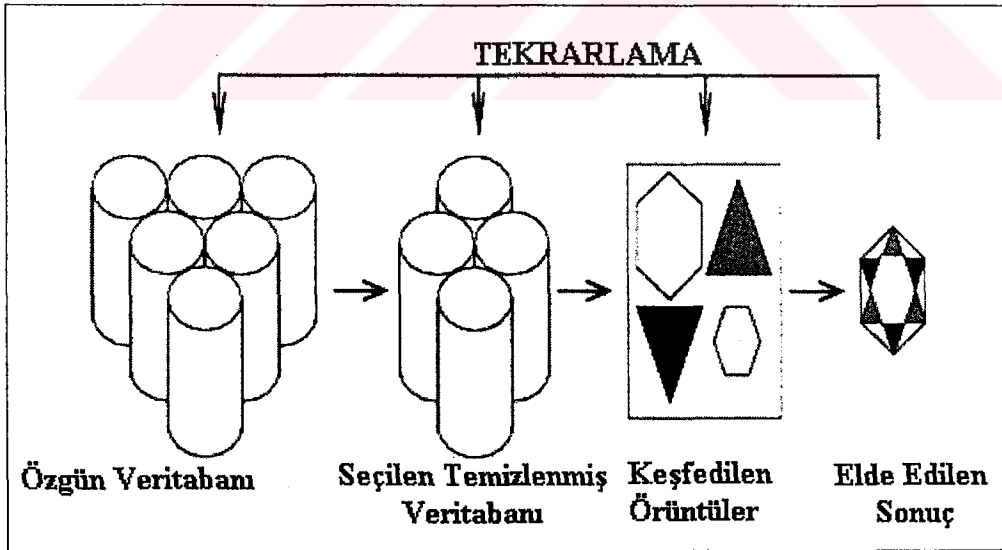


2. VERİ MADENCİLİĞİ

Veri madenciliği, organizasyonların karar aşamaları için yeni bilgiler üreten yada gelecekle ilgili tahminler ve planlar yapmamızı sağlayan, bir dizi teknikler ve anlayışlar bütünü olarak tanımlanmaktadır [6].

Karar aşamalarında çok kritik bazı bilgiler vardır ki, sonuçların etkileri bu bilgilerin doğruluğuyla orantılıdır. Bir çok durumda cevabını tam olarak veremediğimiz sorular doğrultusunda karar verebiliriz. Müşterilerimizin ilgi alanları, bize karşı olan bakış açıları, rakip firmalara olan ilgileri, markalarımıza olan bağlılıkları, gelir düzeyleri gibi bilgiler onlara sağladığımız mal ve hizmetlerin kalitesi üzerinde çok net etkiler yapacaktır. Bu tür bilgiler teorik olarak her ne kadar sistemlerimizde kayıt altında olsa da, kullanılabilir bir şekilde açık ve net cevaplara ulaşabilmemiz mevcut kullanımdaki sistemlerle neredeyse imkansız denecek kadar zordur.

Çok büyük veri yığınları altında saklı olan bu bilgilere ulaşmak için uzun yıllar boyu yapılagelen çalışmaların neticesinde bir dizi metodoloji geliştirilmiştir. Veri madenciliği uzun yıllardır, özellikle batı ülkelerinde üzerinde çalışılan bir konu olmasına rağmen, gerçek hayatta yazılım endüstrisinin son yıllarda üretmiş olduğu ileri teknoloji ürünü yazılımlar ile kullanılmaya başlanmıştır [7].



Şekil 2.1. Veri madenciliği işlemleri

Şekil 2.1, özde veri madenciliği kavramını şematik olarak gösteren bir örnektir. Burada, özgün veritabanı içerisinde ayıklanmış verilerden elde edilen örüntülerin bir araya getirilerek anlamlı bir yapıyı oluşturması gösterilmektedir.

Veri madenciliği Biyomedikal ve DNA veri analizlerinde, finans veri analizlerinde, perakende satış verilerinde, telekomünikasyon endüstrisinde, astronomi ve bir çok alanda uygulanmaktadır [7]. Bu uygulamaların başlıcaları ilgili alanlara göre aşağıda sıralanmıştır:

- Biyomedikal veri madenciliği ve DNA analizi,
- DNA dizisi arasında benzerlik arama ve mukayese. Örneğin, hastalık virüslerin de gen sırası tanımının oynadığı rol,
- İlişki analizi,
- Genetik veri analizleri,
- Finans analizi,
- Kredi dolandırıcılıklarının tespiti,
- Kredi talep değerlendirilmeleri,
- Kullanılan kredilerin türüne göre müşteri özelliklerinin çıkartılması,
- Perakende satışları,
- Satış tahmini,
- Müşteri ürün ilişkisinin kurulması,
- Yeni müşteri kazanma yöntemleri geliştirilmesi,
- Ürün kataloglarının hazırlanması,
- Reyon düzenlemeleri,
- Hastalık tanısı,
- Müşteri satın alma eğiliminin çıkartılması,

Son 20 yıldır, Amerika Birleşik Devletleri'nde çeşitli veri madenciliği algoritmalarının gizli dinlemeden, vergi kaçaklıklarının ortaya çıkartılmasına kadar çeşitli uygulamalarda kullanıldığı bilinmektedir.

Veri madenciliğini istatistiksel bir yöntemler serisi olarak görmek mümkün olabilir. Ancak veri madenciliği, geleneksel istatistikten birkaç yönde farklılık gösterir. Veri madenciliğinde amaç, kolaylıkla mantıksal kurallara yada görsel sunumlara çevrilebilecek nitel modellerin çıkarılmasıdır. Bu bağlamda, veri madenciliği insan merkezlidir ve bazen insan-bilgisayar ara yüzü birleştirilir.

Veri madenciliği konusunda bahsi geçen geniş verideki geniş kelimesi, tek bir iş istasyonunun belleğine sığmayacak kadar büyük veri kümelerini ifade etmektedir. Yüksek

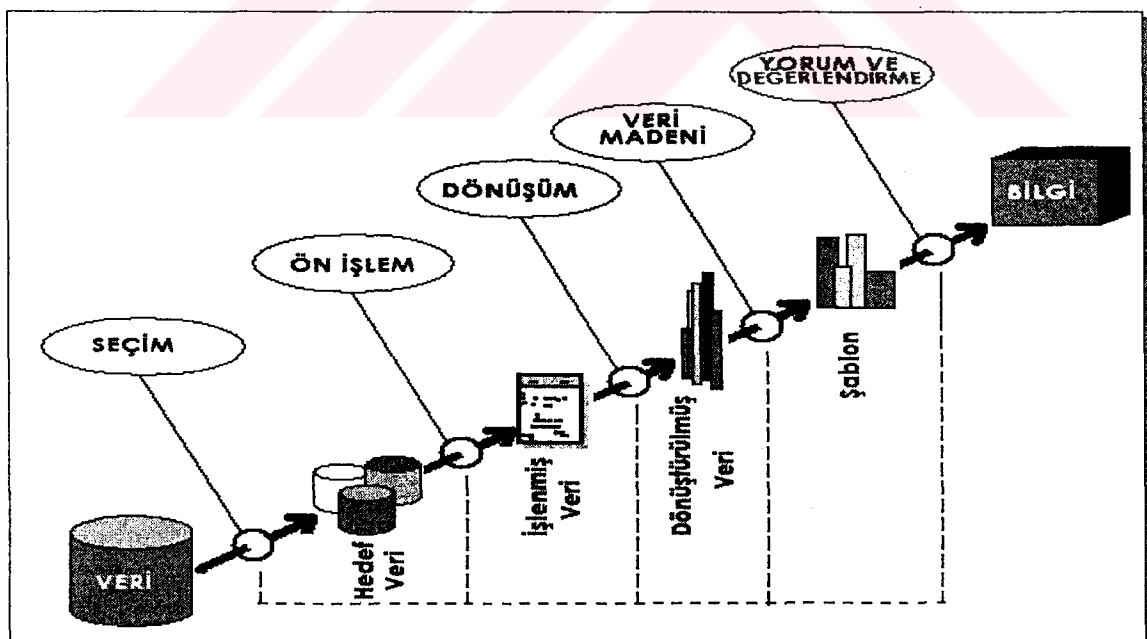
hacimli veri ise, tek bir iş istasyonundaki yada bir grup iş istasyonundaki disklere sığmayacak kadar fazla veri anlamındadır. Dağıtık veri ise, farklı coğrafi konumlarda bulunan verileri anlatır.

Örnek olarak, büyük bir süpermarketin en basit fatura kayıtları incelendiğinde, *traş bıçağı alan müşterilerin %56'sının kalem pil de aldığı ortaya çıkmıştır*. Buna dayanarak firma, traş bıçağı ve pil reyonlarını bir araya getirmek suretiyle kalem pil satışlarını %14 artırmıştır. Bu ve buna benzer örnekler her zaman karşımıza çıkar. Ürünler satışları arasındaki bu ilişkilerin belirlenmesiyle, satış stratejileri değiştirilip kazancın artırılması mümkündür.

2.1. Veriden Maden Elde Etme Süreci

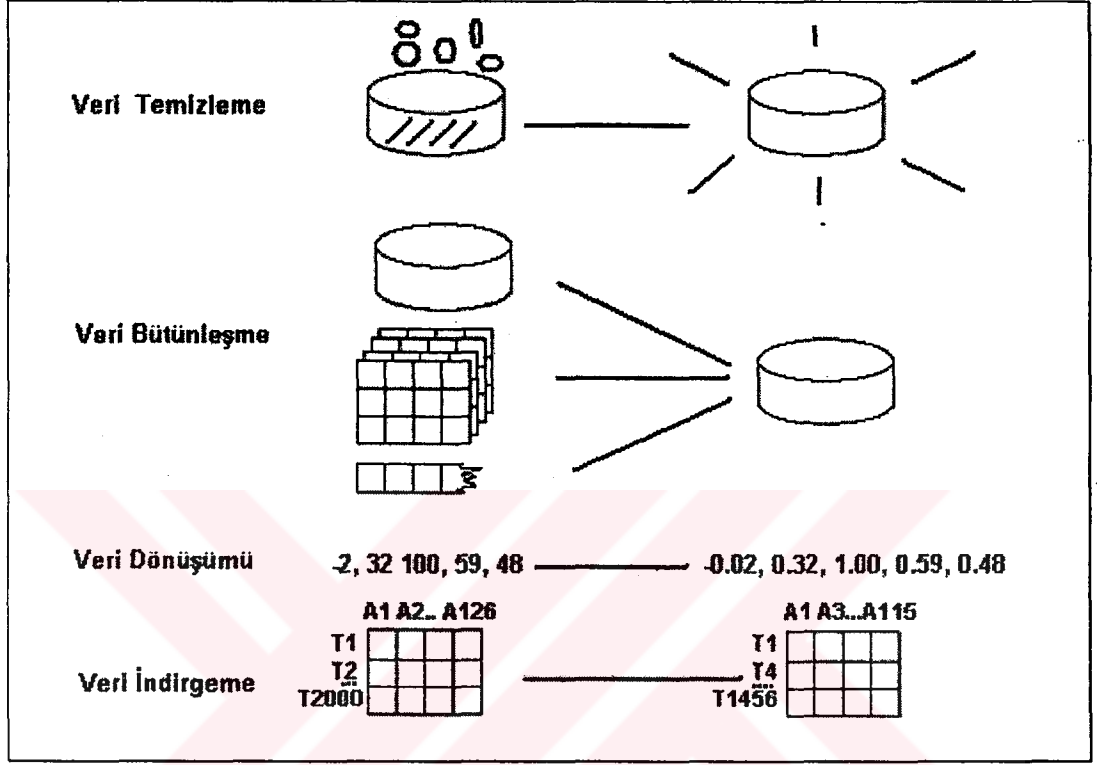
Temelde veri madenciliği, veri birlikteliği sıralaması, kalıp bulma için yazılım tekniklerinin kullanımı ve veri analizi ile ilgilidir. Beklenmedik bir yerde altın bulmak düşüncesi, veri madeninde de daha önce kimsenin fark etmediği yada daha önce veriler arasında keşfedilmeyen ilişkilerin madencilik yazılımı yoluyla elde edilmesi düşüncesiyle paraleldir.

Şekil 2.2' de veri için ön hazırlığın nasıl yapıldığı görülmektedir. Burada verinin istenilmeyen kısımlarının atılması, başka veri tabanından veri birleştirilmesi, verinin istenilen şekilde dönüştürülmesi ve verinin indirgenmesi aşamaları resmedilmektedir.



Sekil 2.2. Verinin hazırlanması.

Şekil 2.2.de değinilen fakat detayları verilmeyen veri ön işlemlerini bu alanın en önemli iki savunucusu Fayyad ve Simoudis [8] tarafından oluşturulan ve Şekil 2.3. de gösterilen veri madenciliği, veri tabanlarında bilgi keşfi yöntemleri aşamalarını detaylı bir şekilde irdelemektedir.



Şekil 2.3. Verinin çıkarım evreleri.

Şekil 2.2’de gösterilen aşamalar aşağıda açıklanmıştır:

Seçim : Verinin hangi kriterlere göre seçileceğini belirten kısımdır. Bir çok veri tabanı temelde birbirleriyle ilgisi olan verilerden oluşmasına rağmen, özelden önemli verilerin belirlenerek seçimlerinin yapılması gerekmektedir. Örneğin, araba sahibi olanlar derken, seçim için araba sahibi olma şartını öne sürmemiz bir seçim kriteridir.

Ön İşlem : Bu aşama, verinin ayıklanması işleminin yapıldığı kısımdır. Gerekli olmadığı düşünülen ve sorguyu yavaşlatabilecek belli bilgiler kaldırılır. Örnek verecek olursak, hamilelik üzerine çalışıldığında hasta cinsiyetini not almak gereksizdir. Bunun yanı sıra verinin bir çok kaynaktan temininden dolayı istemediğimiz yada uygun olmayan şekillerde bulunma ihtimali düşünülerek, belirlenen bir formata göre düzenlenmelidir. Cinsiyet verisi için, E veya K yada 1 veya 0 seçilmesi gibi. Bu durumda veriler arasında ortak bir dil tanımlanmış olacaktır. Böylece sistemimiz hangi başlık altında hangi verinin bulunduğunu kolayca yorumlayabilecektir.

Dönüşüm : Veri sadece transfer edilmez aynı zamanda dönüştürülür. Buna örnek olarak çoğunlukla market araştırmalarında kullanılan müşteri portföyü gibi ekstra bilgilerin mevcut verilere eklenmesini gösterebiliriz. Bu sayede bilgi kullanılabilir ve sevk edilebilir yapıya getirilir.

Veri Madeni : Bu evre ise veri içerisinden örüntülerin elde edilmesiyle ilgilidir.

Yorum ve Değerlendirme: Burada, sistem tarafından tanımlanan örüntüler, insanların karar vermesine yardımcı olmada, önceden haber vermeden ve konuların sınıflanmasında, veritabanının içeriğini özetlemede veya gözlenen olayların açıklamasında daha sonra kullanılabilecek bilgiler şekline dönüştürülür.

2.2. Bağıntı Kuralları ve Sıralı Örüntüler

Bir alışveriş sırasında veya birbirini izleyen alışverişlerde müşterinin hangi mal veya hizmetleri satın almaya eğilimli olduğunun belirlenmesi, müşteriye daha fazla ürün satılmasını sağlama yollarından birisidir. Satın alma eğilimlerinin tanımlanmasını sağlayan bağıntı kuralları ve sıralı örüntüler, pazarlama amaçlı olarak pazar sepeti analizi adı altında veri madenciliğinde yaygın olarak kullanılmaktadır. Bununla birlikte bu teknikler, tıp, finans ve farklı olayların birbiri ile ilişkili olduğunun belirlenmesi sonucunda değerli bilgi kazanımının söz konusu olduğu ortamlarda da önem taşımaktadır.

Bağıntı kuralları aşağıda sunulan örneklerde görüldüğü gibi eş zamanlı olarak gerçekleşen ilişkilerin tanımlanmasında kullanılır[9].

- *Müşteriler bira satın aldığı anda, % 75 ihtimalle patates cipsi de alırlar.*
- *Düşük yağlı peynir ve yağsız yoğurt alan müşteriler, %85 ihtimalle diyet süt de alırlar.*

Sıralı örüntüler ise aşağıda sunulan örneklerde görüldüğü gibi birbirleri ile ilişkisi olan ancak birbirini izleyen dönemlerde gerçekleşen ilişkilerin tanımlanmasında kullanılır[10].

