

T.C.
HARRAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

DİNAMİK AĞLARDA HASTALIK YAYILMA ANALİZİ

Yiğit ALIŞAN

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ŞANLIURFA
2024

İÇİNDEKİLER

Sayfa No

ÖZET.....	ii
ABSTRACT.....	iii
TEŞEKKÜR.....	iv
ŞEKİLLER DİZİNİ.....	v
ÇİZELGELER DİZİNİ.....	vi
1. GİRİŞ.....	1
2. ÖNCEKİ ÇALIŞMALAR.....	4
2.1. Tezin Katkısı ve Organizasyonu.....	9
3. MATERYAL ve YÖNTEM.....	11
3.1. Python Dili ve Kütüphaneler.....	11
3.1.1. Temel python kütüphaneleri.....	16
3.1.2. Networkx kütüphanesi.....	18
3.1.3. Node2vec kütüphanesi.....	24
3.2. Veri Seti.....	25
3.3. Kompartıman Modeller.....	26
3.4. Çizge Teorisi ve Alt Topluluklar.....	27
3.4.1. Çizge teorisi.....	33
3.4.2. Topluluk tespiti.....	36
3.5. Çok Kriterli Karar Verme Yöntemleri.....	37
3.5.1. Topsis yöntemi.....	37
4. ARAŞTIRMA BULGULARI ve TARTIŞMA.....	42
4.1. Temel metriklerin uygulanması.....	42
4.2. Temel metriklerin TOPSIS yöntemi ile uygulanması.....	45
4.3. Louvain yöntemi ve TOPSIS yönteminin uygulanması.....	48
4.4. Node2Vec yöntemi ve TOPSIS yönteminin uygulanması.....	53
4.5. Sonuçların karşılaştırılması.....	54
5. SONUÇLAR ve ÖNERİLER.....	59
KAYNAKLAR.....	60

ÖZET

Yüksek Lisans Tezi

DİNAMİK AĞLARDA HASTALIK YAYILMA ANALİZİ

Yiğit ALIŞAN

Harran Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Nagehan İLHAN
YIL:2024, Sayfa: 67

Covid-19 hastalığının hızlı şekilde pandemi seviyesine yükselmesi nedeniyle tüm dikkatler epidemiyoloji alanına çevrilmiştir. Epidemiyoloji alanında sıklıkla kullanılan SIR temelli kompartıman tipi tahmin modellerinin bazı sınırlılıkları bulunmaktadır. Bunlardan birisi topluluğun homojen ve tüm bireylerin aynı özelliğe sahip olduklarının varsayılmasıdır. Yapılan çalışmada, topluluklarda hastalık yayılımı ile ilgili öngörü sahibi olmak amacıyla geleneksel yöntemlere alternatif olarak kullanılabilecek güncel yöntemler ele alınmaktadır. Gerçek hayatta topluluk heterojen yapıda ve bireyler farklı niteliklere sahip olduğu için ilk olarak bireylere ait temel merkezilik değerleri elde edilmiştir. Farklı bir yaklaşım olarak Louvain yöntemi ile topluluk içerisindeki alt topluluklar ortaya çıkarılarak bireylerin temel merkezilik değerleri hesaplanmıştır. Son olarak Node2Vec yöntemi ile bireylere ait öznitelik çıkarımı yapılmıştır. Elde edilen veriler çok kriterli karar destek yöntemlerinden TOPSIS yöntemine girdi olarak verilmiştir. Bireylere ait yeni özniteliklerin eklenebilmesi, değiştirilebilmesi gibi değişikliklere uyum sağlaması amacıyla TOPSIS yöntemi kullanılmıştır. Enfekte bireylerin seçimi rastgele şeklinde, bağışık bireylerin seçimi ise TOPSIS sonucu elde edilen sıralamalara göre yapılmıştır. Enfekte ve bağışık bireylerin seçimi sonrasında ağdaki hastalık yayılımı izlenerek simülasyon sonucu ortaya çıkan toplam enfekte birey sayıları hesaplanmıştır. Tüm örneklerle ait simülasyonlar sonucu elde edilen enfekte birey değerleri karşılaştırılarak analiz edilmiştir. Sonuç olarak, insan- insan etkileşimine göre oluşturulan iki veri seti için en iyi sonuç Louvain ile TOPSIS yönteminin birleşiminden elde edilmiştir. Sosyal medyadaki arkadaşlık ilişkilerine göre oluşturulan diğer veri setinde ise derece merkeziliği yöntemi başarılı olmuştur. Çalışmanın genel olarak ilgili alanda Türkçe literatüre katkı sağlayacağı ve kullanılan yöntem ve teknikler sayesinde hastalık yayılımının yavaşlatılması veya durdurulması amacıyla yapılacak önleme faaliyetlerinde (örneğin izolasyon, aşı vb.) kullanılacak olan kısıtlı imkanların en etkili şekilde uygulanması için karar vericilere, uygulayıcılara birey seçiminde yardımcı olacağı düşünülmektedir.

ANAHTAR KELİMELER: epidemiyoloji, topsis, louvain, sir, çizge

ABSTRACT

MSc Thesis

Disease Spread Analysis In Dynamic Networks Yiğit ALIŞAN

Harran University Graduate School of Computer Engineering

**Supervisor: Assist. Prof. Dr. Nagehan İLHAN
Year: 2024, Page: 67**

Due to the rapid escalation of the Covid-19 disease to pandemic levels, all attention has turned to the field of epidemiology. Traditional SIR-based compartmental prediction models frequently used in epidemiology have certain limitations. One of these limitations is the assumption of homogeneity within the community, where all individuals are assumed to have the same characteristics. In this study, alternative methods that could be used instead of traditional approaches to gain insights into disease transmission in communities are discussed. In real life, communities have a heterogeneous structure, and individuals possess different attributes. Therefore, the first step was to obtain fundamental centrality values for individuals. As a different approach, the Louvain method was used to identify sub-communities within the larger community, and the fundamental centrality values of individuals were calculated accordingly. Lastly, the Node2Vec method was employed to extract attribute features for individuals. The obtained data were input into the TOPSIS method, a multi-criteria decision support technique. The TOPSIS method was utilized to accommodate changes such as the addition or modification of new attributes related to individuals. The selection of infected individuals was done randomly, while immune individuals were selected based on the rankings obtained from the TOPSIS results. After the selection of infected and immune individuals, the spread of the disease in the network was monitored, and the total number of infected individuals resulting from the simulation was calculated. The values of infected individuals obtained from simulations for all examples were compared and analyzed. As a result, the combination of the Louvain and TOPSIS methods yielded the best results for two datasets created based on human-human interactions. In the other dataset created based on friendship relationships on social media, the degree centrality method was successful. It is believed that this study will contribute to the Turkish literature in the relevant field and provide decision-makers and implementers with assistance in individual selection for preventive activities aimed at slowing down or stopping the spread of the disease, such as isolation and vaccination, by utilizing limited resources in the most effective way possible.

KEYWORDS: epidemiology, topsis, louvain, sir, graph

TEŞEKKÜR

Tezin konusunun seçiminde, uygulamasında ve çalışmamda yardımlarını esirgemeyen ve tez çalışmamda bana sabırla yardımcı olan danışmanım Dr. Öğr. Üyesi Nagehan İLHAN hocama teşekkür ederim.



ŞEKİLLER DİZİNİ

Sayfa No

Şekil 3.1. Projeye yeni paket ekleme	13
Şekil 3.2. Paket listesi	14
Şekil 3.3. Yönsüz çizge örneği.....	20
Şekil 3.4. Yönlü çizge örneği	21
Şekil 3.5. Örnek kod bloğuna ait ekran çıktısı	22
Şekil 3.6. Örnek kod bloğuna ait ekran çıktısı	23
Şekil 3.7. Tüm veri setlerine ait ağ yapısı	25
Şekil 3.8. SIR kompartıman modeli	26
Şekil 3.9. (a) Königsberg'deki yedi köprü görseli ve (b) çizge çizimi	27
Şekil 3.10. (a) Yönsüz çizge ve (b) yönlü çizge örneği.....	28
Şekil 3.11. (a) Örnek arkadaşlık ağının (b) çizge ile gösterimi	29
Şekil 3.12. Louvain algoritmasının adımlarının görselleştirilmiş hali (Blondel ve ark. 2008).....	35
Şekil 3.13. Louvain algoritmasına ait kabakod (Aldabobi ve ark., 2022)	36
Şekil 3.14. Örnek ağ yapısı (Watabe, 1998).....	38
Şekil 4.1. Veri seti-1 üzerinde enfekte yayılım sonucu oluşan ağ yapısı	43
Şekil 4.2. Her veri seti için elde edilen enfekte birey oranı.....	44
Şekil 4.3. Veri seti-1'e ait TOPSIS sonuçlarına göre enfekte yayılım sonucu oluşan ağ yapısı	48
Şekil 4.4. Veri seti-1'e ait alt toplulukların gösterimi	49
Şekil 4.5. Veri seti-1'e ait alt toplulukların gösterimi	50
Şekil 4.6. Veri seti-1 için Louvian yöntemi sonucu oluşan enfekte, bağışık düğüm dağılımı	52
Şekil 4.7. Node2Vec yöntemi sonucu elde edilen çizge.....	54
Şekil 4.8. Simülasyonlar sonucu veri setlerine ait enfekte düğüm oranları.....	55

ÇİZELGELER DİZİNİ

Sayfa No

Çizelge 3.1 Veri setine ait örnek kesit.....	18
Çizelge 3.2. Veri setlerine temel bilgiler.....	26
Çizelge 3.3. Farklı merkezilik ölçülerinin hesaplama karmaşıklığının karşılaştırılması	31
Çizelge 3.4. Düğümlere ait merkezilik değerleri.....	39
Çizelge 3.5. Düğümlere ait sonuçlar	41
Çizelge 4.1. Merkeziyet derecelerine göre ilk 10 düğüm.....	42
Çizelge 4.2 Veri setlerine ait ilk 10 düğümün TOPSIS değeri.....	45
Çizelge 4.3. Veriseti-1 için yöntemlere göre elde edilen ilk 10 düğüm	46
Çizelge 4.4 Veriseti-2 için yöntemlere göre elde edilen ilk 10 düğüm	46
Çizelge 4.5 Veriseti-3 için yöntemlere göre elde edilen ilk 10 düğüm	47
Çizelge 4.6. TOPSIS sonuçlarına göre Veri seti-1'e ait ilk 10 düğüm	51
Çizelge 4.7. TOPSIS sonuçlarına göre Veri seti-2'ye ait ilk 10 düğüm	51
Çizelge 4.8. TOPSIS sonuçlarına göre Veri seti-3'e ait ilk 10 düğüm	52
Çizelge 4.9. Veri setleri için Node2Vec yöntemi ile elde edilen ilk 10 düğüm listesi	53
Çizelge 4.10. Veri seti-1 için tüm yöntemlerde elde edilen ilk 10 düğüm listesi	56
Çizelge 4.11. Veri seti-2 için tüm yöntemlerde elde edilen ilk 10 düğüm listesi	57
Çizelge 4.12. Veri seti-3 için tüm yöntemlerde elde edilen ilk 10 düğüm listesi	58

1. GİRİŞ

Pandemi derecesinde olan büyük hastalık olaylarında hastalık yayılımının kontrol altına alınabilmesi için en etkili yöntemlerden birisi aşıdır. Aşılar, salgını hızlı bir şekilde kontrol altına almanın güvenli ve etkili bir yolunu sunar. Normal şartlarda hastalıklara karşı geliştirilen aşılarda üretilmesi uzun yıllar alan bir prosedürdür (Chen ve ark., 2022). Fakat yeni koronavirüs SARS-CoV-2'nin neden olduğu COVID-19 hastalığının şiddetini hafifletebilmek için çok kısa sürede aşı üretimi yapılmıştır (Corum ve ark., 2020). Ancak pandeminin dünya genelini kapsaması nedeniyle yani aşı talebinin çok fazla olması aşı tedarikinin ise az olması nedeniyle az gelişmiş ülkeler başta olmak üzere birçok ülke tarafından COVID-19 aşıları sınırlı miktarda temin edilmiştir. Nüfusun çoğu enfekte olmadan önce bir salgını kontrol altına almak için aşılarda zamanında nasıl dağıtılacağı bir zorluktur. Aşılama önceliği karmaşıktır ve yaş, ırk, meslek, sağlık eşitliği, coğrafya ve politika ile iç içedir. Veriler, COVID-19'un orantısız bir şekilde komorbiditeleri olan bireyleri ve ayrıca düşük sosyo-ekonomik statüye ve yüksek sosyal savunmasızlığa sahip insanları etkilediğini göstermektedir. Önceliklendirme için pek çok kriter olabilir, örneğin: (i) enfeksiyon riski; (ii) ölüm riski; (iii) enfekte olursa bulaşma riski ve (iv) meslek, örneğin sağlık çalışanları, öğretmenler, kasiyerler vb. insanlara öncelik verilmesi konusunda hemfikir görünüyordu. (Chen ve ark., 2022). Birçok ülke bu bilinçle aşılama öncelikli gruplar belirleme yoluna giderek eldeki aşılarda etkili ve verimli şekilde kullanmak için farklı stratejiler geliştirmişlerdir. COVID-19 salgınının başlangıcından bu yana, çok sayıda matematiksel modelleme yaklaşımı, hastalık bulaşmasını kontrol etmek için dirençli halk sağlığı politikaları tasarlama açısından çok katkıda bulunmuştur (Annas ve ark., 2020; Güzey, 2021). Bununla birlikte, etkili yayıcıların etkin bir şekilde tanımlanması hala çözülmesi gereken temel problemlerden birisidir.

“Salgınlar bilimi” olarak da adlandırılan epidemiyoloji bilimi “Sağlık konusunda toplumu ilgilendiren olayların tespit edilmesi, ortaya çıkış ve yayılma nedenlerinin incelenmesi ve kontrol altına alınmasının sağlanması gibi durumlara odaklanan bilim dalı” şeklinde tanımlanabilmektedir (İşsever ve ark., 2020). Bu temel yaklaşım ile hastalık ortaya çıkması, salgın haline dönmesinin engellenmesi,

önlenmesi, yavaşlatılması, koruyucu ve önleyici tedbirlerin uygulanması gibi konularla ilgili çalışmalar gerçekleştirilmektedir. Epidemiyoloji alanında özellikle salgınlar ile ilgili yapılan çalışmalarda kullanılan yöntem ve teknikler incelendiğinde ilk olarak kompartıman tipi matematiksel modeller görülmektedir. 1927 yılında Kermack – McKendrick tarafından oluşturulan ve SIR (Susceptible- Infectious – Removed / Recovered) olarak adlandırılan model, kompartıman modellerin temel mantığını oluşturmaktadır (Kermack ve ark., 1927). SIR modeli, topluluğun kapalı, nüfusun sabit, bireylerin homojen, doğum veya ölüm (bulaşıcı hastalık haricinde) gibi durumların olmadığı, hastalığın sadece insandan insana geçtiği, bağışıklık kazanan bireyin tekrar hasta olmadığı gibi bazı varsayımlar üzerine oluşturulmuştur. Yayılma süreci biyolojik, sosyal, kimyasal ve farklı birçok alanda görülen doğal bir olgudur. Topluluk içerisinde gerçekleşen yayılma sürecinde süper yayıcılar veya etkili düğümler olarak belirtilen kişiler/ düğümler bulunmaktadır. Bu düğümlerin diğer düğümler ile temas veya iletişim oranı topluluğun geri kalanına göre oldukça fazladır.

Gelişen bilişim sistemleri sayesinde büyük ölçekli sosyal ağlara ait verilerin toplanması ve işlenmesi gün geçtikçe mümkün hale gelmektedir. Kompartıman modellerdeki matematiksel denklemleri basitleştirmek amacıyla gerçek hayat ile bağdaşmayan varsayımların önüne geçmek için bireye dayalı yaklaşımlara ve ağ modellemesine ağırlık veren çalışmalar bulunmaktadır (Bansal ve ark., 2007). Epidemiyolojik bir ağda bulunan her düğüm için yayılma kabiliyetleri hesaplanarak analiz edilebilmesi sayesinde hastalık yayılımı ile ilgili önleyici tedbirler uygulanabilir. Ayrıca, ağı daha iyi anlamak ve analiz etmek için topluluk tespiti, karmaşık ağın önemli yapısal özellikleriyle ilişkili olduğundan, ilerlemek için temel adımlardan birisidir (Rani ve Kumar, 2022b). Sosyal ağlarda topluluk tespiti, sosyal ve biyolojik bilimleri ve daha fazlasını içeren ağların yapısal özelliklerinin keşfedilmesini kolaylaştırdığından, yaygın bir çalışma alanıdır. Topluluk tespiti çalışmaları, tavsiye sistemleri, sağlık hizmetleri, siyaset, ekonomi, e-ticaret, sosyal medya, iletişim ağı gibi birçok farklı alanda kullanılmaktadır (Rani ve Kumar, 2022b).

Merkezilik, ilk olarak topluluktaki etkili düğümleri belirlemek ve ardından elde edilen topluluk yapısındaki etkili düğümlere dayalı olarak alt toplulukların her birini

genişleten birçok topluluk tespit algoritmasında önemli bir rol oynamaktadır. Geleneksel algoritmalar, ağdaki etkili düğümleri tanımak için her zaman tek bir merkezilik ölçüsü kullanır, ancak bu durumda kullanıldığında her merkezilik ölçüsünün hem artıları hem de eksileri vardır; bu nedenle, tek bir merkezilik ölçüsü kullanılarak elde edilen düğümler en iyileri olmayabilmektedir. Bu nedenle çok özellikli karar verme yöntemlerinden İdeal Çözüme Benzerliğe Göre Tercih Tekniği (Technique for Order Preference by Similarity to Ideal Solution-TOPSIS) kullanılarak ağdaki etkili düğümleri belirlemek için çeşitli merkezilik değerlerinin avantajları toplu şekilde değerlendirilebilmektedir.

Çalışmada üç farklı gerçek veri seti kullanılmaktadır. Topluluktaki düğümlerin etkililik değerlerini hesaplamak için, temel metrikler, temel metrikler ile TOPSIS yöntemlerinin birleşimi, Louvain ile TOPSIS yöntemlerinin birleşimi ve Node2vec ile TOPSIS yöntemlerinin birleşimi olmak üzere çizge teorisi temelli dört farklı yaklaşım uygulanmaktadır. Yöntemler sonucu elde edilen sıralamaya göre en iyi %10 oranında düğüm bağışık ve rastgele %10 oranında düğüm enfekte olarak seçilmektedir. Düğüm seçim işlemlerinden sonra ağ üzerinde yapılan simülasyon sonucu enfekte ve bağışık düğüm oranları elde edilmektedir. Son olarak kullanılan yöntemler, simülasyon sonucu ağda oluşan enfekte ve bağışık düğümlerin yayılımlarına göre analiz edilmektedir. Yapılan çalışmanın, hastalık yayılımının yavaşlatılması veya durdurulması amacıyla yapılacak önleme faaliyetlerinde (örneğin izolasyon, aşı vb.) kullanılacak olan kısıtlı malzemelerin en etkili şekilde uygulanması için karar vericilere ve uygulayıcılara birey seçiminde yardımcı olacağı düşünülmektedir.

2. ÖNCEKİ ÇALIŞMALAR

Epidemiyoloji alanında hastalıkların yayılması ile ilgili çalışmalarda sıklıkla tercih edilen çeşitli kompartıman modeller bulunmaktadır. Bir salgının yayılması, genellikle bir sosyal ağ grafiğinde rastgele bir SIR süreciyle modellenir (Babay ve ark., 2022). Kompartıman modellerinin temeli olarak gösterilen SIR (Kermack ve ark., 1927) kompartıman modeline ihtiyaca göre yeni kompartımanlar eklenerek (Jain ve Kumar, 2021), SEIR (Annas ve ark., 2020; He ve ark., 2020), MSIR (Menon ve ark., 2020; Rahman ve ark., 2020), MSEIR (Chen ve ark., 2021), GSIR (Singh ve Gupta, 2022) vb. yeni modeller oluşturulmuştur. Ayrıca kompartıman modellerinin bazı eksikliklerini gidermek için de çalışmalar yapılmıştır (Huo ve Zhao, 2016) (Bansal ve ark., 2007) (Isella ve ark., 2011). Popülasyon yapısının bulaşıcı hastalıklar üzerindeki etkileri ile ilgili de çeşitli çalışmalar yapılmıştır (Onofrio, 2008; Barthélemy ve ark., 2005). Bansal ve ark. (2007), epidemiyoloji alanında kullanılan kompartıman modellerinde homojen ağ modelleri yerine heterojen ağ modellerinin kullanılarak hastalık yayılım değişkenliği ile ilgili çalışmalarda bulunmuşlardır. Hastalık ile ilgili farkındalığın statik ağlarda hastalık yayılımına olan etkisi incelenmiştir (Azizi ve ark., 2020). Ek olarak kompartıman modellerin temelini oluşturan SIR ve SIS modelleri üzerinde rastgele düzenli ağlar kullanılarak ağ parametrelerinin salgın yayılımı üzerine etkileri incelenmiştir (Nadini ve ark., 2020). Olinky ve Stone (2004) tarafından yapılan çalışmada, geleneksel kompartıman tipi modellerde bulaşma oranının belirli bir eşiğin altına düşmesi sonucu hastalığın biteceği varsayımının heterojen ağlarda geçerli olmadığı ve salgın yayılımının ağların karakteristik özellikleri ile bağlantılı olduğu sonucuna varmışlardır. 4 gerçek ağda salgın yayılımını SIR modeli ile simüle ederek elde edilen sonuçlarda, önerilen yöntemin etkin ve uygulanabilir olduğunu göstermiştir. SIR modelinde, homojen senaryolara kıyasla heterojen temas kalıplarının çok çeşitli parametre değerleri için daha erken ve daha büyük salgınlara neden olduğunu, bazı parametre kombinasyonlarında daha küçük salgınlara meydana gelebileceği belirtilmiştir (Rocha ve Blondel, 2013).

