

**KOD DEPOLARINDA BİRLİKTE DEĞİŞEN DOSYALARIN  
SOSYAL AĞ ANALİZİ TEKNİKLERİYLE ANALİZİ**

Levent Türnüklü

161402207

Orcid: 0000-0002-0053-7727

**YÜKSEK LİSANS TEZİ**

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Yüksek Lisans Programı

Danışman: Dr. Öğr. Üyesi Volkan Tunalı

İstanbul,

T. C. Maltepe Üniversitesi

Lisansüstü Eğitim Enstitüsü




Ocak, 2020



# JÜRİ VE ENSTİTÜ ONAYI

## JÜRİ VE ENSTİTÜ ONAYI

LEVENT TÜRNÜKLÜ'ün "Kod Depolarında Birlikte Değişen Dosyaların Sosyal Ağ Analizi Teknikleriyle Analizi" başlıklı tezi 21.01.2020 tarihinde aşağıdaki jüri tarafından değerlendirilerek "Maltepe Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği" nin ilgili maddeleri uyarınca Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans/Doktora tezi oy birliğiyle/oy çokluğuyla, başarılı/başarısız olarak kabul edilmiştir.

Ünvanı, Adı ve Soyadı	İmza
Üye (Tez Danışmanı) Dr. Öğr. Üyesi Volkan TUNALI	
Üye Dr. Öğr. Üyesi Buket DOĞAN	
Üye Dr. Öğr. Üyesi Erdal GÜVENOĞLU	

Prof. Dr. Belma AKŞİT  
Enstitü Müdürü



# ETİK İLKE VE KURALLARA UYUM BEYANI

 maltepe üniversitesi	<b>LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ ETİK İLKE VE KURALLARA UYUM BEYANI</b>	Doküman No	FR-178
		İlk Yayın Tarihi	01.03.2018
		Revizyon Tarihi	23.01.2020
		Revizyon No	01
		Sayfa	1

21/01/2020

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bulguların sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilmeyen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; çalışmamın Maltepe Üniversitesinde kullanılan "bilimsel intihal tespit programı" ile tarandığını ve öngörülen standartları karşıladığımı beyan ederim.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm.



Levent Türnüklü

Hazırlayan: Enstitü Sekreterliği

Onaylayan: Kalite Yönetim Koordinatörlüğü

# İNTİHAL

## KOD DEPOLARINDA BİRLİKTE DEĞİŞEN DOSYALARIN SOSYAL AĞ ANALİZİ TEKNİKLERİYLE ANALİZİ

ORJİNALLİK RAPORU

% <b>11</b>	% <b>9</b>	% <b>4</b>	% <b>10</b>
BENZERLİK ENDEKSİ	İNTERNET KAYNAKLARI	YAYINLAR	ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

<b>1</b>	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Öğrenci Ödevi	% <b>3</b>
<b>2</b>	developer.github.com İnternet Kaynağı	% <b>3</b>
<b>3</b>	cs.bazel.build İnternet Kaynağı	% <b>1</b>
<b>4</b>	www.gitbrowse.com İnternet Kaynağı	% <b>1</b>
<b>5</b>	Submitted to Middlesex University Öğrenci Ödevi	<% <b>1</b>
<b>6</b>	Submitted to Texas A&M University, College Station Öğrenci Ödevi	<% <b>1</b>
<b>7</b>	Submitted to Fırat Üniversitesi Öğrenci Ödevi	<% <b>1</b>
<b>8</b>	www.appsrox.com	

*Volkon Turali*  
Dr. Öğr. Üyesi Volkan TURALI  
27.01.2020

## TEŐEKKÜR

Tez danıŐmanlıđımı üstlenerek araŐtırma konusunun seđimi ve yürütölmesi sırasında kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösterici ve destek olan ayrıca ilgisini ve önerilerini göstermekten kaçınmayan saygıdeđer danıŐman hocam sayın Dr. Öğr. Üyesi Volkan Tunalı'ya, tez dönemi boyunca desteđini esirgemeyen deđerli eŐim Gamze'ye ve kıymetli ođlum Kerem'e sonsuz teŐekkür ederim.

Levent Türnükölü

Ocak 2020

## ÖZ

# KOD DEPOLARINDA BİRLİKTE DEĞİŞEN DOSYALARIN SOSYAL AĞ ANALİZİ TEKNİKLERİYLE ANALİZİ

Levent Türnüklü  
Yüksek Lisans Tezi  
Bilgisayar Mühendisliği Anabilim Dalı  
Bilgisayar Mühendisliği Yüksek Lisans Programı  
Danışman: Dr. Öğr. Üyesi Volkan Tunalı  
Maltepe Üniversitesi Lisansüstü Eğitim Enstitüsü, 2020

Bu tez çalışmasında, 11 farklı yazılım projesindeki birlikte değişen dosyaların arasındaki ilişki ağları modellendikten sonra karmaşık ağ analizi ölçütleri ile değerlendirilmiştir. Veri seti, birlikte değişen dosyalar modellenerek oluşturulmuştur. Görsel analiz, merkezilik analizi, topluluk analizi ve yapısal analiz yöntemleri ile analiz süreçleri gerçekleştirilmiştir. Yapısal analizde, birlikte değişim ağlarının ölçekten bağımsız gerçek hayat ağlarına ait özelliklere sahip olup olmadığını değerlendirmek için modellenen rassal ağlar ile karşılaştırmalı analizi yapılmıştır. Bu çalışmada, karmaşık ağ analizi ölçütlerinin kullanılmasının getireceği faydaların incelenmesi ve karşılaştırmalı olarak sunulmasıyla benzer yapıdaki ağ modelleri için etkili bir analiz altyapısının elde edilmesi amaçlanmıştır.

**Anahtar Kelimeler:** Ağ Bilimi, Karmaşık Ağ Analizi, Yazılım Proje Yönetimi.

## ABSTRACT

### ANALYSIS OF CO-CHANGED FILES IN CODE REPOSITORIES WITH SOCIAL NETWORK ANALYSIS TECHNIQUES

Levent Türnüklü

Master Thesis

Department of Computer Engineering

Computer Engineering Programme

Advisor: Asst. Prof. Volkan Tunalı

Maltepe University Graduate School, 2020

In this thesis, co-changed files networks in 11 different software projects are modeled and analyzed with complex network analysis measures. The data set was created by modeling co-changed files. Visual analysis, centrality analysis, community analysis and structural analysis methods and analysis processes were performed. In the structural analysis, a comparative analysis was performed with the random networks modeled to evaluate whether the co-changed files networks have the characteristics of real life networks. In this study, it is aimed to obtain an effective analysis infrastructure for similar network models by examining and comparing the benefits of using complex network analysis measures.

**Keywords:** Network Science, Complex Network Analysis, Software Project Management.

# İÇİNDEKİLER

<b>ETİK İLKE VE KURALLARA UYUM BEYANI</b> .....	iii
<b>İNTİHAL</b> .....	iv
<b>TEŞEKKÜR</b> .....	v
<b>ÖZ</b> .....	vi
<b>ABSTRACT</b> .....	vii
<b>İÇİNDEKİLER</b> .....	viii
<b>TABLolar LİSTESİ</b> .....	x
<b>ŞEKİLLER LİSTESİ</b> .....	xi
<b>EKLER LİSTESİ</b> .....	xii
<b>SİMGELER VE KISALTMALAR</b> .....	xiii
<b>ÖZGEÇMİŞ</b> .....	xiv
<b>BÖLÜM 1. GİRİŞ</b> .....	1
<b>BÖLÜM 2. İLGİLİ ÇALIŞMALAR</b> .....	3
<b>BÖLÜM 3. AĞ ANALİZİ</b> .....	6
Temel Graf Bilgisi.....	6
3.1.1. Graf.....	7
3.1.2. Düğüm (Node).....	8
3.1.3. Kenar (Edge).....	8
Graf Gösterim Şekilleri.....	9
3.1.4. Komşuluk Matrisi.....	10
3.1.5. Komşuluk Listesi.....	10
Ağ Ölçütleri.....	11
3.1.6. Derece Merkeziliği.....	11
3.1.7. Yakınlık Merkeziliği.....	13
3.1.8. Arasındalık Merkeziliği.....	14
3.1.9. Özvektör Merkeziliği.....	15
3.1.10. Kümelenme Katsayısı.....	16
3.1.11. Ortalama derece.....	17
3.1.12. Derece Dağılımı.....	17



Topluluk Analizi .....	18
3.1.13. Gerçek Hayat Ağları.....	20
3.1.14. Derece Dağılımı .....	20
3.1.15. Ortalama Yol Uzunluğu .....	20
3.1.16. Kümeleme Katsayısı .....	21
<b>BÖLÜM 4. YÖNTEM.....</b>	<b>22</b>
Araçlar .....	22
4.1.1. Github.....	22
4.1.2. Gephi .....	22
4.1.3. Cytoscape .....	22
Yöntem .....	23
4.1.4. Veri Toplama .....	23
4.1.5. GraphQL .....	25
4.1.6. Github API .....	25
4.1.7. Octokit.....	25
4.1.8. Yöntem 1: Mevcut dosyalar üzerinden veri toplama.....	25
4.1.9. Yöntem 2: Tüm Check-in kayıtları ile veri toplama .....	26
Ağ Modelleme .....	28
<b>BÖLÜM 5. BULGULAR VE YORUMLAR.....</b>	<b>30</b>
Karmaşık Ağ Analizi .....	30
5.1.1. Görsel Ağ Analizi.....	30
5.1.2. Yapısal Ağ Analizi.....	31
5.1.3. Derece Dağılımları .....	31
5.1.4. Merkezilik Analizi .....	32
5.1.5. Topluluk Analizi .....	33
<b>BÖLÜM 6. SONUÇ.....</b>	<b>34</b>
EK'LER.....	35
KAYNAKÇA.....	59

## TABLÖLAR LİSTESİ

Tablo 1: Komşuluk matrisi .....	10
Tablo 2: Komşuluk listesi .....	11
Tablo 3. Seçilen proje bilgileri .....	23
Tablo 4. Proje konusu .....	24
Tablo 5. Yöntem 1 işlem sonucu .....	26
Tablo 6. Yöntem 2 işlem sonucu .....	26
Tablo 7. Yöntem 2 model deseni .....	27
Tablo 8: Ağ özet bilgileri.....	28
Tablo 9: Rastgele ağ analiz sonuçları .....	31

## ŞEKİLLER LİSTESİ

Şekil 1: Class of 1880/81 ağı görünümü.....	7
Şekil 2: Yönsüz Graf.....	8
Şekil 3: Yönlü Graf.....	9
Şekil 4: Yönlü ağırlıklı graf.....	9
Şekil 5: Derece merkezliği.....	12
Şekil 6: İç derece merkeziliği .....	12
Şekil 7: Dış derece merkeziliği.....	13
Şekil 8: Yakınlık Merkeziliği .....	14
Şekil 9: Arasında merkezilik gösterimi .....	15
Şekil 10: Özvektör merkeziliği .....	16
Şekil 11: Class of 1880/81 ağının derece dağılımı .....	18
Şekil 12: Modülarite ağı gösterimi .....	19
Şekil 13. Kuvvet yasası dağılımı .....	20
Şekil 14. Merkezilik analiz gösterim ifadeleri.....	32

## EKLER LİSTESİ

Ek 1. Örnek işlem(check-in) açıklaması.....	35
Ek 2. Ağ görüntüleri .....	38
Ek 3. Ağ analizi sonuçları.....	39
Ek 4. Derece dağılımları sonuçları .....	40
Ek 5. Merkezilik analizi görsel sonuçları .....	43
Ek 6. Merkezilik analizi ilk 10 dosya sonuçları.....	49
Ek 7. Topluluk analizi sonuçları .....	57

## SİMGELER VE KISALTMALAR

- API** : Application Programming Interface
- ClusterOne** : Clustering with Overlapping Neighborhood Expansion
- JSON** : JavaScript Object Notation
- SNA** : Social Network Analysis
- SQL** : Structured Query Language



# ÖZGEÇMİŞ

**Levent Türnüklü**

## **Bilgisayar Mühendisliği Anabilim Dalı**

### **Eğitim**

<i>Derece</i>	<i>Yıl</i>	<i>Üniversite, Enstitü, Anabilim/Anasanat Dalı</i>
Y.Ls.	2020	Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı
Ls.	2011	Kocaeli Üniversitesi, Mühendislik Fakültesi Bilgisayar Mühendisliği Anabilim Dalı
Lise	2005	Amasya Yabancı Dil Ağırlıklı Atatürk Lisesi

### **İş/İstihdam**

<i>Yıl</i>	<i>Görev</i>
2014-	Yazılım Uzmanı. Akbank Bilgi Teknolojileri A.Ş.
2013-14	Yazılım Uzmanı. Omsan Lojistik A.Ş.
2011-13	Yazılım Uzmanı. Logo Yazılım Sanayi Tic. A.Ş.

### **Kişisel Bilgiler**

Doğum yeri ve yılı	: Amasya, 1987	Cinsiyet: E
Yabancı diller	: İngilizce (iyi)	
GSM / e-posta	: 0536 607 60 28 / leventturnuklu@hotmail.com	

## BÖLÜM 1. GİRİŞ

Bir yazılım projesi birbirinden bağımsız geliştiricilerin oluşturduğu birtakım dosyalardan oluşmaktadır. Yazılım projelerinin birbirine bağlanması ile de yazılım sistemleri oluşmaktadır. Geliştirilen sistemler yeni ihtiyaçlarla birbirine daha sık bağlarla bağlanarak kolayca çözülemeyen, karmaşık bir ağa dönüşür. Bu karmaşık ağ yazılım bakım maliyetlerini çok artırmaktadır.

Yazılım geliştirme yaşam döngüsünün yazılım bakım aşaması şüphesiz çok önemlidir ve zordur. Bazı istatistiklere göre [1], yıllık yazılım harcamalarının yaklaşık yüzde 60 ila 90'ı mevcut yazılım sistemlerinin bakımını içermektedir. Bakım aşaması, bir yazılım sisteminde, müşterinin kabulü üzerine müşteriye dağıtıldıktan sonra yapılan değişikliklerden oluşur. Bu aşamada meydana gelen değişiklikler bazen son derece zordur, çünkü ilgili program modülleri arasındaki etkiyi ve altta yatan bağımlılığı uygun şekilde değerlendirmeden yapılan değişiklikler yazılım sisteminde hatalara neden olabilir. Her zaman program modülleri arasında düşük bağımlılıklar olması arzu edilir [2]. İyi tasarlanmış bir yazılım sistemi, her zaman bağımlılığı en aza indirir [3].

Yazılım bakımı, yazılım geliştirme yaşam döngüsünün önemli ve zorlu bir aşamasıdır; çünkü bu aşamada program modülleri arasındaki bağımlılıkların farkında olmadan değişiklik yapılması, yazılım sisteminde hatalara neden olabilir. Bu sebeple yazılan kodların bir kod deposunda saklanması ve tarihsel olarak değişikliklerin kayıt altında tutulması tavsiye edilir.

Proje süresince yapılan değişiklikler projenin karakteristik özelliğini barındırır. Proje gelişimi boyunca aynı sayıda değişime uğramış dosya grupları incelendiğinde bir grup dosya değiştiğinde diğer bir dosyanın da sürekli değişikliğe uğradığı gözlemlenmektedir. Bu durum rastgele bir tesadüfün ötesinde bir bağıllığı temsil etmektedir. Bu bağıllık sistemli bir şekilde açıklanırsa, projenin durumu, geleceği hakkında daha fazla yorum yapılmasına, önceden tespit edilemeyecek hataların ön görülmesine yardımcı olabilir.

Sosyal ağ analizi, karmaşık ağları anlama ve yorumlamada etkin bir araştırma yaklaşımıdır. Ağ analizi, birbirine doğrudan veya dolayı bağlı bir ilişkideki aktörleri inceler ve ilişkinin nedenlerini ve sonuçlarını anlamaya çalışır.

Sürekli değişen dosyaların versiyonlama araçları ile tüm hareketleri kaydedilir, bu sebeple versiyonlama araçları bir projenin gelişimi konusunda çok fazla bilgiye sahiptir. Bu çalışmada bir yazılım projesinde birlikte değişikliğe uğrayan dosyaların arasındaki ilişkilerin ağ dönüştürülerek karmaşık ağ analizi yöntemleri ile analiz edilmesi hedeflenmektedir. Yapılacak analiz sonucunda yazılım projelerinin karakteristik ortak özelliklerinin belirlenmesi hedeflenmektedir, ek olarak projenin gizli modüllerini ve birbirine bağımlı olan sistemlerinde anlaşılmasında yol gösterici olacaktır.



## BÖLÜM 2. İLGİLİ ÇALIŞMALAR

Yazılım projelerinde dosya değişim tahminleri kapsamında ilk çalışmalardan birisi Gall ve arkadaşları [4] bir kaynak kodu varlığındaki bir değişiklik diğer varlıklara nasıl yayılır sorusuna cevap aramıştır. Sürüm kontrol sistemlerindeki verileri kullanarak dosyalar arasındaki bağılılığı tespit edebilmişlerdir. Tarihsel değişim verilerinin değişim yayılımı sürecinde geliştiricilere büyük başarı ile yardımcı olmak için sezgisel tarama geliştirmek için kullanılabileceğini göstermişlerdir.

Ball T. Ve arkadaşları [5] versiyon kontrol sistemlerinin yazılım projelerinin karakteristik özelliği hakkında önemli bilgiler içerdiğini öne sürmüştür. Ball T. ve arkadaşlarına [5] göre hangi kodun değiştirildiğini, ne zaman değiştirildiğini, değişikliği kimin yaptığını gibi kod geliştirme sürecinde gerçekte olup bitenler hakkında, bazen geliştiricilerin anılarından daha iyi, değerli bilgiler verebileceğini savunmuştur. Versiyon kontrol sistemleri verileri tamamen analize uygun veriler olduğunu savunmuştur.

