

**T.C.
SÜLEYMAN DEMİREL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**TÜRKİYE'DE YAPILAN AKADEMİK ÇALIŞMALARIN BÜYÜK VERİ
ARAÇLARI İLE BİBLİYOMETRİK ANALİZİ**

Sinan DURGUT

**Danışman
Prof. Dr. Ecir Uğur KÜÇÜKSİLLE**

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
ISPARTA - 2020**



© 2020 [Sinan DURGUT]

TEZ ONAYI

Sinan DURGUT tarafından hazırlanan "**Türkiye'de Yapılan Akademik Çalışmaların Büyük Veri Araçları ile Bibliyometrik Analizi**" adlı tez çalışması aşağıdaki jüri üyeleri önünde Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü **Bilgisayar Mühendisliği Anabilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak başarı ile savunulmuştur.

Danışman

Prof. Dr. Ecir Uğur KÜÇÜKSİLLE
Süleyman Demirel Üniversitesi



Jüri Üyesi

Dr. Öğr. Üyesi Turgay AYDOĞAN
Süleyman Demirel Üniversitesi



Jüri Üyesi

Dr. Öğr. Üyesi Kubilay TAŞDELEN
Isparta Uygulamalı Bilimler Üniversitesi



Enstitü Müdürü

Doç. Dr. Şule Sultan UĞUR



TAAHHÜTNAME

Bu tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

Sinan DURGUT



İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER.....	i
ÖZET	iii
TEŞEKKÜR.....	vii
ŞEKİLLER DİZİNİ	viii
ÇİZELGELER DİZİNİ	ix
SİMGELER VE KISALTMALAR DİZİNİ	x
1. GİRİŞ.....	1
2. KAYNAK ÖZETLERİ.....	3
3. MATERYAL VE YÖNTEM	6
3.1. Temel Kavramlar	6
3.1.1. Bibliyometri	6
3.1.2. Atıf.....	7
3.1.3. Atıf İndeksleri.....	8
3.1.4. ISI ve ISI'nın Atıf Dizinleri	8
3.1.5. SCI	9
3.1.6. SSCI.....	9
3.1.7. AHCI	10
3.1.8. Bibliyometrik Yasalar	10
3.1.8.1. Bradford yasası	10
3.1.8.2. Lotka yasası.....	10
3.1.8.3. Zipf yasası	11
3.1.8.4. Price yasası.....	11
3.1.8.5. Pareto yasası.....	12
3.2. Kullanılan Araçlar	12
3.2.1. Yazılım Araçları	15
3.2.1.1. Python	15
3.2.1.2. requests library	17
3.2.1.3. odoo kütüphanesi.....	19
3.2.2. Odoo Yazılım Çatısı.....	20
3.2.2.1. Modül oluşturma.....	20
3.2.2.2. Model oluşturma	22
3.2.2.3. View oluşturma.....	23
3.2.2.4. Tree View oluşturma	24
3.2.2.5. Form View oluşturma.....	25
3.2.3. Web Servis	26
3.2.3.1. REST Web Servis	26
3.2.3.2. SOAP Api.....	27
3.2.4. Token Based Authentication	27
3.2.5. Veri Tabanı Araçları	29
3.2.5.1. İlişkisel veri tabanı mimarisi	30
3.2.5.2. Postgresql	31
3.2.5.3. Yapısal olmayan veri tabanı mimarisi.....	31
3.2.5.4. MongoDB.....	32
3.3. Büyük Veri	34
3.3.1. Büyük Verinin Karakteri	35

3.3.1.1. Veri büyüklüğü.....	36
3.3.1.2. Veri çeşitliliği.....	37
3.3.1.3. Veri hızı.....	37
3.3.2. Büyük Verinin Avantajları.....	37
3.3.3. Büyük Verinin Zorlukları.....	39
3.4. Büyük Veri Araçları.....	39
3.4.1. Dağıtık Sistemler.....	40
3.4.2. Apache Spark.....	42
3.4.3. Apache Hadoop.....	44
3.5. Birliktelik Kuralları.....	46
3.5.1. Frequent Pattern Growth Algoritması.....	47
4. ARAŞTIRMA BULGULARI VE TARTIŞMA (ARAŞTIRMA BULGULARI).....	49
4.1. WoS Üzerinden Verilerin Elde Edilmesi.....	49
4.2. Verinin MongoDB Üzerinde Saklanması ve Filtrelenmesi.....	50
4.3. Verinin Odo Programlama Çatısına Aktarımı.....	54
4.4. Verinin Spark ile İşlenmesi.....	54
4.4.1. Üniversitelerin Birliktelik Analizi.....	55
4.4.2. Yazarların Birliktelik Analizi.....	59
4.5. Veri Tabanından Elde Edilen Sonuçlar.....	63
5. SONUÇ VE ÖNERİLER (TARTIŞMA VE SONUÇLAR).....	74
KAYNAKLAR.....	76

ÖZET

Yüksek Lisans Tezi

TÜRKİYE'DE YAPILAN AKADEMİK ÇALIŞMALARIN BÜYÜK VERİ ARAÇLARI İLE BİBLİYOMETRİK ANALİZİ

Sinan DURGUT

**Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı**

Danışman: Prof. Dr. Ecir Uğur KÜÇÜKSİLLE

Bilgi, dünden bugüne kadar gelen teknolojik gelişmelerin kaynağı konumundadır. Bilgiyi yönetebilmek, örgütsel ve yönetsel etkinliğini kullanabilmek pazardaki rekabette öne geçmek konusunda çok kritik bir öneme sahiptir. Bilgi yönetiminin önemi; öğrenme eğrisini hızlandırmak, hızlı iyileştirmeler yapmak, doğru bilgiyi doğru insanlara doğru zamanda ulaştırıp hızlandırılmış transformasyonlara imkân sağlayarak, anlamlandırılmış veriyle birlikte yarının rekabetçi ortamında stratejileri belirleyip alınacak aksiyonları belirlemektir.

Hızlı gelişen teknolojiyle birlikte akademik makalelerden oluşan veri büyük verinin alanına girmiştir. Ancak akademik makalelerin bibliyometrik analiz bakış açısıyla büyük veri araçları kullanılarak incelendiği çok fazla araştırma bulunmamaktadır. Bu çalışmanın ana konusu Türkiye de yazılan akademik makalelerin şu anki durumunu bibliyometrik analiz perspektifi ile incelemektir. İstatistiksel ve matematiksel yöntemlerin kullanıldığı bibliyometrik çalışmalarda makaleler hakkındaki bilgiler, atıf veri tabanlarının kullanılmasıyla elde edilmektedir. Bu çalışma kapsamında, Web of Science atıf veri tabanından Türkiye adresli ve 1980-2019 yılları arasında yazılmış 585236 makale amaçlar kapsamında filtrelenip, bunun sonucunda elde edilen 220969 akademik makale analiz edilmiştir.

Bu analiz sonucunda; alan bazında en çok yayın yapan üniversiteler, alan ve üniversite bazında en çok yayın yapan yazarlar, birlikte çalışma yapmış ve yapma

ihtimali olan üniversiteler, birlikte çalışma yapmışlar ve yapma ihtimali olan yazarlar belirlenerek sunulmuştur.

Anahtar Kelimeler: Bibliyometri, Büyük Veri, Birliktelik Analizi

2020, 83 sayfa



ABSTRACT

M.Sc. Thesis

BIBLIOMETRIC ANALYSIS OF ACADEMIC STUDIES IN TURKEY WITH BIG DATA TOOLS

Sinan DURGUT

**Süleyman Demirel University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering**

Supervisor: Prof. Dr. Ecir Uğur KÜÇÜKSİLLE

Information is the source of technological developments from yesterday to today. To be able to manage knowledge, to use its organizational and managerial effectiveness has a critical importance in taking the lead in the competition in the market. The importance of knowledge management is to accelerate the learning curve, to make rapid improvements, to provide the right information to the right people at the right time and to enable accelerated transformations, to determine strategies and actions to be taken in the competitive environment of tomorrow with meaningful data.

With fast-developing technology, data from academic papers has entered the field of big data. But there is not much research in which academic papers are examined using big data tools from a bibliometric analysis perspective. The main subject of this study is to examine the current status of academic articles written in Turkey from the perspective of bibliometric analysis. In bibliometric studies where statistical and mathematical methods are used, information about articles is obtained through the use of citation databases. Within the scope of this study, 585236 articles from the web of Science citation database, written between 1980 and 2019, addressed to Turkey, were filtered for the purposes and 220969 academic articles obtained as a result were analyzed.

As a result of this analysis, the universities that publish the most on a field basis, the authors who publish the most on a field and University basis, the universities

that have done and are likely to do the work together, the authors who have done the work together and are likely to do it were determined and presented.

Keywords: Bibliometrics, Big Data, Coexistence Analysis

2020, 83 pages



TEŐEKKÜR

Bu arařtırma için beni yönlendiren, karşılařtıđım zorlukları bilgi ve tecrübesi ile ařmamda yardımcı olan deđerli Danıřman Hocam Prof. Dr. Ecir Uđur KÜÇÜKSİLLE'ye teőekkürlerimi sunarım.

Tezimin her ařamasında beni yalnız bırakmayan aileme, sonsuz sevgi ve saygılarımı sunarım.

Sinan DURGUT
ISPARTA, 2020



ŞEKİLLER DİZİNİ

	Sayfa
Şekil 3.1. Proje Süreci.....	14
Şekil 3.2. C++ ve C programla dillerinde “Hello Word” yazan kod bloğu.....	16
Şekil 3.3. C# programlama dilinde “Hello Word” yazan kod bloğu.....	16
Şekil 3.4. Java programlama dilinde “Hello Word” yazan kod bloğu.....	16
Şekil 3.5. HTTP get request örneği (Medium, 2020).....	18
Şekil 3.6. XML-RPC çalışma mantığı.....	19
Şekil 3.7. Odoo örnek modül yapısı.....	21
Şekil 3.8. Manifest dosyası içerik örneği.....	22
Şekil 3.9. Minimum model tanımlama örneği.....	23
Şekil 3.10. Örnek bir view XML kaydı.....	23
Şekil 3.11. Tree view kullanım örneği.....	24
Şekil 3.12. Tree view örnek çıktısı.....	24
Şekil 3.13. Form view kullanım örneği.....	25
Şekil 3.14. Form view örnek çıktısı.....	25
Şekil 3.15. Token based authentication yaşam döngüsü.....	28
Şekil 3.16. MongoDB chart örneği.....	33
Şekil 3.17. Büyük veri kelimesinin 2004-2020 yılları arasındaki aranma sıklığı.	35
Şekil 3.18. Büyük veri bileşenleri (5V)(Özdeş, 2017).....	36
Şekil 3.19. Dağınık sistem ve donanım gücü artırma maliyet karşılaştırması ...	41
Şekil 3.20. Apache Spark çalışma prensibi (Spark, 2020a).....	43
Şekil 3.21. Spark Ekosistemi.....	43
Şekil 3.22. HDFS mimarisi (Hadoop, 2020).....	46
Şekil 3.23. Çalışmada kullanılan JSON filtre.....	52
Şekil 4.1. 2007-2018 Yılları arası türkiye adresli akademik çalışma sayısı.....	64
Şekil 4.2. 2007-2018 Yılları arası Türkiye adresli bilim ve teknoloji alanında yazılan makale sayısı.....	65
Şekil 4.3. 2007-2018 Yılları arası Türkiye adresli sosyal bilimler alanında yazılan makale sayısı.....	65
Şekil 4.4. 2007-2018 Yılları arası Türkiye adresli beşeri bilimler alanında yazılan makale sayısı.....	66
Şekil 4.5. 2014-2018 Yılları arası en çok yayın yapan 10 üniversite.....	66
Şekil 4.6. 2014-2018 Yılları arası bilim ve teknoloji alanında en çok yayın yapan 10 üniversite.....	67
Şekil 4.7. 2014-2018 Yılları arası sosyal bilimler alanında en çok yayın yapan 10 üniversite.....	68
Şekil 4.8. 2014-2018 Yılları arası beşerî bilimler alanında en çok yayın yapan 10 üniversite.....	69
Şekil 4.9. 2014-2018 Yılları arası en çok atıf alan 10 üniversite.....	70
Şekil 4.10. 2014-2018 Yılları arası bilim ve teknoloji alanında en çok atıf alan 10 üniversite.....	71
Şekil 4.11. 2014-2018 Yılları arası sosyal bilimler alanında en çok atıf alan 10 üniversite.....	71
Şekil 4.12. 2014-2018 Yılları arası beşeri bilimler alanında en çok atıf alan 10 üniversite.....	72

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 4.1. En çok beraber çalışma yapan üniversiteler	55
Çizelge 4.2. Birlikte çalışma yapabilecek üniversite tahmin sonuçları.....	56
Çizelge 4.3 En çok beraber çalışma yapan yazarlar	60
Çizelge 4.4. Yazarların birliktelik kural sonuçları.....	62



SİMGELER VE KISALTMALAR DİZİNİ

AHCI	Arts and Humanities Citation Index
API	Application Programing Interface
CWI	Centrum Wiskunde & Informatica
ERP	Enterprise Resource Planning
HDFS	Hadoop Distributed File System
HTTP	Hyper-Text Transfer Protocol
ISI	Institute for Scientific Information
JSON	JavaScript Object Notation
ORM	Object-Relational Mapping
RAID	Redundant Array of Independent Disks
RDD	Resilient Distributed Data
REST	Representational State Transfer
RPC	Remote Procedure Call
SCI	Science Citation Index
SDÜ	Süleyman Demirel Üniversitesi
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SSCI	Social Sciences Citation Index
URL	Uniform Resource Locato
WoS	Web of Science
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformations
pip	Preferred Installer Program

1. GİRİŞ

Çeşitli bilim dallarına dair en güncel veri ve bilgileri içermesi nedeniyle akademik yayınlar ve dergiler; bilimsel iletişim, ulaşılabilirlik ve paylaşım açısından şüphesiz önemli bir yere sahiptir. Bilimsel iletişime köprü olan dergilerin belirli zamanlarda incelenmesi, akademik çalışmaların incelendiği dönemdeki gelişme çizgisini çıkartmakla birlikte eğilimler ve popülaritenin neyin üzerine olduğu hakkında fikir sahibi olma imkânı sağlar. Bu incelemelerin sonucu, konu olarak seçilen bilim dalı veya dallarının geçmişten günümüze kadar olan süreçte gösterdiği değişim ve gelişimin takibini kolaylaştırmaktadır. Bununla birlikte, alanın bu süreçte karşılaştığı sorunların ve bu sorunları giderebilmek için yapılan düzenlemelerin tartışılabilmesine ortam sağlamaktadır (Kozak, 1994). Bu amaca hizmet eden yöntemlerden birisi de bibliyometridir.

Bibliyometri alanına dair ilk çalışmalar 1920'lerden önce yapılmış olmasına rağmen terim olarak bibliyometri 1969 yılında Pritchard tarafından kullanılmıştır (Forsman, 2008). Kitapların ve iletişim materyallerinin istatistiksel ve matematiksel yöntemler kullanılarak incelenmesi bibliyometri olarak tanımlanmaktadır (Pritchard, 1969).

Bu teknik aracılığıyla akademik platformlarda yayımlanmış araştırmalar farklı özellikler (üniversite, yazar, yazar sayısı, yıl, atıflar, konu, anahtar kelimeler vb.) üzerinden incelenerek, bilimsel iletişime dair bazı kanaatlere varmak mümkündür. Bu kanaatler ile ülkelerin, kurumların karşılaştırılması, literatürdeki aktif yazarların tespiti ve dergilerin değerlendirilmesi yapılabilmektedir (Yalçın, 2010). Yani bibliyometrik analiz sonuçları ile disiplinlere göre farklılıkların bulunması, kurumların, yazarların, dergilerin değerlendirilmesi yapılabilmektedir (Yalçın, 2010).

Akademik yayınlar üniversitelerin akademik eğilimlerini görmek için çok önemli bir göstergedir. Son yıllarda artan üniversite sayısı ve paralelinde artan dergi sayısı sonrasında, çok çeşitli bilim dallarında yayınların yapıldığı görülmektedir. Bilim dallarının gelişim düzeyinin belirlenebilmesi ve hangi konuların hangi

dönemlerde yoğun olarak çalışıldığının tespit edilebilmesi açısından yazılı bilimsel çalışmaların belirli dönemler itibariyle incelenmesi önem arz etmektedir.

Bu çalışma, Türkiye'deki 1980 – 2019 yılları arasında WoS (Web of Science) veri tabanı tarafından taranan Türkiye adresli çalışmaların analiz edilmesi amacı ile yapılmıştır. WoS veri tabanından veriler alınırken web servis teknolojisi kullanılmış ve alınan veriler NoSql (Not Only Sql) veri tabanına kaydedilmiştir. Veriler incelendiğinde yazarlara benzersiz bir belirteç tanımlanmadığı, aynı isimli yazarları yazar hakkında bilgi sahibi olunmadığı durumlarda ayıklamanın mümkün olmadığı, çeşitli verilerde yazım hataları olduğu gibi durumlar tespit edilmiştir.

Yapılan çalışmaların birçoğunda spesifik bir alan üzerine yazılmış makaleler incelenmiştir ve bu incelenen makale sayısının 1000'in altında olduğu görülmüştür. Aynı zamanda yazar, üniversite birlikteliği birçok makalede bulunmamaktadır. Spesifik alanlarda yazılan makaleler incelendiği için kategori bazında bir karşılaştırma mümkün olmamıştır. Aynı zamanda yapılan çalışmaların hiçbirinde yapay zekâ teknolojilerinin kullanılmadığı görülmüştür. Bu çalışmada MongoDB, Spark gibi güncel teknolojiler kullanılarak birliktelik analizi yapılmıştır.

Elde edilen verilerden; bir yazar id'ye, adrese, anahtar kelimeye sahip olmayan ve 'Article' tipinde olmayan veriler silinmiştir. Bu işlem sonucunda elde edilen 585236 veriden 220969 'u analize tabi tutulmuş ve sonuçları paylaşılmıştır.

2. KAYNAK ÖZETLERİ

Literatüre bakıldığında hem ulusal hem de uluslararası çalışmalarda bibliyometrik profillerin çıkarılmasına dönük birçok çalışma yer almaktadır.

Niemi (1975) çalışmasında, ABD’de iktisat alanında yayım yapan 24 tane derginin makalelerini incelemiştir. ABD’de çalışma yapan üniversiteleri iktisat bölümlerini bazında değerlendirmiştir. Bu çalışmada değerlendirme ölçütü olarak, makale sayıları ve makalelerin sayfa sayılarını kullanmıştır.

Phene ve Guisinger (1998), 1981-1991 yılları arasında “Journal of International Business Studies” dergisinde yayımlanan çalışmalar üzerinde bibliyometrik analiz gerçekleştirmişlerdir. Bu analizin sonucunda elde ettikleri verileri, kendi çıkarım ve önerilerini dergi ile paylaşmışlardır. Bu sayede dergi gelişiminin arttırılmasına katkıda bulunmuşlardır.

Kozak (2003) çalışmasında, 1997-2002 seneleri arasındaki akademik dergilerdeki değişimleri incelemiş, bu süre zarfında sosyal bilimler alanında yayımlanan dergilerin sayısında ve yayımlanma sıklığında artış olduğunu tespit etmiştir.

Jogaratnam vd. (2005) çalışmalarında, 1992-2001 yılları arasında Turizm alanında yayınlanmış olan 11 dergiyi incelemişlerdir. En fazla katkı yapan yazarların; üniversitelerini, dünya genelinde en fazla katkı yapan bölgeleri ve en fazla tekrar yayın yaptıkları dergileri değerlendirmişlerdir.

Dumrul ve Aysu (2006) çalışmalarında, Erciyes Üniversitesi İİBF dergisinde yayımlanan makalelerin belirli parametrelere göre analizlerini yapmışlardır. Bu analiz sırasında; akademik dereceler, makalelerin yayımlandıkları dil ve ana bilim dalları, makaledeki yazar sayıları, makaledeki yazarların daha önce yayımladıkları makale sayısını dikkate almışlardır. Her bir makalenin profilini ortaya çıkarmışlardır. Araştırmanın sonucunda, incelenen makalelerin bazı

anabilim dallarında daha çok yoğunlaştığı, toplamda orta seviye çalışmaların düşük bir yer kapladığı gibi bulgulara ulaşmışlardır.

Al (2008) çalışmasında, yerli literatürler incelendiğinde bibliyometrik alanda yapılan ilk araştırmanın Özinönü tarafından 1970 yılında yayınlanan “Growth in Turkish Positive Basic Sciences” olarak kabul görüldüğünü belirtmiştir. Bu çalışmada, Türkiye'nin temel bilimlerdeki verimliliği incelemiş, 1990 yılı sonrasında benzer araştırmalarda artış olduğunu gözlemlemiştir.

Taşkin ve Çakmak (2010) çalışmalarında, Bilgi Dünyası Dergisi'nde bulunan 104 makale ve 60 görüşün analizini yapmışlardır. Bu çalışma sonucunda, son 4 yılda hakemli dergiler tarafından yayınlanan makale sayısının önceki yıllara nazaran daha fazla olduğu aynı zamanda makalelerin konusunun çoğunlukla bilgi merkezleri ile ilgili olduğunu ortaya koymuşlardır.

Wang ve Hu (2011) çalışmalarında, 1990-2009 yılları arasında pazarlama ve reklamcılık alanında yayınlamış 39 makalenin atıflarını incelemişlerdir. İnceledikleri 1801 atıf ile reklamcılık alanının temel özellikleri üzerine bir bibliyometrik analiz yapmışlardır.

Sakar ve Cerit (2013) çalışmalarında, bibliyometrik araştırmalar ile ülkelerin bilimsel yayın politikaları ve akademik eğilimlerinin değerlendirilmesinin yapılabileceğini aynı zamanda akademik kurumlarının ve araştırmacılarının yayın performanslarını da ölçmenin mümkün olduğunu belirtmişlerdir.

Metin (2013) çalışmasında, bibliyometrik çalışmalar ile elde edilen verilere istatistiksel, matematiksel teknikler uygulanarak; kitap, belge, makale ve dokümanların belirli özelliklerinin çözümlenmesi ve gelişimine ilişkin yorumlar yapılmasının mümkün olduğunu belirtmiştir.

Bakir (2013) çalışmasında, en çok “pazarlama iletişimi” ve “tüketici davranışları” üzerine çalışmaların yapıldığını keşfetmiştir. “E-pazarlama” ve “fiyat” alanlarında

ise çalışmaların en az yapıldığını belirtmiştir. Aynı zamanda, nicel çalışmaların (%62) daha fazla olduğu sonucunu elde etmiştir.

Evren ve Kozak (2014) çalışmalarında, araştırmacılar, dergiler, akademik kurumlar ve çeşitli disiplinler için bibliyometrik araştırmanın öneminin gün geçtikçe arttığını ve son yıllarda çok popüler hale geldiğini belirtmişlerdir.

Ulu ve Akdağ (2015) çalışmalarında, 1999-2013 yılları arasında “Selçuk İletişim Dergisi’nde” yayınlanan 467 makaleyi analiz etmişlerdir. Bu çalışma sonucunda, makalelerin büyük çoğunluğunun tek yazarlı olduğunu belirlemişlerdir.

Aykanat ve Yıldız (2016) çalışmalarında, 1970-2015 yılları arasında dünyada işletme alanındaki, girişimcilik çalışmalarının gelişimini görmek amacıyla bibliyometrik analiz ve bilimsel haritalama yöntemine başvurmuşlardır. Bu çalışma sonucunda, 2006 yılı sonrasında yayın sayılarında artış olduğunu gözlemlemişlerdir. Aynı zamanda ülkeler bazında dağılımları ortaya koymuşlardır.

3. MATERYAL VE YÖNTEM

3.1. Temel Kavramlar

Bu başlık altında, tezde kullanılan temel kavramlar hakkında bilgi verilmiştir.

3.1.1. Bibliyometri

Bibliyometri köken olarak, “biblio” ve “metric” kelimelerinden meydana gelmektedir. Biblio kelimesi eski Yunancada (Grek) kitap anlamına gelen biblion kelimesinden gelmektedir. Metrics kelimesi de yine Eski Yunancada metricus veya metrikos kelimelerinden türeyen ölçü bilimi anlamına gelmektedir (Sengupta, 1992).

Akademik makalelerin belirli unsurlarını istatistiksel ve sayısal analizler aracılığı ile incelemekte olan bibliyometri, ilk defa 1920 yıllarında dile getirilmeye başlanmıştır. E. Wyndham Hulme tarafından Cambridge Üniversitesinde 1922 yılında ders verilmiştir. Bu ders sonrasında kitap olarak da yayımlanmıştır. Hulme bibliyometriyi, “Belge sayma yoluyla bilim ve teknoloji süreçlerinin aydınlatılması.” diyerek tanımlamıştır (Pritchard, 1969).

Pritchard ise, “İstatistiksel, matematiksel yöntemlerin kitaplar ve iletişim ortamlarına uygulanması” olarak tanımlayarak yakın dönemdeki istatistiksel bibliyografinin temelini atmıştır (Forsman, 2008). Bibliyometri, makalelerin sayısal analizler ve istatistikler aracılığıyla incelenmesinin yanında, akademik çalışmaların kendisinden sonra gelen çalışmaları nasıl etkilediğini ortaya çıkarmak ve atıf analizleri yapmak için de kullanılabilir (McBurney ve Novak, 2002).

