Marmara Fen Bilimleri Dergisi 2015, 1: 14-20
DOI: 10.7240/mufbed.83214

Yazılım Hata Logları Kullanılarak Veri Madenciliği Uygulaması Gerçekleştirilmesi

Gökhan ACUN[1], Turgay Tugay BİLGİN [2]

[1] Workcube E-İş Sistemleri A.Ş., İSTANBUL
[2]Maltepe Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, İSTANBUL

ÖZET

Veri madenciliği; genellikle makine öğrenmesi tekniklerini kullanarak önceden bilinmeyen verilerden bilgi çıkarma sürecidir. Bu çalışmada bir kurumsal kaynak planlama yazılımına ait günlük hata loglarını toplayarak birtakım önişleme işlemleri gerçekleştirip, elde edilen veri üzerinde birliktelik analizi algoritması çalıştıran bir yazılım geliştirilmiştir. Bu yazılım sayesinde, bir hatanın oluşmasına yol açan olaylar arasındaki ilişkiler birliktelik analizi algoritması ile tespit edilerek, sonuçları yönetim kademesine e-posta ile raporlayan bir yapı kurulmuştur. Ayrıca hataya sebep olan sık örüntüler tespit edilmiş ve görselleştirilmiştir.

Anahtar kelimeler: Veri Madenciliği, Log Madenciliği, Hata Madenciliği

Developing a Data Mining Application using Software Bug Logs

ABSTRACT

Data mining is the process of posing queries and extracting patterns, often previously unknown from large quantities of data using pattern matching or other reasoning techniques. In this study, a software has been developed in order to perform some data preprocessing steps on the daily bug logs of an Enterprise Resource Planning software. The software also applies association rule mining algorithm on the collected data. It sends discovered associations by email to the corporate managers. Moreover, the frequent patterns which causes bug, has been detected and visualized.

Keywords: Data Mining, Log Mining, Bug Mining

I. GİRİŞ

Veri madenciliği, bilgisayar bilimlerinin disiplinler arası bir sahasıdır [1]. Veritabanlarında bilgi keşfi (Knowledge-Discovery in Databases) olarak da bilinir [1]. Veri madenciliği; yapay zeka, makine öğrenmesi, istatistik ve veritabanı sistemlerinin kesişiminden oluşmaktadır ve büyük miktarlardaki veri setlerinde örüntüler keşfetmenin hesaplamalı bir sürecidir. Veri madenciliği sürecinin nihai amacı bir veri setinden bilgi üretmek ve onu daha ileri kullanımlar için anlaşılabilir bir veriye dönüştürmektir [1]. Veri madenciliğinin, siber güvenliğin yanı sıra ulusal güvenliği sağlamada da birçok uygulaması vardır. Ayrıca şüpheli insanların bulunmasında, bankalara daha iyi performans sağlamada, yeni müşteriler kazanmada, müşterilerin satın alma davranışlarını incelemede de kullanılmaktadır. Kredi kartı dolandırıcılığı ve biyometrik uygulamalar için uygulanabilir [2].

Veri madenciliği, veritabanı ve yapay zekâ teknolojilerinin bir kombinasyonudur. Yapay zekâ kavramı son on yılda ciddi bir sıçrama yapmıştır, ancak hali hazırda var olan bilgisayar bilimlerine de önemli bir katkı sağlayabileceğini de kanıtlamıştır. Birçok uzman, veri madenciliğinin internet ve veri ambarı çalışmalarından sonra endüstrideki en sıcak üçüncü saha olduğuna inanmaktadır [3].

Veri madenciliği gerçekte veriyi analiz etme sürecinden sonraki adımdır. Ancak sadece sorguları ve kullanıcı tanımlı ilişkileri elde etmek yerine bir adım ileri giderek verideki anlamlı ilişkileri de bulmaktadır. Bu ilişkiler veriyi incelemede daha derin bir bakış açısı kazandırmaktadır. Örneğin, bir bilgisayar tarafından oluşturulmuş grafik, kullanıcının anlaması için zor olabilir, ama veri madenciliği kullanıcıya verideki akımların tam olarak ne olduğunu daha net bir şekilde gösterir.

Bu çalışmada, öncelikle çalışmanın temelini oluşturan ve birer madencilik türü olan hata madenciliği ve log madenciliğinden bahsedilmiştir. Ardından çalışmada kullanılan veri setinden bahsedilmiş, veri setine uygulanan ön işleme safhaları anlatılmıştır. İlgili veri ön işlemenin ardından geliştirilen yazılım hakkında bilgi verilmiştir. Son olarak bu yazılımın sonuçlarından bahsedilerek

Sorumlu yazar/Corresponding author: Turgay Tugay BİLGİN Tel: +90 216 626 1050/2384, **e-posta:** ttbilgin@gmail.com
Gönderilme/ Submitted: 28.09.2014, **Kabul/Accepted:**16.03.2015

değerlendirme ve önerilerde bulunulmuştur.

1.1 Hata Madenciliği

Yazılımdaki kusurlara hata (bug) denir [4]. Hatalar, yazılım geliştirirken beklenmedik şekillerde oluşan davranışlardır. Yazılım testçileri ve kalite mühendisleri tarafından yazılımlar test edilirken bu tarz beklenmedik şekilde oluşan davranışlar hata olarak işaretlenir. Hatalar; Bugzilla, Perforce, JIRA vb. araçlarla yönetilir ve izlenirler [4].

