



**ZAMAN SERİLERİNDE ANOMALİ
TESPİTİ VE ÜNİVERSİTE YÖNETİCİLERİ
İÇİN BİR UYGULAMA MODELİ**

Ferhat KAHVECİ
Doktora Tezi
Yönetim Bilişim Sistemleri Anabilim Dalı
Doç. Dr. Yusuf Ziya AYIK
2020
Her Hakkı Saklıdır

ATATÜRK ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI

Ferhat KAHVECİ

ZAMAN SERİLERİNDE ANOMALİ TESPİTİ VE ÜNİVERSİTE
YÖNETİCİLERİ İÇİN BİR UYGULAMA MODELİ

DOKTORA TEZİ

TEZ DANIŞMANI
Doç. Dr. Yusuf Ziya AYIK

ERZURUM – 2020



TEZ BEYAN FORMU

SOSYAL BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

BİLDİRİM

Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Uygulama Esaslarının ilgili maddelerine göre hazırlamış olduğum "ZAMAN SERİLERİNDE ANOMALİ TESPİTİ VE ÜNİVERSİTE YÖNETİCİLERİ İÇİN BİR UYGULAMA MODELİ" adlı tezin tamamen kendi çalışmam olduğunu ve her alıntıya kaynak gösterdiğimi taahhüt eder, tezinin kâğıt ve elektronik kopyalarının Atatürk Üniversitesi Sosyal Bilimler Enstitüsü arşivlerinde aşağıda belirttiğim koşullarda saklanmasına izin verdiğimi onaylarım:

Gereğini bilgilerinize arz ederim *.

- Tezinin/Raporumun tamamı her yerden erişime açılabilir.
 Tezinin/Raporumun makale için **altı ay**, patent için **iki yıl** süreyle erişiminin ertelenmesini istiyorum.

24.12.2020

Ferhat Kahveci
Aslı Islak İmzalıdır.

*** LİSANSÜSTÜ TEZLERİN ELEKTRONİK ORTAMDA TOPLANMASI, DÜZENLENMESİ VE ERİŞİME AÇILMASINA İLİŞKİN YÖNERGE**

.....
ÜÇÜNCÜ BÖLÜM

Çeşitli ve Son Hükümler

Lisansüstü tezlerin erişime açılmasının ertelenmesi MADDE 6– (1) Lisansüstü teze ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.

(2) Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internetten paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç imkanı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulunun gerekçeli kararı ile altı ayı aşmamak üzere tezin erişime açılması engellenebilir.

Gizlilik dereceli tezler MADDE 7– (1) Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.

(2) Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir.



SOSYAL BİLİMLER ENSTİTÜSÜ

Graduate School of Social Sciences

TEZ KABUL TUTANAĞI

SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Doç. Dr. Yusuf Ziya AYIK danışmanlığında Ferhat KAHVECİ tarafından hazırlanan bu çalışma 24/12/2020 tarihinde aşağıdaki jüri tarafından. Yönetim Bilişim Sistemleri Anabilim Dalı'nda Doktora Tezi olarak kabul edilmiştir.

Başkan: Prof. Dr. Sibkat KAÇTIOĞLU

İmza: Aslı Islak İmzalıdır.

Jüri Üyesi: Doç. Dr. Yusuf Ziya AYIK

İmza: Aslı Islak İmzalıdır.

Jüri Üyesi: Prof. Dr. M. Dursun KAYA

İmza: Aslı Islak İmzalıdır.

Jüri Üyesi: Doç. Dr. Abdulkadir ÖZDEMİR

İmza: Aslı Islak İmzalıdır.

Jüri Üyesi: Dr. Öğr. Üyesi Mustafa KESKİNKILIÇ

İmza: Aslı Islak İmzalıdır.

Prof. Dr. Sait UYLAŞ

Enstitü Müdürü

İÇİNDEKİLER

ÖZET	V
ABSTRACT	VI
ÖNSÖZ	VII
KISALTMALAR VE SİMGELER DİZİNİ	VIII
ŞEKİLLER DİZİNİ	XI
TABLOLAR DİZİNİ	XIV
GİRİŞ	1

BİRİNCİ BÖLÜM

ANOMALİ TESPİTİ

1.1. ANOMALİ KAVRAMI VE ÇEŞİTLERİ	5
1.2. ZAMAN SERİLERİNDE ANOMALİ TESPİTİ	8
1.3. ANOMALİ TESPİT YÖNTEMLERİ	12
1.3.1. Kümelemeye Dayalı Anomali Tespiti	13
1.3.2. Yoğunluğa Dayalı Anomali Tespiti.....	14
1.3.3. Destek Vektör Makinesi	15
1.4. ANOMALİ TESPİTİNDE BAŞARI ÖLÇÜMÜ	15
1.5. KURUMSAL METRİKLERİN İZLENMESİ	19
1.5.1. Veri Tabanı Yönetim Sistemleri.....	22
1.5.2. VTYS'lerde Kurumsal Metriklerinin İzlenmesi.....	24
1.5.3. Harici Veri Kaynakları	29
1.5.4. VTYS ile Harici Veri Kaynakları Arasındaki Farklılıklar	30
1.6. MEVCUT YAZILIMLAR VE SERVİSLER	33
1.6.1. Striim ve Confluent	34
1.6.2. Google Analitik	36

İKİNCİ BÖLÜM**ÜNİVERSİTE YÖNETİM FAALİYETLERİNDE BİLİŞİM SİSTEMLERİNİN
ROLÜ**

2.1. ÜNİVERSİTELERDE TEMEL KURUMSAL YAPI VE YÖNETİM.....	40
2.1.1. Yönetimin Fonksiyonları	42
2.1.2. Yönetim Düzeyleri	44
2.1.3. Yöneticinin Rollerini	46
2.1.4. Karar Alma Süreci	47
2.2. ÜNİVERSİTELERDE BİLİŞİM SİSTEMLERİ	48
2.2.1. Bilişim Sistemlerinin Gelişimi	48
2.2.2. Bilişim Sistemi Çeşitleri.....	52
2.2.2.1. Öğrenci Bilgi Sistemleri	52
2.2.2.2. Uzaktan Eğitim Sistemi	52
2.2.2.3. İnsan Kaynakları Bilgi Sistemleri.....	53
2.2.2.4. Muhasebe-Finansman Bilgi Sistemi	53
2.2.2.5. Demirbaş Takip Sistemi	54
2.2.3. Yönetim Düzeylerine Göre Bilişim Sistemleri.....	55
2.2.4. Üniversitelerde Bilişim Sistemi Tedariki ve Prototipleme.....	57
2.2.4.1. Prototip Yazılımın Testinin Yapılması.....	64
2.2.4.2. Prototip Yazılımın Gerçek Yazılıma Olan Faydaları	66
2.3. LİTERATÜR ÖZETİ	67
2.3.1. Kavramsal Literatür	67
2.3.2. Kullanım Alanları	71
2.3.2.1. Sanal Makine Sunucuları	71
2.3.2.2. Sensör Verileri	72
2.3.2.3. Sosyal Ağlar.....	74

2.3.2.4. Performans Karşılaştırma	76
2.3.2.5. Diğer Kullanım Alanları	78

ÜÇÜNCÜ BÖLÜM

TÜRKİYE'DEKİ ÜNİVERSİTE YÖNETİCİLERİNE YÖNELİK BİR UYGULAMA MODELİ ÖNERİSİ: KURUMSAL METRİK İZLEME SİSTEMİ (KMİS)

3.1. METODOLOJİ	80
3.1.1. Modelin Amacı ve Önemi	80
3.1.2. Uygulama Modeli ve Hipotezleri	81
3.1.2.1. Alt Hipotezler	83
3.1.3. Modelin Evreni ve Örneklemi	84
3.1.4. Veri Toplama Yöntemi ve Aracı	85
3.1.4.1. Veri Araçlarının Seçimi	87
3.1.4.2. Kullanıcı Bilgi İhtiyacının Belirlenmesi	88
3.1.4.3. Veri Toplama Arayüzü	89
3.1.5. Kullanıcı Paneli ve Veri Analiz Arayüzü	107
3.1.5.1. Yeni Grafik Talebi	109
3.1.5.2. Grafikler	110
3.1.5.3. Anomali Analizi	113
3.1.6. Model Testi, Başarı Ölçümü ve Değerlendirme	116
3.1.6.1. VTYS'ler ile Yapılan Test	116
3.1.6.2. Harici Veri Kaynağı ile Yapılan Test	119
3.1.6.3. Model Başarısının Ölçümü	125
3.1.6.4. Model Değerlendirmeleri	131
3.1.7. Modelin Sınırlılıkları	132
3.1.8. Modelin Varsayımları	133

3.2. YAPILAN DİĞER ARAŞTIRMALARDAN FARKI.....	134
3.3. MODELİN İŞ ZEKASI SİSTEMLERİNİN TEMEL ÖZELLİKLERİ İLE KARŞILAŞTIRILMASI.....	137
3.4. BULGULAR	139
3.4.1. Model Testi Bulguları.....	139
3.4.1.1. VTYS'ler ile Yapılan Test Bulguları.....	139
3.4.1.2. Harici Veri Kaynağına Erişim Testi Bulguları	142
3.4.2. Model Başarı Ölçümü Bulguları	144
3.4.3. Model Değerlendirme Bulguları.....	148
3.4.3.1. Birinci Alt Probleme İlişkin Bulgular.....	148
3.4.3.2. İkinci Alt Probleme İlişkin Bulgular	150
3.4.3.3. Üçüncü Alt Probleme İlişkin Bulgular	151
SONUÇ VE ÖNERİLER.....	154
KAYNAKLAR.....	159
EKLER.....	169
ÖZGEÇMİŞ.....	182

ÖZET**DOKTORA TEZİ****ZAMAN SERİLERİNDE ANOMALİ TESPİTİ VE ÜNİVERSİTE
YÖNETİCİLERİ İÇİN BİR UYGULAMA MODELİ****Ferhat KAHVECİ****Tez Danışmanı: Doç. Dr. Yusuf Ziya AYIK****2020, 182 sayfa****Jüri: Prof. Dr. Sibkat KAÇTIOĞLU****Doç. Dr. Yusuf Ziya AYIK****Prof. Dr. Muhammet Dursun KAYA****Doç. Dr. Abdulkadir ÖZDEMİR****Dr. Öğr. Üyesi Mustafa KESKİNKILIÇ**

Bu çalışma; kurumsal metrik verileri toplayabilen üniversite yöneticileri için bir uygulama modeli önermektedir. Modelin nitel ve nicel olarak incelenebilmesi için Kurumsal Metrik İzleme Sistemi (KMİS) adında prototip uygulama geliştirilmiştir. Modelin “ne derece var olduğuna” yönelik nicel olarak bilgi edinmek için KMİS iki tür test yapılmıştır. İlk testte Oracle, MySQL, PostgreSQL ve Microsoft SQL Server VTYS’lerine bağlanabilmesi ikinci testte ise Türkiye’deki üniversitelerin kurum dışı paydaş veri kaynaklarından olan Yükseköğretim Kurumu istatistik verileri kullanılmıştır. Doğruluk, duyarlılık, kesinlik, F-Ölçütü ve MCC ölçümlerine göre ortalama % 83 ve ROC eğrisine göre çok iyi derecede bir başarıya sahiptir. Modelin “varlığına ve anlamına” yönelik nitel olarak derinlemesine bilgi edinmek için yarı yapılandırılmış anket yapılmıştır. Konusunda uzman bireylerin görüşlerinin alınmasında etkili olan amaçlı örnekleme yöntemi ile üniversitelerde farklı birimlerde yöneticilik yapmış emekli akademisyenlerin görüşlerine başvurulmuştur. Farklı kurumsal veri kaynaklarını tek bir platform altında toplanabilmesi katılımcıların ilgisini çekmiştir. Katılımcıların beyanları doğrultusunda KMİS’de düzenlemeler yapılmıştır.

Anahtar Kelimeler: Anomali, kurumsal metrik, VTYS, veri izleme.

ABSTRACT**Ph. D. DISSERTATION****TIME SERIES ANOMALY DETECTION AND AN APPLICATION MODEL
FOR UNIVERSITY MANAGERS****Ferhat KAHVECİ****Advisor: Assoc. Prof. Yusuf Ziya AYIK****2020, 182 pages****Jüri: Prof. Dr. Sibkat KAÇTIOĞLU****Assoc. Prof. Yusuf Ziya AYIK****Prof. Dr. Muhammet Dursun KAYA****Assoc. Prof. Abdulkadir ÖZDEMİR****Asst. Prof. Dr. Mustafa KESKİNKILIÇ**

This work proposes an application model for administrators that can collect metric data of universities. In order to analyze the model qualitatively and quantitatively, a prototype application, called the Institutional Metric Monitoring System (KMIS), has been developed. Two types of tests were conducted to obtain quantitative information about “existence potential” of the model. In the first test; Oracle, MySQL, Microsoft SQL Server, and PostgreSQL connection, and the data, collected from Council of Higher Education in Turkey as external stakeholders, has been tested. KMIS has an average of 83% measurements accuracy, sensitivity, precision, F-Criterion and MCC. It also has a very good success according to the ROC curve. A semi-structured questionnaire conducted to obtain qualitative in-depth information about the “existence and meaning” of the model. With the purposeful sampling method, which is effective in obtaining the opinions of individuals who are expert with their subject, the opinions of retired academicians worked in different units of universities were consulted. Gathering different corporate data sources under a single platform attracted the participants. Arrangements made in KMIS in line with the statements of the participants.

Key Words: Anomaly, institutional metric, DBMS, data monitoring.

ÖNSÖZ

Bu araştırmanın planlamasında, yazımında ve raporlaştırılmasında katkı ve desteklerini esirgemeyen başta Prof. Dr. Sibkat KAÇTIOĞLU olmak üzere Prof. Dr. Muhammet Dursun KAYA, Doç. Dr. Abdulkadir ÖZDEMİR ve Dr. Öğr. Üyesi Mustafa KESKİNKILIÇ'a teşekkürlerimi sunmayı bir borç bilirim. Araştırmanın her noktasında değerli bilgi ve paylaşımlarını esirgemeyen, her noktada rehberlik edip, yeri geldiğinde aşk ve şevkimi artıran, sabırla yol gösterici olan kıymetli hocam, tez danışmanım Doç. Dr. Yusuf Ziya AYIK'a çok teşekkür ederim. Ayrıca bu araştırmayı yapmamda bana ilham kaynağı olan Yönetim Bilişim Sistemleri Bölümü hocalarıma da teşekkürlerimi sunuyorum.

Beni bu günlere getiren, okul hayatımdaki maddi sıkıntılara rağmen okumama vesile olan, var olma gayem canım annem Nebahat KAHVECİ, babam Hacı KAHVECİ ve kardeşlerim Rıfat KAHVECİ ve Serkan KAHVECİ'ye minnettarım. Bu çalışmayı yaptığım süre içerisinde ısrarla peşimde duran ve her aşamasında en büyük desteği gösteren ve bana uygun çalışma koşulları sunan en önemli değer varlığım eşim Sultan KAHVECİ'ye ve çalışırken bana hep yardımcı olan oğlum Vedat İsmail KAHVECİ ve Erdem KAHVECİ'ye teşekkürlerimi ve en derin sevgilerimi sunarım.

Erzurum 2020

Ferhat KAHVECİ

KISALTMALAR VE SİMGELER DİZİNİ

ANSI	: Amerikan Ulusal Standartlar Enstitüsü (American National Standards Institute)
ABD	: Amerika Birleşik Devletleri
ASP	: Etkin Sunucu Sayfaları (Active Server Pages)
AQL	: ArangoDB Sorgulama Dili (ArangoDB Query Language)
AVESİS	: Akademik Veri İzleme Sistemi
Bkz.	: Bakınız
DML	: Veri İşleme Dili (Data Manipulation Language)
DN	: Doğru Negatif
DP	: Doğru Pozitif
DSDM	: Dinamik Sistem Geliştirme Metodu (Dynamic Systems Development Method)
Ed.	: Editör
FDD	: Özellik Güdümlü Geliştirme (Feature-Driven Development)
FQL	: Fauna Sorgulama Dili (Fauna Query Language)
HTML	: Hiper Metin İşaret Dili (Hyper Text Markup Language)
HUG	: Hızlı Uygulama Geliştirme
IEEE	: Elektrik ve Elektronik Mühendisleri Enstitüsü (Institute of Electrical and Electronics Engineers)
IIS	: İnternet Bilgi Servisleri (Internet Information Services)
IP	: İnternet Protokolü (Internet Protocol)
ISO	: Uluslararası Standardizasyon Kurumu (International Organization for Standardization)
JSON	: JavaScript Nesne Gösterimi (JavaScript Object Notation)
KMİS	: Kurumsal Metrik İzleme Sistemi

LDAP	: Basit Dizin Eriřim Protokolü (Lightweight Directory Access Protocol)
LOF	: Yerel Aykırılık Faktörü (Local Outlier Factor)
MEB	: Milli Eđitim Bakanlıđı
NoSQL	: Sadece Yapısal Sorgulama Dili Deđil (Not only SQL)
OLAP	: Çevrimiçi Analitik İşlem (Online Analytic Processing)
ÖBS	: Öğrenci Bilgi Sistemi
PHP	: Kişisel Ana Sayfa (Personal Home Page)
RAD	: Hızlı Uygulama Geliřtirme (Rapid Application Development)
RFID	: Radyo Frekanslı Tanımlama (Radio Frequency Identification)
ROC	: Alıcı İşletim Karakteristik Eğrisi (Receiver Operating Characteristic Curve)
PL/SQL	: Prosedürel SQL (Procedural Language SQL)
SAP	: Sistem, Uygulama ve Ürün (System, Application & Products)
SAS	: İstatiksel Analiz Sistemi (Statistical Analysis System)
SQL	: Yapısal Sorgulama Dili (Structured Query Language)
SVM	: Destek Vektör Makineleri (Support Vector Machine)
SWOT	: Güçlü ve Zayıf yönler, Fırsatlar ve Tehditler (Strengts, Weaknesses, Opportunities, Threads)
TCL	: Muamele Denetim Dili (Transaction Control Language)
TUBİTAK	: Türkiye Bilimsel ve Teknolojik Arařtırma Kurumu
T-SQL	: Muame SQL (Transact-SQL)
XAMPP	: Çapraz(Corss) Platform, Apache, MySQL, Php ve Perl
XML	: Geniřletilebilir İşaretleme Dili (Extensible Markup Language)
URL	: Standart Kaynak Belirteci (Uniform Resource Locator)
VTYS	: Veri tabanı Yönetim Sistemi
YN	: Yanlıř Negatif

YP : Yanlıř Pozitif



ŞEKİLLER DİZİNİ

Şekil 1.1. Mesafeye dayalı anomaliler.....	6
Şekil 1.2. Mevsimsel anomaliler.....	7
Şekil 1.3. Bütünsel anomaliler.....	8
Şekil 1.4. Zaman serisi özelliği taşımayan verilerde anomali tespiti.....	9
Şekil 1.5. Dakikalık zaman dilimi grafiği.....	10
Şekil 1.6. Saatlik zaman dilimi grafiği.....	11
Şekil 1.7. Günlük zaman dilimi grafiği.....	12
Şekil 1.8. ROC eğrisinin gösterimi.....	18
Şekil 1.9. Kurumsal metrikleri izleme süreci.....	20
Şekil 1.10. Tetikleyici aktif (a) ve pasif (b) iken sorgu tamamlanma süresi.....	27
Şekil 1.11. Sorgu performansının karşılaştırılması.....	28
Şekil 1.12. Gerçek zamanlı veri toplama ve analiz aracı olarak Striim (Striim, 2016: 1)	35
Şekil 1.13. Google Analytics Uyarıları (Google, 2020: 1).....	36
Şekil 1.14. Google Analytics akıllı uyarı sistemi (Google, 2020: 1).....	38
Şekil 2.1. Üniversitelerin temel kurumsal yapısı.....	40
Şekil 2.2. Yönetim düzeyleri arasında hiyerarşik bilgi akışı (Yavuz, 2014: 18).....	44
Şekil 2.3. Yöneticilerin veri işleme sistemlerinden bilgi edinme süreci.....	55
Şekil 2.4. Bilişim Sistemleri arasındaki etkileşim.....	57
Şekil 2.5. Prototip yazılım geliştirme süreci.....	63
Şekil 2.6. Hipotez, prototip yazılım ve gerçek sistem arasındaki ilişki.....	65
Şekil 2.7. Prototip yazılımın gerçek hayatta kullanımının temsili gösterimi.....	66
Şekil 2.8. Gerçek zamanlı veri akışında anomali tespiti (Solaimani ve diğ., 2016: 1481)	71
Şekil 2.9. Gerçek zamanlı verilerin kullanıcılarına dağıtımını sağlayan mimari (Fernandez ve diğ., 2015: 4).....	75
Şekil 2.10. Gerçek zamanlı veri toplama performans karşılaştırması (Kreps Narkhede ve Rao, 2011: 6).....	77
Şekil 2.11. Gerçek zamanlı veri dağıtım performans karşılaştırması (Kreps Narkhede ve Rao, 2011: 6).....	78
Şekil 3.1. Yöneticilerin bilişim sistemleri aracılığıyla üniversiteyi izlemesi.....	81

Şekil 3.2. Önerilen modelin temel süreçleri.....	82
Şekil 3.3. VTYS'lerin model açısından önemi.	83
Şekil 3.4. Araştırma değişkenlerinin model ile ilişkisi.	84
Şekil 3.5. Önerilen modelin veri işleme süreci.	87
Şekil 3.6. Veri işleme süreci için incelenen ve kullanılan araçlar.	88
Şekil 3.7. Kullanıcı bilgi ihtiyacının belirlenmesi.	89
Şekil 3.8. Yönetici Paneline erişim ve Yönetici Panelinde kullanılan modüller.	90
Şekil 3.9. Kullanıcı tipi seçimi sayfası.	90
Şekil 3.10. Yönetici Paneli ana sayfası.	91
Şekil 3.11. Kullanıcı tanımlama modülü.	92
Şekil 3.12. VTYS bağlantılarını yapılandırma modülü.	95
Şekil 3.13. Tablo erişimi sağlama modülü.	98
Şekil 3.14. Veri toplama işlemi modülü.	104
Şekil 3.15. VERİ tablosunun alanları.	105
Şekil 3.16. Tarih bilgisini içermeyen veri dizisi.	106
Şekil 3.17. Gerçek zamanlı veri izleme ve kayıt penceresi.	107
Şekil 3.18. Yönetici Paneline erişim ve Yönetici Panelinde kullanılan modüller.	108
Şekil 3.19. Uygulama kullanıcı paneli.	108
Şekil 3.20. Kullanıcı yeni grafik talebi modülü.	110
Şekil 3.21. Kullanıcı grafik modülü.	112
Şekil 3.22. Kullanıcı anomali analiz modülü.	114
Şekil 3.23. Kullanıcı anomali listesi.	115
Şekil 3.24. KMİS'in test edilmesi.	116
Şekil 3.25. SQL komutunun prototip uygulamaya girilmesi.	118
Şekil 3.26. Yükseköğretim Kurumu sistemindeki akademisyen sayısı.	119
Şekil 3.27. Yükseköğretim Kurumu sistemindeki akademisyen sayısındaki değişim.	120
Şekil 3.28. Harici veri kaynağı eklenmesi.	121
Şekil 3.29. Yükseköğretim Kurumu toplam akademisyen sayısı parametresine erişimin sağlanması.	122
Şekil 3.30. Yükseköğretim Kurumu toplam akademisyen sayısı verisinin toplanma işleminin girilmesi.	123
Şekil 3.31. Yükseköğretim Kurumu toplam akademisyen sayısı verisinin izlenmesi.	124

Şekil 3.32. Yükseköğretim Kurumu toplam akademisyen sayısı verisinin kaydı.	125
Şekil 3.33. Dakikalık zaman aralığı düzeyinde anomalilerin tahmin aralığına olan birim mesafesine göre frekansı.	126
Şekil 3.34. Saatlik zaman aralığı düzeyinde anomalilerin tahmin aralığına olan birim mesafesine göre frekansı.	127
Şekil 3.35. Günlük zaman aralığı düzeyinde anomalilerin tahmin aralığına olan birim mesafesine göre frekansı.	127
Şekil 3.36. Uygulama anomali tespiti ROC eğrisinin gösterimi.....	130
Şekil 3.37. Anomali tespiti.....	139
Şekil 3.38. Anomalilerin üç seviyede sınıflanması.....	141
Şekil 3.39. Tespit edilen anomalinin ayrıntılı bilgisi.	142
Şekil 3.40. Yükseköğretim Kurumu günlük toplam akademisyen sayısı	143
Şekil 3.41. Yükseköğretim Kurumu günlük toplam akademisyen sayısı	144
Şekil 3.42. Anomalilerin zaman aralığına göre yoğunlukları.	145
Şekil 3.43. Anomalilerin zaman dilimine göre tahmin aralığına uzaklık (%)-frekans (%) grafiği.	146
Şekil 3.44. Tahmin uzaklığı ve frekans arasındaki ilişkiyi sağlamadaki başarısının ROC eğrisi ile gösterimi.....	147
Şekil 3.45. Anomali tespiti başarı ölçüm sonuçları.	148

TABLolar DİZİNİ

Tablo 1.1. Anomali tespiti aracının geçerliliğini ölçen karmaşıklık matrisi (Juba ve diğ., 2015: 1).	15
Tablo 1.2. VTYS’lerde özet veri kaydı gerçekleştiren SQL sorgu örnekleri.	30
Tablo 1.3. Farklı harici veri kaynaklarındaki veri formatı örnekleri.....	32
Tablo 2.1. Anomalilere sebep olan olaylar (Abuaitah, 2013: 35).	73
Tablo 3.1. Sistemde kullanılan tabloların isimleri ve kullanım amaçları.....	86
Tablo 3.2. Farklı VTYS bağlantı değişkeni örnekleri.	93
Tablo 3.3. Farklı VTYS veri tabanlarını listeleme sorgu örnekleri.....	96
Tablo 3.4. Farklı VTYS veri tabanı tablolarını ve harici veri kaynaklarındaki dizileri listeleme sorgu örnekleri.	96
Tablo 3.5. Genel amaçlı verileri listelemek için kullanılan döngü örnekleri.	97
Tablo 3.6. Farklı VTYS için tetikleyici silme komut örnekleri.	101
Tablo 3.7. Farklı VTYS veri tabanı tablo alanlarını listeleme sorgu örnekleri.....	102
Tablo 3.8. Farklı VTYS veri tabanı tablo alanlarını getirme döngü örnekleri.	103
Tablo 3.9. Farklı VTYS veri setini getirme döngüsü örnekleri.....	111
Tablo 3.10. Görev başlıklarına uygulanan örnek veri kuralları.....	117
Tablo 3.11. Veri kurallarının SQL ile uygulanması.	117
Tablo 3.12. Prototip uygulama verilerine göre karmaşıklık matrisi.....	128
Tablo 3.13. Bu araştırma ile yapılan diğer araştırmalar arasındaki benzerlikler ve farklar.	135
Tablo 3.14. İş zekâsı sistemlerinin temel özellikleri ve KMİS.	137
Tablo 3.15 Yöneticiler açısından uygulamanın faydaları konusunda frekans tablosu.	149
Tablo 3.16 Uygulama modeli fikri olarak eksik görülen yönleriyle ilgili görüşlerinin frekans tablosu.	150
Tablo 3.17 Uygulama modeli fikri olarak belirttiğiniz eksiklere çözüm önerileriyle ilgili görüşlerinin frekans tablosu.	151

GİRİŞ

Bilişim teknolojilerindeki gelişmeler birçok bilimsel araştırmaların kapsamını da hızlı bir şekilde genişletmektedir. Bu alanlardan biri de anomali tespittir. Anomali tespiti, veri setinde beklentilere ya da kalıplara uymayan verilerin tespit edilmesidir. Anomali tespit çalışmaları donanım ve yazılımıyla ilgili teknik hataların ve kötü niyetli davranışların tespitinde, sistem yük durumu, servis talep yoğunluğu ve finansal verilerin izlenmesi gibi birçok alanda yapılmaktadır. Ayrıca yönetsel fonksiyonların desteklenmesi amacıyla da kullanılmaya başlanmıştır. Bu amaç doğrultusunda çalışan sayısı, talep yoğunluğu, üretim kapasitesi, bir ürüne olan ilgi gibi bazı temel kurumsal metrikler yöneticilere sunulabilmektedir. Bu tür kurumsal metrikler zaman serisi özelliği taşıyan grafikler ile kullanıldığında iş süreçlerinin gerçek zamanlı olarak izlenebilmesini sağlamaktadır. Böylece yöneticiler gerekli durumlarda erken önlem almalarını sağlayacak bilgiye daha kolay ve zamanında erişebilmektedirler. Ancak, yöneticilerin bir ekranda kurumsal metrikleri sürekli izlemesi beklenemez. Bu nedenle, gerekli durumlarda yöneticiyi bilgilendiren uyarı sistemleri kullanılabildiği görülmektedir. Anomali tespiti gelişmiş uyarı sistemlerinin temelini oluşturmaktadır. Buna göre yöneticilere yönelik olarak tasarlanan bir anomali tespiti çalışmasından beklenen özellikler veri toplama, anomali tespiti ve uyarı olarak sıralanabilir;

Literatüre göre, ilgili veri toplama çalışmaları belirli bir veri kaynağına odaklanmaktadır. Yapılan çalışmalarda veri kaynağı olarak sanal makine sunucuları, sensör verileri, sosyal ağ platformları, veri toplama araçlarının performansı, web sayfalarının görüntülenme sayısı kullanılmıştır. Konuya farklı yönetim düzeylerinin bulunduğu bir kurumda bulunan yöneticiler açısından bakıldığında, farklı veri kaynakları bir arada kullanılabildiği görülmektedir. Kurumsal çerçevede düşünüldüğünde veriler kurum içi kaynaklı olabileceği gibi kurum paydaşlarının kurum dışı kaynaklardan sağladığı verilerden de oluşabilmektedir. Kurum dışı gerçek zamanlı veri aktarımı genellikle XML ve JSON veri formatlarında gerçekleştirilmektedir.

Üniversiteler, günlük iş süreçlerinin çoğunu çeşitli bilişim sistemleri aracılığı ile yürütmektedirler. Bu durum üniversiteler açısından büyük kolaylıklar ve avantajlar sağlamaktadır. Bilişim sistemleri sayesinde günlük işler elektronik ortamda gerçekleştirilebilmektedir. Kullanılan veriler ile ilgili tüm işlemlerin elektronik ortamda

sağlıklı bir şekilde yönetilebilmesi için veri tabanı yönetim sistemlerini kullanmaktadırlar. Kurum içi metrikler veri tabanı yönetim sistemlerinden (VTYS) kontrol noktası oluşturma, sorgu havuzu, sorgulama, günlük kayıtların analizi ve tetikleyici oluşturma yöntemleri kullanılarak toplanabilmektedir. Bu yöntemlerin her birinin kendine göre farklı zorlukları ve dezavantajları bulunmaktadır. Önde gelen dezavantajlardan birisi VTYS genel performansının olumsuz etkilenmesidir. Öne çıkan zorluk, her bir VTYS'nin farklı sürümü üzerinde çalışmalar yürütülmesinin gerekmesidir. Konuyla ilgili uzman görüşlerine göre bu yöntemlerden birini tercih ederken kullanılan VTYS'nin kullanım yoğunluğu dikkate alınarak yapılması gerektiği yönündedir. Birçok kurumun verilerinin bulunduğu veri merkezlerinde kullanılan VTYS'ler saniyeler içerisinde milyonlarca kaydı güncellemek durumunda kalabilmektedirler. Ancak, üniversitelerde işlem yoğunluğu veri merkezlerindeki kadar yoğun olmadığından, ortalama bir tetikleyicinin işlem tamamlama süresi 2 milisaniyenin altında olduğundan ve sadece gerekli görülen tablolara eklenerek işlem yapılmasından dolayı bu çalışmada tetikleyici kullanımının daha uygun olduğu görülmüştür. Ancak, Anadolu Üniversitesi, Atatürk Üniversitesi ve İstanbul Üniversitesi gibi açık öğretim fakültelerinin bulunduğu üniversiteler göz önünde bulundurulduğunda tetikleyiciye alternatif olarak sorgulama, veri dosyası kullanımı ve veri toplama sıklığı seçenekleri eklenmiştir.

Yönetimin fonksiyonları açısından planlama önemli bir konuma sahiptir. Planların yapılmasında, işlerin doğru zamanda planlanması açısından zaman boyutu göz önünde bulundurulur. Bu doğrultuda zaman serilerinde anomali tespiti yöneticiler açısından değerlendirilebilir. Zaman serilerinde anomali tespitinde genellikle denetimsiz makine öğrenmesine dayalı yöntemler kullanılmaktadır. Bu amaçla, kümelemeye, mesafeye, yoğunluğa ve destek vektör makinesine dayalı yöntemleri kullanılmaktadır. Literatürde konuyla yapılan çalışmalara göre anomali tespitinden sonra sistemin uyarı vermesi için destek vektör makineleri kullanıldığı görülmüştür.

Uyarı sistemlerinin verdiği her alarmin gerçekte karşılığının olması başarısının bir göstergesidir. Yöneticilere gereksiz yere yapılan her bir uyarı yöneticinin sisteme olan güvenini azaltır. Bu nedenle, uyarı sisteminin iyi derecede bir başarıya sahip olması öncelikli beklentilerden biridir. Bu başarının ölçümü için doğruluk, duyarlılık, kesinlik, F-Ölçütü, MCC (Matthews Correlation Coefficients) ve ROC (Receiver Operating

Characteristic Curve) hesaplamaları karmaşıklık matrisinden faydalanılarak yapılmaktadır. Karmaşıklık matrisi zaman serilerinin belirli bir kesitinden örneklem alınarak oluşturulmaktadır. Bu amaçla, 2019 Eylül ile 2020 Mayıs ayları arasında Google Trendler “öğrenci bilgi sistemi” teriminin Türkiye’de gerçekleştirilen web aranma yüzdelik oranı değerlerinden oluşan zaman serisi kesiti üzerinde gerçekleştirilmiştir.

Yöneticilerin sorumlu oldukları çalışanların ve birimlerin faaliyetlerinin aksamadan yürütülmesi hızlı çalışmaları ve hızlı karar verme özelliklerinin etkinliğine bağlıdır. Yöneticiler birçok çalışan ve birimle ilgilendiklerinden dolayı faaliyet alanları da çeşitlidir. Gerek bilişim sistemleri aracılığıyla gerekse de ara yazılımlar sayesinde yöneticiler ihtiyaç duydukları bilgiyi raporlar sayesinde elde etmektedirler. Ancak, bazı faaliyetlerin beklenti dahilinde gerçekleştiğini bilmek ve gerektiğinde anormal durumlarda bilgilendirilmeleri muhakkak ki çalışma ve karar verme etkinliklerini olumlu etkileyecektir.

Bu tezin amacı, üniversite yöneticilerine yönelik kurumsal metrikleri izleyebilen ve gerektiğinde uyarı veren bir uygulama modeli önermektir. Konuyla daha önce yapılan çalışmalarda çeşitli yazılım, araç ve teknikler kullanılmıştır. Ancak, bu çalışmalarda kullanılan yazılımlar belirli bir veri kaynağı ile sınırlı olduğundan ortaya çıkan sonuçların kapsamı da çok kısıtlıdır. Bu nedenle, mevcut yazılımların kullanılması yerine araştırma için yeni bir prototip uygulama geliştirilmiştir. Bu kapsamda 2 yıl üzerinde çalışılarak 3500 satır PHP kodundan oluşan Kurumsal Metrik İzleme Sistemi (KMİS) adı verilen bir prototip yazılım geliştirilmiştir. KMİS’in en önemli özelliği farklı veri kaynaklarını kullanabilmesidir. Bunun yanı sıra gerçek zamanlı veri işleme, grafik oluşturma ve uyarı özelliklerinin bulunmasıdır.

Bu tez dört bölüme ayrılmıştır. 1. bölümde anomali tespit ve literatürde yapılan benzer çalışmalar ayrıntılı olarak ele alınmaktadır. Çalışmanın kullanım alanıyla ilgili temel kavramlar 2. bölümde yer almaktadır. Bu kapsamda üniversite yönetim faaliyetlerinde bilişim sistemlerinin rolü bu bölümde bahsedilmektedir. Önerilen modelin amacı, kapsamı, kısıtları, yöntemi ve bulguları 3. bölüm içerisinde yer verilmektedir. Ayrıca bu bölümde KMİS’in özelliklerinin sunulan modele olan katkısı dikkate alınarak ayrıntılı bilgiler verilmektedir.

Bilindiği üzere, kurumsal çapta bir yazılım geliştirme çok karmaşık ve zaman alıcı bir dizi metotların kullanımını gerektiren bir ekip çalışmasıdır. Bu nedenle, tez kapsamında geliştirilen yazılım tüm özelliklere sahip bir yazılım projesi yerine model önerisi ve tasarım şeklinde Hızlı Uygulama Geliştirme (HUG) çalışmalarından olan prototipleme yoluyla gerçekleştirilmiştir. Prototipler büyük yazılım projelerinin fikir kaynağıdır. Gerçek hayatta büyük maddi zararlara ve sistemlerin servis dışı kalmasına sebebiyet verme gibi olumsuzlukları yaşamamak için sürekli düzeltme ve iyileştirme süreçleri uygulanır. Prototiplemeden elde edilen tecrübeler ile daha kapsamlı kurumsal yazılım projelerine gerçek bir ekip ve donanım planlaması ile daha kolay bir şekilde projelendirilebilmektedir.

Sunulan modelin nicel olarak değerlendirilebilmesi amacıyla KMİS istemci-sunucu mimarisine dayalı test ortamı kullanılmıştır. Sunucu olarak VMware yazılımına sahip Intel sunucu bilgisayar kullanılmıştır. Oracle, MySQL, Microsoft SQL Server ve PostgreSQL sanal veri tabanı sunucuları kurulmuştur. Yerel ağ bağlantısı için fiber optik ve kablo bağlantısına sahip Juniper ağ bağlantı cihazı kullanılmıştır. Ayrıca uygulamanın teknik olarak değerlendirmesi amacıyla VTYS, yazılım ve kurumsal uygulamalar konularında uzman kişilerin görüşlerine başvurulmuştur.

KMİS'in harici veri kaynağını kullanabilme özelliği ile Yükseköğretim Kurumu istatistik web sayfasında günlük olarak güncellenen özet akademisyen sayısı verilerinin kullanımı test edilmiştir. Ayrıca üniversitelerde çeşitli yönetim görevlerinde bulunmuş 30 emekli akademisyen ile uygulama modeli hakkında açık uçlu sorulardan oluşan yarı yapılandırılmış anket ile görüşleri alınmıştır.

VTYS ve harici veri kaynakları için JSON veri formatının kullanımı uygulama modelinin geliştirilmesinde odak noktaları olmuştur. Aynı zamanda bu çalışmanın kurumların ve bireylerin kendi kaynakları ve imkanları ile yeni ve değerli bir uygulama modeli geliştirmesine örnek teşkil edeceği düşünülmektedir. Ayrıca hayatımızın vazgeçilmez bir parçası haline gelen İnternet de gerek literatürün taranmasında gerekse de diğer araştırmacıların, uzmanların ve yöneticilerin tecrübe ve önerilerinden istifade edilmesi bakımından bu çalışma için önemli bir araştırma aracı olmuştur.

BİRİNCİ BÖLÜM

ANOMALİ TESPİTİ

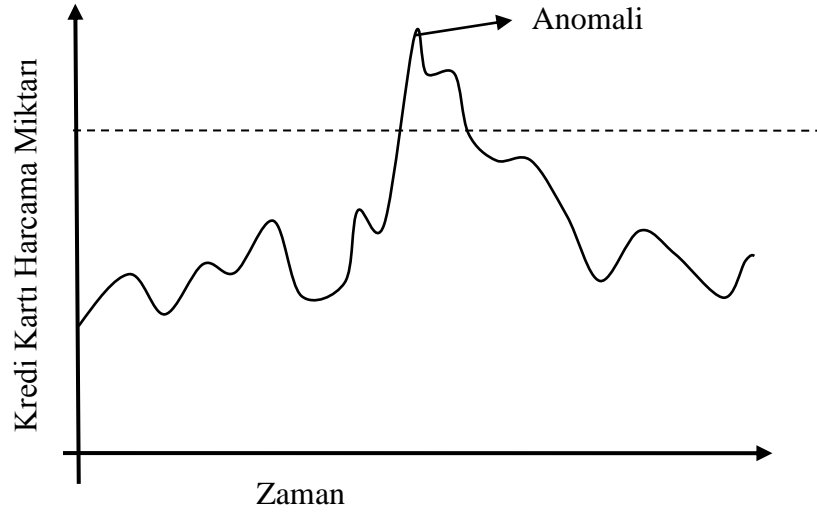
Günlük hayatta her an hem makineler hem de insanlar tarafından çeşitli türlerde veri oluşturulmaktadır. Bu veriler sayısal, metin, görsel, ses, vb. gibi çeşitli türleri olabilmektedir. Kurumlar açısından bakıldığında kurumsal ölçütlerin (metriklerin) raporlanması çoğunlukla sayısal olarak ifade edilmektedir. Öğrenci, akademisyen, derslik, ders, mezun, yeni öğrenci, not ortalamaları, vb. gibi birçok sayısal ifade üniversitelerde sıklıkla kullanılan özet verilerdir. Her bir sayısal veri ise farklı zaman aralıklarında güncellenmesi nedeniyle dinamik bir yapıya sahiptir. Sanal sınıf ortamındaki öğrenci sayısı dakikalık hatta saniyelik zaman dilimlerinde değişkenlik gösterirken, akademisyen sayısı ancak günlük zaman diliminde değişkenlik gösterebilir. Bu verilerin kendine uygun zaman dilimlerinde izlenmesi neticesinde problemler daha başlangıcında tespit edilebilir hale gelir. Bu sayede de gerekli önlemlerin alınmasında zaman kazanılmış olur. Zaman serilerine dayalı anomali tespiti çalışmaları bu tür kurumsal verilerin izlenmesinde kullanılabilir.

1.1. ANOMALİ KAVRAMI VE ÇEŞİTLERİ

Anomali kavramı, veri analiz çalışmalarında kullanılan veri setindeki gözlemler içerisinde beklenilene uymayan değerler olarak tanımlanmaktadır. Anomali terimi aykırılık, sıra dışı olay, uç değer kavramlarıyla eş anlamlıdır. Veri setindeki anomaliler tespit edildiğinde veri setinin kalan kısmı normal değerler olarak etiketlenir. Bu yönü ile anomali tespiti yöneticiler açısından iş süreçlerinin beklentiler dahilinde normal bir seyirde ilerlediğinin bilinmesi yönüyle önem arz etmektedir. Temel olarak üç çeşit anomali vardır (Chandola, Banerjee ve Kumar, 2009: 7):

- Mesafeye dayalı anomaliler
- Mevsimsel anomaliler
- Bütünsel anomaliler

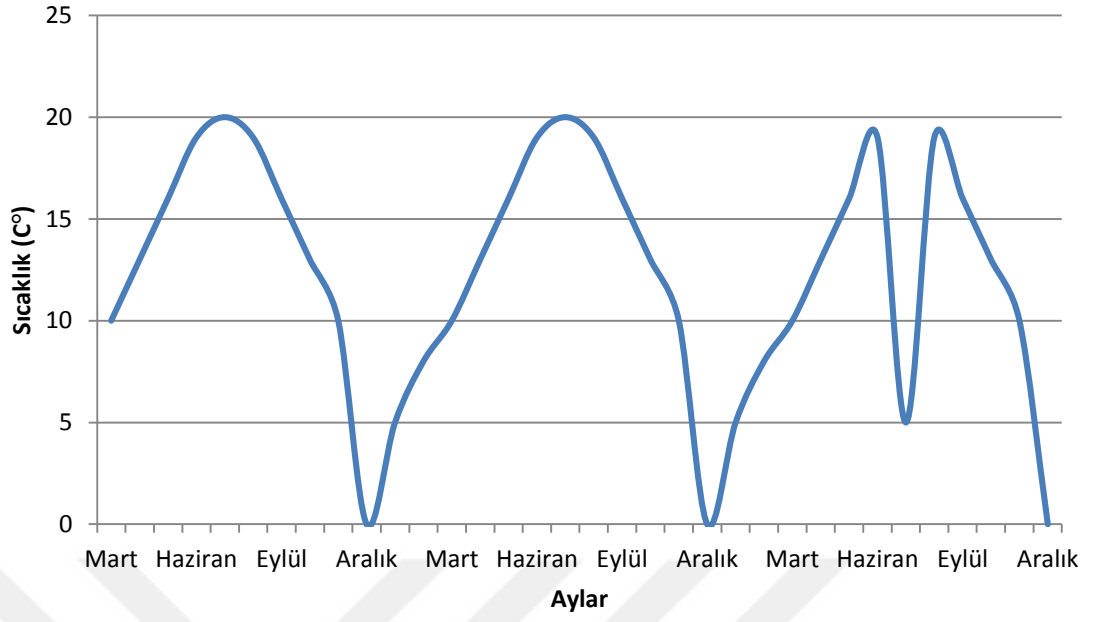
Mesafeye dayalı anomaliler: Anormal olarak görülen bir grup verinin göreceli olarak diğer verilerden uzak kalmasıdır. Harcama miktarına göre kredi kartı sahteciliğinin tespiti, buna bir örnek olarak gösterilebilir.



Şekil 1.1. Mesafeye dayalı anomaliler.

Şekil 1.1’de örnek bir veri setindeki değerlerin zaman serisi grafiği görülmektedir. Grafikte veri setindeki değerlerin kesik çizgi ile gösterilen sınırın içerisinde bulunması beklenmektedir. Ancak, bir kısım değerlerin diğer değerlerden uzaklaşarak beklentinin de üzerine çıktığı görülmektedir. Dolayısıyla, bu değerler Akoğlu, Tong ve Koutra (2015: 3)’nın çalışmalarında da belirttiği üzere anomali olarak değerlendirilmektedir.

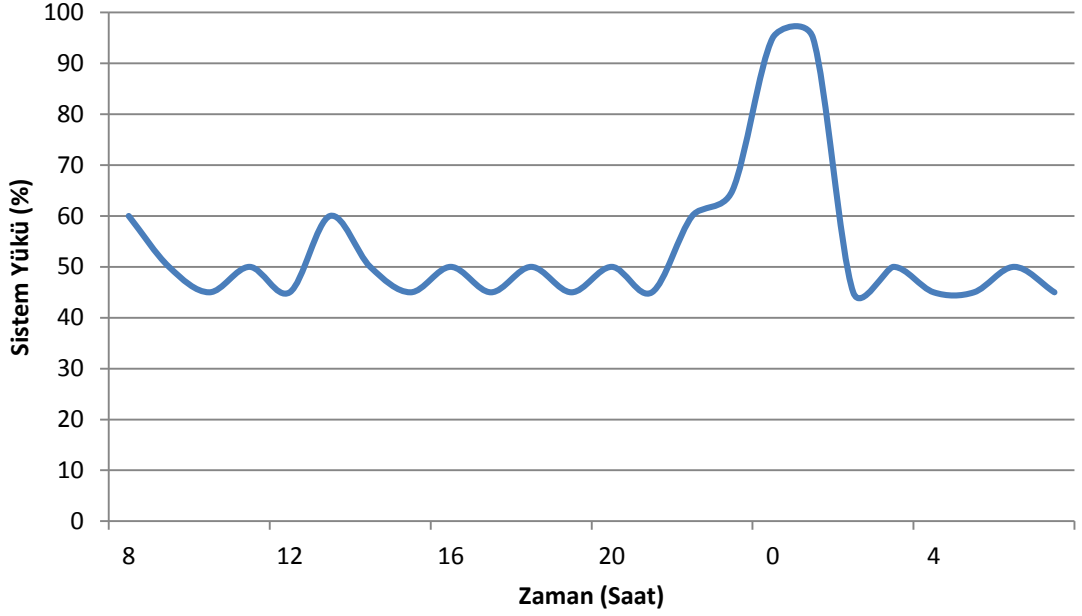
Mevsimsel anomaliler: Bu tür anomali belirli bir periyoda sahiptir. Örneğin; tatil zamanlarında günlük 1000 TL harcama normaldir, ancak diğer zamanlarda bu miktarda günlük harcamanın gerçekleşmesi, anormal olarak kabul edilmektedir.



Şekil 1.2. Mevsimsel anomaliler

Şekil 1.2'de örnek olarak mevsimsel sıcaklık değişimi görülmektedir. Kış aylarında sıcaklığın 5 C° olması normal karşılırken bu sıcaklığın yaz aylarında görülmesi anormal bir değerdir.

Bütünsel anomaliler: Bütünsel anomalinin ortaya çıkarılması bir dizi veri setinin yardımıyla mümkündür. Örneğin; sunucu bilgisayarına program yükleme ve bakım gibi toplu işlemler nedeniyle başka bir bilgisayardan veri kopyalanması olayı çok nadir gerçekleştiğinden, olası bir siber saldırı olarak görülebilir.

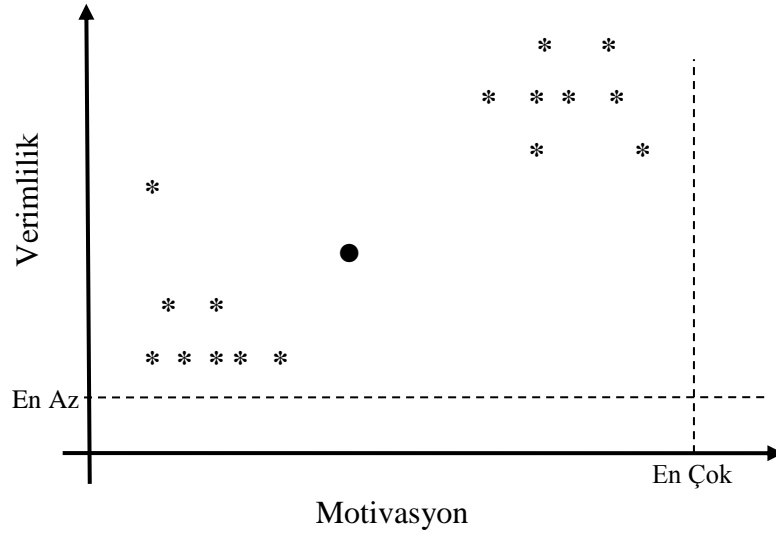


Şekil 1.3. Bütünsel anomaliler.

Şekil 1.3’de bir sunucu sistemin bir günlük yük durumunu gösteren örnek bir grafik verilmiştir. Grafiğe göre sistemin yük durumu gün içerisinde %45 ile %60 değeri arasında seyretmektedir. Ancak, gece saat 00:00’da sistem yük durumunda ani bir artış yaşanmıştır. Bu durum grafiğin bütünü dikkate alındığına anomali olarak görülmektedir. Bu anomalinin gerçekteki karşılığı siber bir saldırı olabileceği gibi sistem yöneticisinin zamanladığı toplu yedekleme işlemi de olabilir. Bu nedenle, tespit edilen anomaliyi meydana getiren nedenlerin dikkatli bir şekilde incelenmesi gerekmektedir.

1.2. ZAMAN SERİLERİNDE ANOMALİ TESPİTİ

Anomali tespitinde zaman serisi ve zaman serisi niteliği taşımayan veri setleri kullanılmaktadır (Mertz, 2018). Zaman serisi özelliği taşımayan veri setlerinde, gözlemlerin herhangi bir kümeye dâhil olup olmadığı incelenir. Şekil 1.4’de örnek olarak verimlilik ve motivasyon arasındaki ilişkiyi gösteren grafik verilmektedir.

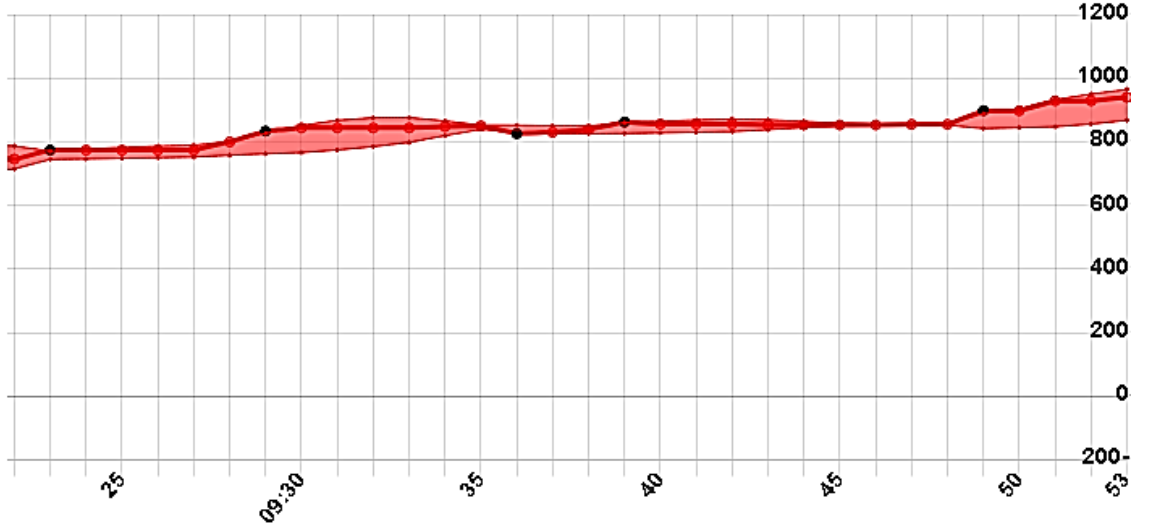


Şekil 1.4. Zaman serisi özelliği taşımayan verilerde anomali tespiti.

Grafikte verimliliği gösteren veriler motivasyonun az ya da çok olması durumuna göre iki ayrı küme oluşturmaktadır. Ancak, grafikte iki küme arasında kalan ve nokta ile gösterilen veri her iki kümenin de dışarısında yer almaktadır. Bu nedenle, bu veri anomali olarak değerlendirilmektedir.

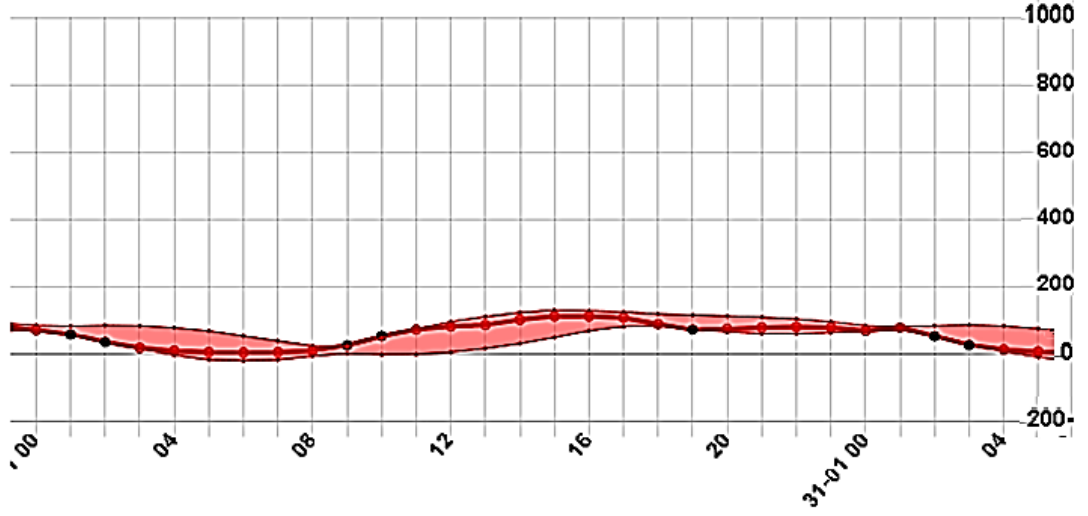
Bu çalışma, zaman serilerinde anomali tespitine odaklandığından zaman serisi grafiği daha ayrıntılı olarak ele alınmıştır. Zaman serisi grafiği belirli bir zaman aralığındaki zaman etiketine sahip verilerden oluşmaktadır. Anomali tespitinde kullanılan modeller sayesinde geçmiş verilere göre beklentilere uymayan gözlemler tespit edilmektedir (Samuelsson, 2016: 12).

Zaman serisi grafiklerinde kullanım alanına ve gereksinimlere göre çeşitli zaman dilimleri kullanılabilir. Bilimsel çalışmalarda örneğin, fizik alanında atom üzerinde yapılan araştırmalarda çoğunlukla mikro ve nano saniyeler gibi zaman dilimleri kullanılırken kurum ile ilgili uzun dönem kararlar için aylık ve yıllık gibi zaman dilimleri kullanılmaktadır. Bu çalışmada dakikalık, saatlik ve günlük olarak üç zaman dilimi kullanılmıştır. Bu grafiklerin temel özellikleri, örnek veri setleri kullanılarak KMİS'den elde edilen aşağıdaki grafikler üzerinde anlatılmıştır.



Şekil 1.5. Dakikalık zaman dilimi grafiği.

Şekil 1.5’de örnek bir dakikalık zaman dilimi grafiği görülmektedir. Grafik incelendiğinde, tahmin aralığının dışında değerlerin çok sık olduğu görülmektedir. Bunun nedeni, dakikalık verilerin hareketliliğinin (volatilité) yüksek olmasından kaynaklanmaktadır. Bu tür zaman aralığı az olan grafikler genelde operasyonel (işlemsel) seviyedeki yöneticiler tarafından günlük işlerin aksamaması için kullanılabilir. Grafikte kullanıcıların yoğun olduğu veya sistemin erişilemediği zamanlar tespit edilerek gerekli önlemlerin alınmasına yardımcı olabilecek bir grafik türü olarak verilmiştir.



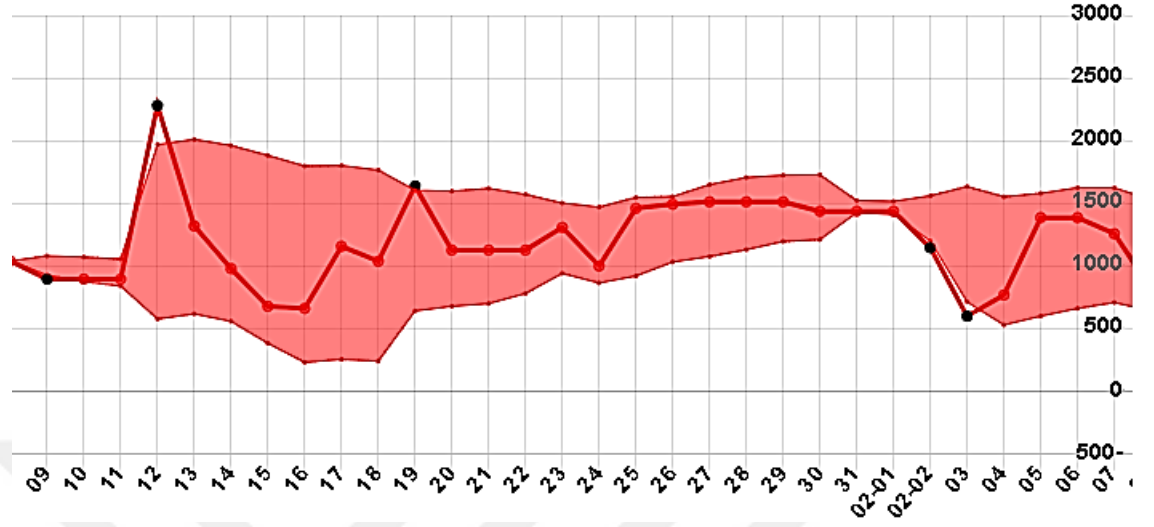
Şekil 1.6. Saatlik zaman dilimi grafiği.

Şekil 1.6'da örnek olarak saatlik zaman dilimi grafiği gösterilmektedir. Grafik yaklaşık bir günlük bir zaman dilimini kapsamaktadır. Bir önceki grafik olan dakikalık grafik ile aynı veri seti kullanılmaktadır. Karşılaştırıldığında dakikalık olarak tespit edilen tahmin aralığı dışına taşmaların tamamı saatlik grafikte gözlemlenmemektedir. Bunun nedeni, dakikalık olarak görülen bazı tahmin dışında kalan değerler saatlik olarak beklentiler dâhilinde olabilmesinden kaynaklanmaktadır. Başka bir deyişle saatlikteki veri değişimleri dakikalık grafiğe göre daha az olduğu görülmektedir.

Veri kaynağına erişilememe durumu için de bir farklılık söz konusudur. Dakikalık zaman diliminde, sisteme bazı zamanlar erişilememiş olabilir. Ancak, saatlik olarak düşünüldüğünde altmış dakikanın birinde dahi erişilmiş olsa saatlik zaman diliminde veri kaynağına erişilmiş kabul edilir. Saatlik grafikler üniversitelerin planlama, organize etme ve analiz gibi faaliyetlerin düzenlenmesinde yardımcı olabilir. Bu özelliğinden dolayı daha çok yönetim seviyesindeki yöneticilere yönelik olarak kullanılabilir.

Şekil 1.6 saatlik grafiğine bakıldığında saat 5:00 ve 6:00 zaman aralığı verinin en düşük değerlere sahip olduğu saatlerdir. Bu grafiğin sistemdeki aktif kullanıcı sayısının olduğu ve planlama açısından düşünüldüğünde sistemin bakım, yedekleme, yükleme, vb. gibi iş yükleri bu saatlerde yapılması daha uygun olacaktır. Aynı şekilde saat 14:00 ve 15:00 zaman aralığı verinin en yüksek değerlere ulaştığı saatlerdir. Bu saatler dikkate alınarak, sistemle sorumlu personelin çalışma saatleri, izin, vb. gibi durumlar yönetim

seviyesindeki yöneticilerin karar vermelerine yardımcı olabilir. Bu sayede sistem kullanıcılarının yoğunluğundan kaynaklanan bazı problemler engellenebilir.



Şekil 1.7. Günlük zaman dilimi grafiği.

Şekil 1.7'de örnek olarak günlük zaman dilimine göre veri dağılımını gösteren grafik verilmektedir. Grafik incelendiğinde daha çok günden güne alçalan, yükselen veya stabil kalan eğilimleri (trend) görmek mümkündür. Günlük grafik sayesinde daha dar bir zaman dilimine sahip olan saatlik ve dakikalık grafiklerin hangi değer aralığında değişebileceğini öngörmek mümkündür.

Günlük grafikler, uzun vadede planların yapılmasına yardımcı olan bilgi içerdiğinden çoğunlukla üst düzey yöneticiler için uygundur. Üst düzey yöneticiler bilindiği üzere, günlük operasyonel seviyedeki rutin işlere çok nadir olarak müdahale ederler. Bundan dolayı, operasyonel ve yönetim seviyesindeki bazı önemli gördükleri iş süreçlere de müdahil olmak isteyebilirler.

1.3. ANOMALİ TESPİT YÖNTEMLERİ

Veri işleme teknolojilerindeki gelişmeler, mevcut çalışmaların gelişmesine yardımcı olmakla birlikte yeni bilimsel çalışmalara ışık tutmaktadır. Bu sayede gelişmekte olan alanlardan biri de anomali tespittir. Anomali tespiti çalışmalarının temelinde, anormal verilerin ortaya çıkarılmasını sağlayan bir çeşit makine öğrenmesine dayalı teknikler bulunmaktadır. Kümeleme ve destek vektör makinaları en çok kullanılan yöntemlerdir. Bu yöntemlerde yoğunluğa dayalı algoritmalar olan k-en yakın

komşu ve göreceli veri yoğunluğu algoritmaları kullanılmaktadır (Chandola, Banerjee ve Kumar, 2009: 25).