Birden fazla tanımı bulunan bibliyometri, belirli alanlar arasında ilişki kurarak korelasyon ve regresyon gibi istatistiksel teknikler ve faktör analizlerinin yapılmasında sunduğu kolaylıklar sayesinde araştırmacılar tarafından kullanılmaktadır (Subramanyam, 1983). Bunun yanı sıra atıf analizlerinde, en çok

kullanılan kaynakların saptanmasında ve çekirdek dergilerin belirlenmesinde kullanılmaktadır (Tatar, 2010).

Makale, kitap gibi bilgi ürünleri üzerinde çalışmalar yapanlar; atıf analiziyle ilgili çalışma yapanlar; kişilerin, kurumların veya ülkelerin akademik durumlarıyla ilgilenenler bibliyometrik analizi en çok kullanan kişiler arasında yer almaktadırlar (Koehler, 2001).

3.1.2. Atıf

Bilimsel ve akademik çalışma yapan araştırmacıların yaptıkları araştırmalarda daha önce bu konu üzerine yapılan araştırmaları kaynak göstermeye başlaması 19. yüzyılda yaygınlaşmaya başlamıştır (Egghe ve Rousseau, 1990). Herhangi bir bilimsel ya da akademik çalışmayı tek başına yapmak doğru olmaz, bunun sebebi araştırmacının içerdiği kaynaklar ve dipnotlarla birlikte o konudaki literatürün bir parçası haline gelmiş olmasıdır. Bir bilimsel araştırmacının kaynakçası çalışmayı yapan kişinin fikir ve bulgularının belli oranlarda daha önceki yapılan araştırmalarla ortak paydada buluştuğunu gösterir (Zan, 2012).

Atıf, kullanılacak olan bilginin belgelendirilmesi işlemini gerçekleştirdiği için, bilimsel bilgiye erişecek olan kişiler için büyük bir önem taşımaktadır (Glänzel, 2009). Literatürde atıf yapmanın, temelde atıf yapan ile birlikte atıf yapılan makale arasında bir bağ kurmakta olduğu belirtilmektedir (Smith, 1981). Aynı zamanda atıf yapmanın genel amacı spesifik bir konu hakkında dokümantasyon ve destek sağlamaktır. Bunun yanı sıra atıf yapmanın çok fazla sebebi bulunmaktadır. Bu nedenler şu şekilde sıralanmıştır (Garfield, 1965):

- Daha önceden konu hakkında makale yazan ve araştırma yapan öncülere saygı göstermek.
- Araştırma Metodoloji, ekipman vb. hakkında bilgi vermek.
- Konu hakkında arka plan okuması sağlama.
- Bir kişinin eserlerini düzeltmesi.
- Başkalarının çalışmalarını düzeltme.
- Önceki çalışmalar hakkında eleştiri yapmak.

- Çalışmadaki iddiaları kanıtlamak.
- Yakın zamanlarda yazılmaya başlayacak çalışmalar için önceden haber vermek ve uyarmak.
- Yanlış bir şekilde tanıtılmış, kötü indekslenmiş veya atıf yapılmamış çalışmaları sergileme.
- Araştırmadaki verileri, fiziksel sabitleri vb. doğrulama.
- Konu veya fikirlerin tartışıldığı orijinal yayını belirleme.
- Başkalarının fikirlerini ya da çalışmalarını reddetmek. (Olumsuz iddialar)
- Başkalarının öncelikli iddialarına itiraz etmek. (Olumsuz saygı)

3.1.3. Atıf indeksleri

Bilimsel dergilerde yayımlanan notlar, mektuplar, düzeltmeler, derlemeler, makaleler ve benzeri araştırmaların hepsinde atıfların mevcut olduğu görülmektedir. Yazarlar araştırmalarını yaparken ve kâğıda dökerken daha önceden konu ile ilgili yapılmış araştırmaları incelemektedirler. Bu inceledikleri çalışmalarda ortak yönleri, farklılıkları, ortak paydaları ya da fikir ayrılıklarını atıf yaparak sergilemektedirler. Bu atıflar, çalışmalar arasındaki resmi bağları kurmaktadır. Atıf indeksleri temelde bu bağ üzerine kurulmuştur (Garfield ve Merton, 1979). Atıf indeksleme kültürüne İngiliz hukukunun öncülük ettiği literatürde görülmektedir. Shepard's Citation (1873) dizini ilk hukuki atıfları içeren dizindir. İngiltere'de mahkemenin dava esaslarına göre kararını daha önceki dava kararlarına dayandırması gerekmektedir. Bu nedenle Shelpard, davalarda daha önceki kararlardan kaç defa yararlandığını gösteren bir liste oluşturmuştur. Bunun sonucunda Shepard atıfları listesi oluşturulmuştur (Garfield ve Merton, 1979).

3.1.4. ISI ve ISI'nın atıf dizinleri

1958 yılında Dr. Eugene Garfield tarafından kurulmuş olan ISI (Institute for Scientific Information) sonrasında Thomson Yayın Grubuna geçmiştir (Cawkell ve Garfield, 2001). 3 ana indeksten oluşmuştur. İlk olarak temel bilimlerde yer alan bilimsel dergileri içeren SCI' yi (Science Citation Index) yayınlamıştır.

SCI, bilgi bankasının oluşturduğu CD ve basılı materyal versiyonudur. SCI-expanded, SCI'nın web ortamındaki versiyonudur, normal versiyonuna göre %60 daha fazla içeriğe sahiptir ve 2000 adetten fazla dergiyi tarayıp indekslemektedir. SCI-expanded sık güncellenmektedir (Asan, 2004).

Thomson Yayın Grubu günümüzde Essential Science Indicators, Journal Citation Reports, WoS, ISI Web of Knowledge ve International Pharmaceutical Abstracts isimli ürünleriyle de bilinmektedir (Scientific, 2007).

ISI, indekslediği dergileri seçerken birçok özelliğini incelemektedir. Bu özelliklerden en önemlisi ise dergilerdeki makalelerin almış olduğu atıf sayısıdır. Aynı zamanda dergiler, indekslerindeki düşüş baz alınarak indeksten çıkarılabilmektedir. Bunların yanı sıra dergilerin buldukları coğrafya ve disiplinleri de etkili faktörlerdendir (Al, 2008).

3.1.5. SCI

SCI, temel bilimler arasındaki bilimsel indekslerden ilk yayınlananı olduğu için en kapsamlı ve geriye dönük en geniş bilgi içerenidir. Kapsamında 20. yüzyılın başlarından başlayarak günümüze kadar Ziraat, Astronomi, Biyokimya, Biyoloji, Biyoteknoloji, Kimya, Bilgisayar, Nöroloji, Onkoloji, Farmakoloji, Fizik, Psikiyatri, Malzeme Bilimi, Matematik, Tıp, Cerrahi, Veteriner Bilimleri gibi alanlarının da içerisinde bulunduğu 150' den fazla konu bulunmaktadır (Al, 2008).

3.1.6. SSCI

SSCI (Social Sciences Citation Index) indeksi, sosyal bilimler alanındaki 1700 adetten fazla bilimsel dergi taranmaktadır. Bazı alanlardaki dergiler(psikiyatri, psikoloji, bilgi bilim gibi) hem SCI hem de SSCI alanındaki alanı altında indekslenmektedir (Al, 2008).

3.1.7. AHCI

AHCI (Arts and Humanities Citation Index), sanat ve insan bilimleri alanlarını içeren 1100'den fazla dergi taranmaktadır. 1975 yılına kadar geriye dönük bilgi içermektedir. SCI ve SSCI indekslerinden bulunmayan bazı alanları (tiyatro oyunu tanıtımı, opera değerlendirmesi, film tanıtımı) indekslemektedir.

3.1.8. Bibliyometrik yasalar

Bu başlık altında, bibliyometrik yasalar hakkında bilgi verilmiştir.

3.1.8.1. Bradford yasası

Enformatik yasalardan en ilgi çekici olarak bilineni Bradford Yasasıdır. Bradford, bilimsel literatürün dağılım kuramı üzerine öncülük eden araştırmasını 1934 yılında tamamlamıştır. Bu araştırma sonucunda oluşturduğu bibliyografyada makaleleri üç grupta değerlendirmektedir. Bu gruplar; özel bir konudaki süreli yayınlar, özel konuyla yakından ilgili süreli yayınlar ve özel konuyla uzaktan ilgili süreli yayınlar olarak belirlenmiştir.

Bu yasa farklı kaynak türlerinden oluşan derme alanları üzerine uygulanmış ve doğruluğu kanıtlanmıştır (Gökkurt, 1994). Bu alanlar ise şu şekildedir:

- Konu bibliyografyaları.
- Bir araştırma kütüphanesinin dermesi.
- Monografi türündeki eserler
- Kütüphanenin dolaşım dermesinde yer alan süreli yayınlar.
- Fotokopi isteklerinin karşılandığı kaynaklar.

3.1.8.2. Lotka yasası

Literatürde enformatik dağılımlar üç grupta incelenmektedir:

1. Ampirik yaklaşım.

2. Teorik yaklaşım
3. Modelleme yaklaşımı.

Lotka, ampirik yaklaşımı rol model alan bir yasadır. Alfred J. Lotka, yazarların bilimsel verimliliklerinin frekans dağılımı üzerinde çalışmıştır (Gökkurt, 1998). Yasa, Lotka' nın kendi adıyla anılmaktadır ve ters kare yasası olarak da bilinmektedir (Birinci, 2008).

3.1.8.3. Zipf yasası

Zipf, gerçekleştirdiği araştırmalar sonucunda, kelime uzunluğu ile kelime kullanım sıklığı arasında bir bağlantı olduğunu tespit etmiştir. Bu tespite göre, kelime ne kadar uzunsa o kadar az kullanılmaktadır. Dolayısıyla kelime ne kadar kısaysa o kadar çok kullanılmaktadır. 1935 yılında yayınladığı eserinde Zipf yasasını anlatmıştır. James Joyce adlı yazarın "Ulysses" isimli kitabını inceleyen Zipf, kitapta geçen 29899 adet kelimeyi kullanım sıklığına göre azalan oranda sıralamıştır. (Zan, 2012).

$$C = r * f \quad (3.1)$$

Her kelimeye 1-29899' a kadar bir sıra vermiştir, denklemden r parametresi olarak yer almaktadır. Sonrasında her sıranın sayısal değerini kitaptaki frekansıyla (f) çarpıp, bu işlem sonucunda C sabitine ulaşmıştır

$$\log C = \log r + \log f \quad (3.2)$$

Denklem 3.1.'deki gibi formüleleştirilmiş yasa matematikte, denklem 3.2.'de görüldüğü gibi ifade edilir (Bayram, 1998)

3.1.8.4. Price yasası

Price, araştırmacıların bilimin gelişimine ne oranda katkı sağladıklarını ölçmek adına Lotka dağılımını da örnek alarak kendi adıyla anılan yasayı oluşturmuştur

(Yılmaz, 2007). Bu yasaya göre, tüm bilimsel makale ve arařtırmaların yarısı, toplam bilim insanlarının karekk sayısına eřit yazar tarafından yazılmaktadır (Sengupta, 1992). rneęin, “Bilgisayar mhendislięi” alanında yazılan makale sayısının 392 olduęunu ve bu alanda makale yazan bilim insanı sayısının 196 olduęu varsayılıں. Bu varsayıma gre, 392 makaleden 196 tanesi 14 bilim insanı tarafından yazılmıř olacaktır.

3.1.8.5. Pareto yasası

İtalyan ekonomist Vilfredo Pareto, 1897 yılındaki arařtırmasında İtalya servetinin %80’ine İtalyan nfusunun %20’sinin; İngiltere topraklarının %80’ine İngiliz nfusunun %20’sinin sahip olduęunu belirlemiř ve sonraki servet daęılımı zerine olan arařtırmalarında oranın genelde korunduęunu keřfetmiřtir (Yılmaz, 2005). Pareto, bahesinde ektięi bezelyelerin %20’sinin mahsuln %80’ini verdięini keřfetmesiyle birlikte, bu gzlemlerinden nemli azınlık ile nemsiz çoęunluk arasında bir modelin olabileceęini fark etmiřtir (McCann, 2001). Ardından, ulusal servete iliřkin daęılım teorisini anlatan modeli, “Cours d’Economie Politique” isimli alıřmasında yayımlanmıřtır (Craft ve Leake, 2002).

Pareto yasasına gre, tahmini olarak ıktıların (makaleler, atıf sayıları, yayın sayıları) %80’inin kaynakların (yazarlar, dergiler) neredeyse %20’si tarafından gerekleřtirileceęi varsayılmaktadır. Bir bařka deyiřle en deęerli dergilerin %20’sinin, kendi alanındaki makalelerin %80’ini yayınladıęı; literatrdeki alıřmaların %80’inin ise, o literatrde arařtırma yapan yazarların %20’si tarafından gerekleřtirildięi oranına dayanmaktadır (Rao ve Neelameghan, 1992).

3.2. Kullanılan Aralar

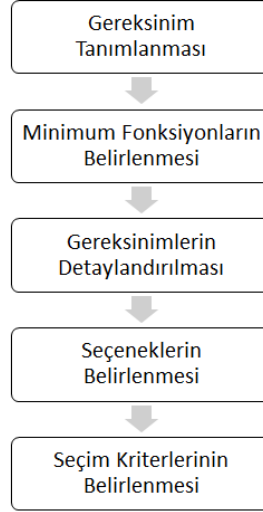
Arařtırma srecinde verilerin servislerden alınmasından, depolanmasına, iřlenip tahmin ve ıkarım yapılmasına kadar birok yazılım aracı kullanılmıřtır. Bu araların her birinin kendine zg zellikleri ve projeye katkıları bulunmaktadır.

Bu bölümde, projede kullanılmış olan tüm araçlar incelenip bu araçlar hakkında bilgi verilecektir.

Araştırmadaki yazılım gereksinimi, büyük ve karmaşık işlemlerin çözümü noktasında oldukça önemli bir role sahiptir. Bunun yanında, sadece büyük ve karmaşık işlemler için değil veriyi elde etme, saklama ve yönetme gibi işlemlerde de önemli katkıları bulunmaktadır. Süreci, yönetimi optimize etmek, senkronizasyonu sağlamak amacıyla ve doğru çözüme en hızlı şekilde ulaşmak için kullanılacak olan araç ve çözümleri proje amacına, planına uygun seçmek önemlidir.

Seçilen araçlar araştırma geliştirme sırasında istenilen süreçlerin bir bölümünü ya da tamamını kapsamalıdır. Sürecin kendi içinde alt süreçlere bölüdüğü durumlarda, seçilen araçların birbirlerine entegre olması yazılım geliştirme döngüsüne girmeden önce planlanmalı ve kolay entegre olabilecek araçlar seçilmelidir. Birbirine entegre olması güç olan yazılımlar seçilmesi, araya farklı entegrasyon yazılımlarının girmesine neden olabilmektedir. Bu da toplam efor gereksinimini arttırmakta ve verimliliğin düşmesine neden olmaktadır. Aynı zamanda araçların doğru seçilmesi iş süreçlerini destekleyecek, geliştiriciye doğru ve güncel bilgi sağlayacaktır.

Bu çalışma kapsamında, yazılım sürecine başlamadan önce gereksinimler tespit edilmiştir. Bu gereksinimler belirlenirken, ilk aşama olarak gereksinim tanımlaması yapılmıştır. Sonrasında, bu tanımlar ışığında detaylara geçilmiştir. Genel tanımlamalar projenin minimum beklenen fonksiyonlarını belirtmektedir. Bu fonksiyonlar detaylandırılırken, tüm alt fonksiyonları ile beraber düşünülerek detaylandırılır. Bu gereklilikleri karşılayabilecek yazılım araçları araştırılmış ve bu araştırma sonucuna göre araç seçimi yapılmıştır.



Şekil 3.1. Proje Süreci

Şekil 3.1. de görülen aşamalardan ilki olan gereksinim tanımlanması şu işlem basamaklarını içermektedir:

- 1- Yazılım geliştirilecek dilin belirlenmesi.
- 2- Esnek sorgular yazılabilecek yapısal olmayan veri tabanına karar verilmesi.
- 3- Verilerin görselleştirilmesi için yazılım çatısı belirlenmesi.
- 4- Yapısal bir veri tabanı belirlenmesi.
- 5- Büyük veri işleme aracı belirlenmesi.

Verilerin internet ortamından yerel bilgisayara çekmek, veriyi düzenlemek, veri tabanına kaydetmek, veri tabanı sorgularına göre işlem yapmak ve işlemleri otomatik hale getirebilmek için güçlü bir programlama diline ihtiyaç bulunmaktadır.

WoS veri tabanından çekilecek veriler web servisi yoluyla alınmıştır. Bu veriler XML (Extensible Markup Language) formatından JSON formatına çevrildikten sonra yerel bilgisayara kaydedilmiştir. JSON data üzerinde sorgu yapılabilmesi ve alanları net inceleyebilmek adına yapısal olmayan NoSql veri tabanına ihtiyaç duyulmaktadır.

Verileri bir arayüzde sunmak ve görselleştirmek için bir web yazılım çatısına ihtiyaç duyulmaktadır. Bu adımda, verileri yapısal bir veri tabanına taşımanın uygun olacağı düşünülmüştür.

Tüm incelemeler tamamlandığında, veriler üzerinden çıkarım ve tahmin yapmak amacı ile bir büyük veri aracına ihtiyaç duyulmaktadır.

3.2.1. Yazılım araçları

Bu başlık altında, kullanılan yazılım araçları hakkında bilgi verilmiştir.

3.2.1.1. Python

Python, ilk versiyonu 1991 yılında Guido Van Rossum tarafından ortaya konulmuş genel amaçlı bir programlama dilidir. Ortaya çıkış ve geliştirilme sürecinde, başka bir genel amaçlı programlama dili olan ABC programlama dilinin etkisi olmuştur (Dierbach, 2012). Rossum, CWI (Centrum Wiskunde & Informatica) araştırma merkezinde, üzerine çalışmakta olduğu programlama dilinin yeterli olmadığını, sınırlı olması nedeni ile geliştirmeye açık bir dil olmadığını fark etmiştir. Bunun üzerine ABC programlama dilinin güzel yanlarını alarak yeni bir dil yani Python programlama dilini geliştirmiştir. Dinamik bir dil olan Python, esas olarak prosedürel ve nesne tabanlı programlama yaklaşımlarını aynı, belli bir oranda ise fonksiyonel programlamayı desteklemektedir. İlk sürümünden şimdiye kadar geçen süreçte geliştiricilerin dikkatini çeken Python, dünya genelinde geniş bir kullanıcı sayısına ulaşmış nesne tabanlı bir programlama dilidir.(Python, 2020)

Python, söz dizimi (syntax) yapısı incelendiğinde ilk göze çarpan sadelik ve kolay okunabilirliğidir. Java, C#, C++, gibi dillerin söz dizimleri ile karşılaştırıldığında çok daha az bir satırla ve daha okunaklı bir şekilde aynı işi yapacak yazılım geliştirilebileceği görülmektedir. Aynı zamanda kod geliştirilirken satırlardaki girintilerin (indentation) çok önemli oluşu, düzenli kod yazma alışkanlığı kazandırmakta ve okunabilirliği arttırmaktadır.

```
main.cpp | main.c
1 #include <iostream> | 1 #include <stdio.h>
2 | 2
3 using namespace std; | 3 int main()
4 | 4 {
5 int main() | 5     printf("Hello World");
6 { | 6
7     cout<<"Hello World"; | 7     return 0;
8 | 8 }
9     return 0; | 9
10 } |
11 |
```

Şekil 3.2. C++ ve C programla dillerinde “Hello Word” yazan kod bloğu

```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         Console.WriteLine("Hello World");
8     }
9 }
```

Şekil 3.3. C# programlama dilinde “Hello Word” yazan kod bloğu

```
Main.java | saved
1 class Main {
2     public static void main(String[] args) {
3         System.out.println("Hello world!");
4     }
5 }
```

Şekil 3.4. Java programlama dilinde “Hello Word” yazan kod bloğu

Şekil 3.2., Şekil 3.3. ve Şekil 3.4’de farklı programlama dillerinde konsola “Hello Word” yazan kod blokları verilmiştir. Python programlama dilinde sadece print(“Hello World”) yazmak aynı işi yapmak için yetmektedir.

Python programlama dili C dili ile geliştirilmiştir. Python'ın standart kütüphaneleri, geliştirme araçları ve bunun yanında birçok ek kütüphanesi internette ücretsiz ve lisans sorunu olmaksızın sunulmaktadır. Sunulan bu programlama dili ve araçlarıyla eğitim, iş gibi akla gelebilecek her alanda ücretsiz geliştirme yapılabilir (Python, 2020). Ayrıca çok fazla sayıda açık kaynak kodlu kütüphanesi bulunması, geliştirme aşamasında avantaj sağlamaktadır. Daha önceden yazılan direkt kütüphaneler kullanılabileceği gibi düzenlenip ihtiyaca göre yeniden düzenlenerek de kullanılabilir.

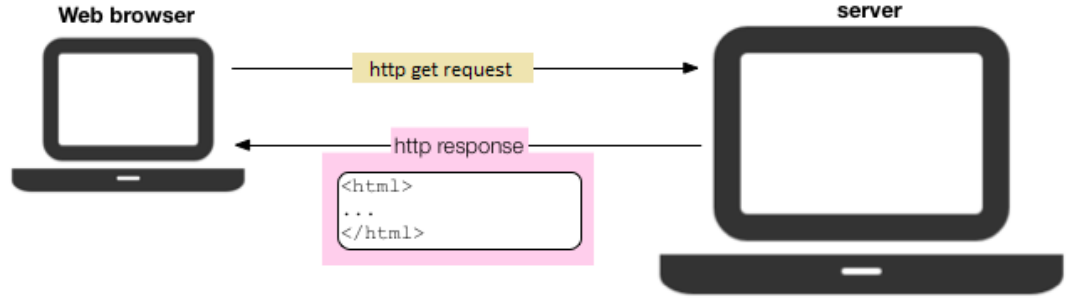
Python'ın kolay öğrenilebilir özelliği her türden ve yaşta kullanıcı için önemli bir avantaj sağlamaktadır. Python programlama dilinin basitliği, öğrenilmek istenilen algoritmalara, bilimsel problemlere veya tasarlanmak istenen sisteme odaklanılmasını kolaylaştırmaktadır.

3.2.1.2. requests library

Python, standart modüllerinin yanında harici yüzlerce kullanışlı modül ile birlikte güçlü bir dildir. Bu gücü veren önemli modüllerden bir tanesi de HTTP (Hyper-Text Transfer Protocol) isteklerinde kullanılan requests modülüdür.

HTTP, web browser ile web server arasında iletişim kurulmasını sağlayan bir protokoldür. HTTP 1.1 versiyonu (RFC 2616) ile tanımlanmış olan ve diğer eklentilerle beraber gelen başlıca HTTP metotları şunlardır:

1. **GET:** Bu metodu sunucudaki veriyi almak için kullanılmaktadır. GET metodu, istemcinin sunucudaki kaynaklara erişmesi için kullanılmaktadırlar (Medium, 2020).



Şekil 3.5. HTTP get request örneği (Medium, 2020)

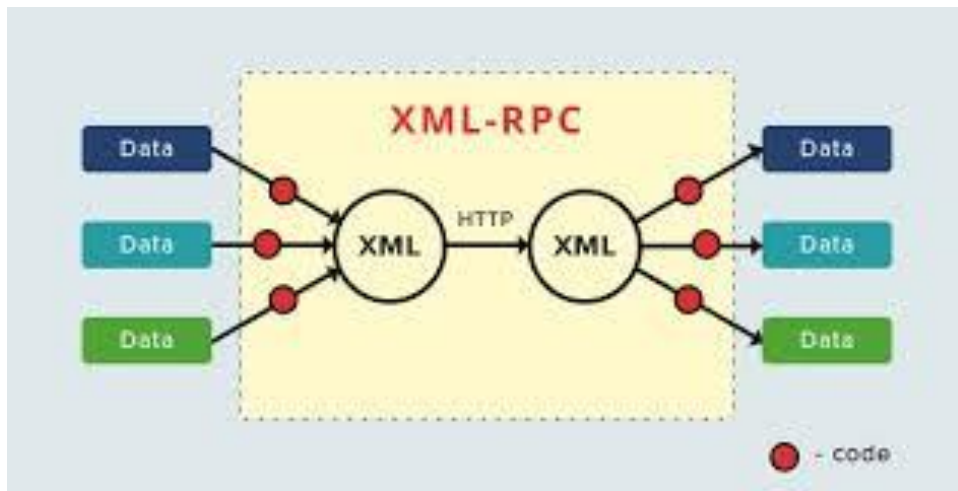
Şekil3.5.'de örnek bir get isteği görülmektedir. Get metodu ile sorgu metinleri URL içinde gönderilebilir. Bunun güzel tarafı; istemcinin bookmark edilebilmesi ve aynı sorguyu içeren isteklerin sonrasında gönderilmesinin sağlanabilmesidir. Ayrıca, tarayıcının “geri” tuşu aracılığıyla tarayıcı geçmişindeki sayfa çağrılarak aynı sayfalara tekrar ulaşabilmektedir. GET metodunda sorgular URL üzerinde açık bir şekilde yazıldığı için güvenlik gerektiren durumlarda kullanılmamaktadır (Medium, 2020).