Statik analiz araçları daha önceden tanımlı hataları bulmada sorun yaşamazlar, ancak yeni hata tiplerini öngörmek ve bulmak, ilgili programlama dilini derinlemesine kavrayan bir anlayış gerektirir [5]. Derinlemesine gözden geçirmeler daha fazla zaman gerektirirler. Hata raporlarındaki metinsel veriler, hatanın düzeltilmesi sürecinde yazılım geliştiricilerine önemli bilgiler sağlar [6]. Hata veritabanı madenciliği hakkındaki daha önceki çalışmalar üç temel hususa odaklanmaktadır [6]:

- Hata düzeltme sürecini doğru kişiye atamak
- Tekrarlı hata raporlarını bulmak
- Raporlanmış bir hataya doğru şiddeti (severity) atamak

Bir yazılım hata deposu, sadece yazılım hataları hakkında bilgi içermez, yazılım geliştiricilerinin, kalite mühendislerinin (testçiler), yöneticilerin ve diğer ekip üyelerinin yazılım hatalarını çözmek için verdikleri çaba hakkında da bilgi içermektedir. Bu bilgi yararlı bilgi örüntüleri çıkarabilmek için analiz edilebilir [7].

Bazen yönetici ve ekip liderleri bazı soruları kendilerine sormak zorunda kalırlar. “Bu yeni gelen yazılım hatasını kime atasam acaba?” veya “Benzer yazılımsal hatayı daha önce çözmüş bir kişi var mı?” bu tarz sorulardır. Aslında ekip liderleri hataları çözerken ekip üyelerinin iş yükünü ve daha önce ilgili hataya yakın benzerlikte hataları çözüp çözmediğini göz önünde bulundururlar. Bu ise elle gerçekleştirilen bir süreçtir. Yazılım geliştiricilerine hata raporlarını atamak, zaman kaybına sebep olan sıkıcı bir iştir.

Bir yazılım hata deposu yazılım hataları hakkında bilgi tutmaktadır. Yazılım hata raporları; hatanın raporlandığı tarih, hatanın tanımı, hatanın özeti (başlığı), hatanın atandığı yazılım geliştirici, hatayı atayan kişi, ekip üyelerinin yorumları vb. hata niteliklerini barındırır [7].

Hata izleyicileri, sürüm kontrol sistemleri ve mesaj panoları, yazılım projelerinin gelişmesine yardımcı olur [7]. Bu tarz araçlar yazılım projeleri hakkında bilgiler içermektedir. Bu bilgiler yazılım geliştirme aşamasına rehberlik etmesi için veri madenciliğinden faydalanabilir. Hata raporlama sistemleri, kullanıcıların hata raporlarını gönderebilmesini, tanımlayabilmesini, izleyebilmesini, yorum yapabilmesini ve hata raporlarını sınıflandırabilmesini sağlar. Bugzilla, hata raporlama aracı için bir örnektir [7].

Bugzilla gibi bir hata-izleme sistemi; geliştirim ekipleri, test ekipleri ve son kullanıcılar gibi çeşitli kaynaklardan toplanmış hata raporlarını tutarlar. Hata raporlayıcıları, bir hata raporlama sistemine hata raporlarını gönderdiğinde gönderilen hataların güvenlik problemine sebep olup olmadığını göstermek için güvenlikle ilgili hata raporu (security bug reports - SBR) ya da güvenlikle ilgili olmayan hata raporu (not-security bug reports - NSBR) diye hata raporlarını etiketleme ihtiyacı duyarlar [8]. Güvenlikle ilgili olan hatalar hatanın düzeltilmesinde daha yüksek önceliğe sahiptirler. Bununla birlikte hata raporlayıcıları güvenlikle ilgili bilgi eksikliği yüzünden bazen SBR’leri NSBR şeklinde etiketlendirebilirler. Bu yanlış anlama güvenlik hatalarını tanımlama ve düzeltilmesinde gecikmeye bağlı olarak yazılımcı ve sistemcilere zarara sebep olmaktadır.

Hata raporlarının analizi, yazılım hata madenciliği alanı içinde önemli bir alt sahadır. Hataları sınıflandırma ve hatalara öncelik atama sürecine dair bilgiler ortaya çıkarmayı amaçlar. Hata verisi, Bugzilla ve JIRA gibi sistemlerden kolaylıkla erişilebiliyor olmasına rağmen, veritabanına aktarıldığında daha kolay kopyalamaya izin verdiği için analiz sürecini ciddi şekilde kolaylaştırabilir [9].

1.2 Log Madenciliği

Yazılım sistemleri log dosyalarında kendi aktiviteleriyle ilgili bilgiler toplamaktadır. “loglamak” terimi tamamlanmış aktivitelerin kaydını tutmak için bir log defterine kayıt girişi yapmak demektir. Log olarak adlandırılan bilgiler, yazılım sisteminin hareketleri ya da olayları, durum bilgileri ve hata bilgilerinden oluşur. Herhangi bir log satırı temel olarak tarih ve zaman bilgisi, kullanıcı bilgisi, uygulama bilgisi ve olay bilgisi içermektedir. Loglar genellikle sistem izleme, sistemin hatalarını ayıklama ve hata teşhisi için toplanırlar.