- X ameliyatı yapıldığında, 15 gün içinde %45 ihtimalle Y enfeksiyonu oluşacaktır.
- IMKB endeksi düşerken A hisse senedinin değeri % 15'den fazla artacak olursa, üç iş günü içerisinde B hisse senedinin değeri %60 ihtimalle artacaktır.
- Çekiç satın alan bir müşteri, ilk üç ay içerisinde % 15, bu dönemi izleyen üç ay içerisinde de % 10 ihtimalle çivi satın alacaktır.

2.3. Bağıntı Kurallarında Temel Kavramlar ve Apriori Algoritması

Veri madenciliğinin amacı, kayıtlarda bazı nesne kümeleri ile diğer bazı nesne kümelerinin ilişkilendirilmesi ve nesne kümeleri arasındaki önemli bağıntılarının bulunmasıdır. Bunu yapmak

için Agrawal ve diğ. [11] kayıtlardaki bağıntı kurallarını bulmak için yoğun nesne kümelerinde birkaç madencilik algoritması sunmuşlardır. İlk aşama kayıt verilerinin taranmasıyla nesne kümeleri sayılır ve aday kümeler bulunur. Kayıtlardaki nesne kümelerinin sayısı önceden tanımlı eşik değerden büyükse yeni bir nesne kümesi oluşturulur. Bu nesne kümesi yoğun nesne kümesi olarak adlandırılır. Önce 1-nesne kümesi içeren nesne kümeleri işleme tabi tutulur. Sadece 2-nesne kümesi içeren yoğun nesne kümeleri, 1-nesne kümesi içeren aday nesne kümelerinden oluşturulur. Bu işlemler bütün yoğun nesne kümeleri bulununcaya kadar devam ettirilir. İkinci aşamada ise ilk aşamada bulunan nesne kümelerinden bağıntı kuralları oluşturulur. Bütün olası bağıntı kuralları her yoğun nesne kümesinin kombinasyonları şeklindedir ve hesaplanan güven değeri önceden tanımlı eşik değerden (minimum güven diye adlandırılır) büyük olmalıdır. Bu işlemin sonucunda bağıntı kuralları elde edilir.

Bugüne kadar ikilik değerli kayıtlardan çıkan bağıntı kuralları metotlarına ek olarak, nicel ve kategorik değer içeren kayıtlara ait yöntemler de sunulmuştur [12].

2.4. Örnek Uygulama

Tablo 2.2 de görüleceği gibi Bira ile Çocuk bezi arasında bir kural yazılabilir. Bira satın alan müşteri kayıtları %50 destek ve %66 güven değeriyle çocuk bezi de alırlar.

Tablo 2.1. Örnek bir veritabanı

<i>Kayıt Numarası</i>	<i>Satılan Nesneler</i>
10	Bira,Süt,Çocuk Bezi
20	Bira, Çocuk Bezi
30	Bira , Ekmek
40	Süt, Yağ, Reçel

Tablo 2.2. Örnek veri tabanı yoğun nesne kümesi

<i>Tekrarlanan desenler</i>	<i>Destek</i>
{ Bira }	3 / 4
{ Çocuk Bezi }	2 / 4
{ Süt }	2 / 4
{ Bira, Çocuk Bezi }	2 / 4

2.5. Minimum Destek ve Minimum Güven Değerleri

Destek değeri nesne kümesindeki kurala ait kümenin tekrarlama sıklığıdır. Yukarıdaki örnek için destek değeri { Bira, Çocuk bezi} kümesinin toplam 4 kayıta da birlikte tekrarlanma sıklığı 2 olduğundan 2/4 yani %50 alınır.

Güven değerinin anlamı ise, yine yukarıdaki örnek için { Bira } içeren kayıtlardan kaç tanesi aynı zamanda çocuk bezi de içermektedir şeklinde söylenebilir. Tablo1.1' den de görülebileceği gibi toplam 4 kayıttan 3 tanesinde { Bira } vardır. Yine bu 3 tane { Bira } içeren kayıttan 2 tanesi Çocuk bezi de içermektedir. Güven değeri ise 2 / 3 yani %66 'dır. O zaman yukarıdaki kuralın minimum destek değeri %50, minimum güven değeri de %66 'dır. Geliştirilen uygulamalarda elde edilecek kural sayısı verilen minimum destek değeri ve minimum güven değerleri ters orantılıdır. Bu iki değer çok yüksek alınırsa elde edilecek kural sayısı buna bağlı olarak azalır bunun yanında da kuralın gerçeklik değeri artar.

2.6 Bulanık Küme Kavramı

Bulanık küme kavramı ilk olarak Zadeh tarafından ortaya atılmış ve bilim dünyası tarafından geniş kabul görmüştür. Bulanık küme bilinen klasik kümelerle benzer şekilde iki yöntemle gösterilir. Bunlardan birincisi küme elamanlarının üyelik derecelerine göre sıralanması diğeri ise matematiksel olarak üyelik fonksiyonu tanımlama şeklindedir.

Bulanık kümelerde üyelik dereceleri arasındaki geçiş yumuşak ve sürekli bir şekilde olmaktadır. Öğeler bulanık kümeye kısmi derecede aittir. Bulanık kümelerde klasik kümelerdeki karakteristik fonksiyon yerini üyelik fonksiyona bırakır ve $\mu_A : E \rightarrow [0,1]$ şeklinde gösterilir.

Genel olarak küme üyelerini değerleri ile değişiklik gösteren eğriye *üyelik fonksiyonu* (önem eğrisi) adı verilir. Üyelik fonksiyonu grafiğinde x eksenini üyeleri gösterirken y eksenini de üyelik derecelerini gösterir. A bulanık kümesi $\mu_A : E \rightarrow [0,1]$, A'nın üyelik fonksiyonu ve $\mu_A(x) \in [0,1]$ $x \in E$ 'nin A' daki üyelik derecesi olmak üzere ;

$$A = \{(\mu_A(x), x)\}$$

olarak yazılabilir. Bu durumda E' deki bulanık küme olan A ;

$$A = \{(\mu_A(x), x)\} = \{\mu_A(x)/x\}$$

$$A = \{\mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n\}$$

ve bu da

$$A = \{\sum \mu_A(x_i / x_i)\} \quad \text{olarak gösterilir.}$$

3. WEB MADENCİLİĞİ

3.1 Giriş

Web madenciliği web dokümanları ve servislerinden bilgi çıkarma ve otomatik keşif için veri madenciliği tekniklerini kullanmaktır. Birçok yazara göre web madenciliği terimi ilk kez Etzioni [1] tarafından 1996'da ortaya atılmıştır. Bu bildiride Etzioni Web madenciliğinin veri madenciliği tekniklerini kullanarak World Wide Web' de bulunan dosya ve servislerden otomatik olarak örüntüler bulmak ve öngörülme yen bilgiye ulaşmak olduğunu iddia etmektedir. Araştırmacıların çoğu, çalışmalarında bu tanımlamayı esas almışlardır.

Madria [13] ise web madenciliğini, Web' de bulunan veriden faydalı bilgiye ulaşmaktır şeklinde tanımlar. Araştırmanın bu alanı son zamanlarda e-ticaret'teki ve web' teki elde edilebilir bilgi kaynaklarının çok büyümesi, değişik araştırma komitelerinin ilgilerinden dolayı oldukça geniştir. Etzioni web madenciliğini alt görevlere ayırtırmayı önerir:

1. **Kaynak Bulma:** Tasarlanan web dokümanlarını bulup getirme görevi.
2. **Bilgi Seçimi ve Temizlenmesi:** Bulunan web kaynaklarından belirlenen bilginin temizlenmesi ve otomatik seçimi.
3. **Genelleştirme:** Çoklu sitelerin karşısında onlar kadar iyi kişisel web sayfalarındaki genel örüntülerin otomatik olarak keşfedilmesi.
4. **Analiz:** Madenciliği yapılan örüntülerin yorumu ve/veya onaylanması.
5. **Kaynak bulma:** Elektronik haberleşme, haber grupları, elektronik mektuplaşma, HTML etiketleri, silinerek elde edilen HTML dokümanlarının metin içeriği ve web kaynaklarının elle seçimi gibi web' teki elde edilebilir metin kaynaklarından çevrimdışı veya çevrimiçi verinin bulunması.

Bilgi seçimi ve temizlenmesi adımı bilgi elde etme sürecindeki orijinal verinin bulunup getirilmesindeki transformasyon süreçlerinin herhangi bir çeşitidir. Bu transformasyonlar Birinci Derece Lojik formu, ilişkileri gösterme, eğitim serisindeki bulma fazları gibi arzulanan gösterimi elde etmede amaçlanan bir temizleme, durdurma veya kelimeleri durdurup silme

işlemidir. Genelleştirme, makine öğrenmesi veya veri madenciliği teknikleri geliştirmek için tipik olarak kullanılır. Web karşılıklı etkileşimli bir ortam olduğu için web’ deki bilgi keşfetme süreçlerini veya insanların bunda oynadığı önemli role önem vermemiz gerekir. Bu özellikle analiz aşamasındaki yorum ve/veya onaylama için önemlidir. Böylece, interaktif sorgu-tetikleme bilgi keşfi, otomatik veri-tetikleme bilgi keşfinden daha önemli olur. Fakat insanlar tarafından el ile yapılan bilgi keşfi bunun dışında tutulur. Böylece web madenciliği web verisinden bilgiyi veya önceden bilinmeyen bilgiyi ve potansiyel olarak keşfin tüm süreçlerini sağlar. Bu bölüm dolaylı olarak veritabanlarındaki (KDD-Knowledge Discovery in Database) bilgi keşfinin standart sürecini kapsar.

3.2. Web Madenciliği ve Bilgiyi Yeniden Elde Etme

Web’ de doküman keşfi, bilgi erişimi veya kaynakları talep etmek akıllı bilgi erişimi ile web madenciliği birleştirmeleri, web içerik madenciliğinin bir örneğidir. Gerçekte bilgi erişimi tüm ilişkili dokümanlarının bulunup getirilmesidir. Bilgi erişimi bir koleksiyondaki yararlı dokümanlar için arama ve metin indekslemenin ilk hedefidir. Günümüzde bilgi erişimindeki araştırmalar, modellemeyi, doküman sınıflandırmayı, kategorizasyonu, kullanıcı ara yüzlerini, filtrelemeyi içermektedir. Bilgi erişimi, web madenciliğinin bir örneği olarak düşünülebilen görev indeksleme için kullanılabilen, kategorizasyon veya web doküman şeklindedir. Bununla birlikte veri madenciliği tekniklerinin bütün indeksleme görevlerinde kullanılamadığını da belirtmek gerekir.

3.3. Web Madenciliği ve Bilgi Çıkarımı

Bilgi Çıkarımı, genellikle bir bilgi erişim sisteminin yardımıyla bilginin daha kolay özümsemesi ve analiz edilmesinde dokümanların anlaşılabilir bir hale dönüştürme hedefine odaklanır. Bilgi erişimi yararlı dokümanları seçerken, bilgi çıkarımı dokümanlardan yararlı gerçekleri çıkarır. Bilgi çıkarımı bir dokümanın gösterimi veya yapısıyla ilgilenirken, bilgi erişimi sırasız kelimelerin tam bir toplamı olarak bir dokümandaki metin bilgiyi inceler.

3.4. Web Madenciliği Kategorileri

Bu bölümde her kategorinin detaylı açıklamalar içeren bir görünümünü verilecektir. Madira ve Blockeel [2]’e benzer olarak madenciliği ilgili üç alana kategorize edeceğiz: *web içerik madenciliği*, *web yapı madenciliği*, *web kullanım madenciliği*. Web içerik madenciliği, web

içerik/veri dokümanlarından kullanışlı bilginin keşfini tanımlar. Bununla birlikte, web verileri, verinin çok geniş bir aralığını kapsar. Günümüzde verilerin çoğuna Web’ den ulaşılabilir ve elde edilebilir. Son birkaç yılda hükümet bilgilerinin miktarındaki artış da çok büyük olmuştur. Çoğu şirketlerin işlerini ve servislerini elektronik ortama aktardıkları da bilinmektedir. Şirket veritabanlarının çoğu sonuç olarak önceden sahip olunan sisteme web’ den erişilebilir hale gelmiştir. Böylece müşteriler veya satıcılar direk olarak web tabanlı ara yüzden şirket veritabanlarına erişebilirler. Kısacası web, verinin bir çok çeşitini içermektedir.

Temel olarak web içerik madenciliği metin içerikli, imaj, audio, video, hyper bağ, meta veri gibi verinin birkaç farklı tipini içerir. Son zamanlarda, çoklu veri tipi madenciliğindeki araştırmalar çoklu ortam veri madenciliği olarak adlandırılmaktadır. Web içerik verisi HTML dokümanları gibi yarı yapısal, serbest metinler ve HTML sayfasından üretilen veritabanı veya tablolardaki veri gibi birçok yapısal veri içerir. Fakat, web içerik verisinin çoğu yapısal olmayan metin verisidir. Yapısal olmayan veriye, veri madenciliği tekniklerini uygulama üzerine yapılan araştırmalar metinlerdeki bilgi keşfi (KDT-Knowledge Discovery in Text), yada metin veri madenciliği veya metin madenciliği diye adlandırılır. Bu nedenle web içerik madenciliğinin bir örneği olarak metin madenciliği düşünülebilir. Bir sonraki bölümde ayrıca metin madenciliği ele alınacaktır.

Şekil 3.1. Web madenciliği sınıflandırması



Web yapı madenciliği, web' in temelinde yatan yol yapılarının modelini keşfetmeyi dener. Model, yolların tanımlanması veya tanımlanmaması ile hiper bağların topolojisine dayanır. Bu model web sayfalarını sınıflandırmak için kullanılabilir ve farklı web siteleri arasında ilişki ve benzerlik gibi bilgi üretmek için kullanışlıdır.

Web kullanım madenciliği web sörfçülerinin oturumları ve davranışları tarafından üretilen verinin anlamını göstermek için çalışır. Web içerik ve web yapı madenciliği web' de gerçek veya birincil veriyi kullanırken, web kullanım madenciliği kullanıcılar web ile etkileşim halindeyken kullanıcıların etkileşimlerinden sağlanan ikincil veriyi kullanır. Web kullanım verisi web sunucu erişim kayıtları, Vekil sunucu kayıtları, tarayıcı kayıtları, kullanıcı profilleri, çerezler, fare tıklamaları ve sayfa kaydırmalar ve etkileşim sonuçları olarak diğer verileri içerir. Şekil 3.1 web madenciliği kategorilerinin üzerinde bir ön görünüş verir.

Bununla birlikte yukarıdaki kategoriler arasında çok farkların açık olmadığı da ortadadır. Web içerik madenciliği kullanıcılar tarafından girilen veya sonucu çıkarılan profilleri ve bağları kullanabilir. Kullanıcı profilleri, kullanıcı modelleme uygulamaları veya kişisel yardımlar için de çok fazla kullanılır. Bağlar hakkındaki bilgiyi kullanabilme web yapı madenciliğinin işidir. Ayrıca sunucu tarafından üretilen kayıtlardan ve kullanıcı oturumu sırasında istenen dokümanlardan da, içinden geçilen bağlar çıkarabilir.

3.4.1. Web İçerik Madenciliği

Web içerik madenciliği web kaynaklarından otomatik bilgi arama tekniklerini tanımlar. Verinin farklı tiplerde oluşu ve yapısal olmayışı bu konudaki tekniklere daha karışık, yaklaşımlar kazandırır. Otomatik anahtar kelime arama ötesinde, metinler içindeki bilinen yapıları bazı veri modellerine indirgeme yöntemidir. İki tip içerik madenciliği stratejisi olabilir; metin içeriklerini doğrudan arama yada arama motorları gibi araçların aramalarına yardımcı olmadır [14,15].

Web' de veritabanı teknikleri, web' de bilgi sorgulaması ve yönetilmesi problemi ile ilişkilidir. Bu problemlerin ilişkili olduğu üç sınıfı vardır. Bunlar web' de modelleme ve sorgulama, bilgi çıkarması ve birleştirilmesi, web site yapısı ve yeniden yapılandırılmasıdır. İlk ikisi web içerik madenciliği uygulamaları ile ilişkili olmasına rağmen çalışmaların hepsi web içerik madenciliği içerisinde olmaz. Temel olarak veritabanı görünümü web' de olanaklı olan sorgulama ve bilgi yönetiminin daha iyi olması için, bir web sitesini veritabanı olarak dönüştürmeye veya web sitesinin yapısından sonuç çıkarmaya çalışır. Web içerik madenciliğinin veritabanı görünümü temel olarak web' deki veriden model çıkarmaya ve onları birleştirmeye çalışır.