Zimmermann ve arkadaşları [6] , ROSE adında bir araç geliştirdi ve üç amaca ulaşmak için ECLIPSE ile bir eklenti olarak entegre etti:

- (i) Gelecekteki değişikliklerin tahmin edilmesi,
- (ii) Tespit edilmesi zor olan bileşen (dosya, yöntem, değişken vb.) kalıplarının belirlenmesi program analizi.
- (iii) Eksik değişiklikler nedeniyle hataların önlenmesi.

ROSE prototipi, vakaların %26'sı için belirli bir değişiklik isteği için hangi dosyaların değiştirilmesi gerektiğini tahmin edebilmiştir.

Hassan ve arkadaşları [7] birlikte değişen dosyaları araştırdılar ve beş büyük açık kaynaklı yazılım projesini analiz ederek ortak değişiklikleri öngörmek için farklı sezgisel tarama kullandılar. Sezgileri veri kaynaklarına ve budama tekniklerine göre tanımladılar. Varlık tabanlı geçmiş ortak değişim verilerini hem sıklık hem de yineleme yöntemlerini birleştiren karma budama yöntemiyle kullanırken en iyi tahmin performansını elde etmişlerdir. Hassas performansı iyileştirmek için Hassan ve ark. [7]

melez sezgisel tarama kullandılar bu yeni yaklaşımla hassasiyet %49'a çıkarken geri çağırma %51'e düştü. Daha iyi performans elde etmek için daha karmaşık bir sezgisel buluşmanın geliştirilmesi gerektiğini belirtilmiştir.

Sayyad-Shirabad ve arkadaşları [8] mantıksal olarak eşleştirilmiş dosyalar arasındaki farklı ilişki kavramlarını öğrenmek için tümevarımsal öğrenme kullandılar. Örneğin iki dosyanın aynı anda güncellenip güncellenmediği bir ilişki ilişkisidir, dosya adı, uzantı ve tanımlanan rutin sayısı gibi basit metrikleri kullanarak dosya değişim ilişkilerini açıklamışlardır.

Canfora ve arkadaşları [9] dosya değişimlerini tahmin etmek için kod değişikliklerinden ilişki kuralları çıkarmışlardır ve %30 başarı sağlamışlardır. Bu kadar düşük başarı olmasının nedeni, ilişkilendirme kurallarının ürettiği yüksek yanlış alarmlardan kaynaklanıyordu. Kurallar ortak değişiklikler bulmakta başarılıydı fakat yanlış tahmin oranı da çok yüksekti.

Macho ve arkadaşları [10] kaynak kod değişiklikleri ve tahmin kategorileriyle ilgili ayrıntılı bilgileri kullanarak, kod değişikliklerinin tahminini geliştirmek için bir çalışma yaptılar. Önceki çalışmalara göre çok daha yüksek performans elde etmişler ve sorun takip sistemini tahmin modellerine dahil ederek performansı daha da arttırmayı planlamışlardır.

Zhou ve arkadaşları [11] tarafından kod değişiklikleri arasındaki davranışını tahmin etmek için bir bayes ağı tabanlı bir yaklaşım sunulmuştur. Statik kaynak kod bağımlılığı, önceki ortak değişikliklerin sıklığı, değişiklik anlamlılık seviyesi, değişim yaşı ve ortak değişim varlıkları gibi bir dizi çıkarılan özelliğe dayanan yaklaşım yapmışlardır.

Giuliano Antoniol ve arkadaşları [12] CVS deposundan veri madenciliği yaparak, birlikte değişen dosya gruplarını bulmak ve tanımlamak için bir yaklaşım sunmuştur. DTW ve Jaccard benzerliğine dayanan, karışan faktörlere göre sağlam olan birlikte değişen dosya algılama grupları için bir yaklaşım geliştirmiştir. Versiyon kontrol sisteminde bugünden geçmişe bakarak birlikte değişen dosya gruplarını tespit etmeyi hedeflemiştir. Bu grupların nitel analizi, bu dosyaların değişimlerinin benzer davranışta olduğunu göstermektedir.

Yapılan literatür çalışmalarında dosya deęişim hareketlerinin sosyal aę analizi ile analiz edilmesiyle ilgili bir çalıřmaya rastlanmamıřtır. Dosya deęişim geiřmiřine ait hareketlerin sosyal aę analizi ile incelenmesi bu çalıřma ile ilk defa yapılacaktır.



## BÖLÜM 3. AĞ ANALİZİ

Bu bölümde ağ analizi ve kullanılacak yöntemler hakkında genel bilgilendirme yapılacaktır.

### Temel Graf Bilgisi

Türkçeye çizge veya çizit olarak da geçmiş olan düğümler ve düğümleri birbirine bağlayan kenarlardan oluşan yapıya graf denir. Graf, bir ağın ve bağlantısının sembolik bir temsidir.

Graf teorisi çizgilerle sağlanan bu gösterim modelini inceleyen, optimize etmeye çalışan, matematiğin bir dalıdır. Graf teorisinin temelinde, karmaşık bir sistemin bir dizi bağlantıyla birbirine bağlanmış bir dizi soyut bileşene indirgenebileceği anlayışı yatmaktadır. Bu fikir, ilk olarak 1735 yılında Prusya kasabasındaki Königsberg 'in Yedi Köprü'ünü çözmek için kullanan 18. yüzyıl matematikçisi Leonhard Euler'e dayanmaktadır. Şehrin yedi köprüsünden en az birini iki kez geçmeden şehrin tüm adalarını ziyaret edemeyeceğini göstermek için çalışan Euler, problemi tüm karmaşıklıklarında soyutlayarak matematiksel temsille adaların düğümlerle ve köprülerin kenarlarla temsil edildiği şehrin soyut bir haritasını oluşturarak açıkladı. Bu yöntem kullanıldıktan 200 yıl sonra, Paul Erdos ve Alfred Renyi'nin çalışmaları ile 20. yüzyılda ayrıntılı, resmileştirilmiş bir matematiksel teori olarak yeniden ortaya çıkmıştır.

Graf yapıları, fen bilimlerinde, bilgisayar bilimlerinde, sosyal bilimlerde kullanım alanları oldukça fazladır. Elektronik devre şemalarında, taşımacılık ağlarında, navigasyon sistemlerinde, mobil şebeke sistemlerinde, bilgisayar bilimlerinde veri yapılarında, sosyal ağ paylaşım sistemlerinde, bilgisayar yazılım projesi geliştirme araçlarında üretilen sınıf ve modellerin birbirine bağlılıklarında, sosyal alanlarda insan ilişkilerinin yorumlanmasında yaygın olarak kullanılmaktadır. 1880/1881 dönemine ait Alman erkek okulunda Johannes Delitsch tarafından toplanan ilk sosyal ağ veri setine ait Class of 1880/81 [13] grafi Şekil-1'deki gibidir.



Şekil 1: Class of 1880/81 ağı görünümü

### 3.1.1. Graf

Durum veya varlıklar arasındaki bağıntıyı görselleştiren bir matematik modeldir. Graf düğüm ve bu düğümleri birbirine bağlayan kenarlardan oluşmaktadır.

Grafın matematik gösteriminde, düğüm (vertex, vertice, node)  $V$  ve kenar (edge)  $E$  harfleri ile ifade edilir. Grafın kendisi de  $G$  notasyonu ile gösterilir. Buna göre grafın gösterimi denklem (1)'de verilmiştir;

$$G = (V, E) \quad (1)$$

Açık bir ifadeyle; graf, düğüm ve kenarların kartezyen kombinasyonudur. Kenarlar ise bağlı oldukları düğüm çiftleri ile ifade edilirler. Bir kenarı anlamlı kılan, bağlı bulunduğu düğümlerdir. Bir düğümü ise anlamlı kılan, hangi diğer düğümlerle bağlı olduğudur. Bir  $G$  çizgesinin kenarları  $E(G)$  ile düğümleri ise  $V(G)$  ile gösterilir.

### 3.1.2. Düğüm (Node)

Graftaki en temel nesne düğümdür. Bir graf oluşturabilmek için en az bir düğüm gereklidir.  $n$  adet düğümden oluşan bir grafa ait düğümlerin kümesi denklem (2)'de verilmiştir.

$$V = \{v_1, v_2, \dots, v_n\} \quad (2)$$

Burada  $n$  grafın boyutudur ve  $|V| = n$  şeklinde gösterilir

### 3.1.3. Kenar (Edge)

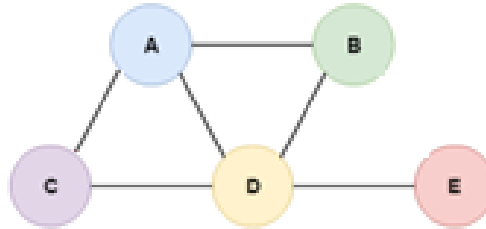
Grafta düğümleri birbirine bağlayan diğer temel nesneye kenar adı verilir. Bir grafa kenar bulunması zorunlu değildir, fakat bir ağın anlamlı olması için birden fazla düğümler arasında bir kenar bulunması beklenir. Düğümleri birbirine bağlayan  $m$  adet kenar içeren kenar kümesi denklem (3)'te verilmiştir.

$$E = \{e_1, e_2, \dots, e_m\} \quad (3)$$

Kenarlar yönlü (directed) veya yönsüz (undirected) olabilirler. Yönlü kenarlar yay (arc) olarak adlandırılır. Yönsüz kenarlar düz çizgi çizilerek ifade edilirken, yönlü kenarlar yönü gösterecek şekilde oklu çizgi çizilerek ifade edilir.

Graflar kenarlarının yönlü veya yönsüz olmasına bağlı olarak yönlü graf ya da yönsüz graf olarak adlandırılır.

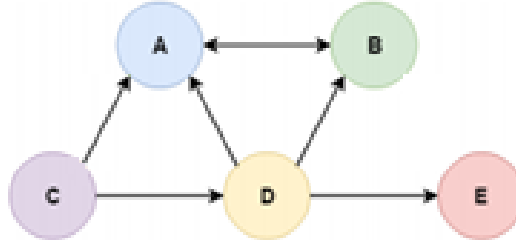
Yönsüz grafa örnek olarak sosyal arkadaşlık sistemleri (Örn: Facebook) gösterilebilir. Böyle bir grafa Şekil 2'deki gibi A kişisi B kişisi ile arkadaş ise B'nin A ile arkadaş olmaması düşünülemez.



Şekil 2: Yönsüz graf

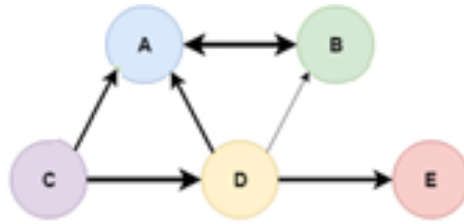
Yönlü grafa örnek olarak Twitter gösterilebilir. Şekil 3'teki A kişisi B kişisini takip edebilir fakat B kişisi A kişisini takip etmek zorunda değildir. B kişisi A'yı takip

ediyor ise B'den A'ya doğru ikinci bir yönlü kenar eklenir ya da iki ucunda ok bulunan tek bir çizgi çizilir.



**Şekil 3:** Yönlü graf

Kenarlar ile alakalı diğer bir öznelik de kenarların ağırlıklı (weighted) olup olmaması durumudur. Şekil 4'te kenarların kalınlığından ağırlıklı graf olduğu anlaşılmaktadır. Kenarın ağırlığı, kenarın birleştirdiği iki düğüm arasında bulunan ilişkiye ait sayısal bir değerdir. Bir karayolu haritasında bulunan şehirleri ve bu şehirler arasındaki yolları temsil eden bir grafa, şehirler arasındaki mesafeyi anlatmak için ağırlık kullanılabilir. Matematiksel notasyonla, ağırlıklı bir graf  $G(V, E, W)$  şeklinde gösterilir.  $W$  Simgesi,  $E$ 'deki her bir kenara ait ağırlık değerinin kümesidir ve daima  $|W| = |E|$  eşitliği vardır [14].



**Şekil 4:** Yönlü ağırlıklı graf

### Graf Gösterim Şekilleri

Düğümün çizgilerle birbirine bağlanarak gösterimi graf gösterim yöntemlerinde en anlaşılır olanıdır. Bu gösterim şekli dışında bilgisayar programları tarafından kolayca işlenebilmesi ve analiz yöntemlerinin kolayca uygulanabilmesi için farklı gösterim şekilleri geliştirilmiştir.

Gösterim şekillerinde graf ile ilgi herhangi bir bilginin kaybolmaması ve herhangi bir özel araç kullanılmadan insanlar tarafından oluşturulabilmesi ve anlaşılır olması gerekmektedir.

#### 3.1.4. Komşuluk Matrisi

Grafları temsil etmenin en basit yollarından biri komşuluk matrisidir (adjacency matrix). Komşuluk matrisi aynı zamanda sosyomatrix (sociomatrix) olarak da adlandırılır. Komşuluk matrisinin hücrelerinde 1 değeri, o hücreye karşılık düşen satır ve sütundaki düğümler arasında bir kenar olduğunu, 0 değeri ise kenar olmadığını gösterir. Ağırlıklı graflarda 1 değeri yerine kenarın ağırlık değeri kullanılır. Komşuluk matrisi matematiksel olarak en doğal gösterim şeklidir [14]. Şekil 2'deki graf'a ait komşuluk matrisi Tablo 1'de verilmiştir.

Tablo incelendiğinde matrisin köşegen etrafında simetrik olduğu gözlemlenir. Bu durum sadece yönsüz ağlar için geçerlidir.

**Tablo 1:** Komşuluk matrisi

	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	1	0
C	1	0	0	1	0
D	1	1	1	0	1
E	0	0	0	1	0

#### 3.1.5. Komşuluk Listesi

Komşuluk matrisinde bağlantılı olmayan düğümlerin sıfır ile gösterimi matrisin okunurluğunu düşürmekte ve gereksiz bellek kullanımına sebep olmaktadır. Ağın yapısının basit temsil edilebilmesi için her bir satırda bir düğüm, karşılığında ilişkili olduğu düğümleri içeren bir liste oluşturulmaktadır, bu gösterim şekline komşuluk listesi denilmektedir. Şekil 2'deki grafa ait komşuluk listesi Tablo 2'de verilmiştir.



**Tablo 2:** Komşuluk listesi

Düğüm	Komşuları
A	B, C, D
B	A, D
C	A, D
D	A, B, C, E
E	D

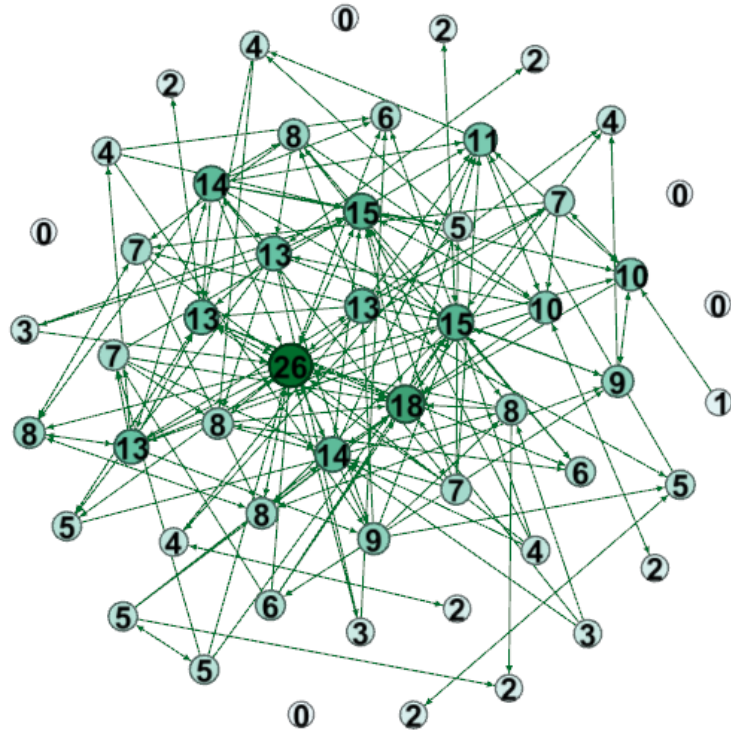
### Ağ Ölçütleri

Bu bölümde ağ analizinde kullanılan ölçütler açıklanmıştır.

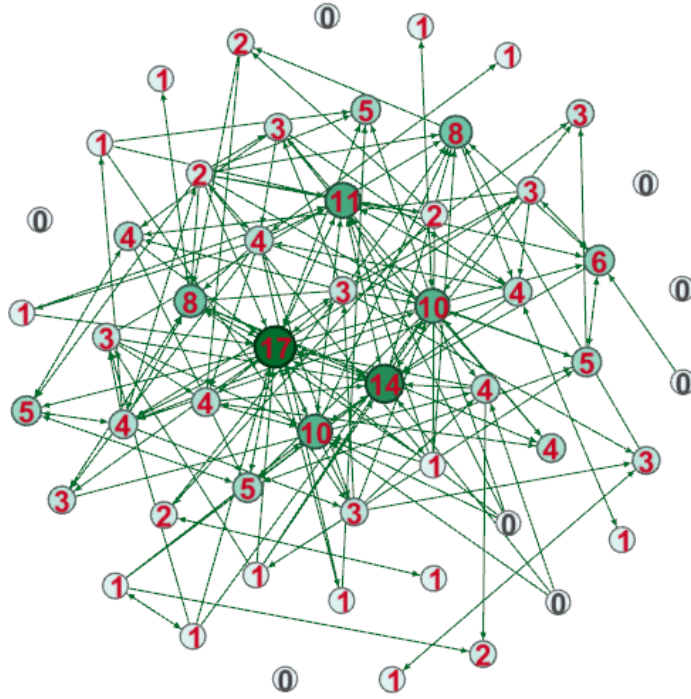
#### 3.1.6. Derece Merkeziliği

Bir düğüm ile ilgili temel ölçüt derecedir (degree). Yönsüz ağlarda bir düğümün derecesi o düğüme bağlı kenarların sayısıdır. Yönlü ağlarda ise iç-derece (in-degree) ve dış derece (out-degree) şeklinde iki derece vardır. Düğüme gelen kenarların sayısı iç derece, düğümünden çıkan kenarların sayısı ise dış-derecedir. Yönlü ağlarda iç ve dış derecelerin toplamı düğümün toplam derecesini verir.