“Ağlar, popülasyonların bağlantı örüntülerini temsil etmek ve hastalığın bulaşma yönlerini tanımlamak için kullanılabilir” (Diekmann ve ark., 1998). İlk defa

John Snow tarafından kolera vakalarının Londra şehir haritası üzerinde çizilerek gösterilmesinden itibaren, grafikler ve görselleştirmeler epidemiyoloji alanında iletişimi, kümelemeyi, analizi, hipotez testi ve karar verme gibi konularda önemli rol oynamaya başlamıştır (Koch ve Denike, 2009; Karlsson ve ark., 2013; Carroll ve ark., 2014). Alanında üretilen birçok başarılı çalışma sayesinde çizge teorisinin kullanım alanı artmıştır. Ağlar, düğüm ve kenar sayılarının yanı sıra düğümler arası ilişkilere göre de farklı karakteristik özellikler gösterebilmektedir. Düğümlerin ağlarda bulundukları yere göre değerlerini belirlemek için farklı çalışmalar yapılmıştır (Borgatti ve Everett, 2006; Katherine, 1997; Freeman ve ark., 1979; Zhang ve Luo, 2017). Wang ve ark. (2017), efficiency centrality (EffC) isimli yeni bir merkezilik ölçüsü önermiştir. Bir düğümün ağ üzerindeki göreceli etkisini ölçmek için Katz (1953), Katz merkeziliğini uygulamıştır. Ahmad ve ark., (2023), akıllı yeşil şehirlerin temelinde kullanılan IoT teknolojilerindeki güvenliği sağlamak için merkezilik değerleri yardımıyla her cihazın ağdaki etkililik değerini hesaplayabilecek “Değiştirilmiş Dinamik Ağırlıklı Toplam Yöntemi” adlı yeni bir çerçeve yöntem önermişlerdir. Yazarlar, karmaşık ağlarda düğümlerin etkililik değerlerinin hesaplanması için çoklu merkezilik ölçütlerinin birleştirildiği ve çıktıların AHP kullanılarak değerlendirildiği yeni ve hızlı bir yaklaşım önermişlerdir (Sabah ve Şimşek, 2023). Li ve ark., (2023), mevcutta kullanılan merkezilik değerleri üzerinde direnç mesafesi kullanarak dirençli Derece Merkeziliği, dirençli Özvektör Merkeziliği, dirençli Harmonik Merkeziliği ve dirençli Dışmerkezlik Merkeziliği de dahil olacak şekilde bir dirençli merkezilikler grubu oluşturmuşlardır. Yazarlar, ağdaki düğümlerin küresel niteliklerinden ziyade yerel niteliklerinin kullanımına dayalı Yerelliğe Dayalı Yapı Sistemi (LSS) adlı yeni bir buluşsal yöntem önermektedir. Yöntemlerinin etkili düğümlerin belirlenmesinde başarılı olduğunu belirtmişlerdir (Ullah ve ark., 2023). Evans ve Chen (2022), derece ve yakınlık merkezilik ölçülerini kullanarak yeni bir ölçüt geliştirmişlerdir. Yaptıkları çalışmada en kısa yol ağacı yaklaşımına dayanarak, yakınlığın tersinin derecenin logaritmasına doğrusal olarak bağlı olduğunu gösteren analitik bir türev önermişlerdir. Yöntemi stokastik ağ modellerinden üretilen ağlar ile 130 gerçek dünya veri setinden üretilen ağlar kullanarak test etmişlerdir. Yapılan test çalışmaları sonucunda çoğu ağın kök düğümlerinden iki veya daha fazla adım uzaklıkta istatistiksel olarak benzer olan en kısa yolu kapsayan ağaçlarla tahmin edilebileceğini belirtmişlerdir. Yazarlar FIP adını verdikleri bir yöntem ile karmaşık

ağlarda etki düzeyi yüksek düğümlerin daha az alanda aramayı amaçlamakta ve bu şekilde algoritmanın zamansal karmaşıklık değerini düşürmeyi hedeflemişlerdir. FIP algoritması kısaca ilk olarak örtüşen toplulukları tespit ederek düğümler arası ilişkileri analiz etmektedir. Bu aşamada arama alanını daraltmak için önemsiz topluluklar kaldırılır. Daha sonra elde edilen aday düğümler, küresel yayılma olasılığının etkisi kullanılarak oluşturulmaktadır. Son aşamada, önemli düğümlerin rolü ve topluluklardaki örtüşen düğümlerin yayılma etkisi nihai düğümleri seçmek için ölçülmektedir. Yazarların gerçek dünya ağları ve yapay ağlarda yaptıkları ölçüm sonuçlarına göre önerilen algoritma diğer algoritmalarından daha başarılı sonuç elde etmiştir (Bouyer ve ark., 2023). Mahadevi ve ark., (2022) yapmış oldukları çalışmada GEC (Grafik Enerji Merkeziliği- Graph Energy Centrality) adı verilen ve $O(n^3)$ zaman karmaşıklığına sahip yeni bir merkezilik metriği geliştirmişlerdir. Bu merkezilik metriğinin temeli, köşe elemelerinden kaynaklanan grafik enerjisine dayalı olarak her köşeye bir merkezilik değeri verilmektedir. Her tepe noktasına ait GEC değerinin döngü grafikleri için asimptotik olarak, tam grafikler için ise tam olarak ikiye eşit olduğunu göstermektedir. Ayrıca, yıldız grafiklerinin sadece iki GEC değeri ile sıralanabileceğini, yol grafiklerinin ise maksimum değerler kullanılarak sıralanabileceğini belirtmişlerdir.

Merkezilik değerleri hesaplama yöntemlerinin, düğümlerin sahip olduğu özellikleri yeterli şekilde kapsayamaması nedeniyle bu yöntemlerin topluca dikkate alınabileceği yöntemler kullanılmaya başlanmıştır. Bu nedenle “Çok kriterli karar verme yöntemleri” ile merkezilik değerlerinin bir arada kullanılması popülerleşmiştir. Liu ve ark. (2022) yapmış oldukları çalışmada, ekspres teslimat ağlarında verimliliği arttırmak ve maliyeti düşürebilmek için geliştirmiş oldukları matematiksel yöntemler ile hizmet noktalarının ağdaki konumuna göre değerlerini belirlemekte ve TOPSIS yöntemi yardımıyla sıralamaktadırlar. Xu ve Jie (2023), karayollarında anahtar kavşakların belirlenmesinde kullanılan geleneksel yöntemler yerine TOPSIS yönteminin kullanılmasını önermişlerdir. Ayrıca TOPSIS yönteminde kullanılan Öklid mesafesinin yerine ağırlıklı Mahalanobis mesafesi kullanarak doğruluk oranını arttırmışlardır. Nguyen ve Woo (2022) tarafından yapılan çalışmada, konteyner taşımacılığı için önemli olan limanların bölgesel olarak oluşturdukları ağ içerisindeki önemlerini belirlemek için bazı kriterler belirleyerek TOPSIS yöntemi yardımıyla

limanların önem sıralamasını gerçekleştirmişlerdir. Yazarlar önermiş oldukları yöntemin, yazılımlarda bulunan ölü kodların tespit edilmesi için ağırlıklandırılmış TOPSIS yardımıyla ölü kodları silme riskini iyi değerlendirebildiğini belirtmişlerdir (Liu ve Tao, 2022). Liu ve ark. (2023) tarafından yapılan çalışmada, ağdaki etkili düğümlerin belirlenmesi için kullanılan K-shell yöntemini optimize ederek ağlara ait merkezilik değerleri ile TOPSIS yöntemini kullanarak başarılı sonuçlar üretmişlerdir. Lu ve ark. (2023), protein-protein etkileşim ağlarında temel proteinleri belirleyebilmek için yeni merkezilik değerleri önermişlerdir. Ağdaki proteinleri etkililiklerine göre sıralamak için önerilen metrikler ile elde edilen değerleri TOPSIS ile önceliklendirmişlerdir. Chang ve ark. (2023), işbirlikçi görev yeteneğine yönelik çok platformlu aviyonik mimarisi için PageRank ve bazı merkezilik değerleri ile TOPSIS yöntemini kullanarak yeni bir düğüm etkililiği değerlendirme yöntemi önermişlerdir. Ishfaq ve ark. (2022) tarafından yapılan çalışmada, sosyal ağlardaki düğümlerin etkililik derecelerini belirlemek için mevcut merkezilik değerlerinin kombinasyonu olan bir yöntem önermişlerdir. Önerilen yöntem ile elde edilen sonuçların TOPSIS yönteminde kullanılarak düğümlere ait elde edilen sıralamanın mevcut merkezilik değerlerinden daha iyi olduğunu belirtmişlerdir. Rani ve Kumar (2022a) tarafından yapılan çalışmada, gerçek ağ yapısına uygun olan heterojen ağlarda merkezilik değerleri yerine makine öğrenmesi yardımıyla öznitelikler çıkartılmakta ve TOPSIS, bulanık AHP gibi yöntemler ile etkili düğümlerin sıralaması gerçekleştirilmiştir. Son olarak önerilen yöntemin performansı ile en son teknolojilerin performans karşılaştırılması yapılmıştır.

Topluluk tespiti, grup içi ilişkilerin gruplar arası ilişkilerden daha fazla olduğu bir çizgedeki düğüm gruplarını keşfetmeye yarayan çizge alanında en çok çalışılan konulardan birisidir (Wang ve ark., 2015; Li ve ark., 2020). (Rocco ve ark., 2022) çeşitli topluluk tespit algoritmalarını karşılaştırarak bazı sonuçlara ulaşmışlardır. Amaç hangi algoritmaların hangi tip ağlarda daha iyi sonuçlar üretebileceği belirlemektir. Karşılaştırma için elektrik güç sistemleri ile ilgili toplamda 16 farklı yapı kullanılmıştır. Sonuç olarak elde edilen çıktılar kullanılarak algoritmaların etkili olduğu ağ grupları ve karakteristik özellikleri kullanılarak genel bir grupta yapılmıştır. Wang ve ark. (2023) LCDPC adlı üç aşamalı bir yerel genişletme algoritması önermişlerdir. Bu çalışmada önerilen LCDPC algoritması ile topluluk

keşfine dayalı yerel toplulukların bulunması hedeflenmiştir. İlk aşamada, tohum olarak verilen düğümü değiştirebilmek için düğüm önemi ve düğüm benzerliği hesaplanarak uygun bir düğüm aranmaktadır. İkinci aşamada, tohum en uygun potansiyel topluluğa katılmakta ve böylece ilk topluluk oluşturulmaktadır. Son aşamada ise uygun görülen düğümler, potansiyel olarak görülen topluluklar ile tohumlar arası benzerlikler hesaplanarak, karşılaştırılarak topluluk genişlemesi için topluluğa dahil edilmektedir. Önermiş oldukları algoritma literatürde bulunan 8 farklı algoritma ile karşılaştırılmıştır. Sonuç olarak potansiyel topluluk keşfi ve ağların topluluk yapısını belirlemeye yönelik çalışmalarda diğer algoritmalarından daha başarılı olduklarını belirtmişlerdir. Rai ve ark. (2023) yapmış oldukları çalışmada bir sosyal içerisinde bulunan potansiyel toplulukların tespiti için TOPSIS ve k-means yöntemlerini kullanmışlardır. TOPSIS'e girdi olarak verilen merkezîyet ölçüm değerleri sayesinde ağ içerisinde etkili olan potansiyel tohumlama düğümlerinin listesini çıkartmışlardır. TOPSIS tarafından üretilen düğüm sıralamasında üst sıralarda olan düğümler seçilerek k-means algoritmasına girdi olarak verilmiştir. K-means algoritması tarafından elde edilen sonuçlar Yasca, LICOD vb. diğer tohum merkezli algoritmalar ile karşılaştırıldığında mevcut yöntemlerinin daha başarılı olduğu sonucuna ulaşmışlardır. Liu ve ark. (2023), Physical Review E (PRE) dergisinde yayınlanan makaleleri kullanarak akademik toplulukların etkisini sosyal ağ analizi ve TOPSIS yöntemi ile bulmayı amaçlamışlardır. Yapmış oldukları bu çalışmada ilk olarak makaleler arasındaki atıf ilişkisini dikkate alarak ortak atıf ağı kurulmaktadır. Daha sonra akademik topluluğun etkisini değerlendirmek amacıyla indeks sistemi kullanılmıştır. Sonraki aşamada ise etkiyi değerlendirmek amacıyla TOPSIS yöntemi kullanılmıştır. Yapmış oldukları çalışmada önerdikleri yöntemin, akademik grupların etkilerini analiz etme konusunda yeni bir bakış açısı sağlamakta ve akademik grupların etkisini ayrıntılı şekilde tanımlayabileceğini belirtmişlerdir. Chaharborj ve ark. (2022) yeni bir ağırlık hesaplama yöntemi geliştirerek ve TOPSIS yöntemine uygulayarak düğümlerin önem sırasına göre sıralanmasını gerçekleştirmişlerdir. Yapılan çalışmada düğümlere ait temel merkezîyet değerleri bulunarak TOPSIS yöntemine girdi olarak verilmektedir. Bu aşamada her metrik için önerdikleri ağırlık hesaplama yöntemi ile ağırlıklar belirlenmektedir. Çıktı sonucu elde edilen sonuçlar eşit ağırlıklı TOPSIS ve düğümlerin elde etmiş olduğu metriklerdeki sıralamasına göre karşılaştırılmıştır.

Sonuç olarak önerdikleri yöntemin eşit ağırlıklı TOPSIS yönteminden daha yararlı olduğunu belirtmişlerdir.

2.1. Tezin Katkısı ve Organizasyonu

Bu çalışmanın katkıları:

1. Epidemiyoloji alanında geleneksel ve güncel yaklaşımların neler olduğu,
2. Kompartıman tipi matematiksel modellerin yapısı ve uygulaması,
3. Epidemiyoloji alanında çizge teorisinin uygulanması,
4. Gerçek ağlar ve alt topluluklarının belirlenmesinde kullanılan yöntemler,
5. Gerçek ağlardaki düğümlerin önemini gösteren temel merkezîyet dereceleri ve metriklerin gelişimi,
6. Çok kriterli karar destek sistemi olan TOPSIS yönteminin kullanılarak etkili düğümlerin nasıl tespit edilebileceği,
7. Güncel yaklaşımlardan “Node2Vec” yönteminin incelenmesi ve örnek uygulaması ile ilgili bir kılavuz oluşturulmuş olup,

Önerilen yaklaşım kullanılarak:

1. Karmaşık modeller yerine, uygulanabilirlik bakımından temel merkezîyet metrikleri kullanılarak verimli şekilde ağdaki etkili yayıcıların tanımlanabilmesi,
2. Ağ içerisinde düğüm etkililiğinin belirlenmesinde tek bir özneteliğin dikkate alınması yerine istenilen özneteliklerin dikkate alınabilmesi,
3. Çok kriterli karar destek sisteminin kullanılması sayesinde merkezîyet metrikleri yerine ya da merkezîyet metrikleri ile önemli görülen farklı metriklerin de kolaylıkla yönteme eklenebilmesi,
4. Elde edilen sonuçların karar vericiler tarafından dikkate alınması durumunda karar vericilerin salgını yavaşlatmak veya durdurmak için yapacakları harcamaları azaltmanın yanı sıra, kısıtlı olan imkanlarında en etkili şekillerde kullanılabilmesi,

gibi konularda kolaylık sağlayacak ve Türkçe literatüre pozitif katkı sunacaktır.

Bu tez dokümanı şöyle düzenlenmiştir: Birinci bölümde giriş yapılarak epidemiyoloji alanından kısaca bahsedilmekte, geçmişten günümüze kullanılan öne çıkmış yöntemler tanıtılmaktadır. İkinci bölümde, literatür taraması yapılarak geçmişten günümüze yapılan önemli çalışmalardan bahsedilmekte ve güncel yöntemlere örneklerle değinilmektedir. Üçüncü bölümde kullanılan materyal ve metotlar örneklerle açıklanmıştır. Dördüncü bölümde, kullanılan yaklaşımlara ait simülasyonlar ve çıktılar paylaşılmaktadır. Son bölümde ise sonuçlar ve öneriler hakkında bilgiler sunulmaktadır.



3. MATERYAL ve YÖNTEM

3.1. Python Dili ve Kütüphaneler

1991 yılında Guidio Van Rossum tarafından yazılan ve daha sonra ise Python Yazılım Vakfı tarafından sürekliliği sağlanan Python dili, “C”, “Java”, “ALGOL 68” gibi dillerden etkilenecek şekilde ortaya çıkmıştır. Python açık kaynak kodlu olması sebebiyle ihtiyaç duyan herkes tarafından ekleme ve çıkarma yapılabilen bir dildir. Ayrıca platform bağımsız ve yüksek seviyeli, nesne yönelimli olma özelliklerine sahiptir. Kullanıcılar tarafından tercih edilmesinin en önemli etkenlerinden birisi okunabilirlik özelliğinin iyi olmasıdır. Kullanıcıların başkaları tarafından yazılan komutları kolayca anlayabilmesi ve üzerinde düzenlemeler yapabilmesi durumuna okunabilirlik denilmektedir. Bu nedenle okunabilirlik yazılım yaşam döngüsünde önemli bir yere sahiptir. Aşağıdaki örnekte Python, Java ve C programla dillerine ait kod blokları görülmektedir. 3. bölümde amacı “Çizge Teorisi” ifadesini ekrana yazdırmaktır. Okunabilirlik açısından Python dilinin diğer popüler dillere göre avantajı bu örnekte görülmektedir.

Python diline ait kod bloğu:

```
print(" Çizge Teorisi.")
```

Java diline ait kod bloğu:

```
public class MerhabaDunya
{
    public static void main(String[] args)
    {
        System.out.println("Çizge Teorisi");
    }
}
```

C diline ait kod bloğu:

```
#include <stdio.h>
int main(void)
{
    printf("Çizge Teorisi.");
}
```

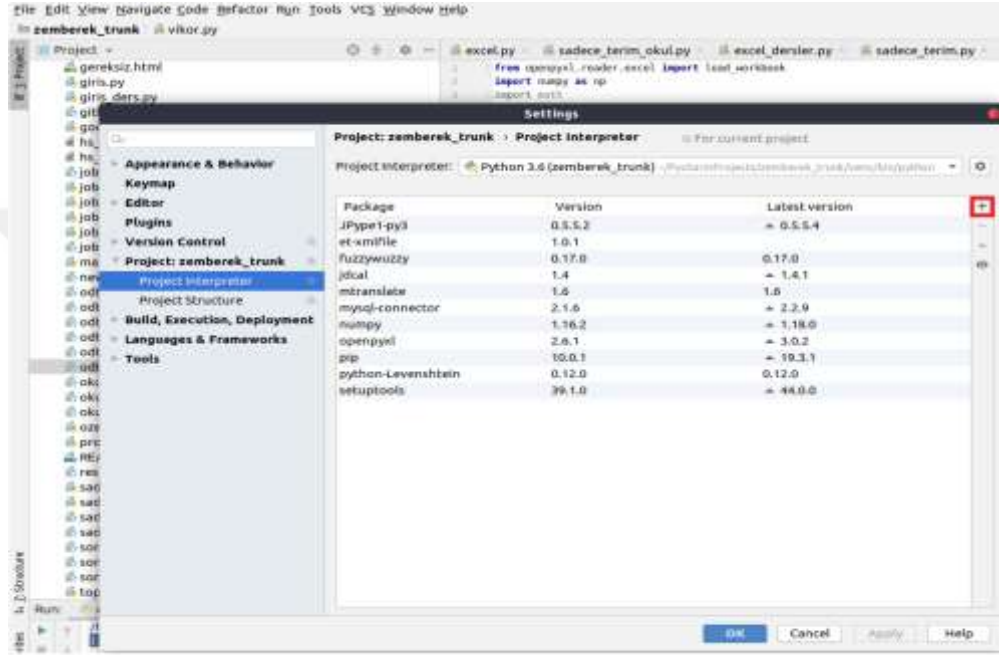
Birçok yüksek seviyeli programlama dili çalışma esnasında yazılan komutları ilk olarak derleyici denilen programlara girdi olarak vermektedirler. Burada yazılan

komutlar bilgisayarın anlayabileceği komutlara çevrilmekte ve daha sonra bilgisayar tarafından işlenmektedir. Bilgisayar tarafından işlenen komutlar kullanıcıya verilmeden önce de aynı işlemin tersi uygulanmaktadır. Bu çalışma mantığına sahip programlama dillerine derlenen diller denilmektedir. Derlenen diller için özel derleyiciler gerekmekte ve bu nedenle de platform bağımlı hale gelmektedirler. Python ise yorumlanan bir programlama dilidir. Yani komutlar çalışma anında okunmakta ve sonrasında işlenmektedir.