Log dosyaları, bir sistemin işleyişi hakkında önemli bilgiler barındırmaktadır. Bu bilgi; genellikle hata ayıklama, operasyonel profilleştirme, anormallikleri bulma, güvenlik tehditlerini saptama, performansı ölçme vb. gibi işlemler için kullanılır. Log dosyalarını elle okuma (manually) hala en geniş kullanılan tekniklerden birisidir. Bu kadar büyük bir bilgi kitlesini böyle okumak hata olasılığını yükseltmektedir. Son yıllarda çeşitli veri madenciliği ve makine öğrenmesi algoritmaları log dosyalarındaki bilgiyi analiz etmek için kullanılmaktadır [10].

Log madenciliği, loglar üzerinde veri madenciliğinin uygulanması işlemidir. Log verisine ön koşul tanımlama ve bir algoritma kullanarak madencilik işlemi uygulama log madenciliğinin en yaygın iki sahasıdır [11]. Ayrıca log

madenciliği, web loglarında gizlenmiş kullanıcı örüntülerini bulabilmek için de kullanılmaktadır.

II. MATERYAL VE YÖNTEM

2.1. Çalışmada Kullanılan Veri Seti

Bu çalışmada metal, mobilya ve perakende sektörü için geliştirilmiş bir ticari programa ait hata logları kullanılmıştır. Program içindeki herhangi bir sayfada yazılımsal bir hata gerçekleştiğinde belli formatta bir hata e-postası sistemde tanımlı belli kişilere gönderilmektedir. Bu e-posta içinde hatayla ilgili önemli bilgilere yer verilmiştir. Bu sayede sistemin yazılım geliştiricileri yazılım hataları hakkında detaylı bilgi elde edebilirler. Hatayı doğru yorumlayarak hataları ayıklayabilirler ve tamir edebilirler.

Hata e-postaları bir tabloda depolanmıştır. Bunun için tabloda tutulması planlanan alanlar belirlenmiştir. Tablo 1'de görüldüğü gibi bu alanlardan oluşan bir veritabanı tablosu oluşturulmuştur. Böylece sistem içinde herhangi bir sayfada yazılımsal bir hata oluştuğunda sistem normal işleyişinde olduğu gibi hatanın detaylı bilgilerini e-posta olarak göndermek yerine hataları ilgili tabloya yazmaktadır.

2.2 Veri Önişleme

Hata verileri bir veritabanında toplandıktan sonra birliktelik analizi uygulayabilmek için uygun bir biçime dönüştürülmeleri gerekmiştir. Genel olarak bu süreç önişleme olarak ifade edilmektedir. Önişleme sürecinde ilk adım veri içindeki gereksiz bilgilerin temizlenmesinden oluşmaktadır. Sayfa (PAGE) ve Hata Detayı (BUG_DETAIL) alanları düzenlenerek, diğerleri de var olan haliyle Apriori algoritmasının kullanacağı metin dosyaya aktarılır.

Düzenleme yapılan alanlar Bölüm 2.2.1 ve Bölüm 2.2.2'de açıklanmıştır:

2.2.1 Sayfa (PAGE)

Verinin toplandığı uygulamadaki sayfa adları PAGE alanında tutulmuştur. Ancak bu alanın tabloda tutulduğu ilk halinde adres çubuğu linki olarak uzun şekli yer almaktaydı. Tablo 2'de görüldüğü gibi her kayıt için aynı olan kısımlar kesilerek sayfa adı sade bir hale getirilmiştir.

Tablo 1. Yazılım hatalarının tutulduğu BUGS tablosu

Alan Adı	Sayfa	Tarayıcı Çeşidi	Kaydeden IP	Kaydeden	Hata Detayı
www.deneme.com	xxx.yyyy	MSIE	192.168.18.1	Ali Veli	Error Executing Database Query

Tablo 2. a) PAGE alanının sadeleştirmeden önceki hali

PAGE alanının veritabanında tutulduğu biçim
http://ep.workcube/index.cfm?fuseaction=settings.popup_form_add_process_cat
http://ep.workcube/index.cfm?fuseaction=report.detail_inventory_report

b) PAGE alanının sadeleştirmeden sonraki hali

PAGE alanının temizlenmiş biçimi
settings.popup_form_add_process_cat
report.detail_inventory_report

2.2.2 Hata Detayı (BUG_DETAIL)

Bir diğer alan da BUG_DETAIL alanıdır. Bu alan, hatanın birkaç cümlelik açıklamasıdır. Alanın tablodaki kayıtlarında yer alan tüm hata tanımları soyutlanmış ve statik alanlarına göre tekil bir şekle indirgenmiştir. Bu sayede her hataya ayrı birer tip atanmıştır. İlgili örnek hatalar ve karşılık gelen tekil değerleri Tablo 3'de görülmektedir.