1.3.1. Kümelemeye Dayalı Anomali Tespiti

Kümeleme, denetimsiz öğrenmede en çok kullanılan tekniklerden biridir. Kümeleme, veri setinde birbirine yakın değere ve özelliklere sahip verileri gruplamak için kullanılır. Sınıflamadan farklı olarak bu grupların isimleri ve sayıları önceden belli değildir. Veri setine yeni veriler eklendikçe gruplar değişkenlik göstermektedir. Bu özelliği nedeniyle sonuçlar dinamik bir yapıya sahiptir (Ceyhan, 2014: 128). Kümeleme işlemi yapılırken verilerin şu özellikleri göz önünde bulundurulur (Silahtaroglu, 2013: 163):

- Benzerlik
- Hiyerarşik
- Toplaşım / yoğunluk
- Bölünürlük

Veri setinde kümelerin oluşturulmasında kullanılan özellikleri ortaya çıkarmak için çeşitli algoritmalar kullanılmaktadır. Benzer verilerin kümelenmesinde şu algoritmalar kullanılmaktadır (Silahtaroglu, 2013: 164):

- Slinq
- Cure
- Chamelon

Hiyerarşik yapıya sahip kümeler oluşturmak için şu algoritmalar kullanılmaktadır (Silahtaroglu, 2013: 164):

- Birch
- Clucduh

Verileri yoğunluklarına göre bir araya toplamak için şu algoritmalar kullanılmaktadır (Silahtaroglu, 2013: 164):

- Dbscan
- Optics
- Denclue

- Grid
- String
- Dalga kümeleme

Veri setini bölerek kümeler oluşturmak için ise şu algoritmalar kullanılmaktadır (Silahtaroglu, 2013: 164):

- K-means
- K-medoids
- Pam
- Clara-ns
- Destek Vektör Makinaları (SVM – Support Vector Machine)

Kümeleme özellikle taksonomide büyük kolaylıklar sağlamaktadır. Taksonomi sayesinde gözlemler düzenli bir yapıya getirilmektedir. Böylece benzer özelliklere sahip gruplardan oluşan sınıflar hiyerarşik hale getirilebilmektedir (Han, Kamber ve Pei, 2012: 20).

1.3.2. Yoğunluğa Dayalı Anomali Tespiti

Literatürde bölgesel uyumsuzluk katsayısı (LOF - Local Outlier Factor) olarak da tanınmaktadır. Erişilebilirlik mesafesi da olarak adlandırılan uzaklık ölçümüne dayalıdır. Parametrik olmayan en basit öğrenme tekniğidir. Uzaklık ölçümlerinin benzerliğine göre veriler sınıflandırılmaktadır. Uzaklık ölçümünde, her bir verinin bölgesel merkezlerden uzaklığı dikkate alınmaktadır. Uzaklık ölçümü için en çok K-ortalama tekniğine dayalı algoritmalar kullanılmaktadır (Değerli, 2012: 15). Bu tekniğe göre verilerden “k” adedince küme oluşturulmaktadır. Bu kümelerin dışında kalan veriler anormal olarak etiketlenmektedir. Bu tekniğin varsayımına göre normal veriler yoğun bölgeler etrafında, anormal veriler ise daha uzak noktalarda gözlemlenmektedir. En yakın noktalar kümesi puanlama yöntemi ile ölçülebilmektedir. Veri tipine göre (kategorik, sayı) Öklid, Manhattan, Minkowski ve Hamming uzaklık ölçümünde en çok kullanılan algoritmalarındandır (Chandola, Banerjee ve Kumar, 2009: 26; Silahtaroglu, 2013: 186).

1.3.3. Destek Vektör Makinesi

Destek Vektör Makineleri (SVM – Support Vector Machine), anomali tespitinde kullanılan etkili yöntemlerden biridir. Destek Vektör Makineleri her ne kadar denetimli öğrenme ile ilişkili de olsa denetimsiz öğrenme tekniği ile de anomali tespiti yapılmaktadır. Algoritma, veri setini kullanarak yeni veriler için değişken bir tahmin aralığı oluşturmaktadır. Tahmin aralığı, verilerin dinamik olarak kümelenmesini sağlayan sınırları belirlemektedir. Tahmin aralığının dışında kalan veriler anomali olarak kabul edilmektedir (Silahtaroglu, 2013: 51).

1.4. ANOMALİ TESPİTİNDE BAŞARI ÖLÇÜMÜ

Anomali tespitinde karşılaşılan en büyük zorluklardan biri, tespit edilen anormal verilerin ne kadarı gerçeği yansıttığıdır. Anormal olarak tespit edilenlerin içerisinde normal veriler de bulunabilir. Başarı ölçütündeki temel kabul anomali tespiti için tasarlanan model ya da uygulama gerçeği yansıttığı oranda başarılı sayılmasıdır (Juba ve diğ., 2015: 1).

Anomali tespiti ile veri setindeki her bir değer için tahmin yöntemi ile normal ya da anormal veri etiketlemesi yapılmaktadır. Başka bir deyişle veri seti pozitif ve negatif olmak üzere iki temel sınıfa kategorize edilmektedir. Ancak, bu iki temel sınıfta yapılan tespitin gerçekte doğru olup olmaması faktörü göz önünde bulundurulduğunda toplam sınıf sayısı dörttür. Anomali tespitinin başarısının nicel olarak ölçümü için dört sınıfı temsil eden Tablo 1.1’de görülen karmaşıklık matrisi kullanılmaktadır (Mertz, 2018).

Tablo 1.1. Anomali tespiti aracının geçerliliğini ölçen karmaşıklık matrisi (Juba ve diğ., 2015: 1).

		Tespit Edilen	
		Hayır	Evet
Gerçekte	Hayır	Yanlış Negatif (YN)	Yanlış Pozitif (YP)
	Evet	Doğru Negatif (DN)	Doğru Pozitif (DP)

Tablo 1.1’de pozitif ve negatif olarak iki esas küme bulunmaktadır. Pozitif küme tespit edilen anormal verileri temsil etmektedir. Negatif küme ise anormal olmayan beklenti dahilindeki verilerdir. Buna göre;

Doğru pozitif (DP): Anormal olarak tespit edilen veriler gerçekte doğrudur.

Yanlış pozitif (YP): Anormal olarak tespit edilen veriler gerçekte yanlıştır.

Doğru negatif (DN): Normal olarak tespit edilen veriler gerçekte doğrudur.

Yanlış negatif (YN): Normal olarak tespit edilen veriler gerçekte yanlıştır.

Karmaşıklık matrisi, anomali tespitinde kullanılan uygulamanın performansının ölçülmesi için gerekli tüm değerleri içermektedir. Doğruluk, duyarlılık, kesinlik, F-ölçütü, MCC (Matthews Correlation Coefficients) ve ROC (Receiver Operating Characteristic) eğrisi başarımlı ölçümünde kullanılan temel hesaplama yöntemleridir (Coşkun ve Baykal, 2011: 3; Alan, 2014: 107; Çavuşoğlu ve Kaçar, 2019: 211):

Doğruluk, doğru pozitiflerin ve negatiflerin toplamının tüm veri setine oranıdır. Hesaplama kullanılan formül şu şekildedir (Coşkun ve Baykal, 2011: 3):

$$\text{Doğruluk} = \frac{DP + DN}{DP + YN + YP + DN} \quad (1)$$

Veri setinde pozitiflerin oranı uygulamanın başarısı hakkında genel bir fikir verebilir. Ancak, doğruluk hesaplamasında tüm verilerin aynı öneme sahip olduğu kabulü vardır. Anomali tespitinde negatifleri doğru bir şekilde tespit etmek kolay iken pozitifleri doğru bir şekilde tespit etmek çok zordur. Dolayısıyla, pozitiflerin sayısı az da olsa daha değerlidirler. Uygulamadan beklenen, tüm verileri gerçeğe en yakın pozitifleri ve negatifleri tespit edebilmesidir. Kullanılan uygulama yalnızca sayıca fazla olan ve tespit edilmesi kolay olan negatifleri doğru bir şekilde tespit ediyor olsa da gerçekte çok başarılı bir uygulama olduğu söylenemez (Juba ve diğ., 2015: 1).

Duyarlılık, uygulamanın doğru pozitifleri tespit etmede ne derece başarılı olduğunu göstermektedir. Hesaplama kullanılan formül aşağıda verilmektedir (Çavuşoğlu ve Kaçar, 2019: 211).

$$\text{Duyarlılık} = \frac{DP}{DP + YN} \quad (2)$$

Tespit edilen pozitiflerin gerçekte pozitif olarak etiketlenmesi gereken tüm verilere oranıdır. Duyarlılık yalnızca toplam pozitiflere odaklandığından uygulama bazı verileri normal olarak etiketlemesi gerekirken yanlışlıkla anomali olarak etiketliyor olabilir (Coşkun ve Baykal, 2011: 3; Çavuşoğlu ve Kaçar, 2019: 211).

Kesinlik, tespit edilen pozitiflerin ne derece doğru olduğunun göstergesidir. Kesinliğin hesaplanmasında kullanılan formül aşağıda verilmektedir (Çavuşoğlu ve Kaçar, 2019: 211).

$$\text{Kesinlik} = \frac{DP}{DP + YP} \quad (3)$$

Kesinlik hesaplaması, yalnızca pozitiflerin sayısına odaklandığından negatiflerin sayısını göz ardı etmektedir (Juba ve diğ., 2015: 1).

Kesinlik ve duyarlılık hesaplamaları, doğru olarak tespit edilen pozitiflerin oranına odaklanması nedeniyle birbirine benzemektedir. Duyarlılık, uygulamanın tüm pozitiflerin başarısını ölçerken kesinlik, tespit edilen pozitiflerin başarısına odaklanmaktadır. Duyarlılıkta uygulamanın verdiği yanlış uyarılar olası iken, kesinlikte uygulama eksik uyarılar olasıdır. Doğruluk hesaplamalarında, yanlış uyarıları ölçmek için kesinliğin de hesaplamada kullanılması gerekmektedir (Juba ve diğ., 2015: 1).

Görüldüğü üzere, duyarlılık ve kesinlik ölçümleri birbirinin eksiklerini gideren hesaplamaları içermektedir. Dolayısıyla, bu iki ölçümün bir arada kullanıldığı bir hesaplama daha başarılı bir sonuç verecektir. Bu iki ölçümü bir arada bulandıran hesaplama F-Ölçütüdür (Coşkun ve Baykal, 2011: 3; Çavuşoğlu ve Kaçar, 2019: 211).

F-Ölçütü, duyarlılık ve kesinlik ölçümlerinin harmonik ortalamasıdır. Böylece bu iki ölçüm tek bir hesaplamada birleştirilmektedir (Alan, 2014: 107).

$$F - \text{Ölçütü} = 2x \frac{\text{Kesinlik} \times \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \quad (4)$$

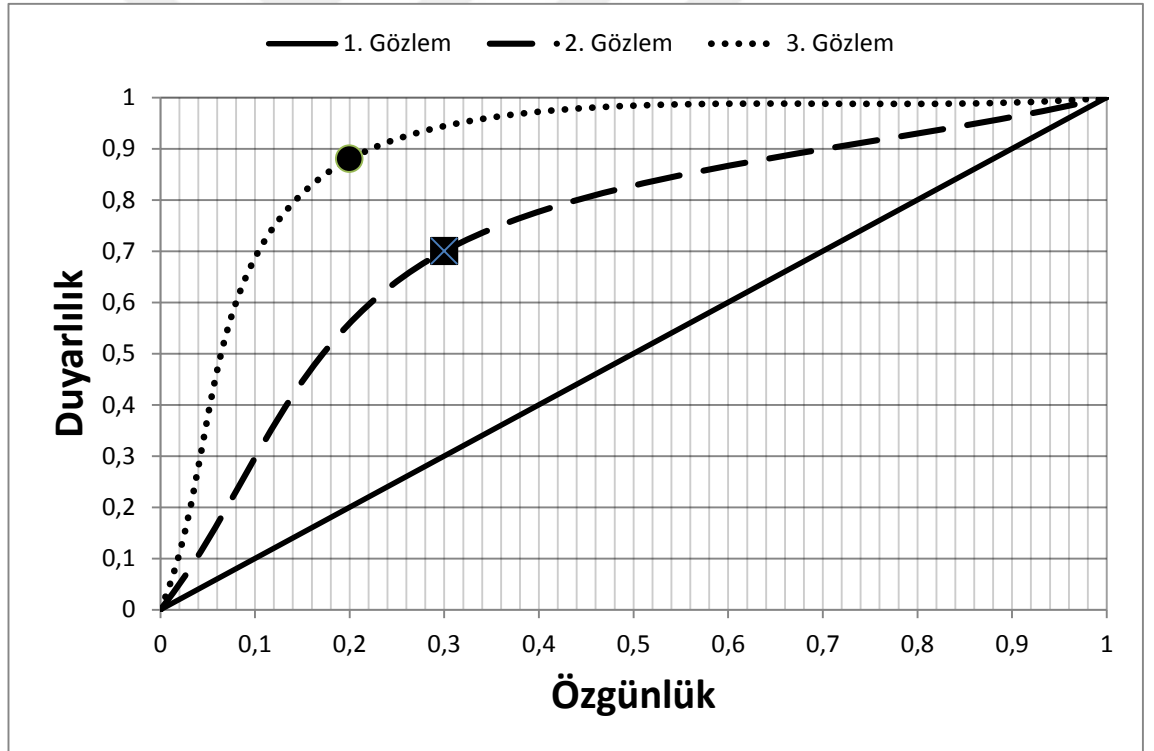
Matthews Correlation Coefficient (MCC), 1975 yılında biyokimyacı Brian W. Matthews tarafından bulunmuştur. Daha önce bulunan Karl Pearson's katsayısına benzemektedir. MCC, özellikle biyoenformatik alanında sıklıkla kullanılmaktadır. Karmaşıklık matrisindeki tüm verileri kullanabilmesi nedeniyle başarılı sonuçlar

vermektedir. MCC değerinin hesaplanmasında kullanılan formül aşağıda verilmektedir (Çavuşoğlu ve Kaçar, 2019: 211).

$$MCC = \frac{(DP \times DN) - (YP \times YN)}{\sqrt{(DP + YP) \times (DP + YN) \times (DN + YP) \times (DN + YN)}} \quad (5)$$

ROC Eğrisi (Receiver Operating Characteristic Curve - Alıcı İşletim Karakteristik Eğrisi), duyarlılık ve özgünlük hesaplaması sonucu elde değerlerin x ve y eksenlerine yerleştirilmesi sonucu ortaya çıkan eğrinin analiz edilmesidir. ROC Eğrisinin hesaplanmasında kullanılan özgünlük, uygulamanın negatif tespitinde ne derece başarılı olduğunu göstermektedir. Doğru negatiflerin tüm gerçek negatiflere oranıdır. Hesaplama kullanılan formül aşağıda verilmektedir (Çavuşoğlu ve Kaçar, 2019: 211).

$$\text{Özgünlük} = \frac{DN}{DN + YP} \quad (6)$$



Şekil 1.8. ROC eğrisinin gösterimi.

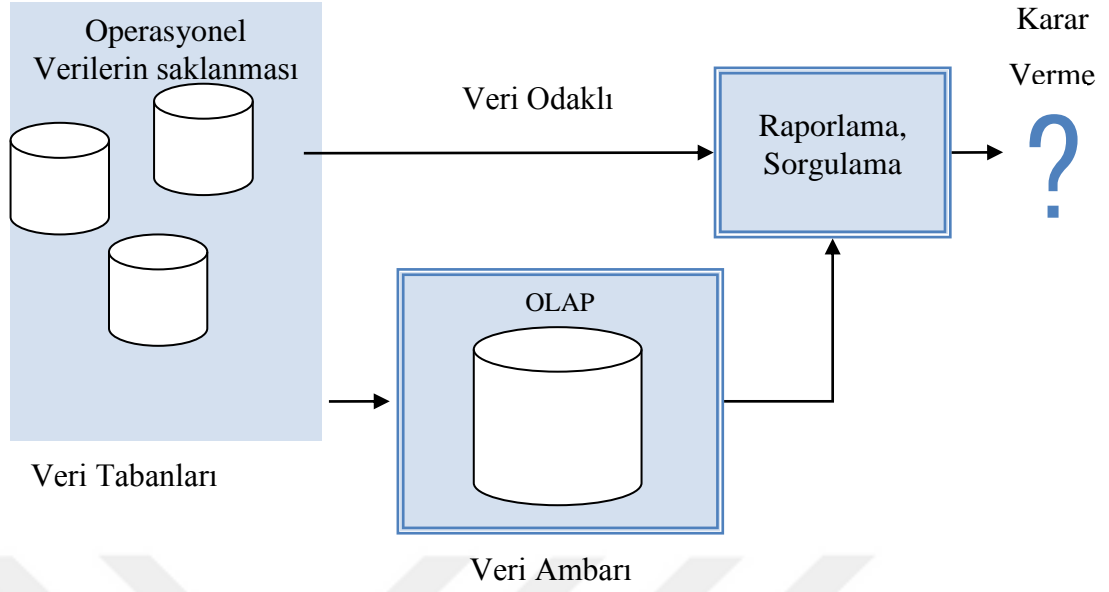
Şekil 1.8’de ROC eğrisinin üç farklı örnek gösterimi görülmektedir. Uygulama başarısı hakkında bir fikir elde edebilmek için, eğri çizgisinin altında kalan alana dikkat edilmesi gerekmektedir. Eğri altında kalan alan arttıkça, uygulama ya da modelin

başarısının da arttığını göstermektedir. Eğrinin, düz bir çizgiye yakın bir şekil alması uygulama ya da modelin başarısız olduğunun bir göstergesidir. (Çavuşoğlu ve Kaçar, 2019: 211).

ROC eğrisi, en üst sınır (peak) değerinin belirlenmesine de yardımcı olmaktadır. Şekil 1.8'de 1. gözlem doğrusal olduğundan dolayı ROC eğrisine göre gözlem sonuçları anlam ifade etmemektedir. Bu nedenle, herhangi bir üst sınır değeri bulunmamaktadır. Üst sınır noktası 2. gözlem eğrisinde kare ve 2. gözlem eğrisinde yuvarlak ile gösterilmiştir. Bu üst sınır değerine yakınlığı ile tespit edilen anomalilerin beklenti değerinden uzaklaşma mesafelerine göre farklı seviyelerde sınıflandırılmaktadır. Bu sayede anomali sınıfları tanımlanmakta ve ayırt edilmektedir (Han, Kamber ve Pei, 2012: 24). Böylece tespit edilen anomaliler yöneticilerin ilgilendikleri ayrıntı düzeyine göre sunulabilir. Veri setine yeni eklenen bir anomalinin hangi sınıfa dahil olabileceği önceden tahmin edilebilmektedir. Örneğin, uygulama tespit ettiği anomalileri yönetim düzeylerine göre üç seviyede sınıflandırabilir. Buna göre uygulama üst düzey yöneticiler için gerçekte olma ihtimali en yüksek olan anomalileri bir sınıfta toplayabilir. Yönetici istediği takdirde, daha alt sınıflardaki anomalileri de görüntüleyebilir. Aynı durum operasyonel düzeydeki yöneticiler için de geçerlidir. Operasyonel düzeydeki yöneticiler, günlük işlerin yürütülmesinde birçok bilgiye ihtiyaç duyabildiklerinden ayrıntı düzeyindeki anomaliler sunulabilir.

1.5. KURUMSAL METRİKLERİN İZLENMESİ

Kurumların temel varlıkları finansal ve beşeri varlıklar ile zihinsel becerileridir. Kurumsal metriklerin izlenmesi, bu temel varlıkların etkili bir şekilde kullanılmasını amaçlamaktadır. Bu doğrultuda kurum çevresinde meydana gelen metrik veriler, organize bir şekilde bir araya getirilmektedir (Han, Kamber ve Pei, 2012: 27; Purohit ve diğ., 2012: 461; Pazarçeviren, Zor ve Gürbüz, 2015: 76).



Şekil 1.9. Kurumsal metrikleri izleme süreci

Şekil 1.9’da kurumsal metrikleri izleme süreci gösterilmektedir. Veri tabanları, farklı veri kaynaklarından toplanan verilerin depolanmasını ve bütünleştirilmesini sağlar. Veri tabanlarındaki bütünleşik veriler kullanılarak önemli bilgileri elde edilebilir (Stonebraker ve diğ., 2005: 42). Veri tabanlarındaki verilere dayalı metriklerin incelenmesiyle yapılan ve veri madenciliği çalışmaları sonucunda elde edilen raporlar ve sorgulamalar veri odaklıdır. OLAP (Online Analytic Processing - Çevrimiçi Analitik İşlem) benzeri sistemlerin oluşturulabilmesi için veri tabanlarındaki veriler veri ambarlarına aktarılmaktadır (Pazarçeviren, Zor ve Gürbüz, 2015: 89).

OLAP, daha nasıl iyi olabilir, ileride ne olabilir gibi sorulara cevap veren karar destek sistemlerinin bir parçasıdır. Çeşitli analiz araçları OLAP verilerini kullanabilmekte ve toplanan veriler yararlı bilgiye dönüştürebilmektedir. Veri ambarlarına aktarılan verilerden daha iyi sonuçlar alabilmek için veri ambarları sürekli olarak iyileştirilmekte ve genişletilmektedir (Pazarçeviren, Zor ve Gürbüz, 2015: 79).

Gerek kurum içi gerekse de kurum dışından toplanan verilerle oluşturulan raporlar ve sorgular sayesinde karar verme davranışları iyileştirilebilmektedir. Kurumu ilgilendiren tüm verilerin toplanmasından, karar verme aşamasına kadar olan süreç içerisinde kullanılan tüm araç, yöntem ve uygulamalar kurumsal metriklerin izlenmesi kapsamında değerlendirilmektedir (Anameriç, 2005: 28).

Veri ambarlarına dayalı olarak gerçekleştirilen analizler karar destek sistemleri tarafından gerçekleştirilmektedir (Yavuz, 2014: 61). Kurumsal metriklerin izlenmesi yalnızca karar destek sistemine yönelik olarak, yöneticilerin kullanabileceği araçları içermemektedir. Veri tabanlarının sorgulanması ve çeşitli raporlama ihtiyaçları, veri madenciliği ve çok boyutlu veri analizi araçları, otomatik karar verme sistemleri, tavsiye algoritmaları, vb. gibi birçok karar destek sistemine yönelik uygulamalar olarak kurum çalışanları tarafından da kullanılabilir (Pazarçeviren, Zor ve Gürbüz, 2015: 82).

Veri ambarları, fiziksel tasarım olarak günlük işlemlerde kullanılan veri tabanları ile aynı teknik özellikleri sahip, ancak mantıksal tasarımı bakımından farklı yapıya sahiptirler. Veri ambarları, araştırma ve veri analiz çalışmaları için daha uygundur. Veri madenciliği aşamalarından olan yapılandırma aşamasından sonra elde edilen verilerin saklanmasında kullanılmaktadırlar. Veri tabanları, veri madenciliği çalışmalarında kullanılmak üzere de veri ambarı olarak tasarlanabilmektedirler. Yapılandırılmış verilerin veri ambarına aktarılma işlemi tek seferde gerçekleştirilir. Veri ambarını güncellemek için veri yapılandırma aşamasında, başka bir deyişle veri madenciliği aşamalarından geçmesi gerekmektedir. Veri ambarlarına direkt olarak ham veri girişi pek tercih edilmemektedir (Özkan Y., 2013: 16).

Veri ambarları, veri madenciliğinde sunum, değerlendirme ve karar verme çalışmalarının alt yapısını oluşturmaktadır. Bu çalışmaların daha iyi sunulabilmesi ve istenilen bakış açısıyla değerlendirilebilmesi için karar destek sistemleri geliştirilmiştir. Bu sistemler direkt olarak veri ambarını kullanılmaktadırlar. Bu sayede yönetici düzeyindeki çalışanların, daha etkin karar vermelerine yardımcı olmaktadır. Veri ambarlarının belirli bir konuya odaklı olması, bütünlük yapıya sahip olması, zaman boyutunun olması, sadece okunur türde olması ve kurum dışı verilere de yer vermesi temel özellikleridir (Özkan Y., 2013: 12).

Özetle, veri ambarları bir tür veri tabanıdır. Ancak, tasarım ve amacı yönüyle farklılık göstermektedir. Veri tabanlarında günlük işlemlerin yürütülmesi amaçlanırken, veri ambarları veri analizi için kullanılmaktadır. Burada şu husus da unutulmamalıdır ki; veri tabanları ve veri ambarları birbirinden tam olarak bağımsız değildirler. Çünkü, veri ambarlarında kullanılan veriler, çoğunlukla veri tabanlarından elde edilmektedir.

1.5.1. Veri Tabanı Yönetim Sistemleri

Bilişim sistemleri, alt yapısı itibari ile yazılım ve veri tabanlarından oluşmaktadır. Kullanıcılar bilişim sistemleri aracılığıyla veri tabanları ile etkileşimde bulunurlar. Bilişim sistemlerinin sahip olduğu kullanıcı arayüzü sayesinde işlemlerin kurumsal iş süreçleri çerçevesinde yürütülmesi sağlanmaktadır. Kullanıcılardan talep edilen veriler, sistem tarafından veri tabanlarına aktarılır. Veri tabanları bu verileri saklayarak kurumun veri deposu görevini görmektedir (Özkan Y., 2013: 5). Dolayısıyla, kurumsal metriklerin izlenmesinde veri tabanları önemli bir konuma sahiptir.

Veri tabanı, bir kurumun aktiviteleriyle ilgili bilgileri içeren birbiri ile ilişkili veriler topluluğudur (Silberschatz, Korth ve Sudarshan, 2011: 1). Örneğin, bir üniversite veri tabanı akademisyen, öğrenci, ders, derslik, öğrencilerin ders kaydı, açılan dersler gibi birbiri ile ilişkili verileri içermektedir.

Kurumsal veriler, bilişim sistemleri aracılığıyla çeşitli amaçlarla birçok iş sürecinde kullanılmaktadır. Verilerin veri tabanlarında saklanması, yetkili kullanıcıların kullanımına sunulması ve yöneticilere raporlanması bu süreçlerden bazılarıdır. Bu süreçlerde kullanılan verilerin yönetimi için VTYS'ler kullanılmaktadır. Bu nedenle, kurumsal metriklerinin gerçek zamanlı olarak izlenmesinde VTYS'ler önemli bir role sahiptir (Ayık ve Kahveci, 2019: 277).

Veri tabanı yönetim sistemi, kısa adı ile VTYS, büyük veri topluluklarının yönetimi için tasarlanmış yazılımlardır. Verilerin devasa boyutlara ulaştığı günümüzde, bu tür yazılımlara olan ihtiyaç ve bu yazılımların kullanımı giderek artmaktadır. VTYS'lerin klasik veri dosyalama sistemine göre birçok avantajı bulunmaktadır (Ramakrishnan ve Gehrke, 2003: 9). VTYS'ler, verilerin tutarlı, güvenilir ve erişilebilir bir şekilde saklanmasını sağlar. Genellikle bilişim teknolojileri alanında kendini kanıtlamış, tanınan uluslararası yazılım şirketleri tarafından geliştirilmektedirler. Günümüzde birçok kurum ve üniversite tarafından kullanılan VTYS'lerden en çok kullanılanları Oracle, MySQL, Microsoft SQL Server, Sysbase ve PostgreSQL'dir (Tecim ve Gökşen, 2009: 2243; Gülpek ve diğ., 2016; *Türkiye Bilimsel ve Teknolojik Araştırma Kurumu [TUBİTAK]*, 2018: 22).

VTYS'ler, veri saklama hiyerarşisine sahiptir. Bu hiyerarşi sırasıyla; veri tabanı-şema, tablo ve veri alanıdır. Veri tabanı diğer bir adıyla şema genellikle belirli bir

bilişim sisteminin kullanımına tahsis edilir. Böylece veriler diğer bilişim sistemlerinden izole edilmektedir. Bir VTYS içerisinde, farklı bilişim sistemlerinin kullanabileceği birçok veri tabanı bulunabilmektedir (Silberschatz, Korth ve Sudarshan, 2011: 15). Tablolar, satır ve sütunlardan oluşan iki boyutlu veri saklama birimidir. Satırlara kayıt, sütunlara da alan adı verilmektedir. Belirli bir kaydın ve alanın kesişim noktasına veri alanı denir. Veri alanı, VTYS veri saklama hiyerarşisindeki en küçük veri saklama birimidir (Kocaeli, 2010: 2).

Yapısı: VTYS'lerin en önemli yapı taşı veri tabanı motoru başka bir deyişle depolama motorudur. VTYS'lerde gerçekleştirilen oluşturma, okuma, güncelleme ve silme işlemleri veri tabanı motoru tarafından gerçekleştirilir. Kullanıcılar ve uygulamalar veri tabanı motorunu direk olarak kullanamazlar. VTYS, veri tabanı motoru ile kullanıcıları arasında bir arayüz görevi görmektedir. Bu nedenle, literatürde sıklıkla veri tabanı motoru yerine veri tabanı sunucusu veya VTYS terimleri kullanılmaktadır. VTYS'de kullanılan veri tabanı örneği terimi kullanılan veri tabanı motorunun hafızadaki yapısını temsil etmektedir. Veri tabanı motorlarından bazıları şunlardır: Aria, Falcon, InnoDB, MyISAM, TokuDB, WiredTiger, XtraDB ve RocksDB (Wikipedia, 2020: 1). Günümüz VTYS'lerin çoğu birden fazla veri tabanı motorunu kullanabilme özelliğine sahiptir. Örneğin, MySQL, InnoDB ve MyISAM veri tabanı motorlarını kullanmaktadır (Tocci, 2013: 17).

VTYS'lerin en önemli iki özelliği, veri işleme ve veri tanımlamadır. Veri işleme özelliğinin uygulanmasında sorgular kullanılmaktadır. Yapısal sorgulama dili (SQL - Structured Query Language), prosedürel SQL (PL/SQL - Procedural Language SQL), muamele denetim dili (TCL - Transaction Control Language), veri işleme dili (DML - Data Manipulation Language), muamele SQL (T-SQL - Transact-SQL), ArangoDB sorgulama dili (AQL - ArangoDB Query Language), basit izin erişim protokolü (LDAP - Lightweight Directory Access Protocol), fauna sorgulama dili (FQL - Fauna Query Language) sorguların yazılması için kullanılan sorgulama dilleridir (Silberschatz, Korth ve Sudarshan, 2011: 872; *Milli Eğitim Bakanlığı [MEB]*, 2012: 3; Özkan S., 2014: 26):

Bu sorgulama dillerinden en çok kullanılanı SQL dilidir. SQL dili ilk defa IBM şirketi tarafından eski adıyla SEQUEL olarak kullanılmıştır. Daha sonra 1986 yılında

Amerikan Ulusal Standartlar Enstitüsü (ANSI - American National Standards Institute) ve Uluslararası Standardizasyon Kurumu (ISO- International Organization for Standardization) tarafından günümüzde kullandığımız SQL dilinin ilk standartları yayımlanmıştır (Silberschatz, Korth ve Sudarshan, 2011: 57).

Veri tanımlama, veri tabanlarında yer alan tüm nesnelere tanımlanmasını sağlar. Tablolar, kullanıcılar, fonksiyon, prosedür, ilişkiler, yetkiler, vb. gibi birçok nesnenin kullanılabilmesi için VTYS’de tanımlı olması gerekmektedir. Nesne ilişki modelleri, tablolar ve veri sözlüğü veri tanımlamada kullanılmaktadır (Uysal, 2011: 14).

Çeşitleri: Normal ve dağıtık olmak üzere iki çeşit VTYS teknolojisi bulunmaktadır. Normal bir VTYS, tek bir sunucu sistemi üzerinde çalışır. Bütün işlemler ve veriler bu veri tabanı üzerinde gerçekleştirilir. Dağıtık VTYS çeşidinde birden fazla veri tabanı eşzamanlı olarak farklı sunucu sistemleri üzerinde çalışabilmektedir. Genellikle coğrafi olarak uzak mesafede şubelere ya da sistemlere sahip kurumlar tarafından kullanılmaktadırlar. Bu sayede, veri işlemleri en yakın dağıtık sunucuda yapılmaktadır. Dağıtık sunucu, belirli periyotlarla merkezi sunucu ile senkronizasyon sağlamaktadır (Ramakrishnan ve Gehrke, 2003: 936).

VTYS’nin dağıtık olarak kullanılmasında iki metot bulunmaktadır. Bunlar bölümlenme ve çoğaltmadır. Bölümlenmede veri tabanı kullanım amacına veya kullanılan uygulamalara göre birçok parçaya ayrılır. Her bir veri tabanı parçası farklı veriler ile ilgilenmektedir Bu parçalar merkezi bir VTYS tarafından kontrol edilir. Çoğaltma yönteminde, veri tabanı birçok parçaya ayrılmakta; ancak bu parçalar arasında belirli periyotlarla senkronizasyon sağlanmaktadır. Bölümlenme yönteminden farkı, bütün parçaların aynı verileri içermesidir. Bu sayede kullanıcılar kendilerine coğrafi olarak en yakın konumdaki dağıtık sunucuya erişim sağlayarak, verilere daha kısa sürede erişme imkânına sahip olurlar (Ramakrishnan ve Gehrke, 2003: 739).

1.5.2. VTYS’lerde Kurumsal Metriklerinin İzlenmesi

Kurumun veri tabanı tablolarında meydana gelen değişiklikler özet veri niteliğindeki metriklerin izlenmesinde temel referans noktasıdır. VTYS teknolojisindeki yeni gelişmeler, kurumsal metriklerin izlenmesi kolaylaştırmaktadır. Değişken yapıdaki veri seti üzerinde yapılan çalışmalar gerçek zamanlı veri analizi olarak

tanımlanmaktadır. (Savaş, Topaloğlu ve Yılmaz, 2012: 6; Solaimani ve diğ., 2016: 1492; Striim, 2016: 1; Confluent, 2018: 1).

Gerçek zamanlı kurumsal metriklerin izlenmesi hali hazırda güvenliğinin sağlanması, performansın iyileştirilmesi, saldırıların önlenmesi, erişim kontrolü, uygulamaların izlenmesi, vb. gibi çeşitli amaçlarla yapılmaktadır. VTYS'lerde bu tür bir izleme işleminin gerçekleştirilebilmesi için kontrol noktası oluşturma, sorgu havuzu, sorgulama, günlük kayıtların analizi ve tetikleyici kullanımı yöntemleri kullanılmaktadır (Stonebraker, Çetintemel ve Zdonik, 2005: 43; Natan, 2009: 287):

Kontrol noktası oluşturma: VTYS ile istemci arasında herhangi bir bağlantı noktasına iletişim izleyicisi yerleştirilmektedir. Bu iletişim izleyicisi VTYS hafızasına yerleştirilebileceği gibi sistemin ağ portlarından birine de yerleştirilebilmektedir. En büyük dezavantajı VTYS ile istemci arasındaki iletişimin yavaşlamasına sebep olmasıdır. Bazı durumlarda ağ sistemi, iletişim izleyicisinden olumsuz etkilenebilmektedir (Natan, 2009: 287).

Sorgu havuzu: VTYS gerçekleştirilen tüm sorguları bir sorgu havuzunda toplamaktadır. Sorgu havuzu, sorguların performansını iyileştirme amacıyla kullanılmaktadır (Natan, 2009: 287). Bu yöntemle tablolar üzerinde en çok kullanılan sorgular incelenebilmektedir. Ancak, tabloyla ilgili gerçek zamanlı bir veriye ulaşmak mümkün değildir. Bu durumun çözümü için tablolara ek bilgi alanı eklenmesi gerekmektedir. Bu bilgi alanı her bir satırın en son güncelleme zamanı, kaç defa güncellendiği gibi verileri saklamak amacıyla kullanılmaktadır. Ancak, tabloda yer alan bir satır silindiğinde bu bilgi alanı da silinmektedir. Ayrıca tüm tablolara bilgi alanı eklemek ve sürekli bu alanın güncel tutulması VTYS'ye ayrı bir işlem yükü getirmektedir.

Sorgulama: Sorgulama yönteminde özet veri toplamak üzere gruplama fonksiyonu içeren bir sorgu yazılır. Gruplama fonksiyonları count, max, min, average, sum, vb. gibi komutlarla tablolardan özet veri elde edilebilmektedir (MEB, 2012: 16). Gerçek zamanlı olarak kurumsal metriklerin izlenmesi açısından sorgulama yönteminin bazı dezavantajları bulunmaktadır. Bunlardan en önemlisi kurumsal metrik izleme sisteminin ilgili veri tabanını sorgulayabilmesi için sürekli aktif halde bulunması gerekmektedir. Bu sayede veri tabanına sorguyu gönderebilir ve sonucu kendi veri tabanına

kaydedebilmektedir. Sorgulamada VTYS haricinde bir sistemden sorgulama gönderildiğinden performans olarak yük oluşturmaktadır. Ancak, sorgulama işleminin sıklığı toplanacak olan verinin özelliğine göre düzenlenmesi performans açısından çok fazla bir yük getirmeyecektir. Örnek olarak, yayımlanan toplam makale sayısı verisi dakikalık olarak sorgulanması yerine saatlik ya da günlük olarak sorgulanabilir. Sorgulama zamanının VTYS yük durumunun az olduğu zamanların tercih edilmesi de ayrıca performans açısından katkı sağlayacaktır.

Günlük kayıtların analizi: VTYS'lerde gerçekleştirilen tüm işlemler günlük kaydı (log) adı verilen dosya parçacıklarına kaydedilmektedir. VTYS'ler bu dosyaları genelde veri kurtarma amacıyla ya da veri tabanını belirlenen bir zamana geri döndürmek için kullanmaktadırlar (Stonebraker ve diğ., 2005: 43). Günlük kayıtları VTYS'de gerçekleştirilen tüm sorguları ve verileri içermektedirler. Literatürde bu kayıtları inceleme çalışmaları günlük kayıt madenciliği (log mining) olarak gerçekleştirilmektedir (Han, Kamber ve Pei, 2012: 630). Oracle Golden Gate, Oracle XStream ve Oracle Log Miner gibi VTYS'lerin günlük kayıt madenciliği için özel olarak çıkardıkları sürümleri ve araçları mevcuttur. Bu araçlar sistemin bir parçası olarak tasarlandığından VTYS'ye ayrıca bir yük getirmemektedir.

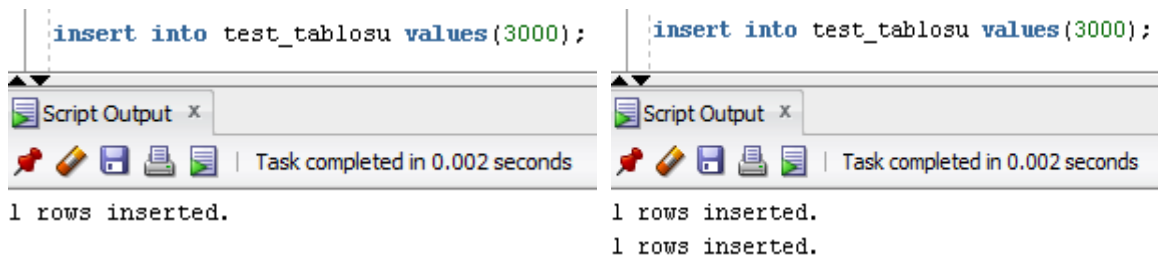
Günlük kayıtların en önemli olumsuzluklarından biri standart bir formata sahip olmamasıdır. Örneğin, her bir Oracle sürümünün günlük kayıt formatı ve içeriği farklıdır. Bu kayıtlar kurtarma amacıyla tutulmasından dolayı analiz çalışması için içerisinde gerekli veriyi bulmak ve çıkarmak oldukça karmaşık bir işlem gerektirmektedir (Ashdown, 2015: 40). Günlük kayıtların sağlıklı bir şekilde analiz edilebilmesi için her sürüme özel bir çalışma yapılması gerekmektedir. Ayrıca mevcut sürümde yapılan bir çalışma yeni bir sürümün çıkması ile sonlanabilmektedir.

Tetikleyiciler: Gerçek zamanlı verileri izlemede en çok kullanılan bir yöntemdir. Tetikleyici (trigger) VTYS içerisinde tanımlanmaktadır. Tetikleyici tanımlamalarında iki ana bölüm vardır. Bunlardan biri tetikleminin başlangıç aktivitelerini içermektedir. Diğeri ise tetiklendikten sonra yapılacak işlemi belirtmektedir. Bir tablo üzerinde herhangi bir değer değişikliği olduğunda (INSERT, UPDATE, DELETE) aktif hale gelmektedirler. Bu değişikliğin izlenmesi tetikleyici ile sağlanmaktadır. Yeni değerın kaydedilmesini sağlayan SQL sorusu tetikleyici içerisine yazılmaktadır. Bu SQL

sorgusu, yeni veriyi zaman değeri ve kaynak bilgisi ile birlikte belirlenen bir tabloya kaydedilmesini sağlayan INSERT komutunu içermektedir. Bu sayede ilgili tabloda değişiklik olduğunda tetikleyici aktif hale gelir ve tetikleyici içerisindeki SQL sorgusu çalıştırılır. Böylece verinin gerçek zamanlı olarak kaydedilme işlemi sağlanmaktadır (Gezgin, 2013: 58).

VTYS performansının tetikleyicilerin kullanımından etkilenmediği söylenemez. Tablolarda meydana gelen her bir değişiklik tetikleyici ile birlikte ek bir işlem yükü getirmektedir (Ramakrishnan ve Gehrke, 2003: 171). Ancak, bu işlem yükü üniversitelerde kullanılan standart bir VTYS açısından milisaniyeler seviyesinde olup, kullanıcılar bu gecikmeden neredeyse hiç etkilenmezler. Anlık olarak milyonlarca kaydın güncellendiği bir veri merkezlerinde kullanılan bir VTYS tablosu açısından gecikme ve iş yükü göreceli olarak çok daha yüksek olacaktır. Tetikleyicinin VTYS performansını ne derece etkilediğini nicel olarak ölçmek amacıyla örnek bir tablo üzerinde tetikleyici tanımlı bir tablonun INSERT işlemini ne kadar sürede tamamladığı ölçülmüştür. Test için kullanılan tabloyu oluşturmak için: CREATE TABLE "TEST_TABLOSU" ("ID" NUMBER). Test tablosundan alınan veriyi tetikleyici ile kopyalanacak tablo için: CREATE TABLE "HEDEF_TABLO" ("DEGER" NUMBER). Son olarak, TEST_TABLOSU'ndaki veriyi HEDEF_TABLO'ya aktaran tetikleyici:

```
CREATE OR REPLACE TRIGGER TEST_TRIGGER
AFTER INSERT ON TEST_TABLOSU
BEGIN
INSERT INTO HEDEF_TABLO
SELECT COUNT(*) FROM TEST_TABLOSU;
END;
```



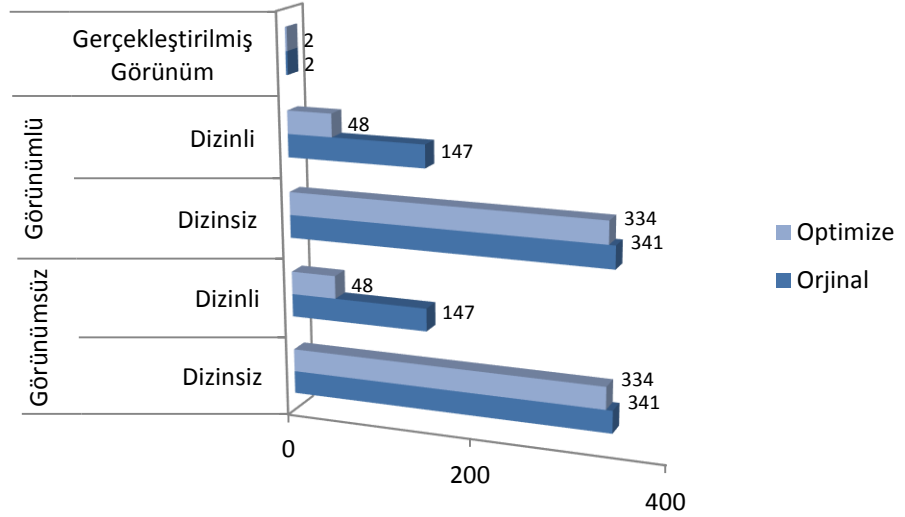
(a) Tetikleyici aktif iken TEST_TABLOSU'na veri eklendiğinde

(b) Tetikleyici pasif iken TEST_TABLOSU'na veri eklendiğinde

Şekil 1.10. Tetikleyici aktif (a) ve pasif (b) iken sorgu tamamlanma süresi.

Şekil 1.10’da tetikleyici aktif (a) ve pasif (b) durumuna göre INSERT komutunun tamamlanma süresi ölçülmüştür. Tetikleyici olmadan komutun tamamlanma süresi 2 milisaniye gibi çok kısa bir zaman sürmektedir. Tetikleyici kullanıldığında tamamlanma süresine etkisi milisaniyenin çok altında bir değer olduğundan 2 milisaniye olarak gerçekleşmiştir.

Tetikleyici kullanımının VTYS performansını olumsuz etkilememesi için kullanılabilir yöntemlerden biri de tetikleyici sorgusunda gerçekleştirilmiş görünüm kullanılmasıdır. Gerçekleştirilmiş görünüm sorgu sonuçlarını tutan ve belirlenen periyotlarla sorgu sonuçlarını güncelleyerek sorgu performansını artıran nesnelere dir. Bu sayede tetikleyici her çalıştığında gerçek tabloları sorgulamak yerine gerçekleştirilmiş görünümün sonuçlarını kullanmaktadır.



Şekil 1.11. Sorgu performansının karşılaştırılması.

Ayık ve Kahveci (2017: 1065) çalışmalarında gerçekleştirilmiş görünümün sorgu performansına etkisini incelemişlerdir. Çalışmada hafıza, kaynak kullanımı, girdi/çıkıtı, işlemci, başlangıç veri kümesinin büyüklüğü ve sonuç olarak dönen satır sayısına göre VTYS tarafından hesaplanan sorgu maliyet gözlemlenmiştir. 10 farklı maliyet hesaplama durumuna göre sorgu maliyeti değerleri Şekil 1.11’de verilmektedir. Gerçekleştirilmiş görünümün maliyetine göre diğer durumlara göre belirgin bir şekilde olumlu bir maliyet farkı bulunmaktadır.

1.5.3. Harici Veri Kaynakları

Günümüzde, sistemlerde kullanılan veri çeşitliliği giderek artmaktadır. Uygulamalar ve kullanıcılar tarafından oluşturulan düzenli verilerin yanı sıra özellikle sosyal medya araçlarının yaygınlaşması ile düzensiz yapıya sahip veriler giderek artmaktadır. Düzensiz yapıya sahip verileri belirli bir kalıba ya da formata yerleştirmek neredeyse imkânsızdır. Bu tür verileri veri tabanlarında daha önceden oluşturulmuş ve tasarlanmış olan tablolara kaydetmek çok güçtür. Bu problemin çözümü için VTYS'ler düzensiz verileri de kaydedecek şekilde kendilerini güncellemiştir. Bu tür verileri saklayabilen veri tabanı modellerine NoSQL veri tabanı denmektedir (*TUBİTAK*, 2018: 23). NoSQL veri tabanı ile ilgili olarak literatürde sıklıkla karıştırılan bir konu İngilizce NoSQL kelimesinde geçen “No” Türkçe karşılığı olan “Hayır” manasında değildir. Not Only kelimelerinin ilk harfleridir. Dolayısıyla, NoSQL kelimesinin İngilizcede tam olarak açılımı “Not Only Structured Query Language” şeklindedir. Türkçe karşılığı “Sadece İlişkisel Sorgulama Dili Değil” olarak hem ilişkisel hem de ilişkisel olmayan verileri sorgulama ve kaydetme yeteneğine sahip anlamındadır

VTYS'ler ilişkisel olmayan verileri normal bir tablo formatı yerine genellikle Genişletilebilir İşaretleme Dili (XML - Extensible Markup Language) ya da JavaScript Nesne Gösterimi (JSON - JavaScript Object Notation) formatında saklamaktadırlar. Bu formatın en önemli özellikleri istenildiği kadar genişletilebilmesi ve değişik türde verileri bir arada saklayabilmesidir. Bu sayede NoSQL veri tabanları, bir anlamda normal tablo düzeninden daha esnek bir yapıya sahiptirler (Munzert, Rubba, Meißner ve Nyhuis, 2015: 10).

XML ve JSON veri formatı sistemler arası veri alışverişinin sağlanmasında da kullanılmaktadır (Mohammed, Humbe ve Chowhan, 2016: 3). Bu amaçla kullanılan ara yazılımlar kurumlar arası kullanılan sistemler arasında köprü görevi görmektedir. Sistemler arası iletişimi sağlayarak veri alışverişini sağlarlar (Silberschatz, Korth ve Sudarshan, 2011: 1008). Ara yazılımlar web sayfalarına tümleşik olarak da kullanılmaktadır. Bu tür ara yazılıma Yükseköğretim Kurumu istatistik sayfası örnek olarak gösterilebilir. Üniversiteler kullandıkları Akademik Veri İzleme Sistemi (AVESİS) verilerini Yükseköğretim Kurumu istatistik sayfasından sağlamaktadırlar.

Uygulama Programlama Arayüzü (API - Application Programming Interface) ve kurumsal portalların kullanımı bu veri akışını sağlayabilmektedir (Ömürbek, 2003: 36).

1.5.4. VTYS ile Harici Veri Kaynakları Arasındaki Farklılıklar

Veri kaynaklarındaki çeşitlilik doğal olarak veri toplamada farklılıklara neden olmaktadır. Veri toplamadaki bu farklılıklar, VTYS'den elde edilen veri ve harici veri kaynaklarından elde edilen veri olmak üzere iki ana başlık altında incelemek mümkündür (Anameriç, 2005: 28).

VTYS'den elde edilen gerçek zamanlı veriler tek bir veri formatına sahiptir. Bunun sebebi verinin sorgu ile elde edilmiş olmasıdır. SQL sorguları verilerin istenilen yapıda kullanıcılarına sunmaktadır (Silberschatz, Korth ve Sudarshan, 2011: 890). Dolayısıyla, VTYS üzerinde sorgu çalıştırılabilen tüm veri tabanlarından yapılandırılmış tek bir veri formatında gerçek zamanlı veri akışı sağlanmaktadır.

Tablo 1.2. VTYS'lerde özet veri kaydı gerçekleştiren SQL sorgu örnekleri.

VTYS	Sorgu
Oracle	<pre>INSERT INTO public.VERI (KAYNAK, DEGER) VALUES ('pgişlem', (SELECT COUNT(*) SAYI FROM public.PGTABLE));</pre>
MySQL	<pre>INSERT INTO TEST.VERI (KAYNAK, DEGER) VALUES ('msislem', (SELECT COUNT(*) SAYI FROM deneme.SINIF));</pre>
Microsoft SQL Server	<pre>INSERT INTO TEST.VERI (KAYNAK, DEGER) VALUES ('Genel gelir gider-farkı verisi', (SELECT SUM(GELIR) - SUM(GIDER) SAYI FROM muhasebe.butce))</pre>
PostgreSQL	<pre>INSERT INTO VIS.VERI (KAYNAK,DEGER) VALUES('Sistem veri girişi',(SELECT COUNT(*) SAYI FROM VIS.VERI));</pre>

Tablo 1.2'de farklı VTYS'lerde gerçek zamanlı veri akışının kaydedilmesi için kullanılan SQL sorguları verilmiştir. Sorguların geneline bakıldığında yapı olarak benzerlik göstermektedirler. Tümü INSERT komutunu kullanarak "KAYNAK" değeri

için veri kaynağının adı ve “DEGER” için SELECT sorgusundan elde edilen değer ile kaydedilmektedir.

Tarih verisi VTYS tarafından eklenebilmektedir. Gerçek zamanlı verilere dayalı olarak gerçekleştirilen çalışmalarda verinin oluştuğu zaman etiketinin kullanılması daha avantajlıdır. Ancak zaman etiketinin bulunmadığı çoğu harici veri kaynaklarında zaman etiketi olarak veriyi toplayan sistemin zamanı kullanılmaktadır. Bu ise verinin ortaya çıktığı zaman ile kaydedildiği zaman arasında saniyeler seviyesinde de olsa farklılık ortaya çıkarabilmektedir.

Görüleceği üzere, VTYS’lerden elde edilen veri akışının kaydedilmesi benzer komut yapısı içerisinde tanımlandığından dolayı veri formatı da birbirine benzemektedir. Gerçek zamanlı veri akışlarından elde edilen veri formatlarındaki en büyük farklılık, harici veri kaynaklarında görülmektedir. Her bir veri kaynağı kendine özgü bir veri formatına sahiptir. Bu farklılığın en önemli sebebi veriyi paylaşım açan kişi ya da kurumun, veri kullanıcılarının hangi tür veri formatına ihtiyacı olduğunu bilgisine sahip olmamasıdır. Bu nedenle, bu tür veriler her ne kadar standart bazı formatlara sahip olsa da içerisindeki verilerin sistemler tarafından kullanılabilmesi açısından farklılıklar bulunmaktadır.

Tablo 1.3. Farklı harici veri kaynaklarındaki veri formatı örnekleri.

Veri kaynağı	Veri Formatı Örneği
Atauni ÖBS	{ "TOPLAM":898, "DESKTOP":253, "MOBILE":644, "TABLET":1 }
Atauni AVESIS	[{"TopicName":"Diğer", "PublicationCount":31081}, {"TopicName":"Temel Bilimler (SCI)", "PublicationCount":16198}, {"TopicName":"Tarım ve Çevre Bilimleri (AGE)", "PublicationCount":14171}, {"TopicName":"Klinik Tıp (MED)", "PublicationCount":13996}, {"TopicName":"Mühendislik, Bilişim ve Teknoloji (ENG)", "PublicationCount":10663}, {"TopicName":"Yaşam Bilimleri (LIFE)", "PublicationCount":10410}, {"TopicName":"Sosyal Bilimler (SOC)", "PublicationCount":10374}, {"TopicName":"Sanat ve Beşeri Bilimler (AHCI)", "PublicationCount":1928}]
Gazi Akademik Bilgi Sistemi	{ "resultDto": [["PROFESÖR",887], ["DOÇENT",363], ["DOKTOR ÖĞRETİM ÜYESİ",211], ["ÖĞRETİM GÖREVLİSİ",563], ["ARAŞTIRMA GÖREVLİSİ",1028]] }

Farklı harici veri kaynakları arasındaki veri formatı farklılıklarını göstermek amacıyla Tablo 1.3’de veri kaynağının adı ve bu veri kaynağından sağlanan ham verinin formatı verilmektedir. Bu tabloda görüleceği üzere harici veriler format olarak birbirine benzemektedir ve iç içe geçmiş parametrelerden oluşmaktadır. Bazı veri formatlarında örneğin, Atauni ÖBS (Öğrenci Bilgi Sistemi)’de başlangıç “{” sembolü ile başlayıp “}” ile bitmektedir. Ancak, bu sembol farklı veri kaynaklarında örneğin, Atauni AVESIS gibi farklı parametrelerin ayrılması için kullanılmaktadır. Parametrelerin değerleri bazılarında virgülden sonra ("PROFESÖR",887) bazılarında iki nokta sembolü ("TOPLAM":898) ile verilmektedir. Veri formatındaki farklılıklar bu aşamada ortaya

çıkmaktadır. Virgül bazı verilerde bir parametre değeri tanımlayıcısı olarak kullanılsa da diğer bazı verilerde parametre ayırıcı ("TopicName":"Diğer", "PublicationCount":31081) olarak kullanılmaktadır. "TopicName" ve "PublicationCount" farklı iki parametredir ve virgül ayırıcı ile ayrılmıştır.

Harici veri kaynaklarındaki veri formatları ilk izlenimde içi içe geçmiş dizi yapısına benzemektedir. Ancak, kullanılan semboller ve bu sembollerin görevleri farklılık arz etmektedir. Dolayısıyla, harici veri kaynaklarından sağlanan verilerde yapısal olarak farklılıklar bulunduğundan her biri ile ayrı ayrı ilgilenilmesi gerekmektedir. Sistemlerin bu tür verileri kullanabilmesi için paylaşımına açılmış olan her bir veri kaynağı için tanımlama yapılması gerekmektedir. Sisteme eklenmesi istenen yeni bir harici kaynağının veri formatı mevcut olanlara benzemesi durumunda kolayca dâhil edilebilmektedir. Ancak, veri formatıyla ilk defa sisteme ekleniyorsa öncelikli olarak bu veri formatının tanımlanması gerekmektedir.

1.6. MEVCUT YAZILIMLAR VE SERVİSLER

Gerçek zamanlı kurumsal metriklerin izlenmesini sağlayan birçok yazılım bulunmaktadır. Bu yazılımlar veri toplama ve veri analizi olmak üzere iki kategoride değerlendirilmektedir:

Veri toplama yazılımları çeşitli veri kaynaklarına bağlanıp ilgili verilere erişebilen araçları içermektedir. Bu yazılımlardan bazıları şunlardır: Apache Kafka, Apache Flume, Apache ActiveMQ ve RabbitMQ'dur. Literatürde bu yazılımlara gerçek zamanlı veri toplama araçları adı verilmektedir. Bu yazılımlar sayesinde akan veriler bir mesaj kuyruğu (queue) içerisinde Hadoop, Spark, Elasticsearch gibi farklı sistemlere transfer edilebilmektedir (Sakınmaz, 2016: 1).

Kurumsal metriklerin gerçek zamanlı olarak izlenmesinde karşılaşılan en büyük sorun verilerin analiz edilmesidir. Gerçek zamanlı olarak değişen verilerin bir kullanıcı tarafından incelenmesi mümkün değildir. Bu nedenle, bu tür verileri inceleme yeteneğine sahip araçlarla kullanılması gerekmektedir. Gerçek zamanlı veri analizi araçlarının bazıları şunlardır: Apache Spark, Streaming, Apache Storm, Apache Samza ve Apache Flink (Terzi, Sağıroğlu ve Koç, 2017: 24). Ancak, veri toplama araçları

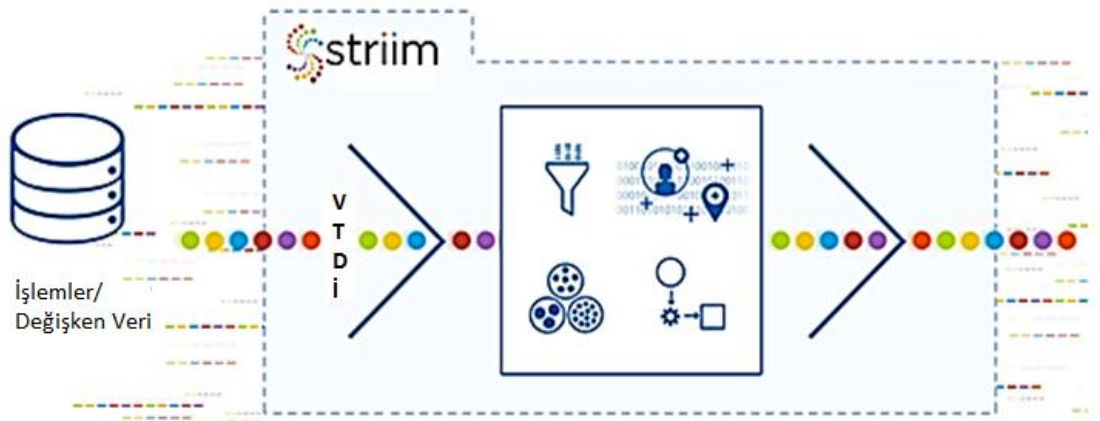
açısından da elde edilen verilerin deęişkenlik göstermesinden dolayı gerçek zamanlı veri analizini gerçekleřtirmede güçlükler bulunmaktadır (Samza, 2019: 1).

Hâlihazırda bulunan gerçek zamanlı veri akışı analiz araçları, arařtırmacılar için büyük kolaylıklar sağlamaktadır. Ancak, yeni bir bakış açısı veya bu araçlarda bulunmayan bir özellik ile gerçek zamanlı veriyi analiz etmek isteyen arařtırmacılar için durum çok farklıdır. Üniversite verilerinin gerçek zamanlı olarak izlenmesi ve veri analiz algoritmalarının gerçek zamanlı olarak uygulanması, mevcut paket yazılım olarak kullanılan gerçek zamanlı veri analiz araçlarında bulunmayan özelliklerdendir. Kurumsal metriklerin gerçek zamanlı olarak izlenebilmesi konusu bu duruma örnek olarak gösterilebilir. Bunun için çok farklı araçların bir arada kullanılması gerekmektedir. Bu ise araçların birbiri ile uyumsuzluğu, araçların ayrı ayrı kurulumu, sistemin süreklilięi ve bakımı açısından birçok olumsuz faktör mevcuttur. Mevcut yazılımların eksiklikleri ve olumsuz yönleri geliştirilen yeni bir prototipleme ile giderilebilir.

1.6.1. Striim ve Confluent

Veri toplama ve veri analiz araçlarının birlikte kullanılmasıyla gerçek zamanlı veriler incelenebilmektedir (Sakinmaz, 2016: 1). Ancak, bu iki aracın bir arada kullanılmasında bazı uyumsuzluklar ortaya çıkarmaktadır. Bu uyumsuzluklar genellikle veri araçları arasındaki veri aktarımıyla ilgilidir. Herhangi bir veri toplama aracından gelen veriler aynı şekilde herhangi bir veri analiz aracıyla analiz edilememektedir. Her bir veri analiz aracı belirli bir veri toplama aracının verilerini kullanmak üzere tasarlanmıştır. Örneęin, her veri toplama aracı istenilen VTYS'den veri toplayamamaktadır. Veri toplama ve analiz araçlarının birlikte çalışabilmesi birçok karmaşık ayarların düzgün bir şekilde yapılandırılması gerekmektedir. Bu sorunu çözmek için veri toplama ve veri analizini aynı anda yapabilme yeteneęine sahip tümleşik veri araçlarının kullanılması gerekmektedir. Striim ve Confluent bu tür tümleşik veri toplama ve veri analiz araçlardandır. Ancak, bu tür yazılımların VTYS'lerin belirli sürümlerini desteklemesi, sürekli bu yazılımların çalışır durumda kalamaması, kurulum ve ayarların çok karmaşık olması ve de lisanslı olması arařtırmacılar açısından karşılaşılabilecek en büyük zorluklardandır.

Striim ve Confluent, araştırma amaçlı deneme süreli ücretsiz lisans ile kullanılabilir. Tezin ön araştırma çalışması olarak bu yazılımlar incelenmiştir. Yapılan gözlemlere göre her iki yazılımın çalışma şekli birbirine benzemektedir. Yazılımların kurulumu herhangi bir özel donanım gerektirmeyen günümüz normal standartlardaki bilgisayarlar üzerinde sorunsuz bir şekilde gerçekleştirilebilmektedir. Kurulum tamamlandıktan sonra yazılımın çalıştırılabilmesi için veri toplama ve kaydetmeyle ilgili ayarların yapılandırılması gerekmektedir. Yazılım çalıştırdıktan sonra kullanıma hazır hale gelmesi yaklaşık 10-15 dakika kadar vakit almaktadır. Sonrasında veri kaynaklarının yazılıma tanıtılması (VTYS sürücülerinin yüklenmesi) gerekmektedir. Bu ise çok karmaşık bazı parametreleri ve ayarları içermektedir. Bu ayarların hem yazılım tarafında hem de VTYS tarafında tam olarak uygulanması gerekmektedir. Bu aşamada karşılaşılan en önemli sorun VTYS'lerin belirli sürümlerini desteklemesidir. Veri kaynağına bağlantı işlemi gerçekleştirildikten sonra veri kaynağından toplanan verilerin filtrelenmesi için ayarların yapılması gerekmektedir. Filtrelenen verilerin kullanımı için SQL komutları kullanılmaktadır. Eğer grafik olarak gösterilecekse ayrıca grafik ayarlarının eksiksiz bir şekilde tamamlanması gerekmektedir. Tüm gerekli ayarlar tam olarak yapılandırıldıktan sonra verilerin analizi gerçekleştirilebilmektedir. Diğer önemli bir husus, gerçek zamanlı verilerin izlenebilmesi için yazılımın sürekli açık kalması gerekmektedir. Yazılım arayüzü kapatıldığında ise sistem servis olarak arka planda çalışmamaktadır.