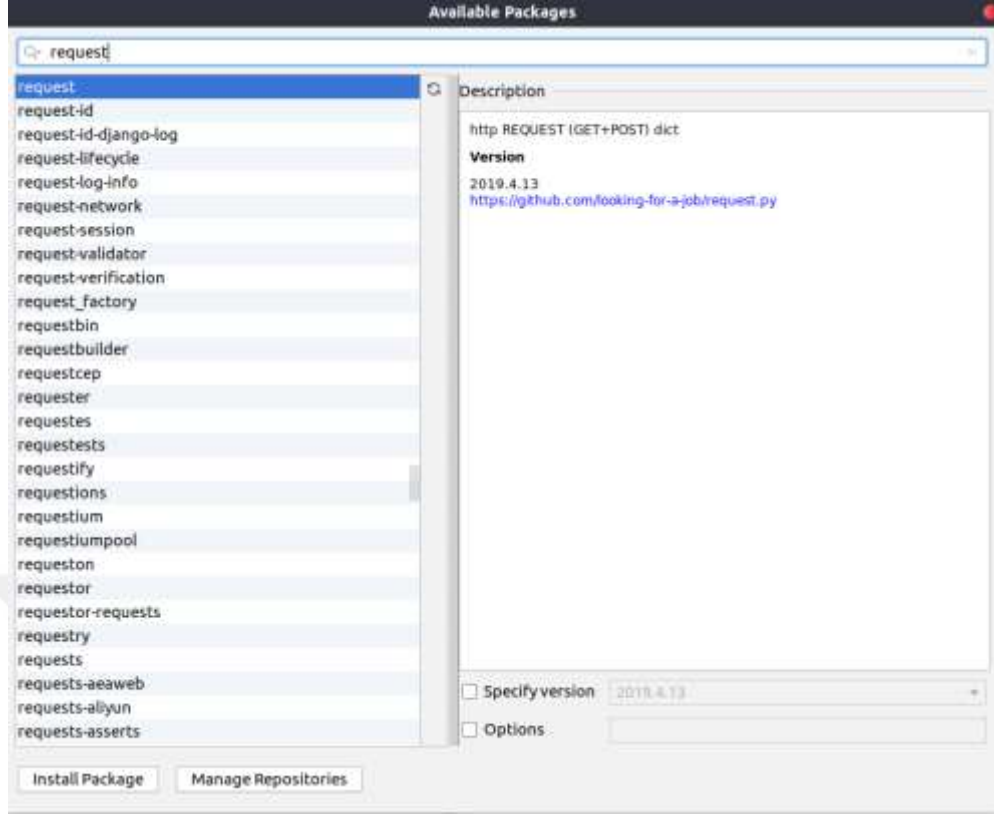
Python, en basit işlemlerden, üst düzey işlemlere kadar kullanılabilirliği kolay olan kütüphaneler sunmaktadır. Ayrıca ihtiyaç halinde “Matlab”, “R” veya “C” gibi dillerle bağlantılı olarak çalışabilmektedir. Python açık kaynak kodlu ve GPL (Genel Kamu Lisansı - General Public Licence) destekli Python Yazılım Vakfı Lisansına sahip olduğu için kullanıcılar, ücretsiz şekilde edindikleri Python’ı kullanarak oluşturdukları programları, ticari amaçlı lisanslayarak dağıtım yapabilme imkanı sunmaktadır (Tiobe, 2023).

TIOBE adlı internet sayfası arama motorlarından elde ettiği arama verilerini işleyerek en sık araması yapılan programlama dillerinin sıralamasını çıkarmıştır. Yapılan sıralama sonucunda Python programlama dili en etkili programlama dili olarak yer almaktadır. Google tarafından yapılan araştırmalarda ise programlama dilleri arasında en popüler olan dil olarak Python programlama dili gösterilmektedir. Programlama dillerini kullanan kullanıcılar tümleşik geliştirme ortamı (Integrated Development Environment- IDE) adı verilen programlardan faydalanmaktadırlar. Kısaca IDE olarak adlandırılan programlar programlama dillerine göre özelliklere sahiptir. Otomatik komut tamamlama, fonksiyon önerme, yazım yanlışlarını otomatik olarak düzeltme ve Github ile tam entegrasyon gibi özellikler örnek olarak sayılabilir. Python programlama dilinin kullanımında en sık tercih edilen IDE’lerden birisi Pycharm adı verilen programdır. Pycharm ücretli ve ücretsiz sürümlerine sahip JetBrains adlı firma tarafından geliştirilen bir IDE’dir. Pycharm ile hızlı şekilde yeni bir proje oluşturulabilmektedir. Ayrıca proje içerisinde istenilen kütüphanelerin eklenebileceği özellikte mevcuttur. Şekil 3.1.’de gösterildiği gibi proje içerisine yeni bir kütüphane kurabilmek için ilk olarak ayarlar sekmesinden “Project Interpreter”

butonuna tıklanmalıdır. Daha sonra açılan pencerede “+” işaretine tıklanarak kütüphane arama penceresi açılmaktadır. Buradan istenilen kütüphanenin adı yazılarak (Şekil 3.2.) aranabilmekte ve “Install Package” butonuna tıklanarak kurulum yapılabilmektedir. Ayrıca aynı ekranda mevcut proje içerisinde kurulu olan kütüphanelere ait liste de mevcuttur.



Şekil 3.1. Projeye yeni paket ekleme



Şekil 3.2. Paket listesi

Değişkenler, karakter ve/veya sayısal değerlerin (kayan noktalı tam, kayan noktasız ondalıklı veya karmaşık sayılar) program içerisinde bir ya da daha fazla defa kullanılması amacıyla saklamaya yarayan veri yapılarıdır. Python programlama dilinde değişken tanımlamaları kolay şekilde yapılabilir. Fakat değişkenlerin isimlendirmesinde bazı kurallar bulunmaktadır. Bu kurallardan ilki diğer birçok programlama dillerinde de olduğu gibi özel ifadelerin değişken ismi olarak kullanılamamasıdır. Örnek olarak Python programlama dilinde kullanılan `break`, `try`, `continue`, `class`, `with`, `pass`, `False`, `True`, `global`, `not`, `finally`, `assert`, `return`, `not`, `if`, `elif` gibi kelimeler gösterilebilir. Ayrıca değişken isimleri sayısal ifadeler ile başlamamalıdır. Ayrıca aşağıdaki örnekte görüleceği gibi değişken isimlendirmede büyük küçük harfe duyarlı “case sensitive” bir sözdizim yapısı vardır.

Örnek;

```
kelime_1 = "Çizge"
Kelime_1 = 2023
print(kelime_1)
# Çizge           #çıktı
print(Kelime_1)
# 2023           #çıktı
```

Program yazarken veriler üzerinde işlem yapmamız gerekmektedir. Böyle durumlarda kullanılan ya da kullanılacak verileri saklamak için değişkenler kullanılmaktadır. Değişkenler muhafaza ettikleri verilere göre değişmektedir. Örnek olarak “string” adı ile tanımlanan karakter ifadelerinin kullanılma şekli aşağıdaki örnekte gösterilmektedir.

Örnek karakter ifadesi kullanımı ve çıktısı:

```
x1 = 'Çizge Teorisi'
print(x1)
# Çizge Teorisi # çıktı
x2 = "Çizge Teorisi "
print(x2)
# Çizge Teorisi # çıktı
x3 = '''Çizge Teorisi '''
print(x3)
# Çizge Teorisi # çıktı
```

Ayrıca karakter ifadeleri kullanırken matematiksel operatörlerden de yararlanılabilmektedir. Aşağıdaki örnekte görüleceği gibi artı işareti (+) kullanılarak iki farklı kelime birleştirilebilmektedir.

Örnek:

```
print('Çizge' + ' ' + 'Teorisi')
# Çizge Teorisi # çıktı
```

Python kullanılarak program yazılması esnasında değişken kullanılacağı zaman değişkenlerin türlerini belirtmek gibi bir zorunluluk bulunmamaktadır. Tanımlama yapılmadan kullanılan değişkenlerin hangi veri tipinde olduğu “type()” komutu ile öğrenilebilir.

Örnek;

```
kelime_1 = "Çizge"
Kelime_1 = 2023
print(type(kelime_1))
# <class 'str'> # çıktı
print(type(Kelime_1))
# <class 'int'> # çıktı
```

Yüksek seviyeli dillerde bulunan integer, float, string gibi temel veri tiplerine (ilkel veri tipleri- primitive data type) ek olarak Python programlama diline özel veri tipleri de bulunmaktadır. Nesne yönelimli programlama dillerinde bulunan “if” şart deyimi Python programlama dilinde de bulunmaktadır. “if”, “if-elif”, “if-elif-else”

yada “if-else” kalıpları şeklinde kullanımı mevcuttur. Bu kod bloklarında bulunan “elif” komutu “else” komutuna benzetilmektedir. Aynı işi tekrarlı şekilde yapmaya yarayan döngü komutlarından birçoğu Python programlama dilinde kullanılmaktadır. En çok kullanılan döngü komutlarından “for” döngü bloğunun kullanımı aşağıdaki örnekte gösterilmektedir.

Örnek:

```
x1='Çizge Teorisi'
for i in range(0,5,1):
    print(x1)

'''
Çizge Teorisi'
Çizge Teorisi'
Çizge Teorisi'
Çizge Teorisi'
Çizge Teorisi'
''' #çıkıktı
```

While döngü komutu, tanımlanan şartın sağlandığı durum boyunca blok içerisine yazılan komutlar tekrar etmektedir. Şartın sağlanma durumu değiştiği zaman kod bloğundan çıkılarak sıra sonraki komutların çalıştırılmasına gelmektedir.

Örnek:

```
x1='Çizge Teorisi'
sayac = 5
while sayac > 1 :
    print(x1)
    sayac -= 1

'''
Çizge Teorisi'
Çizge Teorisi'
Çizge Teorisi'
Çizge Teorisi'
Çizge Teorisi'
''' #çıkıktı
```

3.1.1. Temel kütüphaneler

Matplotlib kütüphanesi, Python dilinde veri görselleştirme amacıyla kullanılan açık kaynak kodlu kütüphanelerden birisidir. Basit bir çizgi grafiklerinden karmaşık üç boyutlu grafiklere kadar birçok çeşitli grafik türünü desteklemektedir. Matplotlib kütüphanesi temel olarak pyplot modülü ve nesne tabanlı arayüz olmak üzere iki farklı bileşenden oluşmaktadır. Kullanıcılar oluşturulan grafik üzerine Pyplot modülü ile

çizgi, nokta, çubuk gibi grafik öğeleri ekleyebilmektedir. Nesne tabanlı arayüz ise genel olarak daha karmaşık grafikler oluşturmak için kullanılmaktadır.

Yapılan çalışmada “matplotlib” kütüphanesi bağışık ve enfekte bireylerin topluluktaki hastalık yayılımlarının görselleştirilmesi aşamasında kullanılmıştır. Kullanılan kod bloğu aşağıda gösterilmektedir.

```
def cizim(df,nm):
    g = nx.from_pandas_edgelist(df,"v1","v2",edge_attr="duration")
    color_map = []
    for node in g:
        x = df[(df['v1'] == node) & (df['enfekte'] == 1)]
        if not len(x) == 0:
            color_map.append('red')
        else:
            color_map.append('green')
    durations = [i['duration'] for i in dict(g.edges).values()]
    labels = {i:i for i in dict(g.nodes).keys()}
    fig, ax = plt.subplots(figsize=(100,100))
    pos = nx.spring_layout(g)
    nx.draw_networkx_nodes(g, pos, ax = ax,node_color=color_map)
    nx.draw_networkx_edges(g, pos, width=durations, ax=ax)
    _ = nx.draw_networkx_labels(g, pos, labels, ax=ax)
    plt.axis('off')
    plt.savefig(nm + ".png")
```

Pandas kütüphanesi, Python programlama dilinde veri analizi ve işleme işlemleri için kullanılan temel kütüphanelerden birisidir. 2008 yılından itibaren geliştirilen Pandas kütüphanesinde temel amaç alanlara özgü istatistiksel hesaplamaların toplu şekilde bir kütüphane üzerinden yapılmasını sağlamaktır. Ayrıca işlem yapılırken veri tabanı dillerinin sahip olduğu esnekliğin entegre edilmesi ile mevcut veri analiz araçlarındaki boşlukların doldurulması hedeflenmiştir. Kütüphanenin adı istatistik ve ekonometride karşılaşılan çok boyutlu veri kümeleri için ortak bir terim olan panel veriden gelmektedir (McKinney, 2011). Kütüphanenin kullanılabilmesi için ilk olarak çalışma ortamına yüklenmesi gerekmektedir. Yükleme işleminden sonra xlsx, csv, txt vb. birçok farklı dosya yapısını otomatik olarak okuyabilmekte ve veri yapısına göre “seri” ya da “dataframe” yapısına çevirebilmektedir.

```
import pandas as pd
data = pd.read_csv('infect.data', delimiter = " ")
print(df.head())
```

Yukarıdaki örnekte çalışmada kullanılan veri setinin yüklenmesi için kullanılan kod bloğu gösterilmektedir. Elde edilen ekran çıktısı Çizelge 3.1.’de gösterilmektedir.

Çizelge 3.1 Veri setine ait örnek kesit

v1	v2	süre	etkilesim_zamani
100	101	5	1247658439
100	101	2	1247660019
100	102	3	1247658579
100	103	3	1247658159
100	103	4	1247658179
100	103	1	1247658419

Numpy (Numerical Python), Python programlama dili kullanırken hızlı şekilde bilimsel hesaplamalar yapabilme imkânı sağlayan matematik kütüphanelerinden birisidir. Dizi yapısında olan Numpy’de bulunan tüm elemanlar aynı tipte olmak zorundadır. Ayrıca boyutları en başta belirlenmek zorundadır. Bu iki zorunluluk sayesinde Numpy kütüphanesi ile “list” veri tipinde işlem yapmaktan çok daha hızlı şekilde sonuçlar elde edilebilmektedir. Python tarafından desteklenen kütüphaneler arasında olmasından dolayı birçok kütüphaneye kolay şekilde eklenebilme özelliğine sahiptir.

3.1.2. Networkx kütüphanesi

Karmaşık ağların yapısı, dinamikleri ve işlevlerinin oluşturulması, manipülasyonu ve incelenmesi amacıyla Python dili ile yazılmış bir kütüphanedir. İlk sürümü 2005 yılı Nisan ayında çıkmıştır. NetworkX kütüphanesi açık kaynak kodlu ve herkes tarafından geliştirilebilir bir kütüphane olması nedeniyle birçok kapsamlı özelliğe ve çizge alanında kullanılan algoritmalara sahiptir. Bu özellikleri sayesinde uzman olmayan kullanıcıların dahi bu kütüphane ile orta seviye çizge işlemlerini gerçekleştirebilmesini sağlamaktadır (Hagberg ve Conway 2020). NetworkX ile örnek bir ağ yapısı oluşturmak için öncelikle kütüphanenin projeye yüklenmesi gerekmektedir. Sonraki aşamada aşağıdaki örnekte gösterilen komutlar yardımıyla örnek bir çizge yapısı elde edilebilmektedir:

Aşağıdaki komut sayesinde NetworkX kütüphanesi projeye yüklenmektedir.

```
import networkx as nx
```

Aşağıdaki komut sayesinde yeni bir ağ oluşturulmaktadır.

```
G = nx.Graph()
```

Aşağıdaki komut sayesinde G ismiyle oluşturulan ağın içerisine A, B ve C isminde 3 adet düğüm eklenmektedir.

```
G.add_node("A")
G.add_node("B")
G.add_node("C")
```

Kütüphaneye ait “edge()” komutu kullanılarak G adlı ağın içerisinde bulunan düğümler arasında aşağıdaki örnekte görüldüğü kenarlar oluşturulmakta, yani birbirlerine bağlanmaktadır.

```
G.add_edge("A", "B")
G.add_edge("B", "C")
G.add_edge("C", "A")
```

Oluşturulan çizgelara ait temel bilgileri elde etmek ve çizge üzerinde matematiksel işlemler yapmak için aşağıdaki örneklerde kullanılan temel komutlardan yararlanılabilmektedir.

“G.number_of_nodes()” komutu ile G çizgeinde bulunan düğümlerin toplam sayısına ulaşılmaktadır. Daha sonra “print” komutu yardımıyla elde edilen sonuç ekrana yazdırılmaktadır.

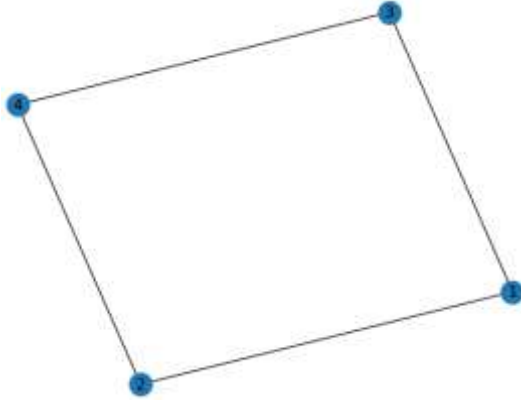
```
print ("Düğüm sayısı:", G.number_of_nodes())
```

“G.number_of_edges ()” komutu ile G çizgesinde bulunan kenarların toplam sayısına ulaşılmaktadır. Daha sonra “print” komutu yardımıyla elde edilen sonuç ekrana yazdırılmaktadır.

```
print ("Kenar sayısı:", G.number_of_edges())
```

“nx.shortest_path_length(G, "A", "C")” komutu ile G çizgesinde bulunan A ve C düğümleri arasındaki en kısa kenarın uzunluğuna ulaşılmaktadır. Daha sonra “print” komutu yardımıyla elde edilen sonuç ekrana yazdırılmaktadır.

```
print ("A ve C arasındaki en kısa yol uzunluğu:",
nx.shortest_path_length(G, "A", "C"))
```



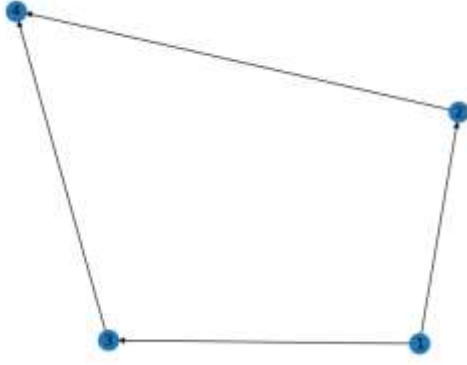
Şekil 3.3. Yönsüz çizge örneği

Yönsüz çizge, herhangi bir düğümün herhangi bir diğer düğüme doğru bir yönelimi olmayan bir çizge yapısıdır. Aşağıdaki örnekte, "nx.Graph()" fonksiyonu ile yönsüz bir çizge (Şekil 3.3.) yapısı oluşturulmaktadır:

```
import networkx as nx
# Yönsüz çizge yapısı oluşturulur
G = nx.Graph()
# Düğümler ve kenarlar eklenir
G.add_nodes_from([1, 2, 3, 4])
G.add_edges_from([(1, 2), (1, 3), (2, 4), (3, 4)])
# Çizge yapısı görselleştirilir
nx.draw(G, with_labels=True)
```

Yönlü çizge yapısı, düğümlerin birbirleriyle yönelimli olarak bağlantılarının bulunduğu yapılardır. Yönsüz çizge oluşturulmasında kullanılan "nx.Graph()" komutunun aksine yönlü çizge oluşturulurken aşağıdaki örnekte olduğu gibi "nx.DiGraph()" komutu kullanılmaktadır. Bu komut ile basit bir yönlü bir çizge yapısı oluşturulabilmektedir.

```
import networkx as nx
# Yönlü çizge yapısı oluşturulur
G = nx.DiGraph()
# Düğümler ve kenarlar eklenir
G.add_nodes_from([1, 2, 3, 4])
G.add_edges_from([(1, 2), (1, 3), (2, 4), (3, 4)])
# Çizge yapısı görselleştirilir
nx.draw(G, with_labels=True)
```



Şekil 3.4. Yönlü çizge örneği

Ağırlıklı çizge, her bir kenarın bir ağırlık veya değerle ilişkilendirildiği yapıdır. Aşağıdaki örnekte ilk olarak "nx.Graph()" fonksiyonu ile "G" isimli bir çizge oluşturulmaktadır. Daha sonra "G.add_nodes_from" komutu ile düğümler eklenmektedir. Düğüm ekleme işleminden sonra "G.add_weighted_edges_from" komutu ile iki düğüm arasında oluşturulacak olan kenarlara ait bilgiler ve kenarın ağırlık değeri girilerek ağırlıklı kenar oluşturulmaktadır. Son olarak "nx.draw()" komutu ile "G" çizgesinin görsel çıktısı ekrana yazdırılır (Şekil 3.4.).

```
import networkx as nx
# Ağırlıklı çizge oluşturulur
G = nx.Graph()
# Düğümler ve kenarlar eklenir
G.add_nodes_from([1, 2, 3, 4])
G.add_weighted_edges_from([(1, 2, 0.5), (1, 3, 1.0), (2, 4, 0.3),
(3, 4, 0.7)])
# Çizge yapısı görselleştirilir
nx.draw(G, with_labels=True)
```

Çizgeler üzerinde yapılacak temel matematiksel işlemlerden birisi çizgesi oluşturan düğümlerin bulundukları çizgedeki konumlarına ve diğer düğümler ile arasında oluşturduğu kenarlara, kenarların sayısına, yönüne, ağırlığına vb. özelliklerine göre merkeziet değerlerini hesaplamaktır. Merkeziet değerlerinin hesaplanabilmesi için, "nx.degree_centrality()", "nx.closeness_centrality()" veya "nx.betweenness_centrality()" gibi kütüphanede önceden tanımlanmış fonksiyonlar bulunmaktadır. Bu fonksiyonları kullanarak sırasıyla düğüm derecesi merkezieti, yakınlık merkezieti ve aralık merkezieti veya diğer merkeziet değerlerinden istenileni hesaplanabilmektedir.