Tablo 3. Örnek hata tipleri

Hata Tanımı	Hata Tipi ID
Could not find the included template … (Dosya bulunamadı hatası)	1
Error Executing Database Query. (SQL sorgusu hatası)	2

Sonuç olarak veri önişleme adımında sistem, "hatalar.txt" adında bir metin dosyası oluşturmuştur. Her veri, tırnak işareti ("") içinde eşittir (=) işaretiyle birleştirilmiş anahtar değer çiftleri halinde bulunmaktadır. Tırnak işareti içindeki ifadeler içinde boşluk karakteri bulunmaması gerekmektedir. Şekil 1'de, verilerin aktarılmasının ardından metin dosyasının görünümü görülmektedir.

hatalar.txt
"Kaydeden=AliVeli" "Sayfa=myhome.welcome" "Tarayıcı_Çeşidi=MSIE9" "Alan_Adi=deger"

Şekil 1. Apriori algoritmasının beklediği veri formatı

Her iki uygulama için de iki ek yazılım kullanılmıştır. Birinci yazılım veri ön işleme adımında sık örüntülerin tespit edilmesini sağlayan Apriori algoritmasıdır. Diğeri de

uygulamanın çalıştırılmasından sonra elde edilen değerlerin görselleştirilmesini sağlayan Graphviz yazılımdır.

Apriori sık öğe seti madenciliği ve birliktelik kuralı öğrenimi için kullanılan klasik bir algoritmadır [12]. Apriori tarafından saptanan sık öğe setleri, veritabanında genel akımları vurgulayan birliktelik kurallarını saptamak için kullanılabilir. Market sepet analizi gibi alanlarda uygulamaları bulunmaktadır.

Graf görselleştirme; soyut graf ve ağların diyagramları olarak yapısal bilgiyi sunmanın bir yoludur. Graphviz, açık kaynak kodlu bir graf görselleştirme yazılımıdır. Graphviz'in programlama dili DOT, basit metinsel formattaki graf tanımlarını okur ve Bitmap, Ölçülebilir Vektör Grafiği (Scalable Vector Graphic - SVG) ya da PostScript (ps) gibi farklı şekillerde kullanışlı diyagramlar üretir [13].

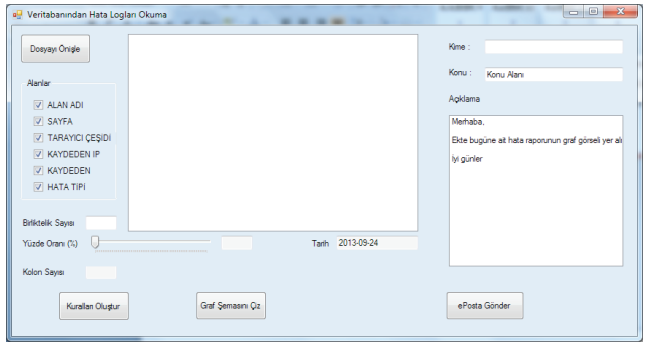
III. BULGULAR VE TARTIŞMA

Bu çalışmada iki farklı uygulama tasarlanmış ve gerçekleştirilmiştir. Bu uygulamalardan ilki "Windows servisi tabanlı hata analizi uygulaması", diğeri de Grafiksel kullanıcı arayüzü tabanlı hata analizi uygulamasıdır. Windows servisi tabanlı uygulama, çalıştığı bilgisayarın windows servisinin desteği altında işlem görmektedir. Bu uygulama periyodik olarak arkaplanda çalıştığı için herhangi bir arayüze sahip değildir. Diğer uygulamada ise uygulamanın ekran arayüzleri bulunmaktadır. Bu uygulamanın çalışması kullanıcının seçimine bağlıdır. Periyodik olarak çalışma özelliği bulunmamaktadır.

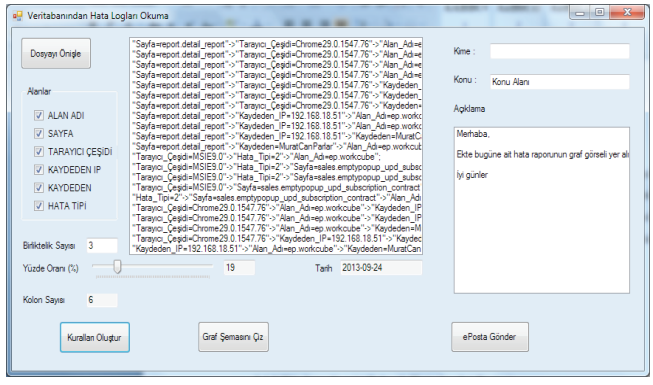
3.1 Grafiksel Kullanıcı Arayüzü Tabanlı Hata Analizi Uygulaması

Bu uygulamanın çalıştırılabilir dosyası (executable file) çağrıldığında Şekil 2'deki pencere açılacaktır. Ekran üzerinde dört adet buton bulunmaktadır. Öncelikle kullanıcıdan sayfada tanımlanması gereken bazı bilgileri girmesi beklenir. Ekranın sağ bloğunda yer alan "Alanlar" çerçevesi içinde hataların hangi alanlarının analiz edileceği bilgisi seçilir. "Dosyayı Önişle" butonuna basıldığında Bölüm 2.2'de anlatılan işlemler gerçekleştirilerek ekranın "Tarih" değerinde yer alan güne ait hatalar belli bir metin dosyasına atanacaktır. İşlem sorunsuz gerçekleştiğinde bunu "İşlem Başarıyla Gerçekleşti" uyarısıyla bildirecektir. Sonrasında "Birliktelik Sayısı" alanından çıkarılması gereken kuralların maksimum değeri girilmelidir. Bu değer "Alanlar" kısmında seçilmiş alan sayısından fazla olmamalıdır. Ayrıca "Yüzde Oranı" alanından kural dosyasında çıkarılması gereken birlikteliklerin genel verinin minimum yüzde kaçında yer alacağı belirtilmelidir.