Şekil 1.12. Gerçek zamanlı veri toplama ve analiz aracı olarak Striim (Striim, 2016: 1)

Şekil 1.12'de gösterildiği üzere Striim VTYS'lerden gerçek zamanlı veri akışı sağlayabilmektedir. Toplanan veriler üzerinde filtreleme, kümeleme, dönüştürme, arama

işlemleri gerçekleştirebilmektedir. Bu sayede Striim kullanıcılarına analiz edilmiş veriyi sunabilmektedir (Striim, 2016: 1). Aynı şekilde bu hususlar Confluent için de geçerlidir (Confluent, 2018: 1).

1.6.2. Google Analitik

Google, kullanıcılarının web site trafiğini izleyebilmesi için Analytics servisini sunmaktadır. Bu servisin amacı kullanıcılarının web site trafiğindeki beklenmeyen değişiklikleri tespit edebilmelerini sağlamaktır. Bunun için Google Analytics servisi ilk zamanlarda Şekil 1.13’de gösterilen basit uyarı sistemini kullanmaktadır.

Uyarı adı:

Şuna Uygula: **Tüm Web Sitesi Verileri**

Süre:

Bu uyarı tetiklendiğinde bana e-posta gönder.

Uyarı Koşulları

Şunun için geçerli olsun:

Şu durumda beni uyar:

Koşul

Şekil 1.13. Google Analytics Uyarıları (Google, 2020: 1).

Şekil 1.13’de Google Analytics uyarı ekleme penceresi gösterilmektedir. Bu pencerede kullanıcılar kişiselleştirilmiş sabit uyarı oluşturabilmektedirler. Örneğin, kullanıcı bu uyarıyı web sitesindeki ziyaretçi sayısı bir önceki güne %10 düşmesi durumuna göre ayarlayabilmektedir.

Bu uyarı sisteminde görülen en büyük problemlerden biri uyarının web sitesinin potansiyel durumunu göz önünde bulundurmamasıdır. Bu nedenle, bazı durumlarda sonuçlar gerçeği yansıtmamaktadır. Örnek olarak, bir üniversitenin yeni kurulduğu ilk

zamanlarda aylık olarak web sitesindeki ziyaretçi sayısının az olmasına rağmen %10'luk bir değişim göstermesi yeterli olmayabilir. Ancak, bu %10'luk değişim köklü bir üniversite için yeterli hatta fazlasını ifade etmektedir. Ziyaretçi sayısındaki değişim yüzdelerle ifade etmek yerine gerçek değerlerle ifade etmek de sorunu çözmemektedir. Örneğin, haftalık ziyaretçi sayısında 50'den fazla artış doğru bir uyarı olmayacaktır. Bu uyarı sistemini kullananlar yukarıda verilen örneklerde görülen gerçek dışı uyarılar ile karşılaştığını belirtmektedirler.

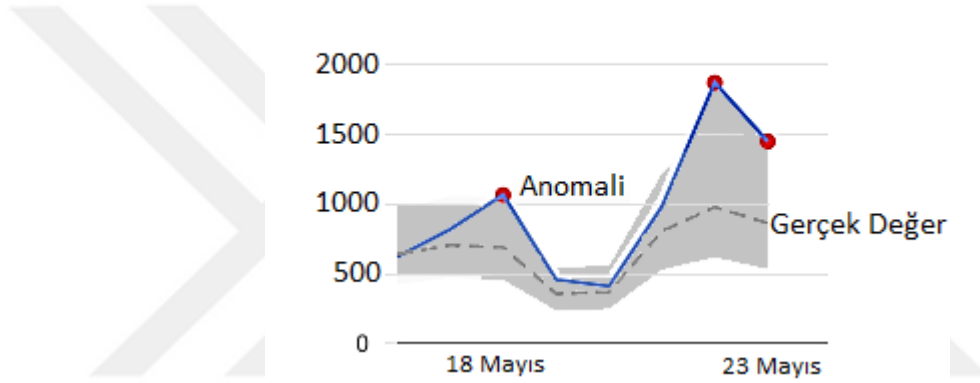
Google Analytics, kullanıcılarını web sitesi ziyaretçilerine bakarak uyarı sistemi geliştirmiştir. Ancak, bu uyarı sisteminin göz ardı ettiği husus web sitelerinin kullanıcı potansiyelinin zamanla değişkenlik göstermesidir. Sadece belirli bazı parametrelerin sabit değerlerle ayarlanması ile oluşturulan bir uyarı sistemi tam olarak gerçek hayatta doğru sonuçlar vermemektedir. Bu sorunun çözümü için Google, yeni bir uyarı sistemi olarak analitik zeka özelliğini geliştirmiştir. Bu özellik makine öğrenmesi algoritmalarına dayalıdır. Bu algoritmalar kullanılarak kullanıcılar için daha gerçekçi sonuçlar elde edilmesi amaçlanmaktadır. Google Analytics verilerindeki anormal gözlemleri tespit etmesi durumunda kullanıcılar otomatik olarak bilgilendirilmektedir.

Anomali Tespiti:
/signin.html web sayfasının
görüntülenme sayısında ani deęişim

23 Mayıs 2018

<https://shop.googlemerchandisestore.com/signin.html>

23 Mayıs 2018 tarihi itibari ile bu web sayfasının görüntülenme sayısı 531-1410 arasında olması beklenmiş olup gerçek görüntülenme sayısı olan 1450 bu beklentinin üzerindedir.



Şekil 1.14. Google Analytics akıllı uyarı sistemi (Google, 2020: 1)

Şekil 1.14’de Google Analytics akıllı uyarı sisteminden kullanıcıya gönderilen mesaj gösterilmektedir. Mesaja göre, signin.html web sayfasında görüntüleme sayısının beklentinin üzerinde bir değere sahip olması anomali olarak tespit edildiği görülmektedir. Bu beklenti değerleri aynı zamanda tahmin aralığı olarak da adlandırılmaktadır. Şekildeki zaman serisi grafiğine göre Google, nokta ile işaretlenen üç adet anormal gözlem tespit etmektedir. Google bu tespiti geçmiş verilere dayanarak yapmaktadır. Geçmiş veriler doğrultusunda beklenti aralığının alt ve üst değerleri oluşturulmaktadır. Normal olarak kategorize edilen verilerin bu tahmin değerleri içerisinde gözlemlenmesi beklenmektedir. Google geliştirdiği bu yeni anomali tespiti özelliğini şu şekilde tanımlamaktadır (Google, 2020: 1):

Öncelikli olarak, tahmin modelinin geliştirilmesi amacıyla geçmiş verilerin bir kısmı incelenir. Günlük anomalilerin tespitinde 90 günlük, haftalık anomalilerin tespitinde 32 haftalık geçmiş verilere bakılır.

Sonra, geçmiş veriler üzerinde Bayesian durum değer-zaman serileri modeli kullanılarak değer tahmini yapılır.

Son olarak, p-değerleri dikkate alınarak anomali olarak işaretlenen verilere istatistiksel anlamlılık testi uygulanır.

Bu ifadelerin daha iyi anlaşılabilmesi için şu şekilde özetlenebilir:

Tahmin modelinin geliştirilmesi amacıyla kullanılan geçmiş veriler: Şekil 1.14'de Google Analytics tahmin aralığının oluşturulması için geçmiş verileri kullanmaktadır.

Bayesian durum değer-zaman serileri modeli: Bu ifade Google'ın anomalilerin tespitinde kullandığı modeli belirtmektedir. Birçok farklı model bulunmaktadır.

P-değerleri dikkate alınarak uygulanan istatistiksel anlamlılık testi: Tespit edilen değerlerin istatistiksel olarak geçerliliğini kontrol edilmesidir. P-değerleri istatistiksel anlamlılığı ölçmede kullanılan bir yöntemdir.

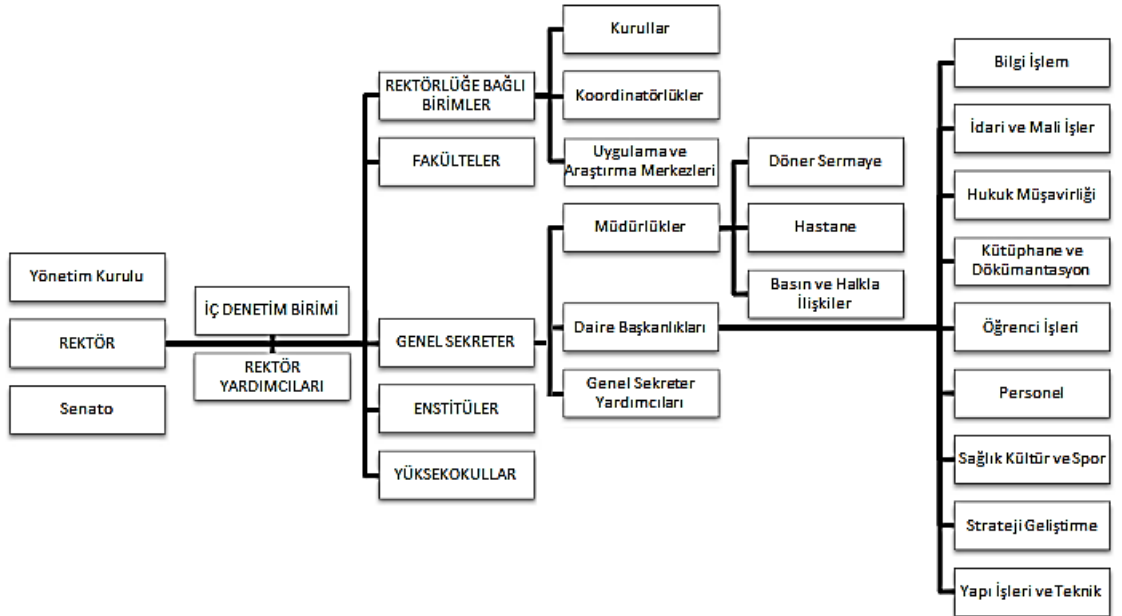
İKİNCİ BÖLÜM

ÜNİVERSİTE YÖNETİM FAALİYETLERİNDE BİLİŞİM SİSTEMLERİNİN ROLÜ

Teorik bilgilerin gerçek hayatta uygulanabilirliği, hiç kuşkusuz bu bilgilerin kullanım alanının iyi bir şekilde araştırılmasına ve anlaşılmasına bağlıdır. Çalışmada uygulama alanı olarak üniversite ele alınmıştır. İşletmeler için geçerli olan çoğu kavram ve tanım üniversiteler için de geçerlidir. Üniversiteler için belirlenen kuralların uygulanmasını ve eğitim sürecinin düzenlenmesini sağlayan yöneticiler bulunmaktadır. Üniversite yöneticileri işletmelerde olduğu gibi çeşitli kararlar vermek durumundadırlar. Bilişim sistemleri de yöneticilerin kararları açısından üniversitenin hem iç çevresinde oluşan hem de dış çevreden gelen bilginin güvenliğini ve tamlığını sağlamasında önemli bir konuma sahiptir.

2.1. ÜNİVERSİTELERDE TEMEL KURUMSAL YAPI VE YÖNETİM

Üniversiteler yüksek düzeyde eğitim ve öğretim veren ve bilimsel araştırma ve yayım yapan, bilimsel özerkliğe sahip bir yükseköğretim kurumudur. Fakülte, bölüm, anabilim dalı, enstitü, yüksekokul, araştırma ve uygulama merkezlerinden oluşmaktadır (Üniversitelerde Akademik Teşkilat Yönetmeliği, 1982: 5348).



Şekil 2.1. Üniversitelerin temel kurumsal yapısı.

Şekil 2.1’de Türkiye’deki üniversitelerin temel kurumsal şeması gösterilmektedir. Bu kurumsal şema Atatürk Üniversitesi, Afyon Kocatepe Üniversitesi, Galatasaray Üniversitesi, İstanbul Kültür Üniversitesi, Ordu Üniversitesi, Bilkent Üniversitesi, Gazi Üniversitesi, Ankara Üniversitesi, Karabük Üniversitesi kurumsal şemalarının ortak yönleri dikkate alınarak hazırlanmıştır. Şekilde de görüleceği üzere üniversitelerde en üst düzeyde rektör, senato, yönetim kurulu, rektör yardımcıları ve iç denetim birimi bulunmaktadır. Ardından fakülteler, yüksekokullar, enstitüler, rektörlüğe bağlı birimler olan kurullar, koordinatörlükler ve uygulama ve araştırma merkezleri ve genel sekreterliğe bağlı müdürlükler, daire başkanlıkları ve genel sekreter yardımcıları yer almaktadır. Döner sermaye, hastane ve basın ve halkla ilişkiler müdürlükleri; bilgi işlem, idari ve mali işler, hukuk müşavirliği, kütüphane ve dokümantasyon, öğrenci işleri, personel, sağlık kültür ve spor, strateji geliştirme ve yapı işleri ve teknik daire başkanlıkları üniversite temel kurumsal şemasında yer alan diğer birimlerdir.

Üniversiteler, içerisinde birçok unsurun uyum içinde işlediği birimlerden oluşmaktadır. Bu uyumun sağlanmasında ve sürdürülmesinde yöneticilere önemli görevler düşmektedir. Yöneticiler istenilen hedeflere ulaştırmak için belirlenen kuralların uygulanması ve eğitim sürecinin düzenlenmesine yönelik birçok faaliyeti yerine getirirler. Kararlar verme, planlar yapma, çalışma ortamının düzenlenmesi, yapılanların kontrol edilmesi bu faaliyetlerden sadece bir kısmıdır.

Üniversitelerin işleyişi, yapısı, eğitim ve öğretim faaliyetleri kanun ve yönetmeliklere göre düzenlenmektedir. Türkiye’deki üniversiteler 6.11.1981 tarihli ve 2547 sayılı Resmi gazetede yayımlanan Yükseköğretim kanununa göre faaliyetlerini yerine getirmektedir Yükseköğretim Kurumu, kanunla kendisine verilen görev çerçevesinde tüm yükseköğretim kurumlarının öğretimini düzenleyen ve faaliyetlerine yön veren bir kuruluştur (Yükseköğretim Kanunu, 1981: 5347).

Üniversitelerde yönetim, başta rektör olmak üzere sırasıyla; senato, üniversite yönetim kurulu, dekan, fakülte kurulu ve fakülte yönetim kurulu olmak üzere çeşitli birimlere ayrılmıştır. Bu yönetim birimlerine göre bilgi ihtiyacı farklılık göstermektedir. Üniversitelerde yönetim, yönetim organlarının sorumlu oldukları birimleri ve görevleri kapsamaktadır.

2.1.1. Yönetimin Fonksiyonları

Üniversiteler birçok faaliyetin bir arada yürütüldüğü kurumlardır. Bunların belirli bir amaca ulaşması için harekete geçirilmesinde yöneticilere önemli görevler düşmektedir. Yönetime doğru bir şekilde destek olunması çalışanlar, yöneticiler hatta üniversitenin dış çevresi ile uyum içinde faaliyetlerin sürdürmesine yardımcı olacaktır. Bu nedenle, yönetimin doğru bir şekilde desteklenebilmesi için yönetim ile ilgili temel kavram ve fonksiyonların iyi tanınması gerekmektedir. Yönetimin fonksiyonları beş kategoride sınıflandırılmıştır. Bunlar (Ertekin, 2006: 79):

- Planlama
- Organizasyon (örgütlenme)
- Yürütme
- Eşgüdümleme (koordinasyon)
- Denetim

Yöneticiler bu fonksiyonların icra edilmesi için çeşitli durumlarda en uygun kararları almak durumundadırlar. (Robledo, Sepulveda ve Archer, 2013: 2074). Yöneticilerin verdikleri bu kararlar yönetim düzeyine göre üst, yönetim ve operasyonel seviyede kararlar olmak üzere üç kategoride sınıflandırılmaktadır. Kararlar için gerekli bilgi ihtiyacı yönetim düzeyine göre farklılık göstermektedir (Acar, 2003: 100). Karar verme eyleminden sonra bu kararın gerçekleştirilebilmesi için bir planlamaya ihtiyaç vardır. Belirlenen hedeflere ulaşma açısından planlama, belirli bir amaca yönelik olması önemlidir.

Organize etme, plana uygun olarak personel, yer, görev tahsisi, zamanlama, vb. gibi birçok unsurun düzeninin sağlanmasıdır. Böylece kimin, ne zaman, nerede ve nasıl bir faaliyet göstereceği belirlenmiş olur. İş bölümü, emir komutu zinciri, bölümlenme ve formalleşme yapısının açıkça belirlenmesi organize etme faaliyetinin düzgün bir şekilde gerçekleştirilmesini sağlar (Yavuz, 2014: 53).

Gerekli organizasyon yapıldıktan sonra planın amaca yönelik olarak gerçekleştirilmesi gerekmektedir. Bu gerçekleştirme işlemi için yönetimin yürütme fonksiyonu uygulanmaktadır. Yürütme fonksiyonunda liderlik, iletişim ve motivasyon becerileri ön plana çıkmaktadır. Otokratik, babacan, katılımcı ve serbest olmak üzere dört çeşit liderlik çeşidi bulunmaktadır. İletişim becerilerinde yöneticilerin

ilgili mesajın, uygun bir dil ile gerekli kanallardan hedefine ulaştırması gerekmektedir. Ayrıca iletişimde mesajın değişime uğrama ihtimali de göz ardı edilmemelidir (Ertekin, 2006: 101; Yavuz, 2014: 54).

Motivasyon becerilerinde süreç ve kapsam olmak üzere iki farklı teori uygulanmaktadır. Kapsam teorileri, çalışma ortamına odaklanırken süreç teorileri motive edici unsurların aralarındaki etkileşim ile ilgilenmektedir (Ertekin, 2006: 106; Küçüközkan, 2015: 100).

İhtiyaçlar hiyerarşisinde çift faktör, erg yaklaşımı ve başarıma ihtiyacı olmak üzere dört adet kapsam teorisi vardır. İhtiyaçlar hiyerarşisine göre öncelik olarak çalışanların fizyolojik ihtiyaçların giderilmesidir. Daha sonra sırasıyla güvenlik, aidiyet, saygınlık ve kendini gerçekleştirme ihtiyaçları izlemektedir. Çift faktör yaklaşımında hijyen ve motivasyon önemlidir. Erg yaklaşımında gelişme, ilişkisel ve var olmak üzere üç önemli konu bulunmaktadır. Başarım ihtiyacı teorisi başarıma, ilişki kurma ve güç kazanma ihtiyaçları üzerine kurulmuştur (Ercan, 2003: 103; Ertekin, 2006: 107).

Pekiştirme, beklenti, eşitlik ve amaç olmak üzere dört adet süreç teorisi bulunmaktadır. Pekiştirme teorisi çalışanları harekete geçiren unsurlara odaklanmaktadır. Beklenti teorisine göre çalışanları motive eden başarıya ulaşmak için duyulan inançtır. Eşitlik teorisinde çalışanlar tarafından sağlanan girdiler onlar için birer çıktı olduğu düşünülür. Amaç teorisinde çalışanların belirli ve zorlayıcı amaçları olması onları motive ettiği kabul edilir (Küçüközkan, 2015: 106; Taşdemir, 2018: 38).

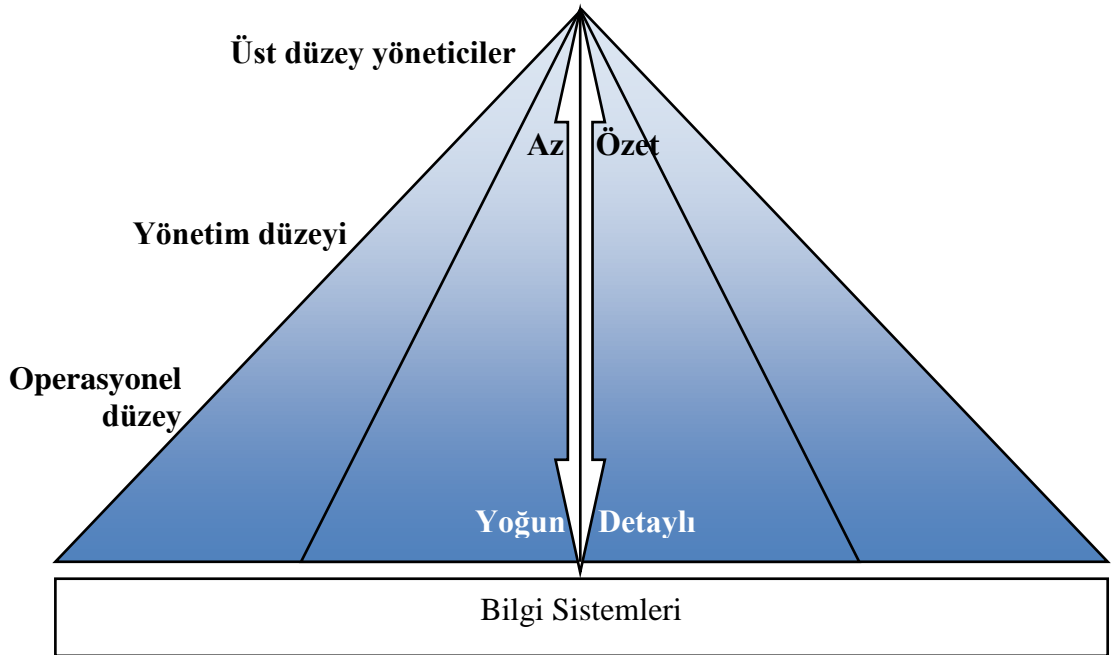
Üniversiteler, birçok iş sürecini amaçları doğrultusunda yürütmeleri gerekmektedir. Bu iş süreçleri ile çalışanlar arasındaki uyum ve iş birliğinin sağlanması için iyi bir koordinasyon gerekmektedir. Bu sayede üniversite kaynakları daha verimli bir şekilde kullanılır. Ayrıca koordinasyon yöneticilerin üniversite iş süreçlerinde disiplinin sağlanmasına yardımcı olur (Koçoğlu, 2010: 26; Yavuz, 2014: 54).

Yöneticiler, yürütülen faaliyetlerin amacına ulaşp ulaşmadığını anlamak için kontrol fonksiyonunu yerine getirirler. Yönetimsel ve finansal faaliyetlerin kontrolü olmak üzere iki temel kontrol mekanizması bulunmaktadır. Yönetimsel faaliyetlerin kontrolünde planlama, organize etme ve yürütme süreçlerinin başarıya ulaşp ulaşmadığı ya da eksiklikler kontrol edilir. Finansal kaynakların ve faaliyetlerin verimli kullanıp kullanılmadığının kontrolü üniversite iş süreçlerinin devamlılığı açısından önemlidir.

Aksi takdirde üniversite faaliyetlerinin kesintiye uğraması kaçınılmaz olur (Ertekin, 2006: 112).

2.1.2. Yönetim Düzeyleri

Literatürde yönetim hiyerarşisini temsil etmek amacıyla genellikle piramit yapısı kullanılmaktadır. Bu piramit yapısı, Max Weber'in organizasyonlar ile ilgili sosyal ve ekonomik görüşlerini takip eden Amerikalı sosyolog Talcott Parsons'ın fikirlerine dayanmaktadır (Weber, 1964). Bu temsili piramit yapısının en dikkat çeken özellikleri operasyonel düzeyde çok sayıda yönetici olması nedeniyle tabanın geniş olması ve yukarı çıkıldıkça sayılarının azalmasıdır. Piramit yapısının bu özelliği yönetim düzeylerinin bazı özelliklerinin daha iyi kavranmasını amaçlamaktadır (Yavuz, 2014: 18). Yönetim düzeylerinin özelliklerinin gösterimi için piramit yapısı referans alınarak düzenlenmiş çeşitli şekiller bulunmaktadır. Literatürde sıklıkla karıştırılan bir konu da, yönetim düzeyleri ile yönetici türleri olan üst, orta ve operasyonel yönetici kavramlarının karıştırılmasıdır. Tüm kurumlarda olduğu gibi üniversitelerde de Şekil 2.2'de gösterildiği üzere üst, yönetim ve operasyonel olmak üzere üç çeşit yönetim düzeyi bulunmaktadır. Aşağıda gösterilen Şekil 2.2'de bilgi sistemleri aracılığıyla yönetim düzeylerine sağlanan bilginin bazı özellikleri gösterilmektedir.



Şekil 2.2. Yönetim düzeyleri arasında hiyerarşik bilgi akışı (Yavuz, 2014: 18)

Şekil 2.2'de bilgi sistemlerinden hiyerarşik olarak yönetim düzeylerine sağlanan bilgi akışı gösterilmektedir. Bilgi akışı yöneticilerin ilgilendiği konuların sayısını ifade etmektedir. Operasyonel yöneticiler, operasyonel seviyede üretilen tüm bilgilerle ilgilidirler. Dolayısıyla, hiyerarşiye dağılan tüm bilgiler Şekil 2.2'de görüldüğü üzere, öncelikli olarak operasyonel düzeydeki yöneticilerin alanında yer almaktadır. Üst düzeye dağılan tüm bilgiler yönetim düzeyindeki yöneticilerle de ilgilidir. Üst düzey yöneticiler, hiyerarşide başka bir üst düzey bulunmadığından sadece kendilerini ilgilendiren bilgilere ulaşırlar. Böylelikle piramit yapıya göre bilgi yoğunluğu, operasyonel yönetim düzeyinden üst yönetim düzeyine doğru azalmaktadır.

Şekil 2.2'de gösterilen diğer bir özellik ise hiyerarşiye dağılan bilginin detaylı ya da özet bilgi olmasıyla ilgilidir. İçerisinde mümkün olduğunca birçok kümeyi barındıran bilgiler en büyük alana sahip olan operasyonel düzeydir. Kısaca yönetim düzeyi arttıkça yöneticiler daha az ama kendileri için önemli ve özet bilgilere ihtiyaç duymaktadırlar. Elde ettikleri bu bilgiler sayesinde yöneticiler, karar alma fonksiyonunu yerine getirirler.

Üst düzeydeki yöneticiler stratejik kararlar alırlar. Bu alınan kararlar çoğunlukla yapısal olmayan problemlerin çözümünde kullanılır. Bu tür problemlerin çözümü için belirli bir prosedür yoktur. Yeni, önemli ve sıradan olmayan problemlerdir. Uzun dönem hedeflerin belirlenmesi gibi konular üst düzey stratejik kararların alınmasını gerektirir (Acar, 2003: 102).

Yönetim düzeyindeki yöneticiler genellikle planlama, organize etme ve analiz gibi faaliyetlere yönelik taktiksel kararlar alırlar (Anameriç, 2005: 30). Taktiksel kararlar genellikle yarı-yapısal problemlerin çözümünde kullanılır. Sorunun çözümünün bir kısmı için belirli bir prosedür bulunmaktadır, ancak kalan kısmı için daha önceden herhangi bir prosedür belirlenmemiştir (Yavuz, 2014: 18).

Operasyonel düzey yöneticileri, çoğunlukla operasyonel kararlar alırlar. Operasyonel kararlar yapısal problemlerin çözümünde kullanılır. Yapısal problemler, tekrar eden ve rutin sorunlar olduğundan bu tür problemlerin daha önceden belirlenmiş bir çözümü vardır. Yönetsel beceriler olarak çalışanları anlama, iyi ilişkiler kurma, liderlik, motivasyon ön plandadır. Operasyonel yöneticilerin bilgi, deneyim ve beceri

sahibi olmaları operasyonel faaliyetlerin daha verimli yürütülmesini sağlar (Anameriç, 2005: 30).

2.1.3. Yöneticinin Rollerini

Üniversitenin çeşitli düzeylerinde bulunan yöneticiler yönetsel fonksiyonlarını yerine getirebilmeleri için bireyler arası ilişkiler, bilgi sağlama ve karar alma gibi bazı temel rollerini üstlenmeleri gerekmektedir (Koçoğlu, 2010: 12).

Bireyler arası ilişkiler, yöneticilerin iletişim halinde olduğu tüm kişiler arasındaki sosyal davranışları temsil etmektedir. Bu konuda yöneticiye üniversite amaçları doğrultusunda kişilerarası sosyal davranışları düzenleyici ve yönlendirici görevler düşmektedir. Yöneticilerden dış çevrede üniversiteyi iyi bir şekilde temsil etmesi, aynı düzeydeki yöneticiler ile işbirliği sağlama, üst yöneticilerin çalışmalarını desteklemesi ve çalışanlarına önderlik ve motive etmesi beklenir (Koçoğlu, 2010: 12).

Bilgi sağlama, yöneticilerin topladıkları bilgilerin gerekli olanları iletişim halinde olduğu tüm kişilere açık ve eksiksiz bir şekilde zamanında bildirmesini ifade etmektedir. Üniversite çalışanlarını ilgilendiren bir bilgi yönetici tarafından uygun bir iletişim kanalı ile ulaştırılmalıdır. Bu sayede üniversitede herhangi bir faaliyetin aksamasının önüne geçilebilir ya da bu faaliyetin performansı daha da artırılabilir. Örneğin, özel bir üniversitede dış çevreden kaynaklı olumsuz bir durumda; üniversitenin mali açıdan zor durumda bulunmasının nedeninin çalışanlar olmadığı, dış etkenlerden kaynaklandığının açıklanması motivasyonunun korunması açısından iyi bir bilgilendirme (Anameriç, 2005: 26; Koçoğlu, 2010: 13).

Karar alma rolü, hiç kuşkusuz yöneticilerin en önemli rollerinden biridir. Üniversite faaliyet ve süreçlerinin devamlılığının sağlanması açısından önemlidir. Yöneticiler karşılaştıkları çeşitli problem karşısında çözüm bulup üniversitenin amaçları doğrultusunda faaliyetlerini devam ettirmeleri gerekmektedir. Karşılaşılan bir problemin ya da olayın üniversite faaliyet süreçlerinde kullanılabilmesi için alternatif birçok çözüm yolu bulunabilir. Bu çözüm yollarından en uygun olanına karar verilmesi görevi yöneticilere düşmektedir (Yavuz, 2014: 56).

2.1.4. Karar Alma Süreci

Yöneticilerin aldığı kararlar üniversitenin amaçlarına ulaşması açısından önemlidir. Yöneticinin sorumluluğu altında bulunan çalışanlar, alınan bu kararlar doğrultusunda iş süreçlerini yerine getirirler. Yöneticinin karar alma eyleminin planlı bir süreç içerisinde gerçekleşmesi kuşkusuz alınan kararın daha olumlu sonuçlar vermesini sağlayacaktır (Şengür ve Tekin, 2013: 7). Bu kapsamda karar alma sürecinin temel aşamaları problemin tanımlanması, alternatif çözümler geliştirme, alternatifleri değerlendirme, en uygun alternatifin seçilmesi ve kararın uygulanmasıdır (Yavuz, 2014: 56).

Problemin tanımlanması, karşılaşılan sorunlara doğru bir teşhis konulmasını sağlar. Konulan teşhisin doğruluğu oranında uygun çözüm yolları geliştirilebilir. Problemin tam olarak tanımlanamadığı ya da yanlış olarak tanımlanması geliştirilen çözümlerin de eksik ya da yanlış olmasına neden olur (Koçoğlu, 2010: 43; Yavuz, 2014: 56).

Alternatif çözümler *geliştirme* en uygun çözümün ortaya çıkarılması adına mevcut çözümlerin ortaya konmasıdır. Böylelikle problemin çözüme ulaşmasında en çok katkı sağlayan alternatifin ortaya çıkarılması amaçlanmaktadır (Koçoğlu, 2010: 49; Yavuz, 2014: 56).

En uygun alternatifin seçilmesi, problemin çözümü için önemlidir. Mevcut alternatifler doğru bir şekilde değerlendirme sürecinden geçirilmesi gerekmektedir. Alternatif çözümlerin her biri problemin tamamı için bir çözüm sunmasa da bir kısmına çözüm getirebilir. Bu nedenle, en uygun alternatif mevcut alternatiflerin bir birleşimi olabilir (Koçoğlu, 2010: 51; Yavuz, 2014: 57).

Kararın uygulanması çözüm için seçilen en uygun alternatifin hayata geçirilmesidir. Her bir alternatifin kendine özgü uygulamaya yönelik bazı kısıtları bulunabilir. Dolayısıyla, seçilen en uygun alternatif zaman, yer, kişi, kapsam, vb. gibi parametreler göz önünde bulundurularak uygulanmalıdır. Kararın uygulanma aşamaları takip edilerek problemin çözüme ne ölçüde yardımcı olduğu gözlemlenebilir. Gerektiğinde oluşan yeni ortama göre problemi yeniden tanımlama aşamasına dönülebilir (Koçoğlu, 2010: 53; Yavuz, 2014: 57).

2.2. ÜNİVERSİTELERDE BİLİŞİM SİSTEMLERİ

Üniversiteler kurumsal faaliyetlerin ve süreçlerin daha iyi düzenlenmesi ve yürütülmesi için bilişim sistemlerini kullanmaktadırlar. Bilişim sistemleri, üniversitelerde bilgi yönetiminin uygulanmasını sağlamakta ayrıca karar verme ve kontrol faaliyetlerini de desteklemektedir. Üniversitelerde birçok farklı birimde çeşitli görevlere sahip çalışanlar bulunmaktadır. Her bir çalışanın bilgi ihtiyacının karşılanması ancak iyi bir bilgi yönetimi sayesinde mümkündür. Bilgi yönetimi, bir üniversitede bilginin toplanmasından paylaşımına kadar olan bilgi süreciyle ilgili kurallar bütünüdür. Bilgi yönetimin etkin bir şekilde uygulanmasını sağlayan bilişim sistemleri sayesinde üniversite yöneticileri ve çalışanlar gerekli bilginin gerektiği kadarına hızlı ve kolay bir şekilde erişim sağlayabilmektedirler. (Yavuz, 2014: 21). Bu bilgi süreci içerisinde bilginin dönüştürülmesi, işlenmesi, sınıflandırılması, standartlaştırılması ve dağıtımı faaliyetleri bulunmaktadır. Bilişim sistemleri teknik olarak veri girişi, işleme (sınıflama, düzenleme ve hesaplama) bilgi çıktısı ve geri besleme faaliyetlerini gerçekleştirmektedir (Tecim ve Gökşen, 2009: 2241).

2.2.1. Bilişim Sistemlerinin Gelişimi

Bilgisayar ve bilişim sistemi kullanımının yaygınlaşması ile birlikte kullanılan verilerin elektronik ortamda saklanma ihtiyacı doğmuştur. Bu ihtiyaç ilk zamanlarda klasik dosyalama sistemleri ile karşılanmıştır. Klasik dosyalama sistemlerinde genellikle verilerin saklanması için işletim sisteminin veya ofis programlarının sunduğu imkânlar kullanılmıştır (Silberschatz, Korth ve Sudarshan, 2011: 3).

Klasik dosyalama sistemlerinde veri erişiminde ve saklanmasında temel olarak sıralı dosyalar, rastgele-direk erişimli dosyalar ve dizinler olmak üzere üç dosyalama düzeni kullanılmıştır (Silberschatz, Korth ve Sudarshan, 2011: 457; Özkan Y., 2013: 2).

Sıralı dosyalar, depolama biriminin fiziki saklama düzenine göre veriler saklanmakta ve verilere erişilmektedir. Sıralı dosyaların kayıtları genellikle belirli bir anahtar alana göre sıralanmakta ve işlem görmektedir. Herhangi bir bilgiye ulaşmak için veri bulunana kadar dosya okunur. Günlük harekete dayalı işlemlerin zamana göre depolama birimine kaydedilmesi işlemi sıralı dosyalamaya örnek olarak gösterilebilir (Silberschatz, Korth ve Sudarshan, 2011: 458).

İlk çalışma anında, veri saklama ve erişim yoğunluğuna sahip uygulamalar için sıralı dosyaların kullanımı daha uygundur. Aylık faturaların veya haftalık olarak kasanın raporunun çıkarılması gibi toplu veri işlemleri yapan uygulamalar için idealdir. Bu tür uygulamalarda herhangi bir alt işleme girmeden tüm veriler sırasıyla kaydedilmekte veya sırayla raporlanmaktadır. Bu da toplu işlemlerde sıralı dosyaların verimliliğini artırmaktadır (Silberschatz, Korth ve Sudarshan, 2011: 459).

Rastgele veya direkt erişimli dosyalar; rastgele veya sıralı erişime sahip dosyalardaki kayıtlar herhangi bir sırada okunabilir ve yazılabilir. Genellikle direkt olarak bir kayda erişmesi gereken uygulamalar tarafından kullanılmaktadır. Uygulama, istenilen kayda direkt erişebilmektedir. Bu sayede erişim hızı çok iyidir. Ancak, toplamda tüm kayıtlara erişim hızı göz önünde bulundurulduğunda erişim hızı sıralı dosyalara göre yavaştır (Ramakrishnan ve Gehrke, 2003: 275).

Dizinler rasgele erişimde kullanılan yöntemlerden biridir. Dizin kaydı, dizin adı verilen bir dosyada saklanır. Bu dosyada yer alan her bir kaydın anahtar alanı ile fiziki depolama birimindeki adresi eşleştirilmiştir (Ramakrishnan ve Gehrke, 2003: 276).

Dizin dosyaları sıralı dosyalarda da kullanılabilir. Sıralı dosyalarda mantıksal işlemler, fiziki depolama biriminden bağımsız olan anahtar alan sırasına göre gerçekleştirilmektedir. Bu nedenle, dizin dosyaları da sıralı dosyalarda anahtar alan olarak kullanılabilir (Silberschatz, Korth ve Sudarshan, 2011: 475).

Dizin dosyasının kontrolü işletim sistemi veya bir yazılım tarafından sağlanmaktadır. Veri dosyası açıldığında dizinin tamamı okunarak geçici hafızaya alınmakta ve dosya kapatılana kadar geçici hafızada tutulmaktadır. Veri dosyasında yer alan bir kaydın okunması veya yazılmasıyla ilgili bir komut geldiğinde, işleme alınacak kaydın anahtar bilgisine göre hafızada yer alan fiziksel adres bilgisine ulaşılır. Daha sonra ilgili kayıt depolama birimine saklanır veya buradan okunur (Ramakrishnan ve Gehrke, 2003: 317).

Dizin dosyalarında yer alan dizin kayıtlarının oluşturulmasında göreceli adresleme ve hesaba dayalı dosyalar olmak üzere iki yöntem kullanılmaktadır (Silberschatz, Korth ve Sudarshan, 2011: 473).

Göreceli adresleme dizin kayıtlarının oluşturulmasında alternatif bir yöntemdir. Veri dosyasının başlangıcı referans alınarak tüm kayıtlara bir numara atanmaktadır.

Birinci kayda 0, ikinci kayda 1, üçüncü kayda 2, vb. numara ataması örnek olarak gösterilebilir. Böylece göreceli olan bu kayıt numarası bir anahtar alan olarak kullanılabilir (Silberschatz, Korth ve Sudarshan, 2011: 476).

Hesaba dayalı dosyalar göreceli kayıt numarası olarak mantıksal bir anahtar alan yerine depolama biriminin fiziki adresleri kullanılır. Göreceli fiziki adresler belirli bir hesaplama ya da algoritma kullanılarak mantıksal bir anahtara dönüştürülmektedir. Bazen hesaplama için tek bir algoritma kullanılması çakışma yaratabilmektedir. Bu çakışmaya yol açmamak için ikincil bir algoritma daha kullanılabilir (Silberschatz, Korth ve Sudarshan, 2011: 514).

Veri saklama için kullanılan klasik dosyalama sistemleri kullanım kolaylığı ve kolay erişim gibi özellikleri sayesinde çok hızlı bir şekilde yaygınlaşmıştır. Kullanılan işletim sistemlerine benzer bir ara yüze sahip olması, kullanıcıların klasik dosyalama sistemlerine kolaylıkla adapte olmasını sağlamıştır. Bilgilerin başka bilgisayarlara basit yöntemler kullanılarak rahatlıkla kopyalanabilmesi, çoğaltılması, paylaşılması klasik dosyalama sistemlerinde kolay erişimi sağlamıştır.

Maalesef, ilk zamanlarda veri dosyalama sisteminin sağladığı bu kolaylıklar zamanla kullanıcılar açısından olumsuzluklar oluşturmaya başlamıştır. Mezun öğrenci kayıtlarını saklama durumunda olan üniversite kurumu için ciddi bir şekilde üzerinde düşünülmesi gereken bir konu haline gelmiştir. Verilerin bu kadar kolay erişilebilmesi, taşınabilmesi, değiştirilebilmesi kişilerin ve üniversitelerin veri gizliliği ve güvenliği konularını ön plana çıkarmıştır. Bu nedenle, önemli verileri içeren bazı veri dosyaları daha dikkatli bir şekilde saklanmaya başlanmıştır. Bilişim sistemleri ve bilgisayarlar kullanılmaya devam ettikçe saklanması gereken devasa veriler birikmeye başlamıştır. Oluşan bu devasa verilere her ne kadar dikkat edilirse edilsin özellikle veri güvenliği ve gizliliği konusu klasik dosyalama sistemleri ile aşılamayacak bir hale gelmiştir.

Geliştirilen her teknoloji zaman içerisinde daha da geliştirilerek yenilenmektedir. Bu bilgisayarlar ve bilişim sistemlerinde olduğu gibi veri saklama sistemlerinde de geçerlidir. Bilgisayar altyapısında ve yazılımdaki gelişmeler yeni veri saklama teknolojilerini ortaya çıkarmıştır. Klasik veri dosyalama sistemlerinde ortaya çıkan olumsuzlukları gidermek üzere veri tabanı teknolojileri geliştirilmiştir. Bu sayede klasik veri dosyalama sistemlerindeki olumsuzluklar giderilmiştir. Bu olumsuzluklar veri

soyutlama, veri erişimi, veri bütünlüğü ve güvenliği, veri yönetimi, eş zamanlı erişim ve gerçek zamanlı kurtarma ve uyumluluk olmak üzere altı maddede sıralanmaktadır (Ramakrishnan ve Gehrke, 2003: 9).

Veri soyutlama: Uygulamalar, veri ve veri depolama birimiyle ilgili ayrıntılı bilgiye ihtiyaç duyarlar. Bu konuda önemli olan uygulamanın sadece ilgili verilerin ayrıntılı bilgisine sahip olmasıdır.

Veri erişimi: Veri, harici depolama kaynaklarında saklansa bile etkin bir şekilde veri kaydı ve veri getirme işlemleri gerçekleştirilebilmelidir.

Veri bütünlüğü ve güvenliği: Verinin saklanması, değiştirilmesi ve silinmesi işlemlerinin belirli bir bütünlük dahilinde gerçekleştirilebilmesi gerekmektedir. Örneğin, çalışanların yaşı girilirken negatif veya küçük değerlerin girilmesi veri bütünlüğünü bozmaktadır. Veriyle ilgili işlemlerin hangi kullanıcı veya kullanıcı grupları tarafından hangi veriler üzerinde işlem yapılabileceği kontrol altında tutularak veri güvenliği sağlanmalıdır.

Veri yönetimi: Birçok kullanıcı ve uygulamanın aynı veri üzerinde gerçekleştirdiği işlemlerin koordinasyonunun sağlanması veri yönetimi ile mümkündür. Verinin belirli bir merkezde toplanıyor olması veri yönetimini kolaylaştıran en önemli etkenlerden birisidir. Özellikle de veri tekrarının önlenmesi ve veriyle ilgili işlemlerde performansın sağlanması iyi bir veri yönetimi sayesinde üstesinden gelinebilmektedir.

Eşzamanlı erişim ve gerçek zamanlı kurtarma: Birçok kullanıcı ve uygulama aynı veri üzerinde işlem yapma durumunda kalabilmektedir. Verilen serviste hiçbir aksaklık meydana gelmemesi için eşzamanlı erişim kontrolü gerekmektedir. Olası bir erişim çakışması durumu tespit edilerek gerçek zamanlı kurtarma yeteneği sayesinde veri bir önceki durumuna dönüştürülebilmelidir. Böylece bir çakışma durumunda uygulama veya kullanıcı işleme tekrar başlatılabilir. Bu sayede de hatalı bir işlemin önüne geçilmiş olur.

Uyumluluk: Uygulamaların ve işletim sistemlerinin sayısı her geçen gün artmakta ve aynı zamanda geliştirilmektedirler. Veriyle ilgili işlemlerden beklenen, aynı ölçüde belki de bir adım önde bu gelişen uygulamalar ve işletim sistemleri ile uyum içerisinde olmasıdır. Veri saklama teknolojileri, uygulamalardaki bu yeniliklere katkı sağlayacak ve yeni gelişmelere öncülük edecek özelliklere sahip olmalıdır.

Teknolojideki gelişmeler veri dosyalama sistemlerinde yeni beklentileri ortaya çıkarmıştır. Yukarıda bahsi geçen altı temel başlık altında sıralanan bu beklentiler veri tabanı teknolojileri ile karşılanabilmektedir. Veri tabanı teknolojisi sayesinde bu temel beklentilerin yanı sıra ayrıntı düzeyinde bulunan birçok farklı türde karmaşık yapıya sahip veri işlemlerinde de beklentileri kolaylıkla karşılayabilmektedir.

2.2.2. Bilişim Sistemi Çeşitleri

Üniversiteler ihtiyaçları doğrultusunda kurum ve birim bazında birçok farklı bilişim sistemleri kullanılmaktadır. Üniversitelerde en çok kullanılan ve veri madenciliği gibi çeşitli analizlerin yapıldığı bilişim sistemi uygulamalarından biri kuşkusuz öğrenci bilgi sistemidir (Kahveci ve Özdemir, 2016: 1). Bunun yanı sıra diğer kurum ve organizasyonlar tarafından kullanılan bilişim sistemleri de kullanılmaktadır. Uzaktan eğitim sistemi, insan kaynakları bilgi sistemi, muhasebe-finance bilgi sistemi ve demirbaş takip sistemi üniversitelerde kullanılan diğer sistemlere örnek olarak gösterilebilir (Özcan, 2003: 2; Tecim ve Gökşen, 2009: 2241).

2.2.2.1. Öğrenci Bilgi Sistemleri

Öğrenci bilgi sistemleri, esas olarak öğrenciyle ilgili tüm işlemleri gerçekleştirmeye odaklanan bir sistemdir. Öğrencilerin üniversiteye ilk kaydından itibaren mezuniyetine kadar öğrenciyle ilgili tüm eğitim-öğretim içeriğinin düzenlenmesine ve uygulanmasına yardım eder (Kaçtıoğlu, Keskinçilic ve Kahveci, 2019: 1263).

Öğrenci bilgi sistemine diğer bilişim sistemleri ile veri alışverişini sağlayan alt modüller de eklenerek kullanılmaktadır. Örneğin, öğrenci ders kaydını öğrenci bilgi sisteminde gerçekleştirebilmesi için ödeme kontrolü ya da ödemeyi kabul edebilen bir modül kullanılabilir. Öğrenci bilgi sisteminde kullanılan bir muhasebe modülü öğrencinin kayıt veya muhasebe ile ilişkili diğer işlemleri sırasında muhasebe sistemi ile veri iletişimini sağlamaktadır.

2.2.2.2. Uzaktan Eğitim Sistemi

Uzaktan eğitim sistemi, eğitim amacıyla farklı coğrafyalardaki öğrenci, öğretmen ve öğretim materyallerinin bir araya getirilmesini sağlayan iletişim teknolojilerini içerir

(Barış ve Mevsim, 2014: 296). Ancak, hiçbir zaman eğitimin tamamı anlamına da gelmemektedir (Al ve Madran, 2004: 7).

Uzaktan eğitim sistemi, eğitim yönetim sistemi ile birebir ilişki içinde bulunması gereklidir. Bu sayede sadece ders içeriğinin elektronik ortamda bulunmasının yanı sıra birçok fonksiyonu içinde barındıran bir sistem oluşturulmaktadır. Ders içeriklerinin hazırlanması, öğrenci kayıtlarının tutulması, sistemin kullanım saatleri ve sıklıkları, öğrencilerin başarı durumu gibi istatistiki bilgilerin sistem üzerinden alınmasına ve web ortamına aktarılmasına olanak sağlamalıdır. Ders içeriğinin erişilebilmesi de diğer bir en önemli unsurdur. Öğrenciler sistem dâhilindeki içeriğe istedikleri zaman ulaşabilmeli ve kaynaklardan istedikleri ölçüde faydalanabilmelidirler (Al ve Madran, 2004: 7).

2.2.2.3. İnsan Kaynakları Bilgi Sistemleri

İnsan kaynakları bilgi sistemi kurum personeliyle ilgili tüm işlemlerin koordinasyonunu sağlamaya yönelik bir sistemdir. Bu nedenle, personel bilgi sistemi olarak da adlandırılmaktadır (Yavuz, 2014: 56). Üniversiteye personel kabulünden ve çıkarılmasına kadar personel ile ilgili üniversite için gerekli tüm bilgilerin toplanması, düzenlenmesi ve raporlanması gibi temel süreçler ile ilgilenmektedir.

Ngai ve Wat (2006: 297)'nin çalışmalarında olduğu gibi kurumlar bir anket veya benzeri bir çalışma ile insan kaynakları bilgi sistemlerinin etkinliğini daha da artırmaya çalışmaktadırlar. Elde edilen değerlendirmeler ayrıca karar alma süreçlerinde de kullanılmaya başlanmıştır. Gelişmiş bir insan kaynakları bilgi sistemi üniversitede yer alan farklı birimlerin faydalanabileceği bir şekilde tasarlanması gerekmektedir. Rutin personel ile ilgili işlemler, araştırma ve rapor üretme ve karar destek insan kaynakları bilgi sistemlerinden beklenen temel özelliklerdir (Samkarpa, 2013: 22).

2.2.2.4. Muhasebe-Finansman Bilgi Sistemi

Üniversiteler de dahil olmak üzere tüm kurum ve organizasyonlar muhakkak muhasebe ve finans konularıyla içi içedir. Miktarı ne kadar olursa olsun ekonomik faaliyetlerin yönetilmesi üniversite için önem arz etmektedir. Muhasebe ve finans bilgi sistemleri üniversitenin gelir, gider ve kaynakları düzenlemede çok büyük kolaylıklar

sağlarlar. Özellikle geçmiş verilerin kolaylıkla analiz edilebilmesi sayesinde kurumsal kararların alınmasında yol göstericidir (Yavuz, 2014: 31).

Muhasebe bilgi sistemleri ve finansman bilgi sistemleri genel olarak muhasebe-finansman bilgi sistemi olarak adlandırılmaktadır. Bu iki bilgi sistemi, uygulamada ayrı ayrı kullanılabilirdiği gibi tek bir bilgi sistemi altında da kullanılabilmektedir. Muhasebe bilgi sistemi kurum içi ayrıntılı verilere odaklanırken finansman bilgi sistemi kurum dışı verilere ve muhasebe bilgi sisteminin sağladığı kurum içi özet bilgilere odaklanmaktadır. Dolayısıyla, iki sistem birbirinin tamamlayıcısı olarak işlevlerini yerine getirmektedir (Ömürbek, 2003: 117).

Üniversitenin muhasebe ve finans birimi iş süreçlerini belirli bir sistem içerisinde yürütmek zorundadır. Muhasebe ve finans bilgi sistemleri üniversite girdilerini, süreçlerini ve çıktılarını mali açıdan kontrol altına alınmasını sağlar. Bu nedenle, muhasebe-finans bilgi sistemleri sistematik olarak işleyen mali sürecin yönetilmesine büyük destek sağlamaktadır.

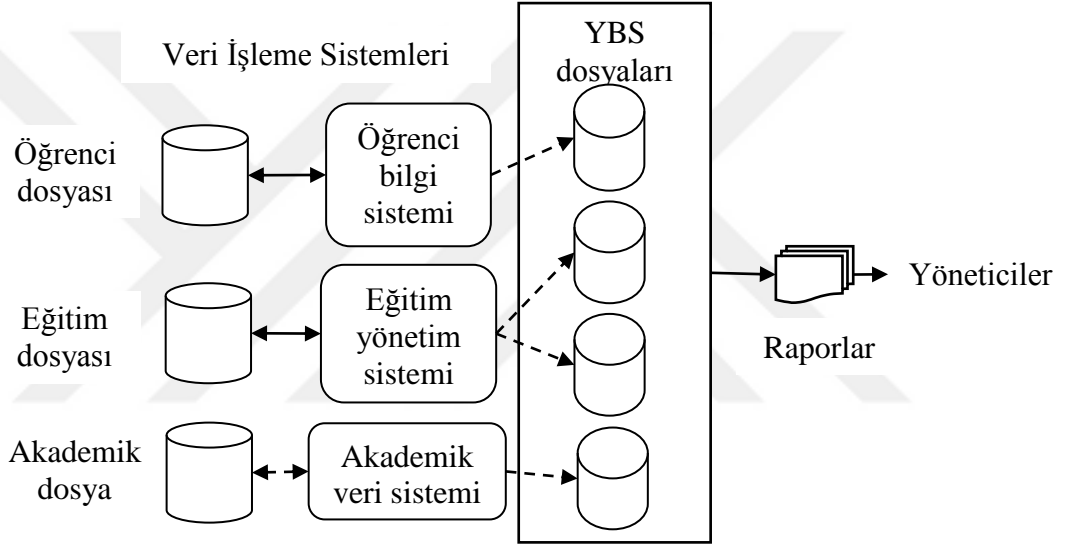
2.2.2.5. Demirbaş Takip Sistemi

Üniversiteler faaliyetlerinin gerçekleştirilebilmesi için bir takım araç ve demirbaşlara ihtiyaç duyarlar. Bu araç ve demirbaşların verimli ve yerinde kullanılabilmesi için kim tarafından, ne zaman ve nerede kullanıldığı bilgisi önemlidir. Örnek olarak, bazen bir personelin ihtiyaç duyduğu bir malzeme üniversite içerisinde pasif halde bulunabilir. Bu malzemenin gereksiz yere tekrar satın alınmaması için üniversitenin sahip olduğu demirbaşlar kayıt altında bulundurulması gerekmektedir. Demirbaş takip sistemi, üniversite varlıklarını ilgili zamanda ilgili personele veya ekibe tahsis edilmesine yardımcı olurlar.

Geliştirilen yeni teknolojiler demirbaş takip sistemlerinin etkinliğini artırmaktadır. Radyo Frekanslı Tanımla (RFID - Radio Frequency Identification) bu teknolojilerden biridir. Bu sayede demirbaşların eşzamanlı olarak takibi ve yönetimi sağlanabilmektedir. Özellikle demirbaş sayımının klasik yöntemlere göre daha hızlı ve kolay bir şekilde gerçekleştirilebilmektedir. Ayrıca demirbaşlar ile ilgili geri dönük bilgilere ulaşma, istatistiksel demirbaş ve personele tahsis raporu alma gibi birçok avantajı bulunmaktadır (Essed, 2019: 1).

2.2.3. Yönetim Düzeylerine Göre Bilişim Sistemleri

Üniversitede farklı düzey ve birim yöneticiler üniversiteyle ilgili farklı bilgilere ihtiyaç duyabilmektedirler. Bu bilgilerin zamanında ve tam olarak sağlanması yöneticilerin kararları açısından büyük önem arz etmektedir. Yöneticilerin veri işleme sistemlerinden bilgi edinme sürecindeki aşamaları veri işleme sistemleri, yönetim bilgi sistemleri, karar destek sistemleri ve üst yönetim bilgi sistemleri olarak sıralanmaktadır (Tanrikulu, 1999: 22). Özbay (2015: 266)'ın aktardığına göre, üniversite yönetimine sunularak stratejik karar alma ilk defa Sanjeev ve Zytkow (1995) tarafından üniversite veri tabanları kullanılarak gerçekleştirilmiştir.



Şekil 2.3. Yöneticilerin veri işleme sistemlerinden bilgi edinme süreci

Şekil 2.3'de K. C. Laudon ve J. P. Laudon (2014: 80)'dan üniversitelere yönelik olarak uyarlanmış yöneticilerin veri işleme sistemlerinden bilgi edinme süreci gösterilmektedir. Şekilde de görüldüğü üzere, yöneticilerin veri işleme sistemlerinden bilgi edinme sürecinde veri işleme sistemleri, YBS dosyaları ve raporlar olmak üzere üç önemli aşama bulunmaktadır.

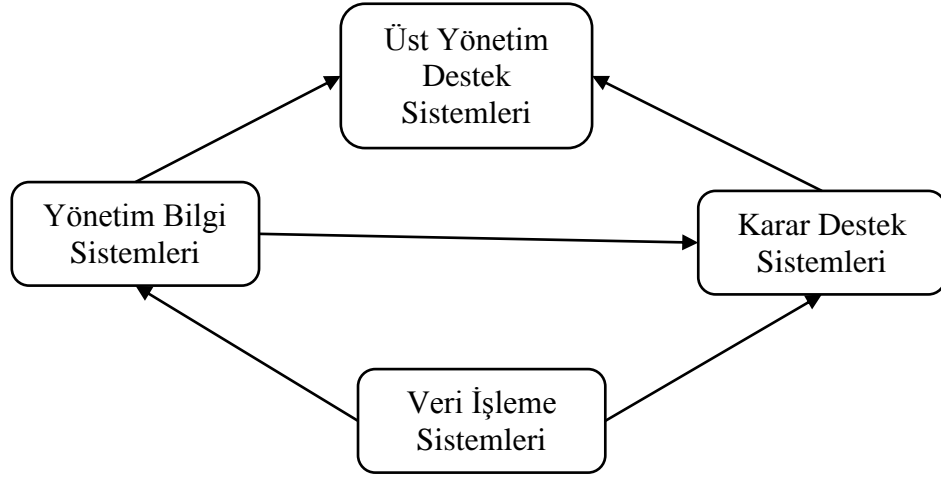
Veri işleme sistemleri, yöneticilere ulaşan her türlü bilgilerin kaynağını teşkil etmektedir. Öğrenci bilgi sistemi, eğitim yönetim sistemi, akademik veri sistemi vb. gibi veri işleme sistemleri iş süreçlerinde kullanılan verileri veri tabanı dosyalarına kaydetmektedir. Her bir veri işleme sistemi kendine ait bir veri tabanı dosyasına sahiptir. Bu veri tabanı dosyasındaki veriler veri işleme sistemleri aracılığıyla yönetim

bilişim sistemlerinin veri tabanı dosyasına aktarılmaktadır (Anameriç, 2005: 34; Koçoğlu, 2010: 100).

Yönetim bilgi sistemleri, kurumun farklı seviyelerinde ve birimlerinde görev yapan tüm yöneticilere hizmet vermeyi amaçlamaktadır. Her bir yönetim düzeyi farklı türde problemler, bilgi ihtiyacı ve kararlar ile ilgilendirilir. Dolayısıyla, bunları destekleyen bilişim sistemleri de yönetim düzeylerine göre farklılık arz etmektedir. Her bir yönetim bilgi sistemi veri tabanı dosyası belirli bir veri işleme sisteminden gelen verileri kullanabileceği gibi farklı veri işleme sistemlerinden gelen verileri de kullanabilmektedir. Yönetim bilgi sistemleri de yöneticilerin karar vermelerine raporlarla yardımcı olmaktadır. Raporlar veri işleme sistemlerinin bir ürünüdür. Başka bir deyişle üniversite içi işlemlerin bir özetidir. Yönetim bilgi sistemleri tarafından veri işleme sistemlerinden toplanan veriler sayesinde gerekli raporlar oluşturulur. Raporların oluşturulmasında yöneticilerin ilgilendiği konularla ilgili ve problem çözümüne yönelik olarak sunulmasına dikkat edilmelidir (Anameriç, 2005: 28; Miman, Yoğun ve Yoğun, 2016: 161).

Karar destek sistemleri, özellikle yönetim düzeyindeki yöneticilerin kararlarına yardımcı olacak şekilde tasarlanmış sistemlerdir. Karar destek sistemleri, çok büyük miktardaki verilere erişim sağlama, birleştirme ve analiz araçlarını içermektedir. Yöneticiler bazı kararlarında üniversite dışı faktörleri de göz önünde bulundurmaya zorundadırlar. Bu noktada karar destek sistemleri gerek üniversite içi gerekse de üniversite dışı bilgileri sunabilmektedir (Yavuz, 2014: 61). Karar destek sistemleri sayesinde yöneticiler kuruma farklı bakış açıları ile bakma imkânı elde ederler. Yöneticiler bu farklı bakış açıları sayesinde daha etkin kararlar verebilmektedirler.

Üst yönetim bilgi sistemleri, özellikle üst düzey yöneticilerin uzun vadeli ve stratejik kararlarına yardımcı olmak üzere tasarlanmış sistemlerdir. Yönetim bilgi sistemleri ve karar destek sistemlerinden gelen bilgilerden faydalanılır. Ancak, üst düzey yöneticiler bu sistemlerden gelen bilgileri direk olarak karar vermede kullanamamaktadırlar. Üst düzey yöneticilerin yapısal olmayan kararlarında kullanabilmeleri için bu bilgileri değerlendirip farklı açılardan yorumlamaları gerekmektedir (Ömürbek, 2003: 102).



Şekil 2.4. Bilişim Sistemleri arasındaki etkileşim.

Şekil 2.4’de üniversite içinde kullanılan bilişim sistemleri arasındaki etkileşim gösterilmektedir (Tanrıkulu, 1999: 20). Buna göre, veri işleme sistemleri diğer bilişim sistemlerinin kaynağı konumundadır. Üst yönetim destek sistemleri üniversitede kullanılan tüm bilişim sistemlerinden faydalanmaktadır. Ancak, veri işleme sistemlerinin kullandığı verileri direk kullanmak yerine yönetim bilgi sistemleri ve karar destek sistemleri aracılığıyla kullanmaktadır. Ayrıca karar destek sistemi, yönetim bilgi sistemi verilerini kullanarak üst yönetim destek sistemlerine bilgi sağlayabilmektedir.

2.2.4. Üniversitelerde Bilişim Sistemi Tedariki ve Prototipleme

Üniversitelerde ihtiyaç duyulan bilişim sistemleri bilgi işlem daire başkanlığı tarafından belirlenir. Bilişim sistemi projelerinin kullanım raporlarını hazırlar. Ayrıca üniversite hiyerarşisinin en üst seviyesinde gerçekleştirilen stratejik planının hazırlanmasına da katkı sağlar (Ertekin, 2006: 81). Bunun için portföy analizi, skor modeli, somut-soyut yararlar, kritik başarı faktörleri, kurumsal analiz yöntemlerinden faydalanılır (K. C. Laudon ve J. P. Laudon, 2014: 566).