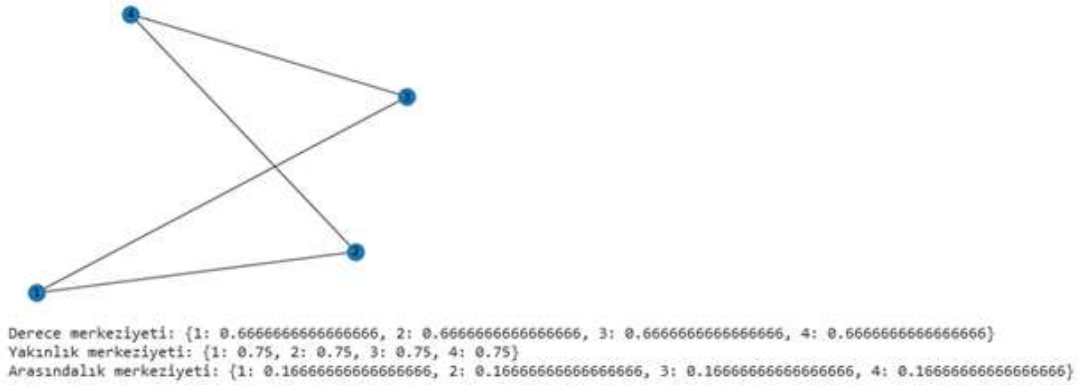
NetworkX kütüphanesi projeye yüklenmektedir.

```

import networkx as nx
# Yeni bir yönsüz çizge oluşturulmaktadır.
G = nx.Graph()
# G ismiyle oluşturulan ağın içerisine 4 adet düğüm eklenmektedir.
G.add_nodes_from([1, 2, 3, 4])
# G ismiyle oluşturulan ağın içerisindeki düğümler arasında kenarlar
oluşturulmaktadır.
G.add_edges_from([(1, 2), (1, 3), (2, 4), (3, 4)])
# “nx.degree_centrality” komutu ile derece merkeziliği değeri hesaplanmaktadır.
degree_centrality = nx.degree_centrality(G)
# “nx.closeness_centrality” komutu ile yakınlık merkeziliği değeri hesaplanmaktadır.
closeness_centrality = nx.closeness_centrality(G)
# “nx.betweenness_centrality” komutu ile arasındalık merkeziliği değeri
hesaplanmaktadır.
betweenness_centrality = nx.betweenness_centrality(G)
# Sonuçlar ekrana yazdırılır.
print ("Derece merkeziliği:", degree_centrality)
print ("Yakınlık merkeziliği:", closeness_centrality)
print ("Arasındalık merkeziliği:", betweenness_centrality)

```

Yukarıdaki kod bloğunun çalıştırılmasıyla birlikte elde edilen sonuç Şekil 3.5.’de gösterilmektedir. Şekil 3.5.’de 4 düğüm arasındaki yönsüz kenar ilişkileri ve bu ilişkiler sonucu her düğüm tarafından elde edilen derece merkeziliği, BC ve yakınlık merkeziliği değerlerine ait sayısal sonuçlar gösterilmektedir.



Şekil 3.5. Örnek kod bloğuna ait ekran çıktısı

Basit ve temel işlemlere ek olarak karmaşık (kompleks) işlemlerde yapılabilmektedir. Örnek olarak aşağıdaki kod bloğunda 9 düğüm ve aralarındaki ağırlıklı kenarların değerlerine göre tek satır komut ile “minimum örten ağaç” algoritması kullanılabilmektedir. Aşağıdaki kod bloğunun çıktısı Şekil 3.6.’da gösterilmektedir.

```

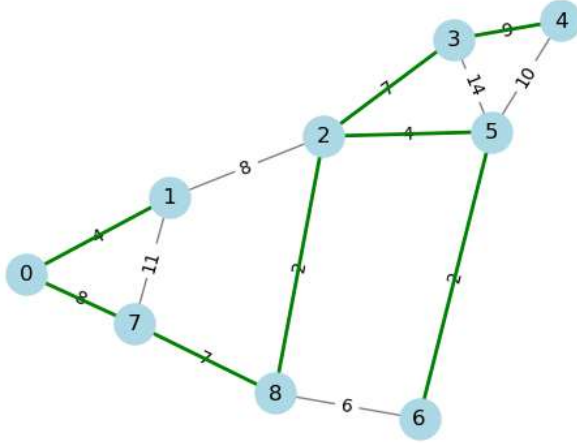
import networkx as nx
import matplotlib.pyplot as plt
G = nx.Graph()
G.add_edges_from(
    [

```

```

        (0, 1, {"weight": 4}),
        (0, 7, {"weight": 8}),
        (1, 7, {"weight": 11}),
        (1, 2, {"weight": 8}),
        (2, 8, {"weight": 2}),
        (2, 5, {"weight": 4}),
        (2, 3, {"weight": 7}),
        (3, 4, {"weight": 9}),
        (3, 5, {"weight": 14}),
        (4, 5, {"weight": 10}),
        (5, 6, {"weight": 2}),
        (6, 8, {"weight": 6}),
        (7, 8, {"weight": 7}),
    ]
)
T = nx.minimum_spanning_tree(G)
pos = nx.spring_layout(G)
nx.draw_networkx_nodes(G, pos, node_color="lightblue",
node_size=500)
nx.draw_networkx_edges(G, pos, edge_color="grey")
nx.draw_networkx_labels(G, pos, font_size=12, font_family="sans-
serif")
nx.draw_networkx_edge_labels(
    G, pos, edge_labels={(u, v): d["weight"] for u, v, d in
G.edges(data=True)})
)
nx.draw_networkx_edges(T, pos, edge_color="green", width=2)
plt.axis("off")
plt.show()

```



Şekil 3.6. Örnek kod bloğuna ait ekran çıktısı

3.1.3. Node2vec kütüphanesi

Node2vec, ağlarda özellik öğrenimi amacıyla kullanılan verimli ve ölçeklenebilir algoritmalarından birisidir. Temel yaklaşımı ağdaki bir düğüm ile komşuları arasında komşuluk ilişkilerini koruma olasılığı yüksek şekilde rastgele yapılan yürüyüşler ile komşu keşfinin verimli ve esnek şekilde öğrenmeye çalışmaktadır (Grover ve Leskovec, 2016). $G = (V, E)$ denklemi yönlendirilmemiş ve ağırlığa sahip olmayan bir ağ olarak kabul edilmektedir. $F: V \rightarrow \mathbb{R}^d$, aşağı akış tahmin görevi için öğrenmenin amaçlandığı düğümlerin özellik temsillerine eşleştirme, d değişkeni gösterilecek özelliklere ait boyut değeri, $f, |V| \times d$ eşdeğerlik parametreleri, Her $u \in V$ kaynak düğümü için, $N_S(u) \subset V$ 'yi, komşuluk örnekleme yöntemi olan S yolu ile oluşturulan u düğümüne ait ağ komşuluğu olarak tanımlanmaktadır. Skip-gram mimarisi kullanılarak ağlara genişletilmekte ve bu şekilde devam edilmektedir (Mikolov ve ark., 2013; Perozzi ve ark., 2014).

Denklem 3.1'de, f ile verilen özellik temsiline göre şartlandırılmış bir u düğümü için bir ağ komşuluğunu $N_S(u)$ gözlemlemenin log-olasılığı maksimize edilmektedir (Grover ve Leskovec, 2016).

$$\max_f \sum_{u \in V} \log \Pr(N_S(u) | f(u)) \quad (3.1)$$

Yukarıda verilen denklemi basitleştirmek için iki varsayım kullanılmıştır. İlk varsayım şartlı bağımsızlıktır. Kaynağa ait özellik temsili verildiği zaman, düğüme ait bir komşu düğümü gözlemleme ihtimali diğer komşu düğümlerin gözlemlenme ihtimali ile bağımsızdır varsayımına dayanılarak olasılık Denklem 3.2'de verilen eşitlik kullanılarak çarpanlarına ayrılmaktadır (Grover ve Leskovec, 2016).

$$\Pr(N_S(u) | f(u)) = \prod_{n_i \in N_S(u)} \Pr(n_i | f(u)) \quad (3.2)$$

İkinci varsayım ise öznitelik uzayında simetriklik varsayımıdır. Kaynak düğüm ve komşu düğüm, öznitelik uzayında birbirleri üzerinde simetrik bir etkiye sahiptir. Buna duruma göre, her kaynak düğüm ve komşu düğüm çiftinin koşullu olasılığını, özelliklerinin bir nokta çarpımı ile parametreleştirilmiş bir softmax unit olarak modellenmektedir (Denklem 3.3.) (Grover ve Leskovec, 2016).

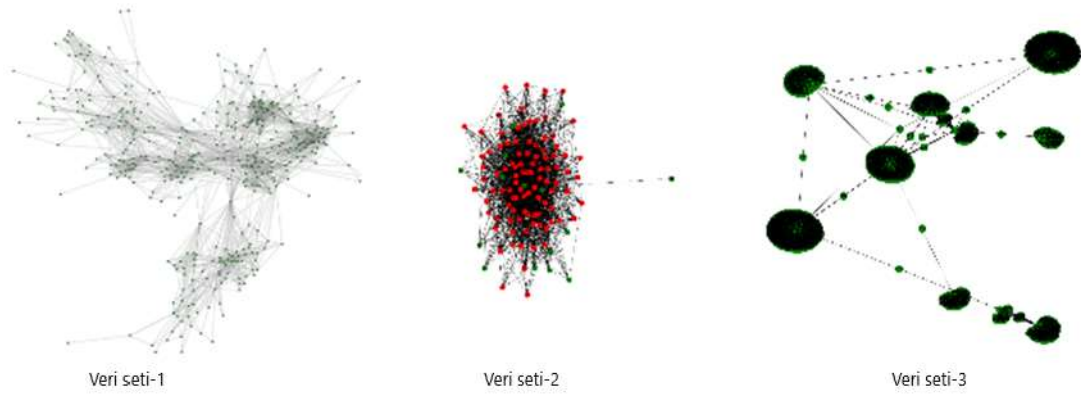
$$\Pr(n_i | f(u)) = \frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))} \quad (3.3)$$

İki varsayımın ilk denkleme uygulanması ile yöntemin sadeleştirilmiş matematiksel formülü Denklem 3.4.’te gösterilmektedir (Grover ve Leskovec, 2016).

$$\max_f \sum_{u \in V} [-\log Z_u + \sum_{n_i \in N_S(u)} f(n_i) \cdot f(u)] \quad (3.4)$$

3.2. Veri Seti

Yapılan çalışmada kullanılan veri setlerinden birincisi, Dublin'deki Bilim Galerisi'nde 2009 yılında yapılan “Infectious: Stay Away” adlı sergide, sergiyi ziyaret eden bireyleri düğüm ve bu bireylerin birbirleriyle en az 20 saniye yüz yüze temaslarının ise kenar olarak temsil edilmesiyle oluşturulmuştur (Keeling ve Eames 2005). İki düğüm arasında birden fazla kenar olabilmekte ve bu durum iki düğüm arasındaki birden çok teması göstermektedir. Veri setleri kullanılarak elde edilen çizgeler Şekil 3.7.’de gösterilmektedir.



Şekil 3.7. Tüm veri setlerine ait ağ yapısı

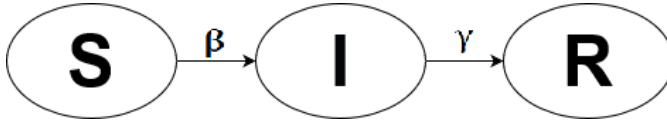
Veri seti-2, “ACM Hypertext 2009” adlı konferansa katılan katılımcılardan elde edilen, yüz yüze iletişim ağıdır. Düğüm ve kenarların elde edilme şartları Veri seti-1 ile aynıdır. Veri seti-3, Facebook üzerinden arkadaşlık ilişkilerine göre elde edilmiş bir ağıdır. Veri setlerine ait temel sayısal bilgiler Çizelge 3.2.’de gösterilmektedir. Veri seti-1 ‘de toplamda 410 düğüm ve aralarında 17.298 kenar bulunmaktadır. Veri seti-2’de 113 düğüm ve aralarında 20.818 kenar bulunmaktadır. Veri seti-3’te 2888 düğüm ve 2.981 kenar bulunmaktadır.

Çizelge 3.2. Veri setlerine temel bilgiler

Özellik Adı	Veri seti-1	Veri seti-2	Veri seti-3
Veri seti tarihi	15.07.2009	29.06.2009	2012
Düğüm bağlantısı	Ziyaretçi	Ziyaretçi	Kullanıcı
Kenar bağlantısı	Temas	Temas	Arkadaşlık
Ağ yapısı	Tek parçalı, yönsüz	Tek parçalı, yönsüz	Tek parçalı, yönsüz
Kenar Tipi	Ağırlıksız, çok kenarlı	Ağırlıksız, çok kenarlı	Ağırlıksız, çok kenarlı

3.3. Kompartıman Modeller

Kermack ve ark., (1927) tarafından literatüre kazandırılan modelde toplamda 3 kompartıman bulunmaktadır (Şekil 3.8.). S (Susceptible – Duyarlı) ile gösterilen ilk kompartımanda bireylerin hastalığa yakalanmadığı fakat hastalığa karşı duyarlı olduğu varsayılmaktadır. S kompartımanın da bulunan ve β sabit hızıyla yayılan hastalığa yakalanan bireyler, ikinci kompartıman olan I (Infectious – Bulaşıcı) kompartımanına geçmektedir. I kompartımanında bulunan ve γ sabit hızıyla hastalığı atlatan, bağışıklık kazanan veya hastalık yüzünden ölen bireyler ise modelin son kompartımanı olan R (Removed / Recovered – Ölüm/ Bağışık) kompartımanında yer almaktadır.



Şekil 3.8. SIR kompartıman modeli

SIR modeli oluşturulurken bazı varsayımlarda bulunulmuştur. Bunlar, toplumun homojen (yaş, sosyal konum, coğrafi konum vb.) yapıda ve dış etkenlere kapalı, nüfusun sabit, doğum veya hastalık dışında ölümün olmadığı, hastalığa karşı bağışıklık kazanan bireylerin tekrar hastalanmadığı, her bireyin hastalığı eşit oranda yaydığı ve hastalığın sadece insanlar arasında geçiş yaptığı durumlarıdır.

$$\frac{dS(t)}{dt} = -\beta S(t)I(t) \quad (3.5)$$

$$\frac{dI(t)}{dt} = \beta S(t)I(t) - \gamma I(t) \quad (3.6)$$

SIR modeline ait denklemlerde (Denklem 3.5-3.7.) kullanılan β ve γ sırasıyla hasta ve iyileşme sabit değerlerini t ise zamanı belirtmektedir (Saeedian ve ark., 2017).

Herhangi bir aşamada $S(t)$, $I(t)$ ve $R(t)$ değerleri toplamı popülasyon büyüklüğüne (N) eşittir.

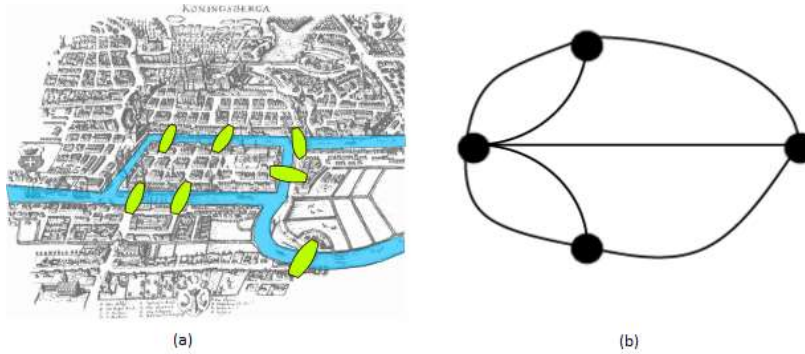
$$\frac{dR(t)}{dt} = \gamma I(t) \quad (3.7)$$

Epidemiyolojide önemli bir parametre, temel üreme katsayısı olan R_0 parametresidir. Tamamen elverişli bir popülasyona yerleştirilen tek bir enfekte birey tarafından bulaştırılan ikincil vakaların ortalama sayısı olarak tanımlanır. Başka bir deyişle, R_0 bize hastalığın ilk yayılma hızını gösterir. Bu nedenle, eğer $R_0 > 1$ ise, bir salgın olacaktır ve $R_0 < 1$ ise, hastalık bulaşan enfekte kişiler yerlerini yeni enfeksiyon bulaşan kişilerle değiştiremeden iyileşecektir (veya ölecektir).

SIR modeli için, R_0 değeri Denklem 3.8.'deki gibi hesaplanır (Saeedian ve ark., 2017):

$$R_0 = \frac{\beta}{\gamma} \quad (3.8)$$

3.4. Çizge Teorisi ve Alt Topluluklar



Şekil 3.9. (a) Königsberg'deki yedi köprü görseli ve (b) çizge çizimi (Biggs ve ark., 1986)

Çizge teorisi, yapısal modellerde kullanılan matematikte önemli bir alandır. Euler, "Königsberg'in yedi köprüsü" (Euler, 1741) adlı makalesi yayımlayarak çizge teorisinin başlangıç adımını atmıştır. Çizge teorisi kavramı, 1735 yılında İsviçreli büyük matematikçi Leonhard Euler tarafından çözülen Königsberg'in Yedi Köprüsü (Şekil 3.9) problemine kadar izlenebilir (Biggs ve ark., 1986; Karadeniz, 2016; Hong ve ark., 2019). Çizge teorisi, toplu taşıma sistemlerini analiz etmek için ulaşım mühendisliği,

yayılma mekanizmalarını keşfetmek için sosyoloji ve hastalıkların yayılmasını izlemek için biyoloji dahil olmak üzere mühendislik alanlarında yaygın olarak uygulanmaktadır (Hong ve ark., 2019).

3.4.1. Çizge teorisi

Çizge, birden fazla nesnenin düğüm olarak gösterildiği, aralarındaki bağlantıların ise kenar olarak gösterildiği bir yapıdır. Literatürde genellikle çizge “G”, çizgeyi oluşturan düğümler “V”, kenarlar ise “E” harfi ile temsil edilmektedir. Düğüm ve kenarların birbirleriyle olan ilişkilerinin gösterildiği bir çizgede elde edilen ağ yapısı birçok yönden ele alınabilmektedir. Wasserman ve Faust’a göre bu düğümler arası ilişkiyi gösteren kenarları sosyal varlıklar arasındaki bağ olarak tanımlamaktadır (Bollobas 1998). Ayrıca Katz ve ark. göre bireyler arası oluşturulan ilişki iş, arkadaşlık, akrabalık gibi farklı yapıları temsil edebilir (Parand ve ark., 2016).

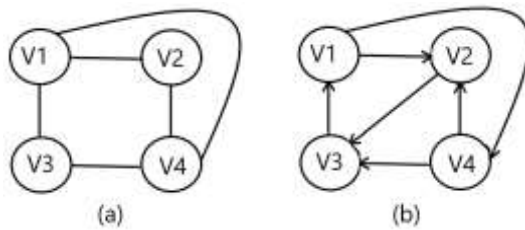
Şekil 3.10’da basit, yönsüz (a) ve yönlü (b) bir çizge örneği gösterilmektedir. Basit ve yönsüz (a) çizgeye ait temel özellikleri şu şekilde gösterebiliriz.

$$V = \{V1, V2, V3, V4\}$$

$$E = \{(V1, V2), (V1, V4), (V2, V4), (V1, V3), (V3, V4)\}$$

$$G = (V, E)$$

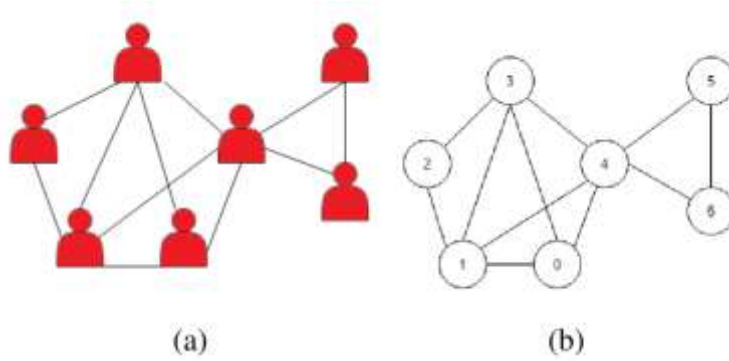
Yukarıdaki matematiksel gösterimde "G" çizgeyi, "V" düğümleri ve "E" kenarları temsil etmektedir. Yukarıdaki bilgilere göre çizgenin 4 düğüme ve düğümler arası toplamda 5 kenara sahip olduğu söylenilebilir. Çizgeleri oluşturan düğümlerin toplam sayısı çizgenin “mertebe” değerini göstermektedir. Örnek olarak Şekil 3.18’te gösterilen iki çizgenin de düğüm sayısı 4 olduğu için mertebesi de 4’tür.