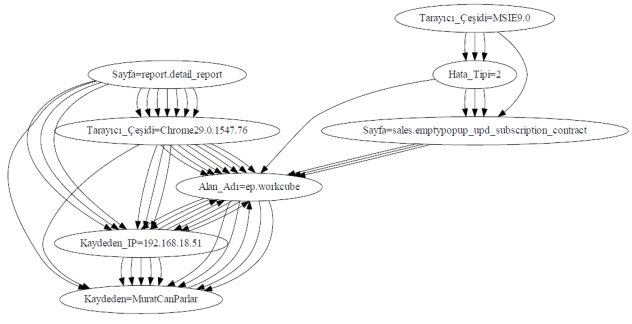
Ardından "Kuralları Oluştur" butonuna basılır. Sistem kullanıcıdan ilk butonda oluşmuş hataların dosyasını seçmesini ister. İlgili dosya seçilir. Ekrana işlemin sorunsuz gerçekleştiğini bildiren bir uyarı göstermektedir. Şekil 3'te görüldüğü gibi bunun dışında arkaplanda yer alan beyaz çerçeve içerisine de kuralların yer aldığı metin dosyasındaki birliktelikleri yansıtacaktır.



Şekil 2. Grafiksel Arayüz tabanlı uygulamanın ekran görüntüsü



Şekil 3. Birliktelik kuralları oluşturulduktan sonra ekran görüntüsü



Şekil 4. Graphviz yazılımı kullanıldığında elde edilen graf yapısı

Bu işlemden sonra kurallar oluşturulmuş ve birliktelikler çıkarılmış olur. Elde edilen birlikteliklerin görselleştirilmesi için ekranın alt bölümünde yer alan "Graf Şemasını

Çiz" butonuna tıklanır. Bu buton arkaplanda graphviz uygulamasını kullanarak ilgili kuralları görselleştirerek yeni bir windows formu üzerinde PDF okuyucu aracılığıyla Şekil 4'deki graf şeklini gösterecektir.

Son olarak bu görsel istenilen kişiyle paylaşılabilecektir. Şekil 3'te "Kime" kısmında görselin gönderileceği kişinin e-posta adresi girilir. Bunun yanısıra "Konu" alanına bir e-posta başlığı girilmelidir. Bu çerçevede "ePosta Gönder" butonuna basılır. Sistem ilgili kişiye e-postayı göndererek elde edilen bilgiyi kullanıcıya raporlar.

3.2 Windows Servisi Tabanlı Hata Analizi Uygulaması

Windows servisleri bir arayüze sahip olmayan arka planında çalışan hizmet programlarıdır. Bunlar sayesinde sistem performansına, sistemdeki hatalara, ağ erişimlerine, veritabanı işlemlerine hükmedilebilir.

Bu uygulama, daha önceden atanmış belli bir tetiklemeye dayanarak çalıştığı için grafiksel kullanıcı arayüzü uygulamasında dört ayrı butonla gerçekleştirilen fonksiyonları tek çatı altında birleştirmiştir. Bu sayede windows servisin her tetiklenmesinde tüm işlemler bir arada gerçekleştirilerek en sonunda ilgili kişiye e-posta gönderilmiştir.

Grafiksel kullanıcı arayüzü uygulamasında "Birliktelik Sayısı", "Yüzde Oranı", "Kolon Sayısı", "E-posta Gönderilecek Kişi" gibi bazı bilgiler dinamik olarak belirlenebiliyordu. Ancak bu uygulamada bu bilgiler statik olarak tanımlanmıştır.

I. SONUÇLAR

Yapılan çalışma kapsamında bir örnek teşkil etmesi için hata veritabanındaki hatalardan kayıt tarihi 24.09.2013 olanlar incelenmiştir. Veritabanında o güne ait 38 tane hata bulunmaktadır. Bu hataların veritabanındaki görünümleri Şekil 5'te yer almaktadır.

Geliştirilen uygulama %20'den fazla ve en az üçlü geçen birliktelikleri bulmak üzere çalıştırılmıştır. Uygulama ilk etapta veritabanından bu 38 kaydı işlemiş ve hatalar.txt dosyasını oluşturarak bu 38 kaydı Apriori algoritması için hazır duruma getirmiştir. Apriori algoritması %20'den fazla ve en az üçlü birliktelikleri kural.txt dosyasına kaydetmiştir.

Buna göre 1 adet beşli birliktelik, 6 adet dörtlü birliktelik ve 14 adet üçlü birliktelik saptanmıştır. Daha sonra bu birlikteliklere Graphviz yazılımı ile graf çizdirilmiştir. Bu yazılım keşfedilen birliktelikleri Şekil 4'te olduğu gibi görselleştirmiştir. Bu görselden aşağıdaki gibi çıkarımlar yapılabilmektedir:

a) "report.detail_report sayfasında oluşan hatalar mutlaka Chrome29.0 tarayıcısında gerçekleşmiştir."

b) "report.detail_report sayfasında ve Chrome29.0 tarayıcısında olan hatalar bazı durumlarda Murat Can Parlar kullanıcısı tarafından gerçekleştirilmiştir."

c) "sales.emptypopup_upd_subscription_contract sayfasında oluşan hatalar MSIE9.0 tarayıcısında gerçekleşmiştir. Bunlardan büyük çoğunluğu 2. Hata tipidir.