2. **POST:** Bu metot aracılığıyla sunucuya veri yazdırılabilmekte ve eklenebilmektedir. Bu metot kullanılırken ihtiyaca göre sorgu parametreleri URL içinde ya da istek gövdesinde gönderilebilmektedir (Medium, 2020).
3. **PUT:** Bu metod aracılığı ile servis sağlayıcı üzerindeki bir kayıt güncellenebilmektedir. Hangi kaydın güncelleneceğinin belirlenebilmesi için, o kaydın benzersiz anahtarının istek içerisinde gönderilmesi gerekmektedir (Medium, 2020).
4. **DELETE:** Bu metod aracılığı ile servis sağlayıcı üzerindeki bir kayıt silinebilmektedir. Aynı PUT metodundaki gibi hangi kaydın silineceğinin istek içerisinde belirtilmesi gerekmektedir (Medium, 2020).

3.2.1.3. odoorpc kütüphanesi

RPC (Remote Procedure Call), bir istemcinin HTTP üzerinde bir web servisini kullanması, uzak işlev çağırma (RPC) olarak adlandırılmaktadır. Aslında RPC, aynı ağ içerisindeki farklı bir makinede bulunan bir fonksiyona ulaşabilme fikri ile ortaya çıkmış bir teknolojidir. HTTP üzerinden yapılan bu çağrılara karşı nesne erişim protokolleri olan SOAP (Simple Object Access Protocol) ya da REST (Representational State Transfer) ile XML ya da JSON formatında çıktıları üretmektedir (Çelik, 2008).

Şekil 3.6.'da XML-RPC çalışma mantığını gösteren bir görsel verilmiştir. odoorpc kütüphanesi Odoo yazılım çatısı üzerindeki veriye RPC aracılığıyla müdahale etmeyi sağlamaktadır. Bu müdahale temel anlamda okuma, yazma, silme ve düzenleme işlemleridir. Yazılım çatısı üzerinde kod yazmaksızın sadece kütüphane kullanılarak bu 4 temel fonksiyon ile işlemler hızlı bir şekilde yapılabilmektedir. Veri tabanındaki verilere ulaşmak için Odoo ORM'sinin (Object Relation Map) sağladığı tüm ekipmanlar bu kütüphane aracılığıyla kullanılmaktadır. Gerekli olduğu durumlarda veriler filtrelenerek işlem yapmak da mümkündür.



Şekil 3.6. XML-RPC çalışma mantığı

3.2.2. Odoo yazılım çatısı

Odoo, Python programlama dili aracılığıyla geliştirilmiş bir yazılım çatısıdır. Dünya üzerinde dört milyonu aşkın kurumsal kullanıcısı bulunmaktadır. Odoo'nun esnek yapısı ve geliştiricilere sunduğu avantajlardan dolayı, son yıllarda farklı amaçlara yönelik yazılımlar geliştirmede Python çatıları arasında önemli bir yer almış durumdadır.

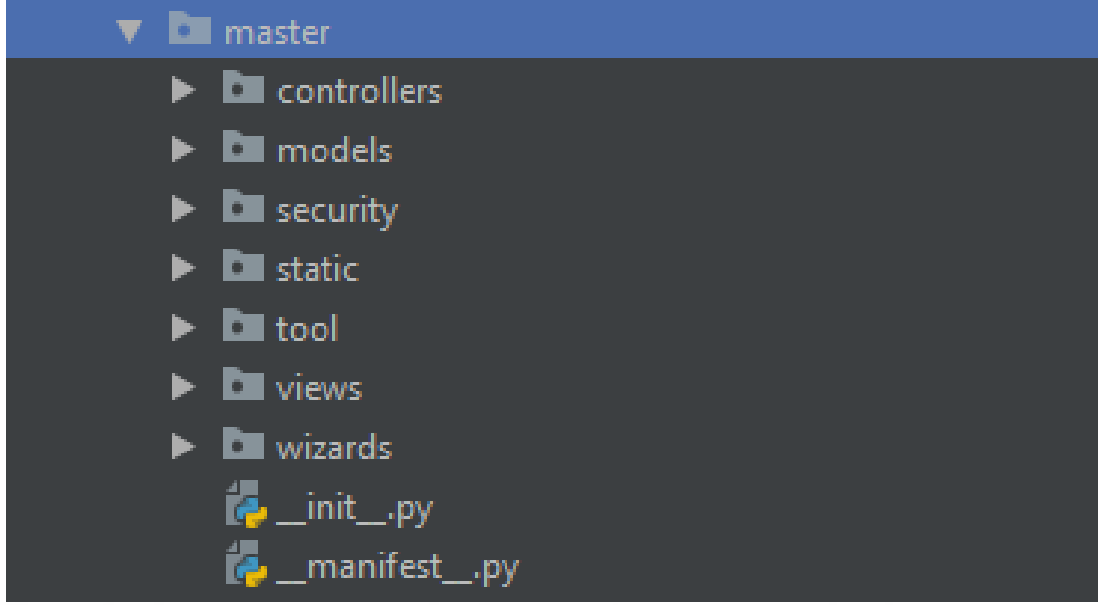
Sektörde çok önemli bir yere sahip olan Odoo, bütün bu alanlarda yüksek performans sunar. Yapısındaki modülerlik sayesinde geliştiricilere büyük kolaylıklar sağlamıştır. Açık kaynaklı yazılımlar için erişilebilirlik, esneklik ve kullanım kolaylığı gibi kelimeler anahtar niteliğindedirler. Bu anahtar kelimelere sahip olan Odoo, kendi içindeki temel modüllerin yanında hem harici modül geliştirmeye imkân sağlamak hem de esnekliği sayesinde kendi modüllerine eklenti geliştirerek genişletmeyi de mümkün kılmaktadır.

Odoo, Türkçe dilinin de arasında bulunduğu yirmi üç farklı dile destek vermektedir. Yazılım altyapısında Python kullanan Odoo kullanıcı ara yüzü tarafında XML, XSLT (Extensible Stylesheet Language Transformations) kullanmaktadır. Veri tabanı çözümü olarak ise PostgreSQL tercih etmektedir.

3.2.2.1. Modül oluşturma

Odoo, indirildiğinde, içerisinde onlarca ücretsiz modülle beraber gelmektedir. Bu modüller temel bir ERP yazılımında olması gereken birçok uygulamayı kapsamaktadır. Aynı zamanda bu modüllere ek olarak Odoo market içerisinde yüzlerce üçüncü parti yazılım modülü bulunmaktadır. Bu modüllerden bazıları Odoo' da bulunmayan özellikleri barındırırken, bazılarının ise hali hazırda içerilen özellikleri geliştirmeye yönelik oluşturulduğu görülmektedir.

Odoo üzerinde geliştirme yapabilmek için öncelikle bir modül oluşturmak gerekmektedir. Modül içerisinde, modül hakkındaki temel bilgileri içeren bir manifest dosyası ve bir init dosyası bulunması gerekmektedir.



Şekil 3.7. Odoo örnek modül yapısı

Şekil 3.7.'de araştırma için oluşturulmuş "master" modülünün dosya yapısı görülmektedir. "views" klasörü, ekranların tasarımlarının tanımladığı XML tipindeki dosyalardan oluşmaktadır. "security" klasörü, modüllerle ilgili yetki grup tanımlamalarını, yetki gruplarının erişim izinlerinin tanımlandığı XML ve csv tipindeki dosyaları içermektedir. Diğer klasörlerde ise, modülün model ve yazılımsal tanımlarının bulunduğu Python dosyaları bulunmaktadır.

```

{
  'name': 'Master',
  'version': '1.0',
  'summary': 'Yüksek lisans bibliyometrik analiz.',
  'description': 'Yüksek lisans için hazırlanan internetteki indeks veritabanlarından indirilmiş makale bilgilerini'
    '(makale,okul,yazar,anahtar kelime vb.) bilgileri saklayabilme ve inceleyebilmeyi sağlayan bir modüldür.',
  'category': 'Education',
  'depends': [
    'mail'
  ],
  'data': [
    'security/crm_group.xml',
    'security/ir.model.access.csv',
    'security/access_rules.xml',
    'views/article.xml',
    'views/author.xml',
    'views/university.xml',
    'views/publisher.xml',
    'views/article_university_author.xml',
    'wizards/views/university_partner.xml',
    'views/menu_item.xml',
  ],
  'demo': [
  ],
  'qweb': [
  ],
  'installable': True,
  'application': True,
  'auto_install': False,
}

```

Şekil 3.8. Manifest dosyası içerik örneği

Şekil 3.8.'de örnek bir manifest dosya içeriği görülmektedir. Manifest dosyası dict tipinde bir objeden meydana gelmektedir. Daha önce de bahsedildiği gibi modül hakkındaki temel bilgileri içermektedir.

3.2.2.2. Model oluşturma

Odoonun temel bileşeni, ORM katmanıdır. Bu katman, basit yapıdaki SQL sorgularını kendisi oluşturabilmektedir. Aynı zamanda, SQL injection gibi güvenlik açıklarını engellemektedir.

İş objeleri, Python sınıflarının Odoonun kütüphanesi içerisinde bulunmakta olan "Model" sınıfını miras alması sonucu oluşmaktadır.

Modeller tanımlarında bir dizi özellik ayarlayarak yapılandırılabilir. En önemli özelliği "_name" özelliğidir. Odoonun sistemi üzerinde olan tüm modellerin içermesi gereken bu özellik, zorunlu alandır ve modelin veri tabanındaki tablo ismini belirtmektedir.

```
from odoo import models
class MinimalModel(models.Model):
    _name = 'test.model'
```

Şekil 3.9. Minimum model tanımlama örneği

Şekil 3.9. bir modeli tanımlamak için gerekli minimum kod içeren bir dosya içeriğini göstermektedir. Fakat model dosyaları sadece isimden oluşmaz ve mutlaka field adı verilen alan bilgilerinin de yer alması gerekir. Fieldlar modelin ne tipte ve nerede veri saklayacağını belirtmektedir. Veri tabanı tablosundaki kolonlara karşılık gelmektedir. Aynı modelde olduğu gibi, model özellikleri de parametreler aracılığı ile konfigüre edilebilmektedir. Bazı parametreler tüm fieldlar için kullanılabilir.

3.2.2.3. View oluşturma

Viewlar, bir modelin kayıtlarının görüntülenme şeklini tanımlamaktadır. Her view türü görselleştirme modunu (tree, form, search, graph, kanban, etc.) temsil etmektedir. Viewlar genel olarak türleriyle veya benzersiz tanımlanmış "id" değerleriyle çağrılabilirler. Türleriyle çağrılmaları durumunda aynı türe ait olan en düşük öncelik değerine sahip view otomatik olarak getirilecektir.

Bir view "ir.ui.view" modelinin bir kaydı olarak tanımlanmaktadır. View tanımlaması sırasında XML içerisinde "arch" elementinin içerisinde tanımlanmaktadır.

```
<record model="ir.ui.view" id="view_id">
  <field name="name">view.name</field>
  <field name="model">object_name</field>
  <field name="priority" eval="16"/>
  <field name="arch" type="xml">
    <!-- view content: <form>, <tree>, <graph>, ... -->
  </field>
</record>
```

Şekil 3.10. Örnek bir view XML kaydı

Şekil 3.10.' da örnek bir view kaydı gösterilmektedir.

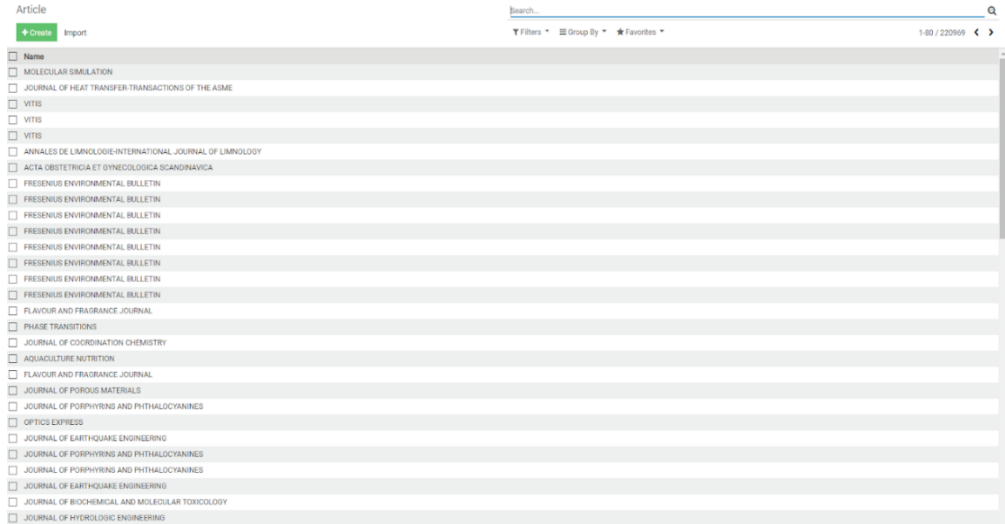
3.2.2.4. Tree View oluşturma

“Tree view” aynı zamanda “list view” olarakta bilinmektedir. Kayıtları liste halinde göstermek amacıyla kullanılmaktadır.

```
<record id="master_article_tree" model="ir.ui.view">
  <field name="name">master.article.tree</field>
  <field name="model">master.article</field>
  <field name="arch" type="xml">
    <tree string="Article">
      <field name="name"/>
    </tree>
  </field>
</record>
```

Şekil 3.11. Tree view kullanım örneği

Şekil 3.11.'de görüldüğü üzere “arch” isimli alan kök elementi “tree” olan içerisinde gösterilmek istenilen alanların bulunduğu bir XML yapı almaktadır. “master.article” isimli modelin “name” alanını göstermek için hazırlanan yapının çıktısı Şekil 3.12.'de görüldüğü gibidir.



Şekil 3.12. Tree view örnek çıktısı

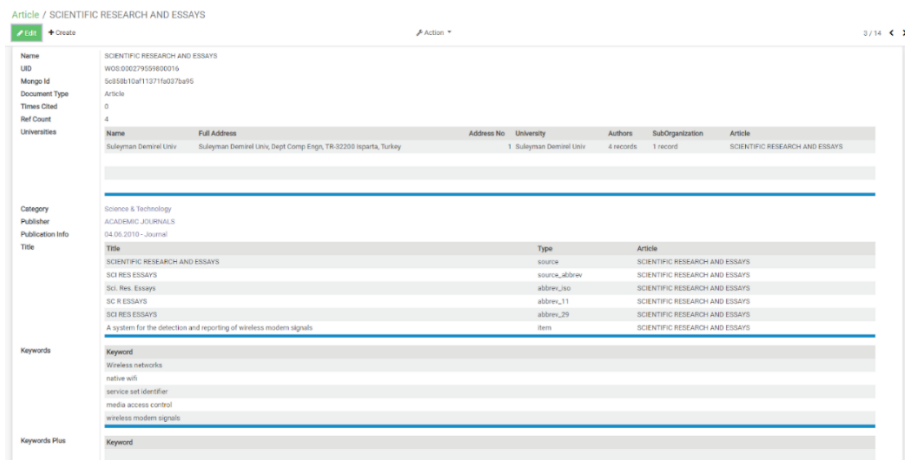
3.2.2.5. Form View oluşturma

“Form view” tekil kayıtları oluşturmak veya düzenlemek amacıyla kullanılmaktadır.

```
<record id="master_article_form" model="ir.ui.view">
  <field name="name">master.article.form</field>
  <field name="model">master.article</field>
  <field name="arch" type="xml">
    <form>
      <sheet>
        <group>
          <field name="name"/>
          <field name="uid"/>
          <field name="mongo_id"/>
          <field name="document_type"/>
          <field name="times_cited_count"/>
          <field name="ref_count"/>
          <field name="university_ids"/>
          <field name="category_id"/>
          <field name="publisher_id"/>
          <field name="pub_info_id"/>
          <field name="title_ids"/>
          <field name="keyword_ids"/>
          <field name="keyword_plus_ids"/>
          <field name="language_ids"/>
        </group>
      </sheet>
    </form>
  </field>
</record>
```

Şekil 3.13. Form view kullanım örneği

Şekil 3.13.'de görüldüğü üzere “arch” isimli alan kök elementi “form” olan içerisinde gösterilmek istenilen alanların bulunduğu bir XML yapı almaktadır. “master.article” isimli modelinin alanlarını düzenleme ve oluşturmak için hazırlanan yapının çıktısı Şekil 3.14.'de görüldüğü gibidir.



Name	Full Address	Address No	University	Authors	SubOrganization	Article
Süleyman Demirel Univ.	Süleyman Demirel Univ, Dept Comp Engn, TR-32200 İsparta, Turkey	1	Süleyman Demirel Univ	4 records	1 record	SCIENTIFIC RESEARCH AND ESSAYS

Title	Type	Article
SCIENTIFIC RESEARCH AND ESSAYS	source	SCIENTIFIC RESEARCH AND ESSAYS
SCI RES ESSAYS	source_abstract	SCIENTIFIC RESEARCH AND ESSAYS
Sci. Res. Essays	abstract	SCIENTIFIC RESEARCH AND ESSAYS
SC R ESSAYS	abstract_11	SCIENTIFIC RESEARCH AND ESSAYS
SCI RES ESSAYS	abstract_29	SCIENTIFIC RESEARCH AND ESSAYS
A system for the detection and reporting of wireless modem signals	item	SCIENTIFIC RESEARCH AND ESSAYS

Keyword
Wireless networks
networks
networks
networks
media access control
wireless modem signals

Şekil 3.14. Form view örnek çıktısı

3.2.3. Web servis

Günümüzde oldukça popüler olmaya başlayan web servis kavramı, yetkilendirme, oturum açma, veriye erişme, herhangi bir yazılım fonksiyonunu başka bir yazılıma verme gibi işlemlerin HTTP(S) aracılığıyla kullanımını ifade etmektedir. Başka bir ifadeyle web servis, yerel nesnelerin dağıtık nesnelere haline getirilmesi için kullanılan bir araçtır. Farklı tipteki platformlar arasında iletişimi kolaylaştırmaktadır. Web, mobil, masaüstü yazılımlar arasındaki iletişimi, hangi dilde yazıldığı fark etmeksizin sağlamakta ve ortak bir paylaşım noktası oluşturmaktadır. Web servis, veriyi anlaşılabilir bir formatta iletebilmek için XML veya JSON dosya formatlarını kullanmaktadır (Cömert vd., 2015).

Servis Yönetimli Mimaride (SOA, Service Oriented Architecture), istemcinin sadece ihtiyacı olan bilgiye ulaşması gerekmektedir. Servis ile ilgili kritik bilgileri bilmemelidir. Geliştirici tarafından veri tabanı, varsa kripto bilgilerinin, yetkilendirme yöntemi ile saklanması ve ulaşılamaz olması sağlanmalıdır (Satoh vd., 2008).

3.2.3.1. REST web servis

Representational State Transfer (Temsili Durum Transferi) kelimelerinin kısaltılması ile REST olarak adlandırılır. Modern Web servis mimarilerinde REST servis kullanımı çok yaygındır. HTTP(S) protokolünü kullanmaktadır. Dağıtık hypermedia sistemleri için kullanılan bir mimari olup SOAP tabanlı mimarinin güçlüklerini ortadan kaldırmak amacıyla ortaya çıkmıştır. İlk defa 2000 yılında Roy Thomas Fielding'in doktora tezinde tanımlanmış yazılım mimarisidir.

SOAP mimarisine göre daha sade, geliştirmesi kolay ve daha hızlı çalışmaktadır. SOAP gibi belli bir standardı bulunmamaktadır. SOAP ile ortak yanları ise ikisinin de HTTP(S) kullanmasıdır. İki web servis mimarisi de HTTP(S) kullanmaktadır. SOAP sunucuya veri gönderirken WSDL olarak tanımlanan XML'leri kullanmaktadır. Bu XML'lerin içerisinde standart şekilde tanımlanmış veriye göre, sunucu ve istemci nasıl işlem yapacağını bilmektedir. REST mimarisinde bu

şekilde bir standart bulunmamaktadır. İstemci sunucuya GET, POST, PUT ve DELETE tiplerinde istek gönderebilir. Bu isteklere karşılık sunucu üzerindeki kod parçacığı veya parçacıkları tetiklenerek işlem gerçekleştirilip sonuç dönülmektedir. Bu sayede proxy ihtiyacı ortadan kaldırılmış olup platform bağımsızlığı sağlanmaktadır (Seyrek vd., 2020).

Aynı zamanda REST mimarisinin tercih edilmesinin en büyük sebeplerinden bir tanesi de birçok yazılım geliştirme aracının harici bir kütüphaneye ihtiyaç duymadan çalıştırabilmesi ve diğer mimarilere göre esnek, öğrenilebilir ve uygulanabilir olmasıdır.

3.2.3.2. SOAP api

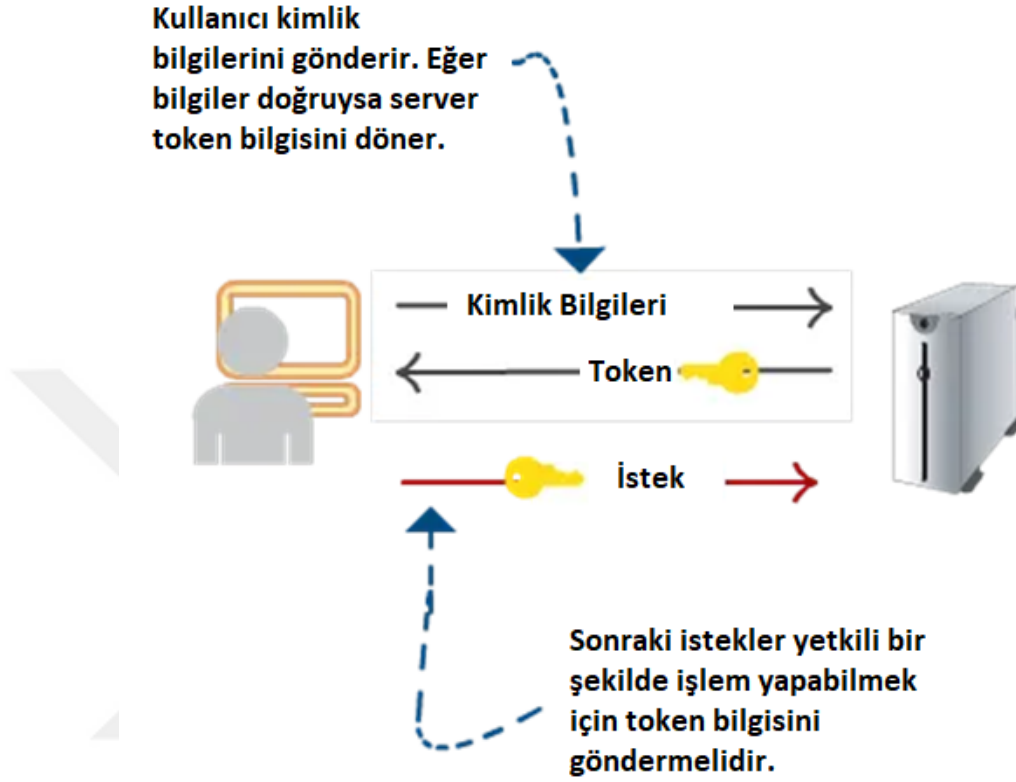
SOAP, Simple Object Access Protocol kelimelerinin baş harfleri kullanılarak kısaltılmış halidir. Türkçe karşılığı basit nesne erişim protokolüdür. HTTP protokolü üzerinden bir istemci ile sunucunun haberleşmesini sağlayan, XML tabanlı mesajlar üzerinden yürümekte olan servis protokolüdür. SOAP servisi hakkında kullanım bilgilerine erişmek için, nasıl kullanıldığı, hangi parametreleri aldığı, ne türde veri geri döndürdüğü gibi bilgileri içeren WSDL dokümanı incelenebilir.

SOAP servis 3 temel bileşenden oluşur. Bunlar “envelope”, “header” ve “body” dir (Louridas, 2006). “Envelope”, XML’in tanımlandığı kısımdır. Kök element olarak da adlandırılmaktadır. “Header”, servis için meta-data gibi bazı bilgileri iletmekte kullanılmaktadır. Kullanımı zorunlu değildir. “Body”, Soap isteği için olmazsa olmazdır. İstekle ilgili gerekli bilgiler içerisinde tanımlanmaktadır. Dönen cevap içerisinde de bulunmaktadır (Mumbaikar vd., 2013).

3.2.4. Token based authentication

Web servisler aracılığı ile birçok işlem yapılabilmektedir. Bu işlemlerin bazıları güvenlik gerektirmeyen herkese açık işlemler iken, bazı işlemler güvenlik gerektirmektedir. Bu güvenliği sağlamak amacı ile yetkilendirme süreçleri

gerçekleştirilir. İstemciyi yetkilendirmek amacı ile çok fazla yöntem bulunmaktadır. Bunlardan bir tanesi de “token based authentication” metodolojisidir.



Şekil 3.15. Token based authentication yaşam döngüsü

Şekil 3.15.'de görülen Token based authentication yaşam döngüsü incelendiğinde şu şekilde işlediği görülmektedir.

- İstemci yetki alabilmek için gerekli güvenlik bilgilerini sunucuya göndermektedir.
- Eğer güvenlik bilgileri doğru ise yetkilendirme sunucusu istemciye bir token cevabı döndürmektedir.
- İstemcinin elde ettiği token, istek “header”ı içerisinde eklenerek yetki gerektiren işlemler tamamlanmaktadır.

3.2.5. Veri tabanı araçları

Veri tabanı en basit tanımı ile, birbiriyle ilişkili olan yada olmayan verilerin birlikte tutulup yönetildiği, üzerinde silme, güncelleme gibi işlemlerin yapılabilirdiği, taşınabilen bilgilerin saklanabildiği yazılımlardır (Garcia Molina vd., 2009).