3.4.2. Web Yapı Madenciliği

Web içerik madenciliğinin web dokümanı içindeki yapıyla ilgilenirken, web yapı madenciliği web'deki hiper bağların yapısıyla ilgilenilir [16-18]. Sosyal ağlar ve ağ analizi ile çıkan ve giren bağların üzerine kurulan sayfaların belirli tipleri keşfedilebilir. Web yapı madenciliği web' in temelindeki linklerin yapısını modellemek için sosyal ağ analizini uygulayarak web' in hiper bağ yapısını kullanır. Web erişim araçlarının çoğu, değerli olabilecek bağlantı verisini göz ardı ederek sadece metin verisine ulaşır, Web yapı madenciliğinin amacı web sitesi ve web sayfası hakkında bağlantı verisine bakarak bilgi üretmektir. Teknik olarak, Web içerik madenciliği dokümanın içeriğine, yapı madenciliği ise dokümanlar arası bağlantılara yoğunlaşır.

3.4.3. Web Kullanım Madenciliği

Web kullanım madenciliği kullanıcı web ile etkileşimde bulunurken kullanıcı davranışları hakkında tahminde bulunabilen tekniklere odaklanır [19-21]. Bu kategoride madenciliği yapılacak veri, etkileşimlerin sonucu olarak web' teki ikincil veridir. Bu veriler çok geniş aralıkta olabilir, fakat veriler; sunucular, vekil sunucular ve web istemcilere dayanan kullanım verisi içinde sınıflandırılır. Web kullanım madenciliği süreci genellikle kullanılan iki yaklaşıma göre hareket eder. İlk yaklaşım bir veri madenciliği tekniği uygulanmadan önce ilişkisel tabloların içerisine web sunucunun kullanım verisinin planını yapmaktır. İkinci yaklaşım özel veri temizleme teknikleri kullanarak doğrudan günlük verisini kullanır. Tipik veri madenciliği teknikleri için bu doğru olduğunda temizleme ve veri kalitesi konuları burada çok önemli olur. Karşılaşılan tipik problemler ise, Vekil sunucular ve ön bellekleme varlığındaki olaylar, sunucu oturumları, benzersiz kullanıcılar arasındaki ayırt etmelerdir.

Bu konudaki çalışmalar Genel Web Kullanım Madenciliği Sistemleri, Site Güncelleme Sistemleri, Sistem İyileştirme ve Kişiselleştirme başlıkları altında toplanabilir. Genel Web Kullanım Madenciliği Sistemleri kullanıcıların genel davranış biçimlerini, bilinen yada önerilen veri madenciliği algoritmalarını, sunucu erişim dosyalarındaki veriye uygulayarak bulmaya çalışır. Site Güncelleştirme Sistemlerinin hedefi ise, site içerik ve yapısında yapılması gereken tadilatları bulmaktır. Sistem iyileştirme üzerine yapılan araştırmalar, web kullanım verisi yardımıyla trafiği etkinleştirmeyi hedefler. Son olarak, kişiselleştirme çalışmaları bireysel taleplere göre değişen siteler oluşturmaya çalışır.

Genelde, tipik veri madenciliği tekniklerinde, veri arzulanan yapı için temizlendikten sonra kullanım verisi için madencilik kullanılabilir. Fakat tipik veri madenciliği tekniklerinin bazı

çeşitleri; karma ilişki kuralları, geleneksel sıralı örüntü ve hiper metin olasılık grameri gibi de kullanılabilir. Web kullanım verisi çizgeler ile de gösterilebilir. Web kullanım madenciliği çoğunlukla hareket şablonu, site topolojisi, kavram hiyerarşileri, sözdizimsel zorlamalar gibi arka plan veya alan bilgisi de kullanır. Web kullanım madenciliği uygulamaları iki ana sınıf altında toplanabilir:

- Bir kullanıcı profilini öğrenme veya uyumlu ara yüzlerde kullanıcı modelleme,
- Kullanıcı hareket örüntülerini öğrenme,

Web kullanıcıları bilgi ihtiyaçlarını ve seçimlerini öğrenebilen tekniklere ilgi duyar, ki bu web içerik madenciliği ile birleştirilmesi olanaklı kullanıcı modelleridir. O halde öğrenilen bilgi kişiselleştirme, sistem geliştirme, site değişikliği, iş zekası ve kullanım karakterizasyonu gibi uygulamalar için kullanılabilir.

3.5. Veri İşleme

Görünümün teknik noktasından bakıldığında, web kullanım madenciliği, geniş web veri depolarının kullanım günlükleri için veri madenciliği tekniklerinin uygulanmasıdır. Sonuçları ise web site tasarımı, web sunucu tasarımı ve bir web sitesinden yönlendirme gibi tasarım görevlerinde kullanılabilir. Fakat, veri madenciliği tekniklerini uygulamadan önce diğer süreçler için gerekli veri soyutlaması için ham veriyi işlemek ve hazırlamak gerekir. Bilgi kaynakları web kullanım günlükleri, web sayfası tanımlayıcıları, web site topolojisi, kullanıcı kayıtları ve anketlerden elde edilebilir. Ön işlemenin üç farklı dönüşüm gerçekleştirdiği düşünülür. Bunlar; kullanım dönüştürme, içerik dönüştürme ve yapı dönüştürmedir.

3.5.1. İçerik Önileme

İçerik önileme kullanım madenciliği tarafından kullanılabilen yapıların içine metin, resim, ve diğer dosyaları dönüştürme işlemidir. Web içeriğinin örüntü keşif algoritmasından giriş veya çıkışı filtrelemek için kullanabildiğini anlamak zor değildir. Cooley ve diğ. [22], sayfa görünümlerinin ön işlemede nasıl önemli rol oynadığını tanımlamıştır.

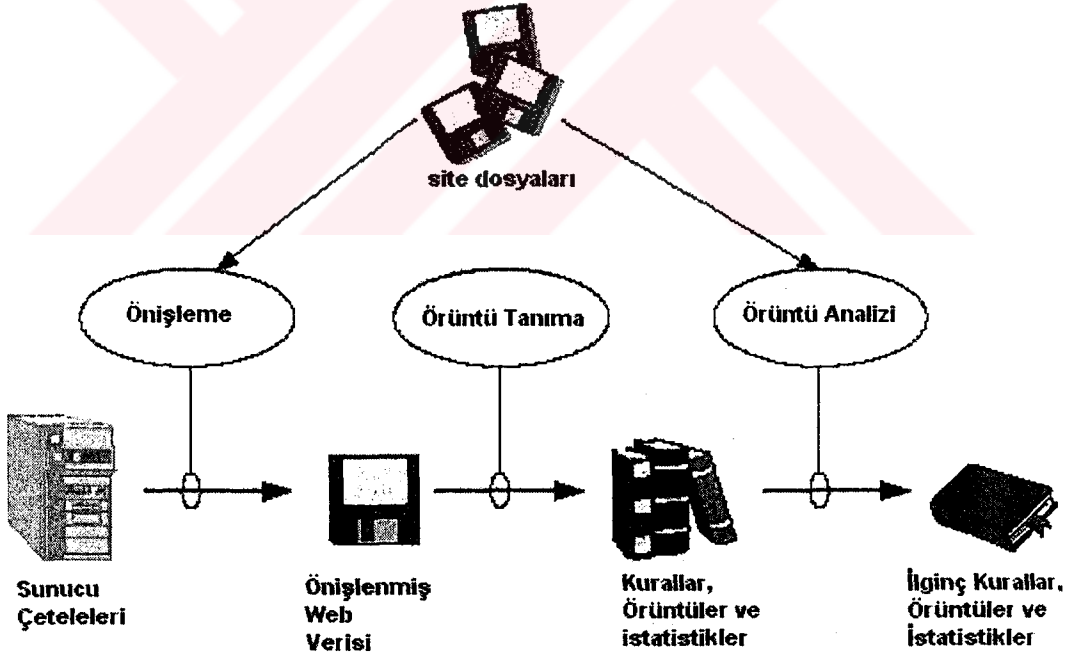
3.5.2. Yapı Önişleme

Bir web sitesinin yapısı, sayfa görünümleri arasındaki hiper bağlar tarafından biçimlendirilir. Yapı ön işleme içerik ön işlemdeki gibi ele alınabilir. Fakat, her sunucu oturumu diğerlerinden farklı site yapısı kurmak zorunda olabilir.

3.5.3. Kullanım Önişleme

Önişlem fazının girişleri; web sunucu günlükleri, kayıt dosyaları, endeks sunucu günlükleri ve önceki bir analizden isteğe bağlı olarak web kullanım istatistikleri olabilir. Çıktılar, kullanıcı oturum dosyaları, geçiş dosyaları, site topolojisi ve sınıflandırmalardır. Analiz sonuçlarıyla ilgili olmayan nesnelerin etkisini yok etmek için veri temizleme teknikleri benimsenmesi gerekir. Kullanım önişleme elde edilebilir, verinin tam olmamasından dolayı web kullanım madenciliğinde en zor görevdir. Yeterli verinin dışında kullanıcıları tanımlamak çok zordur.

Oturum tanımlama da, kullanım önişlemenin bir bölümüdür. Her kullanıcının sayfa erişimlerini ölçme, kişisel oturumlarda, birisinin diğerinden daha fazla siteyi ziyaret etme olasılığını göstermek amacıyla yapılır.



Şekil 3.2. Yüksek seviyeli web kullanım madenciliği süreci

3.6. Örüntü Keşfi

Bu web madenciliğinin anahtar bir bileşenidir. Örüntü keşfi makine öğrenmesi, istatistikler, veri madenciliği ve örüntü tanıma gibi birkaç araştırma alanından teknikler ve algoritmalara yönelir.

3.7. İstatistiksel Analiz

İstatistiksel teknikler, bir web sitesini ziyaret eden kullanıcılar hakkında bilgi çıkarmada en etkili araçlardır. İstatistiksel bilgi çıkarma periyodik olarak web sistem raporları içerir. Çıkarılan rapor, sistem güvenliği, site tanımlama görevi, sistem performansını vermek için kullanışlıdır.

3.8. İlişki Kuralları

İlişki kuralları geçişlerin veritabanındaki bulunan nesneler arasında düzensiz korelasyonlarını keşfetmek için kullanılabilir.

3.9. Kümeleme

Kümeleme benzer karakteristikler ile veri sayfaları veya kullanıcıları birlikte gruplamak için bir tekniktir. Sayfalar veya kullanıcı bilgilerinin gruplaması gelecek market uygulamalarını ve üretimini kolaylaştırır.

3.10. Sınıflandırma

Sınıflandırma ön tanımlı birkaç sınıfın içine bir veri nesnesini yerleştirmek için kullanılan bir tekniktir. Web alanında, web tasarımcısı veya marketçi eğer belirli sınıf veya kategoriler için kullanıcılara ait bir profili saptamak isterse bu tekniği kullanmak zorunda kalacaktır. Karar ağaçları, Bayesian, k-en yakın komşu gibi sınıflandırıcılar, güven destek makinesi gibi eğitici tümevarımsal öğrenme algoritmaları kullanarak gerçekleştirilebilir.

3.11 Örüntü Analizi

Örüntü analizi bütün web kullanım madenciliğinin son aşamasıdır. Bu sürecin hedefi örüntü keşif sürecinin çıkışından ilginç kurallar çıkarmak ve ilişkisiz kural veya örüntüleri yok

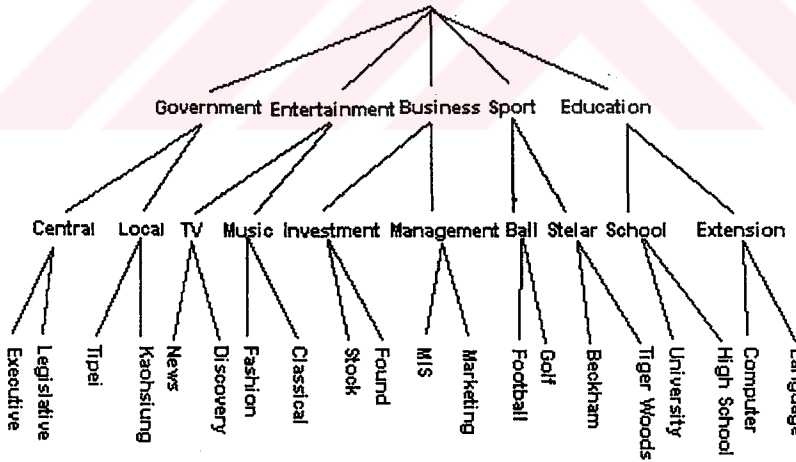
etmektedir. Bazı analiz metodolojileri ve araçlarının yardımı ile yapılabilir. Burada iki çok bilinen yaklaşım vardır. Biri SQL gibi bilgi sorgulama mekanizmasından faydalanırken, diğeri OLAP operasyonlarının kullanılmasını öngörür.



4. ÇOKLU SEVİYELİ BULANIK BAĞINTI KURALLARI VE SIRALI ÖRÜNTÜLERİN ÇIKARILMASI

Bulanık bağinti kural çıkarımı veri madenciliği uygulamalarında bilinen etkili yöntemlerden birisidir[23-25]. Bilindiği gibi ilişki kuralları hangi nesnelerin sıklıkla alındığını veya bir arada kullanıldığını ortaya çıkarmak için kullanılır. Web madenciliği uygulamalarında da bir ziyaretçi tarafından birlikte ziyaret edilen web sayfalarının belirlenmesinde ve birlikteliklerinin ortaya çıkarılmasında kullanılmaktadır. Bilinen algoritmalar bulanık bağinti kuralı çıkarımı için farklı kısıtlama yöntemleri kullanmışlardır. Bu tez çalışmasında, konu web madencilik açısından ele alınmış ve bulanık bağinti kuralları çıkarma sürecinde, birliktelikler için destek değerleri hesaplanırken sayfaların ziyaret zamanları dikkate alınmış ve bir zaman kısıtlaması yöntemi sunulmuştur.

Web madencilik uygulamalarında kullanılan diğer bir yöntem ise, site sınıflandırmaları elde etmek ve kural çıkarım algoritmalarını bu sınıflar üzerinden uygulamaktır. Site sınıflandırması kavramını açıklayacak olursak, bir web sitesi için sayfalar arası bağlar dikkate alınarak elde edilen ağaç yapısı olarak kısa bir tanımlama yapabiliriz. Şekil 4.1 de örnek bir sınıflandırma ağacı görülmektedir.



Şekil 4.1. Örnek site sınıflandırma ağacı

Bu tez çalışmasında Gaziosmanpaşa Üniversitesinin site sınıflandırma ağacı çıkarılmış ve algoritmalar bu sınıflandırmaya ait web günlük verisi üzerine uygulanmıştır.

4.1 Algoritmada Kullanılan Notasyonlar

Algoritmada aşağıdaki notasyonlar kullanılmıştır.

n : Toplam günlük kayıt veri sayısı ;

m : Günlük verisindeki toplam dosya sayısı ;

c : Günlük verisi içerisindeki toplam istemci sayısı ;

n_i : i ' yinci kullanıcının toplam günlük verisi içerisindeki kayıt sayısı ($1 \leq i \leq c$);

D_i : i ' yinci kullanıcının sitedeki sayfalar arasındaki ziyaret sırası. ($1 \leq i \leq c$);

D_{id} : D_i içerisinde d ' yinci kayıt hareketi ($1 \leq d \leq n_i$);

I^g : Günlükteki toplam dosyalar içerisinde g . yinci sayfa , $1 \leq g \leq m$;

R^{gk} : I^g sayfasının k . yinci bulanık bölgesi , $1 \leq k \leq |I^g|$;

$|I^g|$: I^g sayfasının bulanık bölgelerdeki sayısı ;

v_{id}^g : I^g sayfasının D_{id} ziyareti içerisindeki geçen süre;

f_{id}^g : v_{id}^g 'den dönüştürülen bulanık küme;

f_{id}^{gk} : v_{id}^g , Bulanık kümesinin, R^{gk} bulanık bölgesindeki üyelik değeri;

f_i^{kk} : i .yinci ziyaretçinin D_i sıralı ziyaretindeki R^{gk} bulanık bölgesinin üyelik değeri;

$count^{gk}$: R^{gk} bulanık bölgesinin sayısal büyüklüğü;

$\max count^{gk}$: $count^{gk}$ değerleri içerisinde maksimum değer ;

$\max - R^g$: I^g sayfasının maksimum count değerine sahip olduğu fuzz bölgesi ;

α : Önceden tanımlanmış minimum destek değeri ;

λ : Önceden tanımlanmış minimum güven değeri ;

C_r : Her birisi r adet dosya içeren aday nesne kümeleri ;

L_r : Her birisi r adet dosya içeren geniş nesne kümeleri ;

4.2. Algoritmanın Adımları

Giriş: Bir adet sunucu günlük kayıt verisi, bir adet site sınıflandırma ağacı, sınır değerleri tanımlanmış üyelik fonksiyonları, minimum destek değeri (α), minimum güven değeri (λ), ilginç destek eşiği R .