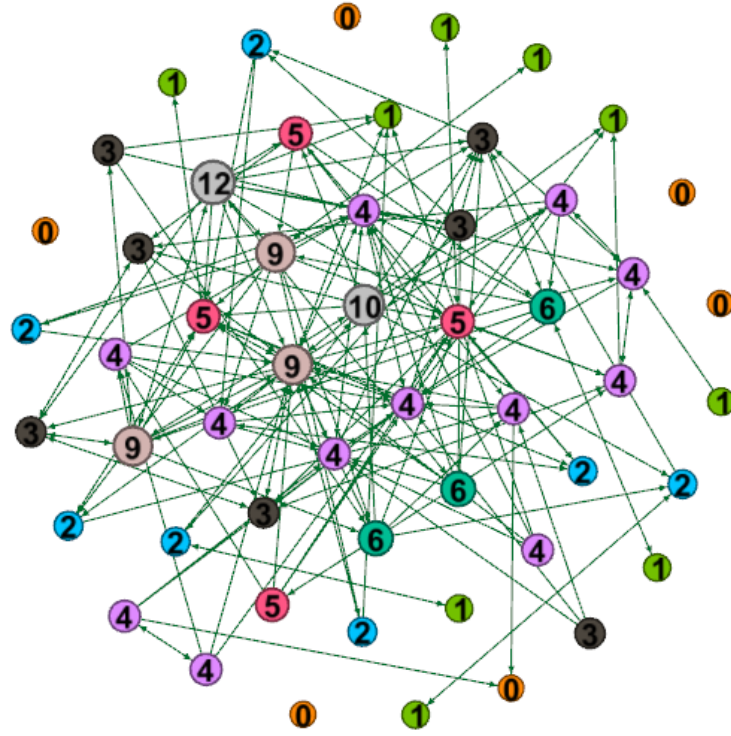
Düğümlerin yalnızca derecelerini kullanarak ağdaki önemini belirlemeye yarayan, hesaplama metodu olarak en basit merkezilik ölçütü derece merkeziliğidir. Şekil 5’ de Class of 1880/81’a ait derece merkeziliği değerleri ağ düğümlerinde yazan etiketler ile gösterilmektedir, gösterimde derecesi büyük olan düğüm daha koyu renkte gösterilmektedir. Sosyal ağlarda bu ölçüt gerçek hayatta bağlantısı en çok olan aktörün en önemli olduğu ilkesine dayanmaktadır. Yani bir düğümün derecesi ne kadar yüksek ise o düğüm o kadar önemlidir. Yönlü ağlarda ise iç-derece merkeziliği bir bireyin sosyal ağdaki popülerliğini, dış derece merkeziliği ise bireyin sokuşlanlık ve sosyalleşebilirliğini (gregariousness, sociability) temsil eder [14].



Şekil 5: Derece merkezliği



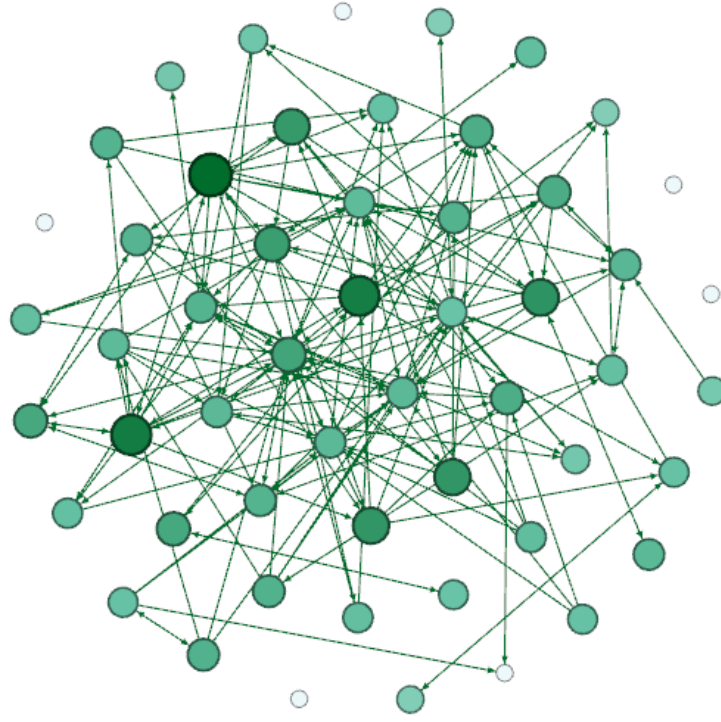
Şekil 6: İç derece merkezliği



Şekil 7: Dış derece merkeziliği

### 3.1.7. Yakınlık Merkeziliği

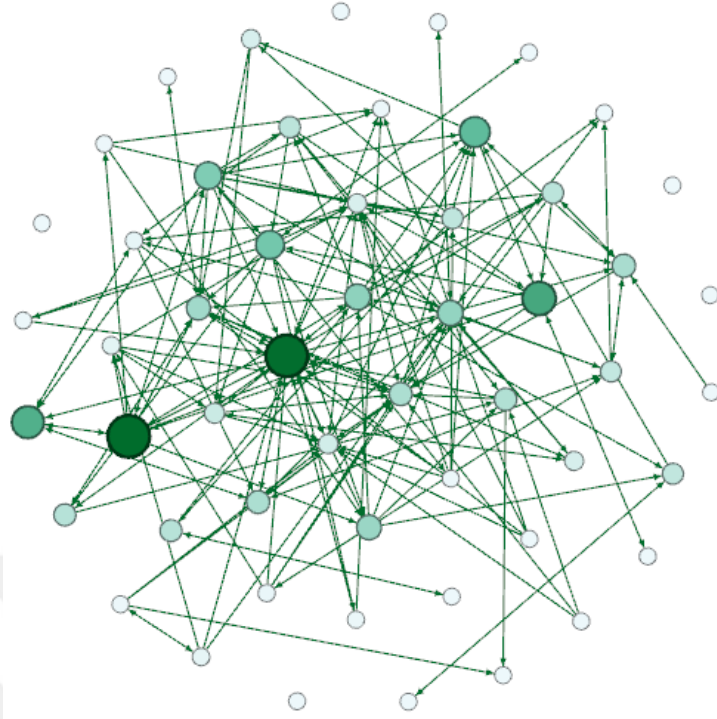
Yakınlık merkeziliği, bir düğümün ağdaki tüm diğer düğümlerle ne kadar yakın olduğunun gösterebilir. Bu ölçüte göre bir düğüm ne kadar merkezi ise o düğüm diğer düğümlere o kadar çabuk ulaşılabilir. Bir düğümün yakınlık merkeziliği hesaplanırken, o düğümün ağdaki diğer tüm düğümlere olan en kısa yol uzunluklarının ortalaması hesaplanır ve ters alınır. Class of 1880/81'a ait yakınlık merkeziliği görünümü Şekil 8 ile gösterilmektedir. Gösterimde derecesi büyük olan düğüm daha koyu renkte gösterilmektedir.



**Şekil 8:** Yakınlık Merkeziliği

### 3.1.8. Arasındalık Merkeziliği

Sıklıkla kullanılan merkezilik ölçütlerinden biri olan arasındalık merkeziliği, bir düğümün ağdaki bilgi akışında ne kadar etkili bir konumda olduğunu ifade eder. Bir düğümün diğer düğümlere yakınlığı ile değil, diğer düğümler arasındaki en kısa yol üzerinde yer almasıyla ilgilenmektedir [14]. Örneğin, bir  $(x,y)$  düğüm çifti arasında 5 adet en kısa yol olsun,  $v$  düğüm bu yollardan 4'ünün üzerinde olsun. Bu durumda  $(x,y)$  düğüm çifti için  $4 / 5 = 0,8$  değeri elde edilir. Bu işlem tüm düğüm çiftleri için hesaplandıktan sonra hesaplanan oranlar toplamı  $v$  düğümünün arasındalık merkeziliği olarak bulunur. Class of 1880/81 ağına ait arasındalık merkezilik görünümü Şekil 9 ile gösterilmektedir. Gösterimde derecesi büyük olan düğüm daha koyu renkte gösterilmektedir.

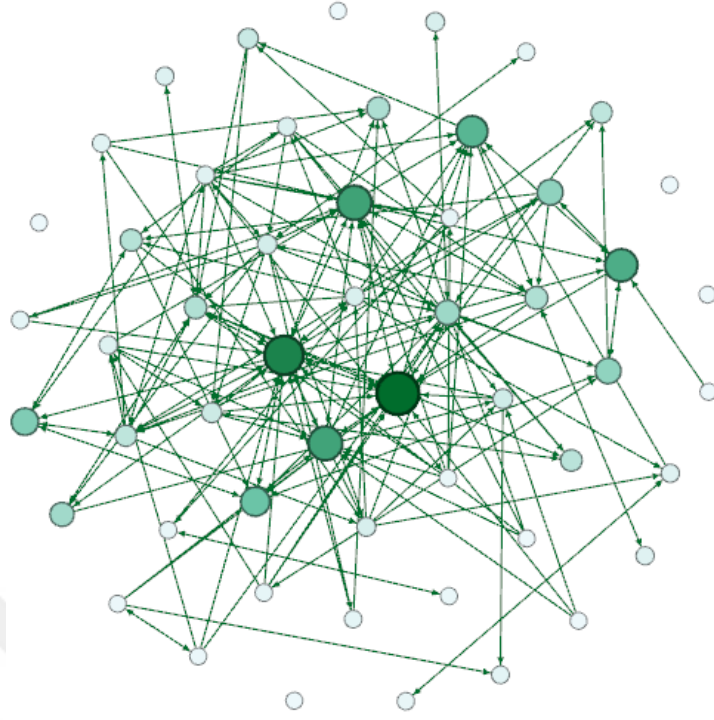


**Şekil 9:** Arasında merkezilik gösterimi

### 3.1.9. Özvektör Merkeziliği

Özvektör merkeziliğinde bir düğümün önemi sadece komşularının sayısına göre değil aynı zamanda komşularının da önemine bağlıdır. Bu yaklaşım gerçek hayat ağlarındaki yüksek yerlerde çok sayıda arkadaş edinme özelliği ile ilgilidir [14].

Özvektör merkeziliği için komşuluk matrisi üzerinde özdeğer (eigenvalue) ve özvektör (eigenvector) hesaplamaları yapılır. Özvektör merkeziliği geribildirim tabanlı (feedback based) merkezilik ölçütleri kategorisindedir. Özvektör merkeziğili ile benzerlik gösteren diğer geri bildirim tabanlı ölçütler Katz Merkeziliği (Katz Centrality), PageRank Merkeziliği, Otorite ve Göbek Merkeziliğidir [14]. Class of 1880/81'a ait özvektör merkeziliği görünümü Şekil 10'da gösterilmektedir, gösterimde derecesi büyük olan düğüm daha koyu renkte gösterilmektedir.



**Şekil 10:** Özvektör merkeziliği

### 3.1.10. Kümelenme Katsayısı

Geçişlilik, bağ oluşturma analizi için kullanılan ölçütlerden birisidir. A, B, C bir ağda herhangi üç düğüm olsun, A-B ve A-C kenarları ağda mevcutken bir B-C kenarı da varsa bu üç düğüm arasında bir geçişlilik davranışı söz konusudur. Bu davranış “arkadaşımın arkadaşı benim de arkadaşımdır” anlamındadır [14].

Kümeleme katsayısı, bir grafın tam grafa ne kadar yakın yoğunlukta olduğunun bir ölçüsüdür. Düğümler için hesaplandığında yerel kümeleme katsayısı, bütün ağ için hesaplandığında global kümelenme katsayısı adı verilir. Kümelenme katsayısı [0,1] aralığında değer alır.

Yerel kümelenme katsayısı denklem (4)'te verilmiştir.

$$C_{cc}(v) = \frac{v'yi \text{ içeren üçgenlerin sayısı}}{v'yi \text{ merkez alan bağlı üçlülerin sayısı}} \quad (4)$$

Global kümelenme katsayısı Denklem (5)'te verilmiştir

$$C_{cc}(v) = \frac{ağdaki \text{ üçgenlerin sayısı} \times 3}{ağdaki \text{ bağlı üçlülerin sayısı}} \quad (5)$$

### 3.1.11. Ortalama derece

Ortalama derece, ağın genel yapısını derece cinsinden tanımlayan, ağda bulunan düğümlerin ortalama derecesinin hesaplanmasıyla bulunan bir ölçüttür. Çoğunlukla  $\langle k \rangle$  olarak ifade edilir. Ağın yönlü ve yönsüz olmasına göre iki farklı hesaplama yolu vardır.  $k_i$ ,  $i$  düğümünün derecesi olmak üzere,  $n$  düğüm ve  $m$  kenara sahip yönsüz ağın ortalama derecesi denklem (6)'da verilmiştir [14].

$$\langle k \rangle = \frac{1}{n} \sum_{i=0}^n k_i = \frac{2m}{n} \quad (6)$$

Yönlü ağlarda ise iç-derece (in-degree) ve dış derece (out-degree) olmak üzere iki derece hesaplanır. Dolayısıyla  $n$  düğümlü yönlü bir ağda ortalama iç ve dış-dereceler denklem (7) ve (8)'deki gibidir [14].

$$\langle k^{in} \rangle = \frac{1}{n} \sum_i^n k_i^{in} \quad (7)$$

$$\langle k^{out} \rangle = \frac{1}{n} \sum_i^n k_i^{out} \quad (8)$$

Yönlü ağlarda ortalama iç-derece ve dış-derece birbirine eşittir ve bu eşitlik denklem (9)'da verilmiştir.

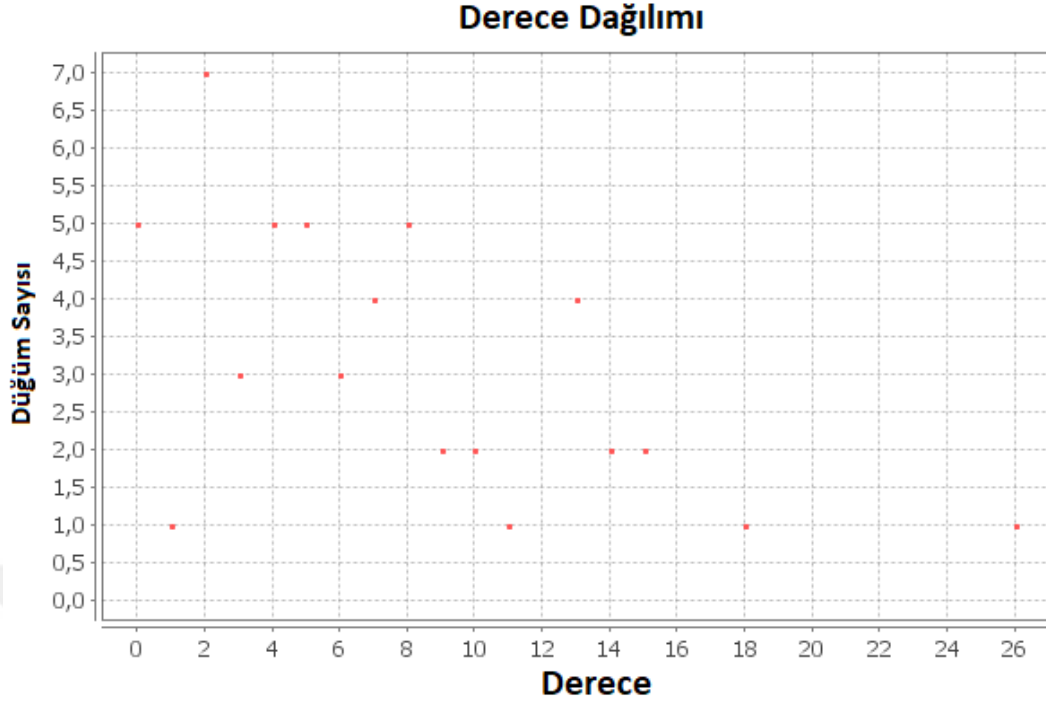
$$\langle k^{in} \rangle = \langle k^{out} \rangle = \frac{m}{n} \quad (9)$$

### 3.1.12. Derece Dağılımı

Derece dağılımı özellikle büyük ölçekli ağlarda ağın karakteristik özelliğini çıkarmak ve ağı tanımlamak için ağ içerisinde hangi dereceye sahip kaç adet düğüm olduğunu ifade eden çok önemli bir ölçüttür.

Derece dağılımını bulmak için ağdaki herhangi bir düğümün derecesi bulunur ve ağda bulunan bu derece ile eşit seviyede kaç adet düğüm olduğu hesaplanarak dağılım hesaplanır. Class of 1880/81'a ait derece dağılımı Şekil 11'de gösterilmektedir.





**Şekil 11:** Class of 1880/81 ağının derece dağılımı

### Topluluk Analizi

Sosyal ağlarda toplulukların analiz edilmesiyle çok önemli bilgilerin elde edilmesi mümkündür. Sosyal ağlarda çoğunlukla iki tür topluluk oluşumu gözlenmektedir. Belirgin topluluklar (explicit community) ve üstü kapalı topluluklar (implicit community). Topluluk tespiti için çeşitli yöntemler geliştirilmiştir.

**Kenar merkeziliği tabanlı topluluk tespiti**, bu yöntemdeki temel prensibi topluluklar arası iletişim için kritik durumda olan kenarların bulunmasıdır.