	WORKCUBE_NAME	PAGE	BROWSER_TYPE	RECORD_IP	KAYDEDEN	BUG_DETAIL
1	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	MSIE 9.0	192.168.18.39	Gökhan Acun	Error Executing Database Query. [Macromedia][SQLS...
2	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	MSIE 9.0	192.168.18.39	Gökhan Acun	Error Executing Database Query. [Macromedia][SQLS...
3	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	MSIE 9.0	192.168.18.39	Gökhan Acun	Error Executing Database Query.
4	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	MSIE 9.0	192.168.18.39	Gökhan Acun	Error Executing Database Query.
5	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	Chrome 29.0.1547.76	192.168.18.51	Murat Can Parlar	Variable order_back_demand_total_11_9_2013 is un...
6	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	Chrome 29.0.1547.76	192.168.18.51	Murat Can Parlar	Variable order_back_demand_total_11_9_2013 is un...
7	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	Chrome 29.0.1547.76	192.168.18.51	Murat Can Parlar	Variable DAY_ORDER is undefined.
8	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	Chrome 29.0.1547.76	192.168.18.51	Murat Can Parlar	Variable TOTAL_20 is undefined.
9	ep.workcube	ep.workcube/index.cfm?fuseaction=product.list_pr...	MSIE 9.0	192.168.18.39	Gökhan Acun	Variable BROWSERDETECT is undefined.
10	ep.workcube	ep.workcube/index.cfm?fuseaction=project.projects	MSIE 10.0	192.168.18.75	Osman Selvi	Variable BROWSERDETECT is undefined.
11	ep.workcube	ep.workcube/index.cfm?fuseaction=product.list_pr...	MSIE 9.0	192.168.18.39	Gökhan Acun	Variable BROWSERDETECT is undefined.
12	ep.workcube	ep.workcube/index.cfm?fuseaction=product.list_pr...	MSIE 9.0	192.168.18.39	Gökhan Acun	Variable BROWSERDETECT is undefined.
13	ep.workcube	ep.workcube/index.cfm?fuseaction=project.projects	MSIE 10.0	192.168.18.75	Osman Selvi	Variable BROWSERDETECT is undefined.
14	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	Chrome 29.0.1547.76	192.168.18.51	Murat Can Parlar	Variable TOTAL_20 is undefined.
15	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	Chrome 29.0.1547.76	192.168.18.51	Murat Can Parlar	Invalid CFML construct found on line 272 at column 1...
16	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	Chrome 29.0.1547.76	192.168.18.51	Murat Can Parlar	Invalid CFML construct found on line 272 at column 1...
17	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	Chrome 29.0.1547.76	192.168.18.51	Murat Can Parlar	Invalid CFML construct found on line 272 at column 1...
18	ep.workcube	ep.workcube/index.cfm?fuseaction=objects.empty...	MSIE 9.0	192.168.18.50	Görkem Bulut	Element ACTION_ID is undefined in ATTRIBUTES.
19	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	MSIE 10.0	192.168.18....	Fatih Ayık	Invalid tag nesting configuration. A query driven query...
20	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	MSIE 9.0	192.168.18.17	Şeyma Santaş	Element POSITION_CODE is undefined in EP.
21	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	MSIE 10.0	192.168.18....	Fatih Ayık	Variable FINISDATE is undefined.
22	ep.workcube	ep.workcube/index.cfm?fuseaction=test.bombos	MSIE 10.0	10.0.0.15	Emin Çokyaşa...	Could not find the included template Emin/bos.cfm. N...
23	ep.workcube	ep.workcube/index.cfm?fuseaction=sales.emptypo...	MSIE 9.0	192.168.18.22	Esra Varlı	Error Executing Database Query. [Macromedia][SQLS...
24	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	MSIE 10.0	192.168.18....	Fatih Ayık	Error Executing Database Query. [Macromedia][SQLS...
25	ep.workcube	ep.workcube/index.cfm?fuseaction=sales.emptypo...	MSIE 9.0	192.168.18.22	Esra Varlı	Error Executing Database Query. [Macromedia][SQLS...
26	ep.workcube	ep.workcube/index.cfm?fuseaction=sales.emptypo...	MSIE 9.0	192.168.18.22	Esra Varlı	Error Executing Database Query. [Macromedia][SQLS...
27	ep.workcube	ep.workcube/index.cfm?fuseaction=sales.emptypo...	MSIE 9.0	192.168.18.22	Esra Varlı	Error Executing Database Query. [Macromedia][SQLS...