Output: Zaman sınırlı bulanık bağıntı kuralları ve sıralı örüntüler.

Adım 1: Günlük verisinden uzantıları html, java, php, cgi, class, olan sayfalarla birlikte kullanıcıların oturumu kapatma zamanlarını da içeren kayıtlar için günlük verisinden tarih, zaman, istemci-ip ve dosya ismi alanları seçilir. Günlük verisi D ile temsil edilir.

Adım 2: Her sayfa belli bir numara sırasına göre kodlanır ve son seviyeden başlanarak ilk seviyeye kadar her seviye için ilgili rakamlar “*” karakteri ile yer değiştirir. Örnek : 2.1.1 -> 2.1.* -> 2.*.*

Adım 3: Her c_j istemcisi için onun (web sayfası, bekleme süresi) n_j ikililerinden oluşan bir D_j sıralısı oluşturulur. d. ikili D_{jd} olarak adlandırılır.

Adım 4: $k = 1$ olarak atanır ve işlem yapılacak seviyeyi tutar.

Adım 5 : Sayfalar k .yıncı seviyeden başlanarak rakamlar “*” karakteri ile yer değiştirilerek tekrar kodlanır.

Adım 6 : D_{id} içerisinde I^k dosyanın v_{id}^k olarak görülen bekleme süresi f_{id}^k olarak temsil edilen bulanık kümelerle dönüştürülür.

$$\left\{ \frac{f_{id}^{g_1}}{R^{g_1}} + \frac{f_{id}^{g_{12}}}{R^{g_2}} + \dots + \frac{f_{id}^{g_l}}{R^{g_l}} \right\}$$

Başlangıçta tanımlanan üyelik fonksiyonları kullanılarak, I^k , g .yıncı dosya ismi, R^{g_k} , I^k sayfasının k .ıncı bulanık bölgesi, $f_{id}^{g_k}$, v_{id}^k nin R^{g_k} bulanık bölgesindeki üyelik fonksiyon değeri ve l ise I^k ‘nin bulanık bölgelerinin sayısıdır.

Adım 7 : Her R^{g_k} için $f_{id}^{g_k}$ değerleri hesaplanır, her bir D_i sıralısı için ;

$$f_i^{gk} = \text{MAX}_{d=1}^{|D_i|} f_{id}^{gk} \text{ formülü kullanılır. } |D_i|, D_i \text{ içerisindeki ikililerin sayısıdır.}$$

Adım 8 : Her bir R^{g_k} bölgesinin sayısal büyüklüğü hesaplanır.

$$\text{count}^{gk} = \sum_{i=1}^c f_i^{gk} \quad c, \text{ yapılan ziyaretler toplamı.}$$

Adım 9 : Maksimum $count^g$ bulunur.

$max count^g = \underset{k=1}{\overset{l}{MAX}}(count^{gk})$, $1 < g < m$. Bu adım sonunda sayfanın hangi bulanık bölge tarafından temsil edileceği elde edilir ve bundan sonraki çıkartım işlemlerinde de artık bu bölge üzerinde işlem yapılır. Bu bölge $max-R^g$ olarak temsil edilir.

Adım 10 : $max-R^g$ bölgeleri içerisinde $max count^g$ değeri minimum destek değerini geçenler kontrol edilir. Eğer bir bölge minimum destek değerine eşit yada büyük ise onu L_1 kümesine kaydet. $L_1 = \{ max-R^g \mid max count^g \geq \alpha, 1 \leq g \leq m \}$

Adım 11 : Eğer L_1 boş küme ise algorithmadan çıkılır aksi taktirde diğer adıma geçilir.

Adım 12 : $r=1$ olarak atanır. r şu an tutulan sıralı örüntülerin uzunluğunu gösterir.

Adım 13 : L_r 'den aday sıralı C_{r+1} kümesi oluşturulur. L_r ile L_r , $r-1$ benzer olacak şekilde ve aynı sırada bileştirilir.

Adım 14 : Yeni oluşturulan ve s simgesi ile ifade edilen $r+1$ sıralılarına aşağıdaki alt adımlar uygulanır. ($s_1, s_2, \dots, s_{r+1}$)

a) D_i içerisindeki s 'lerin üyelik değerleri (f_i^s) aşağıdaki gibi hesaplanır.

$f_i^s = \min_{k=1}^{r+1} f_i^{s_k}$, Burada dikkat edilecek nokta s_k bölgesinin, s_{k-1} 'dan sonra gelmesidir. Bu durum istemcinin hareket örüntüsündeki "time" alanı dikkate alınarak belirlenir. Zaman kısıtlaması bu noktada devreye girmektedir. Eğer aynı sıralı örüntü birden fazla geçmiş ise ; f_i^s bunlar içerisinde maksimum değere sahip olmalıdır.

b) s için sayısal büyüklük hesaplanır.

$$count^s = \sum_{i=1}^c f_i^s \quad (1)$$

c.) Eğer $count^s$, minimum destek değerine eşit veya büyük ise L_{r+1} nesne kümesine alınır.

Adım 15 : Eğer L_{r+1} boş küme ise diğer adıma geçilir. Aksi takdirde $r = r + 1$ yapılır ve 13 – 15 adımları tekrarlanır.

Adım 16 : Bu adım iki aşamalıdır. Birinci aşama sonucunda belirlenen minimum güven değerini aşan birliktelikler kural olarak çıkarılmış olur. İkinci aşamadakiler ise

a) Bulanık Bağıntı Kuralı Elde Etme

Elde edilen birlikteliklerin güven değerleri aşağıdaki formül ile hesaplanır. Minimum güven değerine (λ) eşit yada büyük değere sahip olan birliktelikler zaman kısıtlamalı bulanık bağıntı kuralı olarak anılır.

$$\text{Güven}^s = \frac{\sum_{i=1}^c f_i^{s_{k,l}}}{\sum_{i=1}^c f_i^{s_k}} \quad (2)$$

Örnek :

(2**.kisa $\rightarrow$ 3**.orta) :

$$\frac{\sum_{i=1}^6 count(2**.kisa \rightarrow 3**.Orta)}{\sum_{i=1}^6 count(2**.kisa)} = \frac{2.56}{4.22} = 0.60 \quad (3)$$

$\lambda = 0,60$ alındı ise

2.** Sayfasında kısa süre kalanlar 0,60 güven değeri ile 3.** sayfasında orta süre ziyarette bulunmuşlardır denilir.

b) Sıralı Örüntüler Elde Etme

Elde edilen sıralılar kümesi içerisinde ata bireyler barındıranlar olmayacak şekilde çıkış olarak verilir.

Geriye kalan örüntüler içerisinde en yakın ata örüntüsü elde edilerek İlginç Destek değeri hesaplanır.

$$I_{sup\ port}(s) = \frac{count_s}{\frac{\prod_{k=1}^{r+1} count_{sk}}{\prod_{k=1}^{r+1} count_{tk}}} * count_t \quad (4) \quad \text{Hesaplanan İlginç Destek değeri, belirlenen}$$

minimum R değerinden büyük yada eşit olanlar çıkış olarak verilir.

5. WEB GÜNLÜĞÜNDEN ZAMAN SINIRLI ÇOKLU SEVİYELİ BULANIK BAĞINTI KURALI VE SIRALI ÖRÜNTÜLERİN ÇIKARILMASI İÇİN ÖNERİLEN YÖNTEM

Bu bölümde, dördüncü bölümde anlatılan algoritma adımları izlenerek bir web sunucusuna ait verilerinden zaman kısıtlayıcı bulanık bağıntı kuralları ve beraberinde sıralı örüntüleri elde etmek için önerilen yöntem anlatılır.

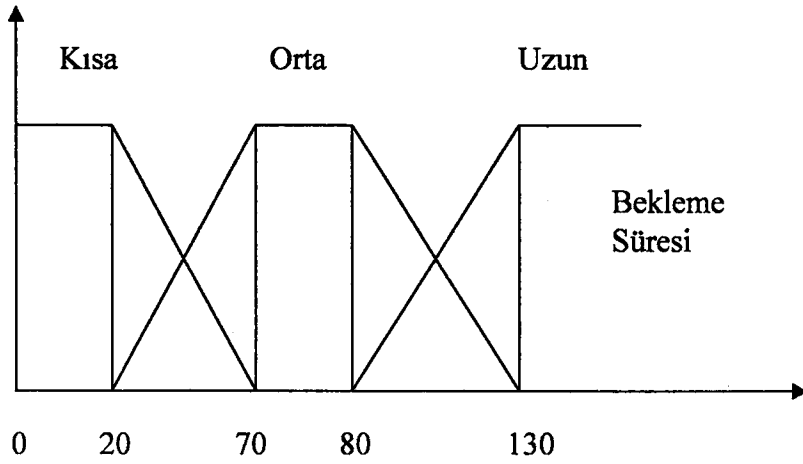
5.1. Önışlemler

Algoritmada kullanılacak günlük verisi Tablo 5.1’ deki yapıdadır. Bilindiği gibi bir web sunucusu, üzerinde barındırdığı web siteleri için sunucu yöneticileri tarafından şekli belirlenen farklı tiplerde günlük dosyaları tutar. Ancak bu günlük dosyaları her zaman Tablo 5.1 ‘de görünen yapıdaki gibi düzenli olmayabilir. Bu sebeple günlük verileri algoritmalarda kullanılmadan önce çeşitli temizleme işlemlerinden geçirilir. Örneğin uzantısı jpg, swf, gif, js, ico, dll, class, jar olan dosyalar bir sayfa istemci tarafından çağırıldığında kendiliğinden yüklenen dosyalar olduğundan hareket örüntüleri elde etmede ayırt edici bir özelliğe sahip değillerdir. Bu nedenle kayıt verisi içerisinde temizlenirler. Uzantısı doc, htm, html, asp, zip , exe olan dosyalar ise örüntü analizleri yapmada ve ilişki kuralları elde etmede önemli olduklarından kayıt verisi içerisinde korunmuşlardır.

Yine algoritmamız için ihtiyaç duyulacak üyelik fonksiyonları Şekil 5.1 ‘de görüldüğü gibidir. İşlemlerde üçgen üyelik fonksiyonu kullanılmıştır. Üyelik fonksiyonlarının taban değerleri sayfalardaki bekleme süreleri tahmini olarak düşünülerek belirlenmiştir. Ziyaretçilerin bekleme süreleri “Kısa”, “Orta”, “Uzun”, olmak üzere üç ayrı bulanık bölgeye ayrılmıştır.

Tablo 5.1 : Örnekte kullanılan web sunucu günlük verisi örneği

Zaman	İstemci IP	Dosya İsmi
05:39:56	140.117.72.1	News.htm
05:40:08	140.117.72.1	Exective.htm
05:40:26	140.117.72.1	University.htm
.....		
05:53:33	140.117.72.6	Closing connection



Şekil 5.1 Üçgen üyelik fonksiyonları

Bunlara ek olarak algoritmada kullanılan site ağacı Şekil 4.1 'de olduğu gibidir. Site ağaçları bize kullanıcıların bir web sitesini ziyaret ettiklerinde hangi yollardan hangi sayfalara ulaştıklarını gösteren basit site haritalarıdır. Site ağacı kullanımı bu tez çalışmasında çoklu seviyeli bulanık bağıntı kuralı ve sıralı örüntüler çıkarılmasında kullanıldığı için önemli bir yere sahiptir.

5.2. Algoritmanın Düzenlenmiş Web Günlüğü Üzerine Uygulanması

Adım 1 : Yukarıda da bahsedildiği gibi .htm,.html,.cgi,.php,.java gibi uzantılara sahip dosyalar kural çıkarmak için seçilirler.

Adım 2 : Site ağacındaki sayfalar Tablo 5.2' de verildiği gibi kodlanırlar. Burada dikkat edilecek nokta hiyerarşinin nasıl sağlandığıdır. Site ağacına bakılacak olursa "*Central.htm*" isimli dosya, "*Government.htm*" isimli dosyadan sonra çağırılmaktadır. Sayfa kodlarına da bakılacak olursa bu durum net olarak ifade edilmiştir.

Adım 3 : Tablo 5.1' de bir parçası görülen kayıt verisi bu adımda Tablo 5.3' de olduğu gibi yeniden oluşturulur. Kullanıcılar IP bilgilerine göre gruplanarak yaptıkları ziyaretler (Sayfa_Kod, Bekleme Süresi) ikilileri şeklinde ifade edilir. "*timestamp*" olarak isimlendirilen alan kullanıcıların sayfalara girme sırasını göstermektedir.

Adım 4 : k= 1 olarak atanır ve üzerinde işlem yapılacak seviyeyi belirlemek için kullanılır.

Tablo 5.2. Dosya isimlerinin site ağacına göre kodlanması

Kod	Dosya İsmi	Kod	Dosya İsmi
111	Executive.htm	1**	Government.htm
211	News.htm	2**	Entertainment.htm
212	Discoversy.htm	3**	Business.htm
313	Stock.htm	4**	Sport.htm
321	Found.htm	5**	Education.htm
411	Football.htm	11*	Central.htm
412	Golf.htm	12*	Local.htm
421	Tigerwoods.htm	21*	TV.htm
511	University.htm	22*	Music.htm
512	Highschool.htm	31*	Investment.htm
		32*	Management.htm
		41*	Ball.htm
		42*	Stellar.htm
		51*	School.htm
		52*	Extension.htm

Tablo 5.3. Kullanıcıların ziyaret ettikleri sayfa bekleme süreleri ile zaman damgası değerleri

C-ID	Ziyaret Edilen Sayfalar	Zaman Damgası
1	(211,30),(511,42),(412,98),(313,91)	0 5 10 25
2	(412,62),(211,31),(421,102)	0 15 25 70
3	(111,92),(411,89)	0 65
4	(412,70),(211,29),(512,118)	0 25 45
5	(412,75),(211,29),(313,74)	0 20 35
6	(421,80),(313,63),(511,72),(212,17)	0 55 89 91 117

Adım 5 : Her sayfa Tablo 5.4’ de olduğu gibi rakamlar “*” karakteri ile yer değiştirecek şekilde yeniden kodlanır.

Adım 6 : Her işlem sırasında ziyaret edilen sayfaların bekleme süreleri daha önce belirlenen üyelik fonksiyonları sayesinde bulanık kümelerle dönüştürülür. Örneğin 2** sayfasının bekleme süresi olan 30 değeri (0,8/ kısa + 0,2/ orta + 0,0 / uzun) şeklinde bulanık kümelerle ifade edilmiştir.

Tablo 5.4. Sayfaların rakamlar yerine “*” karakteri gelecek şekilde yeniden kodlanması

C-ID	Ziyaret Edilen Sayfalar	Zaman Damgası
1	(211,30),(21*,30),(2**,30)	0
	(511,42),(51*,42),(5**,42)	5
	(412,98),(41*,98),(4**,98)	10
	(312,91),(31*,91),(3**,91)	25
2	(412,62),(41*,62),(4**,62)	0
	(211,31),(21*,31),(2**,31)	15
	(421,102),(42*,102),(4**,102)	25
	(311,25),(31*,25),(3**,25)	55
3	(111,92),(11*,92),(1**,92)	0
	(411,89),(41*,89),(4**,89)	65
4	(412,70),(41*,70),(4**,70)	0
	(211,29),(21*,31),(2**,31)	25
	(512,118),(51*,118),(5**,118)	45
5	(412,75),(41*,75),(4**,75)	0
	(211,29),(21*,29),(2**,29)	20
	(312,74),(31*,74),(3**,74)	35
6	(421,80),(42*,80),(4**,80)	0
	(311,65),(31*,65),(3**,65)	55
	(511,72),(51*,72),(5**,72)	89
	(212,17),(21*,17),(2**,17)	117

Adım 7 : Her bulanık bölgenin üyelik fonksiyonu hesaplanır. Örneğin, 2 no’ lu ziyaretçinin 4**.orta sayfası için üyelik fonksiyonu $\text{Max}(0.8,0.0,0.6) = 0.8$ olarak alınmıştır. Buradaki özel durum bu ziyaretçi aynı oturumda ancak farklı zamanlarda iki defa aynı sayfayı ziyaret etmiştir. Bu sebeple bu ziyaretleri içerisinde en büyük üyelik değeri dikkate alınmıştır. Diğer bölgelerin üyelik fonksiyonları da benzer şekilde hesaplanır.