Tarihçesi incelendiğinde, 1960'lı yıllarda kullanımının başladığı görülmektedir. O senelerde özel kuruluşlar tarafından maliyeti azaltması nedeniyle tercih edildiği görülmektedir.

Veri tabanlarının ilk kullanımından günümüze kadar geçen süreçte çok çeşitli mimariler kullanılmıştır. Dosya veri tabanları kullanılan ilk mimaridir. Bugün baktığımızda en çok kullanılan veri tabanı mimarisi ilişkisel ve nesne ilişkisel mimaridir. Geçmişten günümüze kullanılan veri tabanı mimarileri ise şu şekildedir (Önder, 2005):

- Dosya Veri Tabanları
- Hiyerarşik Veri Tabanları
- Ağ Veri Tabanları
- İlişkisel Veri Tabanı Mimarisi
- Nesne Yönelimli Veri Tabanı Mimarisi
- Nesne İlişkisel Veri Tabanı Mimarisi
- Çoklu Ortam Veri Tabanları
- Dağıtık Veri Tabanları
- İlişkisel Olmayan Veri Tabanları

Günümüzde veri tabanı neredeyse her türlü uygulama için neredeyse zorunluluk seviyesinde bir ihtiyaç olmaktadır. En büyük web sitelerden ufak çaplı web sitelerine, mobil, masaüstü uygulamalarına kadar her uygulamanın veriyi aklında tutması gerekmektedir. Uygulamaların verileri hatırlaması için veriyi saklayabileceği bir mekanizma gereklidir. Bu mekanizmayı sağlayan yapılara veri tabanı yönetim sistemi denilmektedir.

Veri tabanı yönetim sistemleri, büyük boyuttaki verileri etkin bir şekilde yönetmek ve uzun süre boyunca güvenli bir şekilde saklama görevlerini sağlamaktadır.

3.2.5.1. İlişkisel veri tabanı mimarisi

1970 yılında IBM laboratuvarında çalışmakta olan matematikçi Edward Frank Codd'un hazırladığı bir araştırma yazısıyla birlikte ilişkisel veri tabanı mimarisi ortaya çıkmıştır. Ağ ve hiyerarşik yapılı veri tabanlarını kullanabilmek için kullanıcıların veri tabanı yapısını öğrenmesi gerekmektedir. Codd, bu gerekliliği ortadan kaldırmak, kayıt, veri tekrarının sorunlarına temel bir çözüm geliştirmek ve fiziksel uygulama üzerindeki bağımlılığı ortadan kaldırmak amacıyla bu bilimsel araştırmaya başlamıştır.

Bu araştırma, matematikteki iki dalı temel almaktadır: Önerme mantığı ve küme teorisi. Küme teorisinin bir bileşeni olan "küme ilişkileri" ilişkisel veri tabanı ifadesine isim verilirken ilham kaynağı olmuştur. Verilerin bağımsızlığını kolaylaştırdığı için verilerin kayıtlar yerine ilişkiler ya da tablolar içerisinde saklanması araştırmasında savunulmuştur (Önder, 2005).

İlişkisel veri tabanı mimarisini diğer mimarilerden ayırmakta olan iki büyük özelliği çok büyük önem arz etmektedir. Bunlardan birincisi, çok büyük ve yaygın bir kullanım alanı olması ve bağımsız veri tabanı tasarımında kullanılmasıdır. İkincisi ise veri tabanı yönetim sistemlerinin temelini atmış olmasıdır (Kroenke ve Auer, 1999).

Yaygın kullanılan ilişkisel veri tabanları şu şekildedir:

- Potgresql
- Mysql
- Mssql
- Oracle
- Firebird
- DB2

- SQLite

3.2.5.2. Postgresql

Postgresql; güçlü, açık kaynak kodlu üzerinde 30 yılı aşkın süredir aktif geliştirme yapılan veri tabanı yönetim sistemidir. Bu geliştirmeler sonucunda kazanılan deneyim sayesinde güvenilirliği, sağlamlığı ve performans açısından gelişmiştir. 1986 yılında California Üniversitesinde proje olarak başlatılmıştır. Çok güçlü bir işlem yapısına sahiptir ve verinin güvenliği daima ön planda tutulmaktadır. Tablo için maksimum 32 TB, satır için maksimum 1.6 TB, alan için maksimum 1 GB veri saklayabilmektedir. Sınırsız sayıda tablo, satır ve index oluşturmaya imkân sağlamaktadır. “Inheritance”, “Transaction” “trigger” ve “stored procedure” özelliklerine sahiptir (PostgreSQL).

3.2.5.3. Yapısal olmayan veri tabanı mimarisi

İlişkisel olmayan veri tabanı mimarisinde veri bölümleri arasında bir ilişki bulunmamaktadır. Dolayısıyla verilerin farklı veri bölümlerinde tekrar ettiği görülmektedir. Verilerin başka tablolarla benzersiz keyleriyle bir ilişkisi olmadığı için ve sorgular tek bir yer üzerinden yapılacağından sonuçlar daha kısa sürede gelmektedir.

İlişkisel olmayan veri tabanlarının tarihçesine bakıldığında ilişkisel veri tabanları kadar eski olduğu görülmektedir. Son 10-15 yıla kadar çok büyük veri kümeleri olmadığı için yakın zamana kadar çok popüler değillerdi (Bolat, 2019).

İlişkisel olmayan veri tabanlarına genellikle ilişkisel veri tabanlarının tersine büyük verilerde ihtiyaç duyulmaktadır. Veriler birden fazla tablodan çekilmek yerine tek bir veri kümesinin içerisinden sorgulanarak getirileceği için daha hızlı sonuç elde edilmektedir.

İlişkisel olmayan veri tabanları Nosql olarak ta adlandırılmaktadır. Nosql “Not Only Sql (Sadece SQL Değil)” ifadesinden akıllarda kalmaktadır. Bu ifadeden anlaşılacağı üzere farklı bir yapı olduğu görülmektedir. Çok fazla artan veri

miktarı ve veriye hızlı ulaşma ihtiyacı sonucunda günümüzde oldukça popülerleşmiştir. İlişkisel olmayan veri tabanları ilişkisel veri tabanlarını reddetmenin aksine eksikliğini kapatmaktadır.

Nosql veritabanları BASE (Basically Available, Soft state, Eventually consistent) özelliklerini kullanılmaktadır (Yıldırım, 2019):

- Kolay Ulaşılabilir: Veri erişim problemini ortadan kaldırmak için birden fazla sunucuda kopyası bulunan veriyi parça parça olarak veriye kolay ulaşılabilirliği sağlamaktadır.
- Esnek Durum: İlişkisel veri tabanları için veri tutarlılığı çok önemlidir ve esnekleştirilemez. Nosql veri tabanlarında veri sürekliliği ve tutarsızlığı çok mühim değildir, barındırılmasına izin verilir.
- Eninde sonunda Tutarlı: Nosql veri tabanlarında verilerin belirli olmayan bir tarihte tutarlılığa ulaşacağı farz edilir.

3.2.5.4. MongoDB

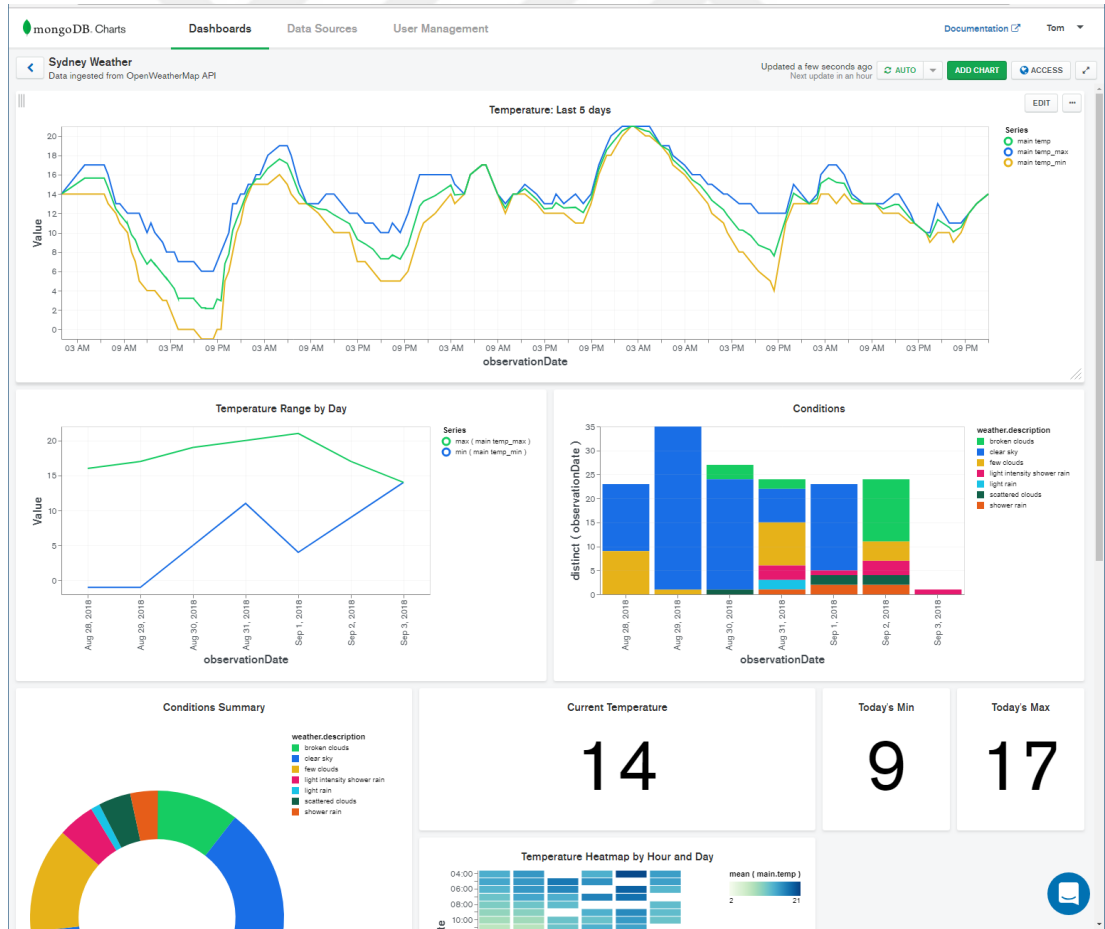
MongoDB, esnek ve ölçeklenebilir, gerektiğinde sorgulama ve indeksleme işlemlerine imkan sağlayan, modern uygulama geliştiriciler ve bulut dönemi için oluşturulmuş genel amaçlı, belge tabanlı dağıtılmış bir NoSql veri tabanıdır (MongoDB, 2019).

MongoDB bir belge veri tabanıdır. Bunun anlamı JSON benzeri bir yapıdaki veri belgeleri depolanmaktadır. Geliştiricileri, verileri düşünmenin en doğal yolunun bu olduğuna ve geleneksel veri saklama yöntemi olan satır / sütun modunda çok daha etkileyici ve etkin bir yöntem olduğuna inanmaktadır.

Kullanıcılarına güçlü bir sorgu dili sunmaktadır. Bu sorgulama dilinin avantajları şu şekilde sıralanmaktadır.

- Bir belge ne kadar iç içe yerleştirilmiş olursa olsun, herhangi bir alana göre filtrelemeye ve sıralamaya imkân sağlayan zengin ve etkileyici bir sorgu dili.
- Toplama ve coğrafi tabanlı arama, grafik araması ve metin araması gibi diğer modern kullanım durumları için destek sağlama.
- Sorguların kendisi JSON'dur ve bu nedenle kolayca oluşturulabilir. Dinamik olarak SQL sorguları oluşturmak için dizeleri birleştirmeye artık gerek yok.

MongoDB, geliştiricilerden analistlere ve veri bilimcilerine kadar herkese verilerle çalışmayı oldukça kolay hale getirmek için kapsamlı bir araç setine sahip gerçek bir veri platformudur. Aynı zamanda Dünya çapında altmıştan fazla bulut bölgesinde çalışmaya imkân sağlamaktadır.



Şekil 3.16. MongoDB chart örneği

Şekil 3.16.'da örneği görülen MongoDB Chart, MongoDB içerisinde bulunan veriyi görselleştirmek için en iyi yollardan bir tanesidir. İstenilen herhangi bir veri kaynağına bağlamak mümkündür. Bağladıktan sonra grafik ve çizelgeleri oluşturabilmekte, sonrasında uygulamalara eklenebilmektedir. MongoDB Charts, iç içe ve hiyerarşik veriler dahil olmak üzere belge modelinin zenginliğini desteklemektedir ve verilerden daha hızlı değer elde edebilmek amacıyla her veri türü için anlamlı işlemler sunmaktadır. MongoDB Chart, şirket içi dağıtım için kullanılabilir ve MongoDB Atlas ile tam olarak entegre edilmiştir. Böylece, herhangi bir harici yazılım veya altyapıya ihtiyaç olmadan hızlı bir şekilde veri görselleştirmeleri oluşturmaya olanak tanımaktadır (MongoDB, 2019).

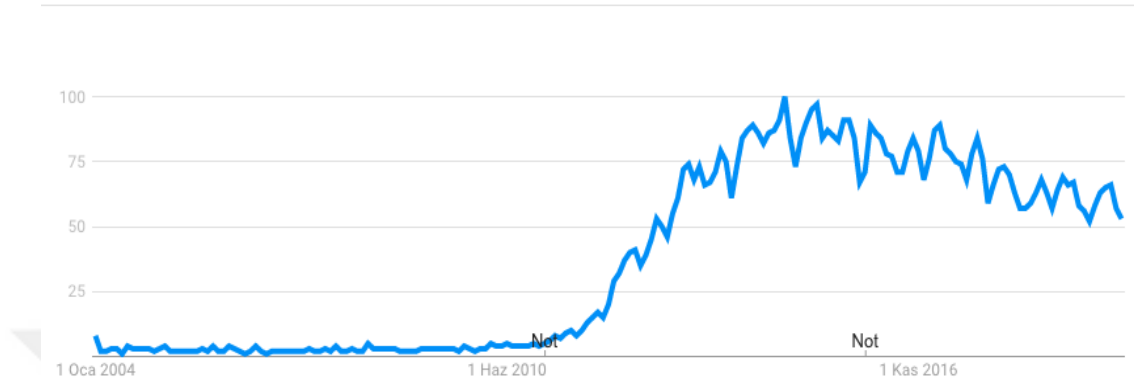
3.3. Büyük Veri

Büyük veri halihazırdaki sistemlerin işleyemeceği hatta saklayamayacağı kadar geniş ve farklı türdeki veri kümelerine verilen genel bir addır. Bir başka deyişle bilinen veri tabanı yönetim sistemlerinin, yazılım araçlarının veri toplama, saklama yönetme ve çözümlene gibi yeteneklerinin yetersiz kaldığı büyüklükteki verilere büyük veri denilmektedir (Özdeş, 2017).

Teknolojinin gelişmesi aynı zamanda yaygın kullanılmasıyla birlikte sensör bilgisi, fotoğraflar, metin, ses, video gibi birçok veri kullanıcılar tarafından üretilmektedir. Bu ortaya çıkan veri işlenip anlamlandırıldığında içerdiği gizli bilgileri ortaya çıkartmak günümüzde çok büyük önem arz etmektedir. Bu veriyi anlamlandırmak neredeyse bütün sektörlerde devrim yaratan bir çağ başlatmak gibi nitelendirilmektedir. Özellikle iş dünyasında veriler anlamlandırıldığında şirketlerin, müşterilerini tanıyarak nelere ne zaman ihtiyaç duyulacağını, hangi müşteri profillerinin neleri alabilme potansiyeli taşıdığını önceden tahmin etmelerine imkân sağlamaktadır.

Bilgi çağında verilerin ortaya çıkışı çok büyük bir hızla artmaktadır. Verinin hacminin büyümesi önceden 1, 2, 3... şeklinde aritmetik dizi katları halinde artmakta iken günümüzde bu hız geometrik dizi katları şeklinde 3, 9, 12... gibi

artmaktadır (Akgül, 2020). Google üzerinde “büyük veri” aramaları incelendiğinde 2004-2010 yılları arasında belirli bir seviyede seyrederken 2010 yılından sonra ciddi bir artış yaşanmıştır. Bu artış Şekil 3.17.’deki grafikte daha net görülmektedir.



Şekil 3.17. Büyük veri kelimesinin 2004-2020 yılları arasındaki aranma sıklığı.

3.3.1. Büyük verinin karakteri

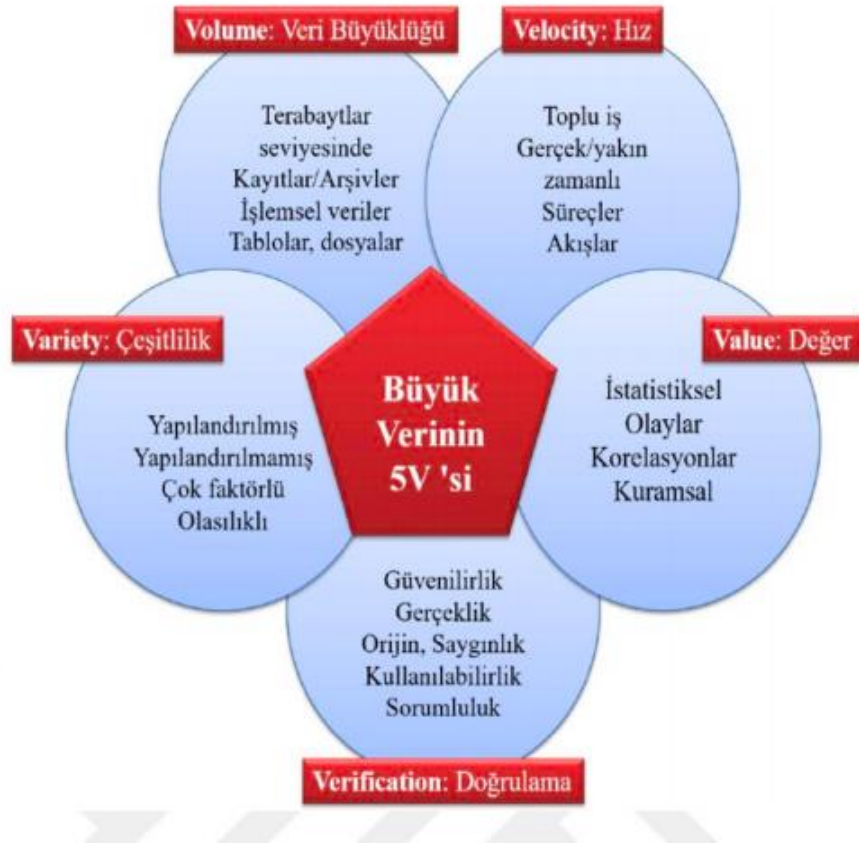
Bir veri kümesini büyük veri olarak adlandırabilmek için sadece boyut olarak büyük olması yeterli değildir. 2001 yılında Doug Lane tarafından hazırlanan araştırma raporunda, büyük veri üç boyutlu olarak yani 3V şeklinde tanımlanmıştır. Verideki değişimleri ve verinin bileşenlerini inceleyen Loan' a göre bunlar şu şekildedir (Laney, 2001):

- Verinin Hızı
- Verinin Büyüklüğü
- Verinin Çeşitliliği

Bu araştırma sonrasında yapılan bazı araştırmalar bu tanımlamalara yenilerini ekledi. Bunlar ise şu şekildedir:

- Verinin Doğruluğu
- Verinin Değeri

Yukarıda bahsedilen tanımlamaların bütünü büyük verinin bileşenleri olarak anılmaktadır.



Şekil 3.18. Büyük veri bileşenleri (5V)(Özdeş, 2017)

Şekil 3.18.'de büyük verinin 5 karakterinin özeti yer almaktadır.

3.3.1.1. Veri büyüklüğü

Veri büyüklüğü, verinin en çok bilinen ve veri miktarını tarif etmekte olan özelliğidir. Çok da uzak olmayan geçmişe bakıldığında, 250 terabyte boyutundaki veri çok büyük görünürken günümüzde petabyte ve üzeri veri büyük olarak adlandırılmaktadır. Gün geçtikçe teknolojinin hızlı ilerlemesi internetin yoğun kullanımı sonucu ortaya inanılmaz derece büyük veri çıkmaktadır. Üretilen bu verinin boyutu şaşırtıcı miktarlara ulaşmış bulunmaktadır. Verilerin boyutunun exabyte ve zettabyte seviyelerine ulaştığı bir zamana gelmiş bulunmaktayız. 2025

yılında dünya üzerinde yaşayan insanların beyin kapasitelerini aşan bir veri büyüklüğü olacağı hesaplanmaktadır (Davis, 2012).

3.3.1.2. Veri çeşitliliği

Veri çeşitliliği verinin birden fazla kaynaktan alındığını, farklı formatlarda ve yapılarda olduğunu temsil etmektedir. Bir kişinin sosyal medyadan, akıllı telefonundan, kullandığı diğer cihaz ve sensörlerden alınan verisi buna örnek verilebilir. Farklı ortam ve kaynaklardan alınan veriler yapısal, yarı yapısal ve yapısal olmayan türde farklı formlarda kategorize edilmektedir (Inc., 2012). Yapısal olan veriler genelde ilişkisel veri tabanlarından gelen veriler tarafından oluşturulmaktadır. Yarı yapısal verilere ise yazılımların ve cihazların logları olarak örneklendirilebilir. Yapısal olmayan veriler ise görüntü, ses, video gibi çıktıları kapsamaktadır. 2003 yılında Soquel grubun yapmış olduğu araştırma sonucunda yapısal olmayan verinin oranının %85'den fazla olduğu ifade edilmektedir (Blumberg ve Atre, 2003). 2010 yılında Cukier bu oranın %95 olduğunu gösteren bir makale ele almıştır (Cukier, 2010). Yapısal veriler üzerinde çalışma yapmak diğer formatlar üzerinde çalışmaktan daha kolaydır. Belli bir düzene ve formata göre saklanan veriler tutarlı oldukları için, üzerinde çalışacak araştırmacının işini birçok noktada kolaylaştırmaktadır.

3.3.1.3. Veri hızı

Teknolojinin baş döndüren bir hızla geliştiği günümüzde, günlük hayatta kullandığımız hemen hemen tüm cihazlar da dahil olmak üzere dünya üzerindeki her şey bir veri üretmektedir. Bu verilerin hızlı bir şekilde analiz edilebilmesi için de büyük veri teknolojilerinin aynı hızda gelişmesi gerekmektedir.

3.3.2. Büyük verinin avantajları

İnsanlık yenilikçi bir türdür. Dijitalleşme, bunun en büyük kanıtlarından bir tanesidir. Dünyanın her geçen gün daha dijital bir ortam olması inkâr edilemez

bir gerçektir. Bundan 10 yıl öncesine kadar insanlar dijital ortamda isimlerini paylaşmaya korkmaktayken; günümüzde tabletler, akıllı cihazlar, bilgisayarlar aracılığıyla internet üzerinden alışveriş yapabilmekte, gidecekleri adresi bulabilmekte, arkadaşlarıyla iletişime geçebilmekte ve çok daha fazlasını yapabilmektedir.

Global seviyede üretilen ve depolanmakta olan verinin miktarı inanılmaz derecede büyük ve gün geçtikçe rasyonel olarak artmaktadır. Çok az bir kısmı analiz edilmiş verilerden bile kritik sonuçlar ve kararlar almak mümkündür. Büyük veri temelde ne kadar veriye sahip olduğundan daha çok bu veriyle neler yapılabileceğiyle ilgilenmektedir.

Büyük veri büyük resmin tamamını görmeye imkân sağlamaktadır. Organizasyonların hizmet verdikleri kişilerin isteklerini anlamalarına imkân sağlamakta, iyi strateji seçmelerini imkân vermekte, kaynaklarını daha etkin bir şekilde ayırmalarını ve bu kaynağı akıllı bir şekilde kullanmalarına imkân sağlamaktadır.

Örnek vermek gerekirse, günümüzün en önemli konularından bir tanesi olan siber güvenlik konusunda oluşan büyük veri incelenerek, bu alanla ilgili önlemler alınması konusunda çalışmalar yapılmaktadır. Büyük veri analizleri kullanılarak, sistem açıklarının tespiti, sistem üzerindeki güvenlik ihlallerini tanımlama işlemleri bir süredir yapılmaktadır (Ismail, 2017).

Diğer faydaları şu şekilde sıralanabilir:

- Bilgi yönetimini geliştirmektedir.
- Operasyonların bakım ve verimliliğini arttırmaktadır.
- Süreçler sırasında hızlı tepki verme avantajı sağlamaktadır.
- Ürün ve market stratejisini geliştirmektedir.
- Gelişmiş talep yönetimi ve ürün planlaması yapmayı sağlamaktadır.
- Finansal anlamda büyük faydalar sağlamaktadır.
- Daha fazla entegrasyon ve iş birliği yapmayı sağlamaktadır.

- Gelişmiş risk yönetim sistemi sağlamaktadır.

3.3.3. Büyük verinin zorlukları

Büyük verinin hızlı bir şekilde büyümesi, veri setinin yönetilmesi aşamasında birçok problemi beraberinde getirmektedir. Bu problemlerden başlıcaları; verinin depolanması, toplanması, bütünlüğün korunması, ölçeklenebilirliği, niteliği ve grafikleştirilmesi olarak sıralanabilir.