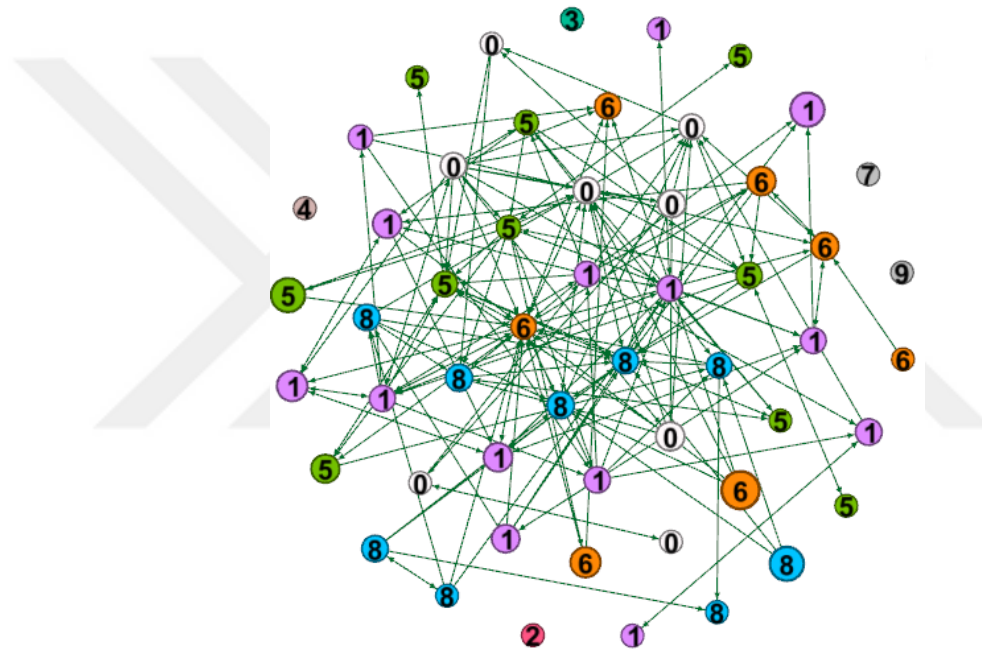
**Benzerlik tabanlı topluluk tespiti**, bu yöntemde ağ yapısı üzerinde hesaplanabilen bir benzerlik ölçütü tanımlanır. Kosinus benzerliği ve jaccard benzerliği veri madenciliği alanında yaygın şekilde kullanılan iki önemli benzerlik ölçütüdür.

**Modülerite**, Newman ve Girman tarafından ortaya atılan, topluluk analizinde çok kullanılan ve oldukça etkili bir kavramdır. Şekil 12’de Class of 1880/81 ağının modülerite yöntemi ile topluluk analizi ağı gözlemlenmektedir. Modülerite kavramına göre bir ağdaki toplulukların oluşumu kesinlikle rastgele bir sürecin sonucu değildir.



Kısaca modülarite, gerçek hayat ağlarında bir topluluktaki olası kenar yoğunluğu ile rastgele olarak oluşması gereken kenar yoğunluğu arasındaki farktır [14].

Modülarite tabanlı yöntemler, çok büyük ağlarda bulunan küçük toplulukları genellikle bunların birleşiminden oluşan büyükçe topluluklar şeklinde tespit eder. Bu duruma çözünürlük problemi (problem of resolution) ya da çözünürlük limiti (resolution limit) adları verilir. Modülarite yöntemli topluluk analizleri bu problemin farkında olunarak uygulanmalıdır. Çeşitli algoritmalarda bu problemin üstesinden gelebilmek için toplulukların boyutunu belirleyen çözünürlük parametresi değeri vardır [14].



Şekil 12: Modülarite ağı gösterimi

**Örtüşen Mahalle Genişlemesi ile topluluk tespiti (ClusterONE)**, bu yöntem temel olarak üç adımdan oluşur. Başlangıçta en yüksek dereceye sahip düğüm bir topluluk olarak kabul edilir, açgözlü (greedy) arama ile onunla bağlantılı düğümleri ekleyerek büyür. ClusterOne büyütülmüş bu grubu hiç düğüm kalmayana kadar besler. Daha sonra Cluster One en uygun üstü üste binen grupları bir üst grupta birleştirir. Son

olarak ClusterOne üçten az düğüm içermeyen, yoğunluğu eşik değerinden (varsayılan değer 0,8'dir.) daha büyük olan grupları üretir [15].

### 3.1.13. Gerçek Hayat Ağları

Gerçek hayat ağları (Real World Network) benzer bazı karakteristik özelliklere sahiptir. Gerçek hayat ağlarının 3 temel özelliği şunlardır:

1. Derece dağılımı kuvvet yasası dağılımı sergilemektedir.
2. Ortalama yol uzunluğu kısadır.
3. Kümeleme katsayısı yüksektir.

### 3.1.14. Derece Dağılımı

Genellikle çok büyük gerçek hayat ağlarında derece dağılımı karakteristik olarak derece büyük olan çok düşük, derecesi düşük olan çok fazla düğüm görülmektedir. Bu dağılıma Kuvvet Yasası Dağılımı denilmektedir [14]. Şekil 13'te Kuvvet yasası dağılımı grafiği görülmektedir.



Şekil 13. Kuvvet yasası dağılımı

### 3.1.15. Ortalama Yol Uzunluğu

Gerçek hayat ağlarında ortak özellikleri herhangi bir düğüm çiftinin arasındaki yol uzunluğu kısa olmasıdır [14].

### 3.1.16. Kümeleme Katsayısı

Gerçek hayat ağları, kümeleme etkisini çok yüksek göstermektedir. Ağdaki iki düğüm ortak bir komşuya sahipse ikisi arasında çok yüksek olasılıkla bir bağ vardır.



## BÖLÜM 4. YÖNTEM

Bu bölümde çalışmada kullanılan araçlar ve kullanım yöntemleri açıklanmıştır.

### Araçlar

Bu çalışmada kod deposu olarak çeşitli kod deposu sistemleri incelenmiş ve en yaygın kullanım ağına sahip Github platformu tercih edilmiştir, ağ analizi için Gephi ve Cytoscape uygulamaları kullanılmıştır.

#### 4.1.1. Github

Git, Linus Torvalds tarafından geliştirilmiş bir versiyon kontrol sistemidir. GitHub, git yazılımı ile entegre olmuş bir depolama alanıdır. 31 milyon kişi tarafından kullanılan sistemde 100 milyonun üzerinde proje bulunmaktadır [16]. Github üzerinden hesap açıldıktan sonra herkese açık olan herhangi bir projeye erişerek, proje hakkında ayrıntılı bilgi edinilebilir. Proje geliştiricileri, yapılan değişiklik adetleri, proje içerisindeki dosyalar, bu dosyalarda yapılan değişiklik tarihçesi vb. bilgilere erişilebilir.

#### 4.1.2. Gephi

Gephi, grafik ve ağ analizi için açık kaynaklı bir yazılımdır. Büyük ağları gerçek zamanlı olarak görüntülemek ve keşfi hızlandırmak için 3D render motoru kullanır. Karmaşık veri kümeleriyle çalışıldığında önemli görsel sonuçlar üretebilmektedir. Ağ verilerine kolay ve geniş erişim sağlar ve filtreleme, gezinme, manipüle etme ve kümeleme gibi özellikleri vardır. [17]

#### 4.1.3. Cytoscape

Cytoscape, moleküler etkileşim ağlarını ve biyolojik yolları görselleştirmek ve bu ağları ek açıklamalar, gen ekspresyon profilleri ve diğer durum verileri ile entegre etmek için açık kaynaklı bir yazılım platformudur [18]. Cytoscape başlangıçta biyolojik araştırma için tasarlanmış olmasına rağmen, karmaşık ağ analizi ve görselleştirme için genel bir platformdur.

## Yöntem

Bu araştırma üç farklı aşamadan oluşur: Veri toplama, ağ modelleme ve karmaşık ağ analizi. Her fazın detayları bu bölümde kendi alt bölümlerinde verilmiştir.

### 4.1.4. Veri Toplama

Bu çalışmada, oluşturulacak ağın yapısının seçilen projelerden en az seviyede etkilenmesi için, seçilen projelerin farklı yazılım geliştirme dilleri ile farklı sürelerde, farklı boyutlarda, birbirine benzemeyen konularda geliştirilmiş projeler olmasına özen gösterilmiştir. Seçilen projelerin bir kısmı tamamen tamamlanmış, bir kısmı bu çalışma hazırlanırken geliştirilmeye devam edilmektedir. Seçilen projelere ait bilgiler Tablo 3'te, proje konuları Tablo 4'te gösterilmiştir.

**Tablo 3.** Seçilen proje bilgileri

Proje Adı	İşlem (Commit) Adedi	Dosya Adedi	Geliştirici Sayısı	Proje Başlangıç Tarihi
angular-cli	4.107	3.347	429	22.03.2015
aspnetboilerplate	5.510	8.504	123	18.08.2013
aws-cli	6.872	6.482	193	11.11.2012
bootstrap-material-design	1.352	1.407	84	14.09.2014
glide	2.303	2.699	96	3.11.2013
http-server	1.733	1.404	30	27.01.2013
johnny-five	3.223	2.004	124	25.03.2012
matlab2tikz	2.767	163	27	2.11.2008
mist	2.049	767	64	24.01.2015
MPAndroidChart	2.018	952	66	20.04.2014
prettier	2.965	5.263	407	27.11.2016

**Tablo 4.** Proje konusu

Proje Adı	Proje Konusu
angular-cli	CLI tool for Angular
aspnetboilerplate	ASP.NET Boilerplate - Web Application Framework
aws-cli	Universal Command Line Interface for Amazon Web Services
bootstrap-material-design	Material design theme for Bootstrap 3 and 4
glide	An image loading and caching library for Android focused on smooth scrolling
http-server	A non-blocking HTTP application server for PHP based on Amp.
johnny-five	JavaScript Robotics and IoT programming framework, developed at Bocoup.
matlab2tikz	This program converts MATLAB®/Octave figures to TikZ/pgfplots figures for smooth integration into LaTeX.
mist	Browse and use Dapps on the Ethereum network.
MPAndroidChart	A powerful Android chart view / graph view library, supporting line- bar- pie- radar- bubble- and candlestick charts as well as scaling, dragging and animations.
prettier	Prettier is an opinionated code formatter.

Veri toplama işleminin özü çalışılan projeye ait dosyaların tespit edilerek her dosyanın check-in(işlem) geçmişine erişmek, tüm check-in işlemlerinde birlikte check-in yapılmış dosyaları veri tabanına ilişkili bir şekilde kaydının atılmasıdır.

Veri seçimi sırasında benzer dosya isimlerinin karışıklığa sebebiyet vermemesi için bütün dosyalar, uzantıları ile değerlendirilmiştir. Bu sayede modele alınan dosyaların tekil kalması sağlanmıştır.

Veri toplamak için bilinen iki yöntem incelenmiştir. Birisi Github'ın destek verdiği GraphQL platformu, diğeri Github'ın hazırladığı Github API dir. Bu çalışma kapsamında yapılan incelemeler sonrasında Github API kullanılarak C#.net platformunda bir uygulama hazırlanmasına karar verilmiştir.

#### **4.1.5. GraphQL**

GraphQL, veri tedariki için client server iletişimde daha fazla esneklik getirecek bir yaklaşım olarak facebook tarafından tasarlanmıştır. Yeni client server haberleşme standartlarındandır. Github'ın da GraphQL desteği bulunmaktadır. Github böyle bir desteği bazı kısıtlarla sunmaktadır. Bir sorgudan en fazla 100 kayıtlık bir sonuç dönmesi beklenmektedir. Bu çalışmada incelediğimiz projelerde bazı işlem (commit)lerde 100'den fazla dosya görünmektedir. Bu kısıtlardan dolayı bu yöntem kullanılamamıştır.

#### **4.1.6. Github API**

Github platformunun yayınladığı uygulama programlama ara yüzü (API) aracılığıyla kod depolarındaki projelere erişim sağlanabilmektedir. Bu çalışma kapsamında geliştirilen uygulama ile github üzerinden seçilen projelere ait tüm check-in (dosya değişim kayıtları) bilgileri bir veri tabanına aktarılmıştır.

Geliştirilen uygulama, Github tarafından .NET platformu için hazırlanmış Octokit kütüphanesi kullanılarak geliştirilmiştir.

#### **4.1.7. Octokit**

GitHub ile bağlantı kuran uygulamalar geliştirmek için kullanılan bir kütüphanedir. GitHub, kullanımı kolay Octokit kütüphanesinin (Ruby, .NET ve Node.js için) GitHub REST API v3 ile çalışan üç resmi çeşidi sunmaktadır. Octokit kütüphanesi kullanılarak C#.NET ortamında kolaylıkla github servislerine erişilebilmektedir.

Github tarafından tekil olarak üretilmiş token, kullanıcı adı ve şifresi ile servise erişim sağlanabilir, bağlı olunan projeye ait bilgilere erişilebilir.

Bu çalışma kapsamında model oluşturmak için 2 farklı yöntem izlenmiştir.

#### **4.1.8. Yöntem 1: Mevcut dosyalar üzerinden veri toplama**

Bu çalışma kapsamında ilk bu yöntem denenmiştir. Kullanılan yöntem, mevcut bir dosyanın check-in geçmişi kontrol edildikten sonra tüm check-in işlemlerine ait dosyalar ile kontrol edilen dosyalar arasında ilişki kurulmasından oluşmaktadır.

Bu yöntemle oluşturulmuş veriler ilişkisel veri tabanına aktarılmıştır. Hazırlanan veriler Tablo 5’te gösterilmiştir. Elde edilen kayıtlar incelendiğinde dosyalar arasındaki karşılıklı ilişki adetlerinde tutarsızlık görülmüştür. Tutarsızlığın sebebinin işlem anında mevcut olan dosyaların bilgilerinin elde edildiği fakat işlem anından önce silinmiş dosyaların check-in bilgilerinin elde edilemediği fark edilmiştir. Bu durumda kurulan ağ yönsüz ağ olmadığı için bu yöntem ile veri toplamadan vazgeçilmiştir.

**Tablo 5.** Yöntem 1 işlem sonucu

Referans Dosya	Birlikte Check-in olan Dosya	Proje Adı
README.md	MPChartLib/src/main/java/com/github/mikephil/charting/formatter/PercentFormatter.java	MPAndroidChart
build.gradle	gradle/wrapper/gradle-wrapper.properties	MPAndroidChart
gradle/wrapper/gradle-wrapper.properties	build.gradle	MPAndroidChart
.gitignore	MPChartExample/src/main/res/layout/list_item.xml	MPAndroidChart

#### 4.1.9. Yöntem 2: Tüm Check-in kayıtları ile veri toplama

Yöntem 1 başarısız olduktan sonra bu yöntem denenmiştir. Github, üzerinden ilgili projeye ait tüm commit bilgilerini içeren bir URL ile paylaşmaktadır. Elde edilen URL json formatında bilgiler içermektedir (örn: check-in sahibi, açıklama, commit edilen dosyalar vb.). Örnek check-in bilgileri json formatında Ek 1 ‘de paylaşılmıştır. Files tagi altındaki dosyalar ayıklanarak check-inlerde bulunan dosyalar için check-in – dosya ilişkisi veri tabanına aktarılmıştır. Elde edilen kayıtların örneği Tablo 6’da gösterilmiştir.

**Tablo 6.** Yöntem 2 işlem sonucu

Commit Id	Dosya Adı	Proje Adı
1	MPChartLib/src/main/java/com/github/mikephil/charting/formatter/PercentFormatter.java	MPAndroidChart
1	README.md	MPAndroidChart
2	build.gradle	MPAndroidChart
2	gradle/wrapper/gradle-wrapper.properties	MPAndroidChart



3	MPChartLib/src/main/java/com/github/mikephil/charting/formatter/PercentFormatter.java	MPAndroidChart
4	MPChartLib/src/main/java/com/github/mikephil/charting/formatter/PercentFormatter.java	MPAndroidChart
5	MPChartLib/src/main/java/com/github/mikephil/charting/formatter/PercentFormatter.java	MPAndroidChart
6	README.md	MPAndroidChart
7	README.md	MPAndroidChart
8	README.md	MPAndroidChart
9	README.md	MPAndroidChart
10	MPChartExample/build.gradle	MPAndroidChart

Tablo 6’da ki gibi ilişki kurulduktan sonra commit id ler referans alınarak, aynı commit id’lere sahip dosyalar arasında çift yönlü ilişki olacak şekilde bir model dosyası oluşturulmuştur. Oluşturulan model dosyası CSV formatında kaydedilmiştir. Ağ analizi uygulamalarına CSV formatında dosyalar aracılığıyla ağ modeli yüklenebilmektedir. Oluşturan ağa ait model örneği Tablo 7’de gösterilmiştir.

**Tablo 7.** Yöntem 2 model deseni

Kaynak Dosya	Birlikte Check-in olan dosya
README.md	gradle/wrapper/gradle-wrapper.properties
MPChartLib/src/main/java/com/github/mikephil/charting/formatter/PercentFormatter.java	MPChartExample/build.gradle
gradle/wrapper/gradle-wrapper.properties	README.md
MPChartLib/src/main/java/com/github/mikephil/charting/formatter/PercentFormatter.java	README.md
README.md	MPChartLib/src/main/java/com/github/mikephil/charting/formatter/PercentFormatter.java
build.gradle	README.md
README.md	build.gradle
README.md	MPChartExample/build.gradle
gradle/wrapper/gradle-wrapper.properties	MPChartExample/build.gradle
build.gradle	gradle/wrapper/gradle-wrapper.properties
gradle/wrapper/gradle-wrapper.properties	build.gradle
build.gradle	MPChartExample/build.gradle

Github api kullanımında karşılaşılan bir diğer sorun ise Github api'nin her bağlanıldığında en fazla 5000 adet sorguya cevap dönebilmesidir. Bu sınırı aştıktan sonra servis 1 saat kullanıma kapatılmaktadır.