Şekil 3.10. (a) Yönsüz çizge ve (b) yönlü çizge örneği

Çizge gösterimlerinde düğümler arası bağlantıları gösteren kenarların işaretsiz olması durumunda çizgeye “yönsüz çizge” denilmektedir. Şekil 3.10’da a örneğinde örnek olarak yönsüz çizge gösterilmektedir. Burada “V1” düğümü ile “V2” düğümü arasındaki kenar yönsüz olduğu için matematiksel gösterimde “(V1, V2)” ile “(V2, V1)” gösterimi aynı kenarı temsil etmektedir. Yönlü çizgelerde ise düğümler arasında ilişkiyi gösteren çizimler genellikle ok işareti ile yapılır. Örnek olarak tüm kenarların ağır değerinin 1 olması durumunda ortaya çıkan çizgeye “ağırlıksız çizge” denilmektedir. Fakat kenarların değerlerinin farklı olması durumunda ortaya çıkan çizgeye “ağırlıklı çizge” ya da “maliyetli çizge” denilmektedir.

Literatürde ağ tanımı ile ilgili birçok açıklama bulunmaktadır. Ağın tanımın basit olarak “Karmaşık yapıya sahip bir sistemin, sistemi oluşturan elemanların ve bu elemanlar arasındaki etkileşimlerin görsel olarak sunulması” (Şekil 3.11) şeklinde yapılabilmektedir. Literatürün incelenmesi durumunda ağ bilimine ilişkin çalışmaların Euler ile başladığı görülmektedir. Bu çalışmayı takip eden “On Random Graphs” (Erdős ve Rényi, 1960), “The small world problem” (Milgram, 1967), “Collective dynamics of ‘small-world’ networks” (Watts ve Strogatz, 1998), “Emergence of Scaling in Random Networks” (Barabási ve Albert, 1999) vb. çalışmalar ağ biliminin gelişmesinde önemli katkıları olan bazı çalışmalardır (Gürsakal ve ark., 2014).



Şekil 3.11. (a) Örnek arkadaşlık ağının (b) çizge ile gösterimi

Özet olarak çalışmalar sayesinde ağların işleyişi, sahip oldukları özelliklerin çıkarımı, anlamlandırılması gibi konular üzerinde durularak açıklanmaya çalışılarak yukarıda örnekleri verilen ağ modelleri ortaya çıkarılmıştır. Hong (2015), ağ topolojisini, bir ağın düğümler ve bağlantılar gibi öğelerinin fiziksel düzenlemesi veya birbirine bağlanması olarak tanımlar. Bir bilgisayarda ulaşım sistemlerinin ağlarını

görselleştirmenin, coğrafi bilgilere atıfta bulunmanın ve çıkarmanın en etkili yolu olmuştur (Lin ve Ban, 2013). Ağların anlamlandırılması aşamasında düğüm, kenar, en kısa yol, komşuluk, yarıçap, yoğunluk, merkezilik derecesi, kümelenme katsayısı gibi yapısal özellikler kullanılmaktadır.

3.4.2. Topluluk tespiti

Genel olarak, bir veri kümesinde bulunan bir grup varlığın diğer varlıklardan daha yoğun ilişki içerisinde olması durumunda bu varlık grubuna topluluk denilmektedir (Ma ve ark., 2020). Varlıklar arasındaki yoğun etkileşim, benzerlik ya da mesafe ile ölçülebilmektedir. Topluluk tespiti, birbirleri ile yoğun ilişkili düğüm gruplarını ortaya çıkarmayı amaçlamaktadır. Topluluk tespit problemi Np-hard olarak sınıflandırılan problemlerden birisidir. Bu nedenle problemlerin önerilen çözümleri yüksek verimliliğe sahip olmalıdır (Huang ve Lakshmanan, 2017). Ağ yapısının anlaşılabilirliği ve analizinin kolaylaştırılması amacıyla çizge teorisinden yararlanılmaktadır. Örnek bir arkadaşlık ağının çizge hali Şekil 3.11’de gösterilmiştir. $G = (V, E)$ denkleminde gösterilen arkadaşlık ağında ilişkiler karşılıklı olduğu için aradaki kenarlar yönsüz ve sadece aralarındaki ilişki bilindiği için ağırlıksız olarak gösterilmektedir.

Topluluk tespiti için ağlarda etkili düğümlerin keşfine odaklanan ve bulmaya çalışan algoritmalar bulunmaktadır (Gao ve ark., 2014; Zhao ve ark., 2016; Xing ve ark., 2019). En yaygın olarak kullanılan yöntemler derece merkeziliği ölçütü (Rong ve ark., 2019), yakınlık merkeziliği ölçütü (Sabidussi, 1966; Ma ve ark., 2019), arasındalık merkeziliği ölçütü (Barthélemy, 2004) ve PageRank (Brin ve Page, 1998) olarak gösterilebilir (Ma ve ark., 2020). Merkezilik ölçütleri, ağ yapısını oluşturan düğümlerin karşılıklı ilişkilerine göre önem derecesini belirtmek için kullanılan bir ölçüttür. Freeman, bilgi aktarımı ve iletişimi ölçmek amacıyla derece merkeziliğini, aracılık durumu veya ilgi kontrolünü ölçmek amacıyla arasındalık merkeziliğini ve verimlilik ve uygun olma seviyesini tahmini için de yakınlık merkeziliğinin kullanılabileceğini belirtmiştir (Freeman, 1977; J. Zhang ve Luo, 2017). Bu merkezilik ölçütleri ile ilgili çalışmaların temeli 1970’ li yıllarda Freeman (Freeman, 1977; Freeman ve ark., 1979; Katz ve ark., 2004) tarafından yapılmıştır. Arasındalık

merkeziliği ölçüsü Freeman (1977) ile yakınlık merkeziliği ölçüsü Beauchamp (1965) sahip olduğu düşük hesaplama maliyeti ile daha iyi sonuçlar üretebilmektedir. Farklı merkezilik ölçütlerinin hesaplama karmaşıklık değerleri Çizelge 3.3’ te gösterilmektedir.

Çizelge 3.3. Farklı merkezilik ölçütlerinin hesaplama karmaşıklığının karşılaştırılması

Yöntem	Hesaplama Karmaşıklığı	Notasyon Açıklama
Derece merkeziliği	$O(n)$	n: düğüm sayısı
Yakınlık merkeziliği	$O(n^3)$	
Arasındalık merkeziliği	$O(n^2 \log n + nm)$	m: kenar sayısı
Özvektör merkeziliği	$O(n^2)$	
PageRank	$O(mi)$	i: iterasyon sayısı

Derece merkeziliği (Degree Centrality- DC), ağ yapısı oluşturan düğümlerin, doğrudan birbirleriyle olan ilişkilerinin toplamına göre hesaplanmaktadır (Zhang ve Luo, 2017; Bolland, 1988). Komşuluk matrisi ($A = (a_{ij})$) kullanılarak Denklem 3.9.’da görülen şekilde formüle edilir. σ_D derece merkeziliğini, j derece merkezilik değeri bulunmak istenilen düğümü, n ağdaki toplam düğüm sayısını ve a_{ij} ise komşuluk matrisinde i ve j düğümleri arasındaki mesafeyi belirtmektedir. Şekil 3.10’da verilen örnek çizge incelendiğinde “V1” ve “V4” düğümünün dereceleri 3, “V2” ve “V3” düğümlerinin derecesi ise 2 olarak elde edilir.

$$\sigma_D(j) = \sum_{i=1}^n a_{ij} \quad (3.9)$$

Yakınlık merkeziliği (Closeness Centrality- CC), bir düğümün diğer düğümlere olan toplam mesafelerini belirtmek için kullanılır. Ağdaki bir düğümün diğer düğümler ile arasında olan mesafe ne kadar küçükse CC değeri o kadar yüksek çıkmaktadır (Bollobas, 1998; Shaw, 1954). CC hesaplamada kullanılan Denklem 5 de gösterilen σ_C yakınlık merkeziliğini, j yakınlık merkezilik değeri hesaplanmak istenilen düğümü, n ağdaki toplam düğüm sayısını ve $d_G(j, i)$ ise i ve j düğümleri arasındaki en kısa mesafeyi belirtmektedir.

$$\sigma_C(j) = \frac{1}{\sum_{i=1}^n d_G(j, i)} \quad (3.10)$$

Arasındalık merkeziliği (Betweenness Centrality- BC), bir ağda bulunan ve doğrudan bağlantısı olmayan iki düğümün bağlantı kurabilmesini sağlayan aracı rolündeki düğümün önemini göstermektedir. Bir düğümün, iletişim, bağlantı, ulaşım veya işlem gibi diğer düğümlerin geçmesi gereken tek yolu buluyorsa, bu düğüm önemli olmalı ve büyük olasılıkla yüksek bir BC değerine sahip olmalıdır (Katz ve ark., 2004). BC hesaplamada kullanılan Denklem 3.11.'de gösterilen σ_B arasındalık merkeziliğini, j arasındalık merkezilik değeri hesaplanmak istenilen düğümü, n ağdaki toplam düğüm sayısını, g_{ix} ise i ve x düğümleri arasındaki en kısa mesafeyi ve $g_{ix}(j)$ ise i ve x düğümleri arasında j düğümünden geçen en kısa mesafeyi belirtmektedir.

$$\sigma_B(j) = \sum_{i=1, i \neq j}^n \sum_{x=1, x < i, x \neq j}^n \frac{g_{ix}(j)}{g_{ix}} \quad (3.11)$$

Özvektör Merkeziliği (Eigenvector Centrality- EC), Bonacich ve Lloyd (2001) tarafından literatüre kazandırılmıştır. Yöntem, bitişik olan bir matrise ait en büyük özdeğerinin sahip olduğu özvektörünün kuvvetli bir ağ merkezilik ölçütü olduğu hipotezi üzerinde kurulmuştur (Parand ve ark., 2016). Temelinde her bir düğümün değeri sadece doğrudan bağlantılarla değil ayrıca dolaylı bağlantıların değerleriyle de belirlenmelidir. Bu sayede bir düğümün değeri belirlenirken ağın tamamı dikkate alınmaktadır. EC'ye ait temel matematik formülüne göre (Denklem 3.12) x iki eşdeğer şekilde, bir matris denklemi ve bir toplam olarak tanımlanmaktadır. Bir köşenin merkeziliği, bağlı olduğu köşelerin merkezizetlerinin toplamı ile doğru orantılıdır. λ değeri, A 'nın en büyük özdeğeridir ve n değeri, köşe sayısını temsil etmektedir (Parand ve ark., 2016).

$$Ax = \lambda x, \lambda x_i = \sum_{j=1}^n a_{ij}x_j, i = 1, \dots, n \quad (3.12)$$

PageRank (PR) yöntemi, Google şirketi tarafından geliştirilen bir yöntemdir. Yöntemin temel amacı sorgu kriterlerine göre en yararlı sayfaları bulmak ve sıralamaktır. Sorgu sonucu elde edilen sayfalar başlık etiketleri, anahtar kelimeler vb. niteliklerine göre değerlendirilmekte ve sonuç olarak sıralanmaktadır. PR yöntemine ait formüle bakıldığında u değeri bir web sayfasını temsil etmektedir. Denklem 3.13'te

gösterilen PR yöntemine ait matematiksel denklemde kullanılan $B(u)$ ise, u yu kapsayan sayfaların toplamıdır (Brin ve Page, 1998). $PR(u)$ ve $PR(v)$ değerleri sırasıyla u ve v sayfalarının elde ettiği sıralama puanlarını göstermektedir. Burada $PR(u)$ ve $PR(v)$, sırasıyla u ve v sayfalarının sıralama puanlarıdır. N_v değeri sayfanın giden bağlantılarının sayısı, d ise sönümlleme faktörüdür (genelde 0,85 değeri olarak alınır). Ek olarak d değeri kullanıcıların bağlantıları takip etme ihtimali olarak görülebilir ve bu durumda $(1 - d)$ değerini doğrudan bağlı olmayan düğümlerden PR dağılımı olarak kabul edilebilir.

$$PR(u) = (1 - d) + d \sum_{v \in B(u)} \frac{PR(v)}{N_v} \quad (3.13)$$

Gerçek ağların çoğu, birçok düğüm ve kenar ile karmaşık olduğundan, birçok algoritma büyük ağlarda verimsiz bir şekilde yürütülmüştür. Son yıllarda toplulukları tespit etmek için modülerlik tabanlı algoritmalar, spektral algoritmalar, yoğunluk tabanlı algoritmalar gibi birçok yöntem önerilmiştir. Clauset ve ark., (2004) tarafından önerilen Ağgözlü Modülerite (Greedy Modularity) aşağıdan-yukarı yaklaşımını kullanmaktadır. Bu yöntemde ilk olarak her köşe tek başına bir topluluk olarak düşünülmektedir. Sonraki aşamada ise toplamda tek bir topluluk oluncaya kadar modüleriteyi arttıran tüm topluluklar birleştirilmektedir. Aralarında bağ olmayan toplulukların birleştirilmesi işlemi modüleriteyi arttırmayacağı için sadece aralarında bağ olan topluluklar birleştirilmektedir. İlerleyen zamanlarda Meghanathan (2016) tarafından “NOVER” adlı ağgözlü modülerite algoritması önerilmiştir. İki algoritma arasındaki temel fark ise NOVER algoritmasında belirlenen eşik değerine göre kenarlar güçlü ve zayıf olarak ayrılmaktadır. Çizgeden çıkartma işlemleri sadece zayıf grupta bulunan kenarlara uygulandığı için bu durum verimlilik yönünden artışı sağlamaktadır.

Bir diğer yöntem olan Klik süzme yöntemi (Clique Percolation Method, CPM) Derényi ve ark., (2005) tarafından 2005 yılında önerilmiştir. Bu algoritmada ilk olarak kaba kuvvet kullanılarak klikler tespit edilmektedir. Başlangıç aşamasında köşelerin her biri 1-klik olarak kabul edilmektedir. Daha sonra derecesi $k-1$ 'e eşit ya da düşük olan köşeler sırayla çizgeden çıkartılır. Kliklerin bulunmasıyla birlikte komşu klikler

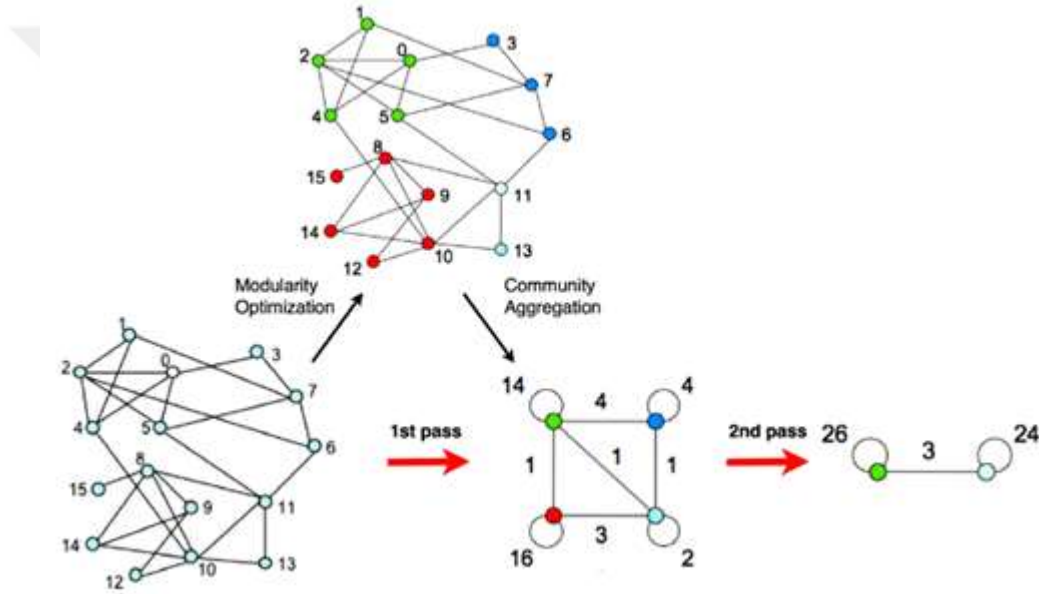
de tespit edilerek birleştirme işlemi yapılır. Temel klik süzme yöntemi üzerinde optimizasyonlar yapılarak farklı yöntemler geliştirilmiştir (Kasoro ve ark., 2019; Wen ve ark., 2017). Fakat doğası gereği klik süzme temelli yöntemler işlem zamanı yönünden yüksek maliyetli olabilmektedir.

Walktrap algoritması (Pons ve Latapy 2005), temelde köşeler arası uzaklık yardımıyla komşu köşeler arasındaki benzerliği hesaplamak için kullanılmaktadır. Köşeler arası mesafe rastgele bir yürüyüşçü tarafından belirlenmektedir. Aradaki mesafesi en yakın olan köşeler benzer köşeler olarak sınıflandırılmaktadır. Yürüyüşçü tarafından gidilebilecek maksimum mesafe algoritmanın en başında belirlenmektedir. Bu nedenle yoğun ağlarda yürüyüş mesafesinin düşük belirlenmesi yararlıdır. Özellikle yürüyüşçünün güzergahı ve yürüyüş mesafesi üzerinde değişik yöntemler kullanılarak algoritmanın farklı türleri üzerinde çalışmalar yapılmıştır (Reichardt ve Bornholdt, 2006; Okuda ve ark., 2021; Wen ve ark., 2017).

Etiket Yayılımı (Label Propagation) algoritması tarafından 2007 yılında literatüre kazandırılmıştır. Bu yöntemde ilk olarak her köşeye özel etiket ataması yapılmaktadır. Daha sonra her adımda komşular arasındaki etiketlere bakılarak en fazla bulunan etikete göre güncelleme işlemi gerçekleştirilir. Bu aşamada en yaygın etiket sayısı eğer iki ya da daha fazla ise aralarında rastgele bir seçme işlemi gerçekleştirilmektedir. Bu adım sürekli tekrar edilmektedir. Bu aşamanın bitmesi için algoritmanın en başında durma kriteri belirlenmelidir. Genelde “durma kriteri” olarak köşelerin etiketlerinin değişmemeye başlaması seçilmektedir. Bu yöntemin olumsuz taraflarından birisi senkron yöntemlerde köşelerin eş zamanlı olarak etiketlerinin değişmesi durumunda sonsuz döngüye girilmesi gösterilebilir. Bu olumsuzluktan kaçınmak için yöntem asenkron biçimde çalıştırılabilmektedir. Fakat bu sefer de zaman maliyeti yüksek çıkmaktadır. Bu olumsuz özelliğin giderilmesi üzerine çalışmalar yapılmıştır. Örnek olarak (Cordasco ve Gargano, 2010) yaptıkları çalışmada, köşeler için minimum sayıda renk kullanılmaktadır. Ayrıca renklendirme işleminde aynı etikete sahip iki komşunun yan yana gelme olasılığı ortadan kaldırılacak şekilde hareket edilmektedir. Aynı renkte olan köşeler aynı zamanda güncellenmekte fakat renklendirmeler ise sıralı şekilde yapılmaktadır. Etiket yayılımı algoritması güncelliğini korumakta ve literatürde sıklıkla çalışılan algoritmalarından

birisidir (Ben El Kouni ve ark., 2020; Lu ve ark., 2019; Shahrivari Joghhan ve Azad, 2019; Hosseini ve Rezvanian 2020).

Blondel ve ark. (2008) tarafından geliştirilen ve “Louvain” adı verilen topluluk arama algoritması, yapılan denemeler (Orman ve Labatut 2009; Orman ve ark., 2011) sonucu performans sonuçlarına göre en iyi algoritmalarından birisi olarak gösterilmektedir. Louvain algoritması, hızlı yakınsama özellikleri, yüksek modülerliği ve hiyerarşik bölümlleme sayesinde birçok uygulama alanında kullanılmaktadır (Pujol ve ark., 2009; Raeder ve Chawla, 2011; Meunier ve ark., 2009; Mohammadi ve ark., 2021; Tang ve ark., 2018; Kempinska ve ark., 2018; Salha-Galvan ve ark., 2022).