Çeşitli cihaz, makina ve araçlardan ortaya çıkan veriler sürekli olarak artmaktadır. Her üç yılda bir veri miktarı kendini ikiye katlamakta olduğu görülmektedir (Hilbert ve López, 2011). Depolama ve yönetim maliyeti gün geçtikçe arttığından dolayı birçok sektörde veriler belli aralıklarla temizlenmekte bunun sonucunda işlendiğinde çok önemli sonuçlar doğurabilecek veri setleri yok olmaktadır (Chen ve Zhang, 2014). Depolama ve bakım maliyetlerini azaltmak amacıyla günümüzde bulut bilişim kullanılmaya başlanmış ve sorunun büyük bir kısmını çözmeyi başarmıştır.

Büyük verinin depolama ve bakım maliyeti çözüldükten sonra, daha büyük problemlerle karşı karşıya kalınmıştır. Özellikle gerçek zamanlı çalışan sistemlerde daha çok görülen bu problem, artan veri sonucu aranan verinin bulunma süresinin artmasıdır. Öyle ki bazı durumlarda sorgular saatler sürmektedir. Bu durum, gerçek zamanlı çalışan sistemlerde sistemin bazen cevap verememesine neden olmaktadır. Bu durum karşısında yeni veri tabanı çözümleri ortaya çıkmış veri tabanları güçlendirilmiştir. Günümüzde oldukça yaygınlaşmaya başlayan Nosql veri tabanlarının esas hedefi bunun gibi durumları ortadan kaldırmaktır.

3.4. Büyük Veri Araçları

Geçtiğimiz yıllarda ortaya çıkmış olan büyük veri kavramı, diskte çok fazla yer kaplayan veri manasında kullanılmamaktadır. Geleneksel yöntemler kullanılarak işlenip yönetilemeyen veri anlamında kullanılmaktadır. Sosyal medyanın

çıkması, işletmelerin bulut sistemine geçmesi, süreçlerin internet üzerinden yönetilmeye başlaması sonucunda internet ortamında devasa miktarda veri yığınları oluşmuştur. Şirketlerin rekabet ortamında öne çıkmak için fark yaratması gerekmektedir. Müşterilerine daha iyi hizmet vermek isteyen, süreçlerini iyileştirmek isteyen ve öne geçmek isteyen firma ya da kuruluşlar internet ortamında oluşan bu verileri kullanarak verilerindeki gizli kalmış örüntüleri keşfetmek istemektedirler.

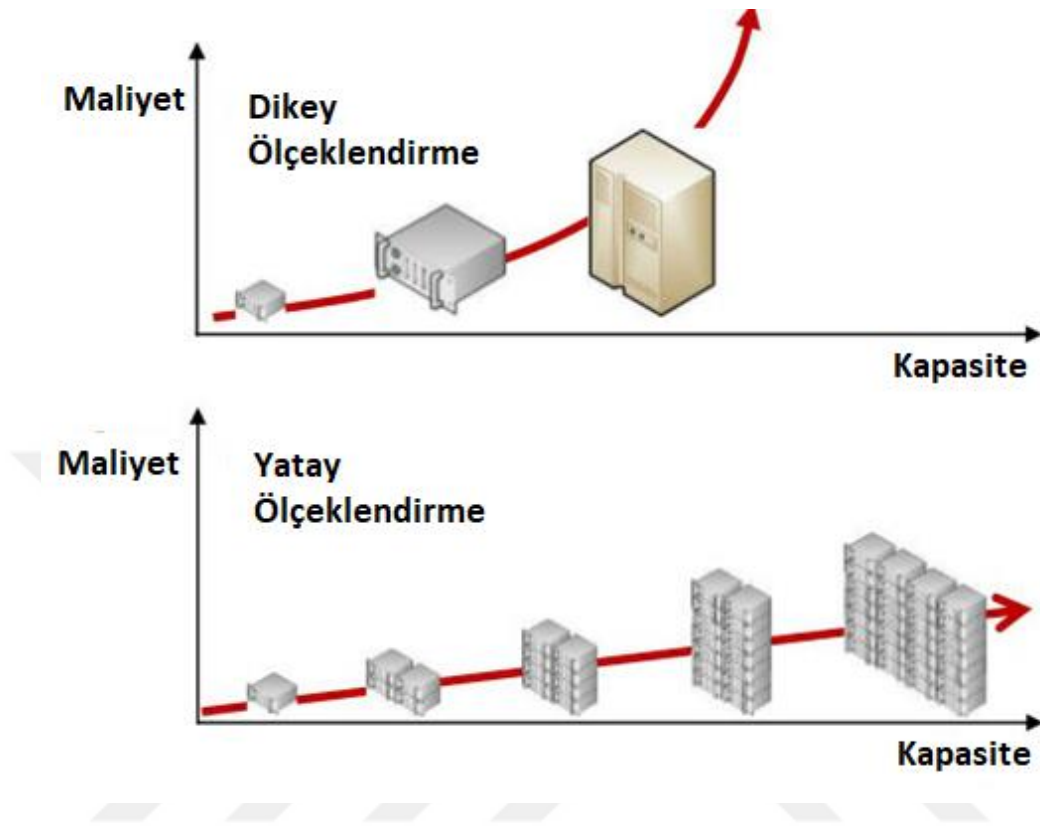
Teknolojinin gelişmesi ile birlikte donanım fiyatları düşmüş ve ev kullanıcıları terabyte seviyesindeki veriyi saklayabilecek donanıma kolayca ulaşabilir hale gelmiştir. Ancak, büyük veri petabyte hatta exabyte seviyesindeki boyutlara ulaşabilmektedir. 2000 senesinde dünya genelinde 800.000 petabyte boyutunda veri saklanmıştır. Bu rakamın 2020 yılında 35 zettabyte olacağı tahmin edilmektedir. Bazı firma ya da kurumlar saatte terabyte seviyesinde veri üretip bu verileri saklamaktadır. Bu veriyi yönetebilmek ve saklayabilmek için sadece donanımsal değil yazılımsal ihtiyaçlar da bulunmaktadır.

Büyük veri ile işlem yapmak istediğinde klasik yöntemler çok yetersiz kalmaktadır. Örnek vermek gerekirse, Google arama motoru üzerinde bir dakikada milyonlarca arama yapılmaktadır. Eğer ki Google klasik veri saklama yöntemlerini kullansa ya da veri arama noktasında geleneksel sql mantığı ile çalışmış olsaydı çok yavaş ve işlevsiz bir konumda olurdu. Bu tür durumlarda klasik ve geleneksel yöntemlerden kopmak ve yapısal olmayan yöntemler kullanmak gerekmektedir.

3.4.1. Dağıtık sistemler

Tek bir bilgisayarla sorun çözümlenmenin yeterli olmadığı, çözüm süresinin uzadığı, yeterli saklama alanına sahip olunmayan durumlarda dağıtık sistemlere ihtiyaç duyulmaktadır. Bilgisayar donanımlarını arttırarak işlem gücünü arttırmak mümkündür. Tek bir bilgisayarın gücünü arttırmak gün sonunda bazı sorunları beraberinde getirmektedir. Donanımın arızalanması, olası bir terslik

durumunda yerine çalışacak sistemin maliyetinin aynı miktarda olması, donanım gücü arttıkça maliyetin ciddi miktarda artması bu sorunların belli başlılarıdır.



Şekil 3.19. Dağıtık sistem ve donanım gücü artırma maliyet karşılaştırması

Şekil 3.19.'dan da anlaşılacağı gibi, bir sistemin gücü artırılarak dikeyde ölçeklendiğinde (scale up), maliyetin ilerleyen süreçte ciddi bir şekilde arttığı gözlemlenmektedir. Yatay ölçekleme (scale out) metodolojisi benimsendiğinde ise, maliyetin lineer bir şekilde arttığı görülmektedir. Bunun yanında, yatay ölçekleme beraberinde bazı avantajları da getirmektedir. Makinenin biri arızalandığında yerine hızlı bir şekilde başka bir makinenin eklenmesi, hali hazırda çalışan makinaların arızalanan makinenin yaptığı işi yapmaya devam etmesi bunlardan bazılarıdır (Erol, 2020).

Hafıza ve saati birbirinden bağımsız işlemciler kümesinin oluşturduğu, haberleşmelerini ağ üzerinden gerçekleştiren sistemlere dağıtık sistemler denmektedir (Galvin vd., 2003).

3.4.2. Apache Spark

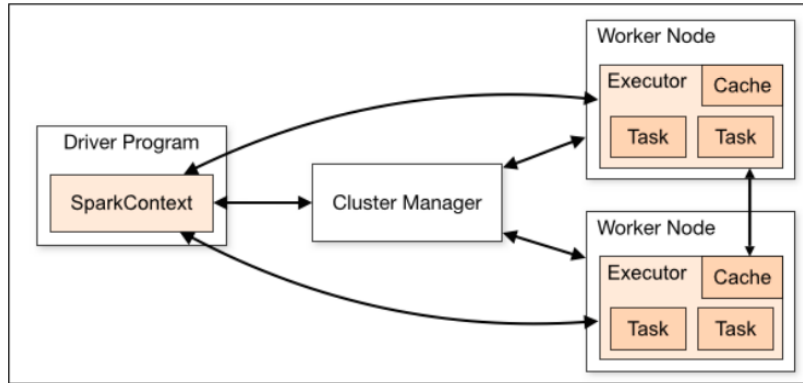
Büyük veri kümeleri kullanılan sistemlerde gelen sorguları daha hızlı cevaplamak ve üzerinde işlem yapmak için geliştirilen Spark, 2009 yılında Berkeley Üniversitesi AMPLab laboratuvarında Matei Zaharia tarafından geliştirilmeye başlanmıştır. Dağıtık sistemlerde veri analizi yapmaya, bellek içi veri işleme özelliğine imkân sağlayan ve Scala dili ile geliştirilmiş olan Apache Spark açık kaynak kodlu olarak geliştirilmeye devam etmektedir.

Mapreduce teknolojisini temel alan Hadoop, petabyte boyutundaki veriyi kolaylıkla ölçekleyebilmekte, hata toleranslı ve maliyeti düşük bir çözüm sağlamaktadır. Her ne kadar bu işlemleri esnek bir şekilde yapabilmeye imkân sağlasa da her bir işlemin bir önceki işleme bağlı olması bekleme sürelerini arttırmaktadır. Spark, bellek içi bilgi işleme özelliği sayesinde veriyi bellekte tutarak tekrar eden işlemler üzerinde çok hızlı işlem yapmaya imkân sağlamaktadır ve bu özelliği ile kendini ön plana çıkarmayı başarmıştır. Hadoop'a rakip olmak yerine eksiklerini tamamladığı görülmektedir ve yapılan çalışmalar Spark'ın Hadoop'a oranla 100 kata kadar daha hızlı olduğu görülmektedir (Akgün, 2016).

Apache Spark'ın kendine ait bir dağıtık dosya sistemi olmadığı için Hadoop tarafından desteklenen HBase, HDFS, Cassandra benzeri dosya sistemlerini kullanmaktadır. Verileri paralel olarak önbellekte saklayabilmek için kendine ait dosya kümeleme yöntemi olan RDD (Resilient Distributed Data) kullanmaktadır. Apache Spark çalışma anında dosya kaynak yöneticisine ihtiyaç duymaktadır. Kendi içerisindeki Spark Standalone kullanılabileceği gibi Hadoop YARN, Apache Mesos araçları da kaynak yöneticisi olarak kullanılabilmektedir.

Spark çalışma anında, bir sürücü (driver) ve buna bağlı birden fazla işçi (worker) çalıştırmaktadır. Her bir işçi çalışma anında gerekli verileri dosya sisteminden okur ve RDD formatında hafızasında saklar. Geliştirici tekrar kullanmak istediği verileri RDD formatında önbellekte saklayabilmektedir. RDD hata toleransı

bulundurmaktadır. Kaybolması ya da zarar görmesi durumunda yeniden oluşturulabilmektedir (Gu ve Li, 2013).



Şekil 3.20. Apache Spark çalışma prensibi (Spark, 2020a)



Şekil 3.21. Spark Ekosistemi (Küçük, 2019)

Şekil 3.20.'de Apache Spark çalışma prensibi gösterilmiştir. Şekil 3.21. incelendiğinde Apache Spark'ın en çok kullanılan özellikleri ve hangi yazılım dilleri ile geliştirilme yapılabileceği görülmektedir. Spark; Scala, Java, Python ve R programlama dilleri kullanılarak işlem yapmaya imkân sağlamaktadır.

Spark Core, isminden de anlaşıldığı gibi Apache Spark'ın işlevlerini yapabilmesi için gerekli olan temel motor görevini üstlenmektedir. İçerdiği kütüphaneler ile Sql ve makine öğrenmesi gibi işlemlerin yapılmasına imkân tanımaktadır. Bellekteki veri yönetimi, bir veri kümesindeki işleri planlama, izleme, depolama ve hata kurtarma gibi işlemlerden sorumlu mekanizmadır.

Spark Sql, Dataframe yapısıyla birlikte Apache Spark 1.3 versiyonunda ortaya çıkmıştır. RDD yapısından sonra ortaya çıkan Dataframe bellekteki yapısal olmayan veri üzerinde Sql benzeri bir dille sorgulamaya imkan sağlamaktadır (Akgün, 2016). Aynı zamanda Mssql, Mysql, Oracle benzeri veri tabanlarından okuma ve yazma işlemlerini yapmaya imkân sağlamaktadır. Bunun yanında Hive, Avro, Parquet, ORC, JSON gibi veri kaynaklarından veriyi okuyup sql deyimiyle "join" yani birleştirme işlemi yapabilmektedir (Spark, 2020b). Bunun sonucu olarak geliştiriciler, yapısal veri tabanındaki veriler ile yapısal olmayan verileri hızlı bir şekilde analiz edebilmektedirler.

Spark MLlib, "Machine Learning Library" kısaltılmasından oluşan "MLlib" isminden de anlaşılabilceği gibi makine öğrenmesi kütüphanelerini içermektedir. Apache Spark dağıtık sistemlerde bellek üzerinde işlem yapabildiği için çok iterasyonlu işler için kullanılmaktadır. MLlib içerisinde birçok makine öğrenmesi bulundurmaktadır.

3.4.3. Apache Hadoop

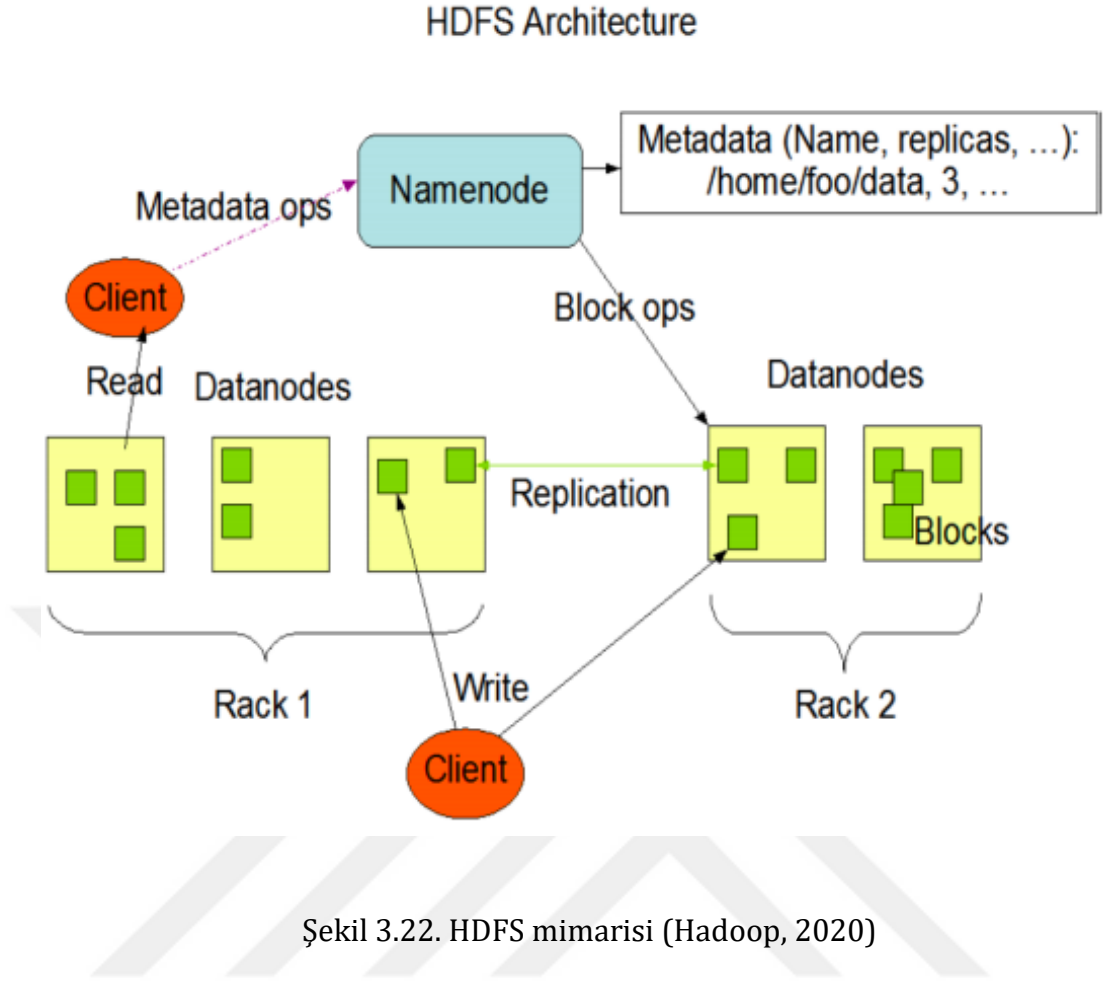
Mapreduce, Google tarafından büyük veri kümelerini işlemek amacıyla geliştirilmiş programlama modelidir (Dean ve Ghemawat, 2004). Mapreduce yaşam döngüsü üç fonksiyondan meydana gelmektedir. Bunlar Map fonksiyonu, Reduce Fonksiyonu ve Ana fonksiyondur. Map fonksiyonu master düğümdeki girişleri dinleyerek alt düğümlerdeki çalışan işçi makinalara işleri bölmektedir. İşçi makinalar işlemlerini bitirdikten sonra cevabı tekrardan master makinasına geri döndürürler. İşçi makinalardan dönen cevapları birleştiren master Reduce aşamasında sonuç üretmektedir (Yavuz vd., 2012).

Google Mapreduce modelini tanıttıktan sonra bu modeli kullanan birçok yazılım çatısı ve mimarisi ortaya çıkmıştır. 2003 yılında Doug Cutting, Mapreduce modelinden esinlenerek Apache Nutch için dağıtık dosya sistemi oluşturmuştur. Bu projenin işlevleri ve çıktıları değerlendirildiğinde farklı alanlarda da kullanılabilmesi fark edilmiş ve sonrasında Hadoop projesi açık kaynak kodlu olarak başlatılmıştır (Özdeş, 2017).

Sonuç olarak Hadoop, depolama yöntemi olarak Hadoop Dağıtık dosya sistemini, hesaplama yöntemi olarak Mapreduce modelini kullanan master ve slave olarak adlandırılmış düğümlere sahip bir mimaridir. Master düğümü, bir veya çok daha fazla slave düğümünü yöneten ve kontrol eden birimdir. Master ve slave düğümlerinden oluşan bu dağıtık sistem ölçeklenebilir, aynı zamanda binlerce bilgisayar eklenerek petabyte seviyesindeki veriyi işleme kabiliyetine sahiptir.

HDFS, sunucu kümelerinin bir araya gelerek oluşturduğu büyük veri üzerinde çalışırken kolaylık sağlayan çok büyük boyuttaki dosyaları saklamak için oluşturulmuş dağıtık dosya yönetim sistemidir (White, 2012). Google dosya modelini rol model almakta olan HDFS, java programlama dili ile geliştirilmiştir. HDFS sayesinde çok büyük verileri okuma süresinde geçen süre oldukça kısaldı. Verileri 64MB-1GB veri blokları halinde, replikaları birden fazla sunucuda bulunacak şekilde saklamaktadır. Bunun sonucunda pahalı RAID (Redundant Array of Independent Disks) sistemlerine gerek kalmamaktadır (Sammer, 2012).

HDFS, Namenode ve Datanode olmak üzere iki kısımdan meydana gelmektedir. Namenode, meta data bilgisini tutmaktadır. Datanode ise uygulama verisini tutmaktadır. Bu sunucular dağıtık sistem üzerinde kurulu oldukları için ağ üzerinden TCP/IP protokolü üzerinden haberleşmektedirler. Datanode sunucuları RAID maliyetinin yüksek olması nedeniyle verilerin güvenliğini sağlamak için verilerin replikasyonunu tutmaktadırlar (Doğan, 2014). Şekil 3.22' de HDFS mimarisi net bir şekilde görülmektedir.



Datanode'lar belli aralıklarla Namenode'lara kalp atımı olarak adlandırılan mesajlar göndermektedir. Bu sayede Namenode hangi replikasyonların ulaşılabilir durumda olduğunu anlamaktadır. İlk kurulumda 3 saniye olarak tanımlı olan bu mesaj aralığını isteğe göre değiştirmek mümkündür. Eğer 10 dakika boyunca Datanode bu mesajı göndermez ise servis dışı olarak tanımlanır ve Namenode yeni bir replikasyon planlar. Namenode tarafından doğrudan Datanode'a mesaj gönderilmemektedir. Bunun yerine kalp atımı olarak tanımlanan mesajların cevapları aracılığıyla iletişim kurmaktadır (Shvachko vd., 2010).

3.5. Birliktelik Kuralları

Veri Madenciliği modelleme tekniklerinden biri de birliktelik kurallarıdır. Tekrar eden verilerin tespit edilmesi ve ortaya çıkarılmasında önemli yer tutmaktadır.

Bunun en bilinen örneđi, bir alışveriş mağazasının en çok birlikte satılan ürünler üzerinden kampanya yaparak müşteriye daha çok ürün satmasıdır.

Satın alma eğilimlerini ortaya çıkaran birliktelik kuralları ve zaman örgülerinin, Pazar Sepet Analizi adı altında kar artışını sağlamak amacıyla kullanıldığı görülmektedir. Diğer kullanım alanları ise; promosyon analizleri, çapraz pazarlama, katalog ve yerleşim düzeni planlamadır (Erpolat, 2012).

3.5.1. Frequent Pattern Growth algoritması

Birliktelik kural analizi ile ilgili birçok algoritma bulunmaktadır. Birliktelik kural analizi için yaygın kullanılan algoritmalarından bir tanesi FP-Growth (Frequent Pattern Growth) Algoritması'dır (Jiawei vd., 2000). Birçok algoritmaya göre daha hızlı çalışmakta olan algoritma veritabanını sadece 2 defa tarayarak FP-Tree(Frequent Pattern Tree) olarak adlandırılan ağaç yapısını kurmaktadır. Veriler ağaç yapısında sıkıştırılmış olarak tutulmaktadır. Algoritma ilk taramasında destek değerini hesaplamaktadır. İkinci taramasında ise, ağaç veri yapısını kurmaktadır. Algoritmanın en önemli özelliđi diğer algoritmaların yapmakta olduđu gibi, aday küme üretmeden yaygın nesne kümelerini test edebiliyor olmasıdır. Bu nedenle büyük veri kümeleri ile çalışırken sistem kaynaklarını daha verimli kullanırken aynı zamanda da daha hızlı çalışmaktadır (Erpolat, 2012).

Destek değerleri algoritmaya verilen destek değerinden büyük ya da eşit olan veriler bir listeye eklenir ve büyükten küçüğe doğru sıralanır. Bu işlem sonrasında destek değerine göre veriler ağaca sıkıştırılarak yerleştirilir. Bu işlem sonrasında destek değeri büyük olan değerler ağacın köküne en yakın olacak şekilde yerlerini almış olurlar. Sıkıştırılma işlemi olarak adlandırılan işlem çok tekrarlı verilerin ilk ekler olarak birleştirilmesidir. Bu işlem sayesinde arama maliyeti önemli bir miktarda düşmektedir. Ağaç oluşturulurken, ağaçta olmayan her bir kayıt için düğüm oluşturulmaktadır. Her bir düğüm için destek değeri olarak başlangıçta 1 atanmaktadır. Eğer nesne önceden oluşturulmuş ise destek değeri 1 arttırılmaktadır.

Ađaç oluřturulduktan sonra Growth Algoritması ađaç üzerine uygulanır. İlk olarak ađaç üzerinde nesnelerin geçtiđi dalların tespiti yapılmaktadır. Birlikte sıkça grlen deđerleri belirleyen algoritma “bl ve ynet yaklařımına” uygun bir Őekilde byk nesne kmelerini kendi aralarında blerek daha kk parçalara blnmesini sađlamaktadır. Bundan dolayı oluřturulan FP-Tree veri yapısı asla asıl kmeden byk olamamaktadır (Pandey ve Pardasani, 2009).



4. ARAŞTIRMA BULGULARI VE TARTIŞMA

4.1. WoS Üzerinden Verilerin Elde Edilmesi

Bu çalışmada geliştirilen araçta, Türkiye adresli akademik çalışmaları incelemek amacı ile öncelikle WoS veri tabanından veriler çekilmiştir. Veriler çekilirken, WoS tarafından sağlanan servisler kullanılmıştır. WoS servislerini kullanabilmek için Ulaknet ağına dahil olmak gerekmektedir. Bu nedenle verilerin bir kısmı üniversite içerisindeki çalışmalarla, bir kısmı ise VPN aracılığıyla SDÜ ağına dahil olunarak yapılan çalışmalarla elde edilmiştir.