Bu sorunun aşılması için api'ye bağlanıldığında her işlemten sonra en son kaydedilen dosya, işlem(commit) id'si ve proje numarası kaydedilerek uygulamanın kaldığı yerin tutulması sağlanmıştır. 5000 sorgu sınırı aşıldıktan sonra herhangi bir zamanda ilgili proje için tekrar sorgu çalıştırıldığında işlem kaldığın yerden devam edebilmiştir. Bu özellik sayesinde birden fazla proje üzerinde asenkron veri toplanabilmiştir.

Toplanan check-in verilerinden, her check-indeki dosyaların birbiri arasında bir ilişki kuracak şekilde bir yönsüz ağ oluşturulmuştur.

### Ağ Modelleme

Ağ yapımı sırasında birlikte değişen dosyaların arasındaki ilişkiler kullanılmıştır. Bir değişim kaydında A dosyası değişirken B dosyası da aynı anda değişmişse bu durum A'dan B'ye bir kenar oluşturmaktadır. Ağdaki tüm dosyalar bir düğüm, aralarındaki ilişki bir kenar oluşturmaktadır. Oluşturulan ağ doğası gereği çift yönlü bir etkileşim sağlamaktadır.

**Tablo 8:** Ağ özet bilgileri

Proje Adı	Düğüm	Kenar
matlab2tikz	158	7.074
mist	763	134.134
MpAndroid Chart	943	143.080
bootstrap material design	1.405	343.522
httpServer	1.404	581.046
johony-five	1.998	538.582
glide	2.692	581.046
angular Cli	3.301	362.084
prettier	5.252	1.129.772
aws-cli	6.480	876.646

<b>aspnet boiler plate</b>	9.974	2.347.328
----------------------------	-------	-----------



## BÖLÜM 5. BULGULAR VE YORUMLAR

Bu bölümde toplanan veriler ile yapılan ağ analiz sonuçları yorumlanmıştır

### **Karmaşık Ağ Analizi**

Karmaşık ağ analizi, düğümlerin her türlü varlığı temsil ettiği büyük ve karmaşık ağların kenarlar, varlıklar arasındaki her türlü ilişkiyi temsil eden, istatistiklerini, yapılarını ve işlevlerini inceleyen bir tekniktir. Matematik, fizik, bilgisayar bilimi, istatistik ve sosyoloji de dahil olmak üzere çeşitli disiplinlerden teorilere ve yöntemlere dayanmaktadır. Bir ağ analizi çalışması, genellikle karmaşık bir ağ temsili elde edildikten sonra aşağıdaki adımları içerir.

İlk önce, ağ verileri görselleştirilir ve desenler görsel olarak incelenir. Ardından, ağın genel özelliklerini bir bütün olarak anlamak için yapısal analiz gerçekleştirilir. Örneğin, düğümlerin derece dağılımı, kümeleme katsayısı, ortalama yol uzunluğu ve çapı gibi yapısal önlemler bize ağ hakkında önemli bilgiler verir. Üçüncüsü, yapısal olarak en merkezi ve önemli düğümlerin derece, yakınlık, iyi olma ve öz vektör merkezi olma gibi uygun merkezi ölçütler kullanılarak tanımlandığı merkeziyet analizi kullanılır. Ağdaki kümeleri veya yoğun grupları tespit etmek için düğümler arasındaki geçişli ilişkileri ortaya çıkarmak için topluluk analizi denilen son bir analiz gerçekleştirilir. [14]

Bu çalışmada, kurulan ağı görselleştirmek için Gephi ve Cytoscape uygulamaları kullanılmıştır.

#### **5.1.1. Görsel Ağ Analizi**

Ağın görsel yapısının incelenmesi, nitel birtakım çıkarımlar yapmaya veya ağın genel yapısının keşfedilmesi için önemli bir çalışmadır. Bu çalışmada Gephi ağ görselleştirme yazılımının geliştiricileri tarafından hazırlanmış Force-Atlas 2 tasarımı kullanılmıştır. Force-Atlas 2 tasarımı küçük-dünya özelliği gösteren ve ölçekten bağımsız ağların görselleşmesi için optimize edilmiş bir algoritmadır.

Ağların görselleştirilmiş hallerinde net bir şekilde belirli kümelenmelerin oluştuğu görülmektedir. Tüm ağlara ait ağ görselleri Ek 2’de gösterilmiştir.

### 5.1.2. Yapısal Ağ Analizi

Cytoscape uygulaması kullanılarak tüm projelere ait yoğunluk, kümeleme katsayısı, çap, merkezileşme ve karakteristik yol uzunluğu gibi genel yapılan ölçümleri yapılmıştır. Ek 3’te yapılan ölçüm değerleri gösterilmiştir.

Ölçümünü yapılan ağların rastgele oluşup oluşmadığını incelemek için Cytoscape uygulaması ile tüm ağlar için Erdős-Rényi modelinde rastgele birer ağ üreterek sonuçları karşılaştırılmıştır. Bulgular genel beklenti ile uyumlu idi. Rasgele ağların, gerçek dünya ağlarının yüksek kümelenmesini göstermediği bilinen bir gerçektir, yapılan ölçümlerin tamamında rastgele oluşturulan ağların kümeleme katsayısı daha düşük çıkmıştır. Rastgele oluşturulan ağ analiz sonucu Tablo 9’da verilmiştir.

**Tablo 9:** Rastgele ağ analiz sonuçları

Proje Adı	Kark. Yol Uzn.	Kümeleme Kat Sayısı
matlab2tikz	1,43	0,570
mist	1,678	0,322
MpAndroid Chart	1,678	0,322
bootstrap material design	1,652	0,348
httpServer	1,86	0,14
johony-five	1,73	0,27
glide	1,84	0,16
angular Cli	1,934	0,066
pretitier	1,918	0,082
aws-cli	1,958	0,042
aspnet boiler plate	1,936	0,064

### 5.1.3. Derece Dağılımları

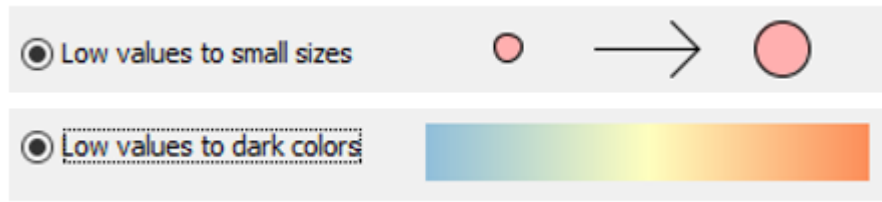
Bir ağdaki düğmelerin derece dağılımları ağı tanımlamak ve ağın karakteristik yapısını anlamak için önemli bir bilgidir. İncelenen projelerin tamamı kuvvet yasası dağılımına uyduğu görülmektedir. Kuvvet yasası dağılımına göre bir ağda derecesi çok

düşük olan çok sayıda düğüm varken, derecesi çok yüksek olan çok az sayıda düğüm bulunur. Kullanılan projelere ait derece dağılım Ek 4’te görülmektedir. Derece dağılım görüntüleri Gephi uygulamasından alınmıştır.

#### 5.1.4. Merkezilik Analizi

Analizin bu aşamasında cytoscape uygulaması kullanarak ağdaki düğümlerin derecesi, yakınlığını ve arasında merkezleri hesaplanmıştır. Bu değerler, ağdaki düğümlerin göreceli önemi hakkında genellikle faydalı bilgiler veren en yaygın kullanılan merkezi önlemlerdir. Merkezilik değeri ne kadar yüksek olursa, bir düğüm o kadar merkezidir.

İncelenen projelere ait merkezilik grafikleri ek 5 ’teki gibidir. Grafik gösteriminde düşük merkezi değerlilikteki düğümü küçük, yüksek merkezi değerlilikteki düğümü büyük yuvarlak olarak gösterilmiştir. Aynı şekilde düşük merkezilikteki düğümler mavi renkli, yüksek merkezilikteki düğümler kırmızı renkli olarak renklendirilmiştir. Gösterim ifadeleri Şekil 13’te ki gibidir.



Şekil 14 : Merkezilik analiz gösterim ifadeleri

Grafiklerde ağın merkezindeki değerler açık renkli ve düğümler büyük görüntülenmiştir. Ağın merkezi düğümlerinin farklı merkezilik yöntemlerine göre de benzer kuvvette çıkması beklenen bir durumdur.

Tüm projelerdeki en yüksek merkezilik değerine sahip dosyalar ek 6’da listelenmiştir. Bu listede öz vektör merkeziliği, derece merkeziliği, arasında merkezilik ve yakınlık merkeziliği sonuçlarına sıralama yapılmıştır. Benzer dosyaların farklı listelerde listelendiği görülmektedir. Farklı merkezilik analizlerinde listelenen dosyalar kalın karakterli olarak listede işaretlenmiştir. Bu sonuçlara göre birden fazla merkezilik analizine göre merkezi kabul edilen dosyaların proje için önemli olduğu kararına

varılmıştır. Bu dosyalarda deęişiklik yapılırken kullanıcının dikkatli olması gerektięi yorumlanabilir.

### **5.1.5. Topluluk Analizi**

Topluluk analizi için Cytoscape uygulaması üzerinden ClusterOne algoritması kullanılmıştır. ClusterOne algoritması, bir protein etkileşim aęındaki çakışan protein komplekslerini bulmak için geliştirilmiştir, büyük ölçekli aęlarda düęümler arasındaki gizli baęları bulmada başarılı sonuçlar vermektedir. Projelere ait tespit edilen ilk 5 alt topluluk grubu bilgileri ek 7’teki gibidir. Topluluk analizi ile projelerdeki gizli modüllerin net bir şekilde ortaya çıktığı görülmektedir.



## BÖLÜM 6. SONUÇ

Bu çalışmadan Github platformunda bağımsız olarak seçilen 11 adet projenin dosya değişim geçmişleri incelenerek bir sosyal ağ oluşturulmuştur. Oluşturulan sosyal ağ CVS dosyası olarak kaydedilerek Gephi ve Cytoscape uygulamaları aracılığıyla ağ analizi yapılmıştır.

Yapılan yapısal ağ analizinde kümeleme katsayısı, karakteristik yol uzunluğu hesaplanmış ve ağların düğüm derece dağılımları elde edilmiştir. Daha sonra incelediğimiz ağlar ile aynı sayıda düğüm ve kenara sahip bir dizi Erdős-Rényi rasgele ağ oluşturulmuştur ve bu rastgele ağlarda aynı ölçüler hesaplanmıştır. Daha sonra, iki değer kümesini ve derece dağılımlarını karşılaştırılmıştır. İncelediğimiz ağların rastgele oluşmadığı, benzer boyutlarda oluşturulan rastgele ağlara göre daha düzenli ve gerçek hayat ağlarına ait özellikler içerdikleri görülmüştür.

Merkezilik ölçütleri kullanılarak ağın tutarlı bir yapıda olup olmadığı incelenmesi sonucunda, ağın tutarlı olduğu, farklı merkezilik yöntemlerinde de benzer desenler gösterdiği tespit edilmiştir. Farklı merkezilik ölçümlerinde de önemli dereceye sahip dosyaların yüksek bağlılık içerdiği ve dosya değişim aşamasında dikkat edilmesi gereken dosyalar olduğu şeklinde yorumlanmıştır.

Ağ üzerinde yapılan topluluk analizinde tüm ağlara ait küçük gruplar tespit edilmiştir. Bu topluluklar analiz edilen projelere ait gizli modüller olarak yorumlanmıştır.

Yapılan incelemelerde bağımsız olarak rastgele seçilen yazılım projeleri incelendiğinde rastgele olmak yerine bilinçli ve anlamlı bir sürecinin sonucu olduğunu göstermektedir.

Son olarak, yazılım projelerindeki dosya değişimlerinin, proje boyutu, konusu, geliştirici sayısı gibi parametrelerden bağımsız olarak gerçek hayat ağlarının tüm kurallarına sahip olduğu için gerçek hayat ağları olarak nitelendirilebilir. Gerçek hayat ağlarında gözlemlene tüm matematiksel açıklamalara dayalı yöntemlerin yazılım projeleri içinde uygulanabileceği tespit edilmiştir.



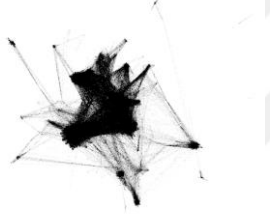



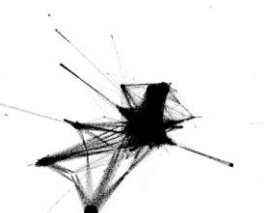


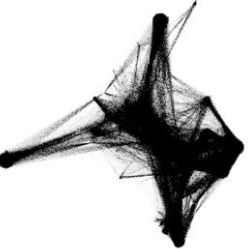
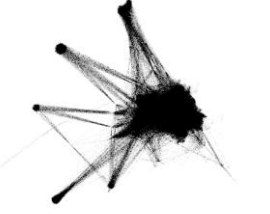


## EK'LER

### Ek 1. Örnek işlem(check-in) açıklaması

```
{
  "commit": {
    "author": {
      "name": "Philipp Jahoda",
      "email": "phil.jahoda@gmail.com",
      "date": "2019-04-27T15:38:41Z"
    },
    "committer": {
      "name": "Philipp Jahoda",
      "email": "phil.jahoda@gmail.com",
      "date": "2019-04-27T15:38:41Z"
    },
    "message": "Merge branch 'master' of github.com:PhilJay/MPAndroidChart",
    "comment_count": 0,
    "verification": {
      "verified": false,
      "reason": "unsigned",
      "signature": null,
      "payload": null
    }
  },
  "author": {
    "login": "PhilJay",
    "id": 6759734,
    "node_id": "MDQ6VXNlcjY3NTk3MzQ=",
    "avatar_url": "https://avatars1.githubusercontent.com/u/6759734?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/PhilJay",
    "html_url": "https://github.com/PhilJay",
    "followers_url": "https://api.github.com/users/PhilJay/followers",
```

```
"following_url": "https://api.github.com/users/PhilJay/following{/other_user}",
"gists_url": "https://api.github.com/users/PhilJay/gists{/gist_id}",
"starred_url": "https://api.github.com/users/PhilJay/starred{/owner}/{/repo}",
"subscriptions_url": "https://api.github.com/users/PhilJay/subscriptions",
"organizations_url": "https://api.github.com/users/PhilJay/orgs",
"repos_url": "https://api.github.com/users/PhilJay/repos",
"events_url": "https://api.github.com/users/PhilJay/events{/privacy}",
"received_events_url": "https://api.github.com/users/PhilJay/received_events",
"type": "User",
"site_admin": false
},
"committer": {
  "login": "PhilJay",
  "id": 6759734,
  "node_id": "MDQ6VXNlcjY3NTk3MzQ=",
  "avatar_url": "https://avatars1.githubusercontent.com/u/6759734?v=4",
  "gravatar_id": "",
  "url": "https://api.github.com/users/PhilJay",
  "html_url": "https://github.com/PhilJay",
  "followers_url": "https://api.github.com/users/PhilJay/followers",
  "following_url": "https://api.github.com/users/PhilJay/following{/other_user}",
  "gists_url": "https://api.github.com/users/PhilJay/gists{/gist_id}",
  "starred_url": "https://api.github.com/users/PhilJay/starred{/owner}/{/repo}",
  "subscriptions_url": "https://api.github.com/users/PhilJay/subscriptions",
  "organizations_url": "https://api.github.com/users/PhilJay/orgs",
  "repos_url": "https://api.github.com/users/PhilJay/repos",
  "events_url": "https://api.github.com/users/PhilJay/events{/privacy}",
  "received_events_url": "https://api.github.com/users/PhilJay/received_events",
  "type": "User",
  "site_admin": false
},
"stats": {
  "total": 24,
  "additions": 16,
```

```
"deletions": 8
},
"files": [
  {
    "sha": "012fab77f9b56c22ea1573d2d0fc09f5a4984cba",
    "filename": "MPChartLib/src/main/java/com/github/mikephil/charting/formatter/PercentFormatter.java",
    "status": "modified",
    "additions": 10,
    "deletions": 2,
    "changes": 12,
    "contents_url":
  },
  {
    "sha": "044a84f9a6aa24ec98f573c95b654598f44741d9",
    "filename": "README.md",
    "status": "modified",
    "additions": 6,
    "deletions": 6,
    "changes": 12,
    "contents_url": "
  }
]
}
```

<b>angular-cli</b> 	<b>Aspnetboilerplate</b> 	<b>johnny-five</b> 	<b>MpAndroidChart</b> 
<b>aws-cli</b> 	<b>bootsrapt-material-design</b> 	<b>matlab2tikz</b> 	<b>Prettier</b> 
<b>Glide</b> 	<b>http-server</b> 	<b>Mist</b> 	