Şekil 3.12. Louvain algoritmasının adımlarının görselleştirilmiş hali (Blondel ve ark., 2008)

Denetimsiz algoritmalarından birisi olan Louvain algoritması, kısmen fırsatçı iyileştirme işleyişine bağlıdır ve elde edilen toplulukları tekrar tekrar oluşturabilme özelliği sayesinde geniş ağlarda kullanım için uygundur. Şekil 3.13'te Louvain algoritmasına ait kaba kod gösterilmektedir. Ayrıca algoritmaya ait temel adımlar Şekil 3.12'de görselleştirilmiştir. Başlangıçta her düğüm birer topluluk olarak kabul edilmektedir ve düğümün komşu topluluklara eklenmesiyle modülerite kazancı hesaplanmaktadır. Hesaplama sonucunda düğüm modülerite kazancı yüksek olan topluluğa eklenir ve eski topluluktan çıkartılır. Eğer kazanç durumu yoksa eski topluluğunda kalmaya devam eder. Louvain algoritmasında bu adım hiçbir iyileşme durumu olmayıncaya kadar devam eder. İyileşme durumunun bitmesi üzerine sonraki

adıma geçilir. Bu adımda ise ilk adımda bulunan topluluklar üzerinden yeni ağ oluşturma işlemi başlamaktadır. Elde edilen yeni ağ üzerinde topluluk içerisinde olan bağlantılar öz döngü olarak, topluluklar arasında oluşan bağlantılarsa ağırlıklı bağlantılar ile gösterilmektedir. Son olarak ilk adım yeni kurulan ağa tekrar uygulanmaya başlanır. Tüm işlemler topluluk yapısı değişmeyene kadar tekrar edilmektedir.

```

1  Input:  $G = (V, E)$  // Graph  $G$  of set of nodes and set of edges representing a network
2  Output: Community structure detected in  $G$ 
3  Initialization: Assign a unique community to each node in the network
4  Begin:
5      Improvement = True
6      While (improvement)
7          Improvement = False
8          Mod = modularity() // Modularity of initial partition is computed
9          Arrange nodes in random order,  $X$ 
10         While (for every node  $i$  in  $X$ ; and one node is moved and there is gain in modularity)
11             Improvement  $\leftarrow$  Modularity_Greedy_Optimization () // Run phase one
12         End While
13         Mod = modularity() // Modularity of the resulted structure is computed
14          $G \leftarrow$  Meta_Graph_Construction () // Run phase two
15     End While
16 End

```

Şekil 3.13. Louvain algoritmasına ait kabakod (Aldabobi ve ark., 2022)

Denklem 3.14'te Louvain algoritmasına ait matematiksel denklem gösterilmektedir. Bu denklemde ΔQ modülarite değerini, $\sum_{in}$ C topluluğunda bulunan ağırlıkların toplam değerini, $\sum_{tot}$ ise C deki köşelere bağlı olan kenarların ağırlıklarının toplam değerini ifade etmektedir. Ayrıca i köşesine bağlı olan kenarların toplam ağırlığını k_i , i köşesi ile C topluluğu arasında bulunan kenarların ağırlıklarının toplam değerini $k_{i,in}$ ve son olarakta ağdaki bütün kenarların ağırlıklarının toplam değeri m ile ifade edilmektedir.

$$\Delta Q = \left[\frac{\sum_{in} + 2k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right] \quad (3.14)$$

3.5. Çok Kriterli Karar Verme Yöntemleri

Çok kriterli karar verme (ÇKKV), çözülmek istenilen problemde, kullanılacak kriterlerin önemleri amaca uygun şekilde ağırlıklandırılarak, çözüm olarak görülen alternatifler arasından seçim işleminin gerçekleştirilmesidir. Çok kriterli karar verme yöntemleri Hwang ve Yoon (1981) tarafından ÇKKV ve Çok Amaçlı Karar Verme (ÇAKV) olarak ikiye ayrılmış olup ÇKKV problemleri, alternatifleri önceden belirlenmiş, genellikle sınırlı ve kriterler açısından mümkün olduğunca az ödün verilerek sonuca ulaşılabilecek problemler olarak tanımlanmaktadır (Hwang ve Yoon 1981).

3.5.1. Topsis yöntemi

Çok kriterli karar verme yöntemlerinden olan TOPSIS'te alternatiflerin sıralanmasında gözetilen temel yaklaşım, kriterlerin göz önünde bulundurulmasıyla birlikte pozitif yönden ideal çözüme en yakın olmalı ve aynı zamanda negatif yönden ideal çözüme en uzak olması şartlarını taşımalıdır (Opricovic ve Tzeng 2004). TOPSIS yöntemine ait işlem basamakları ve kullanılan denklemler sırayla şu şekildedir (Opricovic ve Tzeng 2004):

$$r_{ij} = f_{ij} / \sqrt{\sum_{j=1}^J f_{ij}^2} \quad (3.15)$$

İlk olarak Denklem (3.15) kullanılarak r_{ij} normalize değerler hesaplanmaktadır. Karar matrisinde bulunan normalize değerlerin ağırlıklandırılması için Denklem (3.16) yardımıyla v_{ij} değeri hesaplanmaktadır.

$$v_{ij} = w_i r_{ij} \quad (3.16)$$

$$A^+ = \{v_1^+, \dots, v_n^+\} \left\{ \left(\max_j v_{ij} \mid i \in I' \right), \left(\min_j v_{ij} \mid i \in I'' \right) \right\} \quad (3.17)$$

Matrisin v_{ij} değerleri hesaplandıktan sonra, Denklem (3.17) yardımıyla pozitif (A^+), Denklem (3.18) yardımıyla negatif (A^-) ideal değerleri hesaplanmaktadır.

$$A^- = \{v_1^-, \dots, v_n^-\} = \left\{ \left(\min_j v_{ij} \mid i \in I' \right), \left(\max_j v_{ij} \mid i \in I'' \right) \right\} \quad (3.18)$$

Denklemlerde kullanılan, i parametresi kriter sıra sayısını, w_i parametresi i . sıradaki değişkene ait ağırlık değerini, j parametresi alternatif sıra sayısını, J parametresi toplam alternatif sayısını, n parametresi toplam kriter sayısını, I' parametresi kâr ölçütleri kümesini, I'' parametresi maliyet ölçütleri kümesini ifade etmektedir.

$$D_j^+ = \sqrt{\sum_{i=1}^n (v_{ij} - v_j^+)^2} \quad (3.19)$$

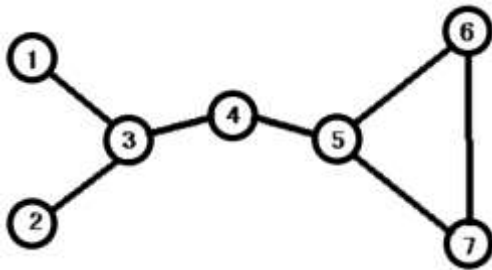
Her bir çözümün ideal çözümlere olan uzaklığını hesaplamak için Öklid kullanılır. Pozitif ideal çözüme olan uzaklık hesabı Denklem (3.19) ile negatif ideal çözüme olan uzaklık hesabı ise Denklem (3.20) ile yapılır.

$$D_j^- = \sqrt{\sum_{i=1}^n (v_{ij} - v_j^-)^2} \quad (3.20)$$

İdeal çözüme olan benzerliklerin hesaplanması kısmında ise bir önceki aşamada elde edilen değerler kullanılarak Denklem (3.21) ile hesaplanır.

$$C_j = D_j^- / (D_j^+ + D_j^-) \quad (3.21)$$

Son adımda ise elde edilen değerler, benzerlik değerleri temel alınarak en yüksekten en düşüğe doğru sıralanır ve ilk sıradaki alternatif önerilir (Opricovic ve Tzeng 2004).



Şekil 3.14. Örnek ağ yapısı (Watabe, 1998)

Şekil 3.14’de verilen çizgeye ait döğümün ağdaki etkililiklerine göre sıralanmasında TOPSIS yöntemi kullanılarak yapılan işlemler adım adım gösterilmektedir.

Şekil 3.14'te gösterilen çizge yapısı incelendiğinde toplamda 7 düğüm ve 7 kenardan oluşmaktadır. Ayrıca yönsüz ve ağırlıksız bir yapıya sahiptir. İlk olarak düğümlere ait BC, CC ve DC merkezilik değerleri hesaplanmaktadır (Çizelge 3.4.). Alternatiflere ait (1,2,3,4,5,6 ve 7 numaralı düğüm) kriterlerin (BC, CC ve DC değerleri) ağırlıkları sırasıyla $w_1 = 0,33$ $w_2=0,33$ ve $w_3=0,34$ olacak şekilde seçilmiştir.

Çizelge 3.4. Düğümlere ait merkezilik değerleri

Düğüm	BC	CC	DC
1	0	0.37	0.16
2	0	0.37	0.16
3	0.35	0.54	0.5
4	0.28	0.6	0.33
5	0.28	0.54	0.5
6	0	0.4	0.33
7	0	0.4	0.33

İlk aşamada “nxm” boyutlarına sahip bir karar matrisi oluşturulmaktadır. Bu aşamada “n” alternatifleri (düğümleri) ve “m” ise kriterleri (düğümlere ait merkezilik değerleri) belirtmektedir.

$$D = \begin{bmatrix} 0 & 0,37 & 0,16 \\ 0 & 0,37 & 0,16 \\ 0,35 & 0,54 & 0,5 \\ 0,28 & 0,6 & 0,33 \\ 0,28 & 0,54 & 0,5 \\ 0 & 0,4 & 0,33 \\ 0 & 0,4 & 0,33 \end{bmatrix}$$

D karar matrisinde satırlar düğümleri, sütunlar ise düğümlere ait merkezilik değerlerini temsil etmektedir. Sonraki aşamada Denklem 3.15 yardımı ile “R” standart karar matrisi oluşturulmaktadır. Aşağıda 1 numaralı düğüme ait ilk kriter ve 2 numaralı düğüme ait ilk kriter için örnek hesaplama gösterilmektedir. Aynı işlemler “D” matrisindeki tüm elemanlar için gerçekleştirilir.

$$r_{11} = \frac{0}{\sqrt{0^2 + 0^2 + 0,35^2 + 0,28^2 + 0,28^2 + 0^2 + 0^2}} = 0$$

$$r_{21} = \frac{0,37}{\sqrt{0,37^2 + 0,37^2 + 0,54^2 + 0,6^2 + 0,54^2 + 0,4^2 + 0,4^2}} = .29845$$

$$R = \begin{bmatrix} 0 & 0,29845 & 0,17076 \\ 0 & 0,29845 & 0,17076 \\ 0,66226 & 0,43557 & 0,53364 \\ 0,52981 & 0,48397 & 0,35220 \\ 0,52981 & 0,43557 & 0,53364 \\ 0 & 0,32264 & 0,35220 \\ 0 & 0,32264 & 0,35220 \end{bmatrix}$$

“R” standart karar matrisinin oluşturulması aşamasından sonra kriterlere ait ağırlıkların kullanılmasıyla birlikte “V” ağırlıklandırılmış standart karar matrisi elde edilir. Bu aşamada her kriter önceden belirlenmiş ağırlık değeri ile çarpılır (Denklem 3.16). Aşağıda 1. Düşüme ait ilk kriter ve 2. Düşüme ait ilk kriter için örnek hesaplama gösterilmektedir. Aynı işlemler “R” matrisindeki tüm elemanlar için gerçekleştirilir.

$$V_{11} = 0 \times 0,33 = 0$$

$$V_{21} = 0,29845 \times 0,33 = 0,09489$$

$$V = \begin{bmatrix} 0 & 0,09849 & 0,05806 \\ 0 & 0,09849 & 0,05806 \\ 0,21854 & 0,14374 & 0,18144 \\ 0,17483 & 0,15971 & 0,11975 \\ 0,17483 & 0,14374 & 0,18144 \\ 0 & 0,10647 & 0,11975 \\ 0 & 0,10647 & 0,11975 \end{bmatrix}$$

“V” ağırlıklandırılmış standart karar matrisi hesaplamasından sonraki aşamada “A⁺” pozitif ideal çözüm değerleri (Denklem 3.17) ve “A⁻” negatif ideal çözüm değerleri (Denklem 3.18) elde edilir. Bu aşamada “V” matrisinde bulunan her kriter için en iyi değer ve en kötü değer tespit edilmektedir.

$$A^+ = \{0,21854 - 0,15971 - 0,18144\}$$

$$A^- = \{0 - 0,09849 - 0,05806\}$$

“A⁺” pozitif ideal çözüm değerleri ve “A⁻” negatif ideal çözüm değerlerinin hesaplanmasından sonra her alternatif için “D⁺” pozitif ideal çözüme olan yakınlık (Denklem 3.19) ve “D⁻” negatif ideal çözüme olan uzaklık (Denklem 3.20) değerleri hesaplanmaktadır. “D⁺” pozitif ideal çözüme olan yakınlık hesaplama aşamasında her alternatif için “V” ağırlıklandırılmış standart karar matrisinde hesaplanmış olan tüm kriterler ve “A⁺” pozitif ideal çözüm değerleri kullanılarak işlem yapılmaktadır. 1 numaralı alternatife ait örnek işlem aşağıda gösterilmektedir.

$$D_1^+ = \sqrt{(0 - 0,21854)^2 + (0,09849 - 0,15971)^2 + (0,05806 - 0,18144)^2} = 0,25832$$

“ D^- ” negatif ideal çözüme olan uzaklık hesaplama aşamasında her alternatif için “ V ” ağırlıklandırılmış standart karar matrisinde hesaplanmış olan tüm kriterler ve “ A^- ” negatif ideal çözüm değerleri kullanılarak işlem yapılmaktadır. 1 numaralı alternatife ait örnek işlem aşağıda gösterilmektedir.

$$D_1^- = \sqrt{(0 - 0)^2 + (0,09849 - 0,09849)^2 + (0,05806 - 0,05806)^2} = 0$$

Alternatifler için “ D^+ ” pozitif ideal çözüme olan yakınlık ve “ D^- ” negatif ideal çözüme olan uzaklık değerlerinin elde edilmesiyle birlikte son olarak “ C_1 ” ideal çözüme göre yakınlık değerleri (Denklem 3.21) hesaplanmaktadır. 1 numaralı düğüme ait örnek işlem aşağıda gösterilmektedir.

$$C_1 = \frac{0}{0,25835 + 0} = 0$$

Çizelge 3.5. Düğümlere ait sonuçlar

Düğüm	D^+	D^-	C	Sıralama
1	0.258	0	0	5
2	0.258	0	0	5
3	0.016	0.255	0.941	1
4	0.076	0.195	0.721	3
5	0.047	0.219	0.825	2
6	0.233	0.062	0.211	4
7	0.233	0.062	0.211	4

Yapılan işlemler sonucunda elde edilen değerler ve düğümlere ait sıralama sonuçları Çizelge 3.5.’te gösterilmektedir. TOPSIS yöntemi sonuçlarına göre Şekil 3.14’te gösterilen çizge yapısında en etkili düğüm 0,941 TOPSIS değeri ile 3 numaralı düğüm çıkmıştır. En az etkili düğüm ise 0 TOPSIS değeri ile 1 ve 2 numaralı düğümler olarak görülmektedir.

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

Yapılan çalışmada ilk olarak, topluluğu temsil eden ağlardaki düğümler, BC, CC, DC, EC ve PR yöntemlerinin uygulanması sonucu elde edilen değerlere göre en iyiden en kötüye doğru sıralanmıştır. Oluşan sıralamada topluluğun %10'una eşit ilk %10'luk düğümler bağışık olarak seçilmiştir. Veri seti-1 için 41 düğüm, veri seti-2 için 11 düğüm ve veri seti-3 için 288 adet düğüm seçilmiştir. Daha sonra her veri setinin %10'una eşit olacak şekilde geri kalan düğümler arasından tek seferlik rastgele düğümler seçilerek enfekte olarak belirlenmiştir. Çalışma başında seçilen enfekte düğümler tüm işlemler boyunca değiştirilmemiştir.

4.1. Temel metriklerin uygulanması

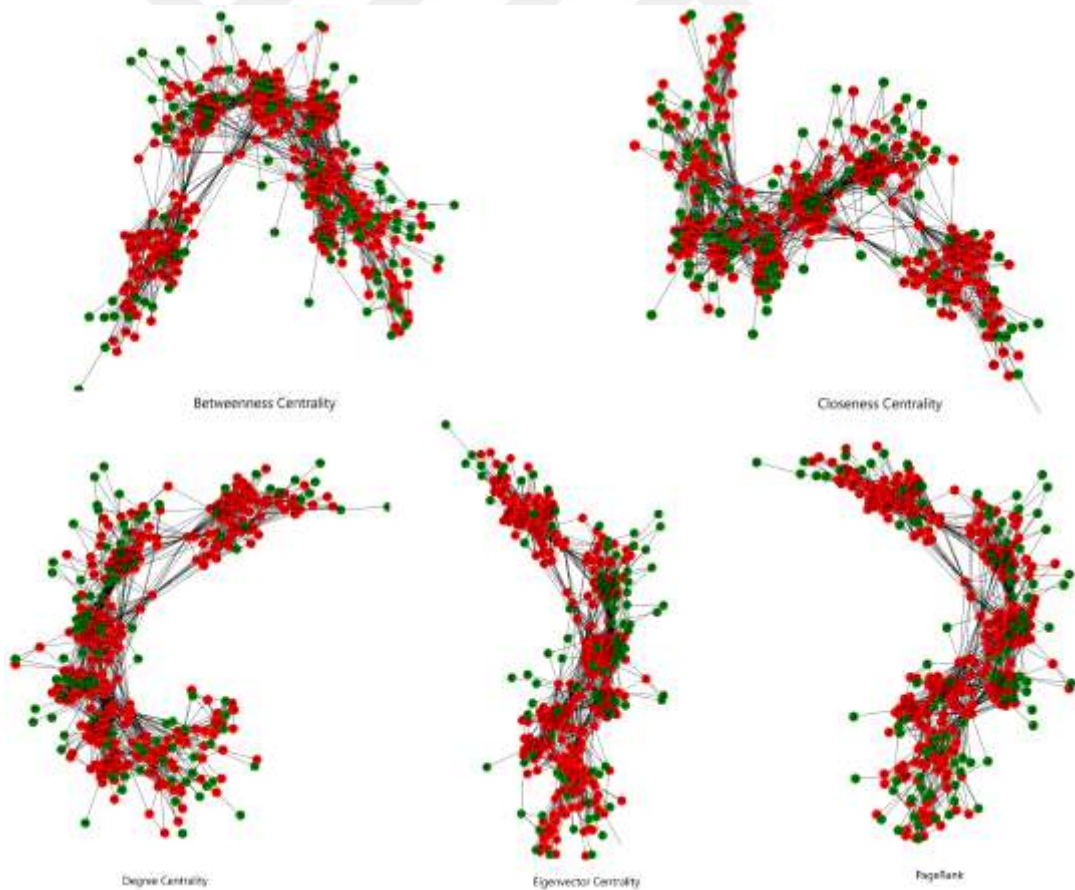
Çizelge 4.1. incelendiğinde, Veri seti-1 için 97 numaralı düğüme ait BC ve CC sonucu en iyi düğüm olarak belirlenirken, diğer merkeziyet derecelerinde ilk 10'a girememiştir.

Çizelge 4.1. Merkeziyet derecelerine göre ilk 10 düğüm

	Sıra	1	2	3	4	5	6	7	8	9	10
Veri seti-1	BC	97	161	20	131	72	66	233	120	130	296
	CC	97	20	131	72	65	161	51	130	66	74
	DC	20	161	198	120	175	69	158	139	256	72
	EC	256	259	254	257	158	258	120	255	280	282
	PR	20	161	120	72	198	19	358	314	304	14
Veri seti -2	BC	9	4	33	5	19	10	6	18	80	17
	CC	9	4	5	19	33	6	10	18	17	80
	DC	9	4	5	19	33	6	10	18	17	80
	EC	9	19	5	4	33	6	18	10	17	80
	PR	9	4	5	19	33	6	10	18	17	80
Veri seti -3	BC	290	288	289	247	1	2629	2231	291	292	2631
	CC	288	247	290	717	720	289	291	1	292	473
	DC	290	289	288	1	291	2631	292	2532	2630	2629
	EC	290	288	247	473	497	514	530	588	598	627
	PR	290	289	288	1	291	2631	292	2532	2630	2629

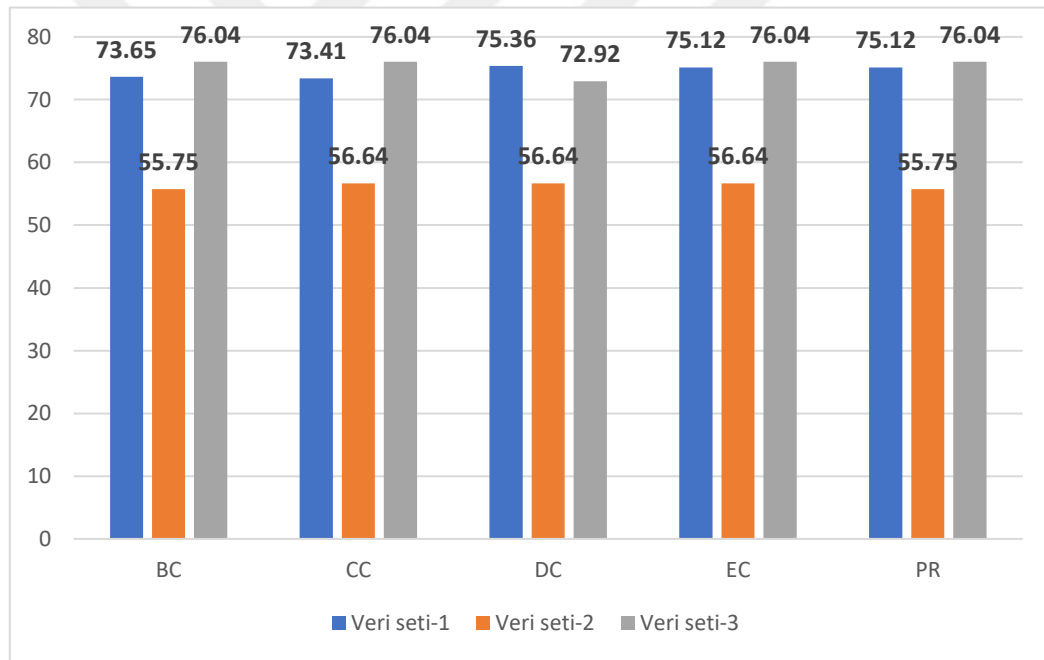
DC ve PR sonucuna göre en iyi düğüm olarak belirlenen 20 sayılı düğüm, BC değerine göre 3. sırada, CC değerine göre de 2. sırada yer almaktadır. Aynı şekilde Veri seti-2'ye ait sonuçlara bakıldığında 9 numaralı düğüm tüm sıralamada ilk sırada yer almaktadır. Veri seti-3'e ait sonuçlarda ise 290 sayılı düğüm CC sonucuna göre oluşturulan sıralama haricindeki tüm sıralamalarda ilk sırada yer almaktadır.