WoS servislerine API aracılığıyla ulaşmak için sadece Ulaknet ağına dahil olmak yetmemektedir. Servis, token tabanlı yetkilendirme kontrolü ile korunmaktadır. Ulaknet ağına bağlı iken yetkilendirme servisine istek yaparak gerekli token'ı almak ve bundan sonraki her isteğin 'Cookie' kısmında bulunan "SID" anahtar kelimesine bu değeri aktarmak gerekmektedir

Yetkilendirme işlemi sonrasında, gerekli verilerin alınacağı servis ile bağlantı yapılabilmektedir. Sonrasında aşağıda verilen parametreler yardımı ile sorgulama işlemleri yapılabilmektedir.

- "databaseId" : Hangi veritabanına bağlanılacağı bilgisi bu parametre ile sağlanmaktadır. Bu çalışma için "WOS" değeri kullanılmıştır.
- "userQuery": Bu parametre, kullanıcının detaylı arama yapabilmesine olanak sağlamaktadır. Bu çalışmada amaç Türkiye adresli çalışmalara erişmek olduğu için, "CU=Turkey" değeri kullanılmıştır.
- "begin" ve "end" parametreleri "timeSpan" tagının altındaki XML tagının altında bulunmaktadır. İsimlerinden de anlaşılacağı gibi, başlangıç ve bitiş tarihlerini belirtmektedir. Çalışmada bu alanlara, yazılan kod aracılığı ile üzerinde çalışılan senenin aylarının başlangıç ve bitiş tarihleri aktarılmıştır.

Sorgulama işlemi sonucunda servis, yapılan sorguya ait bir id ve toplam kaç tane kayıt bulunduğu bilgisini geri döndürülmektedir. Burada önemli olan nokta, bir id üzerinden maksimum 100000 kayıt okunmaya izin veriliyor olmasıdır. Elde edilmek istenen veri seti 585236 kayıt içerdiği için, işlemler aylık bazda sürdürülmüştür. 1980 yılından 2019 yılına kadar olan aylar için her seferinde servisten bir adet id alınmıştır. Bu sayede aynı sorgulama işlemi sonucunda 100000 üzerinde kayıt elde edilmesi imkânsız hale getirilmiştir. Sistemden alınan “queryId” parametresine ek olarak, 2 tane daha parametrenin verilerin alınacağı servise gönderilmesi gerekmektedir. Bu ek parametrelerden birincisi “count”, ikincisi ise “firstRecord” parametresidir. “count” parametresi, servise giden istek sonucunda tek seferde kaç adet kayıt getirileceğini temsil etmektedir. “firstRecord” parametresi ise o an kaçınıcı sayfada bulunduğu bilgisini saklamaktadır.

Bu işlemleri sağlayan bir kod yazılarak veriler çekilmiştir. WoS servisi SOAP alt yapısını kullandığı için veriler XML formatında dönmektedir. Veriler MongoDB veri tabanında saklanacağı için JSON formatına dönüştürülmüştür.

4.2. Verinin MongoDB Üzerinde Saklanması ve Filtrelenmesi

Çekilen ve JSON formatına dönüştürülen veriler, “pymongo” kütüphanesi aracılığıyla lokal makinede önceden kurulmuş olan MongoDB veri tabanına kaydedilmiştir. Veri tabanı olarak MongoDB’nin tercih edilme sebepleri, verilerin ayrıştırılmadan direkt olarak kaydedilebilmesi ve verinin yapısal olmamasıdır.

Verilerin eksik olması veriler üzerinde yapılacak analiz çalışmalarını etkilemektedir. Bu nedenle MongoDB veri tabanına aktarılan veriler üzerinde bir filtreleme yapılmıştır.

MongoDB depolama ve sorgu hızı konusunda başarılı olduğu kadar dinamik sorgular üretmek konusunda da gayet başarılı bir veri tabanı aracıdır. Oluşturduğu sorgular, sakladığı veri tipi gibi JSON formatındadır. Anahtar kelime ve değer şeklinde oluşan sorgularda MongoDB’nin filtre için kullanılan özel

anahtar kelimelerini bilmek gerekmektedir. Bu çalışmada kullanılan anahtar kelimeler ve bu anahtar kelimelerin kullanımları hakkında bilgi aşağıda verilmiştir.

- \$and: Liste tipinde ifade almakta ve bir ya da daha fazla ifade sorgulayabilmektedir. İçerdiği tüm ifadeler için 'true' değerini alan kayıtlar listelenmektedir. Değerlendirme sırasında, ifadelerden bir tanesi 'false' değeri aldığı anda diğer ifadeler için kontrol durmaktadır. 'null', '0' ve 'undefined' değerler 'false', diğer bütün değerler 'true' olarak değerlendirilmektedir. 0 olmayan sayısal değerler ve listeler de buna dahildir. Sql dilindeki 'and' ifadesi ile aynı işlevi görmektedir.
- \$or Liste tipinde ifade almakta ve bir ya da daha fazla ifade sorgulayabilmektedir. İçerdiği ifadelerden herhangi biri için 'true' değerini alan kayıtlar listelenmektedir. Değerlendirme sırasında ifadelerden bir tanesi 'true' değeri döndüğü anda diğer ifadeler için kontrol durmaktadır.
- \$ne: 'not equal' kelimesinin kısaltılmış halidir. Türkçe çevirisi eşit olmayan anlamına gelir. Anlaşıldığı gibi verilen değere eşit olmayan değerleri sorgulamak için kullanılmaktadır.
- \$exists: 'exist' Türkçe çevirisine bakıldığında içermek anlamına gelmektedir. Boolean tipinde değer almaktadır. 'true' değerini aldığı anda aranan alanı içeren veriler listelenmektedir. 'false' değerini aldığı anda ise o alanı içermeyen verileri listelemektedir.
- \$elemMatch: Bu operatör bir liste içerisindeki elemanlar için arama yapılırken kullanılmaktadır.

```

{
  "$and": [
    {
      "static_data.fullrecord_metadata.addresses.@count": {
        "$ne": "0"
      },
      "static_data.summary.doctypes.@count": "1",
      "static_data.fullrecord_metadata.category_info": {
        "$exists": true
      },
      "static_data.summary.doctypes.doctype": "Article",
      "static_data.summary.names.name": {
        "$exists": true,
        "$ne": [],
        "$elemMatch": {
          "@daisng_id": {
            "$exists": true
          }
        }
      },
      "static_data.fullrecord_metadata.reprint_addresses.@count": {
        "$exists": true,
        "$ne": "0"
      }
    },
    {
      "$or": [
        {
          "static_data.fullrecord_metadata.keywords": {
            "$exists": true
          }
        },
        {
          "static_data.item.keywords_plus": {
            "$exists": true
          }
        }
      ]
    }
  ]
}

```

Şekil 3.23. Çalışmada kullanılan JSON filtre

Şekil 3.23.'deki sorguda istenen şartları aşağıdaki sırlamak mümkündür.

- Kökte bulunan '\$and' operatörü sayesinde tüm şartların sağlanması zorunlu kılınmaktadır.
- "static_data.fullrecord_metadata.addresses.@count" anahtar kelimesine sahip verilerden değeri 0 olmayanlar istenmektedir. Veri setinde 0 değerinin altında bir değer bulunmamaktadır. Yani, her kayıt için adres olmak zorundadır.
- "static_data.summary.doctypes.@count" anahtar kelimesine sahip verilerden değeri 1 olanlar istenmektedir.
- "static_data.summary.doctypes.doctype" anahtar kelimesine sahip verilerden değeri 'Article' olanlar istenmektedir. Bu şartı üçüncü maddedeki sorguyla beraber yorumlamak gerekmektedir. Üçüncü maddede doküman tipi 1 den fazla ya da az olamaz şartı konulmuştu. Bu maddeyle birlikte sadece doküman tipi 'Article' olanlar elde edilmiştir.
- "static_data.summary.names.name" anahtar kelimesine sahip ve boş olmayan liste tipindeki kayıtlar içerisinde '@daisng_id' alanı bulunanlar istenmektedir. '@daisng_id' alanı her bir yazar için birincil anahtardır. Yapılan incelemelerde, yazarların isim ve soy isimlerinin yanlış girildiği görülmüştür. Bu alan aracılığı ile olası hatalı sonuçların çıkması engellenmiştir.
- "static_data.fullrecord_metadata.reprint_addresses.@count" anahtar kelimesine sahip ve değeri 0 olmayan kayıtlar istenmiştir.
- Son olarak "static_data.fullrecord_metadata.keywords" ya da "static_data.item.keywords_plus" anahtar kelimelerinden en az birine sahip olması istenmiştir.

Yukardaki sorgu işlemi sonucunda 585236 kayıt içinden 349303 kayıt seçilmiştir. Bulunan kayıtlar Python kodu aracılığıyla yeni bir collection'a aktarılmıştır. Bu aktarım sırasında bir filtreleme daha yapılmıştır. Bu filtreme işlemi sonucunda her yazarın adresinin bulunması zorunlu tutulmuştur ve son kayıt sayısı 220969 olarak bulunmuştur.

4.3. Verinin Odoo Programlama Çatısına Aktarımı

Verileri hem görselleştirmek hem de ilişkili kayıtları bir web kullanıcı arabirimine aktarmak amacı ile Odoo kullanılmıştır. Odoo model first mimarisi ile çalıştığı için veriler Odoo yapısına aktarıldığında, MongoDB üzerinde bulunan kayıtlar hem ilişkisel veri tabanı olan Postgresql'e taşınmış olacak hem de web uygulaması aracılığıyla görselleştirilmiş olacaktır.

Bunu yapabilmek için öncelikle "master" isminde bir odoo modülü geliştirilmiştir. Bu modül, otomatik olarak veri tabanı tablolarını oluşturmada ve ilişkilerini gerçekleştirmektedir. Aynı zamanda web uygulaması aracılığıyla veriyi sorgulama, ekleme, silme ve düzenleme yetkinliği sağlamaktadır. Bunların yanında, yetkilendirme işlemlerini kendi içerisinde bulunan yapı sayesinde yönetme imkânı sağlamaktadır.

Modül içerisinde "controller" aracılığıyla bir web servis yazılmıştır. Bu web servis JSON formatında MongoDB içerisindeki veri listesini almaktadır. Bu liste ayrıştırma işlemleri sonrasında kaydedilmektedir.

Kayıt süreci sonrasında veriler yapısal veri tabanına aktarılmış olmaktadır. Bundan sonraki süreçte, veriler Sql aracılığıyla sorgulanıp istenen bilgilere ulaşılabilecektir. Aynı zamanda elde edilen bu veriler üzerinde Spark ile de çalışılacaktır

4.4. Verinin Spark ile İşlenmesi

Veriler yapısal veri tabanına aktarıldıktan sonra okul, yazar, akademik çalışma ve anahtar kelime tabloları, Sql sorgusu aracılığıyla csv tipindeki dosyalara dönüştürülmüştür. Sisteme Spark için gerekli yazılımlar yüklenmiştir.

Çalışmada ilişki analizini gerçekleştirmek amacı ile FP-Growth algoritması seçilmiştir. Apache Spark aracılığıyla üniversiteler ve yazarlar tabloları üzerinde FP-Growth algoritması kullanılarak birliktelik analizi yapılmıştır.

4.4.1. Üniversitelerin birliktelik analizi

İlk olarak akademik çalışmalarda beraber çalışan üniversiteler incelenmiştir. Bu inceleme sonucunda en çok beraber çalışma yapan üniversiteler Çizelge 4.1.'de sunulmuştur.

Çizelge 4.1. En çok beraber çalışma yapan üniversiteler

Berber Çalışma Yapan Üniversiteler	Berber Çalışma Sayısı
Gazi Univ, Hacettepe Univ	783
Gazi Univ, Ankara Univ	763
Ankara Univ, Hacettepe Univ	676
Dokuz Eylul Univ, Ege Univ	628
Celal Bayar Univ, Ege Univ	487
Marmara Univ, İstanbul Univ	449
Necmettin Erbakan Univ, Selcuk Univ	419
Adnan Menderes Univ, Ege Univ	349
Kirikkala Univ, Gazi Univ	325
Kirikkala Univ, Ankara Univ	305
Anadolu Univ, Eskisehir Osmangazi Univ	302
Recep Tayyip Erdoğan Univ, Karadeniz Tech Univ.	301
Basken Univ, Hacettepe Univ	298
Erzincan Univ, Ataturk Univ	282
Bezmi Alem Vakıf Univ, İstanbul Univ	270
Hacettepe Univ, İstanbul Univ	250
Mersin Univ, Çukurova Univ	243
Inonu Univ, Fırat Univ	242
Erciyes Univ, Gazi Univ	241

Çizelge 4.1. En çok beraber çalışma yapan üniversiteler (Devam)

Ankara Numune Training & Res Hosp, Hacettepe Univ	238
Selçuk Univ, Gazi Univ	237
Ege Univ, Hacettepe Univ	236
Baskent Univ, Gazi Univ	234
Ege Univ, Istanbul Univ	232
Pamukkale Univ, Ege Univ	229
Baskent Univ, Ankara Univ	228

Beraber en çok çalışma yapan üniversiteler incelendiğinde; birbirine yakın üniversitelerin daha çok birlikte çalışma yaptığı gözlemlenmiştir. Özellikle, Ankara ve İstanbul şehirlerindeki üniversitelerin birlikte daha fazla çalışma yaptığı görülmüştür. Gazi, Ankara, Hacettepe ve İstanbul Üniversiteleri'nin, sadece buldukları şehirlerdeki üniversitelerle beraber çalışmadıkları, aynı zamanda farklı şehirlerdeki üniversitelerle de çalışma yaptıkları görülmektedir.

Birliktelik analizi sonucunda, beraber çalışma ihtimali olan ilk 20 üniversite Çizelge 4.2.'de sunulmuştur.

Çizelge 4.2. Birlikte çalışma yapabilecek üniversite tahmin sonuçları

Baskent Univ	Ankara Univ, Hacettepe Univ Gazi Univ
Istanbul Univ	Hacettepe Univ Marmara Univ Ege Univ Bezmialem Vakif Univ

Çizelge 4.2. Birlikte çalışma yapabilecek üniversite tahmin sonuçları (Devam)

Hacettepe Univ Zonguldak Karaelmas Univ	Ankara Univ Ege Univ Gazi Univ Istanbul Univ Baskent Univ Ankara Numune Training & Res Hosp
Selcuk Univ	Gazi Univ Necmettin Erbakan Univ
Ataturk Univ Pamukkale Univ	Ege Univ Erzincan Univ
Gazi Univ	Ankara Univ Hacettepe Univ Erciyes Univ Baskent Univ Selcuk Univ Kirikkale Univ
Hacettepe Univ Ege Univ Rize Univ	Ankara Univ Dokuz Eylul Univ Celal Bayar Univ Gazi Univ Istanbul Univ Baskent Univ Adnan Menderes Univ Pamukkale Univ Ankara Numune Training & Res Hosp
Ankara Univ	Hacettepe Univ Gazi Univ Baskent Univ Kirikkale Univ

Çizelge 4.2. Birlikte çalışma yapabilecek üniversite tahmin sonuçları (Devam)

Ankara Univ Kirikkale Univ	Hacettepe Univ Gazi Univ Baskent Univ
Baskent Univ Minist Hlth Bayindir Hosp	Ankara Univ Hacettepe Univ Gazi Univ
Istanbul Univ Haseki Training & Res Hosp	Hacettepe Univ Marmara Univ Ege Univ Bezmialem Vakif Univ
Baskent Univ Ortoklin	Ankara Univ Hacettepe Univ Gazi Univ
Ankara Univ Dr Sami Ulus Childrens Hosp	Hacettepe Univ Gazi Univ Baskent Univ Kirikkale Univ
Marmara Univ Istanbul Univ Haydarpara Numune Teaching & Res Hosp	Hacettepe Univ Ege Univ Bezmialem Vakif Univ
Kirikkale Univ	Ankara Univ Gazi Univ
Celal Bayar Univ Ege Univ	Hacettepe Univ Dokuz Eylul Univ Istanbul Univ Adnan Menderes Univ Pamukkale Univ

Çizelge 4.2. Birlikte çalışma yapabilecek üniversite tahmin sonuçları (Devam)

Istanbul Univ Kadir Has Univ Univ Ghent	Hacettepe Univ Marmara Univ Ege Univ Bezmialem Vakif Univ
Anadolu Univ Pamukkale Univ Texas A&M Univ SUNY Coll Brockport	Ege Univ Eskisehir Osmangazi Univ
Ankara Univ Kirikkale Univ Ufuk Univ	Hacettepe Univ Gazi Univ Baskent Univ
Istanbul Univ Afyon Kocatepe Univ	Hacettepe Univ Marmara Univ Ege Univ Bezmialem Vakif Univ

Çizelge 4.2. ile, çizelgenin birinci sütunundaki üniversite veya üniversiteler ile ikinci sütunundaki üniversite veya üniversitelerin çalışma yapma olasılıklarının olduğu bilgisi verilmiştir. Çizelge 4.1'e bakıldığında en çok beraber çalışma yapan üniversitelerin aynı ya da yakın şehirlerde olduğu görülmektedir. Birliktelik analizi sonuçları incelendiğinde, sonuçların tutarlı olduğu görülmektedir. Üniversite birliktelik analizi sonuçlarının aynı ya da yakın şehirlerde olan üniversiteleri içerdiği görülmektedir. Aynı zamanda, daha önce fazla sayıda birlikte çalışma yapan üniversitelerin de birlikte yayın yapabilme ihtimali olan üniversiteler arasında olduğu görülmektedir.

4.4.2. Yazarların birliktelik analizi

Yazarların analizi yapılmadan önce verilerin tutarlılığı gözden geçirilmiştir. Bu inceleme sırasında WoS verilerinin bazılarının tutarsız olduğu görülmüştür. WoS'un, yazarları akademik çalışmalar ile birbirine bağlama sırasında hatalı

işlemler yaptığı fark edilmiştir. İsim benzerliği olan yazarların yayınlarının yanlış bağlandığı ortaya çıkmıştır. Genel bir inceleme sonrası birçok sonuç doğru olsa da bu yanlış eşleştirmenin birliktelik analizinin sonuçlarını etkileyeceği ön görülmüştür.

Çalışmanın bu kısmında en çok birlikte yayın yapan yazarlardan sadece ilk 20 tanesi ne yer verilmiştir.

Çizelge 4.3 En çok beraber çalışma yapan yazarlar

Veritabanı Id	Yazar Adı Soyadı	Birlikte Yapılan Makale Sayısı
2362	Bekir Cakir	95
15183	Reyhan Ersoy	
908	Mustafa Soylak	78
10000	Mustafa Tuzen	
8329	Y. Atalay	77
8331	D. Avci	
13177	Mehmet Musa Oezcan	77
86648	Fahad Y. Al Juhaimi	
4223	Kudret Aytemir	77
4224	Ali Oto	
15367	Tahir Durmaz	77
19406	Engin Bozkurt	
15364	Telat Keles	77
15367	Tahir Durmaz	
1059	A. K. Baltaci	75
1060	R. Mogulkoc	
4223	Kudret Aytemir	75
8652	E. Kaya	
15364	Telat Keles	73
19406	Engin Bozkurt	

Çizelge 4.3 En çok beraber çalışma yapan yazarlar (Devam)

7770	A. Aktumsek	72
57782	Gokhan Zengin	
9151	S. Habib Mazharimousavi	71
9152	M. Halilsoy	
4223	Kudret Aytemir	71
36390	Ugur Canpolat	
690	Mehmet Haberal	71
693	Gokhan Moray	
516	Kazim Sahin	71
3713	Mehmet Tuzcu	
15364	Telat Keles	71
15367	Tahir Durmaz	
19406	Engin Bozkurt	
14817	Dilek Berker	70
14820	Serdar Guler	
22781	Ugur Dilmen	69
37912	S. Oguz	
4223	Kudret Aytemir	69
15122	H. Yorgun	
17370	Abdulahdi Baykal	68
45998	Huseyin Sozeri	

Birlikte çalışma yapan yazarlar incelendiğinde 157 adet kayıt ortaya çıkmıştır. Bunlardan ilk 20 si Çizelge 4.3.'te listelenmiştir. İlk 20 kayıt arasında sadece 1 kayıta 3 tane yazar bulunmaktadır. Diğer birliktelik sonuçları 2 şerli yazardan oluşmaktadır. Geriye kalan sonuçlar incelendiğinde en fazla 4 er kişilik yazar grupları olduğu görülmüştür.

Çizelge 4.4. Yazarların birliktelik kural sonuçları

Antecedent	Consequent	Confidence	Lift
Aydin Unal Bulent Tokgoz Oktay Oymak	Murat Hayri Sipahioglu	1.0	3112,23943661971
Ali Oto Lale Tokgozoglu	Kudret Aytemir	1.0	1523,92413793103
Ali Oto Ugur Canpolat	Kudret Aytemir	1.0	1523,92413793103
Saliha Ilican Mujdat Caglar	Yasemin Caglar	1.0	3682,816666666666
Fahad Y. Al Juhaimi	Mehmet Musa Oezcan	1.0	1004,4045454545454
R. Mogulkoc	A. K. Baltaci	1.0	2797,07594936708
Saliha Ilican Yasemin Caglar	Mujdat Caglar	1.0	3876,64912280701
Erhan Guneyisi	Mehmet Gesoglu	1.0	3809,81034482758
Telat Keles Nihal Akar Bayram Engin Bozkurt	Tahir Durmaz	1.0	2569,40697674418
Saliha Ilican	Mujdat Caglar	1.0	3876,64912280701 76
Saliha Ilican	Yasemin Caglar	1.0	3682,816666666666
Nihal Akar Bayram Tahir Durmaz Engin Bozkurt	Telat Keles	1.0	2694,.7439024390 2
Telat Keles Nihal Akar Bayram	Tahir Durmaz	1.0	2569,40697674418

Çizelge 4.4. Yazarların birliktelik kural sonuçları (Devam)

Nihal Akar Bayram Tahir Durmaz	Telat Keles	1.0	2694,74390243902
Murat Hayri Sipahioglu Bulent Tokgoz	Oktay Oymak	1.0	3069,01388888888
E. Kaya Ugur Canpolat	Kudret Aytemir	1.0	1523,92413793103
Murat Hayri Sipahioglu Aydin Unal Bulent Tokgoz	Oktay Oymak	1.0	3069,01388888888
Ali Oto E. Kaya	Kudret Aytemir	1.0	1523,92413793103
Yasemin Caglar Mujdat Caglar	Saliha Ilican	1.0	4803,67391304347
Omer Erdeve S. Oguz	Ugur Dilmen	1.0	2008,80909090909

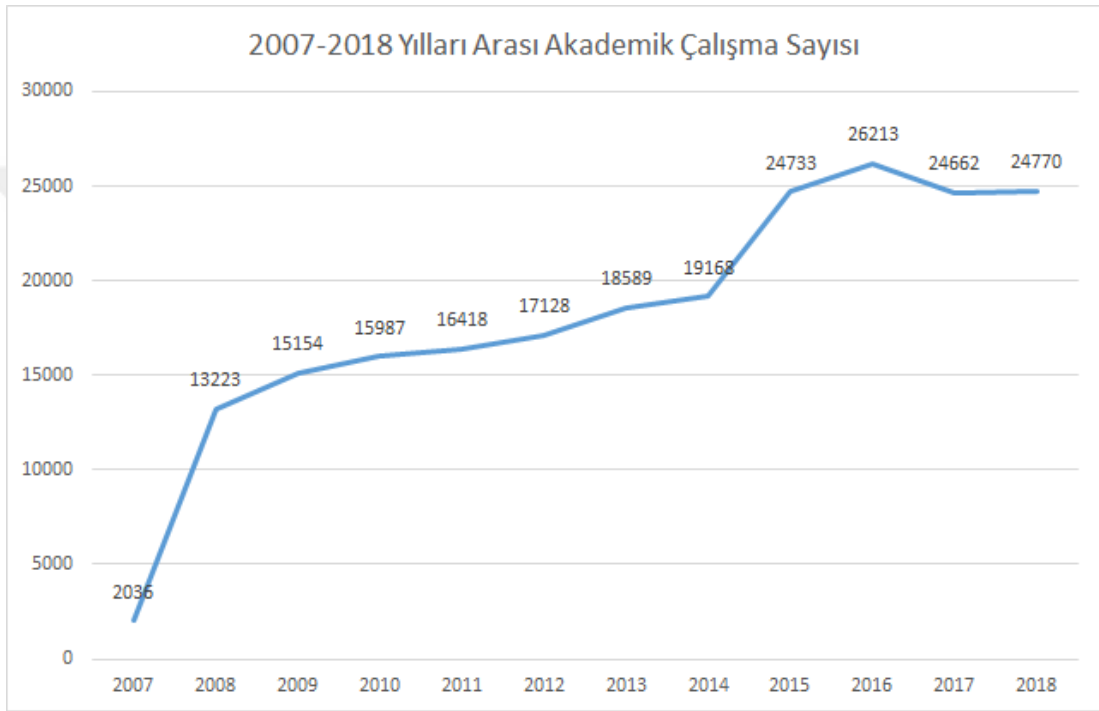
Analiz sonucunda toplamda 356 adet birliktelik kuralı çıkmıştır. Bu kurallardan sadece “confidence” (güven) değeri 1’e eşit olanlar Çizelge 4.4.’te listelenmiştir.