Adım 8 : Her bölge için sayısal büyüklük değerleri ölçülür.

Adım 9 : Her bir nesneye ait üç bulanık bölge arasında en büyük değere sahip olan bulunur ve bundan sonraki çıkarım işlemlerinde artık bu bölge kullanılır. Örneğin 2** nesnesi için, 4.22 - Kısa, 0.78- Orta, 0.00-Uzun değerleri gözönüne alındığında 2** için artık Kısa bölgesinin kullanılacağı anlaşılmaktadır.

Tablo 5.5. Bekleme sürelerinin bulanık kümelere dönüştürülmesi

Sayfa	Bölge	CID 1	CID 2	CID 3	CID 4	CID 5	CID 6
2**	Kısa	0,8	0,78	0,0	0,82	0,82	1
	Orta	0,2	0,22	0,0	0,18	0,18	0,00
	Uzun	0,0	0,0	0,0	0,00	0,00	0,00
21*	Kısa	0,8	0,78	0,0	0,82	0,82	1
	Orta	0,2	0,22	0,0	0,18	0,18	0,00
	Uzun	0,0	0,0	0,0	0,00	0,00	0,00
22*	Kısa	-	-	-	-	-	-
	Orta	-	-	-	-	-	-
	Uzun	-	-	-	-	-	-
211	Kısa	0,80	0,78	0,00	0,82	0,82	0,00
	Orta	0,20	0,22	0,00	0,18	0,18	0,00
	Uzun	0,00	0,00	0,00	0,00	0,00	0,00
212	Kısa	-	-	-	-	-	1
	Orta	-	-	-	-	-	0,00
	Uzun	-	-	-	-	-	0,00
221	Kısa	-	-	-	-	-	-
	Orta	-	-	-	-	-	-
	Uzun	-	-	-	-	-	-

Adım 10 : Adım 9’ da seçilen bölgelerin count değerleri belirlenen minimum destek değeri ile karşılaştırılır ve büyük olanlar L_1 olarak adlandırılan kümede tutulurlar. $\alpha = 2$ olarak alındığında 2**.Kısa, 21*. Kısa, 211. Kısa, 3**.Orta, 31*. Orta, 4**. Orta, 41*. Orta ve 412. Orta nesneleri elde edilmiştir. Elde edilen yoğun nesne kümeleri Tablo 5.8’ de görülmektedir.

Adım 11 : $r=1$ olarak atanır ve L_r ‘den C_{r+1} kümesi elde edilir. Algoritmada söylendiği gibi C_{r+1} kümesi, L_r ile L_r , $r-1$ benzerlik ve aynı sıra göz önünde tutularak elde edilir. Farklı permütasyonlar farklı nesneler anlamına gelir.

Örnek : C_2 aday nesne kümesi:

(2**.Kısa,2**.Kısa), (2**.Kısa, 3**.Orta), (3**.Orta, 2**.Kısa), (3**.Orta, 4**.Orta),...(4**.Orta, 3**.Orta),(4**.Orta,4**.Orta)

Tablo 5.6. Bekleme sürelerinin bulanık kümelere dönüştürülmüş halleri

Sayfa	Bölge	CID 1	CID 2	CID 3	CID 4	CID 5	CID 6
3**	Kısa	0,00	-	-	-	0,00	0,10
	Orta	0,78	-	-	-	1,00	0,90
	Uzun	0,22	-	-	-	0,00	0,00
31*	Kısa	0,00	-	-	-	0,00	0,10
	Orta	0,78	-	-	-	1,00	0,90
	Uzun	0,22	-	-	-	0,00	0,00
32*	Kısa	-	-	-	-	-	-
	Orta	-	-	-	-	-	-
	Uzun	-	-	-	-	-	-
311	Kısa	-	-	-	-	-	0,10
	Orta	-	-	-	-	-	0,90
	Uzun	-	-	-	-	-	0,00
312	Kısa	0,00	-	-	-	0,00	-
	Orta	0,78	-	-	-	1,00	-
	Uzun	0,22	-	-	-	0,00	-
321	Kısa	-	-	-	-	-	-
	Orta	-	-	-	-	-	-
	Uzun	-	-	-	-	-	-
4**	Kısa	0,00	0,16	0,00	0,00	0,00	0,00
	Orta	0,64	0,84	0,82	1,00	1,00	1,00
	Uzun	0,36	0,00	0,18	0,00	0,00	0,00
41*	Kısa	0,00	0,16	0,00	0,00	0,00	-
	Orta	0,64	0,84	0,82	1,00	1,00	-
	Uzun	0,36	0,00	0,18	0,00	0,00	-
42*	Kısa	-	0,00	-	-	-	0,00
	Orta	-	0,56	-	-	-	1,00
	Uzun	-	0,44	-	-	-	0,00
411	Kısa	-	-	0,00	-	-	-
	Orta	-	-	0,82	-	-	-
	Uzun	-	-	0,18	-	-	-
412	Kısa	0,00	0,16	-	0,00	0,00	-
	Orta	0,64	0,84	-	1,00	1,00	-
	Uzun	0,36	0,00	-	0,00	0,00	-
421	Kısa	-	0,00	-	-	-	0,00
	Orta	-	0,56	-	-	-	1,00
	Uzun	-	0,44	-	-	-	0,00

Tablo 5.7. Her bölgenin sayısal büyüklük değerleri

Sayfa	Count	Sayfa	Count	Sayfa	Count	Sayfa	Count
2**.Kısa	4,22	3**. Kısa	0,1	4**. Kısa	0,16	5**. Kısa	0,56
2**.Orta	0,78	3**. Orta	2,68	4**. Orta	5,3	5**. Orta	1,68
2**.Uzun	0	3**. Uzun	0,22	4**. Uzun	0,54	5**. Uzun	0,76
21*. Kısa	4,22	31*. Kısa	0,1	41*. Kısa	0,16	51*. Kısa	0,56
21*. Orta	0,78	31*. Orta	2,68	41*. Orta	4,3	51*. Orta	1,68
21*.Uzun	0	321*.Uzun	0,22	421*.Uzun	0,54	521*.Uzun	0,76
22*. Kısa	0	32*. Kısa	0	42*. Kısa	0	52*. Kısa	0
22*. Orta	0	32*. Orta	0	42*. Orta	1,56	52*. Orta	0
22*. Uzun	0	32*. Uzun	0	42*. Uzun	0,44	52*. Uzun	0
211. Kısa	3,22	311. Kısa	0,1	411. Kısa	0	511. Kısa	0,56
211. Orta	0,78	311. Orta	0,9	411. Orta	0,82	511. Orta	1,44
211. Uzun	0	311. Uzun	0	411. Uzun	0,18	511. Uzun	0
212. Kısa	1	312. Kısa	0	412. Kısa	0,16	512. Kısa	0
212. Orta	0	312. Orta	1,78	412. Orta	3,48	512. Orta	0,24
212. Uzun	0	312.Uzun	0,22	412. Uzun	0,36	512. Uzun	0,76

Tablo 5.8. L₁ yoğun nesne kütmesi

Sayfa	Count
2**.Kısa	4,22
21*.Kısa	4,22
211.Kısa	3,22
3**.Orta	2,68
31*.Orta	2,68
4**.Orta	5,3
41*.Orta	4,3
412.Orta	3,48

Tablo 5.9. C₂ aday nesne kümesi

	2** Kısa	3** Orta	4** Orta	21* Kısa	31* Kısa	41* Kısa	211 Kısa	412 Orta
2**.Kısa	1	2	3	4	5	6	7	8
3**.Orta	9	10	11	12	13	14	15	16
4**.Orta	17	18	19	20	21	22	23	24
21*.Kısa	25	26	27	28	29	30	31	32
31*.Kısa	33	34	35	36	37	38	39	40
41*.Kısa	41	42	43	44	45	46	47	48
211.Kısa	49	50	51	52	53	54	55	56
412.Orta	57	58	59	60	61	62	63	64

Adım 12 : Bu adımda elde edilen yeni kümeye aşağıdaki alt adımlar uygulanır.

a) İkili nesne kümesindeki elemanların üyelik fonksiyonları hesaplanır. Bu hesaplama işlemi yapılırken minimum operatörü kullanılır. (2**.Kısa,3**.Orta) ikilisi ele alınırsa dördüncü ziyaret için $\text{Min}(0.8,0.78)=0.78$ olarak elde edilir. Fakat 6 nolu ziyaretçinin aynı ikilisine göz atacak olursak, normalde minimum fonksiyonu sayesinde reel bir değer elde etmemiz gerekirken algoritmamız gereği bu ziyaretçi için üyelik değeri 0.00 olarak alınır. Çünkü bu ziyaretçi 2** sayfasına 3** dan sonra girmiştir. Algoritmada sıraya dikkat edilerek değer hesaplanmaktadır. Aksi taktirde değer olarak sıfır atanmaktadır.

b) Tablo 5.10' da (2**.Kısa,3**.Orta) ikilisine ait üyelik değerleri görülmektedir. Aynı işlemler diğer ikili nesneler için de uygulanır ve tüm hareketlerdeki değerleri hesaplanır.

c) C₂ içerisindeki her nesne için sayısal büyüklük hesaplanır ve Tablo 5.11 deki değerler elde edildi. Bu adım sonucunda (2**.Kısa,3**.Orta), (4**.Orta, 2**.Kısa), (4**.Orta, 3**.Orta), (41*.Orta,31*.Orta), (4**.Orta, 21*.Kısa), (4**.Orta,31*.Orta), (21*.Kısa, 3**.Orta), (41*.Orta, 3**.Orta), (211.Kısa, 3**.Orta), (412.Orta, 3**.Orta) ve (412.Orta, 31*.Orta) ikili nesneleri minimum destek değerinden yüksek değere sahip olduklarından L₂ yoğun nesne kümesi içerisinde dahil edilirler.

Tablo 5.10. (2**.Kısa,3**.Orta) ikili nesne kümesi için sonuçlar

C-ID	Üyelik Değeri
1	0.78
2	0.00
3	0.00
4	0.00
5	0.82
6	0.00

Tablo 5.11. C₂ içerisindeki her nesne için sayısal büyüklük değerleri

No	Cid-1	Cid-2	Cid-3	Cid-4	Cid-5	Cid-6	Toplam Count
1							0.0
2	0.78	0.98			0.82		2.6
3	0.6	0.6					1.2
4				0.8			0.8
5	0.8				0.8		1.6
6	0.6						0.6
...							
64							0.9

Adım 13 : Bu uygulamada, 3'lü nesneler yeterli minimum destek değerinin üzerine çıkamadıkları için algoritmaya 2'li nesne kümesi üzerinden devam edildi.

Adım 14 :

- a) (2**.Kısa → 3**.Orta), (4**.Orta → 2**.Kısa),
(4**.Orta → 3**.Orta), (41*.Orta → 31*.Orta),
(4**.Orta → 21*.Kısa), (4**.Orta → 31*.Orta),
(21*.Kısa → 3**.Orta), (41*.Orta → 3**.Orta),
(211.Kısa → 3**.Orta), (412.Orta → 3**.Orta),
(412.Orta → 31*.Orta)

Adım 13 'de elde edilen kurallara ait güven değerleri hesaplanarak aşağıdaki kurallar gerçek kurallar olarak ortaya çıkarılmıştır.

$$\frac{\sum_{i=1}^6 \text{count}(2^{**}.Kisa \rightarrow 3^{**}.Orta)}{\sum_{i=1}^6 \text{count}(2^{**}.Kisa)} = \frac{2.56}{4.22} = 0.60$$

$\lambda = 0,60$ olmak koşulu ile aşağıdaki kurallar elde edilmiştir.

($2^{**}.Kisa \rightarrow 3^{**}.Orta$), ($41^{*}.Orta \rightarrow 3^{**}.Orta$), ($412.Orta \rightarrow 3^{**}.Orta$),
($211.Kisa \rightarrow 3^{**}.Orta$)

b.) Sıralı örüntüler elde edilirken, ata bireyler barındıranlar üzerinde algoritma gereği aşağıdaki işlem yapılır.

($2^{**}.Kisa \rightarrow 3^{**}.Orta$), ($4^{**}.Orta \rightarrow 2^{**}.Kisa$),
($4^{**}.Orta \rightarrow 3^{**}.Orta$), ($41^{*}.Orta \rightarrow 31^{*}.Orta$),
($4^{**}.Orta \rightarrow 21^{*}.Kisa$), ($4^{**}.Orta \rightarrow 31^{*}.Orta$),
($21^{*}.Kisa \rightarrow 3^{**}.Orta$), ($41^{*}.Orta \rightarrow 3^{**}.Orta$),
($211.Kisa \rightarrow 3^{**}.Orta$), ($412.Orta \rightarrow 3^{**}.Orta$),
($412.Orta \rightarrow 31^{*}.Orta$)

Yukarıdaki örüntülere dikkat edildiğinde ($412.Orta \rightarrow 31^{*}.Orta$) örüntüsünün ata bireye sahip olduğu görülmektedir ($41^{*}.Orta \rightarrow 3^{**}.Orta$). Bu durumda ilginç destek hesaplaması yapılarak bu örüntünün gerçekten gerekli bir örüntü olup olmadığına karar verilir.

$$\begin{aligned} I_{\text{sup port}}(412.Orta \rightarrow 31^{*}.Orta) \\ &= \frac{(412.Orta \cap 31^{*}.Orta)}{\frac{412.Orta}{41^{*}.Orta} \times \frac{31^{*}.Orta}{3^{**}.Orta} \times (412.Orta \cap 31^{*}.Orta)} \\ &= 1.25 \end{aligned}$$

olarak elde edildi. İlginç destek için verilen minimum destek değeri 1.5 olduğu için ilginç bir örüntü olarak değer kazanmadı.

Sonuç olarak elde edilen sıralı örüntüler listesi aşağıdaki gibidir.

($2^{**}.Kisa \rightarrow 3^{**}.Orta$), ($4^{**}.Orta \rightarrow 2^{**}.Kisa$), ($4^{**}.Orta \rightarrow 3^{**}.Orta$),
($41^{*}.Orta \rightarrow 31^{*}.Orta$), ($4^{**}.Orta \rightarrow 21^{*}.Kisa$), ($4^{**}.Orta \rightarrow 31^{*}.Orta$),
($21^{*}.Kisa \rightarrow 3^{**}.Orta$), ($41^{*}.Orta \rightarrow 3^{**}.Orta$), ($211.Kisa \rightarrow 3^{**}.Orta$)
($412.Orta \rightarrow 3^{**}.Orta$)

6. GELİŞTİRİLEN YAZILIM VE UYGULAMA SONUÇLARI

6.1. Giriş

Bu tez çalışmasında karşılaşılan en büyük problemlerden bir tanesi uygulamada kullanılmak üzere gerçek bir web günlük verisinin temin edilmesidir. İnternet ortamında kullanıma hazır bir çok örnek veri olmasına karşın uygulamamızın gerçek verilerle test edilmesinin daha yararlı olacağı düşünülerek günlük verisini kullanmak istediğimiz birkaç firma ile diyaloga geçildi. Bu firmalardan bir tanesi www.sciencedirect.com idi. Akademisyenlerin sıklıkla ziyaret ettiği bu makale arama sitesinin verileri uygulamanın sonuçları açısından daha ilgi çekici olacaktı ancak firma yetkilileri verilerini güvenlik kriterlerine aykırı bularak vermek istemediler. Benzer bir durumda etkin bir alışveriş sitesi olan www.bascuda.com firmasıyla yaşandı. Yine yapılan görüşmeler sonucunda firma günlük verilerini güvenliği gerekçe göstererek vermek istemedi. Bu sebeple uygulamada, elde edebildiğimiz günlük verilerinden bir tanesi olan www.gop.edu.tr site verileri kullanıldı.

Yapılan uygulama dahilinde Bölüm 4’ de anlatılan algoritma ışığında bir yazılım geliştirildi. Yazılım geliştirme ortamı olarak Delphi 5.0 programlama dili ve Microsoft SQL Sunucu 2000 veri tabanı yönetim sistemi kullanıldı. Geliştirilen yazılımın sonuçları ise Intel Pentium – 4, 2 Ghz işlemcili, 256 Megabyte hafızalı ve üzerinde Windows XP Professional yüklü bir bilgisayarda elde edildi.