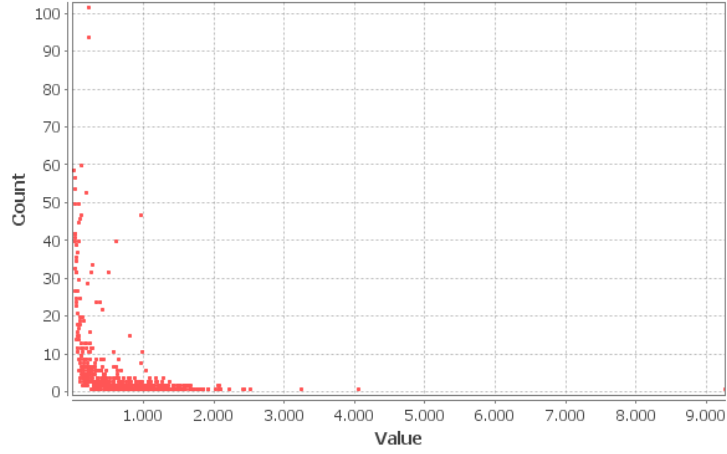
Ek 3. Ağ analizi sonuçları

	Düğüm	Kenar	Kümeleme Kat Sayısı	Karakteristik Yol Uzunluğu	Yoğunluk	Bağlı Bileşenler	Çap	Yarı Çap	Ortalama Komşu Say.
<b>matlab2tikz</b>	158	7.074	0,842	1,904	0,285	2	4	1	44,772
<b>mist</b>	763	134.134	0,841	1,791	0,231	2	4	1	175,8
<b>MpAndroid Chart</b>	943	143.080	0,837	2,01	0,161	2	5	1	151,729
<b>Bootstrap material design</b>	1.405	343.522	0,85	1,953	0,174	4	4	1	244,535
<b>httpServer</b>	1.404	581.046	0,81	2,38	0,07	1	7	4	98,084
<b>johony-five</b>	1.998	538.582	0,848	2,155	0,135	8	6	1	269,622
<b>glide</b>	2.692	581.046	0,836	2,594	0,08	2	10	1	215,84
<b>angular Cli</b>	3.301	362.084	0,84	2,7	0,033	8	7	1	109,689
<b>prettier</b>	5.252	1.129.772	0,881	2,383	0,041	2	6	1	215,113
<b>aws-cli</b>	6.480	876.646	0,933	2,204	0,021	30	6	1	134,98
<b>aspnet boiler plate</b>	9.974	2.347.328	0,723	3,087	0,024	6	7	1	235,565

#### Ek 4. Derece dağılımları sonuçları

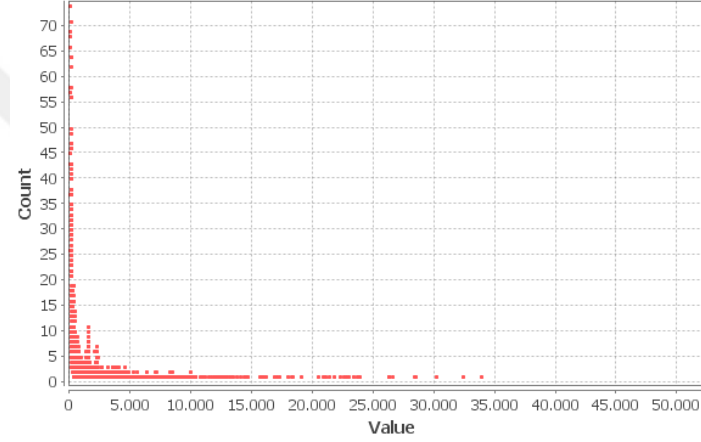
**Angular Cli**

**Degree Distribution**



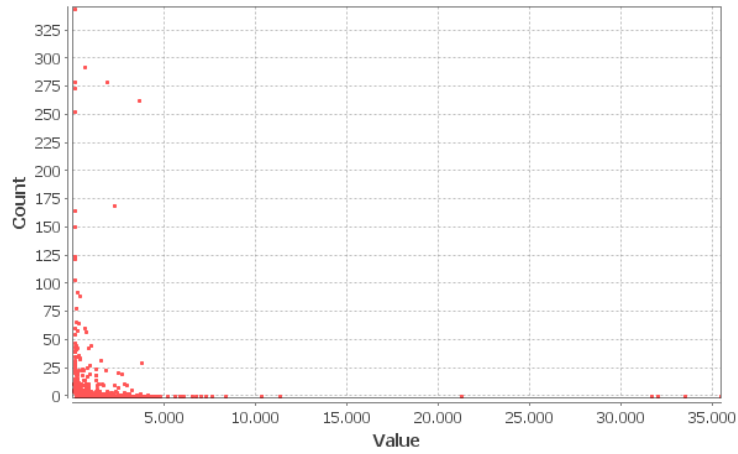
**Aspnetboilerplate**

**Degree Distribution**



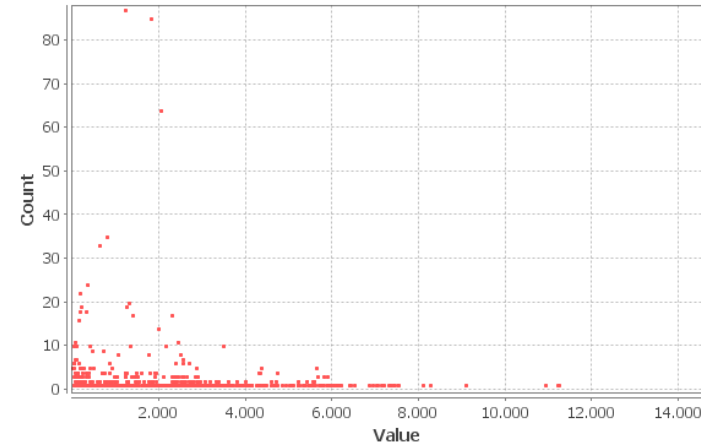
**Aws-cli**

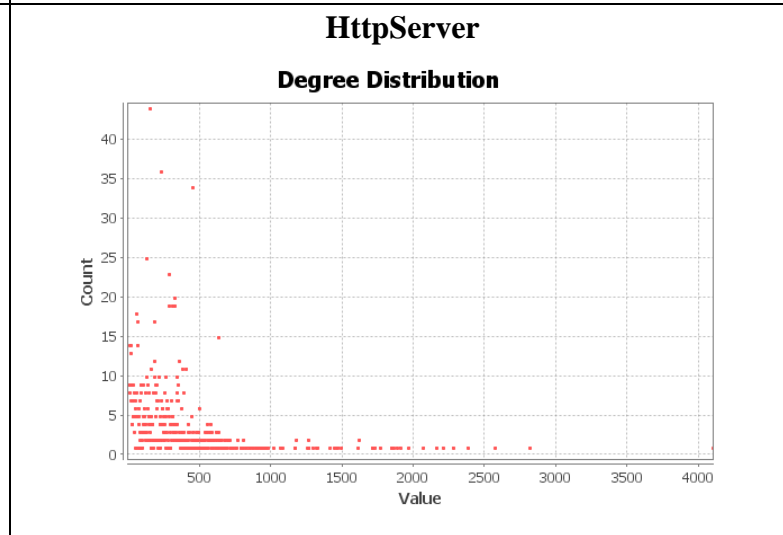
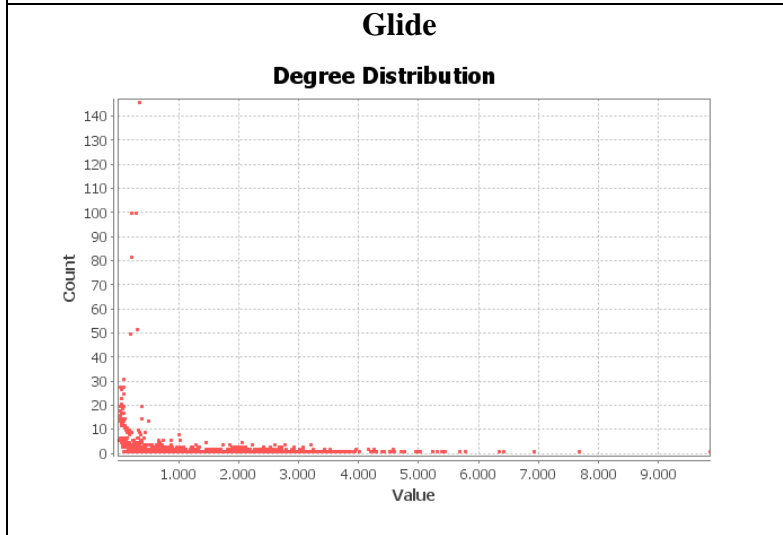
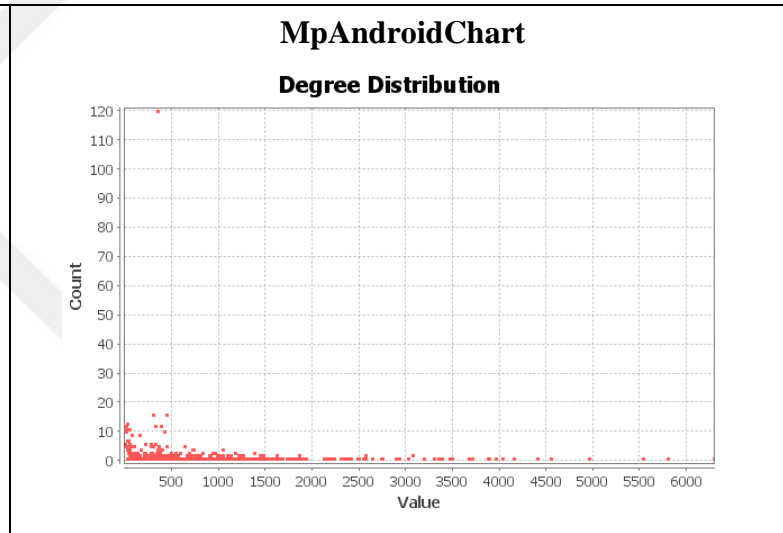
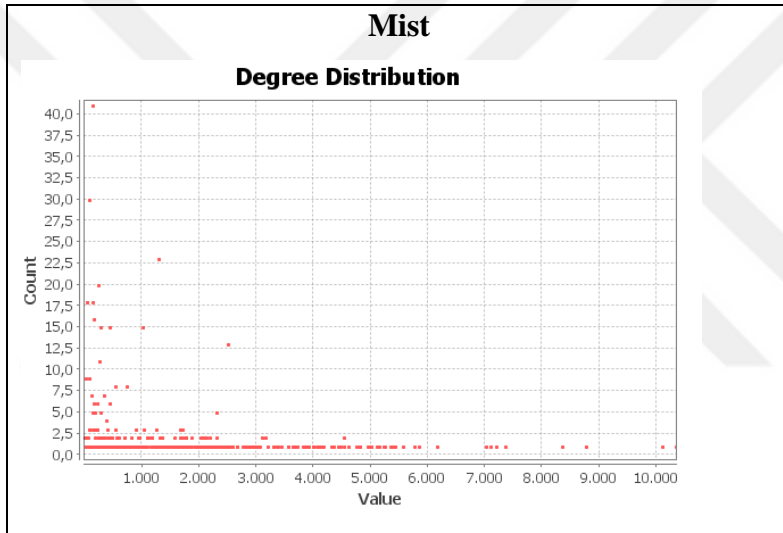
**Degree Distribution**



**Bootstrap-material-design**

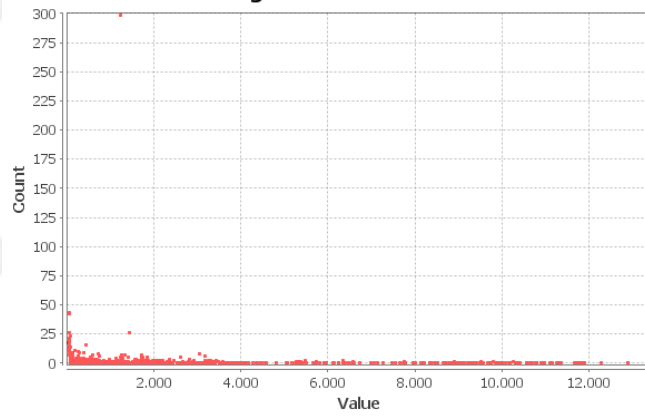
**Degree Distribution**





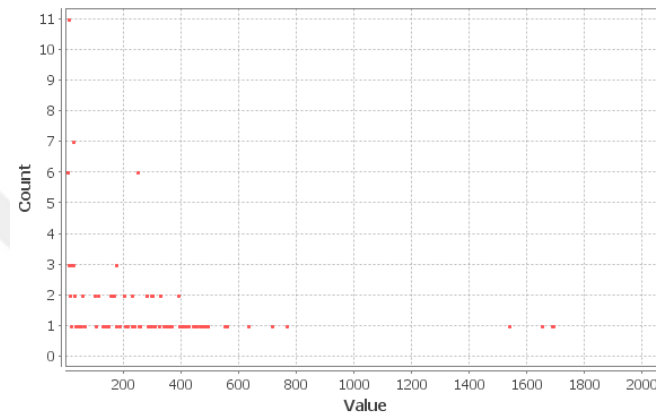
### Johony-five

#### Degree Distribution



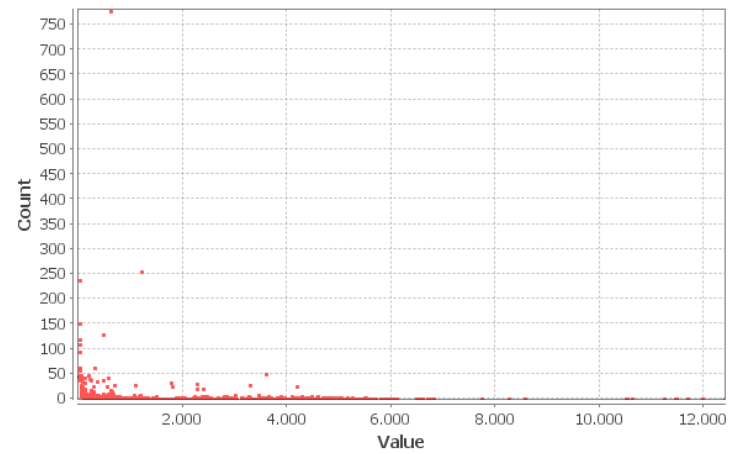
### Matlab2tikz

#### Degree Distribution



### Prettier

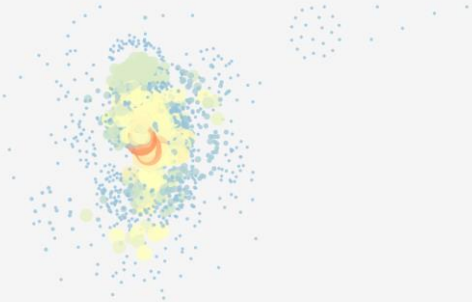
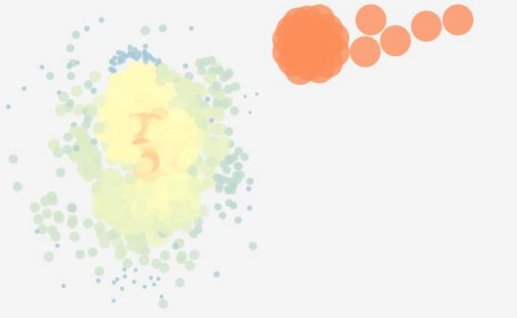