4.5. Veri Tabanından Elde Edilen Sonuçlar

Yapısal veri tabanı olan Postgresql üzerine aktarılmış veriler sql sorguları aracılığıyla analiz edilmiştir. Sene bazında üniversiteler incelenirken, tüm üniversitelerin eşit olarak değerlendirilebilmesi adına son 5 yıla ait veriler dikkate alınmıştır.

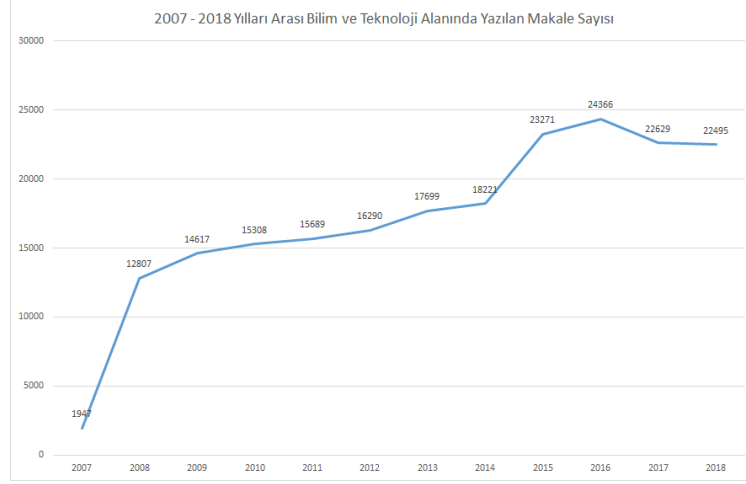
Şekil 4.1.’de 2007-2018 yılları arasında Türkiye adresli olarak yapılan yayın sayılarına ait grafik sunulmuştur. 1989-2007 yılları arasındaki akademik çalışma sayısı çok az olduğu için grafikte yer verilmemiştir. Grafik incelendiğinde 2008

yılında makale sayısındaki yaklaşık %549'luk artış dikkat çekmektedir. 2014 yılına kadar artışın düzenli bir şekilde arttığı görülmektedir. 2015 yılına gelindiğinde yaklaşık %29'luk bir artış dikkat çekmektedir. 2016 yılına kadar sürekli artmakta olan makale sayısı 2017 yılında ilk defa %5'lik oranda düşüş göstermiştir. 2007-2018 yılları arasında toplamda 218081 tane makale yayınlandığı görülmüştür.



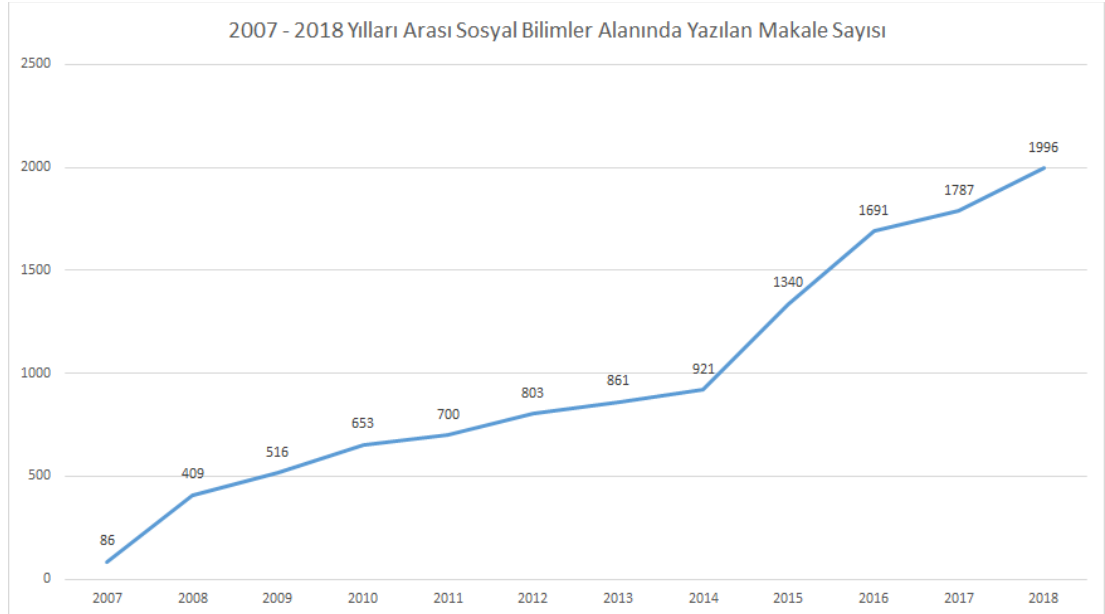
Şekil 4.1. 2007-2018 Yılları arası türkiye adresli akademik çalışma sayısı

Şekil 4.2'de, 2007-2018 yılları arasında bilim ve teknoloji alanında yayımlanan makalelere ait grafik sunulmuştur. Bu alanda yazılan makalelerin toplamı tüm kategorilerde yazılan makalelerin yaklaşık %94'üne karşılık gelmektedir. Yani Türkiye'de yazılan akademik makalelerin yaklaşık %94'ü bu alandadır.



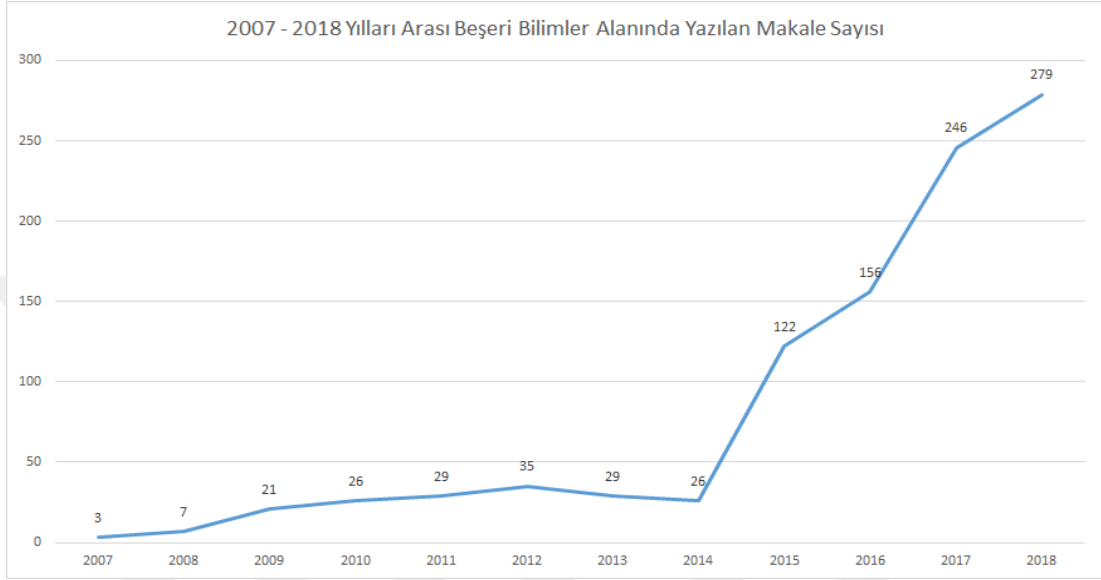
Şekil 4.2. 2007-2018 Yılları arası Türkiye adresli bilim ve teknoloji alanında yazılan makale sayısı

Şekil 4.3'de, 2007-2018 yılları arasında sosyal bilimler alanında yayımlanan makalelere ait grafik sunulmuştur. Grafikte de görüldüğü gibi bu alanda çok az makale yazılmıştır. Makale sayısı her ne kadar az olsa da her sene artan bir ivme vardır. 2007-2018 yılları arasında toplamda 11763 adet makale üretilmiştir. Bu da toplam makale sayısının yaklaşık %5,39'una karşılık gelmektedir.

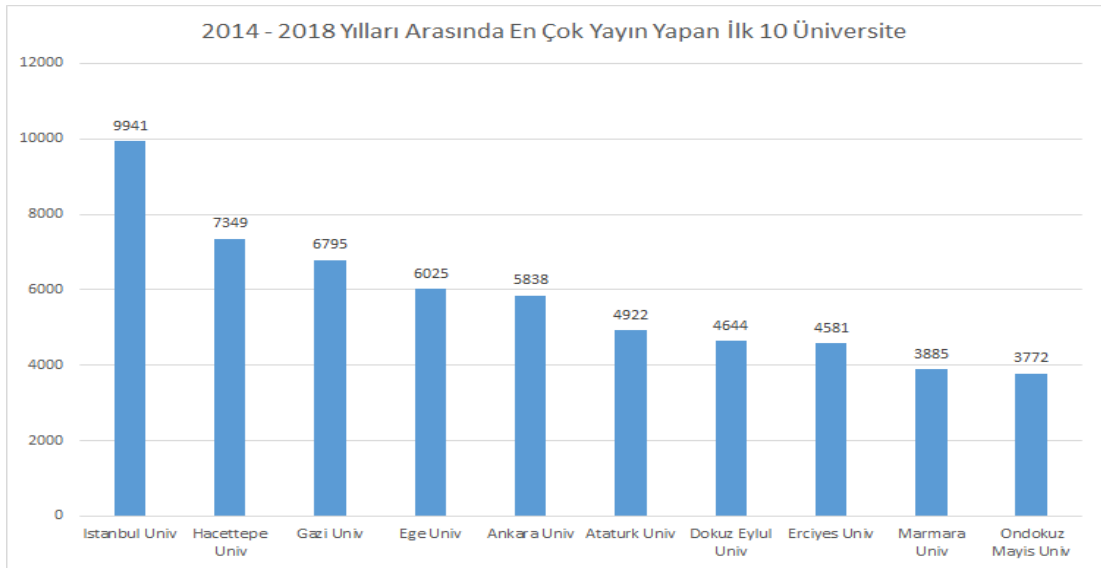


Şekil 4.3. 2007-2018 Yılları arası Türkiye adresli sosyal bilimler alanında yazılan makale sayısı

Şekil 4.4'de, 2007-2018 yılları arasında beşerî bilimler alanında yayımlanan makalelere ait grafik sunulmuştur. Grafik incelendiğinde, Türkiye'de beşerî bilimler alanında çok az makale yazıldığı görülmektedir. 2007-2018 yılları arasında toplamda 979 adet makale yazılmıştır. Bu da toplam makale sayısının yaklaşık %0,45'ine karşılık gelmektedir.

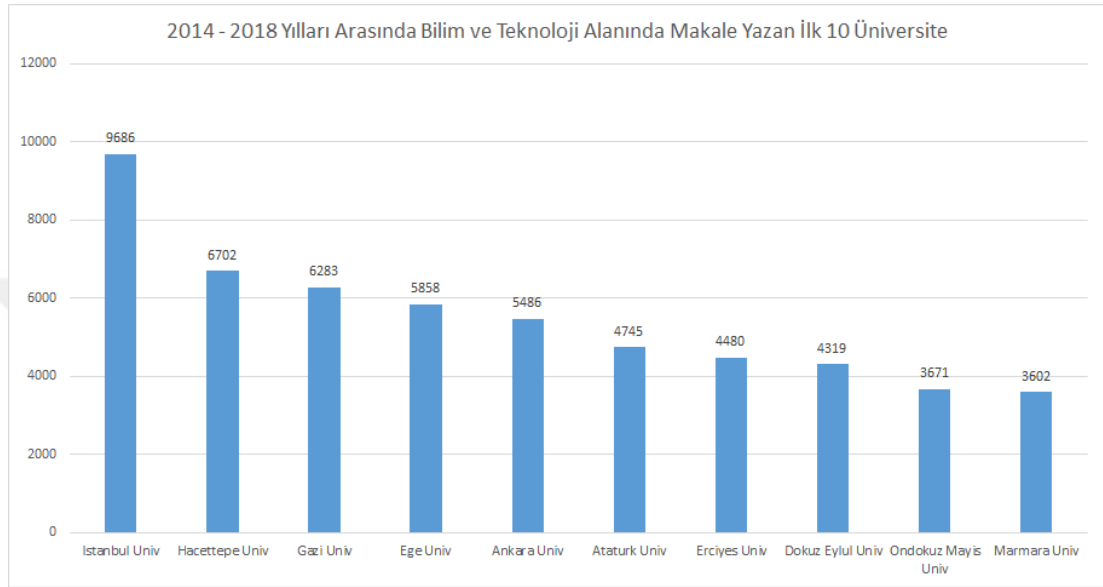


Şekil 4.4. 2007-2018 Yılları arası Türkiye adresli beşerî bilimler alanında yazılan makale sayısı



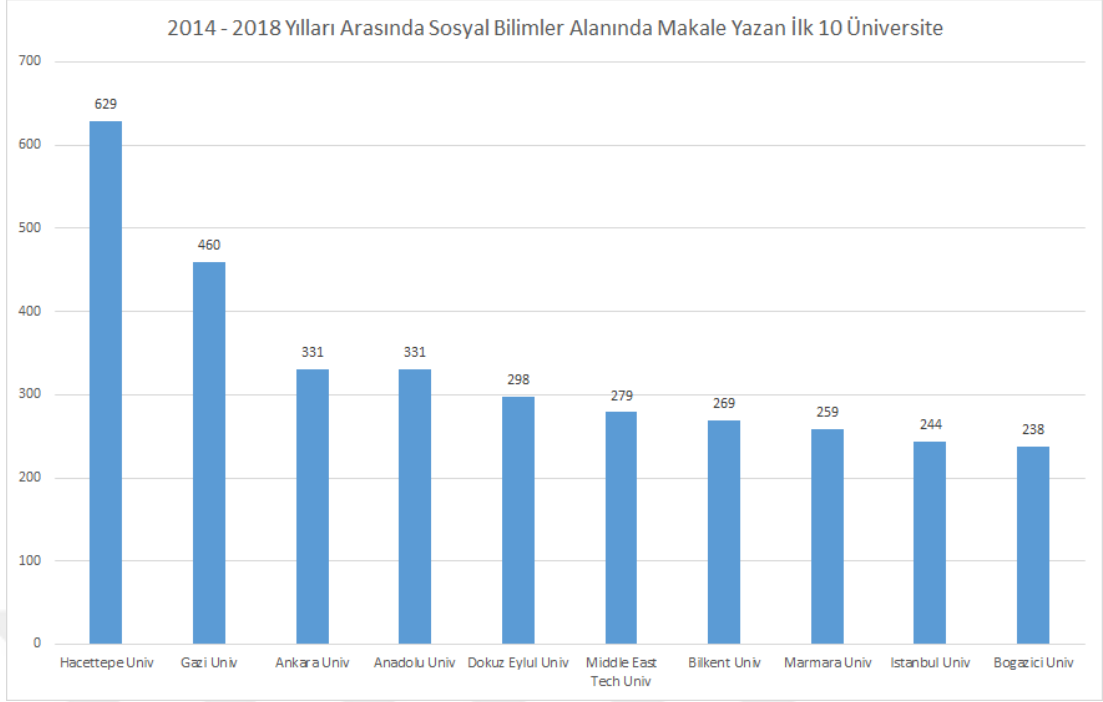
Şekil 4.5. 2014-2018 Yılları arası en çok yayın yapan 10 üniversite

Şekil 4.5.'de, 2014-2018 yılları arasında Türkiye adresli yayımlanan akademik makale sayıları incelenmiştir. En çok makale yayınlayan üniversitenin İstanbul Üniversitesi olduğu görülmüştür. İstanbul Üniversitesine en yakın ikinci üniversitenin ise 7349 makale ile Hacettepe Üniversitesi olduğu görülmüştür. Süleyman Demirel Üniversitesi 3260 makale ile 17. sırada yer almaktadır.



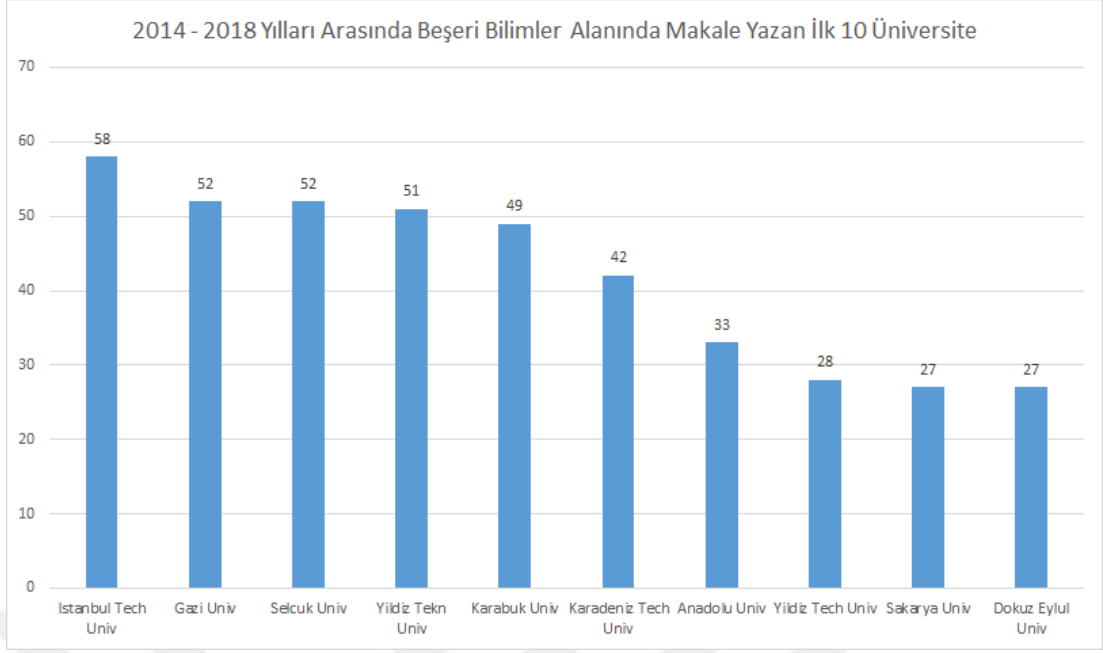
Şekil 4.6. 2014-2018 Yılları arası bilim ve teknoloji alanında en çok yayın yapan 10 üniversite

Şekil 4.6.'da 2014-2018 yılları arasında üniversite bazında, bilim ve teknoloji alanındaki Türkiye adresli makale sayıları analiz edilmiştir. En çok yayın yapan üniversitenin İstanbul Üniversitesi olduğu görülmüştür. Süleyman Demirel Üniversitesi 3168 makale ile 17. sırada yer almaktadır.



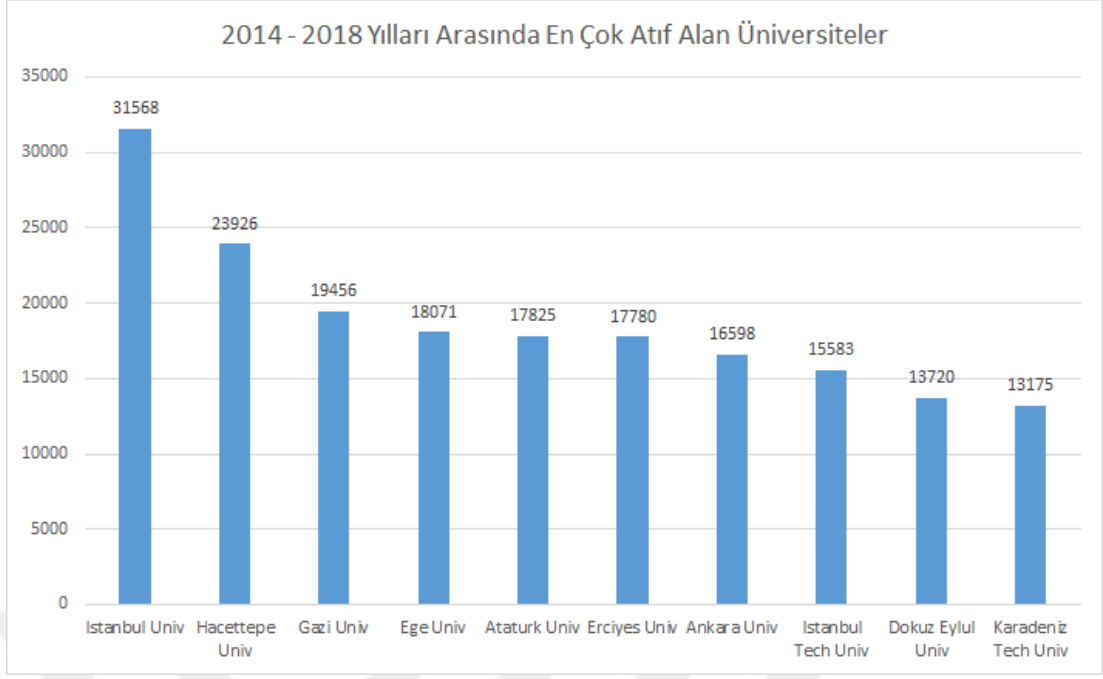
Şekil 4.7. 2014-2018 Yılları arası sosyal bilimler alanında en çok yayın yapan 10 üniversite

Şekil 4.7.'de, 2014-2018 yılları arasında üniversite bazında, sosyal bilimler alanındaki Türkiye adresli makale sayıları analiz edilmiştir. 629 yayınlı en çok yayın yapan üniversite, bilim ve teknoloji alanında 2. sırada olan Hacettepe Üniversitesidir. Bilim ve teknoloji alanında 1. sırada olan İstanbul Üniversitesi sosyal bilimler alanında 9. Sırada olduğu görülmüştür. Süleyman Demirel Üniversitesi 87 yayın ile sosyal bilimler arasında 40. sıradadır.



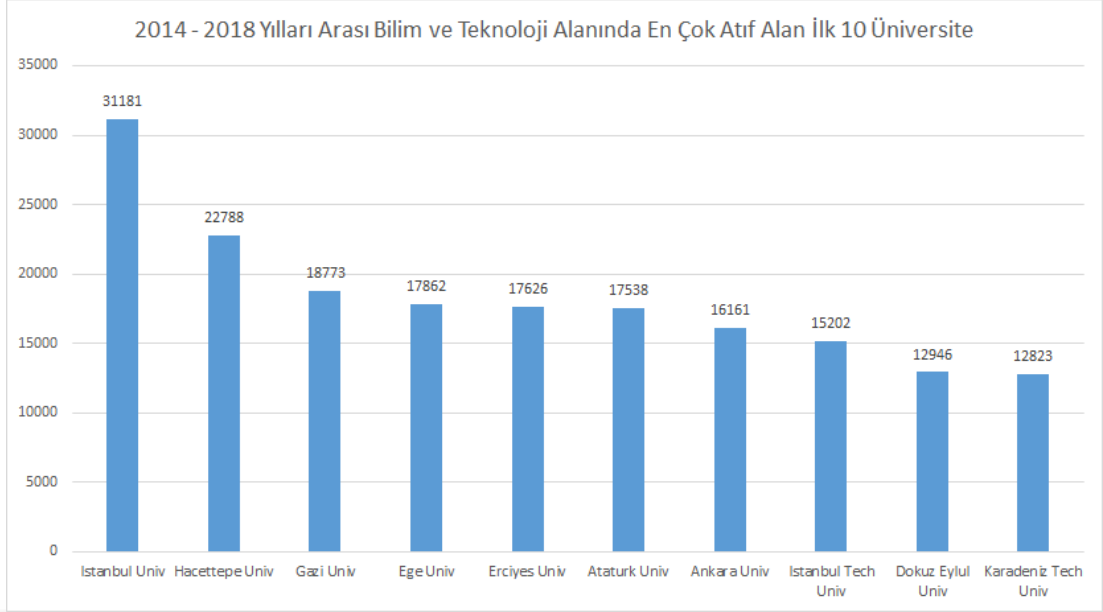
Şekil 4.8. 2014-2018 Yılları arası beşerî bilimler alanında en çok yayın yapan 10 üniversite

Şekil 4.8.'de, 2014-2018 yılları arasında üniversite bazında, beşerî bilimler alanındaki Türkiye adresli makale sayıları analiz edilmiştir. 58 yayınlı İstanbul Teknik Üniversitesi'nin birinci sırada olduğu görülmektedir. Diğer alanlarda ilk sıralarda olan İstanbul Üniversitesi ve Hacettepe Üniversitesi bu alanda ilk 10 da yer almamaktadır. Süleyman Demirel Üniversitesinin bu alanda 5 tane yayını bulunmaktadır.