Veri seti-1 için yapılan işlemlerde düğümlere ait değerlerin elde edilmesi ve sıralanması sonrası bağışık olarak alınan ilk 41 düğüm ve enfekte olarak belirlenen rastgele 41 düğümün ağda yayılım sonucu oluşan ağın son hali Şekil 4.1'de ayrı ayrı gösterilmektedir. Şekil 4.1.'de her merkezîyet derecesine ait ağ yayılım görseli alt kısımlarda belirtilmiştir. Ayrıca yayılım sonucu bağışık kalan düğümler yeşil renkte, enfekte olan düğümler ise kırmızı renkte gösterilmektedir.



Şekil 4.1. Veri seti-1 üzerinde enfekte yayılım sonucu oluşan ağ yapısı

Şekil 4.2.'de, merkeziyet değerlerine göre ağdaki yayılım sonucu enfekte olan toplam düğüm sayılarının toplam popülasyon sayılarına oranı verilmektedir. Bu duruma göre Veri seti-1 için ağda bulunan toplam 410 düğüm arasında en yüksek BC değerine sahip olan ilk 41 düğümün bağışık olarak belirlenmesiyle birlikte gerçekleştirilen yayılım simülasyonu sonucunda enfekte düğüm oranı %73.65 (302 adet) olarak bulunmuştur. Enfekte düğümler aynı kalarak CC sonuçlarına göre en iyi ilk %10'luk düğümün bağışık olarak seçilmesiyle birlikte gerçekleştirilen simülasyon sonucu elde edilen toplam enfekte düğüm oranı %76,04 (301 adet) olarak bulunmuştur. Veri seti-1 için tüm merkezilik değerlerinin uygulanması sonucu elde edilen toplam enfekte düğüm sayılarına bakıldığında en az yayılımın CC, en fazla hastalık yayılımının ise DC değerlerinin uygulanmasında gerçekleştiği görülmektedir.



Şekil 4.2. Her veri seti için elde edilen enfekte birey oranı

Şekil 4.2'de turuncu renk ile temsil edilen Veri seti-2'ye ait sonuçlar incelendiğinde en iyi sonuç %55,75 ile BC ve PR değerleri kullanılarak elde edilmiştir. Gri renk ile gösterilen Veri seti-3'te ise %72,92 oran ile en iyi sonucu DC yöntemi elde etmiştir. Veri seti-2 ve Veri seti-3 kullanılarak elde edilen en yüksek enfekte değerleri sırasıyla %56,64 ve %76,04 ile elde edilmiştir.

4.2. Temel metriklerin TOPSIS yöntemi ile uygulanması

Önceki adımda tüm veri setleri için elde edilen metrik değerler, çok kriterli karar verme yöntemlerinden birisi olan TOPSIS yöntemine girdi olarak verilmektedir. TOPSIS yönteminde alternatifler kriterlere göre ideal sonuca en yakın ve ideal olmayan sonuçtan en uzak olacak şekilde sıralanmaktadır. TOPSIS yönteminin uygulanması sonucu elde edilen düğümlerin sıralaması sonucu ilk 10 sırada olan düğümler ve TOPSIS değerleri Çizelge 4.2’de gösterilmektedir. Çizelge 4.2’ye göre Veri seti-1 için en iyi düğüm olan 97 sayılı düğümün TOPSIS değeri 0.78886854 çıkmıştır. İkinci sırada 0.71776945 değeri ile 161 sayılı düğüm ve üçüncü sırada 0.67449341 değeri ile 20 sayılı düğüm bulunmaktadır. Çizelge 4.2’ye göre Veri seti-2 için en iyi düğüm olan 9 sayılı düğümün TOPSIS değeri 1 olarak görülmektedir. İkinci sırada 0.5267058 değeri ile 4 sayılı düğüm ve üçüncü sırada 0.4985809 değeri ile 5 sayılı düğüm bulunmaktadır. Veri seti-3 için en iyi düğüm olan 290 sayılı düğümün TOPSIS değeri 0.9991710 çıkmıştır. İkinci sırada 0.5716548 değeri ile 289 sayılı düğüm ve üçüncü sırada 0.5134861 değeri ile 288 sayılı düğüm bulunmaktadır.

Çizelge 4.2. Veri setlerine ait ilk 10 düğümün TOPSIS değeri

Veriseti-1			Veriseti -2		Veriseti -3	
Sıra	Düğüm Ad	TOPSIS Değer	Düğüm Ad	TOPSIS Değer	Düğüm Ad	TOPSIS Değer
1	97	0.7888685	9	1	290	0.999171
2	161	0.7177695	4	0.5267058	289	0.5716548
3	20	0.6744934	5	0.4985809	288	0.5134861
4	131	0.4167879	33	0.4975618	1	0.2834928
5	120	0.3407177	19	0.4684542	291	0.2078721
6	72	0.3176364	6	0.3741531	247	0.1825917
7	198	0.3129227	10	0.3730608	2631	0.1638996
8	158	0.3038756	18	0.3554861	292	0.1460799
9	256	0.2881011	17	0.3443001	2629	0.1138524
10	257	0.2828411	80	0.3440523	2231	0.1050855

TOPSIS yönteminin eşit ağırlıklar kullanılması sonucu elde edilen düğüm listesinin ağda düğüm merkezilik hesaplama yöntemleri kullanılarak elde edilen sıralamalar ile karşılaştırma yapabilmek amacıyla sonuçlar toplu şekilde ve veri

setlerine göre ayrı ayrı olarak Çizelge 4.3, Çizelge 4.4 ve Çizelge 4.5'te gösterilmektedir.

Çizelge 4.3. Veriseti-1 için yöntemlere göre elde edilen ilk 10 düğüm

Veri seti-1						
Sıra	TOPSIS	BC	CC	DC	EC	PR
1	97	97	97	20	256	20
2	161	161	20	161	259	161
3	20	20	131	198	254	120
4	131	131	72	120	257	72
5	120	72	65	175	158	198
6	72	66	161	69	258	19
7	198	233	51	158	120	358
8	158	120	130	139	255	314
9	256	130	66	256	280	304
10	257	296	74	72	282	14

Veri seti-1'e ait değerler kullanılarak oluşturulan Çizelge 4.1'de görüldüğü üzere BC ve CC sonuçlarına göre yapılan sıralamada ilk sırada bulunan 97 sayılı düğüm TOPSIS yöntemi sonucunda da ilk sırada çıkmıştır.

Çizelge 4.4. Veriseti-2 için yöntemlere göre elde edilen ilk 10 düğüm

Veri seti-2						
Sıra	TOPSIS	BC	CC	DC	EC	PR
1	9	9	9	9	9	9
2	4	4	4	4	19	4
3	5	33	5	5	5	5
4	33	5	19	19	4	19
5	19	19	33	33	33	33
6	6	10	6	6	6	6
7	10	6	10	10	18	10
8	18	18	18	18	10	18
9	17	80	17	17	17	17
10	80	17	80	80	80	80

DC ve PR yönteminin uygulanması sonucu elde edilen değerlere göre yapılan sıralamada ilk sırada çıkan 20 sayılı düğüm TOPSIS yönteminin uygulanması sonucu oluşan sıralamada en iyi 3. düğüm olarak bulunmuştur. İki farklı merkezilik değerlerinde ilk sırada bulunan 97 ve 20 sayılı düğümlerin TOPSIS sıralamasında farklı çıkmasının temel nedeni, TOPSIS yönteminin tüm kriterlere ait değerleri dikkate alarak sıralama yapmasından kaynaklanmaktadır.

Kısaca iki farklı sıralamada ilk sırada olan düğümün diğer kriterlerdeki sıralaması da dikkate alınınca TOPSIS sıralamasındaki yeri aşağı ya da yukarı yönde değişebilmektedir.

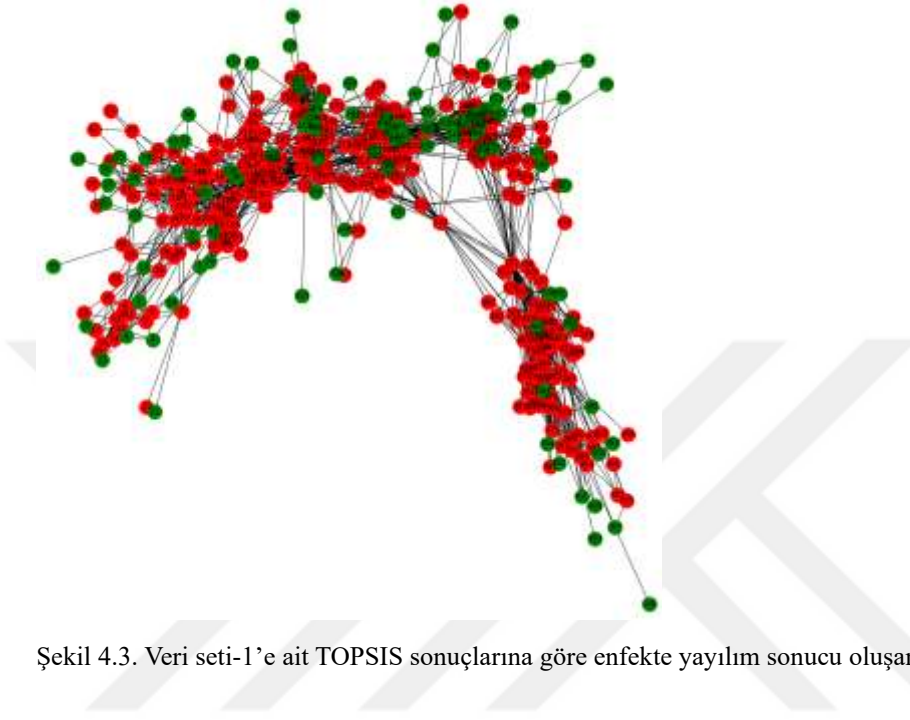
Veri seti-2'ye ait değerler kullanılarak oluşturulan Çizelge 4.4'te görüldüğü üzere BC, CC, DC, EC ve PR sonuçlarına göre yapılan sıralamada ilk sırada bulunan 9 sayılı düğüm TOPSIS yöntemi sonucunda da ilk sırada çıkmıştır. EC sonuçlarına göre yapılan sıralama hariç tüm sıralamalarda ikinci sırada çıkan 4 sayılı düğüm, TOPSIS yönteminin uygulanması sonucu oluşan sıralamada da en iyi ikinci düğüm olarak bulunmuştur.

Çizelge 4.5 Veriseti-3 için yöntemlere göre elde edilen ilk 10 düğüm

Veri seti-3						
Sıra	TOPSIS	BC	CC	DC	EC	PR
1	290	290	288	290	290	290
2	289	288	247	289	288	289
3	288	289	290	288	247	288
4	1	247	717	1	473	1
5	291	1	720	291	497	291
6	247	2629	289	2631	514	2631
7	2631	2231	291	292	530	292
8	292	291	1	2532	588	2532
9	2629	292	292	2630	598	2630
10	2231	2631	473	2629	627	2629

Veri seti-3'e ait değerler kullanılarak oluşturulan Çizelge 4.5'te görüldüğü üzere BC, DC, EC ve PR sonuçlarına göre yapılan sıralamada ilk sırada bulunan 290 sayılı düğüm TOPSIS yöntemi sonucunda da ilk sırada çıkmıştır. DC ve PR sonuçlarına göre yapılan sıralama ikinci sırada çıkan 289 sayılı düğüm, TOPSIS

yönteminin uygulanması sonucu oluşan sıralamada da en iyi ikinci düğüm olarak bulunmuştur. Veri seti-1 üzerinde yayılım sonucu elde edilen enfekte ve bağışık düğümlere ait ağ yapısı Şekil 4.3'te gösterilmektedir. Yeşil renkler bağışık olan düğümleri, kırmızı renkler ise enfekte olan düğümleri temsil etmektedir.

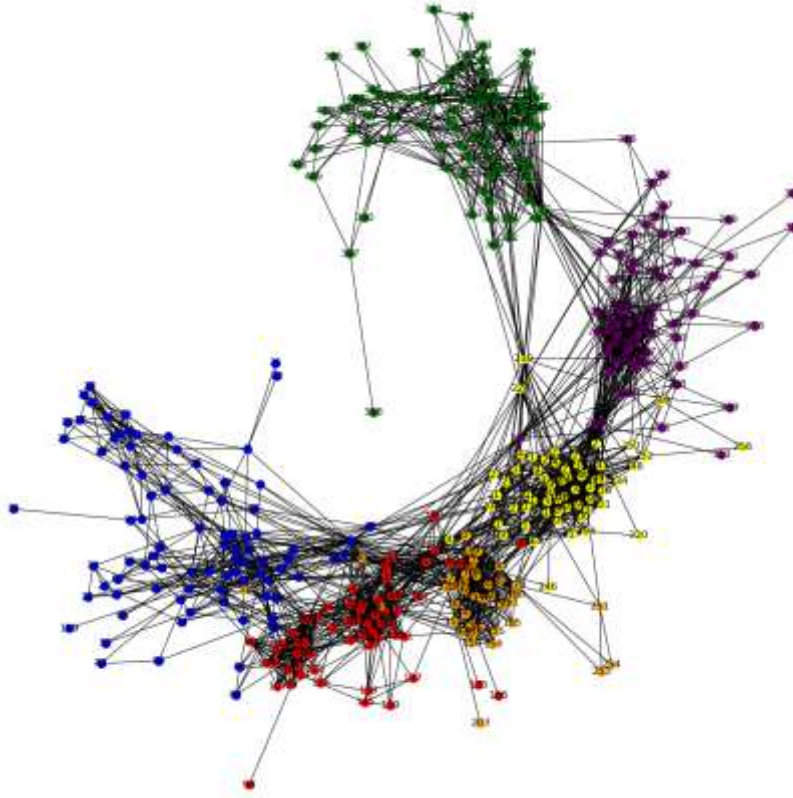


Şekil 4.3. Veri seti-1'e ait TOPSIS sonuçlarına göre enfekte yayılım sonucu oluşan ağ yapısı

Yapılan yayılım simülasyonu sonucunda enfekte düğüm oranı Veri seti-1 için %73,9 (303 adet düğüm), Veri seti-2 için %55,75 (63 adet düğüm) ve Veri seti-3 için %76,04 (2196 düğüm) olarak bulunmuştur. TOPSIS yönteminin kullanılmasıyla elde edilen sonuçlara bakıldığında Veri seti-1 için BC ve CC sonuçlarından sonra üçüncü en iyi sonuç olduğu görülmektedir. Veri seti-2 sonucuna bakıldığında BC ve PR sonuçları ile aynı çıkmıştır. Veri seti-3'te ise CC sonuçları hariç diğer tüm metrikler ile aynı oranda çıkmıştır.

4.3. Louvain yöntemi ve TOPSIS yönteminin uygulanması

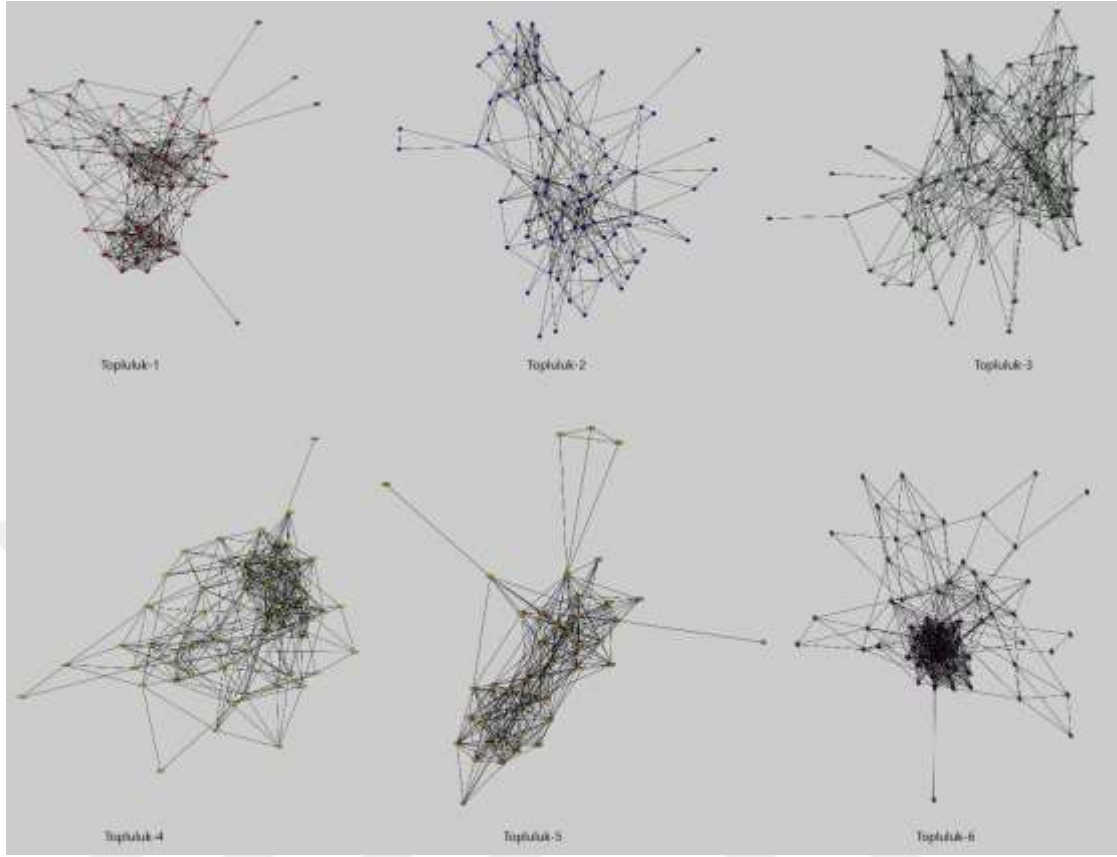
Bu aşamada, topluluk tespit yöntemlerinden birisi olan Louvain yöntemi kullanılarak veri seti tarafından oluşturulan ağın alt toplulukları ortaya çıkarılmıştır. Louvain yöntemi ile ağ içerisindeki benzer niteliklere sahip düğümler gruplanmakta ve topluluk haline getirilmektedir. Veri seti-1 ile oluşturulan ağa ait alt topluluklar Şekil 4.4.'te her topluluk farklı renkte olacak şekilde gösterilmektedir.



Şekil 4.4. Veri seti-1'e ait toplulukların gösterimi

Louvain yönteminin Veri seti-1, Veri seti-2 ve Veri seti-3'e uygulanması sonucu Veri seti-1'de toplamda 6 adet alt topluluk ağı, Veri seti-2'de 5 adet alt topluluk ağı ve Veri seti-3'te 8 adet alt topluluk ağı elde edilmiştir.

Alt topluluklar sırayla numaralandırılmıştır. Veri seti-1 için Topluluk-1, Topluluk-2, Topluluk-3, Topluluk-4, Topluluk-5 ve Topluluk-6 alt toplulukları sırasıyla 63, 87, 86, 61, 41 ve 72 adet düğüme sahiptir. Veri seti-2 için Topluluk-1, Topluluk-2, Topluluk-3, Topluluk-4 ve Topluluk-5 alt toplulukları sırasıyla 35, 4, 29, 23 ve 22 adet düğüme sahiptir. Veri seti-3 için Topluluk-1, Topluluk-2, Topluluk-3, Topluluk-4, Topluluk-5, Topluluk-6, Topluluk-7 ve Topluluk-8 alt toplulukları sırasıyla 284, 315, 59, 465, 707, 757, 98 ve 203 adet düğüme sahiptir. Şekil 4.5.'te Veri seti-1'e ait her alt topluluğun çizgesi ayrı ayrı sıralı şekilde gösterilmektedir. 41 düğüm sayısı ile alt topluluklar arasında en az düğüme sahip olan Topluluk-5 Şekil 4.5.'te turuncu renkte ve düğümlerin oluşturduğu çizge ayrı olarak Şekil 4.5.'te 5. sırada gösterilmektedir. 87 düğüm sayısı ile en fazla düğüme sahip olan Topluluk-2 çizgesi Şekil 4.5'te mavi renkte ve düğümlerin oluşturduğu çizge ayrı olarak Şekil 4.5'te 2. sırada gösterilmektedir.