Portföy analizinde üniversitenin bütünü analiz edilmektedir. Yatırım yapılacak bir bilişim sisteminin üniversiteye olan yararları, zararları ve riskleri ortaya konmaktadır. Bu süreç SWOT analizi, misyon ve vizyon, strateji ve değerlendirme faaliyetleri ile gerçekleştirilebilmektedir. Güçlü ve zayıf yönler, fırsatlar ve tehditler (SWOT - Strengths, Weaknesses, Opportunities, Threads) analizi bilişim sisteminin durumunun değerlendirilmesine yardımcı olan yöntemlerden biridir (Başaran, 2016: 48). Bilişim

sisteminin yerine getireceği görevlerin açıkça ortaya konmasına yardımcı olmaktadır. Belirlenen bilişim sisteminin hedefe nasıl ulaşacağını gösteren bir rehber konumundadır.

Skor modelinde, bilişim sisteminin özelliklerine önem derecesine göre ağırlık katsayısı atanarak bir değerlendirme yapılmaktadır (K. C. Laudon ve J. P. Laudon, 2014: 567). Bilişim sisteminin başarısının sağlanması hedeflerin gerçekleştirilmesine bağlıdır. Bu nedenle, ağırlık katsayıları hedefleri gerçekleştirme derecesine göre belirlenebilmektedir.

Somut yararlar yönteminde, bir bilişim sistemin finansal açıdan getireceği yararlar ele alınmaktadır. Somut yarar sağlamaya odaklanan sistemlerden bazıları veri işleme ve birim yönetim sistemleridir. Soyut yararlar sayısal olarak ifade edilemeyen, ancak üniversiteye değer katan yararları kapsamaktadır (K. C. Laudon ve J. P. Laudon, 2014: 568). Soyut yararlar uzun vadede somut yararlar da sağlamaktadır. Bunlara örnek olarak karar verme fonksiyonunun geliştirilmesi sayılabilir. Karar destek sistemleri ve yönetim bilgi sistemleri daha çok soyut yararlar sağlamaya yönelik sistemlerdir.

Kritik başarı faktörleri stratejik analiz sonucu ortaya çıkmaktadır. Kritik başarı faktörleri üniversitenin iç ve dış çevresini kapsamaktadır. Stratejik analiz sürecinde üniversitenin mevcut sistemler, ne elde etmek istediği, hedeflerini nasıl gerçekleştirmek istediği ve süreç takibinin nasıl yapılacağı açıkça belirtilir (K. C. Laudon ve J. P. Laudon, 2014: 568). Üst düzey yöneticiler ile görüşüldükten sonra odak noktaları belirlenir. Belirlenen odak noktaları sayesinde üst düzey yönetim veya bir karar destek sistemi geliştirilebilir.

Kurumsal analiz yöntemine göre, üniversite bilişim sistemi ihtiyaçlarının tam olarak belirlenebilmesi için üniversite bir bütün olarak ele alınır. Üniversitede kullanılan tüm verilerin özellikleri ayrıntılı olarak açıklanır. Bilişim sistemi ihtiyacının belirlendikten sonra temini veya geliştirilmesi için de bazı temel yöntemler bulunmaktadır. Bunlar (Keskinçilic ve Kahveci, 2019: 1072):

- Klasik yaşam döngüsü
- Paket yazılım kullanımı
- Son kullanıcı geliştirmesi
- Dış kaynak kullanımı
- Prototipleme

Klasik yaşam döngüsü: Ağır süreçler olarak da adlandırılır. Şelale (waterfall), V-şekilli (V-Shaped), artırımsal (incremental), spiral (spiral), birleştirilmiş süreç (Unified Process) en çok bilinen süreç yazılım süreç modelleridir (Awad, 2005: 3). Analiz, tasarım, kodlama, test ve operasyon adımları sırasıyla uygulanır. Adımlarda geriye dönüşler çok maliyetli ve zaman aldığından gereksinimlerin çok iyi bilindiği ve esnek bir yapıya sahip olmayan yazılım projelerinde kullanılmaktadır. Genellikle uzay mekiği, nükleer santraller, askeri yazılımlar gibi özel kullanıma mahsus alanlarda kullanılmaktadır (Ranniko, 2011, 2011: 3).

Paket yazılım kullanımı: Üniversitenin kısa zamanda yazılım tedarikini sağlayan ve en çok tercih edilen seçeneklerden birisidir (Bayraktar ve Efe, 2006: 100). Paket yazılımlar, üniversitelerin belirli üniversite fonksiyonları için program ihtiyacını sağlayan, muhasebe, stok kontrol, personel takip gibi hazır ticari yazılımlardır.

Son kullanıcı geliştirmesi: Dördüncü nesil araçlar sayesinde kullanıcıların teknik uzmanlardan çok az yardım veya hiç bir yardım almadan bilişim sistemi geliştirmesidir. Kısaca hatırlamak gerekirse; birinci nesil makine dilidir, ikinci nesil çevirici assembler dilleri, üçüncü nesil yapısal dillerdir, dördüncü nesil diller kullanıcıların kendi ihtiyaçlarını giderebilecek düzeyde yazılım üretmesini sağlayan temel seviyede teknik bilgi gerektiren yazılım araçlarıdır. Klasik programlama dillerine göre daha az prosedürel yapıya sahiptir. Access, Excel, FrontPage, SQL dördüncü nesil araçlarından bazılarıdır (Ömürbek, 2003: 36). Beşinci nesil diller bilgisayarın çözümü kendisi bulmasıdır. Yapay zekâya dayalıdır ve çevik metodolojide kullanılır.

Çevik metodoloji, yazılımın geliştirilmeye başlanmasından sonra müşterinin isteklerinin değişmesi sonucu sürekli geriye dönüşlerin olduğu esnek yapıya sahip yazılım projelerinde kullanılır (Aybek, 2010: 4). Scrum, XP-Extreme Programming, Gereksinim Güdümlü Uygulama Geliştirme (RDSD- Requirements Driven Software Development), Rasyonel Birleştirilmiş Süreç (RUP- Rational Unified Process), Çevik Birleştirilmiş Süreç (AUP- Agile Unified Process), Temel Birleştirilmiş Süreç (EssUP- Essential Unified Process), Yazılım Yaşam Döngüsü (ALM-Application Lifecycle Management), Test Otomasyonu (TA- Test Automation), İşlerin Ayırımı (SoC- Separation of Concerns), Açık Birleştirilmiş Süreç (OpenUP- Open Unified Process), Takım Yazılım Süreci (TSP- Team Software Process), Microsoft Çözüm Çerçevesi

(MSF- Microsoft Solutions Framework), Test Gdml Yazılım (TDD- Test Driven Development), Hızlı Uygulama Geliřtirme (RAD- Rapid Application Development), Nesne Ynelimli Programlama (OOP- Object Oriented Programming), Servis Odaklı Yazılım Geliřtirme (SOA- Service Oriented Architecture), Bileřen Ynelimli Yazılım Mhendislięi (CBD- Component Based Development), İlgiye Ynelik Programlama (AOP Aspect Oriented Programming), Model Gdml Yazılım Geliřtirme (MDA- Model Driven Architecture), Srekli Entegrasyon (CI- Continuous Integration), Yetenek Olgunluk Modeli (CMM- Capability Maturity Model), ISO 15504 (SPICE), Yeniden Tasarım (Refactoring), Yalın Yazılım Geliřtirme (LSD- Lean Software Development), zellik Gdml Geliřtirme (FDD - Feature-Driven Development), Lean Development, Dinamik Sistem Geliřtirme Metodu (DSDM - Dynamic Systems Development Method) ve Crystal en ok kullanılanlarıdır (Versionone, 2016: 1).

Dıř kaynak kullanımı: niversitelerin bir yazılım firması yardımıyla donanım ve yazılım kaynaklarını ynetmesidir. Uygulama hizmeti saęlayıcılar da niversitelerce dıř kaynak kullanım tercihleri arasında yer almaktadır. niversitenin dnya standartlarında teknoloji kullanımını, kolayca bařarıya eriřimini ve maliyetten tasarruf saęlar (Turgut, Bulut ve Altunel, 2019 :2).

niversitelerin kurumsal yapıları ve iřleyiřleri genellikle buldukları lkenin yksekğrenim kurumu tarafından belirlenmektedir. niversitelerde kullanılan biliřim sistemleri yksekğretim kurumu kurallarını uygulamak durumundadır. Bu nedenle, yksekğretim kuralları gz nnde bulundurularak geliřtirilen niversitelere ynelik genel bir biliřim sistemi herhangi bir niversite tarafından kolaylıkla kullanılabilir. Bu alanda hizmet veren oęu yazılım řirketi alıřmalarını bu hususu gz nnde bulundurarak yrtmektedir. Nitekim yapılan arařtırmalara gre niversitelerin oęunluęu biliřim sistemi tedarikini dıř kaynak kullanımı yoluyla yazılım řirketlerinden saęlamaktadır (zcan, 2003: 2). Bu alanda faaliyet gsteren řirketler yazılımlarını niversitelerin genel kurallarına gre bir řablon oluřturduktan sonra yazılımı niversitenin dięer ihtiyalarına ynelik zeleřtirilebilecek řekilde tasarlamaktadırlar. Yazılım řirketlerinden biliřim sistemi satın alan niversiteler ayrıca teknik destek ve bakım servisi de almaktadırlar.

Dış kaynak kullanımı, gerek üniversiteler gerekse de diğer kurumlar tarafından ilk düşünülen bilişim sistemi temini yöntemidir (Tecim ve Gökşen, 2009: 2243). Üniversiteler açısından yazılım şirketlerinden bilişim sistemi satın alınmasının birçok faydası bulunmaktadır. Dış kaynak kullanımı ile sağlanan sistemin daha önce diğer üniversiteler veya üniversiteler tarafından denenmiş olması, maliyetinin düşük olması, kısa zamanda kullanıma geçilebilmesi, ek personel ve yerleşke gerektirmemesi gibi birçok avantajı bulunmaktadır. Bilişim sistemlerinde meydana gelen arızaların giderilmesi için anlaşma yapılan yazılım şirketi sorumlu tutabilmektedir. Böylece üniversite de eğitim-öğretim faaliyetlerine daha iyi yoğunlaşma imkânı bulmaktadır. Bilişim sistemlerinin üniversite bünyesinde kurulmasından doğan olan bina, donanım, yazılım, personel, vb. gibi maliyet kalemlerinin her biri ile ayrı ayrı ilgilenilmesi gerekmemektedir. Bütün bu konuları yazılım şirketi ile anlaşarak tek kalemde sonuçlandırılabilir. Yeni açılan bir üniversitenin bir öğrenci bilgi sistemi temini etmek istediği varsayılırsa bunun hızlı bir şekilde gerçekleştirebilmesi için yazılım şirketlerine başvurması en muhtemel olasılıklardan biridir. Çünkü bu sistemi kendi bünyesinde geliştirmek zaman alacaktır (Ranniko, 2011: 62).

Üniversiteler açısından bilişim sisteminin yazılım şirketlerinden temin etmenin avantajlarının yanı sıra uzun vadede göz önünde bulundurulması gereken ciddi bazı dezavantajları bulunmaktadır. Üniversitelerde binlerce öğrenci ve çalışanın bulunduğu göz önünde bulundurulduğunda dezavantajlar için önemli tedbirlerin alınması gerekmektedir.

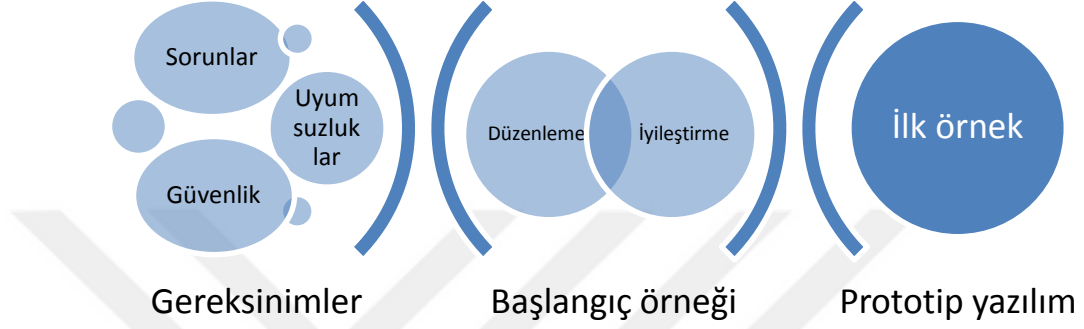
Yazılım şirketleri kar amacı güden ticari işletmelerdir. Yazılım şirketleri ile her ne kadar anlaşmalar yapılsa da şirketinin faaliyetlerine ne kadar süre devam edeceği belirsizdir. Yazılım şirketi bazı ciddi mali sorunlar ya da başka sorunlar yaşaması durumunda faaliyetlerini sonlandırma kararı alabilir. Bu durumda üniversite satın aldığı bu bilişim sistemlerini tamamen bırakarak tekrardan farklı bir yazılım şirketi ile anlaşma yoluna gitmesi gerekmektedir. Bu da üniversite açısından bilişim sistemi maliyetini iyice katlayacaktır. Ayrıca bu geçiş sürecinde binlerce kullanıcısı olan sistemlerin kesintiye uğraması kaçınılmazdır. Yazılım şirketine ileriki zamanlarda ulaşamama durumunda bilişim sisteminin devam ettirilmesi konusu sürekli gündemde tutulmalı ve planlar yapılması gerekmektedir. Bu sorunun çözümü için üniversitenin ikincil bir planı olmadır. Örneğin, anlaşma yapılan yazılım şirketinin ileriki zamanlarda kapanma

ihtimaline karşı ayrıca kurum içi bilişim sistemi ile sorumlu personel tahsis etmelidir. Bu personelin bilişim sistemini satın alma işleminden itibaren bu sistemi ayrıntılı tanınması sağlanmalıdır. İleride yazılım şirketi ile yaşanabilecek olan herhangi bir problemle karşılaşıldığında bu personel devreye sokulmalıdır. Böyle olumsuz bir durum ile karşılaşıldığında sistemi devam ettirmek için yetiştirilen personelin içerisinde bulunduğu bir birim hızlı bir şekilde oluşturulabilmelidir. Bu konunun satın alınan yazılım şirketi ile de görüşülmesi de bu sorunun çözümüne katkı sağlayacaktır.

Dezavantajlardan biri de verilerin yazılım şirketinin veri tabanında bulundurulması veya bulut depolama kullanımınıdır (Ruggiero ve Heckathorn, 2013: 2). Üniversite içerisinde veri tabanı kurulumunu gerçekleştiren yazılım şirketleri de mevcut ama bu hem maliyeti artırmakta hem de yazılım şirketi ile anlaşmanın sağladığı avantajları azaltmaktadır. Kurum içi birim kurulsada yer, donanım, personel, zaman, kurum yapısı vb. gibi kaynaklara ne oranda sahip olduğun iyi değerlendirilmesi gerekmektedir (Keskinç ve Kahveci, 2019: 1072). Verilerin güvenliğinin ve erişilebilirliğinin sağlanması için özel donanım ve uzman personel gerekmektedir. Geliştirilecek sistem için bazen personelin eğitilmesi ve araştırma sürecine girilmesi gerekmektedir. Bu ise istenilen sistemin gerçekleştirme bazen de istenilen kaliteye ulaşma sürecini uzatmaktadır (Tecim ve Gökşen, 2009: 2243).

Bütün üniversitelerin aynı bilişim sistemini kullandığı bir ortamda bilişim sistemi yönüyle bir avantajdan bahsetmek mümkün değildir. Benzer bilişim sistemlerinin kullanılması diğer kurumlara ve üniversitelere bilişim sistemlerinin sağladığı avantajlar açısından fark oluşturulamaz. Bu tür dezavantajları ortadan kaldırmak amacıyla önde gelen bazı kurumlar ve üniversiteler kendi bilişim sistemlerini geliştirme yolunu tercih etmektedirler. Sistem geliştiriciler kendi personeli olduğundan üniversitenin ihtiyaçlarını neredeyse tamamını karşılayacak şekilde tasarlayabilmektedir. Böyle bir sistemde ne fazla ne de eksik bir modül söz konusu değildir. Yazılım personeli bilişim sistemi konusunda her konuda üniversiteye karşı tam sorumluluk altındadır. Bilişim sistemi personeli yeni gelen personeli eğitmektedir. Bilişim sisteminin sürekliliği açısından da üniversitenin herhangi bir kaygısı bulunmamaktadır. Zamanla üniversitenin ihtiyaçları değiştikçe bilişim sisteminin şekillendirilmesi daha kolay ve hızlı gerçekleşmektedir. Bu tür sistemlerin verileri genellikle kurum içinde saklandığından verilerin kontrolü de tamamen üniversiteye aittir.

Prototipleme; faaliyete ya da seri üretime geçirilecek ürünün ilk örneğidir. Bilişim sistemleri açısından bir sistemin tümünün veya bir parçasının çalışan ilk sürümüdür. Şekil 2.5’de görüldüğü üzere, gereksinimler, başlangıç örneği, düzenleme ve iyileştirme olmak üzere prototiplemenin dört adımı bulunmaktadır (Camden ve diğ., 2006: 18).



Şekil 2.5. Prototip yazılım geliştirme süreci.

Prototipleme, sistem geliştirme yaşam döngüsü olarak Hızlı Uygulama Geliştirme (RAD – Rapid Application Development) adıyla kullanılmaktadır. Hızlı, gelişimsel, artırimsal ve uç (extreme) prototipleme olmak üzere dört çeşit prototipleme vardır (Crinnion, 1991: 17; Kis, 2016: 15):

Hızlı prototipleme; prototip yazılımın kolaylıkla test edebileceği bir arayüz geliştirilmektedir. Bu sayede temel gereksinimlerin hızlı bir şekilde ortaya çıkarılması amaçlanmaktadır (Ranniko, 2011: 19).

Gelişimsel prototipleme; temel gereksinimlerin belli olduğu, ancak detay gereksinimlerin belirsiz olduğu durumlarda kullanılmaktadır. Bu detay belirsizliklerin giderilmesi amacıyla geliştirilen prototip yazılım sunulur ve istenilen düzeltmeler doğrultusunda prototip yazılım geliştirilerek tekrar sunulur. Bu şekilde prototip yazılım sürekli geliştirilerek yeni özellikler eklenir veya gereksinimlere uymayan özellikler çıkarılır (Kis, 2016: 15).

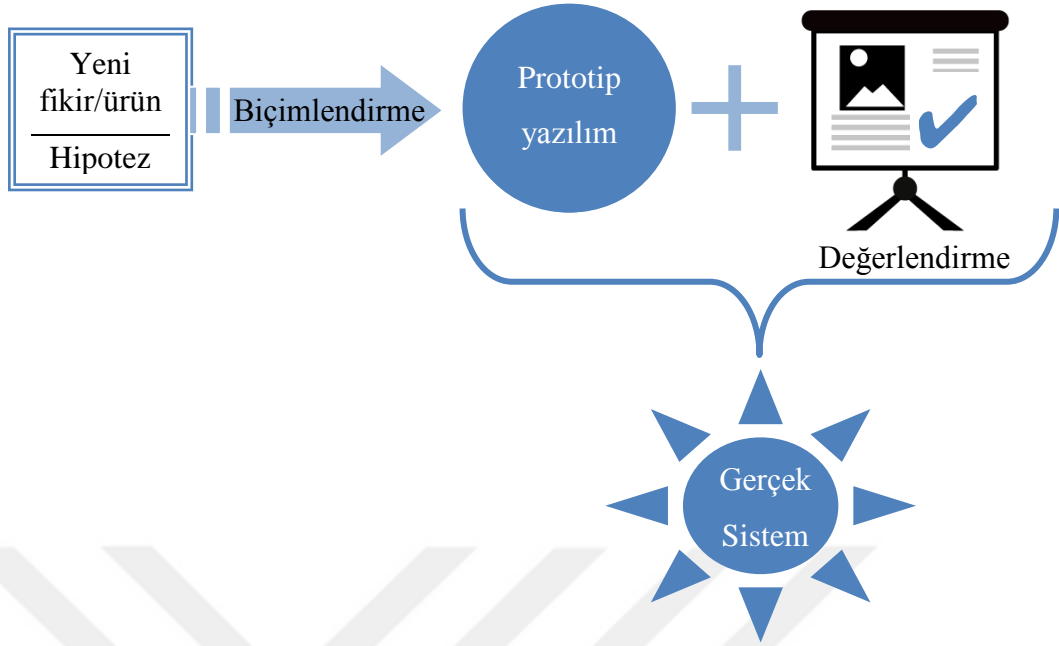
Artırimsal prototipleme; ayrı ayrı üretilen küçük çaplı prototipler tamamlandığında birleştirilerek bütünü oluşturan genel bir prototip yazılım elde edilir. Kısa sürede tamamlanan küçük prototipler değerlendirilirken diğer taraftan yazılım

geliştiriciler küçük prototipler üzerinde çalışmaya devam eder. Bu sayede değerlendirme ve geliştirme arasındaki bekleme süresi azaltılmış olur (Crinnion, 1991: 18).

Uç (extreme) prototipleme; özellikle web uygulamalarının geliştirilmesinde kullanılmaktadır. Her biri bir öncekinin devamı niteliğinde üç aşamadan oluşur. İlk aşamada, genellikle Hiper Metin İşaret Dili (HTML - Hyper Text Markup Language) sayfalarından oluşan statik web sayfaları oluşturulur. İkinci aşamada, sayfalar tüm fonksiyonları yerine getirecek şekilde programlanır. Üçüncü aşamada, web uygulamanın sağlayacağı tüm servisler çalıştırılmaktadır. Uç programlama olarak adlandırılmasının nedeni ikinci aşamaya dayanmaktadır. Bu aşamada tam fonksiyonlu bir kullanıcı arayüzü tasarlanmaktadır. Bu fonksiyonlar talep edilen servislerin haricindekileri de yerine getirebilecek geniş bir servis yelpazesi sunar. Böylece talepteki herhangi bir değişiklik hızlı bir şekilde yerine getirilebilmektedir (Crinnion, 1991: 18).

2.2.4.1. Prototip Yazılımın Testinin Yapılması

Prototipleme, bir süreci veya kavramı test etmek üzere oluşturulmuş bir ürünün ilk örneğidir ve en iyi test modelleridir (Pressman, 2001: 31; Blackwell, 2015: 1624). Prototipleme gerçek yazılımın bir tezahürüdür ve tasarımın oluşmasını sağlayan veya tasarımı olumsuz etkileyen özelliklerin ortaya çıkarılmasına yardımcı olur (Rogers, 2013: 1). Prototiplemeden elde edilen tecrübe gerçek ürüne sorunsuz bir şekilde yansıtılabilmektedir (Blackwell, 2015: 1624).



Şekil 2.6. Hipotez, prototip yazılım ve gerçek sistem arasındaki ilişki

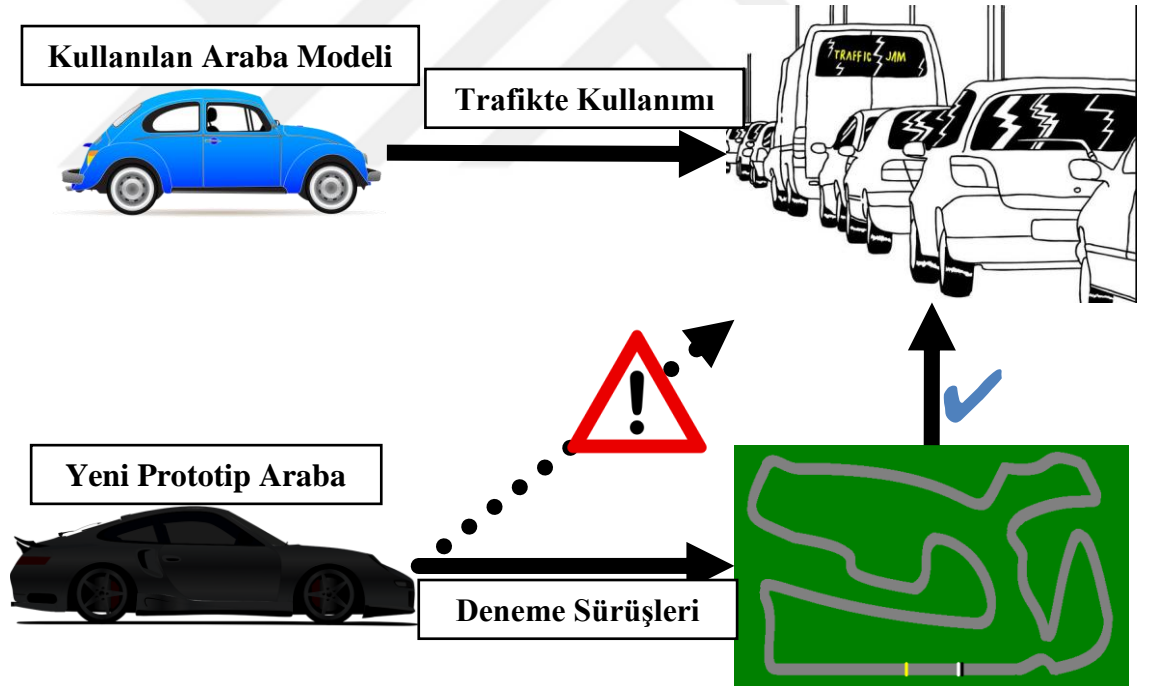
Şekil 2.6’da görüleceği üzere prototip yazılımlar, üniversitelerin yeni fikirlerini gerçek hayatta uygulamadan önce denenmesinde ve yeni bir fikrin sunumunda kullanılmaktadırlar. Yeni fikir veya ürün için geliştirilen teori ve hipotezlerin kanıtlanmasını sağlayan teknik özelliklere sahiptir (Kis, 2016: 15). Bazı tasarım iş akışı modellerinde, somutlaştırma olarak adlandırılan bir süreç olarak kullanılmaktadır (Özdemir, 2017: 1). Prototipleme, bir fikrin biçimlendirilmesi ve değerlendirilmesi arasındaki süreçtir (Soares ve Rebelo, 2012: 331). Palm Pilot’un tasarımcısı ilk zamanlarda tasarladığı cihazı temsilen aynı ebatlarda bir tahta parçasını cebinde taşıyordu. “Rahat kullanım için ebatları uygun mu? Elle tutulması rahat mı? Düğmelerin konumu kullanışlı mı?” gibi sorulara yanıt bulmaya çalışıyordu (Dolek, 2007: 1).

Prototip yazılımının testinde kullanılan önemli kavramlardan biri de veri prototiplemedir. Veri prototipleme, projenin başlangıç aşamasında girdi olarak kullanılacak ilgili tüm veri örneklerini temsil eder. Veri prototiplemenin kullanım amacı veri temizleme ve dönüştürme kurallarını oluşturma ve bu kuralların uygulanması neticesinde elde edilen veri setinin elde edilmesidir (Kis, 2016: 33).

2.2.4.2. Prototip Yazılımın Gerçek Yazılıma Olan Faydaları

Prototipleme ile geliştirilen yazılım çalışır, kullanılabilir, kaliteli, güvenli ve kullanıcı dostu bir arayüz ile tasarlanmış olur. Prototip yazılımlar; gerçek hayatta kullanılan sistemler değil, gerçek hayatta kullanılabilecek sisteme en yakın halidir (Korkmaz, 2015: 1). Sistemin gerçek kullanıcıları açısından kullanıcı arayüzünde, gerçek yazılım ile prototip yazılım arasında hiçbir fark yoktur (Camden ve diğ., 2006:18).

Prototipten toplanan bilgiler ile gerçek yazılım oluşturulabilir ya da gerçek yazılımda değişiklikler yapılabilir (Rogers, 2013: 1). Bu yöntem ile giderilen sorunlar ve problemler sistemin gerçek hayatta çalıştırılması sırasında hiçbiri ile karşılaşılmaz veya sorunlar en aza indirgenmiştir. Prototipleme yöntemi kullanılmadan geliştirilen yazılımlarda, sistemlerdeki uyumsuzluklar nedeni ile kurumlar maddi hasarlara uğramakta, sistem sürekliliğini sağlamakta zorluk çekmekte ve gelişmemektedir.



Şekil 2.7. Prototip yazılımın gerçek hayatta kullanımının temsili gösterimi.

Şekil 2.7’de prototip bir yazılımın gerçek hayatta kullanım süreci temsili olarak yeni bir arabanın trafikte kullanımı üzerinde gösterilmektedir. Şekilde görüleceği üzere bilinen ve kullanılan bir arabanın trafikte kullanılması gayet doğaldır. Aynı şekilde zaten var olan ve kullanılan bir uygulamanın çeşitli üniversitelerce kullanımı ve fayda

sağlaması da normaldir. Kullanımda olan araba ya da uygulama hakkında kullanıcıların memnuniyeti, şikâyetleri, vb. gibi durumlar araştırma söz konusu olabilir. Ancak, henüz daha yeni bir fikir olarak ortaya çıkmış olan prototip bir arabanın ilk örneğinin direk olarak trafikte kullanımı çeşitli tehlikeler arz edebilir. Gerek mal gerekse de can kaybı ile sonuçlanabilecek büyük hasarlar meydana getirebilir. Aynı şekilde yeni bir prototip yazılımın direk olarak gerçek hayatta üniversitelerce kullanılması büyük hasarlara sebebiyet verebilir. Prototip bir araba gerçek hayatta karşılaşılabileceği göz önünde bulundurularak hazırlanan bir deneme pistinde çeşitli özellikleri, kabiliyetleri ve performansı test edilir. Yeni geliştirilen prototip uygulamalar da öncelikli olarak gerçek hayatta kullanımı göz önünde bulundurularak test edilir. Deneme ortamında herhangi bir sorunla karşılaşılmazsa ya da sorunlar giderildikten sonra gerçek hayatta kullanılmaya başlanır.

2.3. LİTERATÜR ÖZETİ

Literatür özeti, kavramsal literatür ve kullanım alanları dikkate alınarak iki başlık altında özetlenmektedir.

2.3.1. Kavramsal Literatür

Üniversitenin istenilen seviyede eğitim ve akademik çalışmalar sunabilmesi için kurumsal kaynakların ve üniversite içi süreçlerin aktif bir şekilde yönetilmesi gerekmektedir. Yöneticiler üniversitenin fiziksel imkânlarını geliştirmek amacıyla çeşitli faaliyetlerde bulunurlar. Ayrıca yöneticiler kaynakları, akademisyenleri, fiziki ortamı ve kurum yapısını en tutarlı bir şekilde bir arada tutmaya ve sürekliliğinin sağlanmasına çalışırlar. Üniversitelerde bilişim sistemlerinin kullanılması gerek yöneticiler gerekse de akademisyenler için büyük avantajlar sağlamaktadır. Üniversitelerde belirli prosedürlerin uygulanması için geliştirilen yapısal bilişim sistemleri akademisyenlerin günlük faaliyetlerine yardımcı olmaktadır. Öğrenci bilgi sistemi, uzaktan eğitim sistemleri, üniversite bilgi yönetim sistemi, akademik veri sistemleri, vb. üniversitelerde kullanılan bazı bilişim sistemlerine örnek olarak gösterilebilir. Yapısal olmayan henüz üzerinde belirli bir prosedür uygulanmayan bazı işlerin yürütülmesi için yöneticilere yönelik olarak geliştirilen karar destek sistemleri kullanılmaktadır.

Üniversiteler, bilişim sistemlerinin temininde genel olarak iki yöntem kullanılmaktadırlar. Bunlardan biri kendi ihtiyaçları doğrultusunda bilişim sistemi geliştirme diğeri ise yazılım şirketlerinden temin etmedir. Kendi bilişim sistemlerini geliştiren üniversitelerin avantaj ve dezavantajları incelendiğinde genel olarak avantajları daha çok bulunmaktadır. Yazılım şirketinden elde edilmesinin dezavantajlarından daha az riskler içermektedir. Üniversiteler birer yükseköğretim kurumlarıdır ve açıldıktan sonra uzun yıllar hizmet vermeyi amaçlayan kurumlardır. Dolayısıyla, kullanılan sistemlerin sürekliliği büyük önem arz etmektedir. Bu açıdan üniversitelerin kendi bilişim sistemlerini geliştirmesi bu bilişim sistemlerinin sürekliliğine katkıda bulunmaktadır (Tecim ve Gökşen, 2009: 2244). Kendi bilişim sistemini geliştirme sürecinde kullanılan en etkili yöntemlerden biri de prototiplemedir. Geliştirilen prototipler sayesinde kurumlar bilişim sisteminin gerçek hayatta karşılaşılabileceği sorunları en aza indirger hatta hiçbir sorunla karşılaşmaz.

Temsili örnek olarak, prototip bir arabanın gerçek hayatta karşılaşılabileceği manevralarını ve performansını test etmek amacıyla deneme pisti kullanılmaktadır. Bu çalışmada üniversitelerin kendi bünyelerinde geliştirebilecekleri yöneticilere yönelik bir model önerilmektedir. Model ile ilgili nicel olarak sonuçlar elde etmek amacıyla da prototip bir uygulama geliştirilmiştir. Uygulamanın test edilmesi için gerçek hayatta üniversitelerce kullanılan VTYS'ler seçilmiştir. Test için gerekli olan veri tabanları, tablolar, alanlar ve gerçek veri ve veri prototiplemesi kullanılmaktadır.

Grafikler, sahip olduğu görsel özellikleri sayesinde karmaşık olarak görünen birçok veriyi daha anlaşılır bir şekilde sunabilmektedir. Bu yönüyle yöneticilerin daha iyi bilgi almasına yardımcı olmaktadır. Ancak, yöneticilerin kurumsal süreçlerin yolunda gittiğini izleyebilmeleri için sürekli olarak bir grafiğe bakmaları beklenemez. Bu nedenle, grafikler uyarı sistemleri ile birlikte kullanıldıklarında daha işlevsel olmaktadır. Uyarı sistemlerinin temelinde anomali tespiti bulunmaktadır. Verileri zaman boyutu ile incelenebilmesi zaman serilerinde anomali tespiti ile mümkündür. Zaman serisi özelliği taşıyan grafikler temelde aylık, haftalık, günlük, saatlik, dakikalık gibi zaman dilimlerinde kullanılmaktadırlar.

Üniversitelerin akademik takvimi genellikle günlük zaman dilimine göre düzenlenmektedir. Bu nedenle, akademik takvimin uygulanmasında en çok tercih edilen

günlük grafikler sayesinde bazı riskler önceden tahmin edilebilir. Örneğin, günlük zaman dilimi grafiğine göre sistemde aktif kullanıcı sayısı 12 Ocak 2019 tarihi itibarı ile yaklaşık olarak 2250 değeri ile en yüksek seviyeyi görmüş olmasının akademik takvimdeki karşılığı öğrencilerin sınav sonuçlarının açıklanması gibi olaylar olabilir. Dolayısıyla, aynı şekilde bir sonraki sınav sonuçlarının açıklanmasında sistemde aşırı bir yoğunluk oluşması muhtemeldir. Üst düzey yöneticinin bu tür bir bilgiye sahip olması diğer operasyonel ve yönetim düzeyindeki yöneticilerin yaşadıkları problemi daha iyi ifade edebilmelerine yardımcı olacaktır. Bu sayede de uzun vadeli planlarda bu durum göz önünde bulundurulabilir.

Zaman dilimi grafiğinin türüne göre elde edilebilecek bilgi farklılık göstermektedir. Dakikalık, saatlik ve günlük zaman dilimlerine göre bakış açısı ve verilerin değişkenliği farklılık arz etmektedir. Dolayısı ile zaman diliminden elde edilebilecek bilgi ile kısa, orta ve uzun vadede planlar yapmak da mümkündür. Böylece yönetim seviyelerine göre yürütme ve kontrol faaliyetleri için bilgi sağlanabilir. Yöneticiler her düzeyde bilgi ile ilgilenebilirler, ancak dakikalık veriler operasyonel yöneticiler için, saatlik veriler yönetim düzeyindeki yöneticiler için ve günlük veriler üst düzey yöneticilerin kullanımına daha uygun görülmektedir.

Zaman serisi grafiğinin iki boyutu bulunmaktadır. Bunlardan biri zaman boyutudur. Diğeri ise zaman etiketine sahip olan değerdir. Değer boyutu genel olarak sayısal olarak kullanılmaktadır. Sayısal veriler veri grubunu istatistikî yöntemlere dayalı olarak özetlenmesinde kullanılabilir. Kurum ile ilgili herhangi bir konuda özet verilere sahip kurumsal metrikler de bu kapsamda grafiğe dönüştürülebilmektedir. Günümüz kurumları iş süreçlerini bilişim altyapısı üzerinden yürütmektedir. VTYS'ler, bu bilişim altyapısının en önemli unsurudur. Kurumsal metrikler kontrol noktası oluşturma, sorgu havuzu, sorgulama, günlük kayıtların analizi ve tetikleyici oluşturma yöntemleri kullanılarak VTYS'lerden elde edilebilmektedir (Stonebraker, Çetintemel ve Zdonik, 2005: 43; Natan, 2009: 287).

Bu yöntemlerin her birinin avantaj ve dezavantajları bulunmaktadır. Standart özellikler sahip bir sunucu sistemi üzerinde çalışan bir VTYS milisaniyede 1000 kadar işlemi gerçekleştirebilme kapasitesine sahiptir. Eğer ki işlem daha önce gerçekleştirildi ise bunu daha önce kaydettiği sorgu havuzundan geri çağırarak bir öncekine göre daha

kısa zamanda sonuçlandırabilmektedir. Üniversite kurumu göz önünde bulundurulduğunda VTYS üzerindeki işlemler birçok kurumun eş zamanlı olarak kullandığı bir veri merkezindeki gibi milisaniyede milyonlarca işlem gerçekleştirilmemektedir. Dolayısıyla, kurumsal metriklerin izlenmesi amacıyla tetikleyici kullanmak elbette performansı olumsuz yönde etkilese de bu etki kullanıcıları ve diğer bilişim sistemlerini etkileyecek seviyede değildir. Tetikleyici kullanımı ile performansın olumsuz etkilenmesi konusunda gerçekleştirilmiş görünüm (Materialized View) kullanımı gibi yöntemlerle en aza indirgeyen çalışmalar bulunmaktadır.

Üniversiteler bağlı olduğu ve diğer paydaş kurumların verilerini de takip etmek durumundadırlar. Örneğin, Türkiye'deki üniversiteler Yükseköğretim Kurumu'ndan gerek akademik gerekse de eğitim-öğretim ile ilgili verileri sağlamaktadırlar. Aynı durum paydaşı olduğu ortak çalışmalarının bulunduğu diğer üniversiteler ile akademik veri paylaşımları söz konusudur. Dolayısıyla, kurumsal metrikler olarak harici veri kaynakları göz ardı edilmemelidir. Bu tür veriler VTYS'lerden sağlanan veriler kadar yapısal değildirler. VTYS'lerde kurumsal metrik verileri sorgu tabanlı olduklarından istenilen yapıda elde etmek mümkündür. Ancak, harici verilerde veri kaynağının sunduğu veri yapısı amaca yönelik olarak tekrar düzenlenmelidir. Harici veride zaman etiketi bulunmaması durumunda bu düzenleme esnasında eklenmesi gerekmektedir.

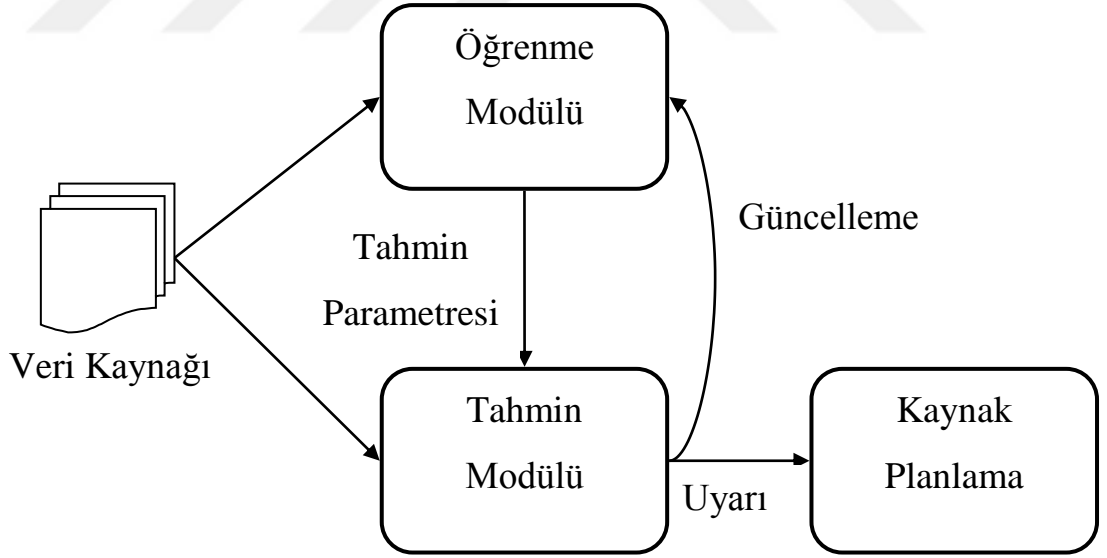
Kurum içi ve kurum dışı metrik verileri toplayabilen, analiz edebilen ve grafik olarak görselleştirip gerektiğinde uyarı olarak bilgilendiren Striim ve Confluent gibi gelişmiş yazılımlar mevcuttur. Ancak, bu yazılımlar VTYS'lerin çok sınırlı bir kısmının belirli sürümlerini desteklemektedirler. VTYS olarak birçok farklı sürümlerinin kullanıldığı üniversiteler açısından bu yazılımların kullanım amacı açısından çok uygun görülmemektedir. Bilimsel çalışmaların ve araştırma geliştirme faaliyetlerinin ön planda olduğu üniversiteler açısından bu çalışmada da olduğu gibi bir model üzerinde çalışılması daha uygun bulunmuştur. Mevcut bilişim sistemlerini etkilemesi açısından önerilen modele yönelik bir prototip yazılım üzerinde gerekli testler tamamlanarak faaliyete geçirilmesi mümkündür. Bu tezde yapılan çalışmalar bu doğrultuda gerçekleştirilmiştir.

2.3.2. Kullanım Alanları

Literatürde sistemlerden gerçek zamanlı veri akışında kullanılan modelleri içeren sınırlı sayıda da olsa değerli çalışmalar bulunmaktadır. Araştırma kapsamında önerilen modelde veri kaynağı kullanımında herhangi bir kısıt olmadığından farklı alanlarda yürütülen çalışmalarda da katkı sağlayabilir.

2.3.2.1. Sanal Makine Sunucuları

Solaimani ve diğ. (2016: 1479) çalışmalarında sanal makine sunucularından veri akışı sağlamaya yönelik bir model geliştirmişlerdir. Geliştirdikleri bu modelde; sanal sunucuların işlemci kullanımı, yük durumu, kullanım yoğunluğu vb. gibi parametrelerin gerçek zamanlı verilerini kullanılmışlardır. Bu verilerin toplanmasında Apache Kafka adı verilen yazılım kullanılmıştır. Apache Kafka kaynak ve hedef bilgisayar sistemleri arasında verilerin hatasız ve hızlı bir şekilde iletimini sağlamaktadır. Bu iletişimin sağlanmasında mesaj kuyruğu denetimini kullanan bir ara yazılımdır (Sakinmaz, 2016: 1).



Şekil 2.8. Gerçek zamanlı veri akışında anomali tespiti (Solaimani ve diğ., 2016: 1481)

Şekil 2.8’de Solaimani ve diğ. (2016: 1481) çalışmalarında kullandığı gerçek zamanlı veri akışında anomali tespiti için kullandığı model görülmektedir. Yukarıdaki şekilde görüldüğü üzere, kullanılan model dört esas üzerine kuruludur. Bunlar; veri

akışı, öğrenme modülü, tahmin modülü ve kaynak planlamasıdır. Sanal makine sunucularından gelen veri akışı veri kaynağı olarak kullanılmaktadır. Bu veriler öğrenme modülüne belirli aralıklarla, tahmin modülüne gerçek zamanlı olarak aktarılmaktadır.

Öğrenme modülünde geçmiş veriler göz önünde bulundurularak anomali tespitinde kullanılacak olan veriler üretilmektedir. Üretilen bu veriler tahmin modülüne iletilir. Tahmin modülü öğrenme modülünden gelen parametrik verileri kullanılmaktadır. Parametrik veriler ile gerçek zamanlı veri akışı karşılaştırılarak anomaliler tespit edilmektedir. Tespit edilen anomaliler kaynak planlayıcısına gönderilmektedir. Sonuç olarak, kaynak planlayıcı gelen anomali uyarısı doğrultusunda sunucu kaynaklarını tekrar düzenleyebilmektedir (Solaimani ve diğ., 2016: 1481).

Doğal olarak, tespit edilen veya edilemeyen uyarılardan bazıları gerçeği yansıtmayabilir. Anomali olarak tespit edilen bazı uyarılar gerçekte doğru olmayabilir ya da normal olarak görülen ve tespit edilemeyen anomaliler olabilmektedir. Bu durumun denetlenmesi daha gerçekçi sonuçların alınmasını sağlamaktadır. Bu denetimin özellikle öğrenme modülünün ilk zamanlarında yapılması çok önemlidir. Denetleme işi tespit edilen uyarının gerçekten bir anomali olup olmadığını işaretleyen biri tarafından denetimli olarak gerçekleştirilmektedir. İşaretlenen bu uyarılar tekrar öğrenme modülüne gönderilerek öğrenme modülünün kendini güncellemesi sağlanmaktadır. Sürekli olarak bu uyarı işaretlemesinin yapılması gerekmemektedir. Ancak, düzenli olarak belirli aralıklarla yapılması daha gerçekçi sonuçların alınmasını sağlamaktadır (Solaimani ve diğ., 2016: 1481).

2.3.2.2. Sensör Verileri

Abuaitah (2013: 4), tez araştırmasında sensör ağındaki verileri toplayan bir sistem geliştirmiştir. Bir binanın içerisine yerleştirilen ısı, nem, ışık, gürültü, duman, hareket vb. gibi çeşitli sensörlerden gelen gerçek zamanlı verileri toplayarak analiz etmiştir. Daha önce bu konuda yapılan çalışmalarda kullanılan sensörlerin özellikleri ve olumlu/olumsuz yönleri konusunda kapsamlı bir araştırma gerçekleştirilmiştir. Sensörlerden alınan veriler ağ bağlantısı yoluyla bir bilgisayarda toplanmaktadır. Elde edilen bu veriler Philip, Nelson, Matt ve David'in TOSSIM (2003: 126) adını verdiği bir yazılım ile analiz edilmektedir. Bu yazılım sayesinde sensör verilerine dayanılarak

anomali tespiti çalışması gerçekleştirilmiştir. Sensörlerden elde edilen veriler direk olarak veri kaynağından elde edilen veriler olduğundan yapılan bu çalışma veri merkezli anomali tespiti olarak adlandırılmaktadır.

Tablo 2.1’de görüldüğü üzere, Abuaitah (2013: 35) çalışmasında anormal olarak tespit edilen olayların donanım, yazılım ve ağ gibi alt yapı hatalardan kaynaklandığı gözlemlenmiştir.

Tablo 2.1. Anomalilere sebep olan olaylar (Abuaitah, 2013: 35).

Normal Hatalar	Kötü Niyetli Davranış	Uyarı
Donanım	Veriye müdahale	Sistem yük durumu
Yazılım	Servis dışı bırakma	Servis talep yoğunluğu
Ağ	Başka bir veriye yönlendirme	Sistem izleme

Elde ettiği bulgulara göre, tespit edilen anomaliler; sensörlerin belirli bir çalışma aralığına sahip olması, sensörlerden alınan verilerin taşınmasını sağlayan ağda tıkanmaların meydana gelmesi, elektrik ve kesintisiz güç kaynağı kesintisi, işletim sistemlerinden ve yazılımlardan kaynaklanan bazı hatalar olarak sıralanmaktadır (Abuaitah, 2013: 35).

Normal hatalar, meydana gelmemesi için önlem alınsa da doğal olarak karşılaşma ihtimali her zaman bulunan hatalardır. Özellikle altyapı kaynaklı tespit edilen ve normal hatalardan kaynaklanan anomaliler olarak sınıflandırılmaktadır (Abuaitah, 2013: 35).

Diğer bir anomali olarak kabul edilen sebeplerin oluşturduğu sınıf, kötü niyetli davranıştır. Veriye müdahalede bulunma, veri kaynağının servis dışı bırakılması ve veri kaynağını başka bir veri kaynağına yönlendirme bu tür hatalara yol açan sebeplerden bazılarıdır (Abuaitah, 2013: 58).

Bu sebeplerin meydana gelmesinde kişi veya personelin kasıtlı olarak kötü niyetli davranışları yer almaktadır. Bu kötü niyetli davranışların ilkinde, veri kaynağında oluşan gerçek veriye müdahale edilerek elde edilen sonuçların değiştirilmesi söz konusudur. Veriye müdahale; var olan verinin silinmesi, değiştirilmesi veya araya yeni bir verinin girilmesi suretiyle gerçekleşmektedir. Elde edilen sonuçların

görüntülenmesini engellemek amacıyla veri kaynağının belirli bir süre kasıtlı olarak devre dışı bırakılması da kötü niyetli bir davranıştır. Diğer bir kötü niyetli davranış ise başka bir veri kaynağının orijinal veri kaynağı olarak gösterilmesi şeklindedir. Böylece kullanıcılar ve uygulamalar farkında olmadan başka bir veri kaynağını kullanmış olurlar. Böylece kötü niyetli kişiler kullanıcıların gerçek verilere erişmesini engellemektedirler (Abuaitah, 2013: 58).

Son olarak, anomaliliği meydana getiren sebeplerden diğeri ise uyarılardır. Başka bir deyişle servis ya da sistemde meydana gelen değişikliklerdir. Sistemin yük durumundaki değişiklikler, kullanıcıların ve uygulamaların servis kullanım yoğunluğu ve sistemin servis sağlayıp sağlamadığının takibi bu tür anomalileri meydana getiren sebeplerden bazılarıdır (Abuaitah, 2013: 71).

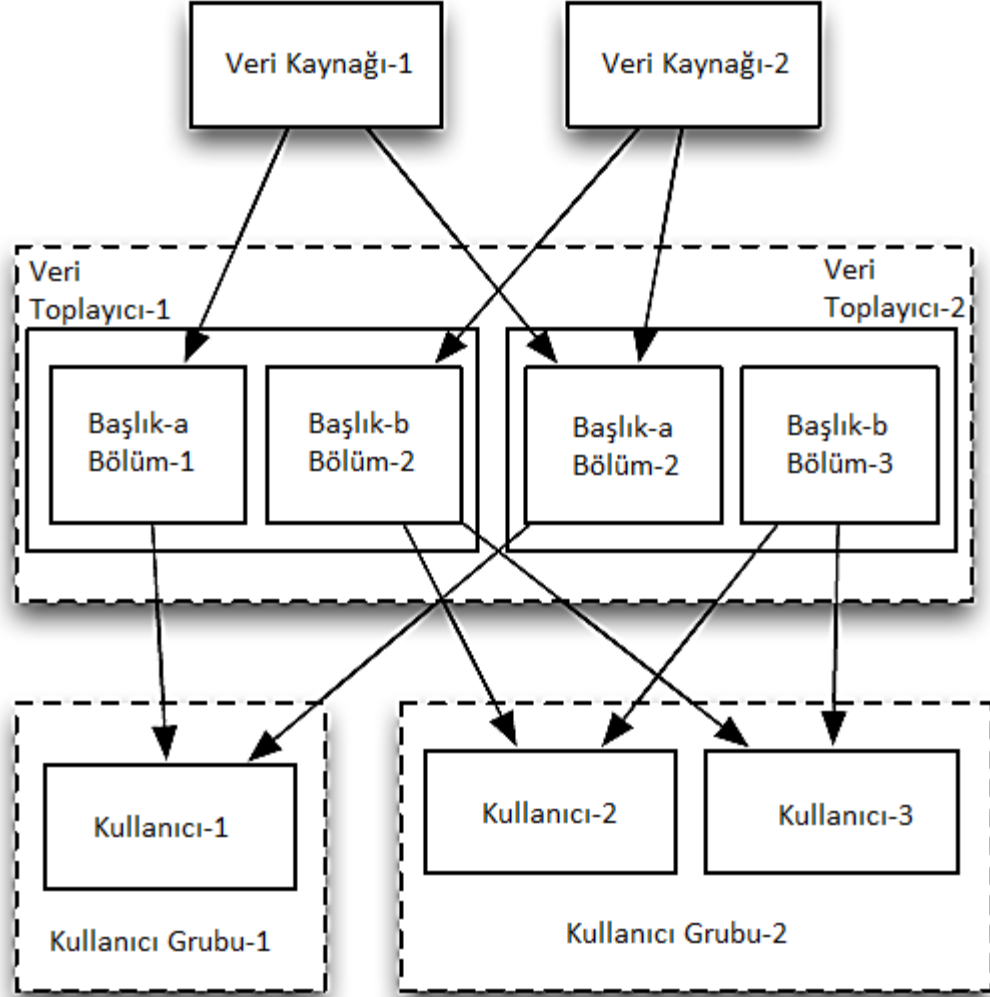
Bu sebeplerin ortaya çıkmasında herhangi bir donanım, yazılım veya kötü niyetli bir kişinin davranışı bulunmamaktadır. Sistemi kullanan kullanıcıların ve uygulamaların davranışları neticesinde meydana gelmektedir. Bu kategoride değerlendirilen anomaliler genellikle gözlem amaçlıdır. Sistemin performansının korunması, servisin devamlılığı ve servis kalitesinin iyileştirilmesine yönelik çalışmalarda kullanılacak türde anomalilerdir (Abuaitah, 2013: 71).

2.3.2.3. Sosyal Ağlar

Guozhang ve diğ., (2015: 1654) çalışmalarında LinkedIn web sayfası tarafından sağlanan verileri toplayan bir model geliştirmişlerdir. LinkedIn (2019: 1) internetteki sosyal ağlardan biridir. Günlük 300 milyonun üzerinde ziyaretçisi bulunmaktadır. Sayfa, ziyaretçilerinin günlük girişlerini ve etkinliklerini kaydetmektedir. “Profilime kim baktı?” ve “Tanıdığımız kişiler” gibi içerik önerileri sunmaktadır. Ayrıca sahte hesap tespiti gibi diğer birçok arka plan izleme ve raporlama araçlarına sahiptir. Çalışmalarında LinkedIn web sayfasından ilgili verileri toplamak için Apache Kafka yazılımı kullanılmıştır.

Fernandez ve diğ. (2015: 1) LinkedIn verilerinin gerçek zamanlı olarak farklı sistemlere dağıtılmasını sağlayan bir model geliştirmişlerdir. Fernandez ve diğ. (2015: 1)’in geliştirdikleri modelde Apache Kafka veri toplama katmanı olarak kullanılmaktadır. Diğer bir katman olan veri işleme katmanında Apache Samza

kullanılmıştır. Böylece daha gelişmiş bir model ortaya koyan bu çalışmada veri işleme ve veri toplama katmanları bir arada yer almaktadır. Geliştirdikleri bu modele Liquid adını vermişlerdir. Apache Samza (2019: 1) dağıtık verinin akışının işlenmesini sağlayan bir yapıya sahiptir. Veri toplama katmanına gelen veri akışının dinamik olarak kontrolünü sağlar. Gerçek zamanlı olarak toplanan verilerin tümleştirilmesini ve kullanıcılarına dağıtımını sağlayan mesajlaşma katmanına sahiptir.



Şekil 2.9. Gerçek zamanlı verilerin kullanıcılarına dağıtımını sağlayan mimari (Fernandez ve diğ., 2015: 4).

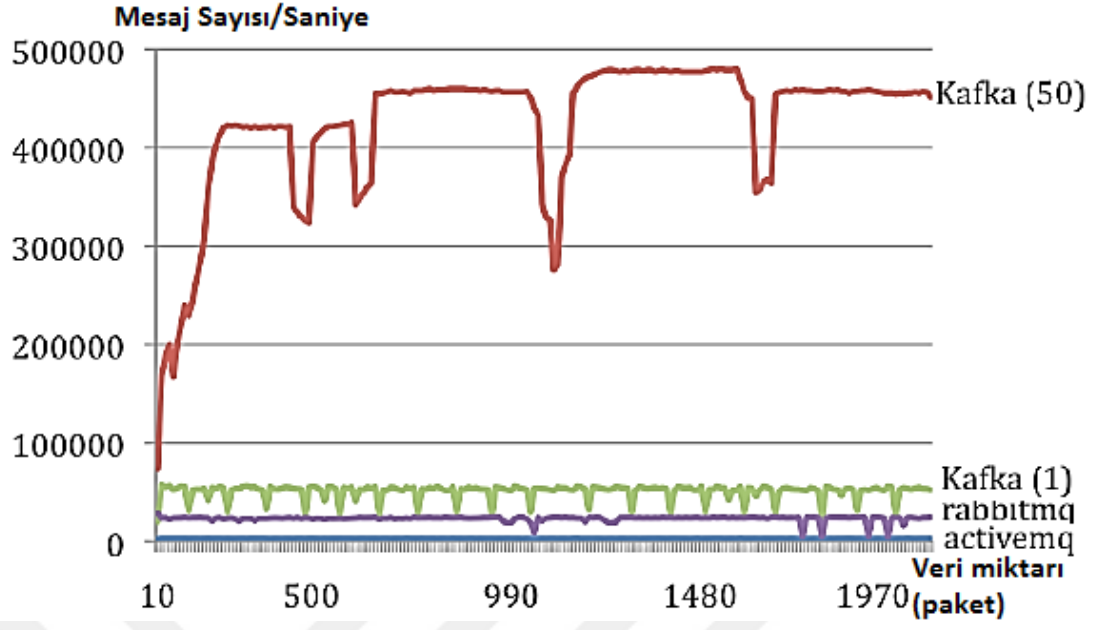
Şekil 2.9'da gösterilen mimari veri kaynağı katmanı, mesajlaşma katmanı ve kullanıcı katmanı olmak üzere üç temel katmandan oluşmaktadır.

Mesajlaşma katmanın görevi, kullanıcılarının istenilen verilere gerçek zamanlı olarak ulaşabilmesi için veri geçişini sağlamaktır. Kullanıcıların ilgilendikleri veri veya

veri grupları deęişkenlik gösterdiğinden bu katman verileri başlıklar halinde hazırda tutmaktadır. Bu başlıklar sayesinde kullanıcı istedięi başlığa erişme imkânına sahip olmaktadır. Mesajlaşma katmanının veri geçişini sağlamasının yanı sıra veri kaynaklarını tümleştirme özellięi de bulunmaktadır. Bu sayede aynı veri kaynağına sahip farklı başlıklar için aynı verilerin tekrar toplanmasına gerek kalmaz. Veri kaynağından toplanan veriler mesajlaşma katmanında genel kullanıma yönelik olarak yapılandırılır. Yeni bir başlık oluşturulmak istenildiğinde mesajlaşma katmanında bulunan yapılandırılmış veriye filtreleme uygulanır. Böylece aynı veri farklı parametrelerde farklı başlıklar altında rahatlıkla çoęaltılabilmektedir (Fernandez ve dię., 2015: 4).

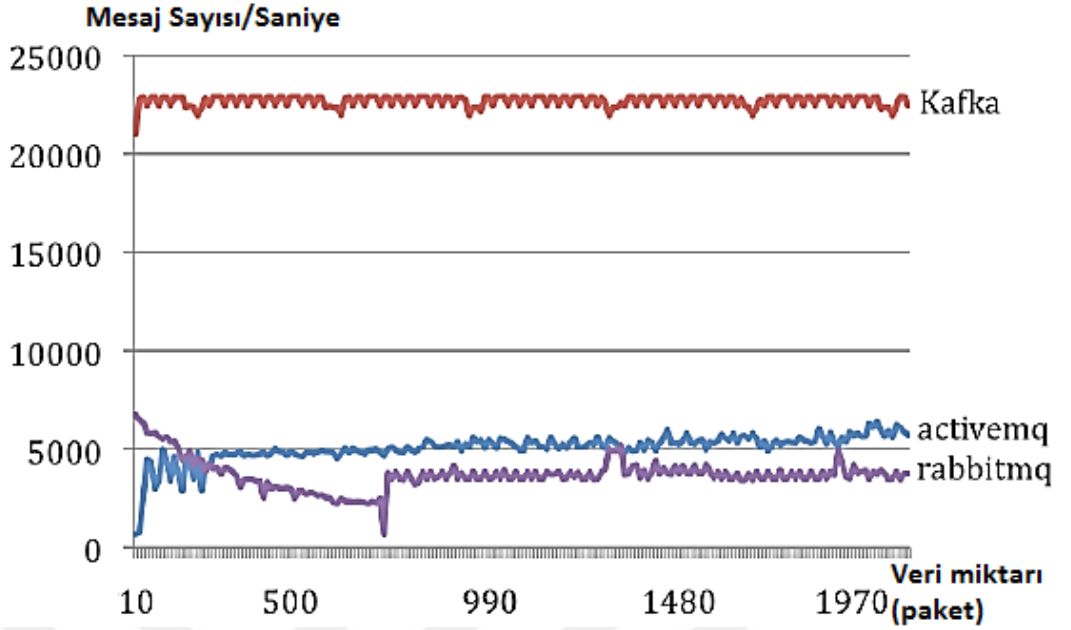
2.3.2.4. Performans Karşılaştırma

Kreps, Narkhede ve Rao (2011: 1) gerçek zamanlı veri toplama araçlarının performans karşılaştırmasını gerçekleştirmişlerdir. Performans karşılaştırması için en çok kullanılan veri toplama araçları olan Apache Kafka, Java ActiveMQ v5.4 ve RabbitMQ v2.4 kullanılmıştır. Performans testi iki adet Linux işletim sisteminin kurulu olduęu bilgisayar üzerinde gerçekleştirilmiştir. Bilgisayarlardan biri veri kaynağı dięeri ise verileri toplayan hedef bilgisayar olarak kullanılmıştır. Yapılan testte 10 milyon adet mesaj içeren verinin transfer hızları ölçülmüştür.



Şekil 2.10. Gerçek zamanlı veri toplama performans karşılaştırması (Kreps Narkhede ve Rao, 2011: 6).

Şekil 2.10'da gerçek zamanlı olarak verilerin toplanmasında kullanılan araçların performans karşılaştırılması verilmiştir. Şekilde verilen grafiğe göre Kafka veri toplama işleminde diğer araçlara oranla çok daha performans göstermiştir.



Şekil 2.11. Gerçek zamanlı veri dağıtım performans karşılaştırması (Kreps Narkhede ve Rao, 2011: 6).

Şekil 2.11’de gerçek zamanlı olarak verilerin dağıtılmasında, başka bir deyişle kullanıcılarına ulaştırılmasında kullanılan araçların performans karşılaştırılması verilmiştir. Şekilde verilen grafiğe göre Kafka verileri dağıtım hızında da diğer araçlara kıyasla çok daha iyi performans göstermiştir.

Genel olarak, elde edilen sonuçlara göre (Bkz. Şekil 2.10 ve Bkz. Şekil 2.11) Apache Kafka gerçek zamanlı verilerin toplanmasında ve dağıtılmasında diğer araçlara oranla daha kararlı bir şekilde yaklaşık 5 kat daha hızlı çalıştığı görülmektedir.

2.3.2.5. Diğer Kullanım Alanları

Anomali tespiti günümüzde birçok alanda kullanılmaktadır. Bankacılık, finans, bilişim sistemleri, güvenlik bu alanların başında sayılabilir (Akoğlu, Tong ve Koutra, 2015: 8). Diğer kullanım alanlarından bazıları aşağıdaki gibi sıralanabilir (Cheboli, 2010: 5):

Elektrokardiyogram (EKG) verileri ile anormal kalp atışlarının tespiti: EKG’den alınan veriler periyodik bir zaman serisi olarak görüntülenebilmektedir. Burada tespit edilen bir anomali sağlık sorununa işaret edebilmektedir.