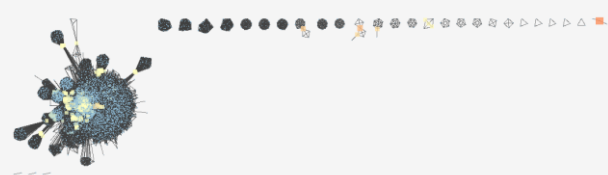


#### Degree Distribution



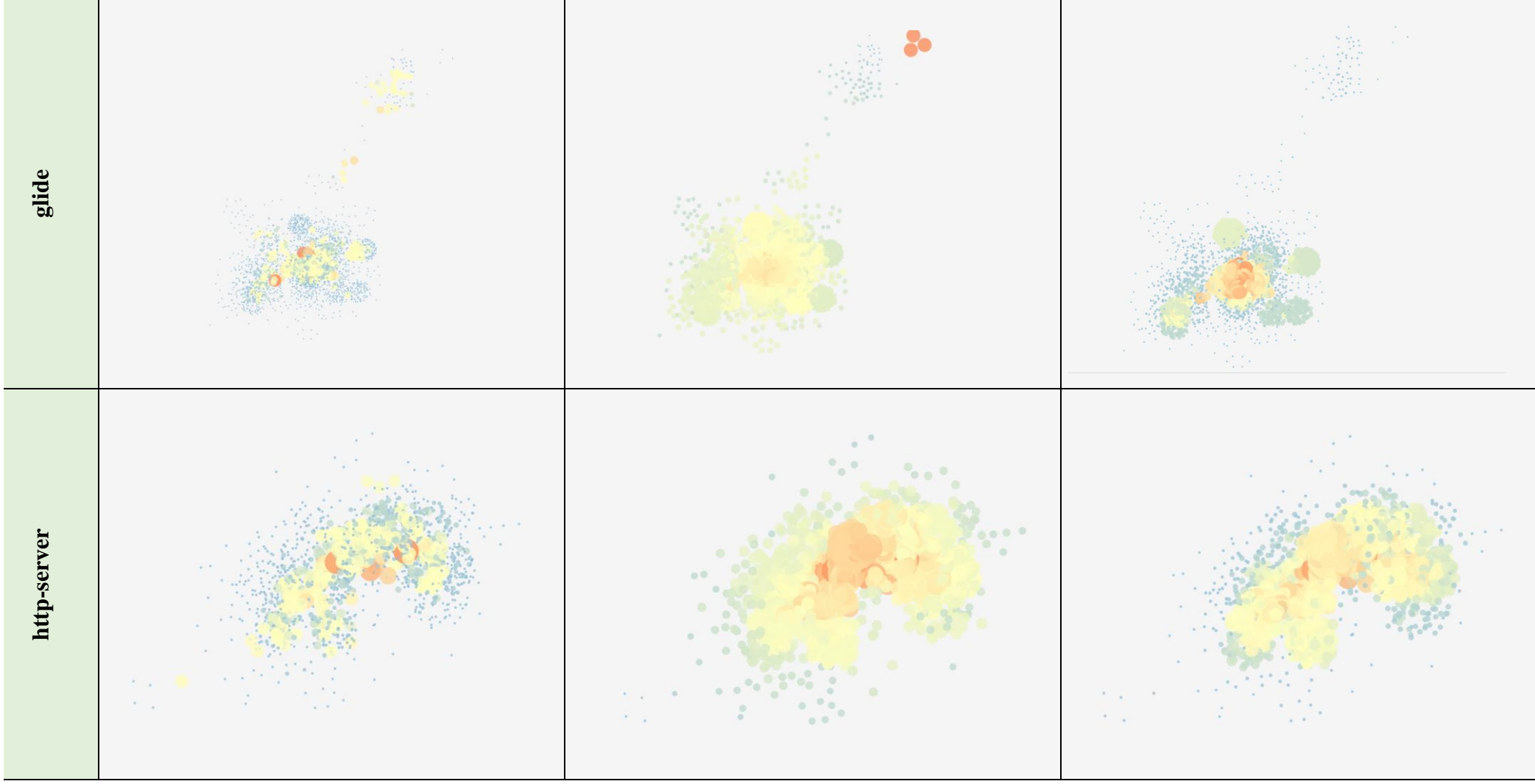


Ek 5. Merkezilik analizi görsel sonuçları

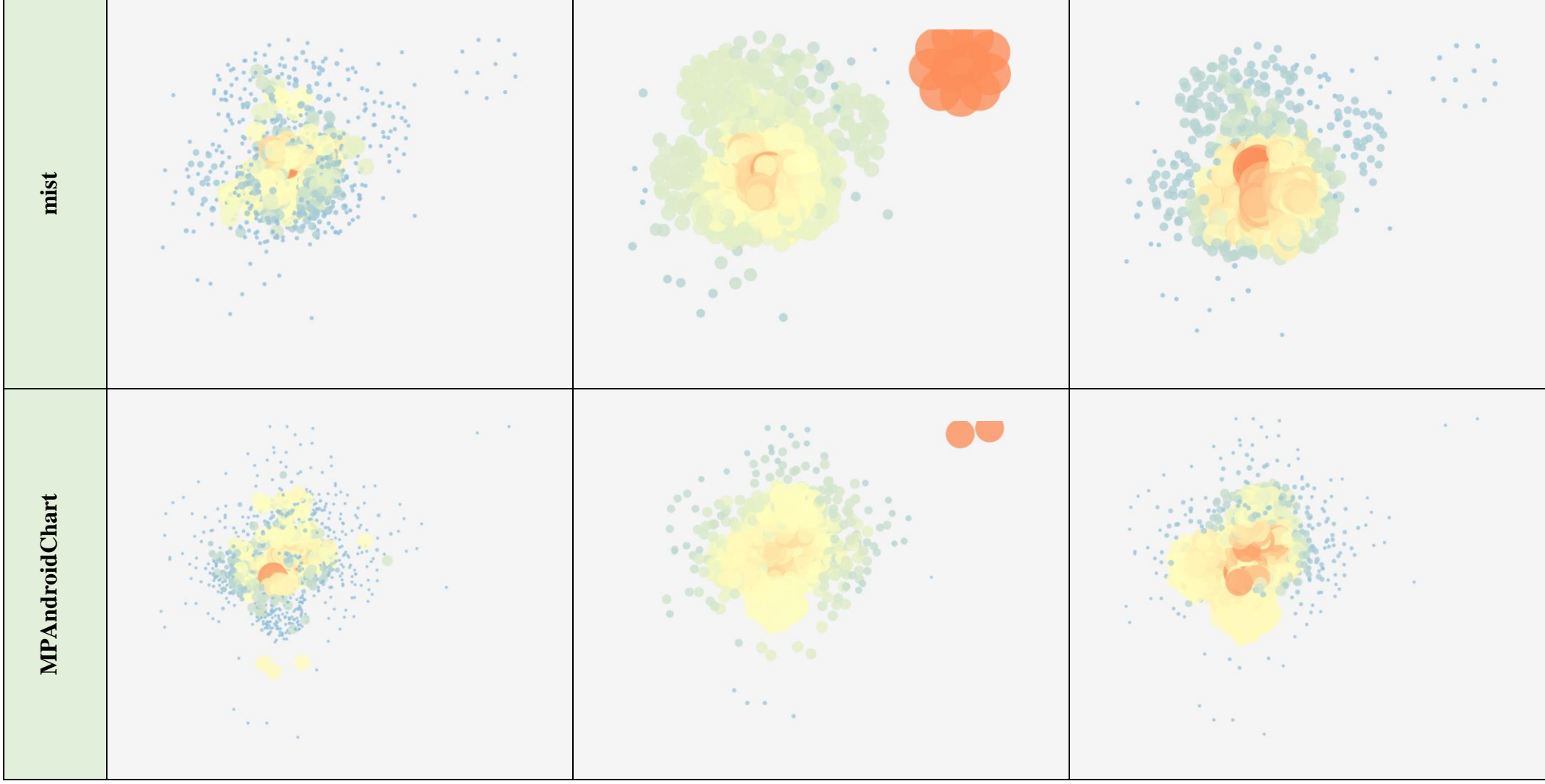
Proje Adı	Arasında Merkeziliği	Yakınlık Merkeziliği	Derece Merkeziliği
Angular-cli			
aspnetboilerplate			

<b>bootstrapMD</b>			
<b>aws-cli</b>			

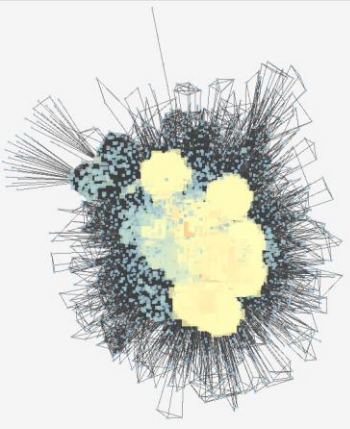
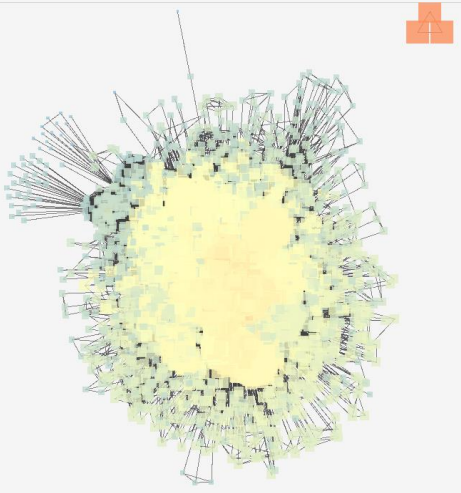
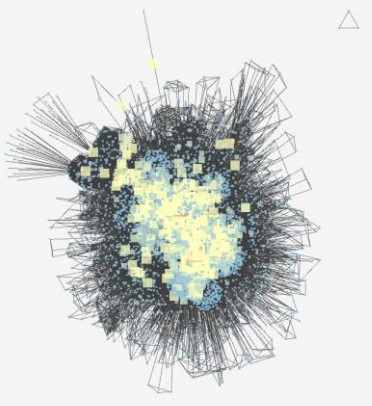
c







prettier



Ek 6. Merkezilik analizi ilk 10 dosya sonuçları

Proje		Özvektör Merkeziliği	Derece Merkeziliği	Arasındalık Merkeziliği	Yakınlık Merkeziliği
angular-cli	1	package.json	package.json	package.json	package.json
	2	packages/@angular/cli/package.json	lib/bootstrap-local.js	yarn.lock	package-lock.json
	3	packages/@angular/cli/commands/build.ts	packages/@angular/cli/package.json	tsconfig.json	yarn.lock
	4	packages/@angular/cli/commands/serve.ts	.eslintignore	README.md	lib/bootstrap-local.js
	5	packages/@angular/cli/commands/e2e.ts	bin/ng	package-lock.json	packages/@angular/cli/package.json
	6	packages/@angular/cli/commands/test.ts	package-lock.json	.circleci/config.yml	.eslintignore
	7	packages/@angular/cli/commands/version.ts	packages/@angular/cli/commands/generate.ts	docs/documentation/1.x/stories.md	bin/ng
	8	packages/@angular/cli/commands/lint.ts	packages/@angular/cli/commands/build.ts	packages/schematics/angular/library/index.ts	packages/@angular/cli/commands/build.ts
	9	packages/@angular/cli/commands/generate.ts	tests/e2e/utils/project.ts	packages/angular/cli/lib/config/schema.json	packages/@angular/cli/commands/generate.ts
	10	packages/@angular/cli/commands/new.ts	packages/@angular/cli/commands/serve.ts	tests/legacy-cli/e2e/utils/project.ts	packages/@angular/cli/commands/serve.ts
aspnetboilerplate	1	src/Abp/Abp.csproj	src/Abp.sln	src/Abp.sln	src/Abp.sln
	2	src/Abp.NHibernate/Abp.NHibernate.csproj	src/Abp/Abp.csproj	doc/README.md	src/Abp/Abp.csproj
	3	src/Abp.EntityFramework/EntityFramework/AbpDbContext.cs	src/Abp.NHibernate/Abp.NHibernate.csproj	doc/WebSite/Documents.md	doc/README.md

	4	<b>src/Abp.Web.Api/Abp.Web.Api.csproj</b>	<b>src/Abp.Web.Api/Abp.Web.Api.csproj</b>	<b>src/Abp/Abp.csproj</b>	<b>src/Abp.NHibernate/Abp.NHibernate.csproj</b>	
	5	<b>src/Abp.Web.Mvc/Abp.Web.Mvc.csproj</b>	<b>src/Abp.Web.Mvc/Abp.Web.Mvc.csproj</b>	src/Abp/Modules/Core/Abp.Modules.Core.Application/Abp.Modules.Core.Application.csproj	<b>src/Abp.Web.Api/Abp.Web.Api.csproj</b>	
	6	src/Abp.MongoDB/Abp.MongoDB.csproj	<b>Abp.sln</b>	src/Abp/Framework/Abp.Application/Abp.Application.csproj	<b>src/Abp.Web.Mvc/Abp.Web.Mvc.csproj</b>	
	7	<b>src/Abp/AbpKernelModule.cs</b>	<b>README.md</b>	<b>Abp.sln</b>	doc/WebSiteContents/documents.html	
	8	<b>src/Abp.EntityFramework/Abp.EntityFramework.csproj</b>	<b>src/Abp/AbpKernelModule.cs</b>	src/packages/AutoMapper.3.1.1/lib/net40/AutoMapper.Net4.dll	doc/WebSiteContents/entities.html	
	9	src/Abp.FluentMigrator/Abp.FluentMigrator.csproj	<b>src/Abp.EntityFramework/Abp.EntityFramework.csproj</b>	src/Abp/Modules/Core/Abp.Modules.Core.Application/packages.config	<b>src/Abp.EntityFramework/Abp.EntityFramework.csproj</b>	
	10	src/Abp.AutoMapper/Abp.AutoMapper.csproj	<b>src/Abp.EntityFramework/EntityFramework/AbpDbContext.cs</b>	src/Abp/Framework/Abp.Web.Mvc/Abp/Framework/scripts/abp.js	doc/WebSiteContents/handling-exceptions.html	
	<b>aws-cli</b>	1	<b>awscli/__init__.py</b>	<b>awscli/__init__.py</b>	<b>awscli/__init__.py</b>	<b>awscli/__init__.py</b>
		2	<b>doc/source/conf.py</b>	<b>doc/source/conf.py</b>	<b>doc/source/conf.py</b>	<b>doc/source/conf.py</b>
		3	<b>setup.py</b>	<b>setup.py</b>	<b>setup.py</b>	<b>setup.py</b>
4		<b>CHANGELOG.rst</b>	<b>CHANGELOG.rst</b>	<b>CHANGELOG.rst</b>	<b>CHANGELOG.rst</b>	
5		<b>awscli/clidriver.py</b>	<b>setup.cfg</b>	<b>setup.cfg</b>	<b>setup.cfg</b>	
6		<b>README.rst</b>	<b>awscli/clidriver.py</b>	<b>awscli/clidriver.py</b>	<b>awscli/clidriver.py</b>	



	7	awscli/customizations/s3/s3handler.py	<b>awscli/data/cli.json</b>	awscli/doc/man/man1/elastictranscoder-create-job.1	<b>README.rst</b>
	8	awscli/customizations/s3/util.py	<b>README.rst</b>	<b>awscli/data/cli.json</b>	<b>awscli/data/cli.json</b>
	9	awscli/argprocess.py	<b>awscli/handlers.py</b>	<b>README.rst</b>	<b>awscli/handlers.py</b>
	10	<b>awscli/handlers.py</b>	awscli/customizations/s3/subcommands.py	MANIFEST.in	awscli/examples/ec2/run-instances.rst
<b>bootstrapMD</b>	1	<b>docs/_includes/header.html</b>	<b>README.md</b>	<b>README.md</b>	<b>README.md</b>
	2	<b>docs/_includes/footer.html</b>	<b>bower.json</b>	<b>package.json</b>	<b>bower.json</b>
	3	<b>_config.yml</b>	<b>docs/_includes/header.html</b>	<b>Gruntfile.js</b>	<b>docs/_includes/header.html</b>
	4	<b>docs/assets/js/vendor/holder.min.js</b>	<b>Gruntfile.js</b>	<b>bower.json</b>	<b>package.json</b>
	5	docs/LICENSE	<b>package.json</b>	<b>.gitignore</b>	<b>Gruntfile.js</b>
	6	docs/assets/img/devices.png	<b>_config.yml</b>	<b>docs/_includes/header.html</b>	<b>Gemfile</b>
	7	docs/apple-touch-icon.png	<b>docs/_includes/footer.html</b>	<b>docs/assets/scss/docs.scss</b>	<b>_config.yml</b>
	8	docs/_data/browser-bugs.yml	<b>Gemfile</b>	<b>_config.yml</b>	<b>docs/_includes/footer.html</b>
	9	docs/_includes/ads.html	<b>docs/assets/scss/docs.scss</b>	LICENSE.md	<b>docs/assets/scss/docs.scss</b>
	10	docs/_data/translations.yml	<b>docs/assets/js/vendor/holder.min.js</b>	<b>.travis.yml</b>	<b>docs/assets/js/vendor/holder.min.js</b>
<b>glide</b>	1	<b>library/src/main/java/com/bumptech/glide/Glide.java</b>	<b>library/src/main/java/com/bumptech/glide/Glide.java</b>	<b>.gitmodules</b>	<b>library/src/main/java/com/bumptech/glide/Glide.java</b>
	2	<b>library/src/main/java/com/bumptech/glide/load/engine/Engine.java</b>	<b>library/build.gradle</b>	library/src/com/bumptech/glide/Glide.java	<b>library/src/main/java/com/bumptech/glide/load/engine/Engine.java</b>

	3	<b>library/src/main/java/com/bumptech/glide/load/engine/EngineJob.java</b>	<b>checkstyle_suppressions.xml</b>	<b>checkstyle_suppressions.xml</b>	<b>library/src/main/java/com/bumptech/glide/RequestManager.java</b>
	4	<b>library/src/main/java/com/bumptech/glide/GlideBuilder.java</b>	<b>library/src/main/java/com/bumptech/glide/load/engine/Engine.java</b>	<b>.gitignore</b>	<b>library/src/main/java/com/bumptech/glide/GlideBuilder.java</b>
	5	<b>library/src/main/java/com/bumptech/glide/load/ResourceDecoder.java</b>	<b>library/src/main/java/com/bumptech/glide/RequestManager.java</b>	library/src/com/bumptech/glide/volley/VolleyUrlLoader.java	<b>library/build.gradle</b>
	6	<b>library/src/main/java/com/bumptech/glide/RequestManager.java</b>	<b>library/src/main/java/com/bumptech/glide/GlideBuilder.java</b>	<b>library/build.gradle</b>	<b>build.gradle</b>
	7	library/src/main/java/com/bumptech/glide/load/engine/EngineKey.java	<b>build.gradle</b>	<b>build.gradle</b>	<b>library/src/main/java/com/bumptech/glide/load/engine/EngineJob.java</b>
	8	library/src/main/java/com/bumptech/glide/load/model/FileLoader.java	<b>library/src/main/java/com/bumptech/glide/load/engine/EngineJob.java</b>	src/com/bumptech/photos/PhotoManager.java	<b>checkstyle_suppressions.xml</b>
	9	library/src/main/java/com/bumptech/glide/load/model/UriLoader.java	<b>library/src/main/java/com/bumptech/glide/load/ResourceDecoder.java</b>	library/src/com/bumptech/glide/resize/bitmap_recycle/Poolable.java	gradle.properties
	10	library/src/main/java/com/bumptech/glide/load/model/ResourceLoader.java	library/src/main/java/com/bumptech/glide/ListPreloader.java	<b>library/src/main/java/com/bumptech/glide/Glide.java</b>	library/src/main/java/com/bumptech/glide/load/engine/bitmap_recycle/LruBitmapPool.java
<b>http-server</b>	1	<b>bin/aerys</b>	<b>src/Aerys/Server.php</b>	<b>src/Aerys/Server.php</b>	<b>src/Aerys/Server.php</b>
	2	<b>src/Aerys/Server.php</b>	<b>bin/aerys</b>	<b>bin/aerys</b>	<b>bin/aerys</b>
	3	<b>lib/Server.php</b>	<b>.gitmodules</b>	<b>.gitmodules</b>	<b>.gitmodules</b>
	4	<b>.gitmodules</b>	<b>lib/Server.php</b>	<b>README.md</b>	<b>lib/Server.php</b>
	5	<b>lib/Response.php</b>	examples/hello_world.php	autoload.php	<b>README.md</b>