6.2. Web Günlük Verisinin Düzenlenmesi ve Uygulama Sonuçları

Web günlük verileri, temel bir veritabanı verisinden daha karmaşık bir yapıya sahiptir. Uygulamada kullanılan G.O.P üniversitesinin web günlük verisi de bu yapıda olduğu için bazı gereksiz bilgilerin temizlenmesi ve kullanılabilir hale gelmesi için çeşitli önlemlerden geçirildi. Microsoft İnternet Information Sunucu tarafından tutulan bir günlük verisinin genel yapısı Şekil 6.1’de görüldüğü gibi, tarih, zaman, sunucu ip adresi, istemci ip adresi, bağlantı yöntemi, ziyaret edilen sayfanın URL bilgisi vb. gibi alanlar içermektedir. Sitenin bir günlük verisi olan verimiz 20277 kayıt içermektedir. İlk bakışta karmaşık gibi görünen bu yapı veri temizleme işlemlerinden geçirildikten sonra uygulamada kullanılabilir hale getirilmiştir.

Web günlük verileri sunucular tarafından metin dosya formatında tutulmaktadır. Veriler üzerinde daha hızlı ve daha kolay bir temizleme işlemi yapabilmek için günlük verisinin

tutulduğu dosya,SQL Sunucu programı aracılığıyla yine SQL Sunucu' a bir veri tablosu olarak ak tarılmış ve bu veri tablosundan çeşitli SQL sorguları ile istenmeyen veriler temizlenmiştir. Örneğin uzantısı .jpg, dll, gif, png, jar vs. olan dosyalar silinmiştir. Yine sitenin özelliğinden dolayı her gelen sayfa ile birlikte yüklenen banner.html ve *.bar gibi dosyalar günlük verisinden SQL sorguları yardımı ile çıkarılmıştır.

ex041227.log - Not Defteri

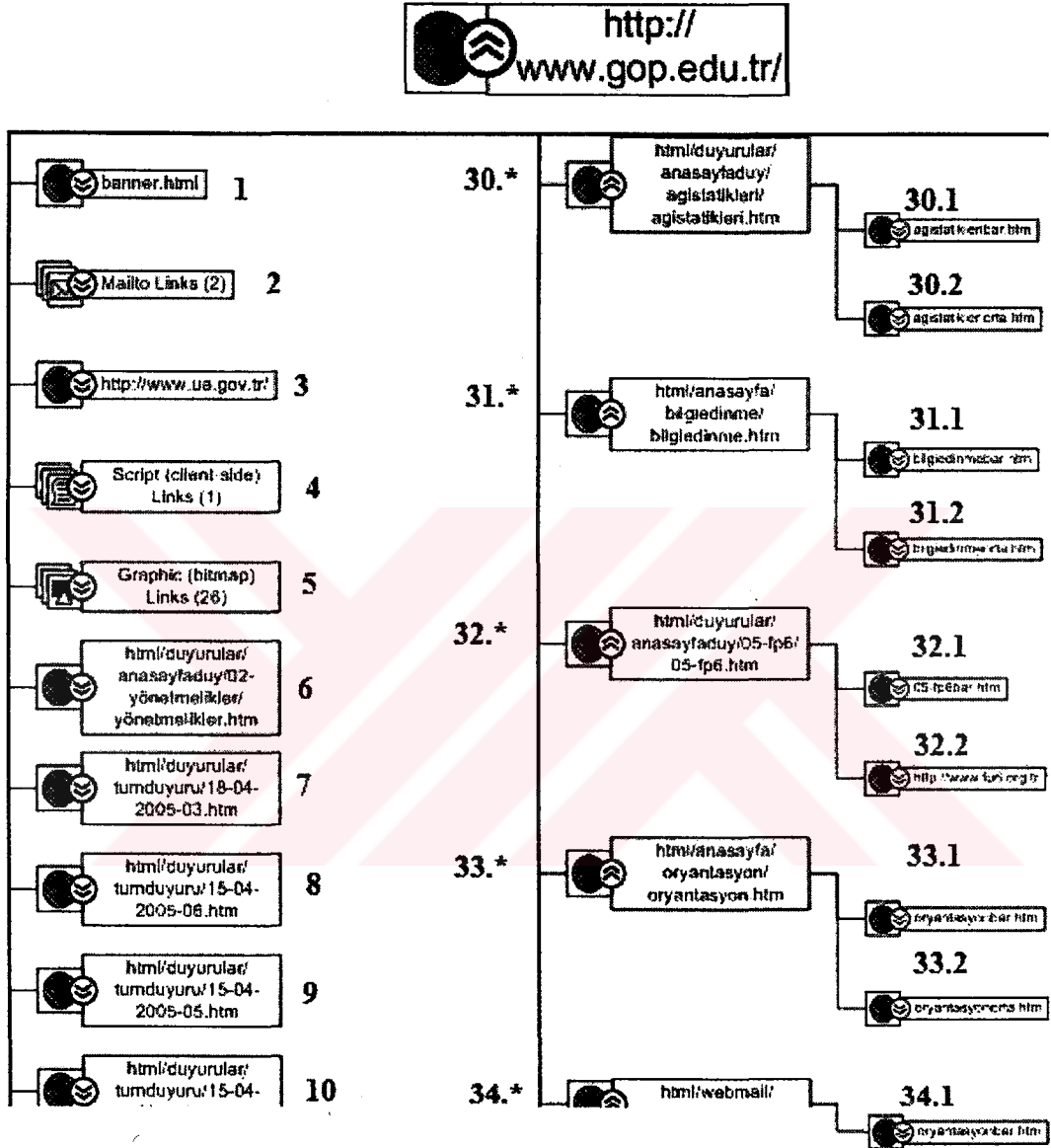
date	time	s-ip	cs-method	cs-uri-stem	cs-uri-query	s-port	cs-username	c-ip	cs(User-Agent)	s
2004-12-27	00:10:57	193.140.182.2	GET	/html/yonetim/yonetimbar.htm	- 80 -	82.152.52.3	Mozi			
2004-12-27	00:10:59	193.140.182.2	GET	/html/yonetim/yonetimorta.htm	- 80 -	82.152.52.3	Moz			
2004-12-27	00:11:01	193.140.182.2	GET	/html/tanitim/tanitimbar.htm	- 80 -	82.152.52.3	Mozi			
2004-12-27	00:11:01	193.140.182.2	GET	/html/tanitim/tanitimorta.htm	- 80 -	82.152.52.3	Moz			
2004-12-27	00:11:45	193.140.182.2	GET	/html/duyurular/anasayfaduy/05-fp6/05-fp6.htm	- 80 -					
2004-12-27	00:12:06	193.140.182.2	GET	/html/akademik/akademikbar.htm	- 80 -	82.152.52.3	Mo			
2004-12-27	00:12:06	193.140.182.2	GET	/html/interaktif/interaktifbar.htm	- 80 -	82.152.52.3				
2004-12-27	00:12:08	193.140.182.2	GET	/html/linkler/linklerbar.htm	- 80 -	82.152.52.3	Mozi			
2004-12-27	00:12:08	193.140.182.2	GET	/html/interaktif/interaktiforta.htm	- 80 -	82.152.52.3				
2004-12-27	00:12:10	193.140.182.2	GET	/html/akademik/akademikorta.htm	- 80 -	82.152.52.3	M			
2004-12-27	00:12:10	193.140.182.2	GET	/html/linkler/linklerorta.htm	- 80 -	82.152.52.3	Moz			
2004-12-27	00:12:12	193.140.182.2	GET	/html/kutuphane/kutuphanebar.htm	- 80 -	82.152.52.3				
2004-12-27	00:12:14	193.140.182.2	GET	/html/kutuphane/kutuphaneorta.htm	- 80 -	82.152.52.3				
2004-12-27	00:12:14	193.140.182.2	GET	/html/bilgiislem/bimbar.htm	- 80 -	82.152.52.3	Mozi			
2004-12-27	00:12:14	193.140.182.2	GET	/html/bilgiislem/bimorta.htm	- 80 -	82.152.52.3	Mozi			
2004-12-27	00:12:15	193.140.182.2	GET	/html/webmail/webmailbar.htm	- 80 -	82.152.52.3	Mozi			
2004-12-27	00:12:15	193.140.182.2	GET	/html/duyurular/anasayfaduy/universitedeyasam/univer						
2004-12-27	00:12:15	193.140.182.2	GET	/html/duyurular/anasayfaduy/universitedeyasam/univer						
2004-12-27	00:12:15	193.140.182.2	GET	/html/kulupler/kuluplerbar.htm	- 80 -	82.152.52.3	Mo			
2004-12-27	00:12:17	193.140.182.2	GET	/html/duyurular/anasayfaduy/01-akademiktakvim/akadem						
2004-12-27	00:12:17	193.140.182.2	GET	/html/duyurular/anasayfaduy/02-yonetmelikler/yonetme						
2004-12-27	00:12:18	193.140.182.2	GET	/html/duyurular/anasayfaduy/02-yonetmelikler/yonetme						
2004-12-27	00:12:18	193.140.182.2	GET	/html/duyurular/anasayfaduy/01-akademiktakvim/akadem						
2004-12-27	00:12:18	193.140.182.2	GET	/html/duyurular/anasayfaduy/agistatikleri/agistatikl						
2004-12-27	00:12:19	193.140.182.2	GET	/html/duyurular/anasayfaduy/agistatikleri/agistatikl						
2004-12-27	00:12:19	193.140.182.2	GET	/html/anasayfa/oryantasyon/oryantasyonbar.htm	- 80 -					
2004-12-27	00:12:20	193.140.182.2	GET	/html/duyurular/tumduyurubar.htm	- 80 -	82.152.52.3				

Şekil 6.1. Web günlük verisinin işlenmemiş hali

Yazılımın Delphi ile ilgili kısmında ise ilk olarak günlük verisinin tutulduğu veri tablosundaki URL alanı üzerinde işlem yapıldı. Bu tablodaki URL alanı içerisinde kullanıcıların ziyaret ettikleri sayfanın yolu tutulmaktadır. Örnek bir URL alanı "html\anasayfa\duyurular\duyurular.html" şeklindedir. Bu alan üzerinde yapılan işlem son "v" işaretinden sonra yer alan sayfa adının alınıp tekrar aynı alan içerisine yazılmasıdır.

Bu işlemin ardından sayfa kodlamak için kullanılan ve Microsoft Visio Programı yardımı ile elde edilen site ağacı Şekil 6.2 'da görülmektedir. Bu site ağacı kullanılarak ilk adımda elde edilen dosya isimleri sayılarla kodlanarak Şekil 6.3' de görülen ve "sayfakod" isimli farklı bir veri tabanı dosyasına kaydedildi. Daha sonra günlük verisinin olduğu tablodaki her bir

kullanıcının “URL” alanındaki dosya isimleri, “sayfakod” isimli tablodaki kendisine karşılık gelen sayfa kodları ile kodlandı. Bu işlem sonucunda elde edilen sayfa kodları günlük verisine ilave edilen “skod” alanına aktarılmış ve günlük verisi Şekil 6.4’deki hale dönmüştür.



Şekil 6.2. Gaziosmanpaşa Üniversitesi site ağacından bir kesit

sayfakod	sayfaad	sid
1.1.*	tumduyuru.htm	1
1.1.1	tumduyuruorta.htm	2
1.1.1.1	tumduyuruorta.htm#	3
1.1.1.2	sonuc.htm	4
1.1.1.3	duysem.htm	5
1.1.1.4	duyburs.htm	6
1.1.1.6	19-08-2004-01.htm	7
1.1.1.8	01-03-2004-01.htm	8
1.1.1.9	03-02-2005-02.htm	9
1.1.1.10	03-02-2005-01.htm	10
1.1.1.11	02-02-2005-01.htm	11
1.1.1.12	21-02-2005-01.htm	12
1.1.1.13	15-02-2005-03.htm	13
1.1.1.14	15-02-2005-02.htm	14
1.1.1.15	15-02-2005-02.htm	15
1.1.1.16	08-02-2005-03.htm	16
1.1.1.17	08-02-2005-02.htm	17
1.1.1.18	08-02-2005-01.htm	18
1.1.1.19	02-02-2005-02.htm	19
1.1.1.20	28-01-2005-03.htm	20
1.1.1.21	28-01-2005-02.htm	21
1.1.1.22	28-01-2005-01.htm	22
1.1.1.23	10-01-2005-02.htm	23
1.1.1.24	10-01-2005-01.htm	24
1.1.1.25	03-01-2005-01.htm	25
1.1.1.26	31-12-2004-03.htm	26
1.1.1.27	31-12-2004-02.htm	27
1.1.1.28	31-12-2004-01.htm	28
1.1.1.29	29-12-2004-02.htm	29
1.1.1.30	28-12-2004-03.htm	30
1.1.1.31	29-12-2004-01.htm	31
1.1.1.32	28-12-2004-02.htm	32
1.1.1.33	28-12-2004-01.htm	33
1.1.1.34	23-12-2004-03.htm	34
1.1.1.35	23-12-2004-02.htm	35

Şekil 6.3. Sayfa kodlarının tutulduğu tablo

	tarih	zaman	ip	url	skod
	2005-04-07	00:02:12	66.196.91.133	yonetimpersonel.htm	7.1.12
	2005-04-07	00:05:05	66.196.91.92	tumduyuru.htm	1.*.*
	2005-04-07	00:08:44	64.242.88.50	02-ogrenci.htm	22.1
	2005-04-07	00:12:25	66.196.90.167	gop_turhal.htm	12.1.3
	2005-04-07	00:17:45	195.175.18.26	02-ogrenci.htm	22.1
	2005-04-07	00:19:05	66.196.91.130	cumhurzamangzt.htm	10.1.14
	2005-04-07	00:21:27	212.156.210.224	03-08-2004-02.htm	1.1.120
	2005-04-07	00:22:47	195.175.18.26	googlearama.htm	31
	2005-04-07	00:24:12	195.175.18.26	02-ogrenci.htm	22.1
	2005-04-07	00:28:18	195.175.18.26	googlearama.htm	31
	2005-04-07	00:28:49	195.175.18.26	02-ogrenci.htm	22.1
	2005-04-07	00:32:56	207.46.98.31	fedindex.htm	11.1.10.1
	2005-04-07	00:33:25	195.175.18.26	02-ogrenci.htm	22.1
	2005-04-07	00:35:33	64.242.88.50	yonetimorta.htm	7.1.*
	2005-04-07	00:35:58	85.99.102.247	googlearama.htm	31
	2005-04-07	00:37:00	85.99.102.247	02-ogrenci.htm	22.1
	2005-04-07	00:44:11	81.213.252.131	googlearama.htm	31
	2005-04-07	00:44:15	81.213.252.131	akademik.htm	9.1.2.2
	2005-04-07	00:44:16	81.213.252.131	akademikorta.htm	11.1.*.*
	2005-04-07	00:44:20	81.213.252.131	akademikfenbilimleri.htm	11.1.23
	2005-04-07	00:47:53	81.213.252.131	akademik.htm	9.1.2.2
	2005-04-07	00:47:53	81.213.252.131	akademikfenbilimleri.htm	11.1.23
	2005-04-07	00:49:31	81.214.145.0	googlearama.htm	31
	2005-04-07	00:49:47	81.214.145.0	02-ogrenci.htm	22.1
	2005-04-07	00:50:29	65.32.129.90	googlearama.htm	31
	2005-04-07	01:07:01	195.175.18.26	googlearama.htm	31
	2005-04-07	01:08:25	195.175.18.26	02-ogrenci.htm	22.1
	2005-04-07	01:09:54	85.100.115.206	googlearama.htm	31
	2005-04-07	01:10:17	85.100.115.206	2004album.htm	16.*
	2005-04-07	01:10:17	85.100.115.206	2004albumorta.htm	16.1
	2005-04-07	23:55:16	195.175.18.26	02-ogrenci.htm	22.1
	2005-04-07	23:59:33	217.193.143.42	kulupler.htm	3.*
	2005-04-07	23:59:59	217.193.143.42	tumduyuru.htm	1.*.*
	2005-04-07	23:34:38	85.96.7.14	kutuphane.htm	17.*
	2005-04-07	23:40:39	81.215.42.169	26-03-2005-02.htm	1.1.154
	2005-04-07	23:59:35	217.193.143.42	agistatikleri.htm	12.*.*

Şekil 6. 4. Günlük verisinin sayfa kodlama işleminden geçmiş hali

Bir sonraki adımda elde edilen sayfa kodları rakamlar yerine “*” karakteri gelecek şekilde Şekil 6.5’da olduğu gibi yeniden kodlandı (2.5.1 -> 2.5.* şeklinde) ve her kullanıcı için yeni birer kayıt olarak eklendi.