Öneri sistemlerinde saldırı tespiti: Saldırganlar öneri sistemlerinden elde edilen sonuçları değiştirmek için kasıtlı olarak puanlama girişleri yapabilmektedirler.

Uçaklarda uçuş güvenliğinin sağlanmasında: Uçaklarda kullanılan sistemlerin tepkileri uçuş sırasında değişen farklı parametrelerin sensör veri bilgileriyle karakterize edilir. Standart sistem tepkilerindeki sapmalar anomali olarak tespit edilmektedir.

Şekilsel anomaliler: Belirli bir zaman sonra diğerlerinden farklılaşan şekillerin tespit edilmesinde kullanılmaktadır. Örneğin, tıp alanında bir türün içerisinde anormal şekil farklılıkları genetik mutasyonları ortaya çıkarmada kullanılmaktadır.

Bitki örtüsü veya sıcaklık gibi yer bilimi verileri kullanılarak eko-sistem bozukluklarının tespit edilmesi çalışmalarında kullanılmaktadır.

Sistemlerin erişim durumu hakkında bilgi elde edilmesi: Uygulamalar veri transferi için bazı veri kaynaklarını ve veri depolama sistemlerini kullanmaktadırlar. Ancak, veri kaynaklarına erişimi zaman zaman kesintiye uğramakta ve herhangi bir veri transferi gerçekleştirilememektedir (Abuaitah, 2013: 69). Erişiminin mümkün olmadığı durumlarda sistemler veri kaynaklarına bağlanamayacak ve görevlerini yerine getiremeyecektir. Veri kaynağındaki teknik bir arıza, bakım, gibi sebepler erişilememe durumuna yol açabilir. Bu tür durumun tespiti için veri kaynağına en son ne zaman erişildiği ve ne kadar süre erişilemediği bilgisi analiz edilebilir. Bu durumun daha detaylı analizi sayesinde veri sağlayan sistemlerin erişilebilirliği hakkında fikir sahibi olmak mümkündür.

ÜÇÜNCÜ BÖLÜM

TÜRKİYE'DEKİ ÜNİVERSİTE YÖNETİCİLERİNE YÖNELİK BİR UYGULAMA MODELİ ÖNERİSİ: KURUMSAL METRİK İZLEME SİSTEMİ (KMİS)

Sistem analizi ve tasarımı çalışmalarında geliştirilen her model sistemin mantıksal çerçevesini belirlemektedir. Önerilen model kapsamında çalışmaya özgü geliştirilen uygulama ise fiziksel tasarımın gerçek hayattaki ilk adımlarından biridir. Bu sürecin her aşamasında farklı araç, teknik ve yöntemlerin belirli bir düzen içerisinde planlanması ile çalışmanın metodolojisini ortaya çıkmaktadır. Bu kapsamda bu bölümde araştırma metodolojisinde önerilen model ile geliştirilen uygulama birlikte ele alınmıştır.

3.1. METODOLOJİ

3.1.1. Modelin Amacı ve Önemi

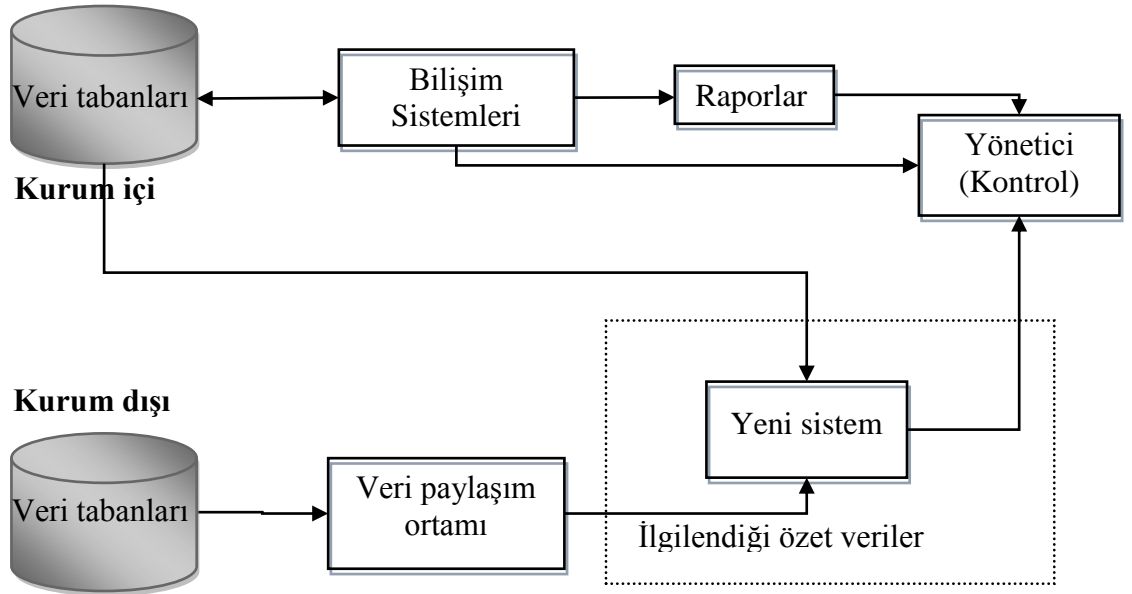
Bu modelin amacı, üniversitelerde özet veri niteliğindeki metrikler üzerinden anomali tespiti ve uyarı ile yöneticilere yönelik bilgi sunmaktır. Modelin nicel olarak incelenebilmesi amacıyla Kurumsal Metrik İzleme Sistemi (KMİS) adı verilen prototip bir uygulama geliştirilmiştir. Geliştirilen prototipleme VTYS ve harici bağlantılar üzerinden kurum ile ilgili çeşitli özet verileri toplayabilme özelliğine sahip olmasından dolayı bu isim verilmiştir. Özkan Y. (2013: 5)'nin da belirttiği üzere veri tabanları kurumların veri deposu görevini gördüğünden üniversite içi metriklerin izlenmesi için çeşitli VTYS'ler ile bağlantı kurulması üzerinde çalışılmıştır. Han, Kamber ve Pei (2012: 27), Purohit ve diğ. (2012: 461), Pazarçeviren, Zor ve Gürbüz (2015: 76) de vurguladığı üzere kurumsal metriklerin izlenmesinde kurum dış çevresinde meydana gelen metrik verilerin de kullanılması gerekmektedir. Üniversite dış çevresindeki metrikler için paydaş kurumların izin verdiği verileri toplayarak kurumun kendi veri tabanına kayıt edilmelidir. KMİS verinin üniversite içi veya üniversite dışı veri olmasına göre gerekli parametreler girildikten sonra verileri gerçek zamanlı ya da veri dosyasından toplayabilme ve saklama özelliğine sahiptir. Yavuz (2014: 18)'in de belirttiği üzere yönetim düzeyi arttıkça yöneticiler daha az ama kendileri için önemli ve

özet bilgilere ihtiyaç duyarlar. Bu doğrultuda anomali tespiti yönetim düzeyleri için üç seviyede sınıflandırılmıştır. Ayrıca model istenilen seviyede uyarıda bulunabilecek şekilde tasarlanmıştır.

Model, üniversite yöneticileri göz önünde bulundurularak yöneticilerin kontrol fonksiyonuna yönelik olarak geliştirilmiştir. Model üniversite yöneticileri tarafından karar verme sürecinde kullanılabilecek yardımcı bilgi sunabilmeyi hedeflemektedir. Böylece, problemler daha başlangıçta büyük sorunlar oluşturmaya başlamadan önce yöneticilerin daha etkin çözüm geliştirmelerine ve tedbir almalarına yardımcı olunabilir. Ayrıca gerçek zamanlı veri akışı analizi ilgili literatürün oluşumuna katkı sağlaması ve modelde kullanılan VTYS'lerin genel olarak farklı üniversitelerde de kullanılması nedeniyle diğer üniversiteler ve kurumlarca uygulanabilir bir model teşkil etmesi düşünülmüştür.

3.1.2. Uygulama Modeli ve Hipotezleri

Kis (2016: 15)'de belirttiği üzere prototipler yeni fikir veya ürün için geliştirilen teori ve hipotezlerin kanıtlanmasını sağlayan teknik özelliklere sahiptir. Bu doğrultuda model geliştirmede süreci olan Hızlı Uygulama Geliştirme (HUG) süreci kullanılmıştır. Bilişim sistemleri ve yöneticiler açısından modelin veri akış şeması aşağıdaki şekilde gösterilmektedir.

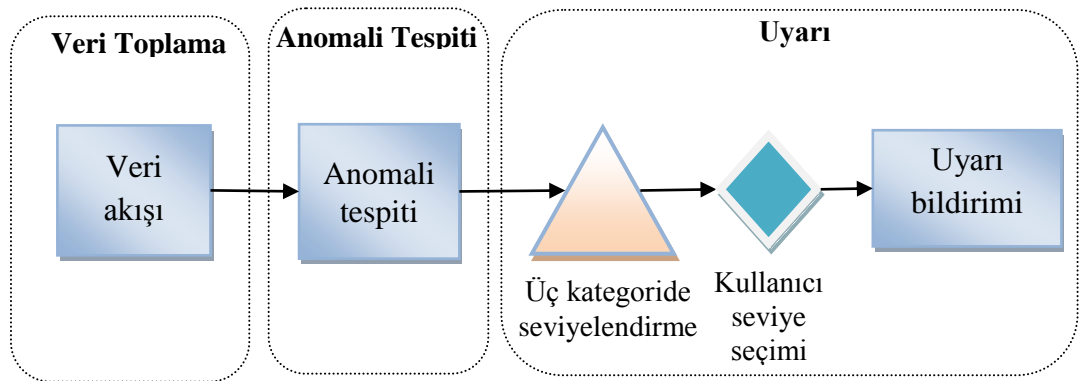


Şekil 3.1. Yöneticilerin bilişim sistemleri aracılığıyla üniversiteyi izlemesi.

Şekil 3.1’de görüldüğü üzere, yöneticiler çeşitli bilişim sistemlerinin sunduğu raporlar sayesinde üniversite verilerini izleme imkânına sahiptir. Ancak, modelin en önemli özellikleri olan gerçek zamanlı veri akışı ve veri dosyasını okuyabilmesi sayesinde yönetici daha erken bilgiye ulaşma imkânına sahip olabilir. Yönetici prototip uygulamadan aldığı bilgiyi raporlarla gelen bilgi ile karşılaştırıp teyit de edebilir. Ayrıca bir süre sonra prototip uygulamadan elde ettiği tecrübe sayesinde bazı kritik durumlarda raporlarla gelen bilgiyi beklemeden gerekli önlemleri alma yoluna gidebilir. Bu doğrultuda araştırmanın ana problemi “Üniversitede kullanılan bilişim sistemlerinden ve kurum dışı veri kaynaklarından gerçek zamanlı veri akışıyla ya da veri dosyasıyla önem derecesine göre uyarı veren bir sistem geliştirilebilir mi?” sorusudur.

Modelin altyapısını oluşturan araçlar seçilip bütünleştirme aşaması tamamlandıktan sonra kullanıma yönelik bir prototip uygulama oluşturulmuştur. Önerilen modelin en önemli çıktılarından biri anomali tespittir. Anomali tespiti için Google Analytics’a benzer bir tahmin aralığı kullanılmaktadır. Tahmin aralığı ile anomalilerin tespitinde kullanılan destek vektör makinesine dayalı öğrenme modeli kullanılmaktadır. Sınıflandırma modeli ile tespit edilen anomaliler önem derecesine göre üç kategoride seviyelendirilmiştir. Uygulamadan elde edilen sonuçlar grafiksel olarak gösterilmektedir. Böylece gerçek zamanlı verilerin görsel olarak daha iyi ve farklı amaçlara göre analiz edilebilmesi amaçlanmıştır.

Şekil 3.2’de sunulan uygulama modelinin temel süreçleri olan veri toplama, anomali tespiti ve uyarı aşamaları gösterilmektedir.

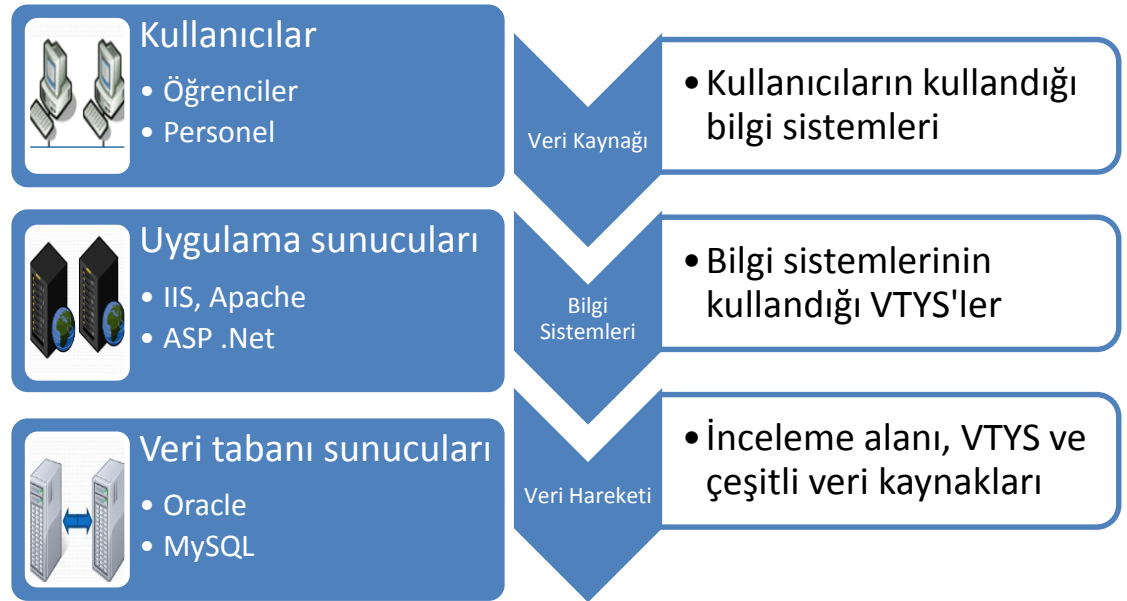


Şekil 3.2. Önerilen modelin temel süreçleri.

Önerilen modelin ilk süreci olan veri toplama farklı veri kaynaklarından mümkün olduğunca gerçek zamanlı veri akışı sağlamayı hedeflemektedir. Toplanan veriler ikinci süreç olan anomali tespiti ile analiz edilmektedir. Bu süreç içerisinde veriler özellikle zaman serisi olarak analiz edilebilecek şekilde yapılandırılmaktadır. Destek vektör makinesi algoritması yardımıyla verilerin beklenti aralığında olup olmadığı kontrol edilmektedir. Beklenti dışında kalan veriler zaman etiketi ve mesafesi ile birlikte kaydedilmektedir. Uyarı süreci ise elde edilen anomali verileriyle işlevini yerine getirmektedir. Öncelikli olarak veriler mesafesine göre üç kategoride seviyelendirilir. Kullanıcı ihtiyacına göre kategorilerden istediğini tercih ederek ilgili seviyede uyarı mesajının içeriği oluşturulmaktadır.

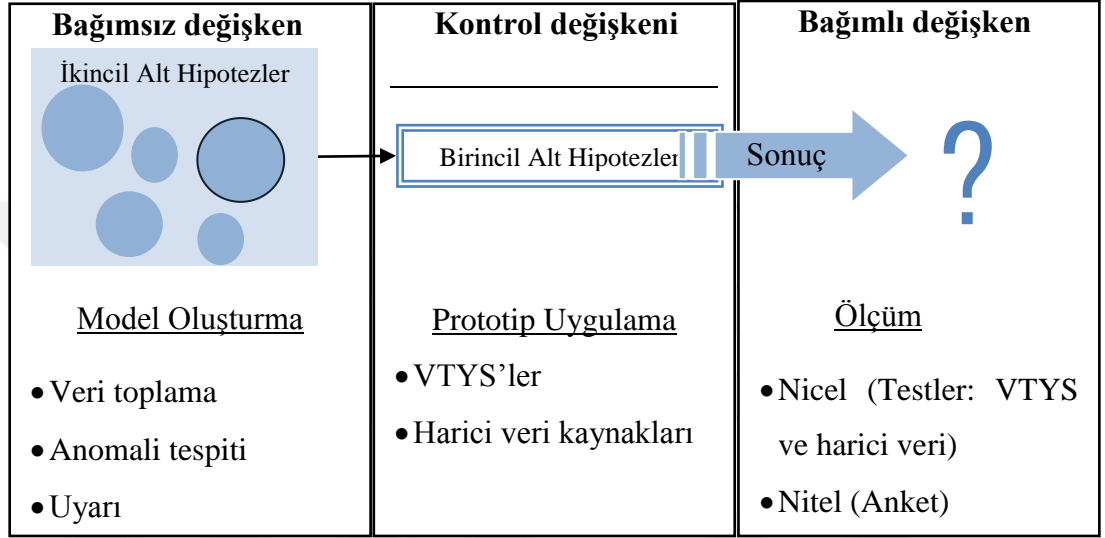
3.1.2.1. Alt Hipotezler

Bilindiği üzere, günümüzde elektronik ortamda gerçekleştirilen birçok işlem bilişim sistemleri aracılığıyla yapılmaktadır. Bilişim sistemlerinde kullanılan veriler veri tabanlarından saklanmaktadır (Ömürbek, 2003: 93). Dolayısıyla, araştırmada öncelikli olarak farklı VTYS'ler ve veri kaynakları ile kolaylıkla bütünleştirilebilecek ve veri toplama işlemini gerçekleştirebilecek bir alt yapının oluşturulması bulunmaktadır (Şekil 3.3).



Şekil 3.3. VTYS'lerin model açısından önemi.

Alt hipotezler önem derecesine göre birincil ve ikincil seviyede sınırlandırılmıştır. Birincil seviyedeki alt hipotezler araştırmada bağımsız değişken olarak model tasarlanmasında kullanılmıştır. İkincil seviyedeki alt hipotezler ise prototip uygulamanın geliştirilmesine yardımcı kontrol değişkenlerini içermektedir. Prototip uygulama sayesinde önerilen model nicel ve nitel olarak değerlendirilmektedir (Şekil 3.4).



Şekil 3.4. Araştırma değişkenlerinin model ile ilişkisi.

Birincil alt hipotezler; tasarlanan sistemin istenilen şekilde geliştirilebilmesini sağlayan problemlerin çözümünü içerir. Önerilen model doğrultusunda geliştirilen uygulamanın başarılı olması için birincil alt hipotezlerin çözümü tam olarak ortaya konması gerekmektedir.

İkincil alt hipotezler; model tasarlandıktan sonra örnek kullanım için gerekli olan prototip uygulamanın geliştirilmesini sağlamaktadır. İkincil alt hipotezler, veri kaynakları ile bire bir ilişkilidir. Sunulan modelin başarısı ya da bir sonuç üretebilmesi farklı veri kaynaklarını kullanılabilmesini gerektirir.

3.1.3. Modelin Evreni ve Örneklemi

Modelin evreni Türkiye'deki üniversitelerdir. Modelin prototip uygulaması olan KMİS yazılımında açık kaynak kod kullanılması nedeniyle daha ileri seviyede bir geliştirmeye açıktır. KMİS kurumların ve üniversitelerin kendi sistemlerine göre

özelleştirilmesine izin verecek şekilde tasarlanmıştır. Dolayısıyla, tasarlanan modelin evreni farklı kurum ve organizasyonları kapsayacak şekilde genişleme potansiyeline sahiptir.

Modelin nicel ve nitel olarak değerlendirilmesi amacıyla iki adet örneklem kullanılmıştır. Nicel örneklem olarak da önde gelen Oracle, MySQL, Microsoft SQL Server, PostgreSQL, harici veri olarak Yükseköğretim Kurumu 2020 Temmuz ve Kasım ayları arasındaki günlük toplam akademisyen sayısı ve 2019 Eylül ile 2020 Mayıs ayları arasında Google Trendler “öğrenci bilgi sistemi” teriminin Türkiye’de gerçekleştirilen web arama yüzdelik oranı kullanılmıştır. KMİS farklı VTYS’leri ve harici veri kaynaklarını kullanabilecek şekilde tasarlandığından nicel örneklem kümesi çeşitli çalışmalarla genişletilebilir. Nitel örneklemde ise Türkiye’deki üniversitelerde çeşitli yönetim pozisyonlarında görev almış 32 adet emekli akademisyen bulunmaktadır. Böylelikle araştırmanın üniversitelerin geneline hitap etmesi amaçlanmıştır.

3.1.4. Veri Toplama Yöntemi ve Aracı

Veri toplama yöntemi kurumsal çevre göz önünde bulundurularak düzenlenmiştir. Kurum içi veri toplama VTYS’lerde tetikleyici ekleme ve sorgulama yöntemleri ile gerçekleştirilmektedir. Tetikleyici ekleme yönteminde veriler ilgili VTYS’de oluşturulan ve Tablo 3.1’de gösterilen “VERİ” tablosuna kaydedilmektedir. Bu yöntem veri toplama işleminin kesintiye uğramaması için avantaj sağlarken, ilgili VTYS’ye ek yük getirme durumu söz konusudur. Diğer bir yöntem olan sorgulamada ise VTYS tabloları direk sorgulanmakta ve KMİS kendi “VERİ” tablosuna kaydetmektedir. Bu yöntemin dezavantajı VTYS’ye erişilememe durumunda veri toplama işleminin kesintiye uğramasıdır. Sorgulama yönteminin VTYS’ye daha az yük getirmesi amacıyla sorgulama sıklığı ek bir seçenek olarak kullanılmaktadır.

Kurum dışı veriler ise dinamik veya sabit veri dosyasından veri aktarımıyla toplanmaktadır. Dinamik veri dosyası olarak json formatı kullanılmış olup, farklı veri formatları da tanımlanabilir. Sabit veri dosyası olarak excel formatı kullanılmaktadır.

Veri toplama için, KMİS’de tanımlı olan tablolar ve tetikleyici kullanılması durumunda ilgili VTYS’de tanımlanacak olan “VERİ” tablosunun kullanım amaçları Tablo 3.1’de verilmektedir.

Tablo 3.1. Sistemde kullanılan tabloların isimleri ve kullanım amaçları.

Tablo Adı	Kullanım Amacı
KULLANICILAR	Sistem kullanıcılarını tanımlayan bilgileri saklar
BAGLANTI	Veri tabanı ve dış veri bağlantı ayarlarını saklamak
ERISIM	Veri tabanı tablolarına erişim ayarlarını saklamak
ISLEM	Veri toplama işlemi ayarlarını tutar
TALEPLER	Kullanıcıların yeni veri taleplerini saklar
ANOMALI	Grafikte tespit edilen anomalilerin katılarını tutar
VERI	Üniversite dışından sağlanan verilerin saklanmasını sağlar

Veri toplama için kullanılacak olan sistem ayarları yukarıda belirtilen tablolarda saklanmaktadır. Bu ayarlar erişim, kaydetme, yapılandırma ve analiz gibi veri işlemleri için kullanılmaktadır.

KULLANICILAR tablosu kullanıcı bilgilerini saklayan alanlara sahiptir. Kullanıcıların adı, soyadı, kullanıcı adı, kullanıcı şifresi, görevi, yönetim düzeyi ve elektronik posta adresi gibi temel kullanıcı verilerini saklamaktadır.

BAGLANTI tablosu farklı veri tabanlarına ve üniversite dışı veri kaynaklarına ait adres bilgileri, bağlantı komutları, mevcut veri tabanlarını, tablolarını ve alanlarını listeleme sorguları, veri çağırma döngüleri, verilerin zamana dayalı sınıflandırma ve veri getirme komutları, vb. gibi veri tabanlarına ve dış veri kaynaklara erişimi sağlayan bilgileri saklamaktadır.

ERISIM tablosu veri tabanlarında ve harici veri kaynaklarında bulunan tablolara erişimi sağlayan değerleri kaydetmektedir. Hangi tablonun hangi veri tabanında bulunduğu bu tablo üzerinde belirtilmektedir.

ISLEM tablosunda veri toplama işlemi gerçekleştirecek olan SQL sorgusu kaydedilmektedir. Verilerin hangi tablodan nasıl toplanacağı bilgisi bu tabloda yer almaktadır.

TALEPLER tablosu kullanıcıların yeni bir veri akışı için talep bilgisini kaydetmesi için kullanılmaktadır. Kullanıcılar ve sistem yöneticisi arasında iletişimi

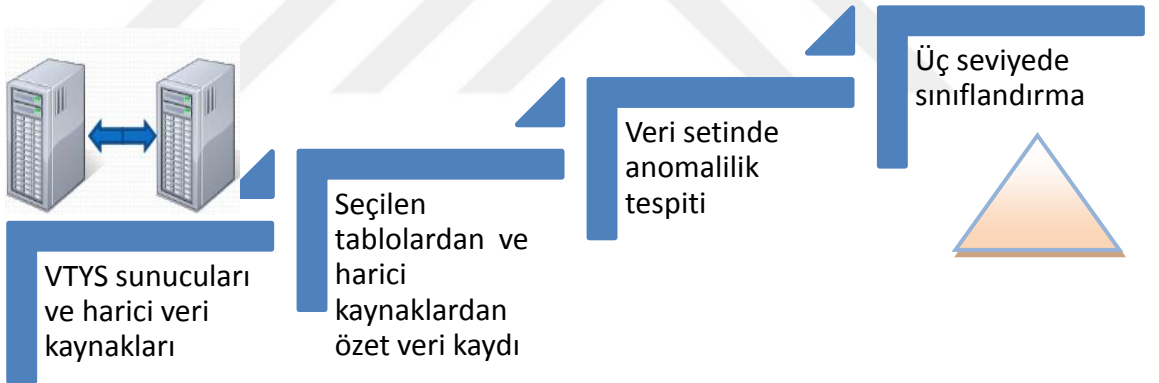
sağlayan bir tablodur. Kullanıcılardan gelen talep ve açıklamalar doğrultusunda sistem yöneticisi bu tablodaki bilgileri okuyarak yeni bir veri toplama işlemi başlatabilir.

ANOMALI tablosu tespit edilen anomalilerin zaman ve işlem adı gibi diğer teknik bilgilerini kaydeder.

VERI tablosu, üniversite dışı veri kaynaklarından sağlanan gerçek zamanlı veri akışının kaydedilmesini sağlar. Üniversite dışında oluşan veriler belirli bir süre veri kaynağında ortaya çıkmakta ve veri kaynağına göre belirli sıklıkta güncellenmektedir. Dolayısıyla, bu verilerin analiz edilebilmesi için yeni veri oluştuğu anda sisteme kaydedilmelidir. Bu görevi VERI tablosu yerine getirmektedir. VERI tablosunda verinin kaynağının adını, oluşma tarihini ve değerini kaydeden üç adet alan bulunmaktadır.

3.1.4.1. Veri Araçlarının Seçimi

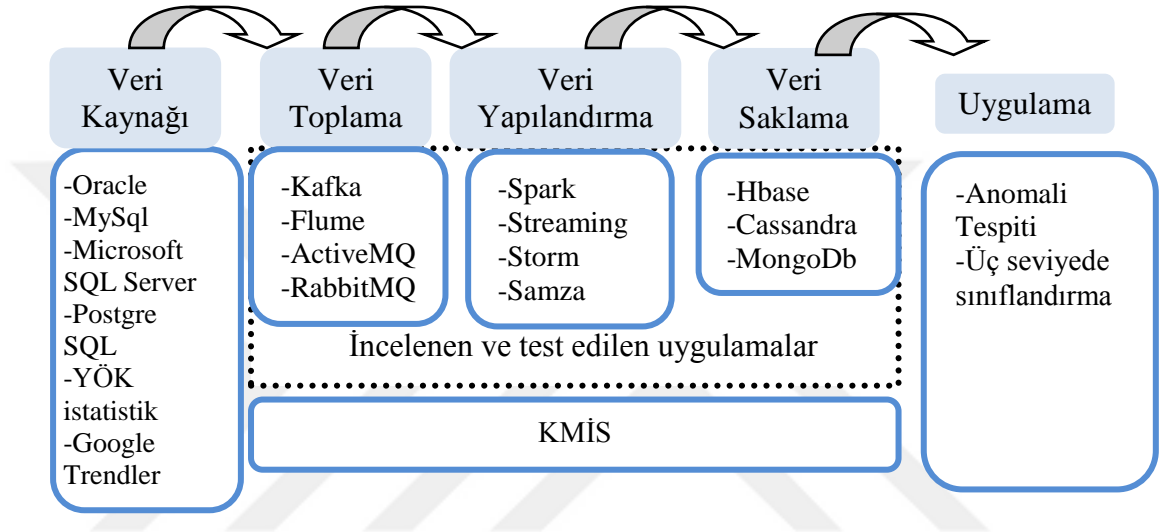
Veri araçlarının belirlenmesinde Şekil 3.5’de gösterildiği üzere önerilen modelin veri işleme süreçleri dikkate alınmıştır.



Şekil 3.5. Önerilen modelin veri işleme süreci.

Şekil 3.5’de önerilen modelin veri işleme süreci dört aşamalı olarak gösterilmektedir. İlk aşamada modele girdi sağlaması için çeşitli veri kaynaklarına bağlanabilme ve veri toplamaya hazır halde bulunması amaçlanmıştır. Bu kapsamda Şekil 3.6’da bu aşama veri kaynağı olarak seçilen VTYS’ler, Yükseköğretim Kurumu istatistik sayfası, Google Trendler kullanılmıştır. İkinci aşamada seçilen tablolardan ve harici veri kaynaklarından özet veri kaydının gerçekleştirilmesi yer almaktadır. Bu

aşamada kullanılmak üzere Şekil 3.6’da gösterilen veri toplama, yapılandırma ve saklama araçları incelenmiş ve test edilmiştir. Ancak, bu araçların önerilen model kapsamında bir arada kullanılmasında çeşitli güçlükler bulunduğundan dolayı KMİS geliştirilmiştir. KMİS önerilen model doğrultusunda ikinci aşamayı tek bir uygulamada gerçekleştirebilmektedir. Üçüncü ve dördüncü aşamalar olan veri setinde anomali tespiti ve üç seviyede sınıflandırma Şekil 3.6’da gösterildiği üzere modelin uygulama çıktılarıdır.



Şekil 3.6. Veri işleme süreci için incelenen ve kullanılan araçlar.

Ayrıca, KMİS’in ön araştırmada kullanılması planlanan araçlar arasındaki yeri Şekil 3.6’da gösterilmiştir. Buna göre veri toplama aracı olarak benzer çalışmalarda sıklıkla kullanılmasından dolayı Apache Kafka kullanılması düşünülmüştür. Ancak, araştırmada sadece veri toplama aracı değil veri analiz aracının da önem arz ettiği görülmüştür. Bu nedenle, farklı kaynaklardan veri toplama özelliğinin yanı sıra toplanan verinin analiz edilmesine de imkân tanıyan Striim ve Confluent gibi Apache Kafka ile bütünleştirilmiş sistemler üzerinde çalışılmıştır. Araştırmada bu iki araçtan birinin tercih edilerek çalışmaların yürütülmesi planlanmıştır. Ancak, daha sonra tasarlanan modelin amacına ulaştıramadığı gözlemlenerek KMİS geliştirilmiştir.

3.1.4.2. Kullanıcı Bilgi İhtiyacının Belirlenmesi

Sistem kullanıcıları olan yöneticilerin ihtiyaç duyduğu bilgi ihtiyacının belirlenmesinde aşağıdaki Şekil 3.7’de gösterilen esaslar göz önünde bulundurulmuştur.

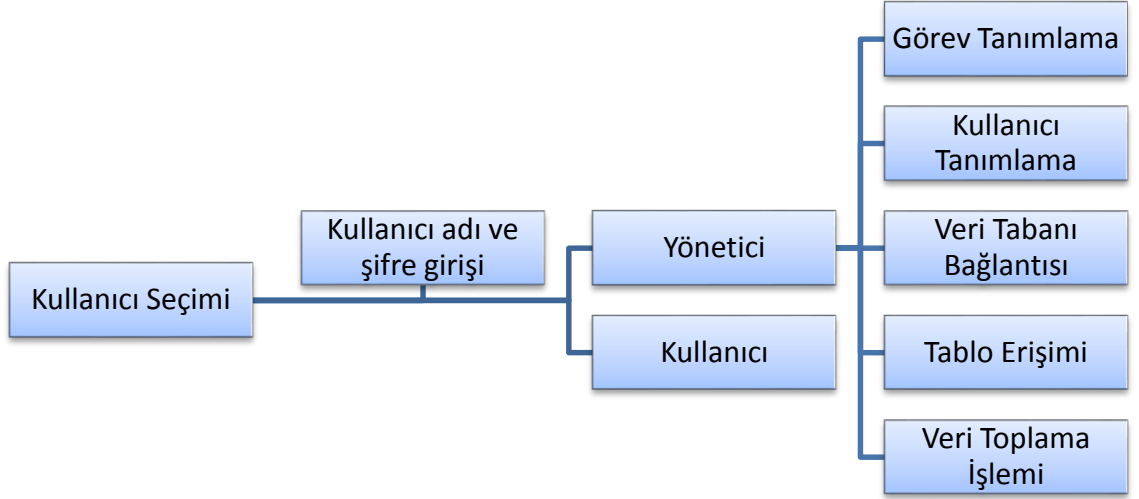


Şekil 3.7. Kullanıcı bilgi ihtiyacının belirlenmesi.

Önerilen modelde kullanıcı bilgi ihtiyacının belirlenmesinde kullanılan iki esas Şekil 3.7’de gösterilmektedir. Buna göre ilki ve en önemlisi kullanıcı talebini dikkate alınmasıdır. Diğeri ise literatür taraması yoluyla üniversite düzeylerine göre genel bilgi ihtiyacı tespit edilip kullanıcıların arasından tercih yapabilecek şekilde hazır olarak modelde sunulmasıdır. Literatür taraması farklı kaynaklarla da desteklenmekle birlikte esas olarak 2547 sayılı Yükseköğretim Kanunu dikkate alınmıştır.

3.1.4.3. Veri Toplama Arayüzü

Modelde iki kullanıcı tipi için farklı ara yüzlere sahip bir kullanıcı paneli tasarlanmıştır. Sistem kullanıcılarına yönelik olarak “Yönetici Paneli” ve normal kullanıcılar için “Kullanıcı Paneli” arayüzü bulunmaktadır. Yönetici panelinde sistem ile ilgili ayarların yapılandırılması için servis modülleri bulunmaktadır. Kullanıcı panelinde çok fazla teknik bilgi gerektirmeyen grafik ağırlıklı modüller bulunmaktadır. Uygulamaya erişim için kaynak dosyaların bulunduğu dizindeki “index.php” dosyasının web tarayıcısında çalıştırılması yeterlidir.



Şekil 3.8. Yönetici Paneline erişim ve Yönetici Panelinde kullanılan modüller.

Şekil 3.8’de Yönetici Panelinde bulunan modüllere erişimi sağlayan kullanım diyagramı gösterilmektedir. Bu modüllere erişimin sağlanması için kullanıcı tipi olarak “Yönetici” seçilir ve şifre girilir.



Kurumsal Metrik İzleme Sistemi (KMIS)

Lütfen, kullanıcı tipini seçiniz.

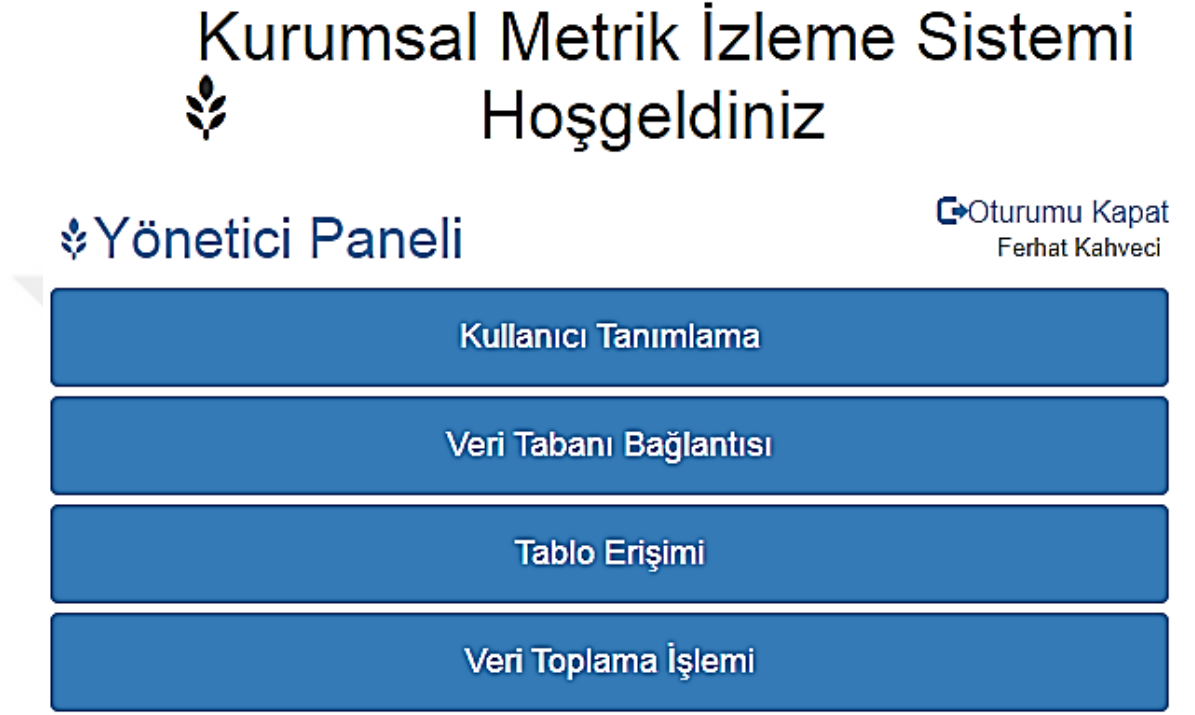
Yönetici

Kullanıcı

Şekil 3.9. Kullanıcı tipi seçimi sayfası.

Şekil 3.9’da görüldüğü üzere, sistemde kullanıcı seçimi için kullanılan sayfa verilmektedir. Yönetici Paneline erişim için “Yönetici” tuşuna basılır. Ardından yönetici şifresi girilir. Sistem yöneticisi olmayan diğer tüm kullanıcılar için “Kullanıcı” seçeneği kullanılmaktadır.

Kullanıcı tiplerinden ilki olan ‘‘Yönetici’’ sistem yöneticisidir. Kullanıcı ve görev tanımlama, VTYS ve dış kaynak veri ayarlarını yapma, tablo erişim ayarlarını düzenleme, veri toplama işlemi için SQL sorgusu yazma ve anlık veri kaydını başlatma görevleri sistem yöneticisine aittir (Şekil 3.10). Bu görevler VTYS hakkında temel bilgi ve deneyim gerektirmektedir.



Şekil 3.10. Yönetici Paneli ana sayfası.

3.1.4.3.1. Kullanıcı Tanımlama

Kullanıcı tanımlama modülünde sistem kullanıcıların bilgileri bulunmaktadır (Şekil 3.11). Kullanıcıların adı, soyadı, görevi, üniversitedeki yönetim düzeyi, elektronik posta adresi, kullanıcı adı ve şifre bilgileri tanımlanabilmektedir. Sistem yöneticisi de ayrıca burada kendi bilgilerini değiştirebilmektedir.

Kullanıcı Tanımlama					Anasayfa	Oturumu Kapat
Adı	Görevi	Kullanıcı Adı	Kullanıcı Şifresi	E-posta		
<input type="button" value="Ekle"/>	<input type="text"/>	<input type="text" value="Görev seçiniz"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Mevcut Kullanıcılar						
Adı	Görevi	Kullanıcı Adı	Kullanıcı Şifresi	E-posta		
<input type="button" value="Sil"/>	<input type="text" value="Ahmet Berk"/>	<input type="text" value="Dekan"/>	<input type="text" value="ahmet"/>	<input type="text" value="....."/>	<input type="text" value="a.berk@hotmail.com"/>	<input type="text"/>
<input type="button" value="Sil"/>	<input type="text" value="Cihan Koşar"/>	<input type="text" value="Bölüm Başkanı"/>	<input type="text" value="cihan"/>	<input type="text" value="....."/>	<input type="text" value="ckoser@gmail.com"/>	<input type="text"/>
	<input type="text" value="Ferhat Kahveci"/>	<input type="text" value="Sistem Kullanıcısı"/>	<input type="text" value="sistem"/>	<input type="text" value="....."/>	<input type="text" value="ferhatkahveci@hotma"/>	<input type="text"/>
<input type="button" value="Kaydet"/>						

Şekil 3.11. Kullanıcı tanımlama modülü.

3.1.4.3.2. VTYS Bağlantısı

Uygulamanın çalışmasını sağlayan en önemli modüllerden biridir. VTYS bağlantı ayarları burada yapılmaktadır (Şekil 3.12). Bu ayarların yapılandırılması için farklı VTYS bağlantıları hakkında bilgi sahibi olmak ve SQL komutları iyi bilinmelidir. Uygulamada bir VTYS bağlantısı için gerekli olan parametreler şunlardır:

IP/URL (İnternet Protokolü – Internet Protocol / Standart Kaynak Belirteci - Uniform Resource Locator) adresi: VTYS'nin bulunduğu sunucu bilgisayarın ağ adres yoludur. Örneğin, kurum içi veri tabanı sunucu ip adresi olarak 10.0.1.12 ya da kurum dışı veri kaynağı olarak <https://obs.atauni.edu.tr/online/online2.txt> gibi bir url adresi kullanılabilir.

Sistem şifresi: VTYS'nin tüm yetkilerine sahip kullanıcılarından biri olan sistem kullanıcısının şifresidir. İstenildiğinde sistem kullanıcısı yerine VTYS'de tablo oluşturma, sorgu çalıştırma, veri tabanı erişimi, tetikleyici oluşturma ve silme yetkilerine sahip bir kullanıcı da oluşturulabilir.

VTYS bağlantı değişkeni: Her bir VTYS'nin kendine özel bağlantı komutları bulunmaktadır. Bu komutlar bir değişkene atanarak uygulama içinde kullanılabilir. Bu sayede uygulama gerektiğinde ilgili bağlantı değişkeniyle ilgili veri tabanına bağlantıyı sağlamaktadır. Farklı VTYS bağlantı değişkeni örnekleri Tablo 3.2'de gösterilmektedir.

Tablo 3.2. Farklı VTYS bağlantı değişkeni örnekleri.

VTYS	Bağlantı Değişkeni
Oracle	<pre>\$sonuc=oci_parse(oci_connect('system', 'oracle', ' (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME = orcl)))'), "sorgu"); oci_execute(\$sonuc);</pre>
MySQL	<pre>\$sonuc=(new mysqli('localhost:3306', 'root', 'oracle', 'Mysql'))->query("sorgu");</pre>
Microsoft SQL Server	<pre>\$consql=sqlsrv_connect('localhost\SQLEXPRESS', array('Database'=>(isset(\$_GET['username'])?\$_GET['username']:'TEST'), 'UID'=>'sa', 'PWD'=>'oracle', 'CharacterSet'=>'UTF- 8')); \$sonuc =sqlsrv_prepare(\$consql, "sorgu"); sqlsrv_execute(\$sonuc);</pre>
PostgreSQL	<pre>\$sonuc =pg_query(pg_pconnect("host=localhost port=5432 dbname=". isset(\$_GET['username'])?\$_GET['username']:'TEST')." user=postgres password=oracle"), "sorgu");</pre>

Yukarıda gösterilen Oracle bağlantı değişkeni \$sonuc değişkenine atanmıştır. \$sonuç değişkeni oci_execute konutuyla SQL sorgusu çalıştırmaktadır. Diğer VTYS bağlantı değişkenleri de bazı küçük farklılıklarla benzerlik göstermektedir.

Veri tabanlarını listeleme sorgusu: VTYS içerisinde farklı iş süreçleri için birçok veri tabanı bulunabilir. Bu veri tabanlarını listelemek için kullanılan SQL komutları VTYS'lere göre farklılık göstermektedir.

Veri tabanı tablolarını listeleme sorgusu: Veri tabanı tablolarının listesini sorgulayan SQL komutudur.

Tablo alanlarını listeleme sorgusu: Tablo alanlarının listesini sorgulayan SQL komutudur.

Tablo alanlarını getirme döngüsü: Tablo alanlarının uygulama içerisinde düzenli bir şekilde listelenmesini sağlar. Seçilen tablo alanlarının özelliklerine göre gruplanarak listelenmesini sağlamaktadır.

Veri tabanı kayıtlarını getirme döngüsü: Veri tabanından alınan bilgilerin herhangi bir filtre kullanmaksızın listelenmesini sağlar. VTYS içerisinde bulunan veri tabanları, tablolar ve tablo içerisindeki bilgilerin listelenmesi gibi görevleri yerine getirir.

Veri izleyicisi oluşturma komutu: Uygulamanın verileri izlemesini sağlayan en önemli SQL komutudur. Bu komut sayesinde tablo üzerinde herhangi bir değer değişikliği olduğunda yeni değerın kaydedilmesini sağlar. Böylece verinin gerçek zamanlı olarak izlenme işlemi sağlanmaktadır.

Veri izleyicisini silme komutu: Kullanıcılar daha önce takip ettikleri bir veri ile artık ilgilenmeyebilirler. Bu durumda uygulamanın gereksiz tablo izleme işlemini gerçekleştirmemesi için tetikleyiciyi silen bir komuttur.

Veri seti sorgusu: Toplanan verilerin istenilen zaman aralığına göre veri setini düzenleyen SQL komutudur. Elde edilen veri seti grafik modülünde kullanılmaktadır.

Veri setini getirme döngüsü: Veri seti sorgusu ile sorgulanan verilerin uygulama içerisinde kullanılmasını sağlayan veri getirme döngüsü komutudur. Uygulama içerisinde kullanılmak üzere alınan bu veriler veri kaynağı, zaman ve değer bilgisini içermektedir. Bu bilgi uygulama tarafından işlenerek grafiğe dönüştürülmektedir.

Veritabanı Bağlantısı

[Anasayfa](#)
[Oturumu Kapat](#)

<
>

	Yeni Bağlantı Adı	VTYS	IP/URL Adresi	Sistem Şifresi
Ekle	<input type="text"/> <small>Şablon seçiniz</small>	<input type="text"/>	<input type="text"/>	<input type="password"/>

Mevcut Bağlantılar

	Bağlantı Adı	VTYS	IP/URL Adresi	Sistem Şifresi
SİL	Microsoft SQL Veri Tabanı	SQLServer	localhost	<input type="password"/> <input style="font-size: small;" type="button" value="Şifre"/> <input style="font-size: small;" type="button" value="Sil"/>
SİL	PostgreSQL	PostgreSQL	localhost	<input type="password"/> <input style="font-size: small;" type="button" value="Şifre"/> <input style="font-size: small;" type="button" value="Sil"/>
SİL	Oracle Veri Tabanı	Oracle	localhost	<input type="password"/> <input style="font-size: small;" type="button" value="Şifre"/> <input style="font-size: small;" type="button" value="Sil"/>
SİL	MySQL	MySQL	127.0.0.1	<input type="password"/> <input style="font-size: small;" type="button" value="Şifre"/> <input style="font-size: small;" type="button" value="Sil"/>
SİL	Döviz Kuru	Harici	https://www.atauni.edu.tr/forms/currencies.json	<input type="password"/> <input style="font-size: small;" type="button" value="Şifre"/> <input style="font-size: small;" type="button" value="Sil"/>
SİL	Atauni Avesis	Harici	https://avesis.atauni.edu.tr/home/getpublicationcountsbyresearchareas	<input type="password"/> <input style="font-size: small;" type="button" value="Şifre"/> <input style="font-size: small;" type="button" value="Sil"/>
SİL	Atauni Obs	Harici	https://obs.atauni.edu.tr/online/online2.txt	<input type="password"/> <input style="font-size: small;" type="button" value="Şifre"/> <input style="font-size: small;" type="button" value="Sil"/>

Şekil 3.12. VTYS bağlantılarını yapılandırma modülü.

3.1.4.3.3. Tablo Erişimi

Uygulamada yer alan VTYS bağlantıları modülünde tanımlanan bağlantı adları burada listelenmektedir (Şekil 3.13). Bu bağlantı adlarından biri seçildiğinde seçili olan VTYS’de bulunan veri tabanları listelenmektedir. Farklı VTYS veri tabanlarını listeleme sorgu örnekleri Tablo 3.3’de gösterilmektedir.

Tablo 3.3. Farklı VTYS veri tabanlarını listeleme sorgu örnekleri.

VTYS	Sorgu
Oracle	select username from all_users where common='NO' order by username
MySQL	select schema_name as username from information_schema.schemata order by schema_name
Microsoft SQL Server	SELECT name as username FROM master.sys.databases where owner_sid<>0x01
PostgreSQL	select datname as username from pg_database where datistemplate='f'

Listelenen veri tabanlarından herhangi biri seçildiğinde bu veri tabanı içerisinde bulunan tablolar sorgulanmaktadır. Farklı VTYS veri tabanı tablolarını ve harici veri kaynaklarındaki dizileri listeleme sorgu örnekleri Tablo 3.4’de gösterilmektedir.

Tablo 3.4. Farklı VTYS veri tabanı tablolarını ve harici veri kaynaklarındaki dizileri listeleme sorgu örnekleri.

Veri Kaynağı	Sorgu
Oracle	SELECT table_name FROM all_tables where owner='".(isset(\$_GET['username'])?\$_GET['username']:')."'' order by table name
MySQL	SELECT table_name as table_name FROM information_schema.tables where table_schema='".(isset(\$_GET['username'])?\$_GET['username'] :')."''
Microsoft SQL Server	SELECT table_name FROM ".(isset(\$_GET['username'])?\$_GET['username']:'BEKLE').".IN FORMATION_SCHEMA.TABLES WHERE TABLE_TYPE='BASE TABLE'
PostgreSQL	SELECT table_name FROM information_schema.tables WHERE table_schema = 'public' json_decode(substr(□\$icerik□, strpos(□\$icerik□, '['), strlen(□ \$icerik□)-strpos(□\$icerik□, '[')-1));
Harici	json_decode(\$icerik) PHPExcel_IOFactory::load("multiTimeline.csv") PHPExcel_IOFactory::load("Özet Akademisyen Sayıları.xls")

Gerek VTYS içerisindeki veri tabanlarının gerekse de tabloların sorgulaması sonucunda elde edilen verileri listelemek için kullanılan döngü örnekleri Tablo 3.5’de gösterilmektedir.

Tablo 3.5. Genel amaçlı verileri listelemek için kullanılan döngü örnekleri.

Veri Kaynağı	PHP Kodları
Oracle	<pre>while (\$row=oci_fetch_array(\$sonuc, OCI_BOTH)) { echo str_replace('secilen', (isset(\$_GET['column'])?(\$_GET['column']==\$row[strtoupper('column')]"selected=selected":""), str_replace("degisken",\$row[strtoupper('column')], "secenek")); }; }</pre>
MySQL	<pre>while (\$row = \$sonuc->fetch_assoc()){ echo str_replace('secilen', (isset(\$_GET['column'])?(\$_GET['column']==\$row['column']?"selected=selected":""):""), str_replace("degisken", \$row['column'], "secenek")); }</pre>
Microsoft SQL Server	<pre>while (\$row=sqlsrv_fetch_array(\$sonuc, SQLSRV_FETCH_ASSOC)) { echo str_replace('secilen', (isset(\$_GET['column'])?(\$_GET['column']==\$row['column']?"selected=selected":""):""), str_replace("degisken", \$row['column'], "secenek")); } }</pre>
PostgreSQL	<pre>while (\$row=pg_fetch_assoc(\$sonuc)) { echo str_replace('secilen', (isset(\$_GET['column'])?(\$_GET['column']==\$row['column']?"selected=selected":""):""), str_replace("degisken", \$row['column'], "secenek")); } }</pre>
Harici	<pre>foreach(\$sonuc as \$key=>\$val){ echo str_replace('secilen', (isset(\$_GET['column'])?(\$_GET['column']==\$key?"selected=selected":""):""),str_replace("degisken", \$key, "secenek")); } }</pre>

Son olarak, oluşturulan bu tablo erişim sürecinde kullanılan parametrelere sahip bir erişim adı atanır. Görüldüğü üzere, uygulamada tanımlanan bir erişim bağlantı adı, veri tabanı ve tablo bilgisini içermektedir. Tablo erişim modülü sayesinde istenilen tablonun erişim yolu tanımlanmaktadır.

Tablo Erişimi Anasayfa Oturumu Kapat

	Bağlantı Adı	Veritabanı	Tablo	Erişim Adı
Ekle	<input type="text" value="Bağlantı seçiniz"/>	<input type="text" value="Veri Tabanı seçiniz"/>	<input type="text" value="Tablo seçiniz"/>	<input type="text"/>
Mevcut Erişimler				
	Bağlantı Adı	Veritabanı	Tablo	Erişim Adı
Sil	Atauni Avesis	AVESIS	TopicName[PublicationCount]	Avesis Akademik Veri
Sil	Atauni Öbs	ÖBS	TOPLAM	Öbs Toplam Kullanıcı
Sil	Döviz Kuru	BANKA	Kod[Alis]	Döviz Alış Kuru
Sil	Döviz Kuru	BANKA	Kod[Satis]	Döviz Satış Kuru
Sil	Microsoft SQL Veri Tabanı	ZIYARETCILER	GELEN	Gelen Ziyaretçi
Sil	MySQL	muhasabe	butce	Bütçe Durumu
Sil	Oracle Veri Tabanı	ARASTIRMA	YAYIMLAR	İndexli Yayimler
Sil	PostgreSQL	KARTLAR	OGRENCI_KARTI	Bina Turnikeleri
<input type="button" value="Kaydet"/>				

Şekil 3.13. Tablo erişimi sağlama modülü.

Seçilen VTYS'ler üzerinde veri tabanı tablolarından özet veri toplamak için kullanılan sorgular aşağıdaki örneklerle gösterilmektedir.

```
create or replace
trigger VIS.VERI_TOPLA10 AFTER INSERT OR DELETE OR UPDATE ON
VIS.VERI
BEGIN
INSERT INTO VIS.VERI (KAYNAK, DEGER) VALUES ('Sistem veri girişi',
(SELECT COUNT (*) SAYI FROM VIS.VERI));
END;
```

Yukarıda verilen kod parçası Oracle'da veri toplama işlemini gerçekleştirmektedir. Yukarıda verilen tetikleyici, veri tabanı tablosunda INSERT, DELETE ve UPDATE işlemleri gerçekleştiğinde aktif hale gelmektedir. Tetikleyici aktifleştikten sonra INSERT işlemi ile istenen tablodan özet veri kaydedilmektedir.

```

CREATE DEFINER=`root`@`localhost`
TRIGGER `VERI_TOPLA10_INS`
AFTER INSERT ON `butce`
FOR EACH ROW
INSERT INTO TEST.VERI
(KAYNAK, DEGER)
VALUES
('Genel gelir gider-farkı verisi',
(SELECT SUM (GELIR) - SUM (GIDER) SAYI FROM muhasebe.butce))

```

Yukarıda MySQL'de veri toplama işlemini gerçekleştiren tetikleyici kodları görülmektedir. Oracle'da olduğu gibi MySQL'de de tetikleyici iki ana kısımdan oluşmaktadır. İlk kısımda, tetikleyicinin aktif olma yöntemi, diğer kısımda ise tetiklendikten sonra yapılacak işlemler yer almaktadır. Buna göre, tetikleyici tabloda INSERT işlemlerinden sonra aktif hale gelmektedir. Oracle tetikleyicilerinden farklı olarak diğer DELETE ve UPDATE işlemleri için ayrı ayrı tetikleyici oluşturulmalıdır. Diğer bir deyişle MySQL tetikleyicileri yalnızca bir eylemle aktifleşmektedirler. Oracle'da bir tetikleyici farklı eylemler aktifleştirecek şekilde tanımlanabilmektedir. Yukarıda verilen tetikleyicinin ikinci kısmı Oracle'da da olduğu gibi INSERT işlemi ile istenilen tablonun alanları arasındaki farkı kaydetmektedir.

```

USE [deneme]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TRIGGER [dbo].[VERI_TOPLA10]
ON [dbo].[SINIF]
AFTER INSERT, DELETE, UPDATE
AS
BEGIN
INSERT INTO TEST.VERI
(KAYNAK, DEGER)
VALUES
('msislem', (SELECT COUNT(*) SAYI FROM deneme.SINIF));
END

```

Yukarıda Microsoft SQL Server'da veri toplama işlemini gerçekleştiren tetikleyici kodları verilmiştir. Kodlar incelendiğinde tetikleyici Oracle'da olduğu gibi farklı işlemlerle aktif hale gelebilme özelliğine sahiptir. Burada da görüldüğü üzere, INSERT, UPDATE ve DELETE işlemleri tablo üzerinde gerçekleştiği takdirde INSERT işlemi uygulanmaktadır. INSERT işlemi istenilen tablodan alınan verileri kaydetmektedir.

```

CREATE FUNCTION public.veri_topla10()
  RETURNS trigger
  LANGUAGE 'plpgsql'
  COST 100
  VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN
INSERT INTO public.VERI
  (KAYNAK, DEGER)
VALUES
  ('pgişlem', (SELECT COUNT(*) SAYI FROM public.PGTABLE));
RETURN NEW;
END;
$BODY$;

ALTER FUNCTION public.veri_topla10()
  OWNER TO postgres;

CREATE TRIGGER veri_topla10
  AFTER INSERT OR DELETE OR UPDATE
  ON public."PGTABLE"
  FOR EACH ROW
  EXECUTE PROCEDURE public.veri_topla10();

```

Yukarıda verilen kodlar PostgreSQL’de tetikleyici tanımlanması için kullanılmaktadır. Burada ilk dikkat çeken husus tetikleyicinin tanımlanmasında kullanılan kodlar diğer seçilen VTYS’lere göre daha karmaşık bir yapıya sahip olmasıdır. Çünkü PostgreSQL, tetikleyicisinin tanımlanabilmesi için öncelikli olarak bir tetikleyici fonksiyon tanımlamasına ihtiyaç duyulmaktadır. PostgreSQL’de tetikleyiciler direk olarak bir SQL sorgusunu çalıştıramamaktadırlar. SQL sorgusu bir fonksiyonun içerisine yazılır. Daha sonra oluşturulan tetikleyicide bu fonksiyon belirtilir. Fonksiyon içerisinde tanımlanan sorgu tetikleyici aktif olduktan sonra gerçekleştirilecek olan işlemleri içermektedir. Tetikleyici ilgili tablo üzerinde INSERT, UPDATE ve DELETE işlemlerinin gerçekleşmesi halinde aktif hale gelmektedir. Tetikleyici aktif hale geldikten sonra INSERT sorgusu ile veri kaydı gerçekleştirilmektedir. Farklı VTYS için tetikleyici silme komut örnekleri Tablo 3.6’da gösterilmektedir.

Tablo 3.6. Farklı VTYS için tetikleyici silme komut örnekleri.

VTYS	Sorgu
Oracle	DROP TRIGGER ".\$part[1]."
MySQL	DROP TRIGGER ".\$part[1]."_INS
Microsoft SQL Server	DROP TRIGGER ".explode('.', \$part[1])[1].";
PostgreSQL	drop trigger ".explode('.', \$part[1])[1]." on ".sprintf('%s', \$_POST['tablo'].\$i)."; drop function ".explode('.', \$part[1])[1].";"

3.1.4.3.4. Veri Toplama İşlemi

Veri kaynaklarından sağlıklı bir şekilde veri elde etmek için öncelikli olarak veri kaynakları iyi tanınmalıdır. Gerek literatür taraması gerekse de yapılan test çalışmalarında veri kaynağıyla ilgili iki önemli hususun öne çıktığı görülmektedir. Bunlar; veri kaynağındaki verilerin toplanması ve veri kaynağındaki tarih verisinin önemidir. Bu hususlar haricinde de veri kaynağına erişim gibi birçok unsur sayılabilir. Özellikle bu hususların bu çalışmada veri analizi esnasında karşılaşılan zorluklardan biri olarak gözlemlenmiştir.

Veri toplama işlemi veri kaynağına göre farklılık arz etmektedir. Ancak, bu farklılıklar sınırlı sayıda olduğundan KMİS’de tanımlanabilmektedir. Veri toplamadaki farklılıklar kullanılan VTYS ve harici veri dikkate alınarak iki ana başlık altında incelenmiştir.

KMİS’de toplanan gerçek zamanlı veriler bir işlem adı ile temsil edilir (Şekil 3.14). Bu sayede kullanıcılar VTYS, veri tabanı, tablo, alan, SQL komutu, erişim adı, vb. teknik ayrıntılara girmeden bir işlem adı altında takip edebilmektedirler.

Tablo erişimi sağlandıktan sonra bu tablodan özet veri toplanması gerekmektedir. Veri toplama işlemi için öncelikli olarak içerisinde VTYS bağlantısı, veri tabanı ve tablo bilgisi bulunduran erişim adı seçilir. Bu tablodaki alanlar türlerine göre gruplanarak listelenir. Alanları listelemedeki amaç veri toplama işlemi gerçekleştirecek olan SQL komutunun düzenlenmesine yardımcı olmaktır. Farklı VTYS veri tabanı tablo alanlarını listeleme sorgu örnekleri Tablo 3.7’de gösterilmektedir. Harici veri kaynaklarındaki alanları listeleme örnekleri Ek 2’de verilmiştir.

Tablo 3.7. Farklı VTYS veri tabanı tablo alanlarını listeleme sorgu örnekleri

VTYS	Sorgu
Oracle	<pre>select data_type, listagg(column_name,', ') within group(order by column_name) column_name from ALL_TAB_COLUMNS WHERE owner='".\$erisimbilgi['VERITABANI'].'" and table_name = '".\$erisimbilgi['SIFRE'].'" group by data type</pre>
MySQL	<pre>"select data_type, GROUP_CONCAT(column_name SEPARATOR ', ') COLUMN_NAME from information_schema.columns WHERE table_schema='".\$erisimbilgi['VERITABANI'].'" and table_name = '".\$erisimbilgi['SIFRE'].'" group by data type"</pre>
Microsoft SQL Server	<pre>SELECT data_type, column_name = STUFF((SELECT ', ' + column_name FROM '".(isset(\$_GET['username'])?\$_GET['username']:'BEKLE')."'.INF ORMATION_SCHEMA.COLUMNS t1 WHERE table_name='".\$erisimbilgi['SIFRE'].'" and t1.data_type = t2.data_type FOR XML PATH ('')), 1, 1, '') from '".(isset(\$_GET['username'])?\$_GET['username']:'BEKLE')."'.INF ORMATION_SCHEMA.COLUMNS t2 where table_name='".\$erisimbilgi['SIFRE'].'" group by data type;</pre>
Postgre SQL	<pre>SELECT data_type, STRING_AGG(column_name, ', ' ORDER BY column_name) AS column_name FROM information_schema.columns where table_name = '".\$erisimbilgi['SIFRE'].'" GROUP BY data_type</pre>

Farklı VTYS veri tabanı tablo alanlarını getirme döngü örnekleri Tablo 3.8'de gösterilmektedir.

Tablo 3.8. Farklı VTYS veri tabanı tablo alanlarını getirme döngü örnekleri.