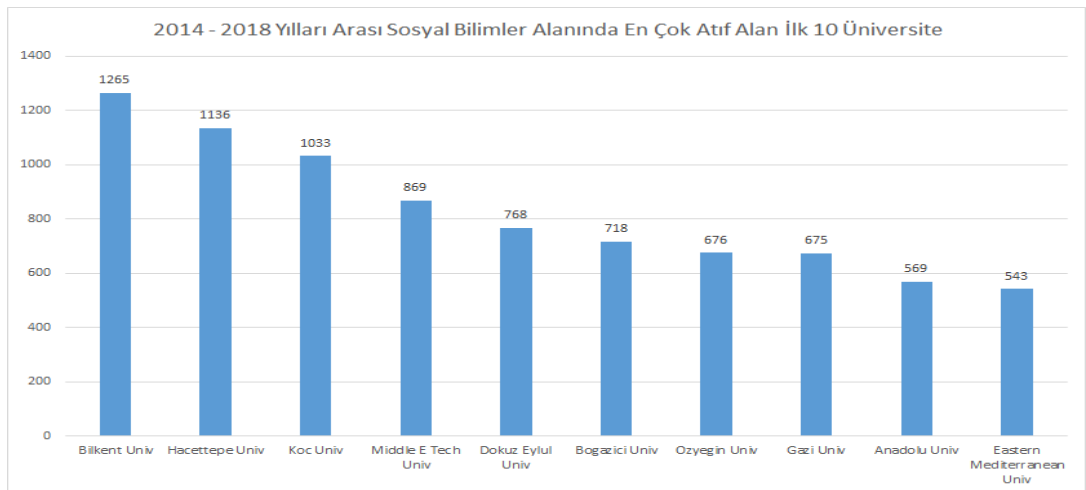
Şekil 4.9. 2014-2018 Yılları arası en çok atıf alan 10 üniversite

Şekil 4.9.'da 2014-2018 yılları arasında üniversite bazında, Türkiye adresli makalelerin aldığı atıf sayıları analiz edilmiş ve ilk 10 üniversiteye ait grafik verilmiştir. Şekil 4.5.'te makale sayısına göre ilk 10 üniversite verilmiştir. İlk 4 üniversiteye bakıldığında hem en çok makale üreten üniversiteler hem de en çok atıf alan üniversitelerin aynı olduğu görülmüştür. İstanbul Teknik Üniversitesi ve Karadeniz Teknik Üniversitesi makale sayısına göre sıralandığında ilk 10 üniversite arasında bulunmamasına rağmen, atıf sayısına göre sıralandığında ilk 10 üniversite arasında yer almıştır. Marmara ve Ondokuz Mayıs Üniversiteleri ise makale sayısına göre sıralandığında ilk 10 üniversite arasında olmasına rağmen atıf sayısına göre bakıldığında ilk 10 üniversite arasında yer alamamıştır. Makale sayısına göre bakıldığında 17. sırada olan Süleyman Demirel Üniversitesi, atıf sayısına göre sıralandığında 9587 atıf ile 18. Sırada yer almaktadır.



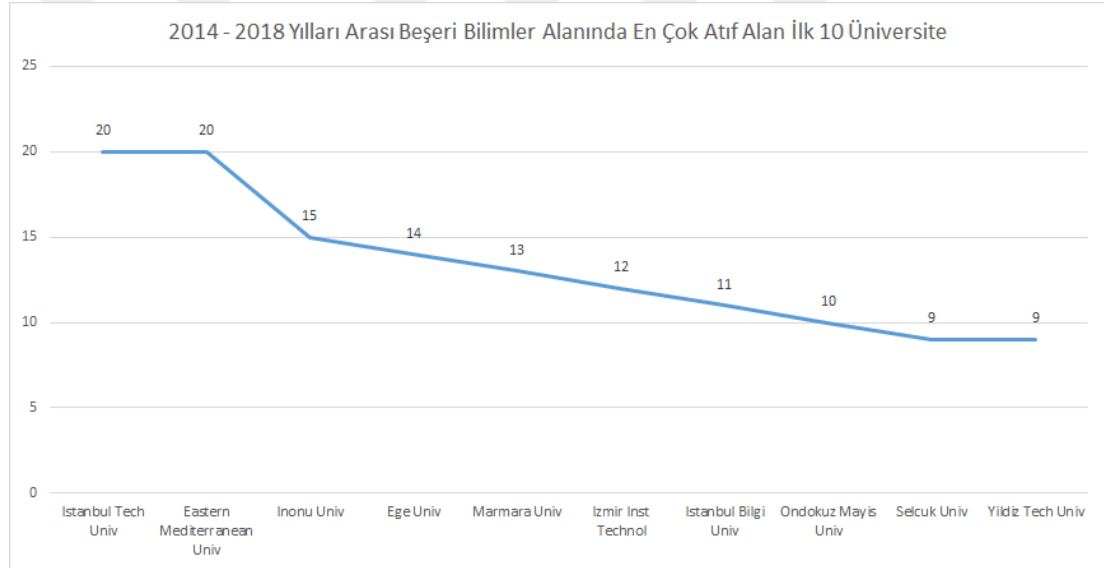
Şekil 4.10. 2014-2018 Yılları arası bilim ve teknoloji alanında en çok atıf alan 10 üniversite

Şekil 4.10.'da 2014-2018 yılları arasında üniversite bazında, bilim ve teknoloji alanındaki Türkiye adresli en çok atıf alan makaleler analiz edilmiştir. Türkiye'de en çok yayın yapılan alan bilim ve teknoloji alanı olduğu için Şekil 4.10. ile Şekil 4.9. arasında sadece sayıların bir miktar değiştiği, sıralamada ise Erciyes ve Atatürk Üniversitelerinin yer değiştiği görülmüştür.



Şekil 4.11. 2014-2018 Yılları arası sosyal bilimler alanında en çok atıf alan 10 üniversite

Şekil 4.11.'de, 2014-2018 yılları arasında üniversite bazında, sosyal bilimler alanındaki Türkiye adresli en çok atıf alan makaleler analiz edilmiştir. Sosyal bilimler alanında makale sayısına göre bakıldığında sıralamada geride olan üniversitelerin atıf sayısında öne çıktıkları da görülmektedir. Bilkent üniversitesi makale sayısına göre yedinci sırada olmasına rağmen atıf sayısında birinci sırada yer almıştır. Hacettepe Üniversitesi makale sayısında birinci sırada yer alırken atıf sayısına göre ikinci sırada yer almıştır. Koç Üniversitesi, Özyeğin Üniversitesi Anadolu Üniversitesi ve Doğu Akdeniz Üniversitesi makale sayısına göre ilk 10 arasında değilken, atıf sayısına göre sıralandığında ilk 10 da yer aldıkları görülmektedir. Süleyman Demirel Üniversitesi sosyal bilimler alanında 66 atfa sahip olduğu görülmüştür.



Şekil 4.12. 2014-2018 Yılları arası beşeri bilimler alanında en çok atıf alan 10 üniversite

Şekil 4.12.'de, 2014-2018 yılları arasında üniversite bazında, beşerî bilimler alanındaki Türkiye adresli en çok atıf alan makaleler analiz edilmiştir. Beşerî bilimler alanı Türkiye' de en az makale yazılan alan olduğu için çok az miktarda atıf vardır. Makale sayısına göre bakıldığında da birinci sırada yer alan İstanbul Teknik Üniversitesi en çok atıf alan üniversiteler arasında da birinci sırada yer almaya devam etmektedir. Şekil 4.11. ve Şekil 4.8. incelendiğinde birinci hariç sıralamanın tamamının değiştiği görülmektedir. Atıf sayılarının makale

sayılarından az olduđu fark edilmektedir. Süleyman Demirel Üniversitesi'nin bu alanda 2 atfa sahip olduđu görülmüştür.



5. SONUÇ VE ÖNERİLER

Bu çalışma sonucunda en çok beraber yayın yapan yazarlar ve üniversiteler tespit edilmiş, birliktelik kuralları çıkartılmış ve bunlara bağlı olarak tahmin sonuçları üretilmiştir. Aynı zamanda üniversitelerin genel ve kategori bazında yayınları incelenmiştir. Bu inceleme sonrasında makale sayısı ve atıf sayılarını baz alan sıralamalar çıkartılmıştır. Üniversiteler karşılaştırılırken 2014-2018 yılları arasındaki veriler kullanılmıştır. Bunun sebebi ise bazı üniversitelerin eski olmasına rağmen bazı üniversitelerin yakın geçmişte kurulmuş olmasıdır. Adil bir karşılaştırma olması için 2014 öncesi veriler dikkate alınmamıştır. Veriler 2019 yılının ilk çeyreğinde topladığı için 2019 yılına ait tüm veriler bu çalışmada değerlendirilememiştir. Bu nedenle 2018 yılı sınır olarak belirlenmiştir.

İnceleme sonrasında en çok makale yayınlayan ve en çok atıf alan üniversitelere bakıldığında İstanbul Üniversitesi, Hacettepe Üniversitesi, Gazi Üniversitesi, Ege Üniversitesi ve Ankara Üniversitesinin ilk 10' da olduğu görülmüştür. Aynı zamanda 2008 ve 2015 yıllarında akademik makale sayılarının ciddi miktarda arttığı görülmüştür. Ülkemizde en fazla bilim ve teknoloji alanında yayın yapıldığı fark edilmiştir. En az yayının ise beşerî bilimler alanında olduğu tespit edilmiştir.

Birliktelik analizi sonuçları incelendiğinde, en çok birlikte çalışma yapan ilk 3 üniversitenin şu şekilde olduğu görülmüştür: 783 akademik çalışma ile Gazi ve Hacettepe üniversiteleri, 763 akademik çalışma ile Gazi ve Ankara üniversiteleri, 676 çalışma ile Ankara ve Hacettepe üniversiteleridir. Beraber çalışma yapan üniversiteler incelendiğinde ilk 3 üniversitede olduğu gibi beraber çalışma yapmayı etkileyen en önemli faktörün buldukları konum olduğu görülmektedir.

En çok birlikte akademik çalışma yapan yazarların ise şu şekilde olduğu görülmüştür: 95 akademik çalışma ile B. Cakir ve R. Ersoy, 78 çalışma ile M. Soylak ve M. Tuzen, 77 akademik çalışma ile Y. Atalay ve D. Avci.

Verilerin elde edilmesi sırasında kullanılan akademik çalışma indekslerini barındıran WoS platformunun eksikleri ve yanlışları olduğu tespit edilmiştir. WoS üzerinde makaleler ve yazarlar birleştirilirken isim benzerliği olan kişiler arasında makalelerin karıştığı görülmüştür. Aynı zamanda yazar ve okul isimlerinde yanlışlıklar olduğu tespit edilmiştir. Bu yanlışlıkların az olmasının tahmin ve çıkarımları çok etkilemediği ve az miktarda bir yanıltmaya sebep olduğu düşünülmektedir.

Bu çalışmadaki kadar büyük veri seti ve büyük veri araçları kullanılarak bir bibliyometrik analiz çalışması literatürde yer almamaktadır. Bu özellikleri ile çalışma literatüre katkı sunmaktadır.

Bu çalışmadan sonra akademik çalışmalar fakülte ve birim bazına indirgenerek yapılabilir. En çok birlikte çalışma yapan fakülte, birimler ve bu birimlerdeki yazarların birliktelik analizleri yapılabilir. Aynı zamanda en çok akademik çalışma yapan fakülte ve birimler sene bazında inceleyerek bibliyometrik analiz yapıp geleceğe dönük tahminler yapılabilir.

KAYNAKLAR

- Akgül, D.M.K., 2020. Büyük (Mega) Veriler ve Süper İşlemler Çağı. Erişim Tarihi: 09.02.2020. <https://anahtar.sanayi.gov.tr/tr/news/buyuk-mega-veriler-ve-super-islemler-cagi-1/526>.
- Akgün, B., 2016. Apache Spark Tabanlı Destek Vektör Makineleri İle Akan Büyük Veri Sınıflandırma. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 73s, İstanbul
- Al, U., 2008. Türkiye'nin bilimsel yayın politikası: Atıf dizinlerine dayalı bibliyometrik bir yaklaşım. Hacettepe Üniversitesi, Sosyal Bilimler Enstitüsü, Doktora Tezi, 230s, Ankara.
- Asan, A., 2004. ISI'nin Kullandığı İndeksler: SCI-Expanded, SSCI Ve AHCI: Tarihsel Gelişim, Bugünkü Durum Ve Etki Faktörü (If). Ortaokul On-Line Mikrobiyoloji Dergisi, 2(5), 1-21.
- Aykanat, Z., Yıldız, T., 2016. İşletme ve Yönetim Alanındaki "Stratejik Yönetim" Kavramının Evrimsel Gelişiminin Bilimsel Haritalama Yöntemiyle İncelenmesi. Ulusal Yönetim ve Organizasyon Kongresi, 29-31 Mayıs, İstanbul, 29-31.
- Bakır, N., 2013. Pazarlama Alanında Yapılan Doktora Tezlerinin Kategorik Olarak Değerlendirilmesi (1994-2012). Öneri Dergisi, 10(40), 1-13.
- Bayram, Ö.G., 1998. Atıf Verisi (Citation Data) Ve Enformetrik Yasalar: Türk Kütüphanecilik Literatüründeki Doktora Tezleri Üzerinde Bir Uygulama. Türk Kütüphaneciliği, 12(1), 21-32.
- Birinci, H.G., 2008. Turkish Journal Of Chemistry'nin Bibliyometrik Analizi. Bilgi Dünyası, 9(2), 348-369.
- Blumberg, R., Atre, S., 2003. The Problem With Unstructured Data. Dm Review, 13(42-49), 62.
- Bolat, N.A., 2019. NOSQL ve RDBSM Yapıların Performans Karşılaştırması. Beykent Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 59s, İstanbul.
- Cawkell, T., Garfield, E., 2001. Institute for Scientific Information. Information Services Use, 21(2), 79-86.

- Chen, C.P., Zhang, C.-Y., 2014. Data-Intensive Applications, Challenges, Techniques And Technologies: A Survey On Big Data. *Information sciences*, 275, 314-347.
- Cömert, Z., Kocamaz, A.F., Çibuk, M., 2015. Web Tabanlı Hibrit Bir Uygulama Modeliyle Personel Bilgi Sistemi Tasarımı. *Akademik Bilişim*.
- Craft, R.C., Leake, C., 2002. The Pareto Principle In Organizational Decision Making. *Management Decision*, 40(8), 729-733.
- Cukier, K., 2010. Data, Data Everywhere: A Special Report On Managing Information. *Economist Newspaper*.
- Çelik, H., 2008. Web Servisleri İle Elektronik Ticaret Uygulaması. Haliç Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 58s, İstanbul.
- Davis, K., 2012. *Ethics Of Big Data: Balancing Risk And Innovation*. O'Reilly Media, Inc.
- Dean, J., Ghemawat, S., 2004. Mapreduce: Simplified Data Processing On Large Clusters. *Communications of the ACM*, 51(1), 107-113.
- Dierbach, C., 2012. *Introduction to Computer Science using Python: A Computational Problem-Solving Focus*. Wiley Publishing.
- Doğan, M., 2014. Büyük Veri'nin Kişiler Ve Kurumlar Üzerindeki Etkileri. İstanbul Bilgi Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, 59s, İstanbul.
- Dumrul, A.G.C., Aysu, A.G.A., 2006. Erciyes Üniversitesi İİ BF Dergisinde Yayınlanan Makaleler: Değerlendirme ve Bibliyografya (1981-2005). *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, (27).
- Egghe, L., Rousseau, R., 1990. *Introduction To Informetrics: Quantitative Methods In Library, Documentation And Information Science*. Elsevier Science Publishers.
- Erol, E., 2020. Erişim Tarihi: 04.01.2020. <https://erkanerol.github.io/post/distributed-systems-1/>.

- Erpolat, S., 2012. Otomobil Yetkili Servislerinde Birliktelik Kurallarının Belirlenmesinde Apriori ve FP-Growth Algoritmalarının Karşılaştırılması. *Anadolu Üniversitesi Sosyal Bilimler Dergisi*, 12(1), 151-166.
- Evren, S., Kozak, N., 2014. Bibliometric Analysis Of Tourism And Hospitality Related Articles Published In Turkey. *Anatolia*, 25(1), 61-80.
- Forsman, M., 2008. Do We Need A Qualitative Approach In Bibliometric Studies. *25(2)*, 249–254.
- Galvin, P.B., Gagne, G., Silberschatz, A., 2003. *Operating System Concepts*. Wiley, 960s.
- Garcia Molina, H., Ullman, J., Widom, J., 2009. *Database Systems: The Complete Book*, Chapter 7. Pearson International Editing, New Jersey.
- Garfield, E., 1965. Can Citation Indexing Be Automated. *Statistical Association Methods For Mechanized Documentation, Symposium Proceedings*, 189-192.
- Garfield, E., Merton, R.K., 1979. *Citation indexing: Its theory and application in science, technology, and humanities*. Wiley New York.
- Glänzel, W., 2009. *History Of Bibliometrics And Its Present-Day Tasks In Research Evaluation*.
- Gökkurt, Ö., 1994. Enformetri, Bradford Yasası ve Citation İndeks Informetrics, *Bradford's Law and Citation Index*. *Türk Kütüphaneciliği*, 8.
- Gökkurt, Ö., 1998. Atıf Verisi (Citation Data) Ve Enformetrik Yasalar: Türk Kütüphanecilik Literatüründe Doktora Tezleri Üzerine Bir Uygulama. *Türk Kütüphaneciliği*, 12(1), 21-32.
- Gu, L., Li, H., 2013. Memory Or Time: Performance Evaluation For Iterative Operation On Hadoop And Spark. *EEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing*.
- Hadoop, A., 2020. HDFS Architecture. Erişim Tarihi: 02.02.2020. <https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>.

- Hilbert, M., López, P., 2011. The World's Technological Capacity To Store, Communicate, And Compute Information. Scienceexpress, 332(6025), 60-65.
- Inc., O.R.M., 2012. Big Data Now. O'Reilly Media.
- Ismail, N., 2017. What Are The Real Opportunities For Big Data In The Digital World? Erişim Tarihi: 09.02.2020. <https://www.information-age.com/real-opportunities-big-data-digital-world-123469428/>.
- Jiawei, H., Kamber, M., Pei, J., 2000. Data Mining: Concepts And Techniques. Morgan Kaufmann Publishers, 560s.
- Jogaratham, G., McCleary, K.W., Mena, M.M., Yoo, J.J.-E., 2005. An Analysis Of Hospitality And Tourism Research: Institutional Contributions. Journal of Hospitality Tourism Research, 29(3), 356-371.
- Koehler, W., 2001. Information Science As "Little Science": The Implications Of A Bibliometric Analysis Of The Journal Of The American Society For Information Science. Scientometrics, 51(1), 117-132.
- Kozak, N., 1994. Anatolia Dergisi'nde Yayımlanan Yazılar Üzerine Bir İnceleme. Anatolia: Turizm Araştırmaları Dergisi, 5(3), 22-33.
- Kozak, N., 2003. Türkiye'de Yayımlanan Akademik Dergilerin Niteliklerindeki Zaman İçerisindeki Değişim Nedenleri: Sağlık, Sosyal Ve Teknik Bilim Alanlarında Yayımlanan Dergiler Üzerine Bir İnceleme. Bilgi Dünyası, 4(2), 146-174.
- Kroenke, D.M., Auer, D.J., 1999. Database Processing: Fundamentals. Pearson Education, 624s.
- Küçük, T.T., 2019. Apache Spark nedir? Ne iş yapar? Erişim Tarihi: 15.02.2020. <https://medium.com/5bayt/apache-spark-nedir-ne-i%C5%9F-yapar-5797c28eb95>.
- Laney, D., 2001. 3d Data Management: Controlling Data Volume, Velocity And Variety. Meta Group Research Note, 6(70), 1.
- Louridas, P., 2006. Soap and web services. IEEE Software, 23(6), 62-67.

McBurney, M.K., Novak, P.L., 2002. What Is Bibliometrics And Why Should You Care? Proceedings. IEEE International Professional Communication Conference, 108-114.

McCann, D., 2001. 80-20 vision. Dairy Industries International, 66(9), 25-25.

Medium. 2020. Erişim Tarihi: 09.02.2020.<https://medium.com/@mbilgil0/http-metotlar%C4%B1-http-request-methods-90d57d574dfa>.

Metin, T.C., 2013. Boş Zaman Literatürünün Dünya'daki Gelişimi: Leisure Science Dergisinde Yayımlanan Makaleler Üzerine İnceleme. Anadolu Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, 234s, Eskişehir.

MongoDB. 2019. Erişim Tarihi: 20.11.2019.<https://www.mongodb.com/>.

Mumbaikar, S., Padiya, P., Publications, R., 2013. Web Services Based On Soap And Rest Principles. International Journal of Scientific, 3(5), 1-4.

Niemi, A.W., 1975. Journal Publication Performance During 1970-1974: The Relative Output Of Southern Economics Departments. Southern Economic Journal, 42(1), 97-106.

Önder, E., 2005. Yönetim Bilişim Sistemleri Kapsamında Web Tabanlı İlişkisel Veritabanı Yönetim Sistemleri ve Bir Uygulama. İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, 251s, İstanbul.

Özdeş, M., 2017. Büyük Veri Araçlarını Kullanarak Duygu Analizi Gerçekleştirimi. Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 75s, Denizli.

Pandey, A., Pardasani, K., 2009. Rough Set Model For Discovering Multidimensional Association Rules. International Journal of Computer Science and Network Security, 9(6), 159.

Phene, A., Guisinger, S., 1998. The Stature Of The Journal Of International Business Studies. Journal of International Business Studies, 29(3), 621-631.

PostgreSQL. Erişim Tarihi: 09.02.2020.<https://www.postgresql.org/about/>.

Pritchard, A., 1969. Statistical Bibliography or Bibliometrics. Journal of documentation, 25(4), 348-349.

Python. 2020. Eriřim Tarihi: 07.01.2020.<https://python.org>.

Rao, I.R., Neelameghan, A., 1992. From Librametry To Informetrics: An Overview And Ranganathan's Contributions. *Libri*, 42(3), 242-257.

Sakar, G., Cerit, A., 2013. Uluslararası Alan İndekslerinde Türkiye Pazarlama Yazini: Bibliyometrik Analizler ve Nitel Bir Arastırma. *Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 27(4), 37-62.

Sammer, E., 2012. Hadoop operations. O'Reilly Media, Inc.

Satoh, F., Nakamura, Y., Mukhi, N.K., Tatsubori, M., Ono, K., 2008. Methodology And Tools For End-To-End Soa Security Configurations. 2008 IEEE Congress on Services-Part I, 307-314.

Scientific, T., 2007. ISI Web of knowledge. Eriřim Tarihi:

Sengupta, I., 1992. Bibliometrics, Informetrics, Scientometrics And Librametrics: An Overview. *Libri*, 42(2), 75-98.

Seyrek, K., Ayas, T., DİNÇER, A., Uraz, B., 2020. Kişiselleştirilmiş Coğrafi Web Servislerinin Hazırlanılması. Eriřim Tarihi: 26.02.2020.http://uzalcbs.org/wp-content/uploads/2016/11/2012_069.pdf.

Shvachko, K., Kuang, H., Radia, S., Chansler, R., 2010. The Hadoop Distributed File System. 2010 IEEE 26th Symposium On Mass Storage Systems And Technologies (Msst), 1-10.

Smith, L.C., 1981. Citation analysis. *Library Trends*, 30, 83 – 106.

Spark, A., 2020a. Cluster Mode Overview. Eriřim Tarihi: 12.02.2020.<https://spark.apache.org/docs/latest/cluster-overview.html>.

Spark, A., 2020b. Eriřim Tarihi: 10.01.2020.<https://spark.apache.org/sql/>.

Subramanyam, K., 1983. Bibliometric Studies Of Research Collaboration: A Review. *Journal of information Science*, 6(1), 33-38.

Taşkin, Z., Çakmak, T., 2010. Başlangıcından Bugüne Bilgi Dünyası Dergisi'nin Bibliyometrik Profili. *Bilgi Dünyası*, 11(2), 332-348.

- Tatar, C.C., 2010. Bilimsel Dergilerdeki Müzik Makalelerinin Bibliyometrik Profili. Abant İzzet Baysal Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, 82s, Bolu.
- Ulu, S., Akdağ, M., 2015. Yayınlanan Hakem Denetimli Makalelerin Bibliyometrik Profili: Selçuk İletişim Dergisi Örneği. Selçuk Üniversitesi İletişim Fakültesi Akademik Dergisi, 9(1), 5-21.
- Wang, C.-C., Hu, W.-C., 2011. Bibliometric Analysis Of Advertising Endorser Research In Marketing. International Proceedings of Economics Development Research, 3, 102-106.
- White, T., 2012. Hadoop: The Definitive Guide. O'Reilly Media, Inc.
- Yalçın, H., 2010. Millî Folklor Dergisinin Bibliyometrik Profili (2007-2009). Millî Folklor, 22(85), 205-211.
- Yavuz, G., Aytekin, S., Akçay, M., 2012. Apache Hadoop Ve Dağıtık Sistemler Üzerindeki Rolü. Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü Dergisi, (027), 43-54.
- Yıldırım, O., 2019. Sql Veritabanlarından Nosql Veritabanlarına Veri Göçü Aracı Geliştirmesi. Düzce Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 53s, Düzce.
- Yılmaz, M., 2005. 80/20 Kuralı. Türk Kütüphaneciliği, 19(3), 308-320.
- Yılmaz, M., 2007. Price Yasası Ve Türkiye’de Fikrî Mülkiyet Hakları Literatürü. Belge ve Bilgi Araştırmaları Dergisi, 1(1).
- Zan, B.U., 2012. Türkiye’de Bilim Dallarında Karşılaştırmalı Bibliyometrik Analiz Çalışması. Ankara Üniversitesi, Sosyal Bilimler Enstitüsü, Doktora Tezi, 264s, Ankara.

ÖZGEÇMİŞ

Adı Soyadı : Sinan DURGUT
Doğum Yeri ve Yılı : Finike, 1995
Medeni Hali : Bekar
Yabancı Dili : İngilizce
E-posta : sinandurgut95@gmail.com

Eğitim Durumu

Lise : Kumluca Anadolu Öğretmen Lisesi, 2013
Lisans : SDÜ, Mühendislik Fakültesi, Bilgisayar Mühendisliği, 2018

Mesleki Deneyim

Artin Bilişim Teknolojileri ve Yazılım Hizmetleri Ticaret Li 2017-2019
Tusaş-Türk Havacılık ve Uzay Sanayi Anonim Şirketi 2019-.....(halen)