	6	<b>lib/Client.php</b>	<b>lib/Host.php</b>	<b>lib/Server.php</b>	<b>lib/Host.php</b>
	7	<b>lib/Websocket/Endpoint.php</b>	<b>lib/Response.php</b>	composer.json	<b>vendor/Amp</b>
	8	<b>lib/Host.php</b>	<b>lib/Client.php</b>	<b>vendor/Amp</b>	<b>lib/Response.php</b>
	9	vendor/Auryn	examples/file_server.php	lib/Bootstrapper.php	<b>lib/Client.php</b>
	10	src/Aerys/Responders/Websocket/Session.php	<b>vendor/Amp</b>	<b>lib/Host.php</b>	<b>lib/Websocket/Endpoint.php</b>
<b>johnny-five</b>	1	<b>README.md</b>	<b>README.md</b>	<b>.gitignore</b>	<b>README.md</b>
	2	<b>docs/led-rainbow.md</b>	<b>tpl/programs.json</b>	<b>README.md</b>	<b>tpl/programs.json</b>
	3	<b>tpl/programs.json</b>	<b>docs/led-rainbow.md</b>	<b>tpl/programs.json</b>	<b>docs/led-rainbow.md</b>
	4	<b>docs/grove-lcd-rgb-temperature-display.md</b>	<b>docs/grove-lcd-rgb-temperature-display.md</b>	<b>lib/johnny-five.js</b>	<b>docs/grove-lcd-rgb-temperature-display.md</b>
	5	<b>docs/sensor-fsr.md</b>	<b>Gruntfile.js</b>	<b>Gruntfile.js</b>	<b>Gruntfile.js</b>
	6	<b>docs/proximity.md</b>	<b>lib/johnny-five.js</b>	<b>docs/grove-lcd-rgb-temperature-display.md</b>	<b>lib/johnny-five.js</b>
	7	<b>docs/servo-continuous.md</b>	<b>docs/proximity.md</b>	<b>docs/led-rainbow.md</b>	<b>docs/proximity.md</b>
	8	docs/motor.md	<b>docs/sensor-fsr.md</b>	<b>package.json</b>	<b>docs/sensor-fsr.md</b>
	9	<b>Gruntfile.js</b>	<b>docs/servo-continuous.md</b>	lib/motor.js	<b>docs/servo-continuous.md</b>
	10	docs/motor-directional.md	<b>lib/accelerometer.js</b>	<b>docs/proximity.md</b>	<b>lib/accelerometer.js</b>
<b>Matlab2tikz</b>	1	<b>test/suites/ACID.MATLAB.8.3.md5</b>	<b>src/matlab2tikz.m</b>	<b>src/matlab2tikz.m</b>	<b>src/matlab2tikz.m</b>


























	2	<b>test/suites/ACID.Octave.3.8.0.md5</b>	<b>test/suites/ACID.MATLAB.8.3.md5</b>	test/matlab2tikz_acidtest.m	<b>test/suites/ACID.MATLAB.8.3.md5</b>
	3	<b>test/suites/ACID.MATLAB.8.4.md5</b>	<b>test/suites/ACID.Octave.3.8.0.md5</b>	test/testfunctions.m	<b>test/suites/ACID.Octave.3.8.0.md5</b>
	4	<b>test/suites/ACID.m</b>	<b>test/suites/ACID.MATLAB.8.4.md5</b>	test/tex/Makefile	<b>test/suites/ACID.MATLAB.8.4.md5</b>
	5	<b>test/testHeadless.m</b>	<b>test/suites/ACID.m</b>	test/ACID.m	<b>test/suites/ACID.m</b>
	6	<b>test/suites/private/herrorbar.m</b>	<b>test/testHeadless.m</b>	<b>test/suites/ACID.MATLAB.8.3.md5</b>	<b>test/testHeadless.m</b>
	7	<b>test/README.md</b>	<b>test/suites/private/herrorbar.m</b>	<b>test/suites/ACID.Octave.3.8.0.md5</b>	<b>test/private/testMatlab2tikz.m</b>
	8	<b>test/private/execute_tikz_stage.m</b>	<b>test/private/testMatlab2tikz.m</b>	<b>test/suites/ACID.MATLAB.8.4.md5</b>	<b>test/suites/private/herrorbar.m</b>
	9	<b>.travis.yml</b>	<b>test/README.md</b>	ChangeLog	<b>test/README.md</b>
	10	test/testGraphical.m	<b>test/private/execute_tikz_stage.m</b>	test/data/.gitignore	<b>test/private/execute_tikz_stage.m</b>
<b>mist</b>	1	<b>main.js</b>	<b>main.js</b>	<b>package.json</b>	<b>main.js</b>
	2	<b>package.json</b>	<b>package.json</b>	<b>main.js</b>	<b>package.json</b>
	3	<b>interface/i18n/mist.en.i18n.json</b>	<b>gulpfile.js</b>	<b>gulpfile.js</b>	<b>gulpfile.js</b>
	4	<b>modules/ipc/ipcProviderBackend.js</b>	<b>modules/ipc/ipcProviderBackend.js</b>	<b>modules/ipc/ipcProviderBackend.js</b>	<b>modules/ipc/ipcProviderBackend.js</b>
	5	<b>gulpfile.js</b>	<b>interface/i18n/mist.en.i18n.json</b>	<b>.gitignore</b>	<b>interface/i18n/mist.en.i18n.json</b>
	6	<b>interface/client/appStart.js</b>	<b>interface/.meteor/versions</b>	modules/preloader/splashScreen.js	<b>interface/.meteor/versions</b>
	7	modules/menuItems.js	<b>interface/.meteor/packages</b>	<b>interface/.meteor/versions</b>	<b>interface/.meteor/packages</b>

	8	<b>interface/.meteor/versions</b>	<b>interface/client/appStart.js</b>	modules/i18n.js	<b>interface/client/appStart.js</b>
	9	<b>interface/client/styles/layout.import.less</b>	<b>interface/client/styles/layout.import.less</b>	<b>interface/.meteor/packages</b>	<b>interface/client/styles/layout.import.less</b>
	10	<b>modules/preloader/mistUI.js</b>	<b>modules/preloader/mistUI.js</b>	modules/preloader/wallet.js	<b>modules/preloader/mistUI.js</b>
<b>MPAndroidChart</b>	1	<b>MPChartExample/build.gradle</b>	<b>MPChartExample/build.gradle</b>	<b>README.md</b>	<b>MPChartExample/build.gradle</b>
	2	<b>MPChartLib/build.gradle</b>	<b>MPChartLib/build.gradle</b>	<b>MPChartExample/build.gradle</b>	<b>MPChartLib/build.gradle</b>
	3	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityScatter.java</b>	<b>README.md</b>	<b>MPChartLib/build.gradle</b>	<b>README.md</b>
	4	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityLine.java</b>	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/BarChartActivity.java</b>	MPChartExample/AndroidManifest.xml	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/BarChartActivity.java</b>
	5	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityBar.java</b>	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityScatter.java</b>	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/BarChartActivity.java</b>	MPChartExample/src/com/xxmassdeveloper/mpchartexample/BarChartActivityMultiDataset.java
	6	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmBaseActivity.java</b>	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityLine.java</b>	MPChartLib/src/com/github/mikephil/charting/charts/BarLineChartBase.java	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityScatter.java</b>
	7	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityBubble.java</b>	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityBar.java</b>	MPChartExample/src/com/xxmassdeveloper/mpchartexample/ScatterChartActivity.java	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmBaseActivity.java</b>
	8	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityPie.java</b>	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmBaseActivity.java</b>	MPChartExample/src/com/xxmassdeveloper/mpchartexample/LineChartActivity1.java	<b>MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityLine.java</b>

	9	MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityRadar.java	MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityPie.java	MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityScatter.java	MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityBar.java
	10	MPChartExample/src/com/xxmassdeveloper/mpchartexample/BarChartActivity.java	MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityBubble.java	MPChartLib/src/com/github/mikephil/charting/charts/Chart.java	MPChartExample/src/com/xxmassdeveloper/mpchartexample/realm/RealmDatabaseActivityPie.java
prettier	1	tests/classes/__snapshots__/jsfmt.spec.js.snap	package.json	package.json	package.json
	2	tests/arrows/__snapshots__/jsfmt.spec.js.snap	src/printer.js	src/printer.js	src/printer.js
	3	tests/async/__snapshots__/jsfmt.spec.js.snap	tests/classes/__snapshots__/jsfmt.spec.js.snap	yarn.lock	yarn.lock
	4	package.json	yarn.lock	README.md	tests/classes/__snapshots__/jsfmt.spec.js.snap
	5	tests/arrays/__snapshots__/jsfmt.spec.js.snap	tests/arrows/__snapshots__/jsfmt.spec.js.snap	src/language-js/printer-estree.js	tests/arrows/__snapshots__/jsfmt.spec.js.snap
	6	tests/computed_props/__snapshots__/jsfmt.spec.js.snap	tests/async/__snapshots__/jsfmt.spec.js.snap	tests/classes/__snapshots__/jsfmt.spec.js.snap	tests/destructuring/__snapshots__/jsfmt.spec.js.snap
	7	tests/array_spread/__snapshots__/jsfmt.spec.js.snap	tests/arrays/__snapshots__/jsfmt.spec.js.snap	index.js	README.md
	8	tests/es6modules/__snapshots__/jsfmt.spec.js.snap	tests/computed_props/__snapshots__/jsfmt.spec.js.snap	tests_config/run_spec.js	tests/comments/__snapshots__/jsfmt.spec.js.snap
	9	tests/export_default/__snapshots__/jsfmt.spec.js.snap	tests/array_spread/__snapshots__/jsfmt.spec.js.snap	src/util.js	index.js
	10	tests/destructuring/__snapshots__/jsfmt.spec.js.snap	tests/es6modules/__snapshots__/jsfmt.spec.js.snap	tests/comments/__snapshots__/jsfmt.spec.js.snap	tests_config/run_spec.js

Ek 7. Topluluk analizi sonuçları

Proje Adı	Top. adedi	Grup 1	Grup 2	Grup 3	Grup 4	Grup 5
Anglar Cli	143	 Nodes: 345 Density: 1,420 Quality: 0,758 P-value: 0,000	 Nodes: 259 Density: 1,275 Quality: 0,624 P-value: 0,000	 Nodes: 232 Density: 1,295 Quality: 0,547 P-value: 0,000	 Nodes: 200 Density: 0,501 Quality: 0,610 P-value: 0,000	 Nodes: 188 Density: 1,746 Quality: 0,622 P-value: 0,000
aspnetboilerplate	293	 Nodes: 2057 Density: 0,491 Quality: 0,754 P-value: 0,000	 Nodes: 337 Density: 1,435 Quality: 0,723 P-value: 0,000	 Nodes: 240 Density: 1,999 Quality: 0,999 P-value: 0,000	 Nodes: 238 Density: 1,893 Quality: 0,999 P-value: 0,000	 Nodes: 269 Density: 1,957 Quality: 0,708 P-value: 0,000
aws-c	486	 Nodes: 596 Density: 0,643 Quality: 0,599 P-value: 0,000	 Nodes: 458 Density: 1,413 Quality: 0,741 P-value: 0,000	 Nodes: 330 Density: 1,709 Quality: 0,953 P-value: 0,000	 Nodes: 307 Density: 1,994 Quality: 0,933 P-value: 0,000	 Nodes: 291 Density: 2,000 Quality: 0,930 P-value: 0,000
bootstarpMD	17	 Nodes: 709 Density: 0,791 Quality: 0,810 P-value: 0,000	 Nodes: 313 Density: 0,676 Quality: 0,726 P-value: 0,000	 Nodes: 295 Density: 1,618 Quality: 0,668 P-value: 0,000	 Nodes: 44 Density: 2,000 Quality: 0,565 P-value: 0,000	 Nodes: 21 Density: 2,000 Quality: 1,000 P-value: 0,000
glide	56	 Nodes: 1343 Density: 0,500 Quality: 0,961 P-value: 0,000	 Nodes: 232 Density: 1,089 Quality: 0,905 P-value: 0,000	 Nodes: 163 Density: 1,978 Quality: 0,931 P-value: 0,000	 Nodes: 137 Density: 2,000 Quality: 0,945 P-value: 0,000	 Nodes: 100 Density: 2,000 Quality: 0,892 P-value: 0,000
http-server	33	 Nodes: 301 Density: 0,797 Quality: 0,697 P-value: 0,000	 Nodes: 185 Density: 0,853 Quality: 0,574 P-value: 0,000	 Nodes: 135 Density: 1,180 Quality: 0,744 P-value: 0,000	 Nodes: 158 Density: 1,026 Quality: 0,581 P-value: 0,000	 Nodes: 184 Density: 0,502 Quality: 0,482 P-value: 0,000

<b>johnny-five</b>	68	 <p>Nodes: 1369 Density: 0,473 Quality: 0,992 P-value: 0,000</p>	 <p>Nodes: 301 Density: 2,000 Quality: 0,994 P-value: 0,000</p>	 <p>Nodes: 16 Density: 2,000 Quality: 0,800 P-value: 4,485E-8</p>	 <p>Nodes: 14 Density: 2,000 Quality: 1,000 P-value: 1,173E-7</p>	 <p>Nodes: 7 Density: 2,000 Quality: 1,000 P-value: 2,062E-4</p>
<b>matlab2tikz</b>	10	 <p>Nodes: 122 Density: 0,910 Quality: 0,971 P-value: 0,000</p>	 <p>Nodes: 31 Density: 0,503 Quality: 0,351 P-value: 4,569E-4</p>	 <p>Nodes: 26 Density: 0,523 Quality: 0,392 P-value: 0,009</p>	 <p>Nodes: 5 Density: 1,000 Quality: 0,333 P-value: 0,335</p>	 <p>Nodes: 8 Density: 1,214 Quality: 0,121 P-value: 0,435</p>
<b>Mist</b>	7	 <p>Nodes: 600 Density: 0,690 Quality: 0,973 P-value: 0,000</p>	 <p>Nodes: 59 Density: 1,890 Quality: 0,638 P-value: 0,000</p>	 <p>Nodes: 30 Density: 2,000 Quality: 0,592 P-value: 0,000</p>	 <p>Nodes: 31 Density: 1,781 Quality: 0,421 P-value: 5,439E-9</p>	 <p>Nodes: 38 Density: 1,952 Quality: 0,472 P-value: 9,859E-8</p>
<b>MPAndroidChart</b>	15	 <p>Nodes: 541 Density: 0,500 Quality: 0,780 P-value: 0,000</p>	 <p>Nodes: 209 Density: 1,328 Quality: 0,684 P-value: 0,000</p>	 <p>Nodes: 152 Density: 1,974 Quality: 0,715 P-value: 0,000</p>	 <p>Nodes: 158 Density: 1,023 Quality: 0,515 P-value: 0,000</p>	 <p>Nodes: 3 Density: 2,000 Quality: 0,500 P-value: 0,023</p>
<b>prettier</b>	246	 <p>Nodes: 460 Density: 1,580 Quality: 0,743 P-value: 0,000</p>	 <p>Nodes: 487 Density: 0,755 Quality: 0,612 P-value: 0,000</p>	 <p>Nodes: 455 Density: 1,279 Quality: 0,639 P-value: 0,000</p>	 <p>Nodes: 328 Density: 1,611 Quality: 0,870 P-value: 0,000</p>	 <p>Nodes: 303 Density: 1,861 Quality: 0,928 P-value: 0,000</p>



## KAYNAKÇA

- [1] J. Koskinen, *Software Maintenance Costs.*, 2015.
- [2] M. Page-Jones, *The practical guide to structured systems design*, New York: YOURDON Press, 1980.
- [3] M. J. H. a. P. K. A. Jefferson Offutt, «A software metric system for module coupling,» *Journal of Systems and Software*, cilt 3, no. 20, pp. 295-308, 1993.
- [4] K. H. M. J. Harald C. Gall, «Detection of Logical Coupling Based on Product Release History,» %1 içinde *ICSM '98: Proceedings of the International Conference on Software Maintenance*, 1998.
- [5] K. J. P. A. S. H. Ball T, «If Your Version Control System Could Talk,» %1 içinde *ICSE Workshop on Process Modelling and Empirical Studies of Software Engineering*, 1997.
- [6] P. W. Thomas Zimmermann, «Mining Version Histories to Guide Software Changes,» %1 içinde *ICSE '04: Proceedings of the 26th International Conference on Software Engineering*, 2004.
- [7] R. C. H. Ahmed E. Hassan, «Predicting Change Propagation in Software Systems,» %1 içinde *ICSM '04: Proceedings of the 20th IEEE International Conference on Software Maintenance*, 2004.
- [8] T. L. a. S. M. J. Sayyad-Shirabad, «Supporting Maintainance of Legacy Software with Data Mining Techniques,» 2001, pp. 22-31.
- [9] M. C. L. C. M. G. Canfora, «Using multivariate time series and association rules to detect logical change coupling: An empirical study,» %1 içinde *26th IEEE International Conference on Software Maintenance (ICSM 2010)*, Timisoara, Romania, 2010.
- [10] C. M. S. P. M. Macho, %1 içinde *IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, Suita, Japan, 2016.
- [11] M. W. E. G. H. C. G. Y. Zhou, «A Bayesian Network Based Approach for Change Coupling Prediction,» %1 içinde *WCRE 2008, Proceedings of the 15th Working Conference on Reverse Engineering*, Antwerp, Belgium, 2008.
- [12] V. F. R. G. V. Giuliano Antoniol, «Detecting groups of co-changing files in CVS repositories,» %1 içinde *Eighth International Workshop on Principles of Software Evolution (IWPSE'05)*, Lisbon, Portugal, 2005.

- [13] M. G. A. H. F. E. Richard Heidler, «relationship patterns in the 19th century: The friendship network in a German boys' school class from 1880 to 1881 revisited,» *Social Networks*, no. 37, pp. 1-13, 2014.
- [14] V. Tunalı, *Sosyal Ağ Analizine Giriş*, 1. dü., Ankara: Nobel Akademik Yayıncılık, 2016.
- [15] H. Y. A. P. Tamás Nepusz, «Detecting overlapping protein complexes in protein-protein interaction networks. *Nat. Methods*,» pp. 471-472, 2012.
- [16] P. H. Sarah Guthals, *Github for dummies*, New Jersey, Hoboken: Jhon Willey & Sons, Inc., 2019, p. 1.
- [17] D. Khokhar, *Gephi Cookbook*, Packt Publishing Ltd., 2015.
- [18] G. Su, *Instant Cytoscape Complex Network Analysis How-To*, 2013.