VTYS	PHP Kodu
Oracle	<pre>while ((\$row=oci_fetch_array(\$sonuc, OCI_BOTH)) != false){ echo "<tr><td>".\$row['DATA_TYPE']. "</td><td>".\$row['COLUMN_NAME'] . "</td><tr>"; } </pre>
MySQL	<pre>while (\$row = \$sonuc->fetch_assoc()){ echo "<tr><td>".\$row['DATA_TYPE']. "</td><td>".\$row['COLUMN_NAME'] . "</td><tr>"; } </pre>
Microsoft SQL Server	<pre>while (\$row=sqlsrv_fetch_array(\$sonuc, SQLSRV_FETCH_ASSOC)) { echo "<tr><td>".\$row['data_type']. "</td><td>".\$row['column_name'] . "</td><tr>"; } </pre>
Postgre SQL	<pre>while (\$row=pg_fetch_assoc(\$sonuc)) { echo "<tr><td>".\$row['data_type']. "</td><td>".\$row['column_name'] . "</td><tr>"; } </pre>

Ayrıca erişim adı seçildiğinde veri tabanı bağlantısı modülünde tanımlanan VTYS parametreleri doğrultusunda veri tabanı ismi ve tablo adını içeren SQL komutunun temel yapısı otomatik olarak yapılandırılmaktadır. SQL komunu istediğimiz veriyi toplayacak şekilde düzenledikten sonra veri toplama yöntemi seçilir. Veri toplama yöntemi olarak veri kaynağına göre tetikleyici ekleme, sorgulama ve veri dosyası seçeneklerinden biri seçilir. Veri kaynağının performansının olumsuz etkilenmesi için veri toplama sıklığı olarak dakikalık, saatlik ve günlük seçeneklerinden biri seçilir. Son olarak, bu işleme bir ad verilir ve ekle tuşuna basılır. Veri toplama yöntemi olarak tetikleyici ekleme seçilmesi durumunda; ekle tuşuna basıldığı an itibari ile tanımlanan veri tabanı bağlantısı modülünde VTYS parametreleri doğrultusunda uygulama otomatik olarak düzenlediğimiz SQL komutunu içeren bir tetikleyiciyi bağlantıyı gerçekleştirdiğimiz veri tabanı yönetim sistemine eklemektedir. Bu sayede izlediğimiz tabloda bir değişiklik olması durumunda sıklıkta SQL komutu tetiklenerek veriler toplanmaktadır. Sorgulama ya da veri dosyası seçilmesi durumunda; istenilen sıklıkta

veri kaynağına bağlantı gerçekleştirilerek güncel veriyi sorgulayıp ya da veri dosyasından verileri alarak kendi veri tabanına kaydetmektedir.

Veri Toplama İşlemi

[Anasayfa](#)
[Oturumu Kapat](#)

Erişim Adı

Alanlar

Yöntem

Tetikleyici ekleme
 Sorgulama
 Veri dosyası

Güncelleme

Dakikalık Saatlik Günlük

İşlem Adı

Lütfen, gruplama fonksiyonu içeren sorgu yazınız.

```
SELECT COUNT(*) SAYI FROM
```

Anlık Veri İzleme ve Kayıt Penceresi

Mevcut İşlemler

	İşlem Adı	İşlem Adresi
<input type="button" value="Sil"/>	Toplam Akademisyen Sayısı	Harici@ISTATISTIK.VERI_TOPLA9
<input type="button" value="Sil"/>	Öbs Toplam Kullanıcı Sayısı	Harici@ÖBS.VERI_TOPLA8
<input type="button" value="Sil"/>	Gelen Ziyaretçi Sayısı	SQLServer@ZIYARETCILER.VERI_TOPLA10
<input type="button" value="Sil"/>	Bütçe gelir-gider farkı	MySQL@muhasabe.VERI_TOPLA10
<input type="button" value="Sil"/>	Toplam indexli yayımlar	Oracle@ARASTIRMA.VERI_TOPLA10
<input type="button" value="Sil"/>	Avesis Toplam Yayınlar	Harici@AVESIS.VERI_TOPLA10
<input type="button" value="Sil"/>	Usd Alış Kuru	Harici@BANKA.VERI_TOPLA9
<input type="button" value="Sil"/>	Usd Satış Kuru	Harici@BANKA.VERI_TOPLA10
<input type="button" value="Sil"/>	Gbp/Usd Alış Kuru	Harici@BANKA.VERI_TOPLA10

Şekil 3.14. Veri toplama işlemi modülü.

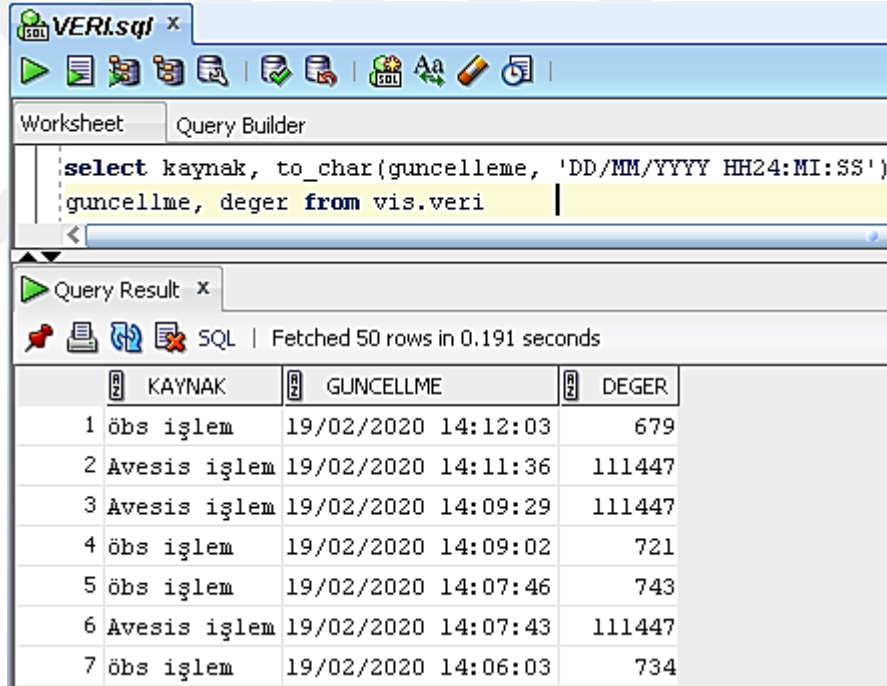
KMİS çeşitli kaynaklardan topladığı verileri tutarlı bir şekilde kendi veri tabanına kaydetmektedir. Verilerin kaydedildiği tablonun ismi de “VERI”dir. Veri tablosunun sorgusal tanımı aşağıda verilmektedir.

```

CREATE TABLE "VIS"."VERI"
  ( "KAYNAK" VARCHAR2(100 BYTE),
    "GUNCELLEME" DATE DEFAULT SYSDATE NOT NULL ENABLE,
    "DEGER" NUMBER(10,3)
  ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
  NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS";

```

Yukarıda verilen sorgusal tablo tanımlamasına göre VERI tablosunun üç alanı bulunmaktadır. Bunlar sırasıyla KAYNAK, GUNCELLEME ve DEGER alanlarıdır. Bu alanların tanımlarına bakıldığında; KAYNAK alanı VARCHAR2 veri tipinde karaktere dayalı değerler için, GUNCELLEME alanı DATE veri tipinde tarihsel değerler için, DEGER alanı NUMBER veri tipinde sayısal değerler için tanımlanmıştır.



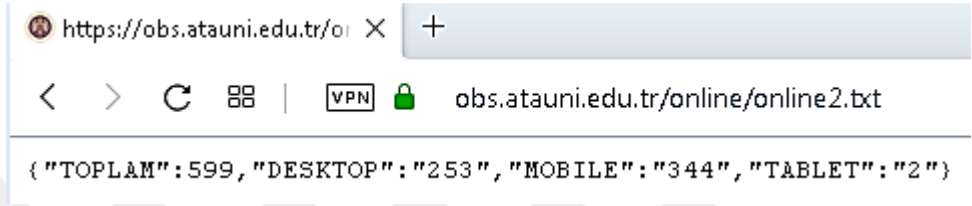
The screenshot shows a SQL query editor window titled 'VERI.sql'. The query is: `select kaynak, to_char(guncelleme, 'DD/MM/YYYY HH24:MI:SS') guncelleme, deger from vis.veri`. Below the query, the 'Query Result' window shows the following data:

	KAYNAK	GUNCELLEME	DEGER
1	öbs işlem	19/02/2020 14:12:03	679
2	Avesis işlem	19/02/2020 14:11:36	111447
3	Avesis işlem	19/02/2020 14:09:29	111447
4	öbs işlem	19/02/2020 14:09:02	721
5	öbs işlem	19/02/2020 14:07:46	743
6	Avesis işlem	19/02/2020 14:07:43	111447
7	öbs işlem	19/02/2020 14:06:03	734

Şekil 3.15. VERI tablosunun alanları.

Şekil 3.15'de KMİS verilerin saklandığı tablo olan VERI tablosu gösterilmektedir. Şekilde de görüleceği üzere tabloda KAYNAK, GUNCELLEME ve DEGER olmak üzere üç adet alan içermektedir. KAYNAK alanı veri kaynağının tanımlayan karakter tipinde veriyi kaydetmektedir. GUNCELLEME alanı tarih verisini ve DEGER alanı verinin aldığı sayısal değeri saklamaktadır.

Şekil 3.15’de GÜNCELLEME alanında bulunan tarih değeri tablo alan tanımlamalarında da görüleceği üzere SYSDATE komutu ile oluşturulmaktadır. SYSDATE, komutu Oracle’da sistem tarihini GÜNCELLEME alanına kaydetmektedir. Dolayısıyla, burada dikkat edilmesi gereken husus bu tarih verisinin tabloya kaydedilme zamanıdır. Gerçek zamanlı olarak oluşturulan verinin zaman etiketinde çok az da olsa farklılıklar olabilir. Bu nedenle, daha sağlıklı bir veri seti için kaynaktaki tarih verisinin kullanılması gerekmektedir. Ancak, dış kaynaklardan alınan verilerin çoğunda tarih etiketi bulunmamaktadır.



Şekil 3.16. Tarih bilgisini içermeyen veri dizisi.

Şekil 3.16’da veri kaynağından alınan veriler görülmektedir. Görüldüğü üzere, veri kaynağında herhangi bir tarih bilgisi bulunmamaktadır. Dolayısıyla, bu veri dizisi tarih etiketi olarak VTYS’nin VERI tablosuna kaydettiği andaki tarih değeri ile kaydedilmektedir.

3.1.4.3.5. Veri İzleme ve Kayıt Penceresi

Veri toplama işlemi için gerekli olan SQL komutu belirlendikten sonra gerçek zamanlı olarak erişim ve kaydetme işlemi Veri İzleme ve Kayıt Penceresi ekranında gerçekleştirilmektedir (Şekil 3.17).

Veri İzleme ve Kayıt Penceresi, tanımlı mevcut işlemlerin yürütülmesini ve elde edilen verilerin görüntülenmesini sağlamaktadır. Böylece sistem yöneticisi gerektiğinde gerçek zamanlı veri toplama işlemlerini kontrol edebilmektedir. Bu pencerede işlemin adı, değeri ve güncelleme için kalan saniye görüntülenmektedir. Her bir işlemin güncelleme sıklığı veri kaynağını olumsuz etkilememesi için farklılık arz edebilmektedir. Farklı veri kaynaklarında bağlanma ve veri getirme işlemleri sırasında milisaniye bazında da olsa gecikmeler yaşanabilmektedir. Tüm işlemler için veri kaydetme işlemine başladıktan belirli bir süre sonra güncelleme sıklığı aynı da olsa

güncellemeye kalan süresi birbirinden bağımsız olarak azalarak birbirinden farklılaşmaktadır.

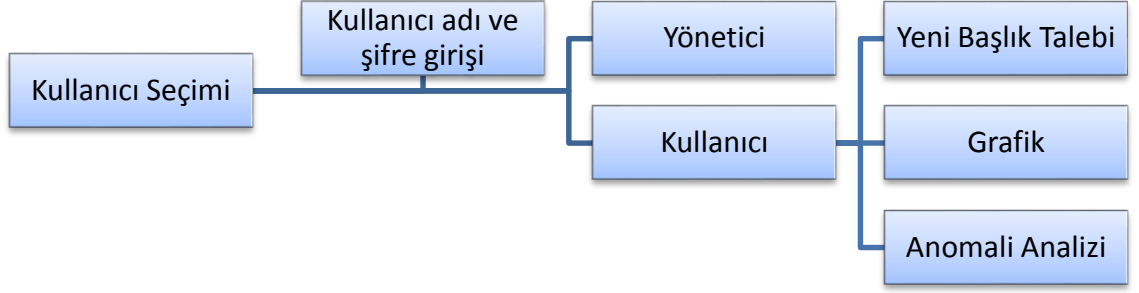
ANLIK VERİ İZLEME VE KAYIT

Veri	Değer	Güncelleme kalan saniye
Gelen Ziyaretçi Sayısı	30	594
Usd Alış Kuru	7.872	7
Öbs Toplam Kullanıcı Sayısı	1871	3157
Avesis Toplam Yayınlar	115735	6734
Gbp/Usd Alış Kuru	1.305	24
Usd Satış Kuru	7.918	7
Bütçe gelir-gider farkı	85600	6743
Toplam indexli yayımlar	5002	6703

Şekil 3.17. Gerçek zamanlı veri izleme ve kayıt penceresi.

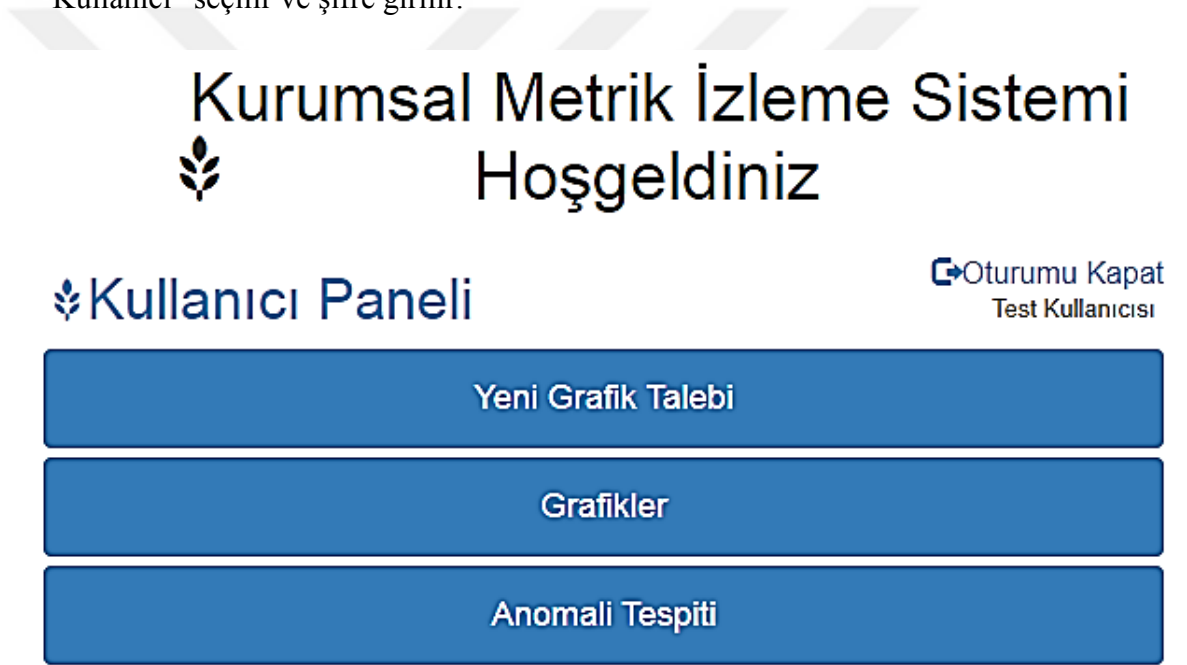
3.1.5. Kullanıcı Paneli ve Veri Analiz Arayüzü

Kullanıcı tiplerinden diğer bir kullanıcı çeşidi olan “Kullanıcı” sistemi kullanacak olan kişiler için tanımlanmıştır. Kullanıcılara sistem yöneticisi tarafından kullanıcı adı ve şifre tanımlanmış olmalıdır. Bu sayede kullanıcılar sisteme giriş yapabilmektedirler.



Şekil 3.18. Yönetici Paneline erişim ve Yönetici Panelinde kullanılan modüller.

Şekil 3.18’de Kullanıcı Panelinde bulunan modüllerin erişimi sağlayan kullanım diyagramı görülmektedir. Bu modüllere erişimin sağlanması için kullanıcı tipi olarak “Kullanıcı” seçilir ve şifre girilir.



Şekil 3.19. Uygulama kullanıcı paneli.

Şekil 3.19’da prototip uygulamanın kullanıcı paneli görülmektedir. Kullanıcıların gerekli modülleri kolay kullanabilmesi için fazla teknik bilgi gerektirmeyen, basit ve görsel bir arayüz hazırlanmıştır. Bu panelde üç adet modül erişimi bulunmaktadır. Sistem yöneticisine talepte bulunma, veri akışının grafik üzerinde izlenmesi ve tespit edilen anomalileri görüntüleme gibi uygulama servisleri sunulmaktadır.

3.1.5.1. Yeni Grafik Talebi

Yöneticilerin ihtiyaç duyduğu bilgi ihtiyacının belirlenmesinde iki esas göz önünde bulundurulmuştur. Bunlardan ilki ve en önemlisi kullanıcının talebine göre. Bir diğer anlamda sistemin kişiselleştirilebilme özelliğine sahip olmasıdır. Bu kişiselleştirme özelliği Şekil 3.20’de görülen KMİS’in Yeni Grafik Talebi modülü ile sağlanmaktadır.

Kullanıcılar yeni bir grafiğe ihtiyaç duyduklarında bunu sistem yöneticisine bu modül ile bildirebilmektedirler. Yeni grafik talebinde kullanıcı istediği verinin özelliklerini tanımlayabilmektedir. Yapılan taleplerin takibi bu modül üzerinden yapılabilmektedir. Uygulama yöneticisi bu talepleri inceleyerek mevcut VTYS bağlantıları üzerinden ya da yeni bir VTYS bağlantısı oluşturarak bir grafik oluşturur. Sistem yöneticisi talep edilen grafiği oluşturulduktan sonra kullanıcı talebini “Hazır” olarak işaretleyerek kullanıcıya bildirir. Kullanıcı ise oluşturulan bu grafiği kendi panelinde görüntüleyebilmektedir.

Yeni Grafik Talebi
[Anasayfa](#)
[Oturumu Kapat](#)

	No	Tanım	
<input type="button" value="Ekle"/>			

Mevcut Talepler

	No	Tanım	Durum	Kullanıcı No
<input type="button" value="Sil"/>	1	İndexli dergilerde yayımlanan makalelerin toplam sayısının izlenmesi	Hazır	1003
<input type="button" value="Sil"/>	2	ÖBS aktif toplam kullanıcı sayısı	Hazır	1003
<input type="button" value="Sil"/>	3	Bina içerisine giriş yapan öğrenci sayısı	Hazır	1003
<input type="button" value="Sil"/>	4	Döviz alış ve satış kur bilgisi	Beklemede	1003

Şekil 3.20. Kullanıcı yeni grafik talebi modülü.

3.1.5.2. Grafikler

Kullanıcılar açısından en önemli modüldür (Şekil 3.21). Grafikler, açık kaynak kodlu chart.js JavaScript eklentisinin 2.8.0 sürümünün KMİS ile tümleştirilmesi sayesinde çizilmiştir. Bu sayede, kullanıcının seçtiği veri seti zaman serisi grafiği üzerinde gösterilmektedir. Oracle veri seti sorgusu örneği aşağıda gösterilmektedir. Diğer VTYS ve harici veri seti sorgu örnekleri Ek 3’de verilmiştir.

```
select guncelleme, sum(degers) degers from
((select guncelleme, round(avg(deger),3) degers from (
SELECT substr( TO_CHAR(guncelleme, 'YYYYMMDDHH24MISS'), 0, ".$grup.")
guncelleme, to_number(deger) deger
from vis.veri where GUNCELLEME<(sysdate +
numtodsinterval($(".$kaydir."), ".$zaman_ing."))
and guncelleme > (sysdate - numtodsinterval($(".($uzunluk+$kaydir)."),
".$zaman_ing."))
and kaynak=".$rowa['ADI'].") dt
group by guncelleme)
```



```

union
(select substr( TO_CHAR(sysdate +
numtodsinterval((rownum+(".$kaydir-$uzunluk).")), '$zaman_ing.'),
'YYYYMMDDHH24MISS'), 0, ".$grup.") araliklar, 0
from all_objects where rownum<=".$uzunluk.") )
group by guncelleme
order by guncelleme
fetch first ".$uzunluk." rows only

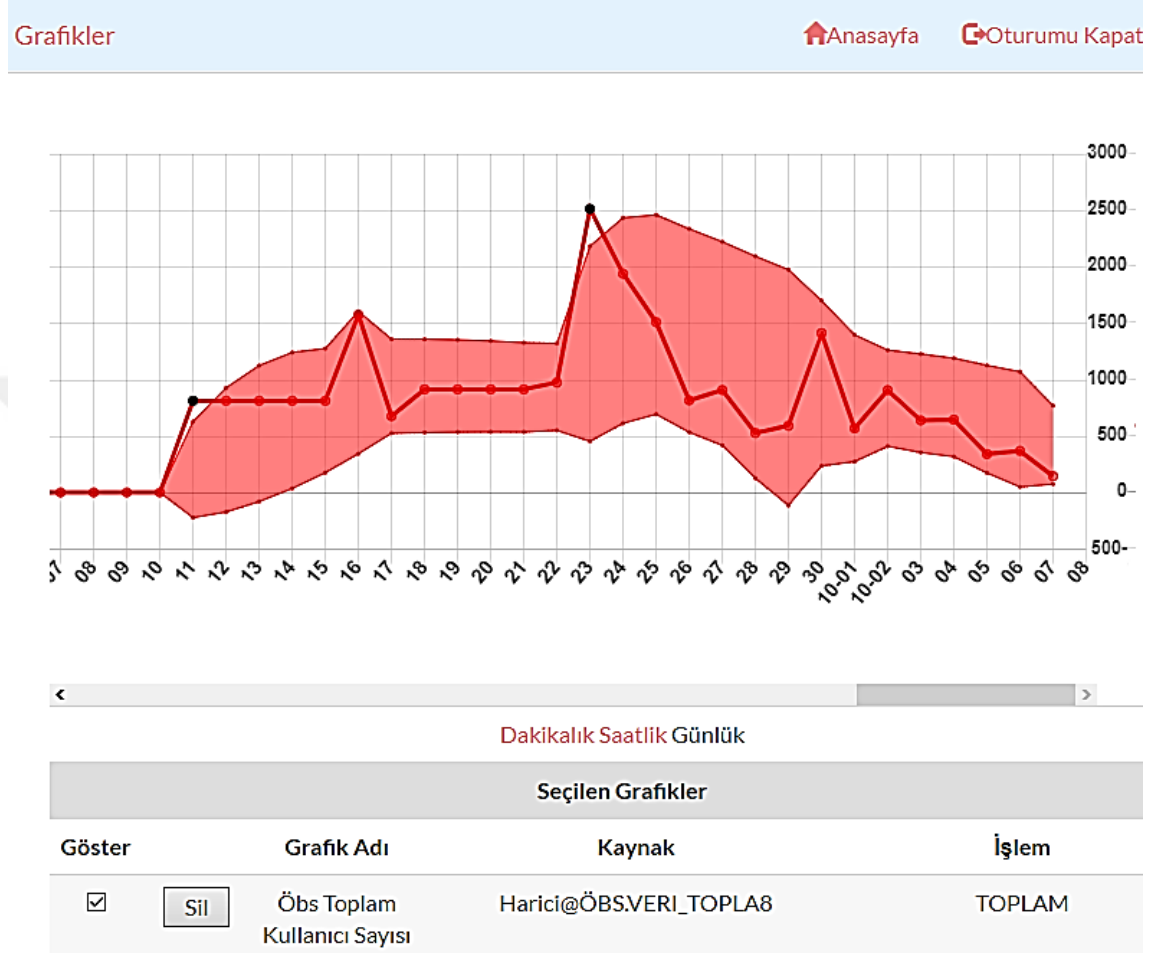
```

Farklı VTYS veri setini getirme döngüsü örnekleri Tablo 3.9'da gösterilmektedir. Harici veri kaynaklarındaki örnekleri Ek 4'de verilmiştir.

Tablo 3.9. Farklı VTYS veri setini getirme döngüsü örnekleri.

VTYS	PHP Kodu
Oracle	<pre> while ((\$row=oci_fetch_array(\$sonuc, OCI_BOTH)) != false){ \$say++; if(\$row['DEGERS']==0) {\$row['DEGERS']=\$onceki;} \$deger =\$deger.",",".str_replace(',','.', \$row['DEGERS']); \$onceki=\$row['DEGERS']; } </pre>
MySQL	<pre> while (\$row = \$sonuc->fetch_assoc()){ \$say++; if(\$row['DEGERS']==0.000) \$row['DEGERS']=\$onceki; \$deger =\$deger.',',".str_replace(',','.', \$row['DEGERS']); \$onceki=\$row['DEGERS']; } </pre>
Microsoft SQL Server	<pre> while (\$row=sqlsrv_fetch_array(\$sonuc, SQLSRV_FETCH_ASSOC)) { \$say++; if(@\$row['degers']==0) {\$row['degers']=\$onceki;} \$deger =\$deger.",",".str_replace(',','.', \$row['degers']); \$onceki=\$row['degers']; } </pre>
Postgre SQL	<pre> while (\$row=pg_fetch_assoc(\$sonuc)) { \$say++; if(\$row['degers']==0) {\$row['degers']=\$onceki;} \$deger =\$deger.',',".str_replace(',','.', \$row['degers']); \$onceki=\$row['degers']; } </pre>

Anomalilerin tespit edilmesini sağlayan tahmin aralığı bu grafik üzerinde yer almaktadır. Bazı verilerin bu tahmin aralığı dışına çıkması anomali olarak kabul edilmektedir (Şekil 3.21).



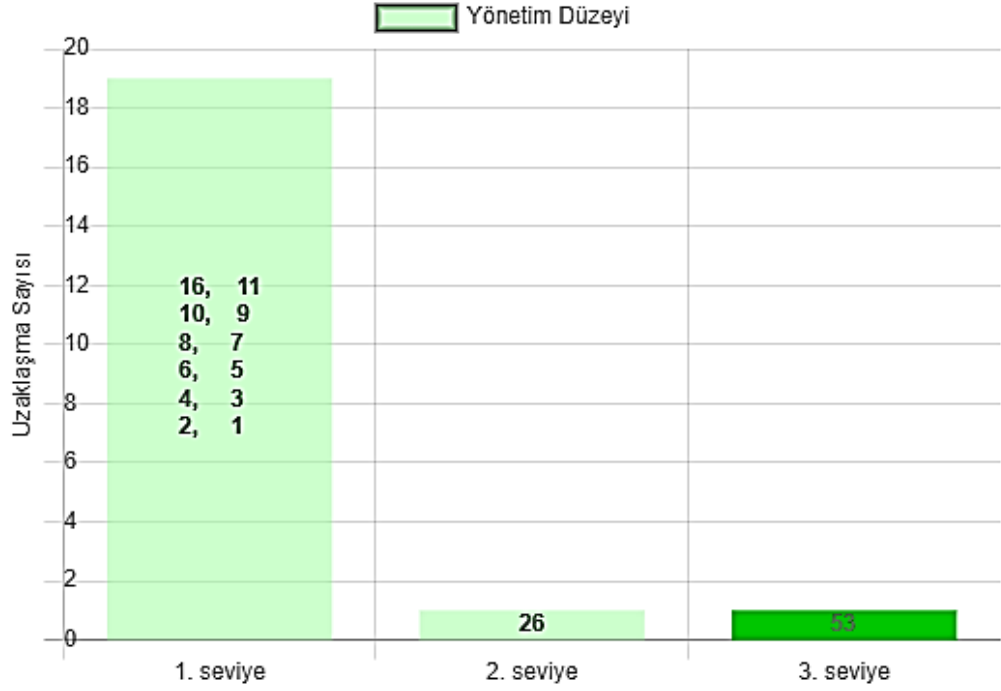
Şekil 3.21. Kullanıcı grafik modülü.

Her bir anomali zaman ve değer bilgisi ile birlikte uygulama veri tabanına kaydedilmektedir. Kullanıcı isterse birden fazla grafiği aynı grafik üzerinde görüntüleyebilmektedir. Bunun için Göster alanı altında ilgili grafiğin sol tarafında bulunan onay kutusu seçilir. Böylece bu veri grafiğe eklenir. Kullanıcılar isterlerse diğer seviyelerde bulunan, ancak ilgilendikleri bir grafiği Ekle düğmesi ile Seçilen Grafikler arasına ekleyebilmektedir. İlgilenmediği bir grafiği Sil düğmesi ile çıkarabilmektedir.

3.1.5.3. Anomali Analizi

Kullanıcıların görüntülediği grafik üzerinde tespit edilen anomaliler uygulama tarafından otomatik olarak grafik adı, zaman ve değer bilgileri ile birlikte kaydedilmektedir (Bkz. Ek 5). Anomalilerin bir kısmı tahmin aralığına çok yakın bir kısmı ise çok uzak birim mesafededir. Genellikle anomaliler tahmin aralığına yakın mesafede yoğunlaşmaktadır. Uzak mesafedeki anomaliler daha seyrek dağıldıkları görülmektedir. Anomaliler mesafesine göre üç seviyede sınıflandırılmaktadır. Tahmin aralığına yakın mesafede bulunan ve sık rastlanan noktalar üçüncü seviye olarak sınıflandırılır. Biraz daha uzak mesafede olanlar ikinci seviye ve en uzak mesafede bulunanlar üçüncü seviye olarak sınıflandırılır.





Dakikalık Saatlik Günlük

Seçilen Grafikler

	Grafik	Vtys	Veri Tabanı	Düzey
<input type="radio"/>	Gelen Ziyaretçi Sayısı	SQLServer	ZİYARETCILER	Orta
<input checked="" type="radio"/>	Avesis Toplam Yayınlar	Harici	AVESIS	Üst

Şekil 3.22. Kullanıcı anomali analiz modülü.

Şekil 3.22’de görüldüğü üzere, her bir sınıfın tahmin aralığından uzaklaşma mesafesi ilgili seviye sütununda görüntülenmektedir. Kullanıcı anomali analizi modülünü ilk defa açtığında varsayılan değer olarak birinci seviyede tespit edilen anomalileri görüntülemektedir. Ancak, kullanıcı istediği takdirde daha alt (ikinci ve üçüncü) seviyedeki anomalileri de görüntüleyebilmektedir. Ayrıca kullanıcı daha az sayıdaki üst düzeye sınıflandırılmış olan anomalileri de seçebilmektedir.

Anormal Veriler				
	Uzaklık	Tarih	Grafik Adı	Düzyey
<input checked="" type="radio"/>	53	03:00:00 18-02-2020	Avesis Toplam Yayınlar	Üst
<input type="radio"/>	26	03:00:00 09-06-2020	Avesis Toplam Yayınlar	Orta

Açıklama:

Üst düzeyde tanımlı Avesis Toplam Yayınlar adlı Günlük grafikte 2020-02-18 03:00:00 tarihinde beklenti aralığından 53 birimlik uzaklaşma anomali olarak tesbit edilmiştir.

Teknik Detay:

Veri Tabanı Sistemi	:	Harici
Veri Tabanı Adı	:	AVESIS
Tablo Adı	:	TopicName[PublicationCount]
Tablo Erişim Adı	:	Avesis Akademik Veri
Tablo İşlem Adı	:	Avesis Toplam Yayınlar
Veri Tabanı Bağlantı Adı	:	Atauni Avesis

Şekil 3.23. Kullanıcı anomali listesi.

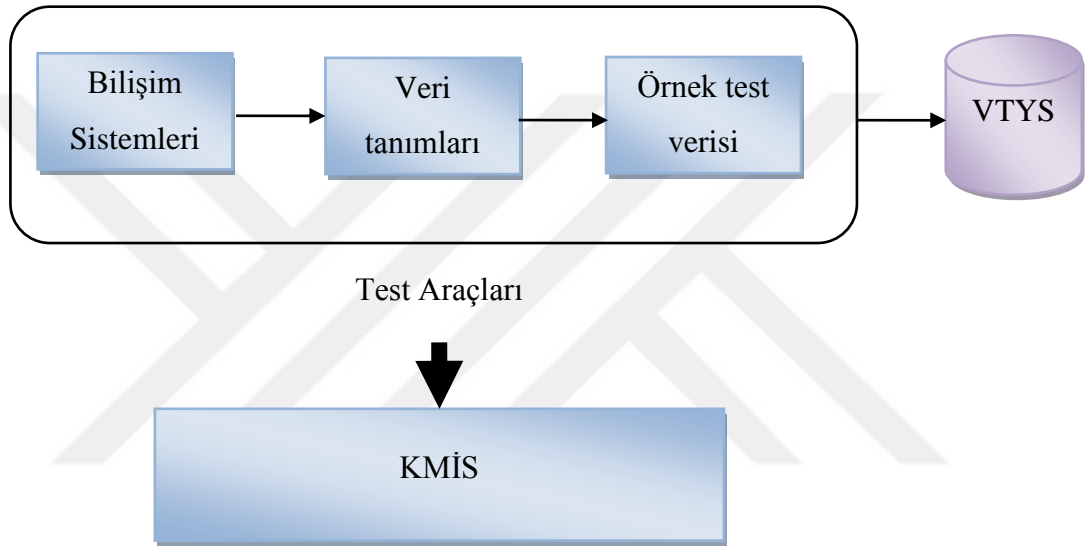
Şekil 3.23'de görüldüğü üzere, kullanıcı her bir anomalinin ayrıntılı bilgisini görüntüleyebilmektedir. Bunun için ilgili anomalinin solunda bulunan radyo düğmesini işaretleyerek ayrıntılı bilgi olarak anomalinin uygulama tarafından oluşturulan kısa bir açıklamasını ve teknik detayını görüntüleyebilmektedir. Açıklamada anomalinin seviyesi, adı, grafik türü, tarihi ve kaç birim uzaklaşma olduğu belirtilmektedir. Teknik detayda anomalinin gerçekleştiği VTYS, veri tabanının adı, tablonun adı, uygulama yöneticisi tarafında oluşturulan tablo erişim adı, tablo işlem adı ve veri tabanı bağlantı adı belirtilmektedir.

3.1.6. Model Testi, Başarı Ölçümü ve Değerlendirme

Modelin “ne derece var olduğuna” yönelik derinlemesine bilgi edinmek için nicel araştırma yapılmıştır (Karataş, 2015: 76; Yıldırım ve Şimşek, 2008: 255). Bu doğrultuda gerekli sayısal sonuçların alınmasında etkili olan prototip yazılım testleri ve model başarı ölçümleri gerçekleştirilmiştir.

3.1.6.1. VTYS’ler ile Yapılan Test

Bu test kapsamında veri kaynağı olarak örnek test verisi kullanılmıştır.



Şekil 3.24. KMİS’in test edilmesi.

KMİS üzerinde gerçekleştirilecek işlemler için Şekil 3.24’de yer alan model kullanılarak örnek test verisi oluşturulmuştur. Örnek test verilerinin tanımlanması amacıyla açık kaynak kodlu bilişim sistemleri referans alınmıştır. Üniversitelerde kullanılan öğrenci bilgi sistemini temsil etmesi amacıyla OpenSIS, muhasebe bilgi sistemi için Front Accounting ve ders programı hazırlama için Float gibi sistemler incelenmiştir. Bu sistemlerdeki veri tanımları doğrultusunda veri prototipleme yöntemi ile örnek test verisi oluşturulmuştur. Oluşturulan örnek test verileri VTYS veri tabanına kaydedilmiştir. Böylece, örnek test verisi ile tasarlanan KMİS’in testi gerçekleştirilmiştir.

Tablo 3.10. Görev Başlıklarına Uygulanan Örnek Veri Kuralları.

Görev Başlıkları	İlgili Sistem	Tablo	Tablo Alanı	Veri Filtresi
Bilimsel araştırmalar	Araştırma	Yayımlar	Satır sayısı	indexli='E'
Öğrenci sayıları	Öğrenci	Ogrenciler	Satır sayısı	Durum='devam_eden'
Kadro sayıları	Personel	Kadro	Satır Sayısı	Durum='calisan'
Sınavlar	Öğrenci	Sınavlar	Satır sayısı	Turu='final'

Veri prototiplemesi için gerekli veri kuralları görev başlıklarına göre Tablo 3.10'da belirtilmiştir. Görev başlıkları üniversitelerde rektör, rektör yardımcıları, dekan, enstitü ve yüksekokul müdürleri, bölüm başkanı ve temsilcilerin görev tanımları doğrultusunda oluşturulmuştur. Her bir görev başlığı için veri setinin elde edilebilmesi için veri akış süreci belirtilmektedir. İlk aşamada, verinin hangi bilişim sisteminden toplandığı belirtilmektedir. İkinci aşamada; bilişim sisteminin kullandığı VTYS veri tabanında ilgili başlık verisinin kullandığı tablo verilmektedir. Üçüncü aşamada; tablo içerisinde hangi alanın kullanılacağı “Tüm satırlar mı?”, “Belirli bir sütunun verileri mi?” yer almaktadır. Son olarak, tablo verilerine kullanıcının ihtiyacına göre filtre kuralı belirtilmiştir.

Tablo 3.10'da görev başlıklarına uygulanan örnek veri kuralları için gerekli olan SQL komutu Tablo 3.11'de verilmektedir. Tabloda belirtilen SQL komutları sistem tarafından ilgili tablolarda değişiklik olması durumunda tablonun bulunduğu VTYS tarafından tetiklenmektedir. Elde edilen veriye zaman etiketi eklenerek görev başlığının veri setine kaydedilmektedir. Elde edilen veri seti analiz çalışmalarında kullanılmaktadır.

Tablo 3.11. Veri kurallarının SQL ile uygulanması.

Görev Başlıkları	SQL komutu
Bilimsel araştırmalar	Select count(*) from arastirma.yayimlar where indexli='E'
Öğrenci sayıları	Select count(*) from ogrenci.ogrenciler where durum='devam_eden'
Personel sayıları	Select count(*) from personel.kadro where durum='calisan'
Sınavlar	Select count(*) from ogrenci.sinavlar where turu='final'

Şekil 3.25’de SQL komutunun sisteme giriş ekranı görülmektedir. Daha farklı filtreleme seçeneği kullanılabilmesi amacıyla tablonun tüm alanlarının adları ve veri türleri sol tarafta veri türüne göre gruplayarak listelemektedir. Elde edilen SQL dizilimi istenilen işlem adıyla kaydedilmektedir. Veri toplama yöntemi olarak tetikleyici seçilirse veri tabanı tablosu üzerinde bir işlem gerçekleştirildiğinde belirtilen SQL komutunun çalıştırarak o andaki toplam indexli yayım sayısını veri setine kaydetmektedir. Sorgulama yöntemi seçildiğinde, ilgili tablo VTYS’ye bağlantı kurma sureti ile veri toplama işlemi gerçekleştirilmektedir. Toplanan verinin özelliğine göre ve VTYS performansının olumsuz etkilenmemesi amacıyla bağlantı kurma sıklığı dakikalık, saatlik ve günlük olarak ayarlanabilmektedir. Veri dosyası kullanımı Excel veri formatına sahip harici veri kaynaklarının aktarımı için düşünülmüştür.

Veri Toplama İşlemi

[Anasayfa](#)
[Oturumu Kapat](#)

Erişim Adı Indexli Yayınlar

Alanlar

DATE	YAYIM_TARIHI
NUMBER	BOL_NO, CILT, FAK_NO, ID, SAYI
VARCHAR2	BASLIK, INDEXLI, INDEX_ADI, LINK, ODEME, SAYFA, TURU, YAYIMCI, YAZARLAR

Yöntem

Tetikleyici ekleme
 Sorgulama
 Veri dosyası

Güncelleme

Dakikalık Saatlik Günlük

İşlem Adı Toplam indexli yayınlar

Lütfen, gruplama fonksiyonu içeren sorgu yazınız.

```
SELECT COUNT(*) SAYI FROM
ARASTIRMA.YAYIMLAR
```

Anlık Veri İzleme ve Kayıt Penceresi

Şekil 3.25. SQL komutunun prototip uygulamaya girilmesi.

Örnek veri kuralları (Bkz. Tablo 3.10), önce SQL komutuna dönüştürülmekte sonrasında ise SQL komutu sisteme girilmesiyle istenilen özet verinin ilgili tablodan izlenmesi sağlanmaktadır.

3.1.6.2. Harici Veri Kaynağı ile Yapılan Test

İkinci uygulama testinde üniversite dışı veri kaynaklarına erişim test edilmektedir. Yükseköğretim Kurumu ve üniversiteler kendi izni doğrultusunda birçok paydaşı ile bazı verileri paylaşabilmektedir. KMİS harici veri kaynağı olarak bu tür verileri kullanabilmektedir. Bu amaçla üniversiteleri ilgilendiren harici veri kaynaklarının KMİS tarafından kullanımı test edilmiştir. Bu kapsamda araştırmanın gerçekleştirdiği zaman aralığında istatistik.yok.gov.tr adresinde paylaşılan günlük olarak güncellenen akademisyen sayısı verisi ele alınmıştır (Yükseköğretim Kurumu, 2019).



Üniversite Türü	Profesör			Doçent			Doktor Öğretim Üyesi		
	E	K	T	E	K	T	E	K	T
DEVLET	16735	7914	24649	9125	5882	15007	18753	13964	32717
VAKIF	3125	1497	4622	1087	930	2017	4199	4380	8579
VAKIF_MYO	5	1	6	0	0	0	6	11	17
TOPLAM	19865	9412	29277	10212	6812	17024	22958	18355	41313

Şekil 3.26. Yükseköğretim Kurumu sistemindeki akademisyen sayısı.

Şekil 3.26’da görüldüğü üzere, Yükseköğretim Kurumu istatistik sayfasına giriş yapıldığında üst menüde mevcut akademisyen sayısı gösterilmektedir. Bu veri günlük olarak güncellendiğinden test için de uygundur. Sağlanan günlük veri içerisinde toplam akademisyen sayısının yanı sıra devlet, vakıf ve vakıf meslek yüksekokulu olmak üzere üç kategoride üniversite türünde sayılar verilmektedir. Bu kategorik sayılar da ileriki çalışmalarda incelenebilir.



YÜKSEKÖĞRETİM BİLGİ YÖNETİM SİSTEMİ

Öğrenci İstatistikleri ▾ Öğretim Elemanı İstatistikleri ▾ Birim İstatistikleri ▾ Yükseköğretim İstatistikleri ▾

Özet Öğretim Elemanı Sayıları Raporu

Excel Çıktısı Pdf Çıktısı (12.10.2020 tarihli verilerdir.)

Üniversite Türü	Profesör			Doçent			Doktor Öğretim Üyesi		
	E	K	T	E	K	T	E	K	T
DEVLET	16741	7915	24656	9132	5888	15020	18755	13964	32719
VAKIF	3125	1497	4622	1088	929	2017	4194	4382	8576
VAKIF_MYO	5	1	6	0	0	0	6	11	17
TOPLAM	19871	9413	29284	10220	6817	17037	22955	18357	41312

Şekil 3.27. Yükseköğretim Kurumu sistemindeki akademisyen sayısındaki değişim.

Şekil 3.27’de Yükseköğretim Kurumu sistemindeki akademisyen sayısının farklı bir zamanda oluşan değeri gösterilmektedir. Bu şekilde de görüleceği üzere aktif akademisyen sayısı değişken bir veridir.

Harici veri kaynağını prototip uygulamaya eklemek için VTYS’lerde de olduğu gibi veri kaynağının adresi ya da veri dosyasının bulunduğu dizin yolu girilmektedir. Toplam akademisyen sayısı ile ilgili veriler Yükseköğretim Kurumu sayfasından Excel dosyası formatında erişilebilmektedir. VTYS’lerden farklı olarak harici veri paylaşımına açık olduğundan herhangi bir şifre girilme zorunluluğu bulunmamaktadır. Bu örnekte veri dosyasının isminin girilmesi yeterlidir.

YÖK		Anasayfa Oturumu Kapat
VTYS		Harici
IP/URL Adresi		https://istatistik.yok.gov.tr
Sistem Şifresi		
Vtys Bağlantı Değişkeni	<input type="text" value="\$sonuc=sorgu;"/>	
Veri Tabanlarını Listeleme Sorgusu	<input type="text" value="array('ISTATISTIK'=>'0')"/>	
Veri Tabanı Tablolarını Listeleme Sorgusu	<input type="text" value="PHPExcel_IOFactory::load('Özet Akademisyen Sayıları.xls')"/>	
Tablo Alanlarını Listeleme Sorgusu	<pre> \$objWorksheet = \$sonuc->getActiveSheet(); \$satir_say=0; foreach (\$objWorksheet->getRowIterator() as \$row) { \$satir_say++; \$sutun_say='A'; \$cellIterator = \$row->getCellIterator(); \$cellIterator->setIterateOnlyExistingCells(false); foreach (\$cellIterator as \$cell) { if(\$sutun_say==\$sutun&&\$satir_say==\$satir) { \$deger=\$cell->getValue(); } } </pre>	

Şekil 3.28. Harici veri kaynağı eklenmesi.

Şekil 3.28’de görüldüğü üzere, Yükseköğretim Kurumu akademisyen sayısı gerçek zamanlı veri dosyasının ismi Özet Akademisyen Sayıları.xls olarak sisteme eklenmiştir.

Tablo Erişimi				Anasayfa	Oturumu Kapat
	Bağlantı Adı	Veri Tabanı	Tablo	Erişim Adı	
Ekle	YÖK	ISTATISTIK	Kayıt Alan	Akademisyen Sayısı	
Mevcut Erişimler					
	Bağlantı Adı	Veri Tabanı	Tablo	Erişim Adı	
Sil	Döviz Kuru	BANKA	Kod[Alis]	Döviz Alış Kuru	
Sil	Döviz Kuru	BANKA	Kod[Satis]	Döviz Satış Kuru	
Sil	Microsoft SQL Veri Tabanı	ZİYARETCILER	GELEN	Gelen Ziyaretçi	
Sil	Atauni Öbs	ÖBS	TOPLAM	Öbs Toplam Kullanıcı	
Sil	MySQL	muhasabe	butce	Bütçe Durumu	
Sil	Atauni Avesis	AVESIS	TopicName[PublicationCount]	Avesis Akademik Veri	
Sil	Oracle Veri Tabanı	ARASTIRMA	YAYIMLAR	İndexli Yayimler	

Şekil 3.29. Yükseköğretim Kurumu toplam akademisyen sayısı parametresine erişimin sağlanması.

Yükseköğretim Kurumu sistemdeki toplam akademisyen sayısı bu veriyi oluşturan parametrelerden biridir. Aktif olarak devlet, vakıf, vakıf meslek yüksekokulu sayıları aynı anda sağlanan diğer verilerdir. Gerçek zamanlı veri analizi için toplam akademisyen sayısı kullanılacağından tablo adı olarak “TOPLAM” parametresinin izlenmesi için sisteme eklenir.

Veri Toplama İşlemi

[Anasayfa](#)
[Oturumu Kapat](#)

Erişim Adı

Lütfen, ilgili alanın adını yazınız.

U9

Alanlar	D6:16741	E6:7915	F6:24656	G6:9132
	H6:5888	I6:15020	J6:18755	K6:13964
	L6:32719	M6:16112	N6:14219	O6:30331
	P6:22918	Q6:23214	R6:46132	S6:83658
	T6:65200	U6:148858	D7:3127	E7:1497
	F7:4624	G7:1088	H7:930	I7:2018
	J7:4197	K7:4383	L7:8580	M7:2410
	N7:4540	O7:6950	P7:1831	Q7:2860
	R7:4691	S7:12653	T7:14210	U7:26863
	D8:5	E8:1	F8:6	G8:0
	H8:0	I8:0	J8:6	K8:11
	L8:17	M8:85	N8:153	O8:238
	P8:0	Q8:0	R8:0	S8:96
	T8:165	U8:261	D9:19873	E9:9413
	F9:29286	G9:10220	H9:6818	I9:17038
	J9:22958	K9:18358	L9:41316	M9:18607
	N9:18912	O9:37519	P9:24749	Q9:26074
	R9:50823	S9:96407	T9:79575	U9:175982

Yöntem Tetikleyici ekleme Sorgulama Veri dosyası

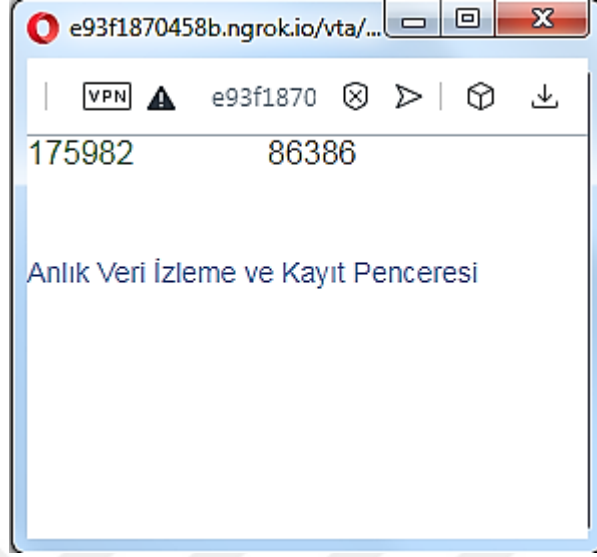
Güncelleme Dakikalık Saatlik Günlük

İşlem Adı

Şekil 3.30. Yükseköğretim Kurumu toplam akademisyen sayısı verisinin toplanma işleminin girilmesi.

Yükseköğretim Kurumu toplam akademisyen sayısı ile ilgili prototip uygulamaya veri dosyasının adı ve veri parametresi girildikten sonra veri toplama işlemine başlanabilir. Şekil 3.30’da görüldüğü üzere, bunun için veri işlem adının girilmesi yeterlidir. VTYS’lerde olduğu gibi veri toplama için SQL sorgusu kullanılmasına gerek yoktur. Harici verilerde SQL sorgusuna gerek kalmadan istenilen verilere erişilebilmektedir. Bunun nedeni veri sağlayıcısının kendi veri tabanını sorgulamaya

izin vermemesidir. Harici veriler kullanıma hazır olarak paylaşılma açık olarak gerçek zamanlı yayımlanmakta ya da veri dosyası içerisinde yer almaktadır.



Şekil 3.31. Yükseköğretim Kurumu toplam akademisyen sayısı verisinin izlenmesi.

Şekil 3.31’de Yükseköğretim Kurumu toplam akademisyen sayısı internet tarayıcısında görüntülenebilmektedir. Burada beliren veri için veri toplama sıklığı günlük olarak ayarlanmış olup veriler KMİS veri tabanına kaydedilmektedir.

ANLIK VERİ İZLEME VE KAYIT

Veri	Değer	Güncelleme kalan saniye
Toplam Akademisyen Sayısı	175984	6731
Gelen Ziyaretçi Sayısı	30	594
Usd Alış Kuru	7.872	7
Öbs Toplam Kullanıcı Sayısı	1871	3157
Avesis Toplam Yayınlar	115735	6734
Gbp/Usd Alış Kuru	1.305	24
Usd Satış Kuru	7.918	7
Bütçe gelir-gider farkı	85600	6743
Toplam indexli yayımlar	5002	6703

Şekil 3.32. Yükseköğretim Kurumu toplam akademisyen sayısı verisinin kaydı.

Yükseköğretim Kurumu toplam akademisyen sayısının kullanıcılar tarafından çok fazla teknik detaya girmeden erişebilmesi için Şekil 3.32’de görüldüğü üzere, veri toplama işlem adı veri adı olarak kullanılmaktadır. Veri adı aynı zamanda kullanıcı tarafından grafik adı olarak kullanılmaktadır. Böylece eklenen veri diğerleri ile birlikte veri izleme ve kayıt penceresinde görüntülenmektedir.

3.1.6.3. Model Başarısının Ölçümü

Modelin nicel olarak başarısının ölçümü için karmaşıklık matrisi kullanılarak istatistiki hesaplamalar yapılmıştır. Karmaşıklık matrisi değerleri KMİS tarafından tespit edilen anomaliler ile oluşturulan veri setinden elde edilmiştir. Anomali tespiti

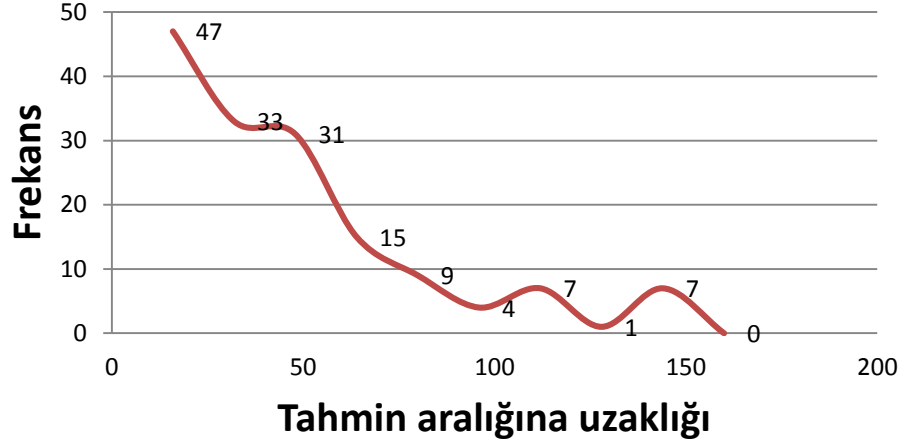
2019 Eylül ile 2020 Mayıs ayları arasında Google Trendler “öğrenci bilgi sistemi” teriminin Türkiye’de gerçekleştirilen web arama yüzdeler oranı değerleri üzerinde gerçekleştirilmiştir. Zaman aralığına göre dakikalıkta 880, saatlikte 154 ve günlükte 57 olmak üzere toplam 1091 anomali tespiti uygulama tarafından yapılmıştır.



Şekil 3.33. Dakikalık zaman aralığı düzeyinde anomalilerin tahmin aralığına olan birim mesafesine göre frekansı.

Şekil 3.33’de zaman aralığı olarak dakikalık düzeyde tespit edilen anomalilerin tahmin aralığından uzaklaşma mesafesine göre frekans grafiği gösterilmektedir. Şekilde tahmin uzaklığının 0 ile 20 arasındaki frekans seviyesi yüksektir. Bu ise tahmin aralığına yakın mesafede bulunan anomalilerin sayısının fazla olduğunu göstermektedir. Tahmin aralığından uzaklaştıkça anomalilerin sayısı azalmaktadır.

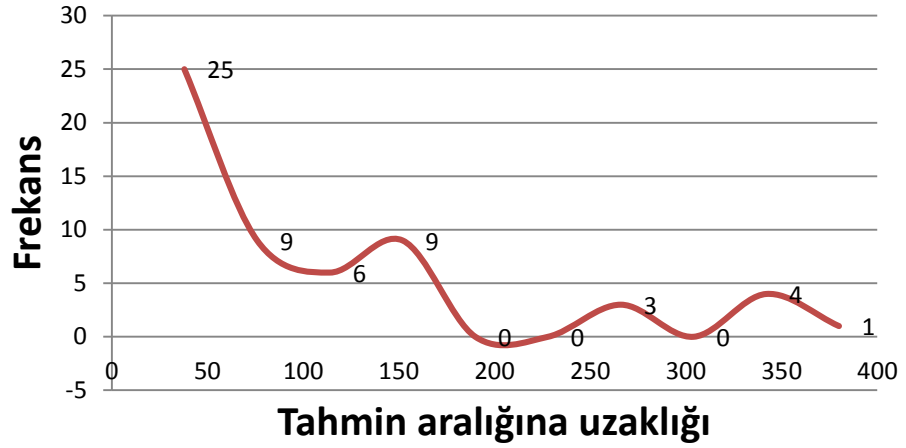
Saatlik Zaman Aralığı



Şekil 3.34. Saatlik zaman aralığı düzeyinde anomalilerin tahmin aralığına olan birim mesafesine göre frekansı.

Şekil 3.34'de zaman aralığı olarak saatlik düzeyde tespit edilen anomalilerin tahmin aralığından uzaklaşma mesafesine göre frekans grafiği gösterilmektedir. Şekilde de görüleceği üzere tespit edilen anomalilerin tahmin aralığından uzaklaştıkça kademeli olarak frekansının da azaldığını göstermektedir.

Günlük Zaman Aralığı



Şekil 3.35. Günlük zaman aralığı düzeyinde anomalilerin tahmin aralığına olan birim mesafesine göre frekansı.

Şekil 3.35’de zaman aralığı olarak günlük düzeyde tespit edilen anomalilerin tahmin aralığından uzaklaşma mesafelerinin frekans grafiği gösterilmektedir. Şekilde görüleceği üzere bir önceki grafik olan saatlik düzey grafik ile benzerlik göstermektedir. Bu grafikte de tespit edilen anomalilerin tahmin aralığına olan uzaklığı arttıkça anomalilerin frekansı azalmakta, ancak gerçekte daha anlamlı sonuçlar vermektedir. Bu nedenle, uygulamanın başarı testi için kullanılacak olan örneklem veri seti dakikalık, saatlik ve günlük düzeyde uzak mesafede bulunan veriler kullanılmıştır (Bkz. Ek 6). Bu doğrultuda prototip uygulamadan elde edilen verilere göre Tablo 3.12’de verilen karmaşıklık matrisi oluşturulmuştur.

Tablo 3.12. Prototip uygulama verilerine göre karmaşıklık matrisi.

		Tespit Edilen	
		Hayır	Evet
Gerçekte	Hayır	6 Yanlış Negatif (YN)	4 Yanlış Pozitif (YP)
	Evet	78 Doğru Negatif (DN)	24 Doğru Pozitif (DP)

Tablo 3.12’de tespit edilen 112 adet anomali içerisinde 24+78 adet anomali gerçekte doğrudur ve bunlar doğru pozitifler ile doğru negatiflerin toplamıdır. 4+6 adet anomali gerçekte yanlıştır ve bunlar yanlış pozitifler ve yanlış negatiflerin toplamıdır. Karmaşıklık matrisi verilerine göre uygulamaya yönelik anomali tespitinin başarısının ölçümü doğruluk, duyarlılık, kesinlik, F-Ölçütü, Matthews Correlation Coefficient (MCC) ve ROC eğrisi yöntemleri kullanılarak yapılmıştır.

Doğruluk, doğru olarak tespit edilen anomalilerin ve normal verilerin sayısının toplamının tüm tespit sayısına oranıdır. Hesaplama kullanılan formül şu şekildedir:

$$\text{Doğruluk} = \frac{DP + DN}{DP + YN + YP + DN} = \frac{24 + 78}{24 + 6 + 4 + 78} = 0.91 \quad (1)$$

Duyarlılık, uygulamanın verileri anomali olarak etiketlemedeki başarısını göstermektedir. Doğru olarak tespit edilen anomalilerin gerçekte anomali olarak etiketlenmesi gerekenlere oranıdır. Hesaplama kullanılan formül ve hesaplama aşağıdaki gibidir.

$$\text{Duyarlılık} = \frac{DP}{DP + YN} = \frac{24}{23 + 6} = 0.8 \quad (2)$$

Kesinlik, uygulamanın tespit ettiği anomalilerin ne derece doğru olduğunun göstergesidir. Kesinliğin hesaplanmasında kullanılan formül ve hesaplama aşağıda verilmektedir.

$$\text{Kesinlik} = \frac{DP}{DP + YP} = \frac{24}{24 + 4} = 0.85 \quad (3)$$

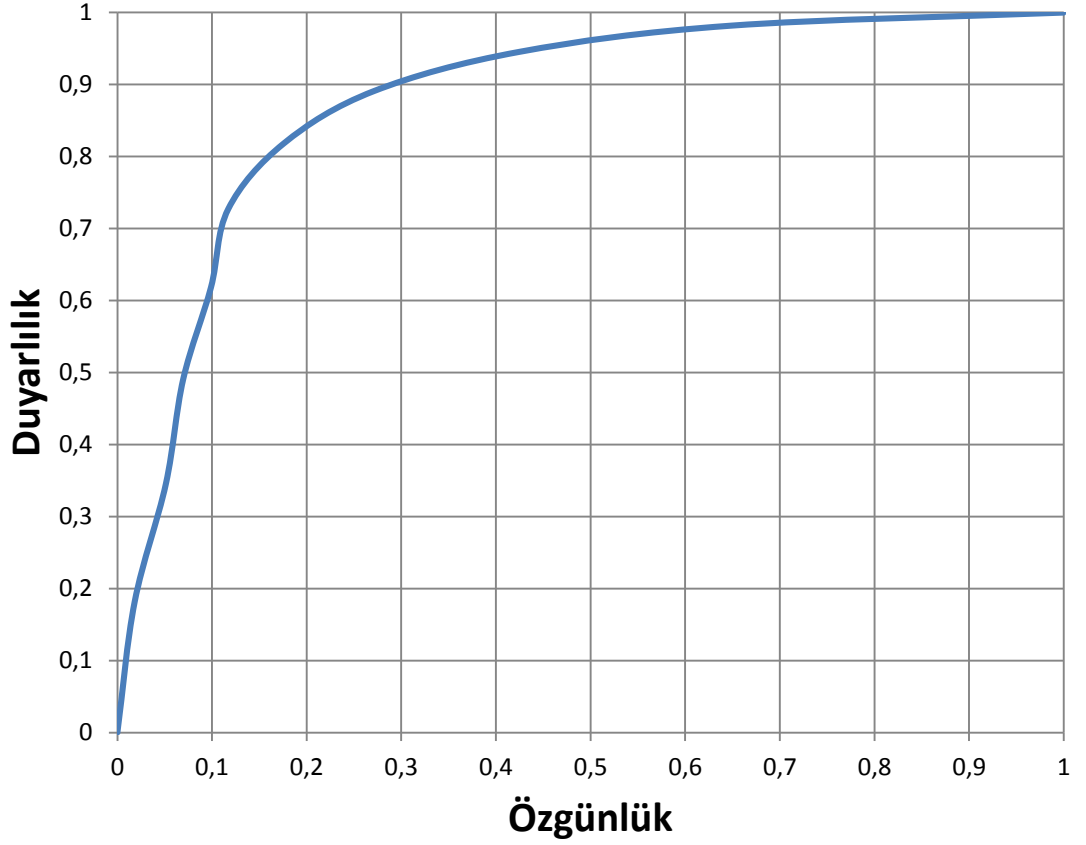
F-Ölçütü, duyarlılık ve kesinlik ölçümlerinin harmonik ortalamasıdır. Böylece bu iki ölçüm tek bir hesaplamada birleştirilmektedir.

$$F - \text{Ölçütü} = 2x \frac{\text{Kesinlik} \times \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} = 0.82 \quad (4)$$

Matthews Correlation Coefficient (MCC) değerinin hesaplanmasında kullanılan formül ve hesaplama aşağıda verilmektedir.

$$\text{MCC} = \frac{(DP \times DN) - (YP \times YN)}{\sqrt{(DP + YP) \times (DP + YN) \times (DN + YP) \times (DN + YN)}} = 0.77 \quad (5)$$

ROC Eğrisi (Receiver Operating Characteristic Curve - Alıcı İşletim Karakteristik Eğrisi), duyarlılık ve özgünlük hesaplaması sonucu elde değerlerinin x ve y eksenlerine yerleştirilmesi sonucu ortaya çıkan eğri analiz edilmektedir. Duyarlılık, bir önceki hesaplamalarda yapıldığı üzere doğru pozitif oranının hesaplanması ile özgünlük, doğru negatif oranının hesaplanması elde edilmektedir.