	WORKCUBE_NAME	PAGE	BROWSER_TYPE	RECORD_IP	KAYDEDEN	BUG_DETAIL
28	ep.workcube	ep.workcube/index.cfm?fuseaction=sales.emptypo...	MSIE 9.0	192.168.18.22	Esra Varlı	Error Executing Database Query. [Macromedia][SQLS...
29	ep.workcube	ep.workcube/index.cfm?fuseaction=test.bombos	MSIE 10.0	10.0.0.15	Emin Çokyaşa...	Variable SAAT2 is undefined.
30	ep.workcube	ep.workcube/index.cfm?fuseaction=test.bombos	MSIE 10.0	10.0.0.15	Emin Çokyaşa...	Variable SAAT2 is undefined.
31	ep.workcube	ep.workcube/index.cfm?fuseaction=ehesa	MSIE 9.0	192.168.18.15	Şenay Gargacı	Invalid list index 2. In function ListGetAt(list, index [, d...
32	ep.workcube	ep.workcube/index.cfm?fuseaction=report.detail_re...	MSIE 9.0	192.168.18.17	Şeyma Santaş	Element RECORDCOUNT is undefined in GET_POSI...
33	ep.workcube	ep.workcube/index.cfm?fuseaction=objects.empty...	MSIE 9.0	192.168.18.17	Şeyma Santaş	Error Executing Database Query. [Macromedia][SQLS...
34	ep.workcube	ep.workcube/index.cfm?fuseaction=objects.empty...	MSIE 9.0	192.168.18.90	Şafak Bilir	Element CANDIDATE_POS_1 is undefined in GET_E...
35	ep.workcube	ep.workcube/index.cfm?fuseaction=sales.emptypo...	MSIE 9.0	192.168.18....	Merve Kılınçar...	Error Executing Database Query. [Macromedia][SQLS...
36	ep.workcube	ep.workcube/index.cfm?fuseaction=sales.emptypo...	MSIE 9.0	192.168.18....	Merve Kılınçar...	Error Executing Database Query. [Macromedia][SQLS...
37	ep.workcube	ep.workcube/index.cfm?fuseaction=sales.emptypo...	MSIE 9.0	192.168.18....	Merve Kılınçar...	Error Executing Database Query. [Macromedia][SQLS...
38	ep.workcube	ep.workcube/index.cfm?fuseaction=objects2.empt...	MSIE 9.0	192.168.18....	Merve Kılınçar...	Invalid list index 2. In function ListGetAt(list, index [, d...

Şekil 5. 24.09.2013 tarihinde oluşan hataların veritabanı görünümü

```
"Sayfa=report.detail_report"->"Tarayıcı_Çeşidi=Chrome29.0.1547.76"->"Alan_Adı=ep.workcube"->"Kaydeden_IP=192.168.18.51"->"Kaydeden=MuratCanParlar";
"Sayfa=report.detail_report"->"Tarayıcı_Çeşidi=Chrome29.0.1547.76"->"Alan_Adı=ep.workcube"->"Kaydeden_IP=192.168.18.51";
"Sayfa=report.detail_report"->"Tarayıcı_Çeşidi=Chrome29.0.1547.76"->"Alan_Adı=ep.workcube"->"Kaydeden=MuratCanParlar";
"Sayfa=report.detail_report"->"Tarayıcı_Çeşidi=Chrome29.0.1547.76"->"Alan_Adı=ep.workcube";
"Sayfa=report.detail_report"->"Tarayıcı_Çeşidi=Chrome29.0.1547.76"->"Kaydeden_IP=192.168.18.51"->"Kaydeden=MuratCanParlar";
"Sayfa=report.detail_report"->"Tarayıcı_Çeşidi=Chrome29.0.1547.76"->"Kaydeden_IP=192.168.18.51";
"Sayfa=report.detail_report"->"Tarayıcı_Çeşidi=Chrome29.0.1547.76"->"Kaydeden=MuratCanParlar";
"Sayfa=report.detail_report"->"Kaydeden_IP=192.168.18.51"->"Alan_Adı=ep.workcube"->"Kaydeden=MuratCanParlar";
"Sayfa=report.detail_report"->"Kaydeden_IP=192.168.18.51"->"Alan_Adı=ep.workcube";
"Sayfa=report.detail_report"->"Kaydeden_IP=192.168.18.51"->"Kaydeden=MuratCanParlar";
"Sayfa=report.detail_report"->"Kaydeden=MuratCanParlar"->"Alan_Adı=ep.workcube";
"Tarayıcı_Çeşidi=MSIE9.0"->"Hata_Tipi=2"->"Alan_Adı=ep.workcube";
"Tarayıcı_Çeşidi=MSIE9.0"->"Hata_Tipi=2"->"Sayfa=sales.emptypopup_upd_subscription_contract"->"Alan_Adı=ep.workcube";
"Tarayıcı_Çeşidi=MSIE9.0"->"Hata_Tipi=2"->"Sayfa=sales.emptypopup_upd_subscription_contract";
"Tarayıcı_Çeşidi=MSIE9.0"->"Sayfa=sales.emptypopup_upd_subscription_contract"->"Alan_Adı=ep.workcube";
"Hata_Tipi=2"->"Sayfa=sales.emptypopup_upd_subscription_contract"->"Alan_Adı=ep.workcube";
"Tarayıcı_Çeşidi=Chrome29.0.1547.76"->"Alan_Adı=ep.workcube"->"Kaydeden_IP=192.168.18.51"->"Kaydeden=MuratCanParlar";
"Tarayıcı_Çeşidi=Chrome29.0.1547.76"->"Alan_Adı=ep.workcube"->"Kaydeden_IP=192.168.18.51";
"Tarayıcı_Çeşidi=Chrome29.0.1547.76"->"Alan_Adı=ep.workcube"->"Kaydeden=MuratCanParlar";
"Tarayıcı_Çeşidi=Chrome29.0.1547.76"->"Kaydeden_IP=192.168.18.51"->"Kaydeden=MuratCanParlar";
"Kaydeden_IP=192.168.18.51"->"Alan_Adı=ep.workcube"->"Kaydeden=MuratCanParlar";
```