Bu işlemten sonra bütün kayıtlara ait ziyaret süreleri hesaplandı. Hesaplanan değerler “gecensure” adı ile tanımlanan ve yine günlük tablosuna ilave edilen alana yazıldı. Bu alandaki değerler bir kullanıcının herhangi bir sayfayı ziyareti için geçen süreyi göstermektedir. Sayfa kodları ve süreler elde edildikten sonra yapılan işlem Şekil 5.1’de verilen üyelik fonksiyonları kullanılarak “gecensure” değerlerinin bulanıklaştırılmasıdır. Bu üyelik fonksiyonlarından geçirilerek elde edilen “Kısa”, “Orta” ve “Uzun” değerleri hesaplanmıştır. Verinin son durumu Şekil 6.6’da görülmektedir

ip	url	skod	gecensure
133.227.51.84	webmail.htm	15	180
128.227.51.80	googlearama.htm	31	71
128.227.82.84	googlearama.htm	31	68
130.238.5.5	akademikorta.htm	11.*.*	7
130.238.5.5	akademikfened.htm	11.*.*	7
130.238.5.5	akademikfened.htm	11.1.*.*	7
130.238.5.5	akademikorta.htm	11.1.*.*	7
130.238.5.5	akademikfened.htm	11.1.10.*	7
130.238.5.5	googlearama.htm	31	31
130.238.5.5	akademik.htm	9.*.*	37
130.238.5.5	akademik.htm	9.1.*.*	37
130.238.5.5	akademik.htm	9.1.2.*	37
130.238.5.5	akademik.htm	9.1.2.2	37
133.44.1.221	fedindex.htm	11.*.*	25
133.44.1.221	fedindex.htm	11.1.*.*	25
133.44.1.221	fedindex.htm	11.1.10.*	25
133.44.1.221	fedindex.htm	11.1.10.1	25
133.44.1.221	akademik.htm	9.*.*	5
133.44.1.221	akademik.htm	9.1.*.*	5
133.44.1.221	akademik.htm	9.1.2.*	5
133.44.1.221	akademik.htm	9.1.2.2	5
140.109.90.95	tanitimorta.htm	10.*.*	137
140.109.90.95	tanitim.htm	10.*.*	149
140.109.90.95	tanitimorta.htm	10.1.*	137
140.109.90.95	akademikorta.htm	11.*.*	121
140.109.90.95	akademikorta.htm	11.1.*.*	121
140.109.90.95	03-dahilitel.htm	23	47
140.109.90.95	03-dahilitel.htm	23	273
140.109.90.95	googlearama.htm	31	64
140.109.90.95	yönetim.htm	7.*.*	203
140.109.90.95	yönetimsenato.htm	7.*.*	78
140.109.90.95	yönetimorta.htm	7.*.*	128
140.109.90.95	yönetimkurulu.htm	7.*.*	144
140.109.90.95	yönetimsenato.htm	7.1.*	78
140.109.90.95	yönetimkurulu.htm	7.1.*	144

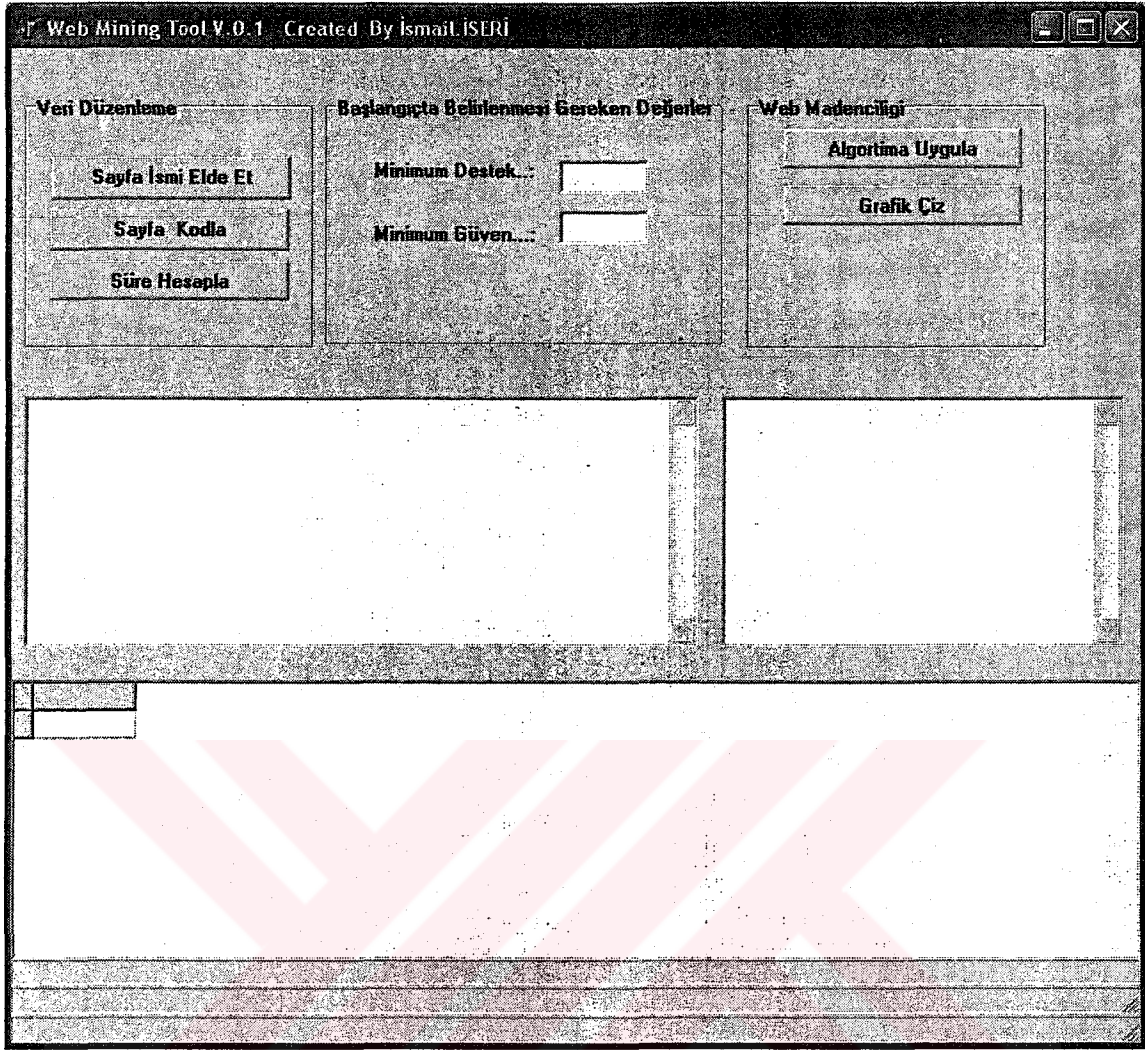
Şekili 6.5. Sayfa kodlarının '*' karakteri ile kodlanmış hali

ip	uri	skod	gecensure	short	middle	high
128.227.51.80	googlearama.htm	31	71	0	1	0
128.227.51.80	webmail.htm	15	180	0	0,55	0,45
128.227.82.84	googlearama.htm	31	68	0,04	0,96	0
130.238.5.5	googlearama.htm	31	31	0,78	0,22	0
130.238.5.5	akademik.htm	9.1.2.2	37	0,66	0,34	0
130.238.5.5	akademik.htm	9.1.2.*	37	0,66	0,34	0
130.238.5.5	akademik.htm	9.1.*.*	37	0,66	0,34	0
130.238.5.5	akademik.htm	9.*.*.*	37	0,66	0,34	0
130.238.5.5	akademikorta.htm	11.*.*.*	7	1	0	0
130.238.5.5	akademikorta.htm	11.1.*.*	7	1	0	0
130.238.5.5	akademikfened.htm	11.1.10.*	7	1	0	0
130.238.5.5	akademikfened.htm	11.1.*.*	7	1	0	0
130.238.5.5	akademikfened.htm	11.*.*.*	7	1	0	0
133.44.1.221	akademik.htm	9.1.2.*	5	1	0	0
133.44.1.221	akademik.htm	9.1.*.*	5	1	0	0
133.44.1.221	akademik.htm	9.*.*.*	5	1	0	0
133.44.1.221	akademik.htm	9.1.2.2	5	1	0	0
133.44.1.221	fedindex.htm	11.1.10.1	25	0,9	0,1	0
133.44.1.221	fedindex.htm	11.1.10.*	25	0,9	0,1	0
133.44.1.221	fedindex.htm	11.1.*.*	25	0,9	0,1	0
133.44.1.221	fedindex.htm	11.*.*.*	25	0,9	0,1	0
140.109.90.95	googlearama.htm	31	64	0,12	0,88	0
140.109.90.95	yönetim.htm	7.*.*	203	0	0,44	0,56
140.109.90.95	yönetimorta.htm	7.1.*	128	0	0,78	0,22
140.109.90.95	yönetimorta.htm	7.*.*	128	0	0,78	0,22
140.109.90.95	yönetimkurulu.htm	7.1.*	144	0	0,71	0,29
140.109.90.95	yönetimkurulu.htm	7.*.*	144	0	0,71	0,29
140.109.90.95	yönetimkurulu.htm	7.1.9	144	0	0,71	0,29
140.109.90.95	yönetimsenato.htr	7.1.10	78	0	1	0
140.109.90.95	yönetimsenato.htr	7.1.*	78	0	1	0
140.109.90.95	yönetimsenato.htr	7.*.*	78	0	1	0
140.109.90.95	tanitim.htm	10.*.*	149	0	0,69	0,31
140.109.90.95	tanitimorta.htm	10.1.*	137	0	0,74	0,26
140.109.90.95	tanitimorta.htm	10.*.*	137	0	0,74	0,26
140.109.90.95	akademik.htm	9.1.2.*	19	1	0	0

Şekil 6.6. Bulanıklaştırılmış veriden bir görünüm

Bu adım da tamamlandıktan sonra programımız artık madencilik için hazır hale gelmiş veriden kural çıkarımı işlemine geçmiştir. Yazılan “counthesapla” isimli yordam aracılığıyla dışardan alınan minimum destek ve minimum güven değerleri kullanılarak bulanık kurallar ve sıralı örüntüler elde edilmiştir.

Temizlenmiş ve düzenlenmiş veriden kural çıkartımı yapan programımızın arayüz görüntüsü de Şekil 6.7’de olduğu gibidir.



Şekil 6.7. Web madencilik programının ana sayfa ekran görüntüsü

Program aracılığıyla veri üzerinde şu işlemler yapılabilir: Sayfa isimlerinin elde edilmesi, sayfaların site ağacına göre kodlanması, sayfa ziyaret sürelerinin hesaplanması ve algoritmamızın adımlarına göre bulanık bağıntı kurallarının ve sıralı örüntülerin çıkarılması.

Programın grafik çizme bölümünde ise, farklı minimum destek değerlerine göre elde edilen geniş nesne kümelerinin sayısı ve farklı minimum güven değerlerine göre elde edilen kural sayıları gösteren grafikler elde edilebilir.

Program dışardan girişi yapılabilen minimum destek ve güven değerleri ile çalıştırılır. Elde edilen geniş nesne kümeleri, zaman kısıtlamalı bulanık bağıntı kuralları ve sıralı örüntüler program sonlandıktan sonra otomatik olarak dosyalara yazdırılır. Şekil 6.8’de, Şekil 6.9’da ve Şekil 6.10 ‘da programın çalıştırılması sonucunda elde edilen dosyaların ekran görüntüleri verilmiştir.

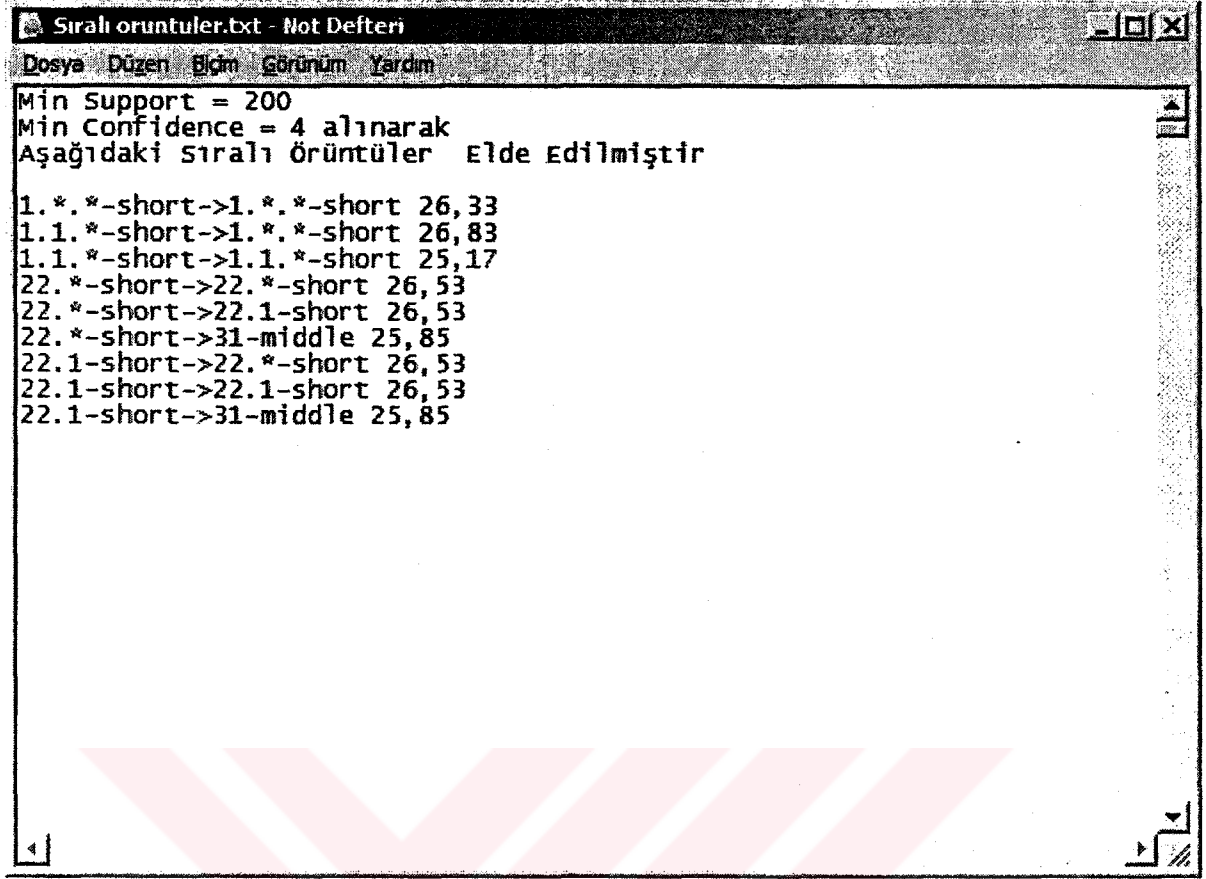
```
ikilinesne.bkt - Not Defteri
Dosya Düzen Ekle Görünüm Yardım

Minimum Support değeri...:50
Elde Edilen İkili Geniş Nesneler :
1.*.* . short - 1.*.*short-10962,75999999989
1.*.* . short - 1.1.*short-10251,59999999991
1.*.* . short - 10.*.*short-4046,580000000016
1.*.* . short - 10.1.*short-3338,340000000013
1.*.* . short - 11.*.*short-4199,580000000016
1.*.* . short - 11.*.*.*short-1055,539999999999
1.*.* . short - 11.1.*short-4199,580000000016
1.*.* . short - 11.1.*.*short-1055,539999999999
1.*.* . short - 13.*.*short-1375,22
1.*.* . short - 15short-939,719999999992
1.*.* . short - 16.*short-1356,419999999999
1.*.* . short - 17.*short-1228,9
1.*.* . short - 19.*short-1566,480000000001
1.*.* . short - 2.*.*short-2442,260000000004
1.*.* . short - 2.1.*short-1517,780000000001
1.*.* . short - 2.1.6short-131,68
1.*.* . short - 20.*short-1341,12
1.*.* . short - 22.*short-4695,200000000005
1.*.* . short - 22.1short-4695,200000000005
1.*.* . short - 23short-904,519999999997
1.*.* . short - 25short-1508,12
1.*.* . short - 29short-745,759999999999
1.*.* . short - 3.*short-1418,24
1.*.* . short - 30.*.*short-1404,519999999999
1.*.* . short - 30.1.*short-702,139999999999
```