Şekil 3.36. Uygulama anomali tespiti ROC eğrisinin gösterimi.

Şekil 3.36’da uygulama anomali tespitinin değerlerine göre ROC eğrisi görülmektedir. Değerler, (Bkz. Şekil 3.33, Bkz. Şekil 3.34 ve Bkz. Şekil 3.35) anomalilerin tahmin aralığından uzaklaşma mesafesine göre frekansı kullanılmıştır. Uygulamada anomalilerin uzaklaşma mesafesi olarak gösterilen özgünlük (doğru negatiflerin oranı) grafikte x eksenini ile gösterilmektedir. Anomalilerin frekansı olan duyarlılık (doğru pozitiflerin oranı) grafikte y eksenini ile gösterilmektedir. Uygulamanın başarısı uzaklaşma mesafesi ve frekans arasındaki ilişki ile doğru orantılıdır. ROC eğrisi, uygulamanın bunu sağlamadaki başarısını grafik olarak göstermektedir. Eğri çizgisinin altında kalan alan uygulamanın başarısını göstermektedir. Eğri altında kalan alan arttıkça uygulama ya da modelin başarısının da arttığını göstermektedir. Şekil 3.36’da görüldüğü üzere, eğri grafiğin sol üst köşesine doğru kavis çizdiğinden eğri altında kalan alan daha fazladır.

3.1.6.4. Model Değerlendirmeleri

Modelin “varlığına ve anlamına” yönelik derinlemesine bilgi edinmek için nitel araştırma yapılmıştır (Karataş, 2015: 76). Bir konu hakkında oldukça fazla bilgiye sahip olan bireylerin görüşlerinin alınmasında etkili olan amaçlı örnekleme yöntemi kullanılmıştır (Yıldırım ve Şimşek, 2008: 107). Bu doğrultuda üniversitelerde farklı birimlerde yönetici konumunda bulunmuş emekli akademisyenlerin görüşlerine başvurulmuştur. Farklı düzeyde görev yapmış olmasına özellikle dikkat edilmiştir.

Modelin değerlendirilmesine ilişkin olarak zengin bilgi kaynağı olabilecek çalışma grubunun (örneklem) büyüklüğünün belirlenmesinde kartopu örnekleme yöntemi kullanılmıştır. Bu yöntemde ilk başta birçok farklı cevaplar ortaya çıkmıştır, ancak bir süre sonra bazı cevaplar tekrarlamaya başladığında yeterli örneklem büyüklüğüne ulaşıldığı kanaatine varılmıştır.

Araştırmanın geçerliliğine katkı sağlaması amacıyla esnek bir yapıya sahip olan yarı yapılandırılmış anket formu uygulanmıştır (Bkz. Ek 7). Uygulama geliştirme alanında deneyimli ve uzman kişilere inceletilerek, meslektaşların görüşü alınarak ve katılımcının onayı da alınarak, güvenilirliği önceden denenerek hazırlanmış anket formu kullanılmıştır. Anket formunda sorular kolay ve anlaşılır olmasına dikkat edilerek açık uçlu sorular sorulmuştur. Katılımcılardan demografik bilgi olarak cinsiyet, görev yılı ve görevi talep edilmiştir. Bu konuda ilgili literatür incelenmiş ve bilimsel araştırmalarda sakınılacak ve dikkat edilecek konulara özen gösterilmiştir. Model değerlendirme soruları yönlendirilme olmadan, rahatsız ve rencide etmeyecek şekilde bir konuya odaklı hazırlanmıştır. Hazırlanan anket formundaki sorular aşağıda belirtilmiştir.

1. Yöneticiler açısından bu tür bir uygulama modeli fikri ne gibi faydalar sağlayabilir?
2. Uygulama modeli fikri olarak eksik bulduğunuz yönler nelerdir?
3. Uygulama modeli fikri olarak belirttiğiniz eksikliklere çözüm olarak önerileriniz nelerdir?

Araştırmaya katılan 32 yönetici (27 erkek, 5 kadın) bulunmaktadır. Araştırma gurubuna ait demografik bilgiler EK 8’de verilmiştir. EK 8’de görüldüğü gibi katılımcıların çoğunluğu farklı birim ve görevlerde yer almaktadır. Katılımcıların

cinsiyet verilerine göre çoğunluğu erkektir ve görev yılı 24 ile 35 arasında bulunmaktadır.

Güvenirliliğinde sağlanması için araştırmaya katılan katılımcıların demografik bilgileri açık bir şekilde tanımlanmıştır. Araştırma grubundaki katılımcılar emekli akademisyen olduklarından anket formu elektronik ortamda hazırlanmış olup e-posta yoluyla erişim linki ve gerekli açıklamalar yapılarak her katılımcıya ayrı ayrı gönderilmiştir. Soruların samimi cevaplanması için kişisel bilgilerin gizli tutulacağı ve kodlama yapılarak kullanılacağı açıklaması yapılmıştır. Her bir katılımcı Y1, Y2, ... şeklinde kodlanmıştır. Anket formu elektronik ortamda doldurulduğundan katılımcıların soruları yanıtlamalarında araştırmacı tarafından herhangi bir ek yorum yapma gibi müdahale söz konusu değildir. Katılımcılar formu doldurduktan sonra katılımcıların anlamakta güçlük çektiği konular ve sorular daha ayrıntılı açıklanarak bir sonraki katılımcılar tarafından da daha iyi anlaşılması sağlanmıştır. Aynı şekilde bazı katılımcılar formu doldurmadan önce daha geniş bir açıklama talebinde bulunmuş olup detaylı bir şekilde araştırma hakkında bilgi verilmiştir. Araştırmaya ait veriler tüm yarı yapılandırılmış anket formları ve üzerindeki kodlamalar incelenebilecek şekilde elektronik ortamdan kâğıda basılmış ve araştırmacı tarafından saklanmaktadır.

3.1.7. Modelin Sınırlılıkları

Modelin sınırlılıkları tez konusunun kapsamı, veri kaynakları, veri işleme ve analiz süreçleri doğrultusunda ele alınmıştır.

Bu araştırma mevcut bir uygulamanın incelenmesi ya da proje çalışması olmayıp yalnızca yeni bir uygulama modeli geliştirmeye yöneliktir. Bu kapsamda, VTYS ve harici veriler üzerinden kurumsal özet verileri toplayıp analiz edebilen prototip bir uygulama geliştirme çalışması yapılmıştır.

Kullanılan veri kaynakları VTYS ve harici veridir. Çalışmada VTYS olarak Oracle, MySQL, PostgreSQL ve Microsoft SQL Server kullanılmıştır. Ancak, prototip uygulamaya yeni bir VTYS eklemek mümkündür.

Uygulama zaman serilerine verilerini kullandığından yalnızca zaman etiketine sahip sayısal verileri kullanılabilir. Metin, resim, ses, video, vb. veri türlerini desteklememektedir. Veri analizi de yalnızca zamansal boyutuyla incelenebilmektedir.

Verilerin zaman boyutu haricinde farklı boyutla incelenebilmesi için seçenek bulunmamaktadır.

Modelde uygulamanın işlevselliği ön planda tutulmuştur. Bu nedenle, performans ve arayüz görselliği arka planda kalmıştır. Görsel programlama ile daha profesyonel bir web arayüzü geliştirilebilir.

Kurumsal boyutta yazılım geliştirme için şüphesiz bir ekip çalışması gereklidir. Ancak bu çalışmada yeni bir modelin temel yapı taşlarını ortaya koymak amacıyla bir yazılım geliştirilmiştir. Ortaya çıkan yazılım bir paket yazılım niteliğinde olmayıp, açık kaynak kodlu, araştırma ve geliştirmeye yönelik bir prototiplemedir. Yazılımı kullanmak isteyen üniversiteler tasarımı örnek alarak kendi sistemleri ve ihtiyaçları doğrultusunda yazılım ekibi ile daha gelişmiş bir yazılım ürünü ortaya koyabilir.

3.1.8. Modelin Varsayımları

Araştırmanın uygulama alanına yönelik olarak gerçekleştirilmesinde bazı varsayımlarda bulunulmuştur. Bunlar:

- Gerçek zamanlı veri akış analizinin üniversiteler tarafından da kullanılabilceği
- Bilişim sistemlerine verilerin tutarlı bir şekilde girildiği
- Kullanılan bilişim sistemlerinin VTYS'ye sahip olduğu
- Kurumun en az bir VTYS uzmanına sahip olduğu
- Yöneticinin talebine göre veri tabanlarında izlenecek olan tabloların düzenli bir şekilde seçildiği
- Prototip uygulamanın kurum içi sunucu bilgisayarında çalıştırıldığı varsayılmıştır.

Kullanıcıların sistemi ilk defa kullanmaya başladıklarında sistem hakkında bir fikir sahibi olabilmeleri amacıyla varsayılan değerler kullanılmıştır. Bu değerler daha sonra kullanıcının isteğine göre özelleştirilebilen uygulama başlangıç değerleridir. KMİS'in başlangıç değerleri:

- Anomali seviyelerinin rakamsal karşılığı sırasıyla 1 (daha az değerli), 2 (değerli) ve 3 (daha çok değerli)'tür. Uygulamanın ilk çalışma anındaki seviye değeri 3'dir.
- Veri toplama sıklığı ve grafik zaman aralığı değerleri dakikalık, saatlik ve gündüktür.
- Veri kaynağına ulaşılama durumunda tekrar deneme süresi 3 saniyedir.

Zaman aralığı olarak dakikalık, saatlik ve günlük seçilmesinin sebebi SQL sorgulamalarında verileri bu zaman aralıklarında gruplama işlemini daha basit olmasıdır. Örneğin, 10/03/2020 14:10:20 zaman etiketi 202003101410 olarak veri setinde dakikalık, 2020031014 olarak saatlik ve 20200310 olarak günlük düzeyde veriler kolaylıkla gruplanabilmektedir.

Sistem herhangi bir sunucu sistemi üzerine kolaylık kurulabilecek şekilde tasarlanmıştır. KMİS temelde VTYS ve web uygulamasından oluşmaktadır. Uygulamada kullanılan parametrelerin, kullanıcı ayarlarının ve kurum dışından toplanan verilerin kaydedilmesi amacıyla sistem tarafından kullanılan bir veri tabanı bulunmaktadır. KMİS'de Oracle Database 11g Release 2 (11.2.0.1.0) sürümünün Windows 32 bit dosyası kullanılmıştır.

3.2. YAPILAN DİĞER ARAŞTIRMALARDAN FARKI

Yapılan çalışmaların bu araştırmayla ilgili benzerlikleri ve farklılıkları aşağıdaki tabloda gösterilmektedir.

Tablo 3.13. Bu araştırma ile yapılan diğer arařtırmalar arasındaki benzerlikler ve farklar.

Arařtırma ve Uygulama	Amaç	Veri Kaynakları	Kullanım Alanı
Solaimani ve diğ. (2016: 1479)	Sanal makine sunucularını izleme	İřlemci kullanımı, yük durumu, kullanım yođunluđu	Bilgi Teknolojileri
Abuaitah (2013: 3)	Bina iine yerleřtirilen çeřitli sensörleri izleme	Isı, nem, ıřık, gürültü, duman, hareket sensörleri	Enerji tasarrufu ve güvenlik
Guozhang ve diğ., (2015: 1654)	LinkedIn web sayfasının verilerini izleme	LinkedIn	Sosyal medya
Fernandez ve diğ. (2015: 1)	LinkedIn web sayfasının verilerini farklı sistemlere dađıtımı	LinkedIn	Sosyal medya
Kreps, Narkhede ve Rao (2011: 1)	Gerçek zamanlı veri toplama araçlarının performans karşılařtırması	Apache Kafka, Java ActiveMQ v5.4 ve RabbitMQ v2.4	Bilgi teknolojileri
Google Analytics	Web site ziyaretçi sayılarının takibi ve saldırı tespiti	Web site ziyaretleri	Bilgi teknolojileri
Elektronik Posta Servisleri	İzinsiz giriř tespiti	Elektronik posta giriř kayıtları	Bilgi teknolojileri
Bu arařtırma	Anomali tespiti	VTYS'ler ve harici veriler	Üniversite ve üniversitelerde yönetici kararlarında

Tablo 3.13'de bu arařtırmanın yapılan diđer arařtırmalarla benzerlikleri ve farklılıkları dikkate alınarak incelenebilmesi amacıyla özet bilgiler sunulmuřtur. Bu bilgiler arařtırmaların amacı, kullandıđı veri kaynakları ve kullanım alanına göre düzenlenmiřtir.

Yapılan arařtırmalara (Bkz. Tablo 3.13) göre; sanal makine sunucularının performansını iyileřtirme, bina güvenliđinin sađlanması, sosyal ađ verilerinin kaydedilmesi ve bu verilerin dađıtımı, gerçek zamanlı veri toplama araçlarının

performansı ve bir servise izinsiz giriş tespiti gibi amaçlar yer almaktadır. Ayrıca Google Analytics de kullanıcılarının web sitelerinin ziyaretçileriyle ilgili istatistiki bilgileri analiz edebilmelerini sağlayan araçlar geliştirmiştir. Yapılan çalışmalardaki amaçların bütününe bakıldığında belirli ve sınırlı bir konuya yönelik olarak gerçekleştirildiği görülmektedir. Kurumlar açısından belirli bir konu ile ilgilenen veya teknik personelin ve idarecilerin bazı sorunlarına çözüm bulmaya yönelik olduğu anlaşılabilir. Ancak, yapılan bu araştırma üniversitede belirli bir birimin ya da personelin kullanımına yönelik olmayıp herhangi bir sorunun çözümüne yönelik olarak gerçekleştirilmiştir.

Tabloda bu araştırmayı diğer yapılan araştırmalardan ayıran diğer bir özellik, farklı veri kaynaklarının kullanılabilmesidir. Diğer araştırmalarda belirli bir veri kaynağı üzerine yoğunlaşarak analiz çalışmaları yürütülmüştür. Ancak, bu araştırma farklı veri kaynakları ile çalışabilmeyi hedeflemektedir. Bilindiği üzere, üniversitelerde farklı VTYS'ler ve veri kaynaklarının kullanımı söz konusudur. Dolayısıyla, tek bir veri kaynağı ile ilgilenmek belirli ve sınırlı bir konuda bilgi elde etmemizi sağlar. Ancak, farklı veri kaynakları ilgilenildiği takdirde üniversite veya üniversite genelinde özet veri toplamak mümkündür.

Kullanım alanıyla ilgili benzer diğer araştırmalara bakıldığında konular genel olarak bilgi teknolojileri, enerji tasarrufu ve güvenlik gibi teknik altyapıyı destekleyen alanları ilgilendirmektedir. Bu çalışmalar içerisinde sosyal medya takibi ve reklam gibi sosyal ve dış çevreye yönelik kullanım alanı olan çalışmalar da bulunmaktadır. Ancak, bu çalışmalar daha dikkatli incelendiğinde belirli bir veri kaynağı olarak LinkedIn web sayfası kullanıldığı görülmektedir. Sosyal medya takibi ve reklam çalışmalarının yalnızca bir web sayfasının verilerine dayandırılarak gerçekleştirmek uygulamada pek doğru sonuçlar vermeyebilir. Ancak, destekleyici bir bilgi olarak kullanılabilir. Bu araştırmada, her ne kadar çalışma alanı olarak üniversite kurumu seçilmiş olsa da her çeşit kurumu kapsayabilecek bir kullanım alanına sahiptir. Gerek üniversite içi gerekse üniversite dışı veriler araştırmanın ilgi odağı olması nedeniyle geniş bir kullanım alanına sahiptir.

Bu arařtırmayı diđer yapılan alıřmalardan (Bkz. Tablo 3.13) ayıran en nemli zelliđi, geniř bir kullanım alanına sahip olmasıdır. Bu zelliđe sahip olmasında farklı veri kaynaklarını ve herhangi bir amaca ynelik olarak kullanılabilmesi etkili olmuřtur.

3.3. MODELİN İŐ ZEKASI SİSTEMLERİNİN TEMEL ZELLİKLERİ İLE KARŐILAŐTIRILMASI

Bu alıřmada, iŐ zekâsı sistemlerine benzer bir uygulama gerekleřtirmek istenmiřtir. Bu amala, iŐ zekâsı sistemlerinde kullanılan bir uygulamadan beklenen zellikler ile KMİS'in bu zellikleri ne derece karŐıladıđı Tablo 3.14'de ele alınmaktadır.

Tablo 3.14. İŐ zekâsı sistemlerinin temel zellikleri ve KMİS.

İŐ Zekâsı Sistemleri		KMİS		Uyum
zellikleri	Beklenti	Gzlemlenen		
Veri kaynakları	Yazılı, elektronik, zihinsel	Elektronik		Kısmen
Veri tabanı	Var	Var		Uyumlu
Veri Analizi	Veri odaklı veya Veri ambarı	Veri odaklı		Uyumlu
Veri Madenciliđi	Var	Var		Uyumlu
Raporlama, Sorgulama	Var	Var		Uyumlu

Tablo 3.14'de iŐ zekâsı sistemlerinin temel zelliklerine bakıldıđında iŐ zekâsı kapsamındaki uygulamalardan beklentiler ‘‘Beklenti’’ alanında verilmektedir. KMİS'in bu beklentileri karŐılayıp karŐılamadıđı ‘‘Gzlemlenen’’ alanında belirtilmektedir. Elde edilen beklenti ve gzlemlerin karŐılařtırma sonuları ‘‘Uyum’’ bařlıklı alanda yer almaktadır. Bu tabloya gre;

İŐ zekâsı sistemlerindeki uygulamalar niversiteyi ilgilendiren tm verileri toplama abasındadır. Bu veriler yazılı, elektronik ve bazen de zihinsel olabilir. Ancak, KMİS VTYS'lerden ve harici verileri toplama yeteneđine sahip olduđundan yalnızca

elektronik ortamdaki verileri toplayabilmektedir. Bunun sebebi KMİS'in gerçek zamanlı verilere yönelik bir çalışma gerçekleştirmesinden kaynaklanmaktadır. Veri kaynağına gerçek zamanlı olarak erişilebilmesi için verinin genellikle elektronik ortamda üretilmesi gerekmektedir. Dolayısıyla, iş zekâsı sistemleri yazılı, elektronik ve zihinsel veri kaynaklarını kullanabilmesi beklenmektedir. Ancak, prototip uygulama yalnızca elektronik ortamdaki verilere erişebildiğinden kısmen veri kaynağı özelliği yönüyle uyumludur.

Çeşitli veri kaynaklarından toplanan veriler iş zekâsı sistemlerinde kullanılabilmesi için düzenli bir şekilde veri tabanlarına kaydedilmektedir. İş zekâsı sistemleri veri tabanlarını kullanması gerekmektedir. KMİS'de farklı veri kaynaklarından toplanan veriler veri tabanlarına kaydedilmektedir. Bu nedenle, veri tabanına sahip olma özelliği yönüyle iş zekâsı sistemleri ile KMİS arasında uyum bulunmaktadır.

Veri analizi iş zekâsı sistemleri kapsamında veri odaklı ya da veri ambarları vasıtasıyla gerçekleştirilmektedir. Veri odaklı analizlerde veri tabanındaki veriler direk olarak analiz edilmektedir. KMİS'de de buna benzer bir analiz söz konusudur. Veri tabanına kaydedilen zaman parametrelili verilerin analizi gerçekleştirilmektedir. Bu sebeple veri analizi yönüyle iş zekâsı sistemleri ve KMİS arasında uyum söz konusudur.

Gerek veri odaklı gerekse de veri ambarlarına dayalı analizlerdeki amaç değerli bilginin ortaya çıkarılmasıdır. Bu doğrultuda KMİS destek vektör makinesi yöntemi ile anomali tespiti yapmakta ve tespit edilen anomalileri seviyelendirmektedir. Model bu yönü ile iş zekâsı sistemleri ile uyumludur.

Elde edilen sonuçların anlaşılır bir şekilde gerekli kişilere bildirilmesi için raporlama ve sorgulama yapılabilmesi iş zekâsı sistemlerindeki beklentileri arasındadır. KMİS gerek grafiksel olarak gerekse de teknik detayları dikkate alınarak elde edilen sonuçların kullanıcılarına raporlanmasını ve sorgulanmasını sağlamaktadır. Raporlama ve sorgulama özelliği yönüyle iş zekâsı sistemlerinde ve KMİS arasında uyum bulunmaktadır.

Genel olarak, iş zekâsı sistemlerinin temel özellikleriyle KMİS arasındaki bu özelliklerin gözlemlerine bakıldığında uyumlu sonuçlar bulunmaktadır. Yalnızca,

KMİS'in gerçek zamanlı veriler ile ilgilendiğinden dolayı veri kaynağı olarak elektronik verileri kullanmaktadır. Bu nedenle, veri kaynağı ile uyumluluk kısmen bulunmaktadır.

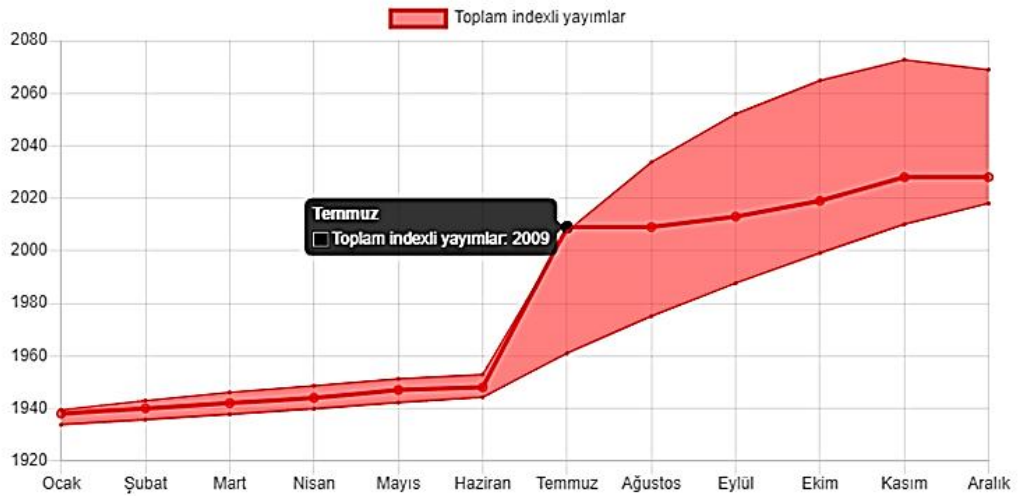
3.4. BULGULAR

3.4.1. Model Testi Bulguları

KMİS üzerinde yapılan testlerden elde edilen bulgular bu alt başlık altında ele alınmaktadır. KMİS'in VTYS'ler ve harici veri kaynakları ile yapılan testler olmak üzere iki farklı test yapılmıştır.

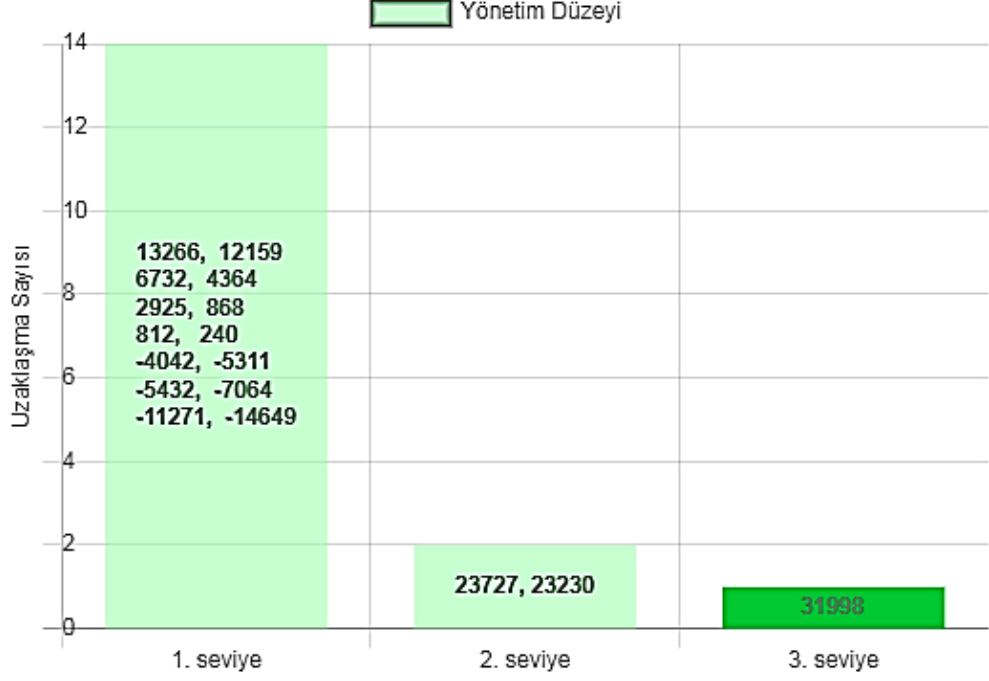
3.4.1.1. VTYS'ler ile Yapılan Test Bulguları

Bu testte, veri tabanı tablolarından özet veri elde edilmesi amaçlanmaktadır. Bu doğrultuda veri tabanı tablolarında meydana gelen herhangi bir değişiklik sonucunda elde edilen özet verinin zaman etiketi ile birlikte kaydedilmesi gerekmektedir. Ortaya çıkan dinamik veri seti üzerinde, destek vektör makinası algoritması kullanılarak anomalilerin tespit edilmesi düşünülmüştür. Algoritma veri seti üzerinde istatistiki hesaplamalara dayalı değişken bir tahmin aralığı oluşturmaktadır. Bu tahmin aralığı verileri kümelemede kullanılmaktadır. Tahmin dışarısında kalan veriler anomali olarak etiketlenmektedir.



Şekil 3.37. Anomali tespiti.

Şekil 3.37’de örnek veri prototiplemesi yolu ile hazırlanmış Ocak ile Aralık ayları arasında veri seti görüntülenmektedir. Veriler seçilen VTYS’ler üzerinde ayrı ayrı uygulanmış ve aynı grafik elde edilmiştir. Taralı alan, veri setinin tahmin aralığıdır. Veri setine eklenecek olan yeni değerlerin tahmin aralığı içerisinde olması beklenmektedir. Taralı alan dışına çıkma durumu beklenti dışı gerçekleştiğinden yeni anomali olarak işaretlenmektedir. Şekilde de görüleceği üzere Ocak ayından başlayıp Mayıs ayına kadar doğrusal değer artışı bulunmaktadır. Tahmin aralığı doğrusal olarak muhtemel hareket alanını işaretlemektedir. Tahmin aralıklarının birbirine yakın olması veri setindeki değişimin normal ve düzgün olduğunu göstermektedir. Şekilde, Temmuz ayında veri setinde ani bir yükseliş Temmuz değerinin tahmin aralığının üzerine çıkmasına sebep olmuştur. Bu nokta anomali olarak sisteme eklenmiştir. Temmuz ve Aralık aylarına bakıldığında ani yükseliş sonrası sakin bir veri değişimi görülmektedir. Ancak, tahmin aralığı birbirinden uzaklaşmıştır. Çünkü vektör makine algoritması Temmuz ayında yaşanan ani değer artışını da göz önünde bulundurarak tahmin aralığını artırmaktadır. Temmuz ayından sonraki tahmin aralığının üst noktalarında yaşanan genişleme tahmin aralığının alt noktalarına oranla daha fazladır. Çünkü Temmuz ayında yaşanan ani artış tahmin aralığının üst limitini aşmıştır.



Şekil 3.38. Anomalilerin üç seviyede sınıflanması.

Anomali tespitinin sonucu zaman etiketine sahip sayısal bir veridir. Bu sayısal veri tahmin aralığından uzaklaşma mesafesidir. Bu elde edilen sonuç verileri ile analiz edilmek üzere anormal verilerinden oluşan veri seti oluşturulmaktadır. Buradaki amaç, tespit edilen anormal verileri önem derecesine göre kategorize edebilmektir. Gözlemlere göre anormal veriler tahmin aralığına yakın olma eğilimi göstermektedirler. Veri ne kadar tahmin aralığından uzak ise bu verinin anormal olma ihtimali o kadar kuvvetli olduğu anlamına gelmektedir. Bu nedenle, bu tip anormal veriler üçüncü seviye olarak etiketlenmektedir. Aralığa daha yakın veriler mesafesine göre ikinci ve birinci seviye olarak etiketlenmektedir (Şekil 3.38).

Açıklama:

Alt düzeyde tanımlı Toplam indexli yayımlar adlı Günlük grafikte 2020-07-07 03:00:00 tarihinde beklenti aralığından 3 birimlik uzaklaşma anomali olarak tesbit edilmiştir.

Teknik Detay:

Veri Tabanı Sistemi	:	Oracle
Veri Tabanı Adı	:	ARASTIRMA
Tablo Adı	:	YAYIMLAR
Tablo Erişim Adı	:	İndexli Yayımlar
Tablo İşlem Adı	:	Toplam indexli yayımlar
Veri Tabanı Bağlantı Adı	:	Oracle Veri Tabanı

Şekil 3.39. Tespit edilen anomalinin ayrıntılı bilgisi.

Anomaliler tespit edilip önem derecesine göre sınıflandırıldıktan sonra kullanıcı tespit edilen anomalinin ayrıntılı bilgisine ihtiyaç duyulmaktadır. Bunun için Şekil 3.39'da görüldüğü üzere; anomalinin düzeyi, tarihi, uzaklaşma miktarı, VTYS adı, veri tabanı, tablo adı vb. gibi ayrıntılara ulaşabileceği teknik detaylar özetlenmektedir.

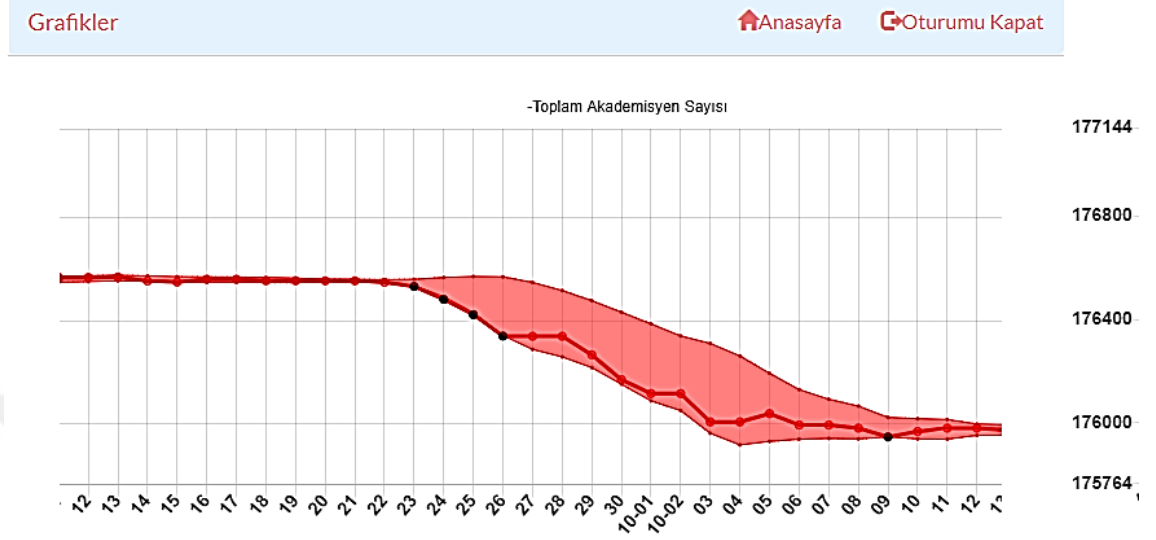
Birinci prototip uygulama testinden elde edilen sonuçlara göre KMİS herhangi bir VTYS'nin veri tabanı tablolarından veri akışı sağlayabilme yeteneğine sahip olmaktadır. Bu veri akışına göre VTYS, veri tabanı tablolarından özet veri toplama ile gerçekleştirmektedir. Bu sayede elde edilen veri seti üzerinde tespit edilen anomaliler önem derecesine göre yöneticilere sunularak bilgi sağlanabilmesi mümkün olmaktadır.

3.4.1.2. Harici Veri Kaynağına Erişim Testi Bulguları

İkinci testte, harici veri kaynaklarından sağlanan veri seti kullanılmıştır. Harici kaynaktan sağlanan veriler dolaylı olarak VTYS'lerden sağlanmaktadır. Ancak, VTYS veri tabanının tamamına erişme söz konusu değildir. Üniversiteler ve bağlı kurumlar sadece belirli bazı verileri paylaşmaktadır. Genellikle kendi VTYS'lerden bu verileri veri paketi olarak paylaşmaktadırlar. Bu veri paketi içerisinde farklı parametrelerde

veriler bulunabilmektedir. Bu arařtırmada Yükseköğretim Kurumu toplam akademisyen sayısı veri paketinde yer alan “TOPLAM” verisi bu testte kullanılmıřtır.

14-10-2020 04:44:52



řekil 3.40. Yükseköğretim Kurumu günlük toplam akademisyen sayısı
(2020 Eylül –Ekim).

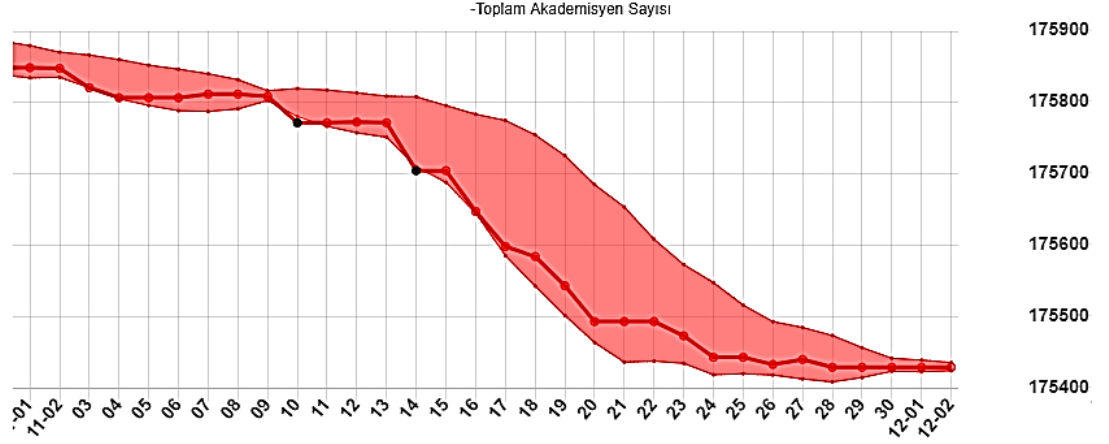
řekil 3.40’da, Yükseköğretim Kurumu toplam akademisyen sayısı deęiřimi ile elde edilen grafięi verilmiřtir. Grafikte Eylül ayının bařından ortasına kadar veriler her ne kadar deęiřkenlik gösterse de tahmin aralıęında bulunduęu görölmektedir. Ancak, 23. günden bařlamak üzere toplam akademisyen sayısında düşüř görölmektedir. Bu düşüř tahmin aralıęının alt sınırını dört gün üst üste ařılmasına sebep olmuřtur. İlerleyen günlerde de tahmin aralıęının alt sınırlarına yakın bir seyir izledięi görölmektedir. Bu da toplam akademisyen sayısında düşüř eęiliminin devam ettięini göstermektedir. Ekim ayının 9. gününde yařanan tahmin aralıęının alt sınırının tekrar ařılması, ilerleyen günlerde yeni bir düşüř sürecinin bařlayabileceęinin bir göstergesi olabilir.

03-12-2020 00:45:20

Grafikler

Anasayfa

Oturumu Kapat



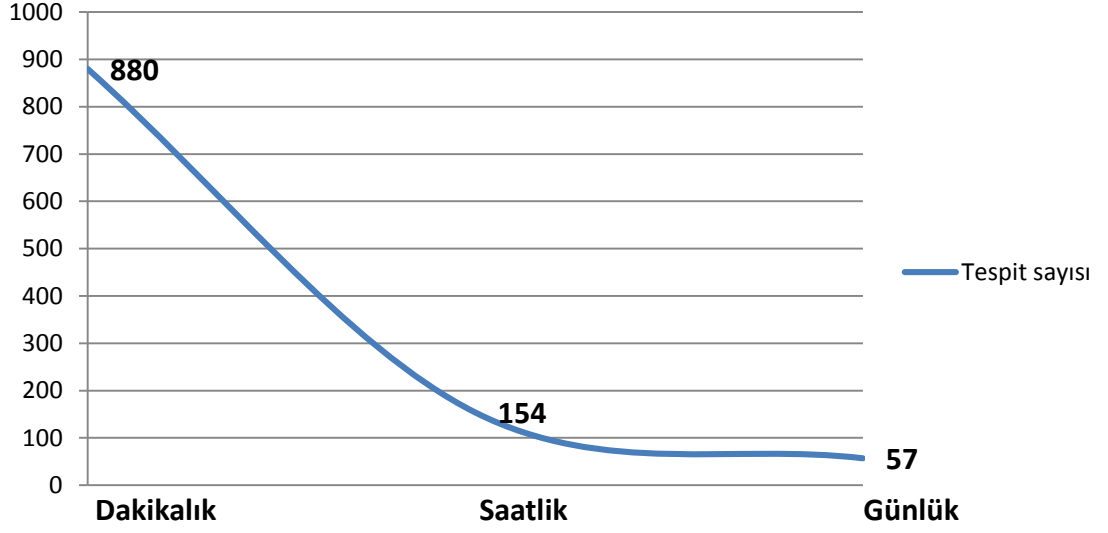
Şekil 3.41. Yükseköğretim Kurumu günlük toplam akademisyen sayısı (2020 Kasım).

Şekil 3.41’de görüldüğü üzere, Yükseköğretim Kurumu toplam akademisyen sayısı Kasım ayında da izlenmeye devam edilmiştir. Nitekim yapılan tahmin doğrultusunda Kasım ayının ortalarına doğru düşüşün hızlandığı görülmektedir. Kasım ayında düşüşün devam ettiği, ayın 10. ve 14. gününde yaşanan tahmin aralığının alt sınırının aşılması sonrası düşüş ivmesinin hızlandığı görülmektedir. Dolayısıyla, yöneticiye olayın bir sonraki aylara sarkmadan akademisyen tahsisi konusunda gerekli önlemi alabilmesi için zaman kazanmasını sağlayabilmektedir. Ayrıca sistem anomalileri tespit ederek yöneticiler ilgilendiği kurumsal verilerin beklentiler dahilinde olup olmadığı bilgisini edinebilir.

3.4.2. Model Başarı Ölçümü Bulguları

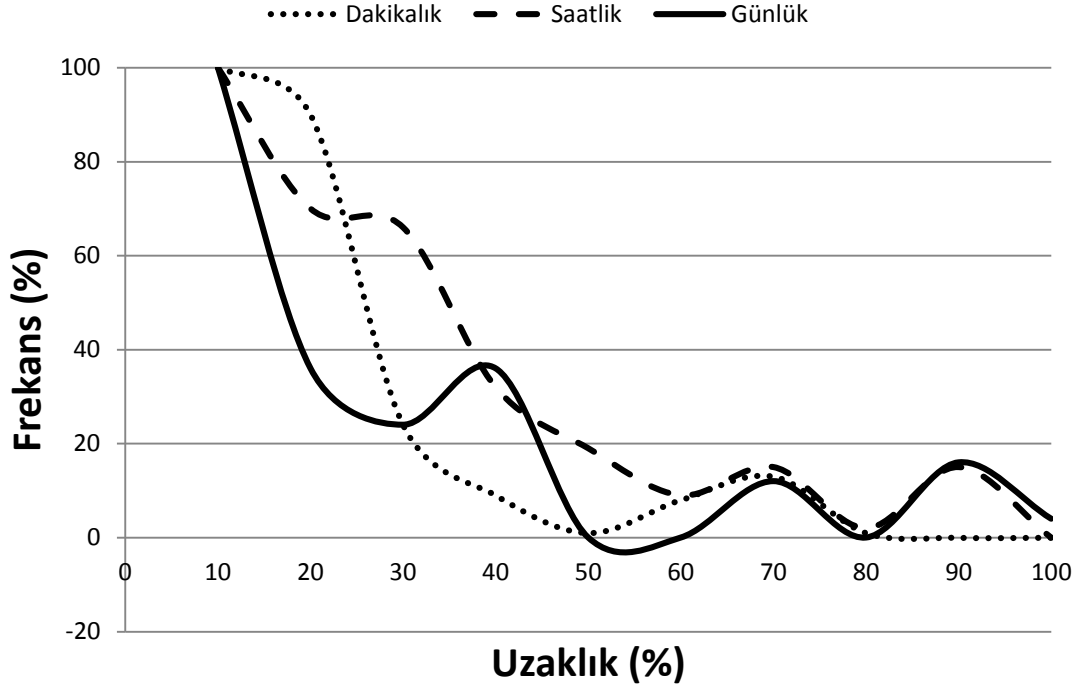
Modelin nicel olarak değerlendirilebilmesi için anomali tespiti başarı ölçümleri yapılmıştır. Başarı ölçümü için 2019 Eylül ile 2020 Mayıs ayları arasında Google Trendler “öğrenci bilgi sistemi” teriminin Türkiye’de gerçekleştirilen web arama yüzdelik oranı verileri kullanılmıştır. Öncelikli olarak tespit edilen anomalilerin zaman aralığına göre yoğunlukları incelenmektedir. Sonraki aşamada ise doğruluk, duyarlılık, F-ölçütü, MCC ve Roc eğrisi bulgularına yer verilmiştir.

Tespit sayısı



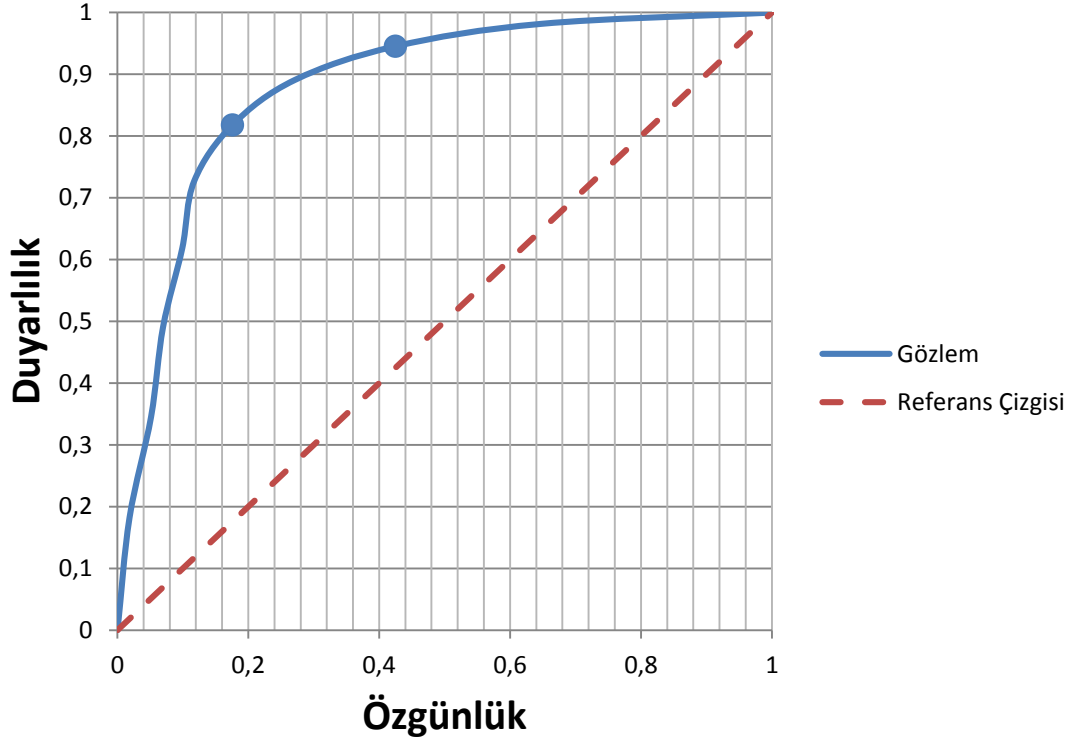
Şekil 3.42. Anomalilerin zaman aralığına göre yoğunlukları.

Şekil 3.42’de, anomalilerin zaman aralığına göre yoğunlukları görülmektedir. Grafiğe göre zaman aralığı arttıkça anomalilerin sayısı azalmaktadır. Bu sonuç literatür taraması yoluyla elde edilen bilgiler ile uyumluluk göstermektedir. Bilindiği üzere, yönetim düzeyinin seviyesi arttıkça yöneticiler daha az ama önemli bilgilere ihtiyaç duymaktadırlar. Operasyonel düzeylerdeki yöneticiler günlük işlerin yürütülmesi için daha fazla ve ayrıntı bilgiler ile ilgilenirler. Elde edilen sonuçlara göre dakikalık zaman aralığında tespit edilen anomalilerin sayısı diğerlerine göre daha fazladır. Bu nedenle, operasyonel seviyedeki yöneticilerin kullanımına uygundur. Günlük zaman aralığında tespit edilen anomali sayısı diğerlerine oranla daha azdır. Dolayısıyla, bu zaman aralığındaki sonuçlar üst düzey yönetimin kullanımına daha uygundur.



Şekil 3.43. Anomalilerin zaman dilimine göre tahmin aralığına uzaklık (%) - frekans (%) grafiği.

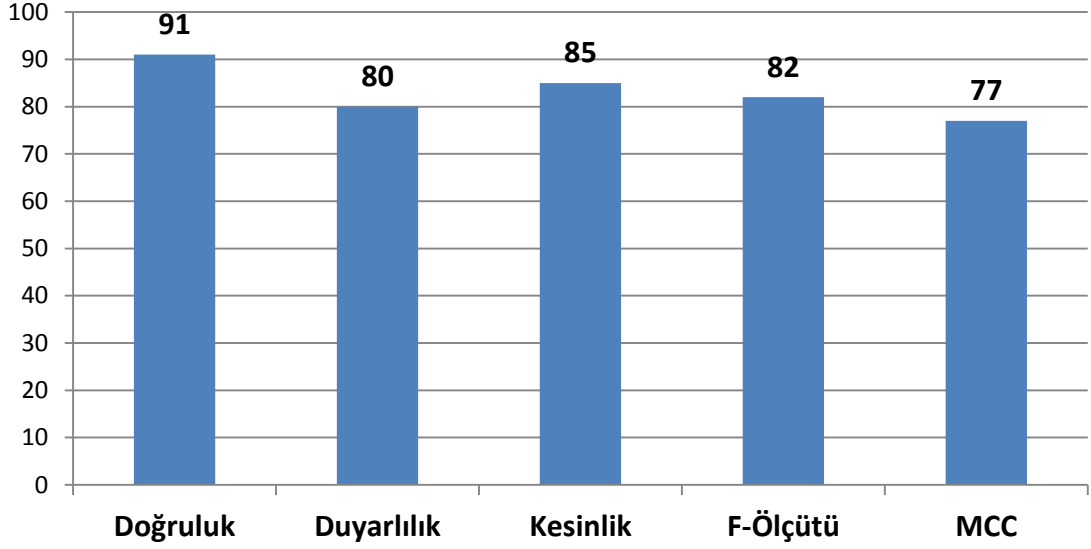
Şekil 3.43 grafiği, (Bkz. Şekil 3.33, Bkz. Şekil 3.34 ve Bkz. Şekil 3.35) grafiklerinin yüzdelerik veri aralığına yerleştirilmesi sonucu elde edilmiştir. Şekil 3.43 grafiği anomalilerin zaman dilimine göre tahmin aralığına olan uzaklık yüzdesi ve frekans yüzdesini göstermektedir. Grafiğe göre dakikalık, saatlik ve günlük zaman aralıklarında tahmin aralığına uzaklık yüzdesi arttıkça verilerin frekansı, başka bir deyişle görülme yüzdesi azalmaktadır. Bu modelin anomali tespitindeki başarısının bir göstergesi olarak kabul edilebilir. Uygulamanın bu başarıyı ne derece sağladığını ölçmek amacıyla ROC eğrisi kullanılmıştır.



Şekil 3.44. Tahmin uzaklığı ve frekans arasındaki ilişkiyi sağlamadaki başarısının ROC eğrisi ile gösterimi.

Şekil 3.44’de, anomalilerin tahmin aralığına olan uzaklığı ile frekansı arasındaki ilişkiyi ne derece sağladığı ROC eğrisi ile gösterilmektedir. ROC eğrisi aynı zamanda anomalilerin derecelendirilmesini sağlayan eşik değerlerinin belirlenmesine yardımcı olmaktadır. Bu doğrultuda ilk eşik değeri x-ekseninde yaklaşık 0.2 olarak görülmektedir. Bir sonraki eşik değeri olarak yaklaşık 0.4 değeri görülmektedir.

Başarı Ölçümleri (%)



Şekil 3.45. Anomali tespiti başarı ölçüm sonuçları.

Şekil 3.45’de başarı ölçüm sonuçları görülmektedir. Bu sonuçlar uygulamanın anomali tespitindeki başarısının ölçümüdür. Elde edilen sonuçlara göre uygulama anomali tespitinde ölçüm yöntemine göre % 77 ile % 91 arasında başarıya sahiptir. F-Ölçütü diğer ölçüm yöntemlerine göre daha doğru sonuçlar vermektedir. Grafikte F-Ölçütüne göre başarısı % 82 olarak görülmektedir. Genel olarak, farklı alanlarda yapılan bu tür başarı ölçümlerinde alt seviye % 80 ile % 85 arasındadır.

3.4.3. Model Değerlendirme Bulguları

Model değerlendirme bulgularında nitel araştırmadan elde edilen sonuçlara yer verilmektedir. Bu amaçla kullanılan anket formundaki her bir soruya yanıt olarak ortaya çıkan tablolara yer verilmiştir. Problemler veya çözüm önerileriyle ilgili frekans (F) tablosu verilerek, tablolar katılımcı ifadeleriyle desteklenmiştir.

3.4.3.1. Birinci Alt Probleme İlişkin Bulgular

“Yöneticiler açısından bu tür bir uygulama modeli fikri ne gibi faydalar sağlayabilir?” sorusuna katılımcılardan gelen cevaplara göre 32 katılımcının verdiği cevaplardaki denetmen için belirlenmiş, toplamda 8 kod tespit edilmiştir. Bunlar Tablo 3.15’de verilmiştir.

Tablo 3.15 Yöneticiler açısından uygulamanın faydaları konusunda frekans tablosu.

Kodlar	F
Karar alma	16
Denetleme	13
Planlama	11
Sistemlerin durumu	4
Aksaklıkları fark etmek	3
İş sürekliliğinin sağlanması	2
Kontrolün sağlanması	1

Tablo 3.15 incelendiğinde, en fazla frekansa sahip kod karar alma olmuştur. Katılımcı Y25 bu konuyla ilgili “Yeni donanıma ihtiyaç olup olmadığına karar vermede zorlanıyorduk. Bu uygulamada yer alan grafik bu konuda yol gösterebilir” ifadelerini kullanmıştır. Bu ifadelere dayanarak katılımcının uygulamada yer alan grafiğin karar almaya yardımcı olduğunu ifade etmiştir. Katılımcı Y14’ün ifadelerinde “Verilen kararların desteklenmesi açısından bu grafik iyi bir kanıt olur” ifadelerini kullanmıştır. 16 katılımcı ise, karar almayla ilgili ifadelere yer vermiştir. Bu durum Tablo 3.15’deki karar alma kodunun frekansı ile görülmektedir.

İkinci frekansa sahip olan kod “Denetleme” dir. Katılımcı Y10’un ifadesinde “Kimin işini düzgün yaptığı burada daha net görünüyor.” olarak ifade etmiştir. Bu ifadelere dayanarak iş süreçlerinin denetlenmesinde yöneticilere bir fikir veriyor olabilmesi oldukça önemli olduğu görülmüştür.

Üçüncü frekansa sahip kod ise, ileriye dönük planların yapılmasında ve mevcut planların gözden geçirilmesine yardımcı olduğunu göstermektedir. Katılımcı Y3 bu konuyla ilgili “Öğrencilerimizin için gerekli olan sınıf ve laboratuvar planlamasında yardımcı olabilir.” ifadelerini kullanmıştır.

Dördüncü ve beşinci sıradaki frekansa sahip olan kod “sistemlerin durumu” ve “aksaklıkları fark etmek” kodlarıdır. Katılımcı Y24’ün konuya yönelik “Sistemin çalışmadığının başkası tarafından bildirilmesi hiç hoş olmuyor. Bu uygulama bize anlık olarak sistemlerin durumunu gösterebiliyor.” açıklamasını kullanmıştır.

Altıncı ve yedinci sıradaki frekansa göre, “İş sürekliliğinin sağlanması” ve “kontrolün sağlanması” kodlarıdır. Katılımcı Y16 “İş sürekliliğinin sağlanmasına yardımcı olur” ifadelerini kullanmıştır. Katılımcı Y19 “Oluşan yoğunluğa göre farklı geçici birimlere eleman tahsis etmede kullanabiliriz” ifadesini kullanmıştır. Buradan anlaşılacağı gibi uygulama iş sürekliliğinin ve kontrolün sağlanmasına kolaylık sağlamaktadır.

Özetle, toplam 50 frekansın bulunduğu temalarda uygulamanın faydaları açısından farklılık gösterdiği görülmüştür. Kimi katılımcı, planlamada faydalılık gösterebileceğine ifade ederken kimi katılımcıların, sistemlerin ve iş sürekliliğinin sağlanmasına yardımcı olacağı kanısındadır. Bunun yanında karar almalarına yardımcı olabilecek bir uygulama olarak görülmektedir.

3.4.3.2. İkinci Alt Probleme İlişkin Bulgular

Araştırmada ilk olarak, “Uygulama modeli fikri olarak eksik bulduğunuz yönleri nelerdir?” sorusu incelenmiştir. 32 yöneticinin verdiği cevaplar listelenmiş denetmen tarafından toplamda 5 problem kodundan rahatsız oldukları belirlenmiştir. Karşılaşılan ve rahatsız olunan problemlerin tümü frekans olarak Tablo 3.16’da gösterilmiştir.

Tablo 3.16 Uygulama modeli fikri olarak eksik görülen yönleriyle ilgili görüşlerinin frekans tablosu.

Kodlar	F
Görsel arayüz	8
Akıllı telefonda görüntüleme	5
Kullanım kolaylığı	4
Zaman aralığı	3
Sesli uyarı vermemesi	2

En çok frekansa sahip kodda yöneticilerin büyük çoğunluğunun ifadelerinde “görsel arayüz üzerinde biraz çalışılmalıdır” konusunu belirttikleri görülmüştür. Katılımcı Y2’nin bu konuda “kullanıcılar kendini özelleştiremiyor” ifadesini kullanmıştır. Katılımcı Y2’nin de ifadeleriyle kullanıcının uygulamada kişisel bir arayüz kullanmayı tercih ettiğini ortaya çıkmıştır. Diğer bir katılımcı Y29’un verdiği yanıt

göre “İlk açtığımda acaba Yönetici olarak mı yoksa Kullanıcı olarak mı gireceğim diye tereddüt ettim.” özellikle ilk uygulama ile karşılaştığında görsel ara yüzde bir tereddüt yaşadıkları görülmüştür. Buna göre ilk giriş ekranı çok fazla karmaşık olmamalı ve kullanıcı istediği arayüz renklerini seçerek, rahat çalışabilmeleri için uygun ortam sağlanması gerekmektedir.

İkinci en çok frekansa sahip olan kod “akıllı telefonlarda görüntülenme” olmuştur. Katılımcı Y27’nin ifadesine göre “Yoğunluktan dolayı çok fazla masaüstü bilgisayar başında çalışmıyorum. Genellikle akıllı telefonumla hallediyorum. Uygulama akıllı telefonlar ile tam uyumlu değil.” katılımcıların akıllı telefonlarını çok sık kullandığını ortaya koymuş ve mutlaka uygulamanın akıllı telefonlar ile uyumlu olması gerektiğini vurgulamıştır.

Diğer önem arz eden bir koddaki kullanıcıların “kullanım kolaylığı” konusunda zorluk yaşadıkları görülmektedir. Katılımcı Y10’a göre “Çok farklı bir uygulama olmuş. Hangi seçenekleri kullanmam gerektiği açık değil” diyerek kullanım zorluğu çektiğini ifade etmiştir. Buna göre uygulama içerisinde açıklayıcı bazı bilgilerin yer alması gerekliliğini savunmuştur. Böylelikle kullanıcılar için uygulama daha rahat kullanılabilir hale getirilmelidir.

3.4.3.3. Üçüncü Alt Probleme İlişkin Bulgular

Araştırmada “Uygulama modeli fikri olarak belirttiğiniz eksikliklere çözüm olarak önerileriniz nelerdir?” sorusuna 32 yöneticinin verdiği cevaplardaki denetmen için belirlenmiş toplamda 6 frekans elde edilmiştir. Bu frekanslar Tablo 3.17’de verilen cevapların çözüm olarak neler sunulduğunu göstermektedir.

Tablo 3.17 Uygulama modeli fikri olarak belirttiğiniz eksiklere çözüm önerileriyle ilgili görüşlerinin frekans tablosu.

Kodlar	F
Görsel arayüz gözden geçirilmeli	11
Ek kişiselleştirme özellikleri eklenmeli	9
Daha sade bir kullanıcı girişi	8
Bir adet kullanıcı doğrulaması olmalı	7
Akıllı telefonlar ile uyumlu hale getirilmeli	3
Açıklayıcı bilgilere yer verilmeli	1

Tablo 3.17 incelendiğinde katılımcıların büyük çoğunluğunun “görsel ara yüzün tekrar gözden geçirilmesi” gerektiğini belirtmişlerdir. Katılımcı Y7’ye göre “Görsel arayüz yeterli ancak daha iyi tasarlanabileceği” ifadelerini kullanmıştır. Katılımcı Y1 çözüm olarak şu şekilde öneri sunmuştur. “Ara yüzde kullanıcıların farklı renk temaları seçebilmesi daha iyi olur. Ancak, bu da iş görür.” demektedir. Bu bağlamda katılımcı Y7 ve Y1’nin de belirttiği gibi uygulamanın ara yüzünün işlevini yerine getirdiği, ancak iyileştirilmesi gerektiğini öngörmüşlerdir. Bu şekilde uygulama kullanıcıların ilgisini çekmeyebilir demektir. Bu bağlamda, uygulama ara yüzünün önemi ön plana çıkmaktadır.

Frekansı fazla olan ikinci kod ise “ek kişiselleştirme özellikleri eklenmeli” yani kullanıcıların kendi renk tonlarını seçebilmesi oldukça önemli olduğunu ifade etmiştir. Bu konuda katılımcı Y11’in ifade ettiği gibi “Önemli olarak gördüğüm bazı verilerin rengini değiştirmek isterdim.” Katılımcılar, renklerle işaretleme yapmanın daha uygun olacağını ifade etmişlerdir. Uygulamanın çok sayıda kişiselleştirilebilir özelliğine sahip olması kullanıcılar için ayrıca bir kullanım kolaylığı sağlayabilir.

Diğer bir önemli konu katılımcıların da ifade ettiği gibi “Daha sade bir kullanıcı girişi” sahip olması gereken bir özelliktir. Bu durumu katılımcı Y14 ve Y15 şöyle açıklamaktadır: “Sadece kullanıcı girişi olsa daha iyi olur”. Kullanıcılar, kendilerini ilgilendirmeyen nesnelere görmek istemediklerini ifade etmektedirler. Bu doğrultuda giriş ekranında sadece bir adet kullanıcı doğrulamasının kullanılması gerektiği ortaya çıkmaktadır.

En az frekansa sahip olan kodlar arasında “Akıllı telefonlar ile uyumlu hale getirilmeli” ve “Açıklayıcı bilgilere yer verilmeli” kodları yer almıştır. Y9 akıllı telefonlar ile uyumluluk koduyla ilgili olarak “Akıllı telefon ile uyumlu olursa istediğim zaman kullanabilirim.” ifadesini kullanmış ve bu ifadeye göre kullanıcıların bazıları masaüstü bilgisayar yerine akıllı cep telefonlarını tercih ettiği görülmektedir. Gelişen teknolojiler ile birlikte bu kullanıcı grubunun gereksinimi göz önünde bulundurulmalıdır. Açıklayıcı bilgiler kodu hakkında Y27 “Menülerde ve modüllerde kısa açıklamalar olsa iyi olur” ifadesini kullanmıştır. Bu ifade ile bazı kullanıcıların ilk

defa karşılaştıkları uygulamayı kullanırken o uygulamayı tanımak istedikleri görülmektedir.

Genel olarak bakıldığında, modelin “varlığına ve anlamına” yönelik derinlemesine bilgi edinmek için yapılan bu anket araştırmasında katılımcıların modelin varlığını benimsedikleri ve katılımcı Y25 bu konuyu “Tabii ki, böyle bir modelin düşünülmesi bile birçok fayda sağlayacaktır.” ifadeleriyle de vurgulamaktadır. Modelde çok ciddi anlamda sorunların bulunmadığı gözlemlenmiştir. Ancak, basit ama önemli görülen katılımcıların istekleri makuldür ve uygulama modeli açısından da yerine getirilmesi modelin başarısını artıracaktır.



SONUÇ VE ÖNERİLER

Kurumsal kaynakların verimli bir şekilde kullanarak belirlenen hedeflere ulaşılmasında yönetim faaliyetlerinin önemli rolü bulunmaktadır. Yöneticilere bu konuda birçok önemli görevler düşmektedir. Karşılaşılan ya da ileride karşılaşılabilecek birçok sorununun çözümü için planlar yapmak ve kararlar almak bu önemli görevlerin başında yer almaktadır. Yöneticiler karar alırken mevcut bazı bilgilerin yanı sıra henüz daha raporlanmamış bilgilere de ihtiyaç duyabilmektedirler. Bu bilginin hızlı bir şekilde sağlanması kararın zamanında ve doğru bir şekilde alınmasını katkı sağlayacaktır. Ayrıca yöneticinin gerçek zamanlı bilgiye sahip olması hızlı hareket edebilme kabiliyetini geliştiren bir konudur. Günümüz teknolojileri yöneticilerin istedikleri bilgilere ulaşabilmesine yardımcı olmaktadır. Bu teknolojilerden biri de bilişim sistemleridir. Bilişim sistemleri sayesinde yöneticilere kolaylıkla çeşitli bilgiler sunulabilmektedir.

Bilişim teknolojilerindeki hızlı gelişmelerle birlikte günümüz üniversiteleri de teknolojinin sağladığı kolaylıkları kullanma eğilimindedirler. Özellikle evrak üzerinde yapılan çoğu işlemler artık elektronik ortamda da yapılabilir hale gelmiştir. İşlemlerin elektronik ortamda yapılması kâğıt tasarrufu sağlama, işlemlerin hızlandırılması, işlem maliyetinin azalması gibi daha sayılabilecek birçok avantajı bulunmaktadır. Üniversiteler özellikle eğitim ve öğretim faaliyetleri için bu avantajları kullanmaktadır. Üniversiteler hizmet sunarken kaynak kullanım miktarı, kaynakların tahsis edilmesi ve ayrıca değişen çevre şartlarına zamanında uyum sağlanabilmesi adına bilişim sistemlerine ihtiyaç duymaktadırlar. Bilişim sistemleri mevcut hizmetin verimliliğinin artırılmasına ve geliştirilmesine, yeni bir hizmetin planlanmasına ve kendine özgü bir iş sürecini ortaya konulmasına yardımcı olmaktadır.