Şekil 6. 24.09.2013 tarihinde oluşan hata birliktelikleri

Bu çalışmada, hata birliktelikleri tespit edilerek hatalar hakkında çıkarımlarda bulunulmuştur. Bu sayede hataların en çok hangi koşullarda oluştuğu tespit edilebilir ve bu koşullar düzeltilerek veya iyileştirilerek hatalar minimize edilebilirler. Böylece, daha stabil bir yazılım sistemi elde etmek mümkün olacaktır.

Gelecekteki çalışmalarda graf görsellerinin belli bir klasörde tutulması yerine bir dosya yönetim sistemi aracılığıyla istenilen zamanda istenilen tarihe ait görsel elde edilebilecek şekilde daha gelişmiş bir arşiv yapısında saklanması planlanmaktadır. Ayrıca, graf görselini otomatik olarak yorumlayabilecek bir yapı inşa edilmesi planlanmaktadır. Bu yapı, graf görselini yorumlayarak görsel hakkındaki önermeleri otomatik olarak sunabilecektir.

Bu çalışmadaki analizde sadece firma çalışanlarının sebep olduğu hatalar incelenmiştir. Gerçek uygulamada, ilgili sistemdeki kurumsal veya bireysel üyelerin oluşturdukları hatalar da loglanabilmektedir. Geliştirilen yazılımın kapsamı genişletilerek sistemdeki diğer birey ve öğelerin oluşturduğu hatalar incelenebilir. Bu sayede sisteme bağlı çalışan e-ticaret sitelerinde oluşabilen hataların ilişkileri analiz edilebilecektir. Bu alanlarda uğraş verecek araştırmacılara hata analizi uygulamasını web tabanlı bir yapıya getirilerek e-ticaret sitelerindeki hataları gerçek zamanlı analiz edebilecek hale dönüştürmeleri önerilebilir.

KAYNAKLAR

[1] http:// en.wikipedia.org/wiki/Data_mining, (Erişim tarihi: Ocak 2014) .

[2] Thuraisingham B., (2009) “Data Mining for Malicious Code Detection and Security Applications”, 2009. WI-IAT ‘09. IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, ISBN : 978-0-7695-3801-3, Milan.

[3] Bora S.P., (2011) “Data mining and ware housing”, 2011 3rd International Conference on Electronics Computer Technology (ICECT), ISBN: 978-1-4244-8678-6, Kanyakumari.

[4] Nagwani N. K., Bhansali A., (2010) “A Data Mining Model to Predict Software Bug Complexity Using Bug Estimation and Clustering”, 2010 International Conference on Recent Trends in Information, ISBN 13: 978-1-4244-5956-8, Kochi, Kerala.

[5] Breckel A., (2012) “Error Mining: Bug Detection Through Comparison with Large Code Databases”, 9th IEEE Working Conference on Mining Software Repositories (MSR), ISBN : 978-1-4673-1760-3, Zürich.

[6] Wijayasekara D., Manic, M., Wright, J.L., McQueen, M., (2012) “Mining Bug Databases for Unidentified

Software Vulnerabilities", 5th International Conference on Human System Interactions (HSI), ISBN : 978-1-4673-4498-2, Perth, WA

[7] Nagwani N. K., Verma S., (2012) "Predicting Expert Developers for Newly Reported Bugs Using Frequent Terms Similarities of Bug Attributes", 2011 9th International Conference on ICT and Knowledge Engineering (ICT & Knowledge Engineering), ISBN : 978-1-4577-2161-8, Bangkok.

[8] Gegick, M., Rotella, P., Tao Xie, (2010) "Identifying security bug reports via text mining: An industrial case study", 7th IEEE Working Conference on Mining Software Repositories (MSR), ISBN: 978-1-4244-6802-7, Cape Town

[9] Lamkanfi, A., Perez, J., Demeyer, S., (2013) "The Eclipse and Mozilla defect tracking dataset: A genuine dataset for mining bug information", 10th IEEE Working Conference on Mining Software Repositories (MSR), ISBN : 978-1-4799-0345-0, San Francisco, CA

[10] Nagappan, M., Vouk, M.A., (2010) "Abstracting log lines to log event types for mining software system logs", 7th IEEE Working Conference on Mining Software Repositories (MSR), ISBN : 978-1-4244-6802-7, Cape Town

[11] Lei-Yue Yao, Jian-Ying Xiong, (2009) "The research and implementation of a correlative degree mining algorithm based on IIS logs", 2009, GRC '09. IEEE International Conference on Granular Computing, ISBN: 978-1-4244-4830-2, Nanchang

[12] http://en.wikipedia.org/wiki/Apriori_algorithm, (Erişim tarihi: Ocak 2014).

[13] Schneider A., Walter S., Langer J., Heinkel U., (2006) "Automatic Visualization of Abstract System Specifications", QSIC 2006. Sixth International Conference on Quality Software, ISBN: 0-7695-2718-3, Pekin