Şekil 6.8. Program tarafından elde edilen ikili yoğun nesne kümesi

```
kurallar.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
Minimum Güven Değeri... :3 alınarak
Aşağıdaki kurallar elde edilmiştir
1.*.*-short->1.*.*-short 26,33
1.*.*-short->1.1.*-short 24,62
1.*.*-short->10.*.*-short 9,72
1.*.*-short->10.1.*-short 8,02
1.*.*-short->11.*.*-short 10,09
1.*.*-short->11.1.*-short 10,09
1.*.*-short->13.*.*-short 3,3
1.*.*-short->16.*.*-short 3,26
1.*.*-short->19.*.*-short 3,76
1.*.*-short->2.*.*-short 5,87
1.*.*-short->2.1.*-short 3,65
1.*.*-short->20.*.*-short 3,22
1.*.*-short->22.*.*-short 11,28
1.*.*-short->22.1.*-short 11,28
1.*.*-short->25.*.*-short 3,62
1.*.*-short->3.*.*-short 3,41
1.*.*-short->30.*.*-short 3,37
1.*.*-short->31.*.*-middle 15,43
1.*.*-short->4.*.*-short 3,73
1.*.*-short->7.*.*-short 8,61
1.*.*-short->7.1.*.*-short 7,21
1.*.*-short->9.*.*.*-short 7,68
1.*.*-short->9.1.*.*-short 5,82
1.*.*-short->9.1.2.*.*-short 3,36
1.1.*-short->1.*.*-short 26,83
1.1.*-short->1.1.*-short 25,17
```

Şekil 6.9. Program tarafından elde edilen bulanık bağıntı kuralları

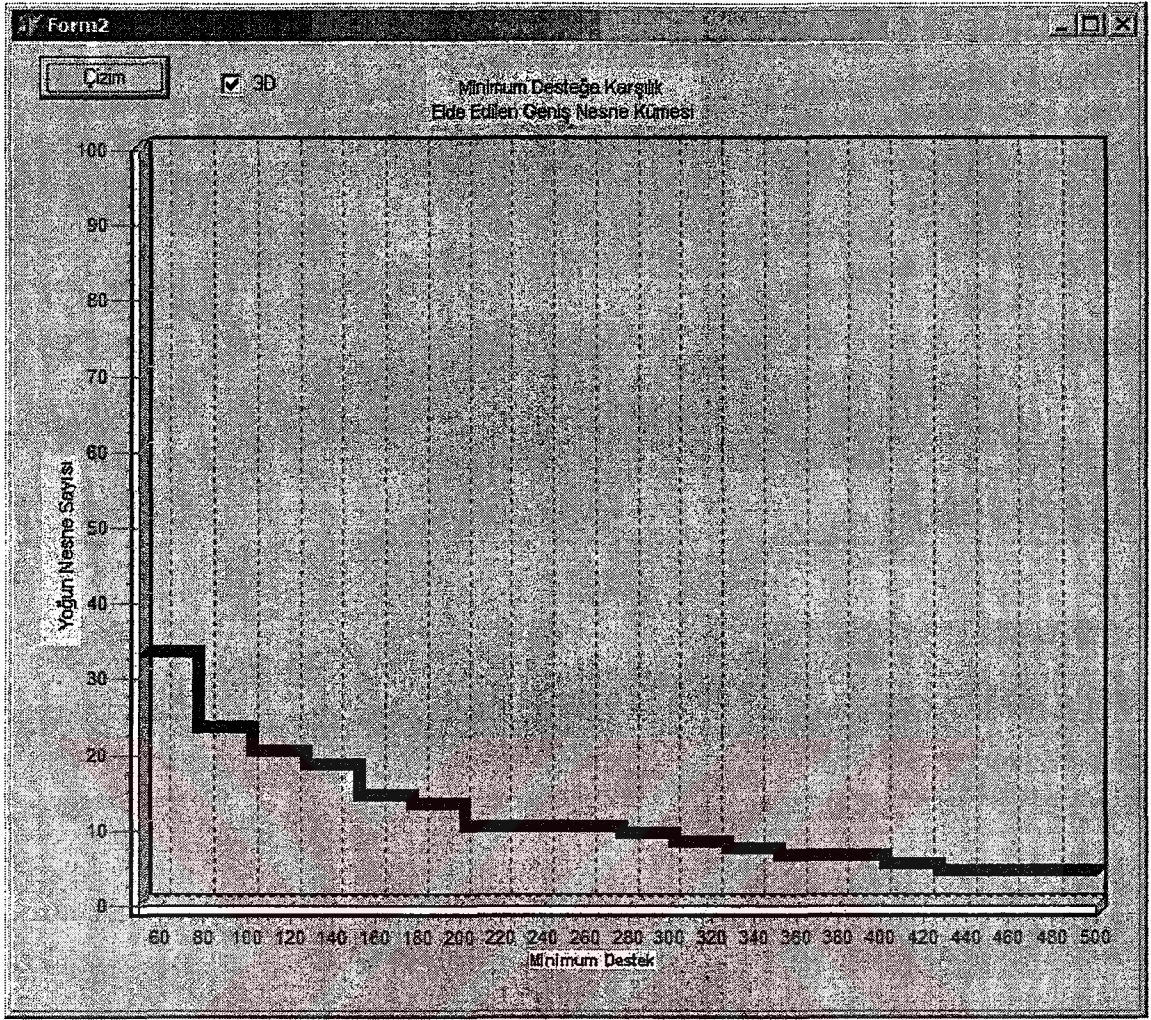


```
Min Support = 200
Min Confidence = 4 alınarak
Aşağıdaki sıralı örüntüler elde edilmiştir

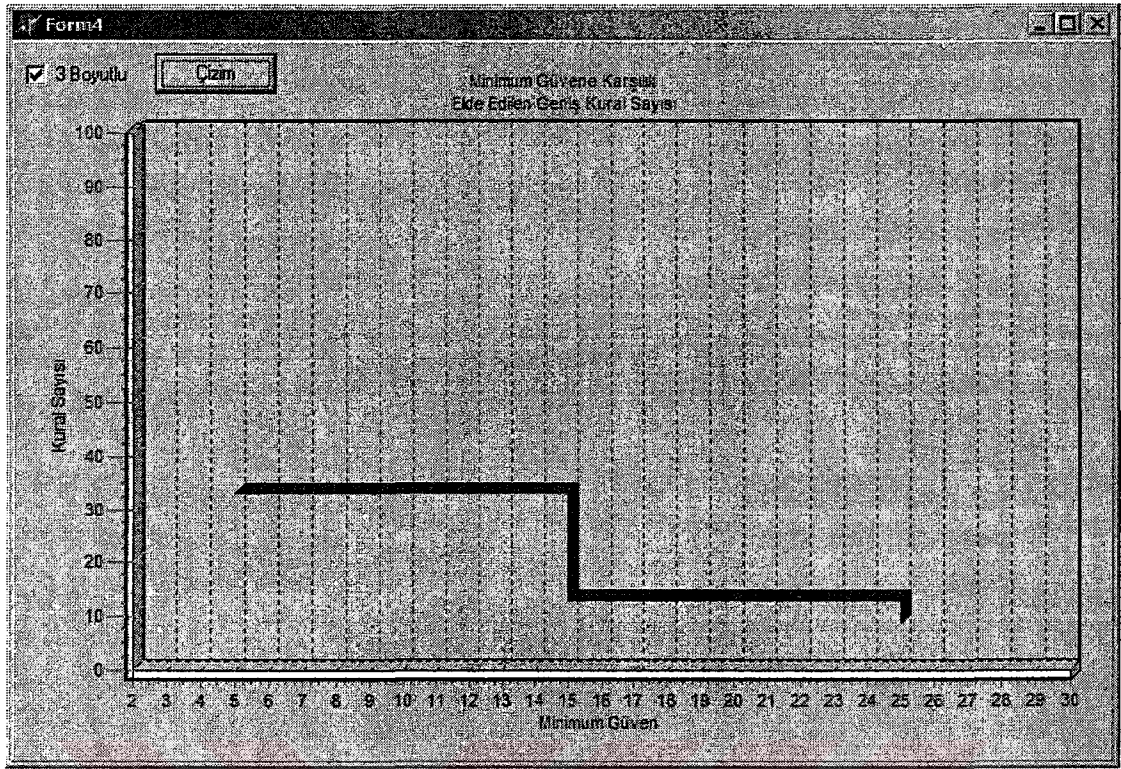
1.*.*-short->1.*.*-short 26,33
1.1.*-short->1.*.*-short 26,83
1.1.*-short->1.1.*-short 25,17
22.*-short->22.*-short 26,53
22.*-short->22.1-short 26,53
22.*-short->31-middle 25,85
22.1-short->22.*-short 26,53
22.1-short->22.1-short 26,53
22.1-short->31-middle 25,85
```

Şekil 6.10. Program tarafından elde edilen sıralı örüntüler

Programın farklı destek değerlerinde çalıştırılarak elde edilen geniş nesne kümelerinin sayısı ve farklı minimum güven değerlerinde çalıştırılarak elde edilen bulanık bağıntı kurallarının sayısı sırasıyla Şekil 6.11 ve Şekil 6.12’ de görülmektedir. Bu şekillerden de görülebileceği gibi minimum destek değeri yükseldikçe elde edilen yoğun nesne küme sayısı azalmaktadır. Buna benzer olarak minimum destek değeri yükseldikçe de elde edilen kural sayısı azalmaktadır.



Şekil 6.11. Minimum Destek – Yoğun Nesne Kümesi ilişkisi



Şekil 6.12. Minimum Güven– Kural Sayısı ilişkisi

7. SONUÇLAR

Bu tez çalışmasında, son yıllarda oldukça popüler hale gelen web kullanım madenciliği konusu üzerine çalışılmıştır. Çalışma sürecinde site ağaçları ve bulanık küme kavramı kullanılarak çoklu seviyeli bulanık ilişki kuralları ve sıralı örüntüler elde edilmiştir. Web madenciliği uygulamaları günümüzde bir çok alanda kullanılmaktadır. Temel olarak kullanıldığı alanlar ise ziyaretçi karakteristikleri elde etmek, kendi kendini güncelleyen siteler oluşturmak ve sunucu performansını artırmaktır. Madenciliği yapılan verilerden elde edilen meta-data, ihtiyaca göre yukarıda bahsedilen alanlardan herhangi birisi için girdi olarak kullanılarak bir çok uygulama geliştirilebilir.

Bu çalışmada Gaziosmanpaşa Üniveristesine ait web sunucu günlük dosyası verisi alınarak üzerinde madencilik algoritması uygulanmış ve siteyi ziyaret eden ziyaretçilerin sıklıkla ve ardı ardına ziyaret ettikleri sayfalar tespit edilmiştir. İşlemlerde bulanık küme teorisinden faydalanılarak sayfalara yapılan ziyaretler sırasında gecen süreler bulanıklaştırılarak daha kullanışlı bulanık bağıntı kuralları elde edilmiştir. Normalde A sayfasına girenler, B sayfasına da girmiştir şeklinde elde edilen kuralların yerine, A sayfasında kısa süre bekleyenler, B sayfasında uzun süre beklemektedirler şeklinde bir zaman eklentisi getirilmiştir. Yine çalışmada bulanık bağıntı kuralları ve sıralı örüntüler elde edilirken sayfaların ziyaret edildikleri zaman değerleri dikkate alınarak zaman kısıtlama kavramı kullanılmış ve bu sayede daha etkin kurallar ve daha etkin sıralı örüntüler elde edilmiştir.

Uygulama sonuçlarına dikkat edildiğinde yüksek destek değerlerinde elde edilen geniş nesne küme sayısının azaldığı, benzer olarak yüksek güven değerlerinde de elde edilen kural sayılarının azaldığı görülmüştür. Buradan çıkarılabilecek sonuç yüksek destek ve güven değerleri sonucu elde edilen kuralların ve örüntülerin daha gerçekçi ve sağlam olduklarıdır. Bunların dışında kullanılan çoklu seviye yapı sayesinde bilinen sıralı örüntülerden farklı olarak çoklu seviyeli sıralı örüntüler elde edilmiştir. Bu işlem sayesinde normalde elde edilemeyecek türden örüntüleri seviyeli ağaç yapısı sayesinde dolaylı yoldan elde edilebilmesi sağlanmıştır.

KAYNAKLAR

1. Etzioni O., 1996, "The World Wide Web: quagmine or gold mine?" Communications of the ACM 39(11), 65 – 68.
2. Kasala R., Blackeel H., 2000, "Web Mining Resource : A Survey", SIGKDD Explorations, ACM 2(1), 1-15.
3. Chen L, Sycara K., 1998, "WebMate: a personal agent for browsing and searching", The Second International Conference on Autonomous Agents, ACM.
4. Chen M. S, Park J. S, Yu P. S, 1998, "Efficient data mining for path taversal patterns", IEEE Trans. Knowledge and Data Eng 10: 209–221.
5. Cohen E, Krishnamurthy B, Rexford J., 1999, "Efficient algorithms for predicting requests to web sunucus", The Eighteenth IEEE Annual Joint Conference on Computer and Communications Societies 1: 284–293.
6. Fayyad U., Shapiro G., and Smyth P., 1996, "From Data Mining to Knowledge Discovery in Database", AI Magazine, 37-54.
7. Han J., 2001, "Data Mining : Concept and Techniques", Morgan Kaufman.
8. Simoudis E., 1996, "Reality Check for Data Mining," IEEE Expert: Intelligent Systems and Their Applications, 26-33.
9. Agrawal R., Imielinski T. and Swami A., 1993, "Mining association rules between sets of items in large databases", ACM SIGMOD Conf. Management of Data, 207-216.
10. Agraval R. And Srikant R., 1995, "Mining Sequential Patterns", Proc. Of the 11th International Conference on Data Engineering, pp. 3-14.
11. Agraval R. And Srikant R., 1994, "Fast Algorithms for Mining Association Rules", Proc. of the 20th Int. Conf. Very Large Databases, VLDB.
12. Srikant R., and Agrawal R., 1996, "Mining quantative association rules in large relational tables", Proc. of ACM SIGMOD International Conference on Management pf Data, 1-12
13. Madria S.K, Raymond C., Bhowmick S.S, Mohania M., "Association Rules for Web Data Mining in WHOWEDA", Proceedings of the International Conference on Digital Libraries: Research and Practices . Kyoto, Japan, November 13 – 16.
14. Dumais S., Plott J., and Sahani M., 1998, "Inductive learning algorithms and representation for text categorizatation "In proc. of the ACM 7th International Conference ob Information and Knowledge Management, 148-155.
15. Scott S., and Matwin S., 1999. "Feature engineering for text classification", ICML-99.

16. Kleinberg J.M, 1998, "Authoritative Sources in a hyperlinked environment", In Proc. of ACM_SIAM Symposium on Discrete Algorithms, 168-677
17. Brain S., Poge L., 1998, "The Anatomy of a large scale hypertextual web search engine", In seventh International WWW Conference
18. Chakraboti S., 2000, "Data Mining for hypertext : A tutorial survey", ACM SIGKAD Explorations, 1(2):1-11
19. Srivastava J., Cooley R., Desphade M., Tan P.N, 2000, "Web usage mining : discovery and applications of usage patterns from web data ", SIGKAD Explorations, 1(2): 12-33.
20. Pal S., Talwar V., Mitra P, 2002, "Web mining in soft computing framework: relevance, state of the art and future directions", IEEE Transactions on Neural Network, 13(5):1163-1177
21. Facca F.M, Lonzi P.L., 2004, "Mining interesting knowledge from weblogs: a survey ", Data and Knowledge Engineering.
22. Cooley R., Mobasher B., Srivastava J., 1999,"Data preparation for mining world wide web browsing patterns", Knowledge and Information Systems, 1(1):5-32.
23. Iseri.I, Gungor.S, Kaya.M,2004,"Hastalık teşhisinde etkin bir bulanık bağıntı çıkarım yöntemi", TAINN2004 Symposium, İzmir, Türkiye.
24. Kaya M., and Alhajj R., 2004, "Integrating Multi-objective Genetic Algorithms into Clustering for Fuzzy Association Rules", IEEE int. Cong. on Data Mining.
25. Kaya M., and Alhajj R., 2005,"Genetic Algorithm based Framework for mining Fuzzy Association Rules", Fuzzy Sets and Systems, 152(3): 587-601

ÖZGEÇMİŞİM

iiiseri@gop.edu.tr

Tlf: 0 (356) 252 16 16

Gaziosmanpaşa Üniversitesi

Tokat Meslek Yüksekokulu

TOKAT

1980 yılında Tokat ili Turhal ilçesinde doğdum. İlk orta ve lise öğrenimimi Tokat'ta tamamladıktan sonra 2000 yılında Fırat Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünü kazandım. 2003 yılında bölüm üçüncüsü olarak mezun oldum. 2004 yılında Fırat Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana Bilim Dal'ında yüksek lisansa başladım. 2005 yılında Gaziosmanpaşa Üniversitesi tarafından yapılan öğretim görevliliği sınavını kazanarak öğretim görevlisi olarak göreve başladım. Halen Gaziosmanpaşa Üniversitesi Tokat Meslek Yüksekokulunda öğretim görevlisi olarak görev yapmaktayım.