Teknolojik gelişmelerin sağladığı alt yapısı sayesinde anomali tespiti çalışmaları birçok alanda kullanılmaktadır. Kurumlar bu sayede problemleri daha başlangıcında tespit ederek gerekli önlemleri alabilmektedirler. Aynı amaç doğrultusunda bu çalışmada yöneticilere yönelik bir uygulama modeli üniversite kurumu göz önünde bulundurularak geliştirilmiştir. Bu kapsamda kurum içi ve kurum dışı çeşitli veri kaynaklarını bir araya toplayarak analiz edebilen KMİS adında prototip bir uygulama

geliştirilmiştir. Bu sayede gerçek zamanlı olarak ortaya çıkan ve kısa vadede bir anlam ifade etmeyen veriler anlamlı bilgiye dönüştürülmeye çalışılmıştır.

Tez çalışması farklı veri kaynakları kullanılarak anomali tespiti yöntemi ile yöneticilere yönelik bir uyarı modeli oluşturulmasını amaçlamaktadır. Bu kapsamda üniversite yöneticilerine yönelik araştırma gerçekleştirilmiştir. Öncelikli olarak bu araştırmanın fizibilite çalışması için konu teknik ve kullanım alanı boyutlarıyla incelenmiştir.

Ön inceleme sonucunda anomali tespitinin temel unsurlarıyla ilgili bulgular elde edilmiştir. Mevcut araçların veya yazılımların araştırmayı teknik olarak desteklediği görülmüştür. Ancak, çalışmanın kullanım alanı olarak ele aldığı üniversitelere yönelik olarak mevcut yazılımlar tez amacı doğrultusundaki beklentileri tam olarak karşılayamamıştır. Araştırmanın tamamlanmasında önemli bir yere sahip olan bu sorunun çözümü için KMİS geliştirilmiştir.

Günümüz araştırmacılarının fikir paylaşımı ve sorunlara farklı açılardan çözüm üretmelerine büyük katkı sağlayan stackoverflow.com, medium.com, academia.edu, researchgate.net gibi teknik ve akademik sosyal platformlar bu çalışmaya büyük katkı sağlamıştır. Bu platformlar sayesinde diğer araştırmacıların bilgi ve deneyimlerinden istifade edilmiştir. Araştırma sırasında karşılaşılan en kritik problemlerin bile bu sayede çözüldüğü görülmüştür. Böylece ön araştırma neticesinde KMİS'in istenilen amaca yönelik olarak tasarlanabileceği kanısı oluşmuştur.

Prototip uygulama gerek ön araştırma sonucu elde edilen bilgilerin teknik ve kullanım alanı boyutuyla gerekse de mevcut araçlar ve yazılımlar gözlemlenerek hazırlanmaya çalışılmıştır. KMİS'in diğer araştırmacılar tarafından da faydalanılabilmesi ve geliştirebilmesi amacıyla açık kaynak kod kullanılmıştır. Uygulamanın farklı alanlarda kullanılabilmesi için en çok kullanılan veri kaynağı teknolojileri uygulamada kullanılmıştır. Ayrıca uygulamada kullanılan mevcut veri kaynaklarının haricinde yeni bir veri kaynağı da tanımlanabilmektedir.

Prototip uygulamaya iki yönlü olarak test uygulanmıştır. Bu uygulanan testlerin ilkinde VTYS veri tabanı istenilen şekilde sorgulanabilmektedir. Bu sayede veri tabanlarından özet veri akışı sağlanmıştır. Gerçekleştirilen ilk test seçilen VTYS'ler

üzerinde sorunsuz bir şekilde çalıştığı gözlemlenmiştir. Elde ettiği verileri grafiğe yansıtmakta ve anomali tespiti yaparak bu tespitleri üç seviyede sınıflandırmaktadır.

İkinci testte ise, paydaş kurumların paylaşmış olduğu veriler göz önünde bulundurulmuştur. Bu doğrultuda T.C. Yükseköğretim Kurumu istatistik web sayfasında (istatistik.yok.gov.tr) bulunan ve günlük olarak güncellenen toplam akademisyen verisi kullanılmıştır. Bu testte de başarılı bir şekilde harici veri kaynaklarını kullanabildiği görülmüştür. Bu sayede KMİS üniversitelerin ya da kurumların paydaşları ile paylaştıkları verileri takip edebilme avantajına sahiptir. İlk testte olduğu gibi bu veriler grafiğe yansıtılabilmekte, anomali tespiti yapılabilmekte ve tespit edilen anomaliler üç seviyede sınıflandırılabilir. Modelin nicel olarak değerlendirilebilmesi amacıyla uygulamanın başarısı ölçülmüştür. Başarı ölçümünde araştırma kapsamı doğrultusunda iki kriter göz önünde bulundurulmuştur. Bu kriterlerden ilki tespit edilen anomalilerin ne derece gerçek hayatta karşılığı olduğuyla ilgilidir. Bu başarı ölçümü için doğruluk, duyarlılık, kesinlik, F-Ölçütü ve MCC yöntemleri kullanılmıştır. Bu yöntemler 2019 Eylül ile 2020 Mayıs ayları arasında Google Trendler “öğrenci bilgi sistemi” teriminin Türkiye’de gerçekleştirilen web aranma yüzdelik oranı değerlerinden oluşan zaman serisi kesiti üzerinde elde edilen anomaliler üzerinde uygulanmıştır. Elde edilen sonuçlara göre KMİS % 77 ile % 91 arasında başarıya sahiptir. Farklı alanlarda bu yöntemlerin kullanılarak yapıldığı başarı ölçümlerine göre başarılı sayılabilmesi için en az % 80 ile % 85 oranında değerlere sahip olması gerekmektedir. Prototip uygulama, başarı ortalamasında bu alt seviyenin üzerinde kalabildiği gözlemlenmiştir.

Prototip uygulamanın başarısının gözlemlendiği diğer bir kriter tespit edilen anomalileri üç seviyede sınıflandırmadaki başarısıdır. Yönetim düzeyi arttıkça yöneticiler daha az ve öz bilgiye ihtiyaç duymaktadırlar. Yönetim düzeyi azaldıkça daha yoğun ve detaylı bilgiye ihtiyaç duyarlar. Bu ilişki prototip uygulamada tespit edilen anomalilerin tahmin aralığından uzaklaşma mesafesi ve frekansı ile temsil edilmiştir. Bu ilişkinin sağlanmasındaki başarı ölçümü ROC eğrisi ile gösterilmiştir. Prototip uygulama tahmin uzaklığı ve frekansı arasındaki ilişkiyi sağlamada başarılı olduğu gözlemlenmiştir. ROC eğrisinin sağladığı diğer bir sonuç ise eşik değerlerinin

belirlenmesini sağlamasıdır. Bu eşik değerleri sayesinde anomalileri üç seviyede sınıflandırılmasında kullanılan sınırlar belirlenebilmektedir.

Geliştirilen modelin kullanım alanına yönelik olarak bir anlam ifade edip etmediğini ortaya koymak amacıyla nitel araştırma gerçekleştirilmiştir. Araştırmada çıkan temel sonuç, fikir olarak bile ortaya çıkmasının önemli olduğu ve ileriye yönelik çeşitli faydalar sağlanabileceğini göstermiştir. Genel olarak, bu faydaların yönetimin fonksiyonlarını desteklemeye yönelik olduğu gözlemlenmiştir. Araştırma sonuçlarına göre arayüz başta olmak üzere kullanım kolaylığıyla ilgili bazı sorunların olduğu görülmüştür. Buna göre uygulama ara yüzünün iyileştirilmesi adına birçok ifade kullanırken bazı katılımcılar ara yüzden daha çok işlevselliği ile ilgilenmişlerdir. Ara yüzün tasarımı kişilerin bakış açılarının farklı olmasının doğal bir sonucu olarak uygulamalarda en çok eleştirilen bir konudur. Ancak, herkes tarafından genel kabul gören ve kullanıcıya özel bazı arayüz iyileştirmelerinin yapılması gereklidir. Böylece kullanıcılar uygulamayı daha rahat kullanma imkanına sahip olacaktır.

Genel olarak, katılımcıların modeli benimsedikleri ve işlevsel olarak amacını yerine getirerek anlam ifade ettiği kanısına varılmıştır. Ancak, uygulama ara yüzünde yapılabilecek küçük çaplı iyileştirmeler daha iyi sonuçlar verebilir.

Bu araştırmanın diğer araştırmalardan en büyük farkı geniş bir kullanım alanı potansiyeline sahip olmasıdır. Farklı veri kaynaklarını kullanabiliyor olması ve farklı alanlarda çeşitli amaçlara yönelik kullanım esnekliğinin bulunması bu farkın elde edilmesinde önemli rol oynamaktadır.

İş zekâsına bakan yönüyle; veri tabanı, veri analizi, veri madenciliği algoritmalarının kullanımı, raporlama ve sorgulama özellikleri açısından iş zekâsı ile uyumludur. Ancak, iş zekâsının diğer bir özelliği olan veri kaynakları ile uyum göstermemektedir. İş zekâsı, veri kaynağı olarak yazılı, elektronik ve zihinsel verileri kullanabilmektedir. Ancak, önerilen model zaman serisi verilerine odaklandığından yalnızca elektronik veri kaynaklarını kullanabilmektedir.

Araştırma üniversite kurumuna yönelik olarak gerçekleştirilmesine rağmen diğer kurumlarca kullanımına ve geliştirilmesine açıktır. KMİS'in sahip olduğu bu esneklik sayesinde bir sonraki çalışma farklı alanlarda farklı veriler ile yürütülebilir. Daha ileriki

çalıřmalarda yalnız zaman serisi verileri deęil ayrıca gerek zamanlı metin verilerinin izlenebilmesi saęlanarak daha ileri seviyede bir model geliřtirme alıřması yrtlebilir.



KAYNAKLAR

- Abuaitah, G. R. (2013). *Anomalies in Sensor Network Deployments Analysis, Modeling, and Detection*. (Yayımlanmış Doktora Tezi). ABD, Ohio: Wright State Üniversitesi Fen Bilimleri Enstitüsü.
- Akoğlu, L., Tong, H. ve Koutra, D. (2015). "Graph Based Anomaly Detection and Description: A Survey [Grafik Tabanlı Sıradışılık Tespiti ve Tanımlama: Bir Anket]". *Data Mining and Knowledge Discovery*, 29 (3), 626-688.
- Al, U. ve Madran, O. (2004). "Web Tabanlı Uzaktan Eğitim Sistemleri: Sahip Olması Gereken Özellikler ve Standartlar". *Bilgi Dünyası*, 5(2), 259-271.
- Alan, M. A. (2014). "Karar Ağaçlarıyla Öğrenci Verilerinin Sınıflandırılması". *Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 28 (4), 101-112.
- Anameriç, H. (2005). "Yönetim Bilgi Sistemlerinin Yönetim Fonksiyonları Üzerine Etkisi". *Ankara Üniversitesi Dil ve Tarih-Coğrafya Fakültesi Dergisi*, 45 (2), 25-43.
- Ashdown, L. (2015). *Oracle Database Backup and Recovery User's Guide* (11g Release 2b.). Oracle.
- Awad, M. A. (2005). *A Comparison between Agile and Traditional Software Development Methodologies*. The University of Western Australia, Computer Science and software Engineering.
- Aybek, A. (2010). *Çevik Yazılım Geliştirme "Agile"*. İstanbul: Acm Software.
- Ayık, Y. Z. ve Kahveci, F. (2017). "Improving Effects of Materialized View in Database Query Processing and Specific Applications [Veri Tabanı Sorgulama İşleminde Gerçekleştirilmiş Görünümün İyileştirici Etkisi ve Bazı Uygulamaları]". *Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 31 (5), 1057-1067.
- Ayık, Y. Z. ve Kahveci, F. (2019). "Monitoring Instant Data Flow in Organizations" [Bildiri]. R. Durak ve diğ. (Ed.). *4th International Conference on Advances in*

Natural & Applied Sciences, 19-22 Haziran 2019, (ss. 277). Ağrı: Ağrı İbrahim Çeçen Üniversitesi.

- Barış, M. F. ve Mevsim, E. (2014). “Türkiye’de Uzaktan Eğitim Merkezi Kuruluşu ve Organizasyonu: Namık Kemal Üniversitesi Örneği”. *Eğitim ve Öğretim Araştırmaları Dergisi*, 3 (3), 295-303.
- Başaran, H. (2016). *Üniversitelerde Stratejik Yönetim Anlayışının Temel Yönetim Süreçleri Açısından Analizi (Kamu Üniversiteleri Örneği)*. (Yayımlanmış Doktora Tezi). Kütahya: Dumlupınar Üniversitesi Sosyal Bilimler Enstitüsü.
- Bayraktar, E. ve Efe, M. (2006). “Kurumsal Kaynak Planlaması Kurulum Süreci: Kritik Başarı Faktörleri”. *Yönetim Bilimleri Dergisi*, 4 (2), 91-109.
- Blackwell, A. H. (2015). *Encyclopedia of Science* (Üçüncü Baskı). Michigan, ABD: UXL.
- Camden, R., Danesh, A., Helms, H., Rish, G., Kim, E., Gantz, S. ve diğ. (2006). *ColdFusion mx developer's handbook*. San Francisco: Sybex Inc.
- Çavuşoğlu, Ü. ve Kaçar, S. (2019). “Anormal Trafik Tespiti için Veri Madenciliği Algoritmalarının Performans Analizi”. *Academic Platform Journal of Engineering and Science*, 7 (2), 205-2016.
- Ceyhan, İ. F. (2014). *Bağımsız Denetim Kalitesini Artırıcı Bir Yöntem Olarak Veri Madenciliği: Borsa İstanbul Uygulaması*. (Yayımlanmış Doktora Tezi). Sakarya: Sakarya Üniversitesi Sosyal Bilimler Enstitüsü.
- Chandola, V., Banerjee, A. ve Kumar, V. (2009). *Anomaly Detection: A Survey (Teknik Rapor No: TR 07-017)*. Minneapolis, Minnesota, ABD: ACM computing surveys (CSUR).
- Cheboli, D. (2010). *Anomaly Detection o Time Series*. (Yayımlanmış Yüksek Lisans Tezi). Minneapolis, Minnesota, ABD: Minnesota Üniversitesi.
- Confluent. (2018). *Compare Streaming Platforms*. (Erişim: 10 12 2018), <https://www.confluent.io/product/compare/> adresinden alındı

- Coşkun, C. ve Baykal, A. (2011). “Veri Madenciliğinde Sınıflandırma Algoritmalarının Bir Örnek Üzerinde Karşılaştırılması” [Bildiri]. *13. Akademik Bilişim Konferansı, 2-4 Şubat 2011*, (ss. 1-8). Malatya: İnönü Üniversitesi.
- Crinnion, J. (1991). *Evolutionary Systems Development, a practical guide to the use of prototyping within a structured systems methodology*. New York: Plenum Press.
- Değerli, O. (2012). *Naive Bayes Yöntemi ile Blog İçeriklerinin Sınıflandırılması*. (Yayımlanmış Yüksek Lisans Tezi). Ankara: Gazi Üniversitesi Bilişim Enstitüsü.
- Dolek, C. (2007). *Prototip (ilkörnek) Yazılım Geliştirme*. (Erişim: 13 05 2019), <http://dugumkume.org/prototip-ilkornek-yazilim-gelistirme/>
- Ercan, L. (2003). “Motivasyon”. L. Küçükahmet (Ed.), *Sınıf Yönetiminde Yeni Yaklaşımlar* (Üçüncü. Baskı, ss. 103-118) içinde. Ankara: Nobel Yayıncılık.
- Ertekin, A. (2006). *Üniversite Yönetimi ve Organizasyonu ve Trakya Üniversitesi'nde Bir Uygulama*. (Yayımlanmış Yüksek Lisans Tezi). Edirne: Trakya Üniversitesi Sosyal Bilimler Enstitüsü.
- Essed. (2019). *RFID Demirbaş Takip Sistemi*. (Erişim: 10 03 2019), <http://www.essed.com.tr/rfid-demirbas-takip-sistemi/>
- Fernandez, R. C., Pietzuch, P., Kreps, J., Narkhede, N., Rao, J., Koshy, J. ve diğ. (2015). “Liquid: Unifying Nearline and Offline Big Data Integration” [Bildiri]. *7th Biennial Conference on Innovative Data Systems Research, 4-7 Ocak 2015*, (ss. 1-8). Asilomar, Kaliforniya, ABD: The Conference on Innovative Data Systems Research.
- Gezgin, D. M. (2013). *Oracle Veri Tabanı Yönetim Sistemi & PL/SQL*. (Erişim: 07 11 2019), <https://www.researchgate.net/publication/320531402>
- Google. (2020). *Anomaly Detection*. (Erişim: 04 04 2020), https://support.google.com/analytics/answer/9181923?hl=en&ref_topic=6317489
- Gülpek, E. B., Oruç, F. ve Ustalar, E. (2016). *Veri Tabanı Sistemleri*. (Erişim: 07 11 2019), <https://groupdatabase.wordpress.com/veri-tabani-sistemleri/>

- Guozhang, W., Koshy, J., Subramanian, S. ve Paramasivam, K. (2015). "Building a Replicated Logging System With Apache Kafka" [Bildiri]. *41st International Conference on Very Large Data Bases, 31 Ağustos-4 Eylül 2015*, (ss. 1654-1655), Hawaii: Very Large Data Base Endowment.
- Han, J., Kamber, M. ve Pei, J. (2012). *Data Mining Concepts and Techniques 3rd Edition*. Waltham, Massachusetts, ABD: Elsevier Inc.
- Juba, B., Musco, C., Long, F., Sidiroglou-Douskos, S. ve Rinard, M. (2015). "Principled Sampling for Anomaly Detection" [Bildiri]. *22nd Annual Network and Distributed System Security Symposium, 8-11 Şubat 2015*, (ss. 1-14). San Diego, Kaliforniya, ABD: The Internet Society.
- Kaçtıoğlu, S., Keskinçilç, M. ve Kahveci, F. (2019). "Yazılım Kalitesi Faktörü Olarak Yazılım Güvenliğinin Öğrenci Bilgi Sisteminde RIPS Testi ile Değerlendirilmesi". *Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 33 (4), 1261-1278.
- Kahveci, F. ve Özdemir, A. (2016). "Öğrenci Bilgi Sisteminde Değerlendirmenin Veri Madenciliği ile Yapılması". *Yönetim Bilişim Sistemleri Dergisi*, 1 (3), 1-10.
- Karataş, Z. (2015). "Sosyal Bilimlerde Nitel Araştırma Yöntemleri". *Manevi Temelli Sosyal Hizmet Araştırmaları Dergisi*, 1(1), 62-80.
- Keskinçilç, M. ve Kahveci, F. (2019). "Yazılım Mühendisliğinde Çevik Yöntemler Üzerine Kavramsal Bir İnceleme ve Sınıflandırma". *Atatürk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 23 (3), 1067-1091.
- Kis, F. (2016). *Prototyping with Data: Opportunistic Development of Data-Driven Interactive Applications*. (Yayımlanmış Doktora Tezi). İsveç, Stockholm,: Kraliyet Teknoloji Enstitüsü.
- Kocaeli, G. M. (2010). *Veri Tabanı*. Orman İdaresi ve Planlama Daire Başkanlığı.
- Koçoğlu, E. (2010). *Üniversitelerde Yöneticilerin Karar Verme Süreci ve Bu Süreçte Bilişim Sistemlerinin Kullanımı: Ankara İli Örneği*. (Yayımlanmış Yüksek Lisans Tezi). Ankara: Atılım Üniversitesi Sosyal Bilimler Enstitüsü.

- Korkmaz, Ö. (2015). *Makaleler: Prototipleme*. (Erişim: 13 05 2019), <https://sherpa.blog/makale/prototip-islevsel-ve-pratik-olmalidir>
- Kreps, J., Narkhede, N. ve Rao, J. (2011). “Kafka: A Distributed Messaging System for Log Processing” [Bildiri]. *The 6th International Workshop on Networking Meets Databases, 12-16 Haziran 2011*, (ss. 1-7), Atina: ACM SIGMOD/PODS.
- Küçüközkan, Y. (2015). “Liderlik ve Motivasyon Teorileri: Kuramsal Bir Çerçeve”. *Uluslararası Akademik Yönetim Bilimleri Dergisi*, 1 (2), 86-105.
- Laudon, K. C. ve Laudon, J. P. (2014). *Management Information Systems: Managing the Digital Firm* (13. Baskı b.). Londra, ABD: Pearson Education Limited.
- LinkedIn. (2019). *About LinkedIn*. (Erişim: 06 05 2019), <https://about.linkedin.com/>
- MEB. (2012). *Veritabanında Sorgular*. Ankara: Milli Eğitim Bakanlığı.
- Mertz, N. B. (2018). *Anomaly Detection in Google Analytics—A New Kind of Alerting*. (Erişim: 23 05 2019), <https://medium.com/the-data-dynasty/anomaly-detection-in-google-analytics-a-new-kind-of-alerting-9c31c13e5237>
- Miman, M., Yoğun, A. E. ve Yoğun, E. Ö. (2016). “Yönetim Bilgi Sistemleri Kullanım Özellikleri Arasındaki İlişkiler: Lojistik Sektörü Üzerine Bir Çalışma”. *Akademik Bakış Dergisi*, Sayı: 54, 161-173.
- Mohammed, A. F., Humbe, V. T. ve Chowhan, S. S. (2016). “A Review of Big Data Environment and Its Related Technologies” [Bildiri]. *International Conference on Information Communication and Embedded System (ICICES 2016), 25-26 Şubat 2016*, (ss. 1-5). Chennai, Tamilnadu, Hindistan: Institute of Electrical and Electronics Engineers (IEEE).
- Munzert, S., Rubba, C., Meißner, P. ve Nyhuis, D. (2015). *Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining*. Chichester, UK.: John Wiley & Sons
- Natan, R. B. (2009). *HOWTO Secure and Audit Oracle 10g and 11g 1st Edition*. Boca Raton, Florida: Auerbach Publications.

- Ngai, E. ve Wat, F. (2006). "Human Resource Information Systems: A Review and Empirical Analysis [İnsan Kaynakları Bilgi Sistemi: Bir Değerlendirme ve Gözlemsel Analiz]". *Personnel Review*, 35 (3), 297-314.
- Ömürbek, V. (2003). *Kurumsal Kaynak Planlamasında Muhasebe Bilgi Sisteminin Rolü Gıda Sektöründe Uygulama*. (Yayımlanmış Doktora Tezi). Isparta: Süleyman Demirel Üniversitesi Sosyal Bilimler Enstitüsü.
- Özbay, Ö. (2015). Veri Madenciliği Kavramı ve Eğitimde Veri Madenciliği Uygulamaları. *Uluslararası Eğitim Bilimleri Dergisi*, 2 (5), 262-272.
- Özcan, E. (2003). "Türkiye'deki Üniversitelerde İnternet Tabanlı Akademik Kayıt ve Not Takip Sistemleri" [Bildiri]. *Akademik Bilişim 2003, 3-5 Şubat 2003*, (ss. 1-11). Adana: Çukurova Üniversitesi.
- Özdemir, S. (2017). *Prototipleme nedir?* (Erişim: 13 05 2019), <https://www.muhendisbeyinler.net/prototipleme-nedir/>
- Özkan, S. (2014). *Personel Web Sistemi*. (Erişim: 06 04 2020), http://personel.klu.edu.tr/dosyalar/kullanicilar/selcuk.ozkan/dosyalar/dosya_ve_belgeler/dersler/Ders1.pdf
- Özkan, Y. (2013). *Veri Madenciliği Yöntemleri*. İstanbul: Papatya Yayıncılık Eğitim.
- Pazarçeviren, S. Y., Zor, Ü. ve Gürbüz, F. (2015). "İş Zekâsı: Kavramsal Çerçeve, Bileşenler ve İşleyiş". *Siyaset, Ekonomi ve Yönetim Araştırmaları Dergisi*, 3 (1), 75-91.
- Philip, L., Nelson, L., Matt, W. ve David, C. (2003). "TOSSIM: Accurate and Scalable Simulation of Entire Tinyos Applications" [Bildiri]. *Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys)*, 5-7 Kasım 2003, (ss. 126-137). New York, ABD: Association for Computing Machinery.
- Pressman, R. S. (2001). *Software engineering a practitioner's approach*. New York, ABD: The McGraw-Hill Companies.

- Purohit, N., Purohit, S. ve Purohit, R. K. (2012). “Data Mining, Applications and Knowledge Discovery [Veri Madenciliği: Uygulamaları ve Bilgi Keşfi]”. *International Journal of Advanced Computer Research*, 2 (6), 458-462.
- Ramakrishnan, R. ve Gehrke, J. (2003). *Database Management Systems*. Boston: McGrawHill Companies.
- Ranniko, P. (2011). *User-Centered Design in Agile Software Development*. (Yayımlanmış Yüksek Lisans Tezi). Finlandiya, Tampere: Tampere Üniversitesi Bilişim Enstitüsü.
- Robledo, L. F., Sepulveda, J. ve Archer, S. (2013). “Hybrid Simulation Decision Support System for University Management” [Bildiri]. R. Pasupathy, S.H. Kim, A. Tolk, R. Hill ve M. E. Kuhl (Ed.). *Winter Simulation Conference, 5-7 Aralık 2013*, (ss. 2066-2075). Washington, ABD: IEEE Press.
- Rogers, T. (2013). *Key Differences Between Prototypes and Production*. (Erişim: 13 05 2019), <https://www.creativemechanisms.com/blog/bid/313597/Key-Differences-Between-Prototypes-and-Production>
- Ruggiero, P. ve Heckathorn, M. A. (2013). *Data Backup Options*. United States Computer Emergency Readiness Team.
- Sakınmaz, S. (2016). *Apache Kafka Nedir?* (Erişim: 12 10 2018), <http://www.buyukveri.co/apache-kafka-nedir/> adresinden alındı
- Samkarpa, S. (2013). “Status of Human Resource Information Systems (HRIS) in Universities and Affiliated Colleges [İnsan Kaynakları Bilgi Sistemlerinin Üniversitelerde ve Bağlı Enstitülerdeki Durumu]”. *The IUP Journal of Organizational Behavior*, 12 (2), 21-42.
- Samuelsson, M. (2016). *Anomaly Detection In Time Series Data: A Practical Implementation For Pulp And Paper Industry*. (Yayımlanmış Yüksek Lisans Tezi). Gothenburg, İsviçre: Chalmers Teknoloji Üniversitesi, Matematik Bilimleri Bölümü.
- Samza. (2019). *Apache Samza*. 06 05 2019, <http://samza.apache.org/>

- Sanjeev, A.P. ve Zytchow, J.M. (1995). "Discovering Enrollment Knowledge in University Databases" [Bildiri]. U. M. Fayyad ve R. Uthurusamy. *First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, (ss. 246-251). Montreal: AAAI Press.
- Savaş, S., Topaloğlu, N. ve Yılmaz, M. (2012). "Veri Madenciliği ve Türkiye'deki Uygulama Örnekleri". *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 11 (21), 1-23.
- Şengür, D. ve Tekin, A. (2013). "Öğrencilerin Mezuniyet Notlarının Veri Madenciliği Metotları ile Tahmini". *Bilişim Teknolojileri Dergisi*, 6 (3), 7-16.
- Silahtaroglu, G. (2013). *Veri Madenciliği Kavram ve Algoritmalar*. Ankara: Papatya Yayınları.
- Silberschatz, A., Korth, H. F. ve Sudarshan, S. (2011). *Database System Concepts*. New York: McGrawHill.
- Soares, M. M. ve Rebelo, F. (2012). *Advances in Usability Evaluation*. Florida, ABD: CRC Press, Taylor and Francis Group.
- Solaimani, M., Iftekhar, M., Latifur, K., Thuraisingham, B., Ingram, J. ve Şeker, S. E. (2016). "Online Anomaly Detection For Multi-Source Vmware Using A Distributed Streaming Framework [Dağıtık Veri Akışı Bakış Açısıyla Çok-Kaynaklı Vmware İçin Canlı Anomali Tespiti]". *Software: Practice and Experience*, 46 (11), 1479-1497.
- Staimer, M. (2009). *Why Cloud Backup/Recovery (Private, Public, or Hybrid) WILL be Your Data Protection?* Toronto: Asigra.
- Stonebraker, M., Çetintemel, U. ve Zdonik, S. (2005). "The 8 Requirements of Real-Time Stream Processing [Gerçek Zamanlı Veri Akışının İşlenmesinde 8 Gereksinim]". *ACM Sigmod Record*, 34 (4), 42-47.
- Striim. (2016). *Striim Announces Partnership with Confluent*. (Erişim: 12 10 2018), <https://www.striim.com/press/striim-announces-partnership-with-confluent/>

- Tanrıkulu, Z. (1999). *Bilişim Sistemlerindeki Gelişmelerin İşletme Yönetimine Etkileri, Yönetim Bilişim Sistemleri Geliştirme ve Bir Uygulama Örneği*. (Yayımlanmış Doktora Tezi). İstanbul: İstanbul Üniversitesi Sosyal Bilimler Enstitüsü.
- Taşdemir, S. (2018). *Motivasyon Kavramına Genel Bir Bakış Motivasyon Araçları ve Bilgi Teknolojileri ve İletişim Kurumu Ölçeğinde Bir Model Önerisi*. (Yayımlanmış İdari Uzmanlık Tezi). Ankara: Bilgi Teknolojileri ve İletişim Kurumu.
- Tecim, V. ve Gökşen, Y. (2009). “Bilişim Teknolojilerinin Üniversitelerde Etkin Kullanımı Üzerine Bir Çalışma”. *Journal of Yasar University*, 4 (14), 2237-2256.
- Terzi, R., Sağiroğlu, Ş. ve Demirezan, M. U. (2017). “Büyük Veri ve Açık Veri: Temel Kavramlar”. Ş. Sağiroğlu ve O. Koç (Ed.), *Büyük Veri ve Açık Veri Analitiği* (ss. 13-30) içinde. Ankara: Grafiker Yayınları.
- Tocci, G. (2013). *A Comparison of Leading Database Storage Engines in Support of Online Analytical Processing in an Open Source Environment*. (Yayımlanmış Yüksek Lisans Tezi). ABD, Tennessee: Doğu Tennessee Eyalet Üniversitesi Lisansüstü Eğitim Enstitüsü.
- TUBİTAK. (2018). *PostgreSQL Dünyanın En Gelişmiş Açık Kaynaklı Veritabanı*. Ankara: BİLGEM Yazılım Teknolojileri Araştırma Enstitüsü.
- Turgut M., Bulut F. G. ve Altunel H., (2019). “Yazılım Geliştirme Sektöründe Kurumsal Ölçümlerin Belirlenmesi ve Ölçüm Altyapısının Oluşturulması” [Bildiri]. A. Tolga, E. Murat S. ve Tekir (Ed.). *13. Ulusal Yazılım Mühendisliği Sempozyumu, 23-25 Eylül 2019*. İzmir İleri Teknoloji Enstitüsü.
- Uysal, A. Ö. (2011). *Veritabanı Sistemlerinde Sorgu Optimizasyonlarının Veri Analiz Teknikleriyle Geliştirilmesi* (Yayımlanmış Doktora Tezi). İstanbul: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü.
- Üniversitelerde Akademik Teşkilat Yönetmeliği. (1982). *T. C. Resmi Gazete*, 17609, 18 Şubat 1982.
- Versionone. (2016). *What is Agile Methodology?* (Erişim: 28 04 2016), <https://www.versionone.com/agile-101/agile-methodologies/>

- Weber, M. (1964). *Max Weber: The Theory of Social and Economic Organization*. T. Parsons (Ed.), A. M. Henderson (Çevirmen). New York: Free Press.
- Wikipedia. (2020). *Database Engine*. (Erişim: 05 04 2020), https://en.wikipedia.org/wiki/Database_engine
- Yavuz, H. (2014). *Muhasebe Bilgi Sisteminin Üniversite Yönetim Kararları Üzerinde Etkileri: Bartın İlindeki Kobiler Üzerinde Bir Araştırma*. (Yayımlanmış Yüksek Lisans Tezi). Bartın: Bartın Üniversitesi Sosyal Bilimler Enstitüsü.
- Yıldırım, A., ve Şimşek, H. (2008). *Sosyal Bilimlerde Nitel Araştırma Yöntemleri* (Altıncı Baskı). Ankara: Seçkin Yayıncılık.
- Yükseköğretim Kanunu. (1981). *T.C. Resmi Gazete*, 17506, 6 Kasım 1981.
- Yükseköğretim Kurumu. (2019). *Türlerine Göre Mevcut Üniversite Sayıları*. (Erişim: 05 05 2019), <https://istatistik.yok.gov.tr/>

EKLER

Ek 1. Yükseköğretim Kurumu Veri Kullanım İzni



T.C.
ATATÜRK ÜNİVERSİTESİ REKTÖRLÜĞÜ
Sosyal Bilimler Enstitüsü Müdürlüğü

Saya : 48553601-000-E.2000198931

17.08.2020

Konu : Bilgi Edinme (Ferhat KAHVECİ).

KURUMSAL İLETİŞİM DİREKTÖRLÜĞÜNE

İlgi : 12.08.2020 tarihli ve 39657895-000-E.2000195962 sayılı belge.

İlgide kayıtlı yazı ile gönderilen **Ferhat KAHVECİ**'ye ait Bilgi Edinme başvuru dilekçesi ile ilgili bilgiler aşağıya çıkarılmıştır.

Adı geçen, Yönetim Bilişim Sistemleri Ana Bilim Dalı doktora programı öğrencisi olup, danışmanı Doç.Dr. Yusuf Ziya AYIK'ın "<https://istatistik.yok.gov.tr>" adresinde bulunan verileri kullanmak isteğini, geliştirmekte olduğu doktora tez çalışmasının sürekli değişebilen olağanüstü verilerin takibini gerektirmesi ve uygulamanın gerçek veriler üzerinde anlam kazanacağını belirterek, yetkili kurumların izin vermesi halinde bilimsel amaçlı kullanılmasının uygun olacağı kanaatinde olduğunu bildirmiştir. İstekle ilgili olarak, Enstitümüzce de herhangi bir sakınca bulunmamaktadır.

Bilgilerinize arz ederim.

Prof.Dr. Sait UYLAŞ
Müdür

Ek 2: Harici veri kaynaklarındaki alanları listeleme örnekleri

Yükseköğretim Kurumu Toplam Akademisyen Sayısı

```
$objWorksheet = $sonuc->getActiveSheet();
$satir=9; $sutun='U'; $satir_say=0;
foreach ($objWorksheet->getRowIterator() as $row) {
    $satir_say++; $sutun_say='A';
    $cellIterator = $row->getCellIterator();
    $cellIterator->setIterateOnlyExistingCells(false);
    foreach ($cellIterator as $cell) {
        if($sutun_say==$sutun&&$satir_say==$satir) {
            $deger=$cell->getValue(); } $sutun_say++; }
    echo "<table><tr><td width=120>
    <font color=green>".round($deger,3)."</font></td><td><span
    id='countdown'>Başlatılıyor</span></td></tr></table>";
```

Atatürk Üniversitesi ÖBS

```
$deger=$sonuc->veri;
echo "<table><tr><td width=120>".(isset($deger)?'
<font color=green>'. $deger: '<font color=red>Bağlantı
kesildi.')."</font></td>
<td><span id='countdown'>Başlatılıyor</span></td></tr></table>";
```

Akademik Veri İzleme Sistemi (AVESİS)

```
$alan=explode("[",$row['TABLO']);
$alan[1]=substr($alan[1],0,-1); $deger=1;
$row['SORGU']="/".$row['SORGU']; $sorgu=explode( "/",$row['SORGU']);
foreach($sonuc as $key=>$val){
    if( array_search($val){$alan[0]},$sorgu) {
        $deger= $val->{$alan[1]} / $deger;} }
echo "<table><tr><td width=120>
<font color=green>".round($deger,3)."</font></td>
<td><span id='countdown'>Başlatılıyor</span></td></tr></table>";
```

Google Trendler

```
$objWorksheet = $sonuc->getActiveSheet();
$satir=4; $sutun='B'; $satir_say=0;
foreach ($objWorksheet->getRowIterator() as $row) {
    $satir_say++;
    $sutun_say='A';
    $cellIterator = $row->getCellIterator();
    $cellIterator->setIterateOnlyExistingCells(false);
    foreach ($cellIterator as $cell) {
        if($sutun_say==$sutun&&$satir_say==$satir) {
            $deger=$cell->getValue();
        } $sutun_say++; } }
echo "<table><tr><td width=120>
<font color=green>".round($deger,3)."</font></td> <td><span
id='countdown'>Başlatılıyor</span></td></tr></table>";
```

Döviz Kuru

```
$alan=explode("[",$row['TABLO']); $alan[1]=substr($alan[1],0,-1);
$row['SORGU']="+".$row['SORGU']; $sorgu=explode( "+",$row['SORGU']);
$deger=0;
foreach($sonuc as $key=>$val){
    if( array_search($val->{$alan[0]},$sorgu) ) {
        $deger= $deger+ $val->{$alan[1]}; } }
echo "<table><tr><td width=120>
<font color=green>". $deger. "</font></td> <td><span
id='countdown'>Başlatılıyor</span></td></tr></table>";
```

Ek 3: VTYS ve harici veri seti sorgu örnekleri.

Oracle

```
select guncelleme, sum(degers) degers from
((select guncelleme, round(avg(deger),3) degers from (
SELECT substr( TO_CHAR(guncelleme, 'YYYYMMDDHH24MISS'), 0, ".$grup.")
guncelleme, to_number(deger) deger
from vis.veri where GUNCELLEME<(sysdate +
numtodsinterval($(".$kaydir."), ".$zaman_ing."))
and guncelleme > (sysdate - numtodsinterval($(".($uzunluk+$kaydir)."),
".$zaman_ing."))
and kaynak=".$rowa['ADI'].") dt
group by guncelleme)
union
(select substr( TO_CHAR(sysdate + numtodsinterval((rownum+($(".$kaydir-
$uzunluk).")), ".$zaman_ing."), 'YYYYMMDDHH24MISS'), 0, ".$grup.")
araliklar, 0
from all_objects where rownum<=".$uzunluk." )
group by guncelleme
order by guncelleme
fetch first ".$uzunluk." rows only
```

MySQL

```
select a.guncelleme, sum(a.deger) DEGERS from
((select guncelleme, avg(deger) deger from (
SELECT substr( DATE_FORMAT(guncelleme, '%Y%m%d%H%i%s'), 1, ".$grup.")
guncelleme,deger
from test.veri where guncelleme <= sysdate()
and guncelleme > (sysdate() - interval ".$uzunluk." ".$zaman_ing.")
and kaynak=".$rowa['ADI'].")
) dt
group by guncelleme
limit ".$uzunluk.")
union
(SELECT substr( DATE_FORMAT(@zaman:=@zaman - interval 1
".$zaman_ing."), '%Y%m%d%H%i%s'), 1, ".$grup.") araliklar, 0 FROM
test.veri
cross join (select @zaman:=sysdate() ) b
LIMIT ".$uzunluk."
)) a
group by a.guncelleme
order by a.guncelleme
limit ".$uzunluk.";
```

Microsoft SQL Server

```
WITH mycte AS
(SELECT CAST(dateadd($(".$zaman_ing.", $(".$kaydir."), getdate()) as
datetime) DateValue
UNION ALL
SELECT dateadd($(".$zaman_ing.", -1, DateValue) DateValue FROM mycte)
select top ".$uzunluk." a.guncelleme, sum(a.degers) degers from
(select dt.guncelleme, round(avg(dt.deger),3) degers from (
SELECT substring(replace(replace(replace(convert (varchar, guncelleme,
20), '-',''), ':',''), ' ',''), 1, ".$grup.") guncelleme, deger
from TEST.dbo.VERI v where v.GUNCELLEME < dateadd($(".$zaman_ing.",
+($(".$kaydir."), getdate() )
and v.guncelleme > dateadd($(".$zaman_ing.", -
($(".$uzunluk."+($(".$kaydir.")), getdate() )
and v.kaynak=".$rowa['ADI'].")
) as dt
```

```

group by guncelleme
UNION ALL
SELECT top ".$uzunluk."
substring(replace(replace(replace(convert(varchar, m.DateValue, 20), '-
',''),':',''),' ',''), 1, ".$grup.") guncelleme, 0 degers FROM mycte
as m ) as a
group by a.guncelleme
order by a.guncelleme
option (maxrecursion ".$uzunluk.")

```

PostgreSQL

```

select guncelleme, sum(degers) degers from
((select guncelleme, cast(avg(deger) as numeric) degers from (
SELECT substring( TO_CHAR(guncelleme, 'YYYYMMDDHH24MISS'), 1,
".$grup.") guncelleme, deger
from VERI where guncelleme<(current_timestamp::timestamp +
(".$kaydir." || ' ".$zaman_ing."')::interval)::timestamp
and guncelleme > (current_timestamp::timestamp -
(".$uzunluk."+".$kaydir." || ' ".$zaman_ing."')::interval)::timestamp
and kaynak=".$rowa['ADI'].") dt
group by guncelleme)
union
(select substring( TO_CHAR( (current_timestamp::timestamp -
(n+".$kaydir." || ' ".$zaman_ing."')::interval), 'YYYYMMDDHH24MISS'),
1, 12) araliklar, 0
from generate_series(1, "。(.$uzunluk-1)." n) )
tum
group by guncelleme
order by guncelleme

```

Yükseköğretim Kurumu Toplam Akademisyen Sayısı

```

require_once 'Classes/PHPExcel/IOFactory.php';
$sonuc = PHPExcel_IOFactory::load("Özet Akademisyen Sayıları.xls");
$objWorksheet = $sonuc->getActiveSheet();
$satir=9; $sutun='U'; $satir_say=0; $sayac=1;
foreach ($objWorksheet->getRowIterator() as $row) {
$satir_say++;
$sutun_say='A';
$cellIterator = $row->getCellIterator();
$cellIterator->setIterateOnlyExistingCells(false);
foreach ($cellIterator as $cell) {
$deger=$cell->getValue();
if(is_numeric($deger)) {
$dongu=($sayac++)% 4;
if($dongu == 1){$dizi[1]= $sutun_say. $satir_say." ".$deger;}
if($dongu == 2){$dizi[2]= $sutun_say. $satir_say." ".$deger;}
if($dongu == 3){$dizi[3]= $sutun_say. $satir_say." ".$deger;}
if($dongu == 0){echo
"<tr><td>".$dizi[1]."</td><td>".$dizi[2]."</td><td>".$dizi[3]."</td><t
d>".$sutun_say. $satir_say." ".$deger."</td><tr>";}
} $sutun_say++; } }

```

Atatürk Üniversitesi ÖBS

```

foreach($sonuc as $key=>$val) {
echo "<tr><td>".$key."</td><td>".$val."</td><tr>";}

```

Akademik Veri İzleme Sistemi (AVESİS)

```

foreach($sonuc as $key=>$val) {
echo "<tr><td>".$val->{$alan[0]}."</td>
<td>".$val->{$alan[1]}."</td><tr>"; }

```

Google Trendler

```

require_once 'Classes/PHPExcel/IOFactory.php';
$sonuc = PHPExcel_IOFactory::load("multiTimeline.csv");
$objWorksheet = $sonuc->getActiveSheet();
$satir=4; $sutun='B'; $satir_say=0; $sayac=1;
foreach ($objWorksheet->getRowIterator() as $row) {
    $satir_say++;
    $sutun_say='A';
    $cellIterator = $row->getCellIterator();
    $cellIterator->setIterateOnlyExistingCells(false);
    foreach ($cellIterator as $cell) {
        $deger=$cell->getValue();
        if(is_numeric($deger)) {
            $dongu=($sayac++)% 4;
            if($dongu == 1){$dizi[1]= $sutun_say. $satir_say.":". $deger; }
            if($dongu == 2){$dizi[2]= $sutun_say. $satir_say.":". $deger;}
            if($dongu == 3){$dizi[3]= $sutun_say. $satir_say.":". $deger;}
            if($dongu == 0){echo
                "<tr><td>".$dizi[1]."</td><td>".$dizi[2]."</td><td>".$dizi[3]."</td><t
                d>".$sutun_say. $satir_say.":". $deger."</td><tr>";}
            } $sutun_say++; } }

```

Döviz Kuru

```

foreach($sonuc as $key=>$val){
    /*echo str_replace('secilen', (
    isset($_GET['column'])?($_GET['column']==$val-
    >{$_GET['username']}?"selected=selected":""),""), str_replace("degisken
    ", $val->{$_GET['username']}, "secenek");*/
    echo "<tr><td>".$val->{$alan[0]}."</td><td>".$val-
    >{$alan[1]}."</td><tr>";
}

```

Ek 4: Harici Veri Setini Getirme Döngüsü Örnekleri

Yükseköğretim Kurumu Toplam Akademisyen Sayısı

```
$array = array('Kayıt', 'Alan');
foreach(array_unique($array) as $key=>$val){
echo str_replace('secilen', ( isset($_GET
['column'])?($_GET['column']==
$val?"selected=selected":""),""),str_replace
("degisken",$val,"secenek"));}
```

Atatürk Üniversitesi ÖBS

```
foreach($sonuc as $key=>$val){
echo str_replace('secilen',
(isset($_GET['column'])?($_GET['column']==$key?"selected=selected":""
:""),
str_replace("degisken",$key,"secenek"));}
```

Akademik Veri İzleme Sistemi (AVESİS)

```
$array = array();
foreach($sonuc as $key=>$val){
foreach($val as $keykey=>$valval){
$array[] = $keykey;
} }
foreach(array_unique($array) as $key=>$val){
echo str_replace('secilen', ( isset($_GET
['column'])?($_GET['column']==
$val?"selected=selected":""),""),str_replace
("degisken",$val,"secenek")); }
```

Google Trendler

```
$array = array('Kayıt', 'Alan');
foreach(array_unique($array) as $key=>$val){
echo str_replace('secilen', ( isset($_GET
['column'])?($_GET['column']==
$val?"selected=selected":""),""),str_replace
("degisken",$val,"secenek")); }
```

Döviz Kuru

```
$array = array();
foreach($sonuc as $key=>$val){
foreach($val as $keykey=>$valval){
$array[] = $keykey;} }
foreach(array_unique($array) as $key=>$val){
echo str_replace('secilen', ( isset($_GET
['column'])?($_GET['column']==
$val?"selected=selected":""),""),str_replace
("degisken",$val,"secenek")); }
```

Ek 5. Anomali Tespitini Sağlayan Uygulama Kodları

```
/*ANLIK VERİ AKIŞI YOLUYLA GELEN VERİ SETİNİN TEMİZLEME İŞLEMİ.*/
```

```
$dizi=substr($deger,1,strlen($deger));
$ilkdeger=strrpos($dizi,'0')+2;
$ilkuzunluk=strpos($dizi,',',$ilkdeger);
$doldur= substr($dizi,$ilkdeger,$ilkuzunluk-$ilkdeger);
```

```
/*VERİ SETİNİN YAPILANDIRILMASI*/
```

```
$dizi=explode(",",$dizi);
$dizi= array_replace($dizi,
    array_fill_keys(
        array_keys($dizi, 0),
        $doldur
    )
);
```

```
/*VERİ SETİ ÜST VE ALT BANTLARININ BELİRLENMESİ*/
```

```
$dizib=trader_bbands( $dizi,
    7,
    1.5,
    1.5,
    TRADER_MA_TYPE_WMA
);
```

```
/*VERİ SETİNİN ÜST VE ALT BANT NOKTALARININ GRAFİKSEL GÖSTERİMİ İÇİN HAZIRLANMASI*/
```

```
for($i=0;$i<$say-6;$i++){
    $diziy[$i]=$dizi[$i+6];
}
```

```
if(!empty($dizib)) {
    $degeriust[$saya]=implode(",",$dizib[0]);
    $degerialt[$saya]=implode(",",$dizib[2]);
    $degerorta[$saya]=implode(",",$diziy);
}
```

```
//TARİH VERİSİNİN GRAFİKSEL GÖSTERİMİ İÇİN HAZIRLANMASI
```

```
for ($k=0;$k<$say;$k++){
    $hesap=strtotime($simdi)-(($k-7)*60*$yenileme);
    $hesap=$hesap-($hesap%(60*$yenileme));
    $zamantam[$k]=date("Y-m-d H:i:s", $hesap);
```

```

}
$zamantam=array_reverse($zamantam);

/*ÜST VE ALT BANT NOKTALARINI AŞAN VERİLERİN UZAKLIK TESPİTİ
(ANOMALİ TESPİTİ)*/
for ($i = 0; $i <$say-7 ; $i++) {
    $nokta[$i]="";$bust=$dizib[0][$i+6]-$diziy[$i];$balt=$diziy[$i]-
$dizib[2][$i+6];
    if(($bust<0)||($balt<0)) {
        $sqluzaklasma="insert                into                vis.siradisi
values('.$rowa['ADI'].','to_date('.$zamantam[$i].','yyyy-mm-dd
hh24:mi:ss'),'abs(round(($bust<0?$bust:$balt),0)).','.$_GET['zaman'].')";
        $sqluzak="select count(*) sayi from vis.siradisi where
baslik='.$rowa['ADI'].'' and tarih=to_date('.$zamantam[$i].','yyyy-mm-dd
hh24:mi:ss') and uzaklik="abs(round(($bust<0?$bust:$balt),0))." and
zaman=".$_GET['zaman'].''";
    }

/*TESPİT EDİLEN ANOMALİLARIN VERİ TABANINA KAYDEDİLMESİ*/
    $uzak=oci_parse($conn, $sqluzak);
    oci_execute($uzak);
    $bilgi=oci_fetch_assoc($uzak);
    if($bilgi['SAYI']==0) {
        $uzaklasma=oci_parse($conn,$sqluzaklasma);
        oci_execute($uzaklasma);
    }

/*TESPİT EDİLEN ANOMALİLARIN GRAFİK GÖSTERİMİ İÇİN RENGİNİN
FARKLILAŞTIRILMASI*/
    $nokta[$i]="rgb(0, 0, 0)";

}

}

```

Ek 6. Uygulama Başarısı Hesaplamasında Kullanılan Veri Seti

Tarih Gözlem Uzaklaşma			Tarih Gözlem Uzaklaşma			Tarih Gözlem Uzaklaşma		
12.9.2019	DP	169	8.1.2020	DN	0	28.2.2020	DP	34
18.9.2019	DN	0	13.1.2020	DP	30	1.3.2020	DN	0
21.9.2019	DP	325	15.1.2020	DN	0	8.3.2020	DN	0
22.9.2019	DN	0	16.1.2020	YN	0	9.3.2020	DN	0
23.9.2019	DN	0	19.1.2020	YN	0	11.3.2020	DN	0
25.9.2019	DN	0	20.1.2020	DN	0	14.3.2020	DN	0
28.9.2019	DN	0	22.1.2020	DN	0	15.3.2020	DN	0
3.10.2019	DN	0	23.1.2020	DN	0	18.3.2020	DP	4
4.10.2019	DP	6	25.1.2020	DN	0	19.3.2020	DP	4
9.10.2019	DN	0	28.1.2020	DP	52	20.3.2020	DP	40
17.10.2019	DP	24	29.1.2020	DP	105	21.3.2020	DP	114
23.10.2019	YP	3	30.1.2020	DN	0	22.3.2020	DN	0
10.11.2019	YP	69	1.2.2020	DN	0	23.3.2020	DN	0
11.11.2019	DP	119	2.2.2020	DN	0	29.3.2020	DN	0
12.11.2019	YN	0	3.2.2020	DN	0	1.4.2020	DN	0
15.11.2019	YN	0	4.2.2020	DN	0	2.4.2020	DN	0
21.11.2019	DP	100	5.2.2020	DN	0	3.4.2020	DP	140
18.12.2019	YP	1	6.2.2020	DN	0	6.4.2020	DN	0
19.12.2019	YP	10	7.2.2020	DP	224	7.4.2020	DN	0
20.12.2019	DP	131	9.2.2020	DN	0	9.4.2020	DN	0
24.12.2019	DN	0	10.2.2020	DN	0	14.4.2020	DP	21
25.12.2019	DN	0	11.2.2020	DN	0	15.4.2020	DP	51
28.12.2019	DN	0	13.2.2020	DN	0	19.4.2020	DN	0
29.12.2019	DN	0	14.2.2020	YN	0	22.4.2020	DN	0
3.1.2020	DN	0	15.2.2020	DN	0	23.4.2020	DP	5
4.1.2020	DP	27	18.2.2020	DN	0	24.4.2020	DP	127
5.1.2020	DN	0	20.2.2020	DN	0	26.4.2020	DN	0
6.1.2020	DP	310	21.2.2020	YN	0	27.4.2020	YP	0
7.1.2020	DN	0	24.2.2020	DN	0	28.4.2020	DN	0

0: Gözlemin beklenti aralığı içerisinde olduğunu göstermektedir.

DP: Doğru Pozitif

YP: Yanlış Pozitif

DN: Doğru Negatif

YN: Yanlış Negatif

EK 7: Veri Toplama Aracı (Yarı Yapılandırılmış Anket Formu)

Değerli Hocam,

Atatürk Üniversitesi Sosyal Bilimler Enstitüsü Yönetim Bilişim Sistemleri Programı tez aşaması doktora öğrencisiyim. Üniversitelerimizin daha kaliteli hizmet vermesine yardımcı olmak adına tez kapsamında üniversite yöneticilerine yönelik Kurumsal Metrik İzleme Sistemi adında bir uygulama modeli geliştirilmiştir.

Uygulama üniversite ile ilgili metrikleri:

- Öğrenci, akademisyen, çalışan sayıları
- Kampüs/bina içerisindeki öğrenci sayısı
- Akademik yayımlar ile ilgili sayılar
- Üniversite sistemlerindeki yoğunluk ve kesinti durumu
- Genel/fakülte/bölüm bütçe durumu

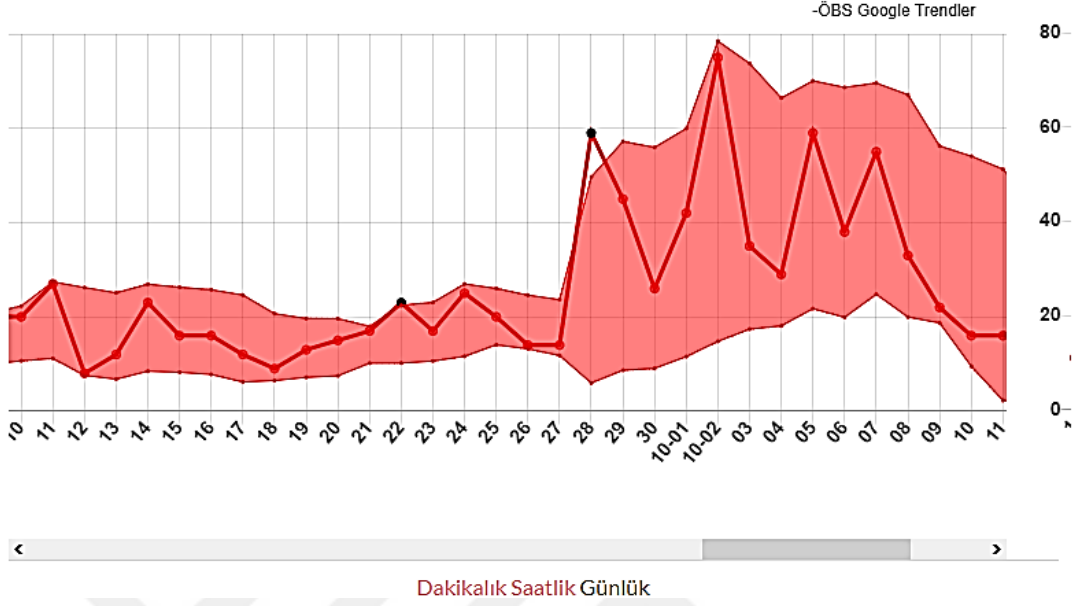
vb. birçok veriyi veri tabanı yönetim sistemlerinden (Oracle, MySQL, Microsoft Sql Server, Microsoft SQL Server ve PostgreSQL) ve harici veri kaynaklarından (Yükseköğretim Kurumu istatistiki bilgilerini, diğer paydaş üniversite ve kurum verilerini veri kullanım izni alınarak) toplayabilme, zaman serisi grafiğine dönüştürme ve yöneticiye uyarı mesajı gönderebilmektedir. Aşağıda uygulamadan alınan örnek bir grafik (Şekil 1) ve uyarı mesajı (Şekil 2) verilmektedir.

22-10-2020 00:03:39

Grafikler

Anasayfa

Oturumu Kapat



Şekil 1. Google Trendler “öğrenci bilgi sistemi” teriminin Türkiye’de gerçekleştirilen web arama yüzdelik oranı.

Açıklama:

Üst düzeyde tanımlı ÖBS Google Trendler adlı Günlük grafikte 2020-09-28 03:00:00 tarihinde beklenti aralığından 9 birimlik uzaklaşma anomali olarak tesbit edilmiştir.

Teknik Detay:

Veri Tabanı Sistemi	:	Harici
Veri Tabanı Adı	:	TRENDLER
Tablo Adı	:	Kayıt[Alan]
Tablo Erişim Adı	:	Türkiye Web Arama
Tablo İşlem Adı	:	ÖBS Google Trendler
Veri Tabanı Bağlantı Adı	:	Google Trendler

Şekil 2. Yöneticiye bildirilen mesaj.

Uygulama modeli fikriyle ilgili bu formu özenle ve içtenlikle cevaplandırmanızı rica ediyorum. Katılımınız için teşekkürler.

#Cinsiyet:.....

#Üniversite kurumunda toplam görev yılınız:.....

Daha önce yaptığınız yönetici görevleri (Örn. Bölüm Başkanı, birden fazla görev yazılabilir):.....

.....

1) Yöneticiler açısından bu tür bir uygulama modeli fikri ne gibi faydalar sağlayabilir?

.....

2) Uygulama modeli fikri olarak eksik bulduğunuz yönleri nelerdir?

.....

3) Uygulama modeli fikri olarak belirttiğinizne eksikliklere çözüm olarak önerileriniz nelerdir?

.....

Ferhat Kahveci

EK 8: Çalışmada yer alan yöneticilerin birim ve görevlerine göre dağılımı

KOD	CİNSİYET	GÖREV YILI	GÖREV
Y1	Erkek	34	Bölüm başkanı
Y2	Erkek	28	İdari ve mali işler daire
Y3	Erkek	33	Bölüm başkanı
Y4	Kadın	29	Dekan
Y5	Erkek	28	Bölüm başkanı
Y6	Erkek	31	Dekan
Y7	Erkek	32	Muhasebe müdürü
Y8	Erkek	28	Muhasebe müdürü
Y9	Erkek	30	Genel sekreter
Y10	Erkek	35	Rektör yardımcısı
Y11	Erkek	28	Rektör yardımcısı
Y12	Kadın	26	Rektör yardımcısı
Y13	Erkek	29	Rektör yardımcısı
Y14	Erkek	25	Bölüm başkanı
Y15	Erkek	29	Dekan
Y16	Erkek	31	Bilgi işlem daire başkanı
Y17	Kadın	24	Bölüm başkanı
Y18	Erkek	33	Yönetim kurulu üyesi
Y19	Erkek	26	Yönetim kurulu üyesi
Y20	Kadın	27	Bölüm başkanı
Y21	Erkek	30	Bölüm başkanı
Y22	Erkek	35	Bölüm başkanı
Y23	Erkek	33	Bölüm başkanı
Y24	Erkek	28	Bilgi işlem daire başkanı
Y25	Erkek	32	Bilgi işlem daire başkanı
Y26	Kadın	34	Öğrenci işleri daire başkanı
Y27	Erkek	35	Dekan
Y28	Erkek	33	Öğrenci işleri daire başkanı
Y29	Erkek	32	Bilgi işlem daire başkanı
Y30	Erkek	28	Yönetim kurulu üyesi
Y31	Erkek	29	Dekan
Y32	Erkek	32	Genel sekreter yardımcısı

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Ferhat Kahveci
Doğum Yeri ve Tarihi	Çaycuma, Zonguldak 10 Mart 1982
Eğitim Durumu	
Lisans Öğrenimi	Gazi Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Sistemleri Öğretmenliği
Y. Lisans Öğrenimi	Gaziantep Üniversitesi Fen Bilimleri Enstitüsü Yönetim Bilişim Sistemleri Programı
Bildiği Yabancı Diller	İngilizce
Bilimsel Faaliyetler	<p>Makaleler:</p> <p>Ayık, Y. Z. ve Kahveci, F. (2017a). “Improving Effects of Materialized View in Database Query Processing and Specific Applications [Veri Tabanı Sorgulama İşleminde Gerçekleştirilmiş Görünümün İyileştirici Etkisi ve Bazı Uygulamaları]”. <i>Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi</i>, 31 (5), 1057-1067.</p> <p>Ayık, Y. Z. ve Kahveci, F. (2017b). “Materialized View Effect On Query Performance [Gerçekleştirilmiş Görünümün Sorgu Performansı Üzerindeki Etkisi]”. <i>World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering</i>, 11 (9), 1020-1023.</p> <p>Ayık, Y. Z. ve Kahveci, F. (2017c). “Ön Hesaplama Tekniğinin Bilişim Sistemlerinde Kullanım Amaçlarına Göre Sınıflandırılması”. <i>International Periodical for the Languages, Literature and History of Turkish or Turkic</i>, 12 (3), 593-606.</p>

	<p>Ayık, Y. Z. ve Kahveci, F. (2019a). "Comparison of Calculating Operations Performances on Database Server and Web Server [Hesaplama İşlemlerinin Veri Tabanı Sunucusu ve Web Sunucusu Üzerindeki Performansının Karşılaştırması]". <i>The Journal of International Social Research</i>, 12 (63), 1324-1329.</p> <p>Kaçtıoğlu, S., Keskinliç, M. ve Kahveci, F. (2019). "Yazılım Kalitesi Faktörü Olarak Yazılım Güvenliğinin Öğrenci Bilgi Sisteminde RIPS Testi ile Değerlendirilmesi". <i>Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi</i>, 33 (4), 1261-1278.</p> <p>Kahveci, F. ve Özdemir, A. (2016). "Öğrenci Bilgi Sisteminde Değerlendirmenin Veri Madenciliği ile Yapılması". <i>Yönetim Bilişim Sistemleri Dergisi</i>, 1 (3), 1-10.</p> <p>Keskinliç, M. ve Kahveci, F. (2019). "Yazılım Mühendisliğinde Çevik Yöntemler Üzerine Kavramsal Bir İnceleme ve Sınıflandırma". <i>Atatürk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi</i>, 23 (3), 1067-1091.</p> <p>Konferansa Katılım ve Bildiriler:</p> <p>Ayık, Y. Z. ve Kahveci, F. (2019b). "Monitoring Instant Data Flow in Organizations" [Bildiri]. R. Durak ve diğ. (Ed.). <i>4th International Conference on Advances in Natural & Applied Sciences, 19-22 Haziran 2019</i>, (ss. 277). Ağrı: Ağrı İbrahim Çeçen Üniversitesi.</p>
İş Deneyimi	
Stajlar	2006 Zonguldak Çatalağzı Termik Santrali Bilgi İşlem 2007 Ankara Keçiören Meslek Yüksekokulu Bilgisayar Programcılığı
Çalıştığı Kurumlar	2011-2020 Erbil ili özel üniversite bilgi işlem personeli öğrenci bilgi sistemi sorumlusu

	2015-2020 Erbil Selahattin Üniversitesi'nden alınan öğretim görevlisi ünvanı ile akademisyen
İletişim	
E-Posta Adresi	ferhatkahveci@hotmail.com
Tarih	24.12.2020