Şekil 4.5. Veri seti-1'e ait alt toplulukların gösterimi

Ağın alt topluluklara bölünmesiyle elde edilen alt topluluklar üzerinde ilk olarak merkezilik değerlerini elde etme işlemi gerçekleştirilmiştir. Her topluluğa ait düğümler için elde edilen metrikler TOPSIS yöntemine girdi olarak verilmiştir. Eşit ağırlıklı TOPSIS yönteminin uygulanması sonucu elde edilen sonuçta en yüksek TOPSIS değerine sahip ilk 10 düğüme ve en düşük TOPSIS değerine sahip ilk 10 düğüme ait TOPSIS değerleri, düğümler ve içerisinde bulundukları topluluk sınıfları Veri seti-1 için Çizelge 4.6'da, Veri seti-2 için Çizelge 4.7'de ve Veri seti-3 için Çizelge 4.8'de gösterilmektedir. Çizelge 4.6 incelendiğinde ilk 10 düğümün 6 tanesi 2 numaralı topluluğa 4 tanesi ise 3 numaralı topluluğa aittir. 0.852345381 TOPSIS değeri ile ilk sırada bulunan 3 numaralı düğüm Topluluk-2 adlı topluluğa aittir.

Çizelge 4.6. TOPSIS sonuçlarına göre Veri seti-1'e ait ilk 10 düğüm

Veri seti-1			
Sıra	Düğüm Ad	Topluluk Sınıfı	TOPSIS
1	3	2	0.852345381
2	70	2	0.826212574
3	383	3	0.678138062
4	347	3	0.665649323
5	272	3	0.613668507
6	27	2	0.601463303
7	407	3	0.596279033
8	88	2	0.589871778
9	49	2	0.589262893
10	59	2	0.566572631

Veri seti-2'ye ait çıktılarla oluşturulan Çizelge 4.7' ye bakıldığında ilk 10 düğümün 4 tanesi 1 numaralı topluluğa 3 tanesi ise 2 numaralı ve 3 numaralı topluluğa aittir. 0.600008503 TOPSIS değeri ile ilk sırada bulunan 61 numaralı düğüm Topluluk-2 adlı topluluğa aittir.

Çizelge 4.7. TOPSIS sonuçlarına göre Veri seti-2'ye ait ilk 10 düğüm

Veri seti -2			
Sıra	Düğüm Ad	Topluluk Sınıfı	TOPSIS
1	61	2	0.600008503
2	112	3	0.50147
3	3	2	0.47034578
4	68	2	0.470345784
5	18	3	0.465817813
6	107	1	0.417683884
7	100	1	0.391543433
8	75	1	0.36872007
9	83	3	0.361998973
10	19	1	0.355831352

8 farklı alt topluluğa sahip Veri seti-3'e ait Çizelge 4.8 incelendiğinde ilk 10 düğüm listesinde her topluluktan en az bir adet düğüm bulunmaktadır. 0.999021821 TOPSIS değeri ile ilk sırada yer alan 603 sayılı düğüm Topluluk-6 ya, ikinci sırada

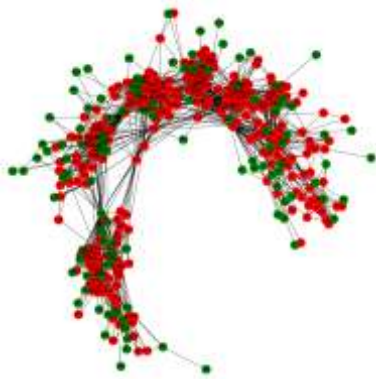
bulunan 1709 sayılı düğüm Topluluk-5'e, üçüncü sırada bulunan düğüm ise Topluluk-4'e aittir.

Çizelge 4.8. TOPSIS sonuçlarına göre Veri seti-3'e ait ilk 10 düğüm

Veri seti -3

Sıra	Düğüm Ad	Topluluk Sınıfı	TOPSIS
1	603	6	0.999021821
2	1709	5	0.905311177
3	288	4	0.559236228
4	1	1	0.411753061
5	2674	2	0.338344612
6	335	7	0.336329948
7	2570	3	0.327874888
8	1524	8	0.327181705
9	2678	2	0.205982655
10	2697	8	0.107756829

Her veri seti için TOPSIS yöntemi sonucu elde edilen sıralamadaki ilk 41 düğüm bağışık olarak alınarak enfekte yayılım simülasyon işlemi gerçekleştirilmiştir. Veri seti-1 için yapılan simülasyon sonucu enfekte yayılım oranı %72,92 (299 adet düğüm) olarak bulunmuştur. Veri seti-1 için yapılan simülasyon sonucu oluşan enfekte, bağışık düğümleri gösteren çizge yapısı Şekil 4.6.'da gösterilmektedir. Veri seti-2 için yapılan simülasyon sonucu enfekte yayılım oranı %54,86 (62 adet düğüm), Veri seti-3 için yapılan simülasyon sonucu enfekte yayılım oranı % 74,96 (2165 adet düğüm) olarak bulunmuştur.



Şekil 4.6. Veri seti-1 için Louvian yöntemi sonucu oluşan enfekte, bağışık düğüm dağılımı

4.4. Node2Vec yöntemi ve TOPSIS yönteminin uygulanması

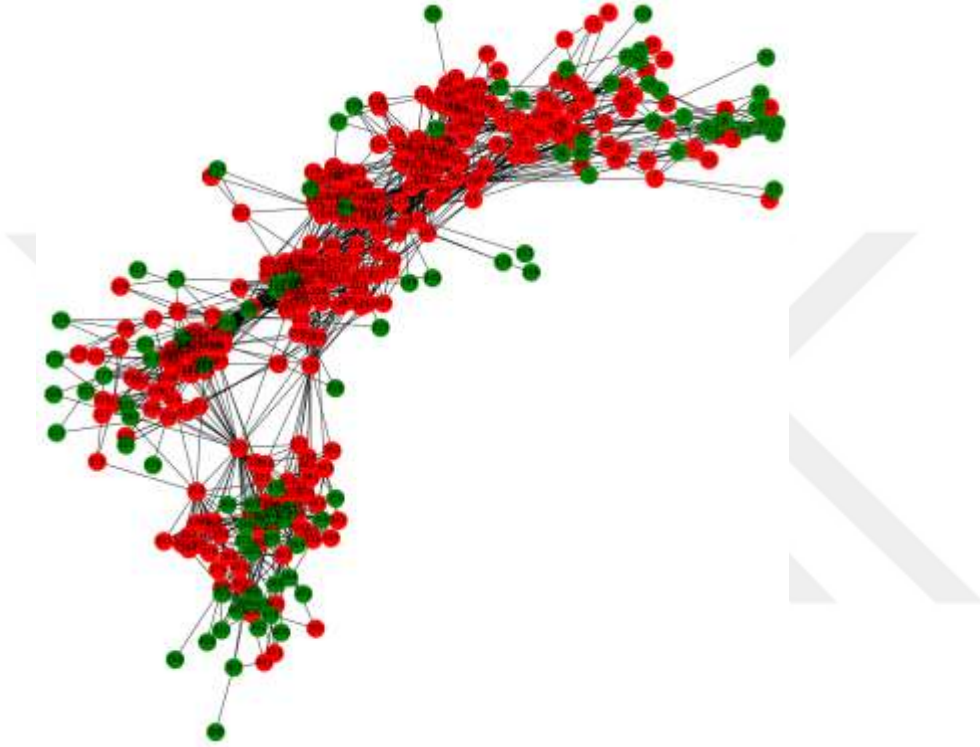
Node2vec algoritmasında kullanılan düğüm özellikleri, yürüyüş uzunluğu ve toplam yürüyüş sayısı parametreleri önemli parametrelerdir. Ayrıca Node2Vec tarafından özellik hesaplaması sonrası model için uyumlu hale getirme aşamasında kullanılan pencere sayısını da ekleyebiliriz. Yapılan çalışmada boyut sayısı olarak 16, yürüyüş uzunluğu olarak 100, toplam yürüyüş sayısı olarak 1000 ve pencere boyutu olarak 7 kullanılmıştır. Ayrıca çıktıyı model nesnesine uyumlu hale getirme aşamasında tüm düğümlerin dikkate alınması için "min_count" parametresine 1 değeri verilmiştir. Tüm parametre değerlerinin tanımlanması ve Node2Vec yönteminin kullanılması sonucu Veri seti-1 için 16x410 boyutlu, Veri seti-2 için 16x113 boyutlu ve Veri seti-3 için 16x2888 boyutlu matrisler elde edilmiştir. Elde edilen matrisler daha sonra TOPSIS yöntemine girdi olarak verilmiştir. Tüm veri setleri için TOPSIS sonucu elde edilen düğüm sıralamasına göre ilk 10 sırada bulunan düğümler Çizelge 4.9'da gösterilmektedir. Çizelge 4.9'a göre Veri seti-1 için 402 sayılı düğüm, Veri seti-2 için 2 sayılı düğüm ve Veri seti-3 için 603 sayılı düğüm ilk sırada bulunmaktadır.

Çizelge 4.9. Veri setleri için Node2Vec yöntemi ile elde edilen ilk 10 düğüm listesi

Sıra	Veri seti-1	Veri seti-2	Veri seti-3
1	402	2	603
2	394	20	2261
3	391	107	2325
4	405	18	2265
5	398	87	2288
6	388	48	2263
7	403	9	2308
8	406	33	2247
9	397	73	2291
10	346	84	2306

Düğüm sıralarının elde edilmesi aşamasından sonra önceki simülasyon için gerekli olan işlem basamakları gerçekleştirilmiştir. Bu aşamada Veri seti-1 için ilk 41 düğüm, Veri seti-2 için ilk 11 düğüm ve Veri seti-3 için ilk 288 düğüm bağışık olarak alınarak ve rastgele seçim sonucu aynı sayıda düğüm enfekte olarak alınarak

simülasyon işlemi gerçekleştirilmiştir. Veri seti-1 ağında simülasyon işlemi sonrasında oluşan sonuca göre enfekte düğüm oranı %76,58 (314 adet düğüm) olarak bulunmuştur. Ayrıca Veri seti-2 için bu oran % 86,72 (93 adet düğüm) ve Veri seti-3 için %79,98 (2310 adet düğüm) olarak bulunmuştur. Yayılm simülasyonu sonucu Veri seti-1'e ait enfekte ve bağışık düğümlerin işaretlenmesi ile oluşturulan çizge Şekil 4.7.'de gösterilmektedir.

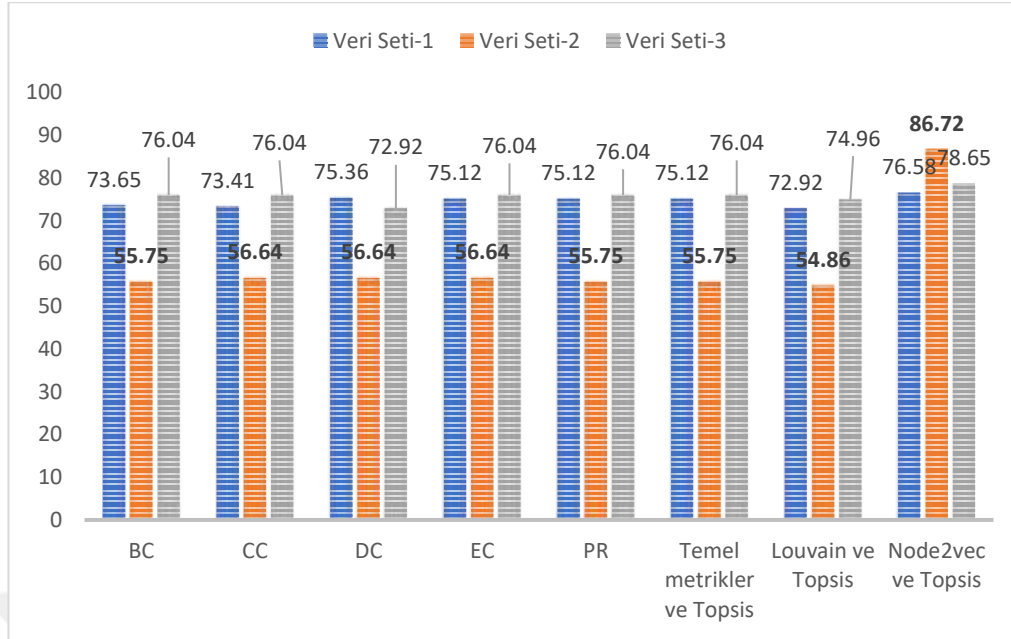


Şekil 4.7. Node2Vec yöntemi sonucu elde edilen çizge

4.5. Sonuçların karşılaştırılması

Bu alt bölümde, yapılan örnek simülasyonlar sonucunda elde edilen değerler ve sonuçlar toplu şekilde, karşılaştırmalı olarak sunulmakta ve özetlenmektedir.

Şekil 4.8'de her veri seti ve uygulama için ayrı ayrı olmak üzere elde edilen enfekte düğüm oranları gösterilmektedir. Veri seti-1 (mavi renk) üzerinde yapılan uygulamalarda en başarılı sonuç Louvain ve TOPSIS yöntemlerinin birarada kullanılmasıyla elde edilmiştir. Veri seti-2 (turuncu renk) için en başarılı sonuca Louvain ve TOPSIS yöntemlerinin birlikte kullanılmasıyla elde edilmiştir. Veri seti-3 (gri renk) için derece merkeziliği yöntemi en başarılı sonucu üretmiştir.



Şekil 4.8. Simülasyonlar sonucu veri setlerine ait enfekte düğüm oranları

Çizelge 4.10.'da Veri seti-1 kullanılarak elde edilen sıralamalara ait ilk 10 düğüm gösterilmektedir. Veri seti-1 için BC'ye göre elde edilen sıralama sonucu %73,65 enfekte oranı, CC'ye göre elde edilen sıralama sonucu %73,41 enfekte oranı, DC'ye göre elde edilen sıralama sonucu %75,36 enfekte oranı, EC'ye göre elde edilen sıralama sonucu %75,12 enfekte oranı, PR'ye göre elde edilen sıralama sonucu %75,12 enfekte oranı elde edilmiştir. Daha sonra düğümlerin alternatif ve merkezilik değerlerinin kriter olarak TOPSIS yönteminde kullanılmasıyla her düğüm için TOPSIS değerleri elde edilmiştir. TOPSIS değerlerine göre sıralama yapılarak ilk %10'luk düğüm bağışık olarak seçilmiştir. Bağışık düğümlerin haricinde kalan düğümler arasından da aynı oranda rastgele düğümler seçilerek enfekte olarak işaretlenmiştir. Yapılan simülasyon sonucu enfekte düğüm oranı %75,12 olarak bulunmuştur. Louvain yöntemi ile mevcut ağ alt topluluklara bölünerek düğümlerin merkezilik değerleri tekrardan hesaplandı. Elde edilen değerler TOPSIS yöntemi için girdi olarak kullanılmıştır. TOPSIS değerlerine göre yapılan sıralamaya göre bağışık düğümler seçilmiş ve simülasyon tekrarlanmıştır. Yapılan simülasyon sonucu enfekte düğüm oranı olarak %72,92 bulunmuştur. Son olarak Node2Vec yöntemi ile düğümlere ait toplamda 16 öznitelik çıkartılmış ve TOPSIS yöntemine girdi olarak verilmiştir. Yapılan işlem sonucu oluşan sıralamadaki ilk %10'luk düğüm bağışık

olarak işaretlenerek simülasyon işlemi gerçekleştirilmiştir. İşlem sonrası enfekte düğüm oranı %76,58 olarak bulunmuştur.

Çizelge 4.10. Veri seti-1 için tüm yöntemlerde elde edilen ilk 10 düğüm listesi

Veri seti-1								Node2Vec
Sıra	BC	CC	DC	EC	PR	TOPSIS	Louvain ve TOPSIS	ve TOPSIS
1	97	97	20	256	20	97	3	402
2	161	20	161	259	161	161	70	394
3	20	131	198	254	120	20	383	391
4	131	72	120	257	72	131	347	405
5	72	65	175	158	198	120	272	398
6	66	161	69	258	19	72	27	388
7	233	51	158	120	358	198	407	403
8	120	130	139	255	314	158	88	406
9	130	66	256	280	304	256	49	397
10	296	74	72	282	14	257	59	346

Çizelge 4.11’de Veri seti-2 kullanılarak elde edilen sıralamalara ait ilk 10 düğüm gösterilmektedir. Veri seti-2 için BC’ye göre elde edilen sıralama sonucu %55,75 enfekte oranı, CC’ye göre elde edilen sıralama sonucu %56,64 enfekte oranı, DC’ye göre elde edilen sıralama sonucu %56,64 enfekte oranı, EC’ye göre elde edilen sıralama sonucu %56,64 enfekte oranı, PR’ye göre elde edilen sıralama sonucu %55,75 enfekte oranı elde edilmiştir. Daha sonra düğümlerin alternatif ve merkezilik değerlerinin ise kriter olarak TOPSIS yönteminde kullanılmasıyla her düğüm için TOPSIS değerleri elde edilmiştir. TOPSIS değerlerine göre sıralama yapılarak ilk %10’luk düğüm bağışık olarak seçilmiştir. Bağışık düğümlerin haricinde kalan düğümler arasından da aynı oranda rastgele düğümler seçilerek enfekte olarak işaretlenmiştir. Yapılan simülasyon sonucu enfekte düğüm oranı %55,75 olarak bulunmuştur. Louvain yöntemi ile mevcut ağ alt topluluklara bölünerek düğümlerin merkezilik değerleri tekrardan hesaplandı. Elde edilen değerler TOPSIS yöntemi için girdi olarak kullanılmıştır. TOPSIS değerlerine göre yapılan sıralamaya göre bağışık düğümler seçilmiş ve simülasyon tekrarlanmıştır. Yapılan simülasyon sonucu enfekte düğüm oranı olarak %54,86 bulunmuştur. Son olarak Node2Vec yöntemi ile

düğömlere ait toplamda 16 öznitelik çıkartılmış ve TOPSIS yöntemine girdi olarak verilmiştir. Yapılan işlem sonucu oluşan sıralamadaki ilk %10'luk düğüm bağışık olarak işaretlenerek simölasyon işlemi gerçekleştirilmiştir. İşlem sonrası enfekte düğüm oranı %86,72 olarak bulunmuştur.

Çizelge 4.11. Veri seti-2 için tüm yöntemlerde elde edilen ilk 10 düğüm listesi

Veri seti-2								
Sıra	BC	CC	DC	EC	PR	TOPSIS	Louvain ve TOPSIS	Node2Vec ve TOPSIS
1	9	9	9	9	9	9	61	2
2	4	4	4	19	4	4	112	20
3	33	5	5	5	5	5	3	107
4	5	19	19	4	19	33	68	18
5	19	33	33	33	33	19	18	87
6	10	6	6	6	6	6	107	48
7	6	10	10	18	10	10	100	9
8	18	18	18	10	18	18	75	33
9	80	17	17	17	17	17	83	73
10	17	80	80	80	80	80	19	84

Çizelge 4.12'de Veri seti-3 kullanılarak elde edilen sıralamalara ait ilk 10 düğüm gösterilmektedir. Veri seti-3 için BC'ye göre elde edilen sıralama sonucu %76,04 enfekte oranı CC'ye göre elde edilen sıralama sonucu %76,04 enfekte oranı, DC'ye göre elde edilen sıralama sonucu %72,92 enfekte oranı, EC'ye göre elde edilen sıralama sonucu %76,04 enfekte oranı, PR'ye göre elde edilen sıralama sonucu %76,04 enfekte oranı elde edilmiştir. Daha sonra düğömlerin alternatif ve merkezilik değerlerinin ise kriter olarak TOPSIS yönteminde kullanılmasıyla her düğüm için TOPSIS değerleri elde edilmiştir. TOPSIS değerlerine göre sıralama yapılarak ilk %10'luk düğüm bağışık olarak seçilmiştir. Bağışık düğömlerin haricinde kalan düğömler arasından da aynı oranda rastgele düğömler seçilerek enfekte olarak işaretlenmiştir. Yapılan simölasyon sonucu enfekte düğüm oranı %76,04 olarak bulunmuştur. Louvain yöntemi ile mevcut ağı alt topluluklara bölünerek düğömlerin merkezilik değerleri tekrardan hesaplandı. Elde edilen değerler TOPSIS yöntemi için girdi olarak kullanılmıştır. TOPSIS değerlerine göre yapılan sıralamaya göre bağışık düğömler seçilmiş ve simölasyon tekrarlanmıştır. Yapılan simölasyon sonucu enfekte düğüm oranı olarak %74,96 bulunmuştur. Son olarak Node2Vec yöntemi ile

düğömlere ait toplamda 16 öznitelik çıkartılmış ve TOPSIS yöntemine girdi olarak verilmiştir. Yapılan işlem sonucu oluşan sıralamadaki ilk %10'luk düğüm bağışık olarak işaretlenerek simölasyon işlemi gerçekleştirilmiştir. İşlem sonrası enfekte düğüm oranı %78,65 olarak bulunmuştur.

Çizelge 4.12. Veri seti-3 için tüm yöntemlerde elde edilen ilk 10 düğüm listesi

Veri seti-3								
Sıra	BC	CC	DC	EC	PR	TOPSIS	Louvain ve TOPSIS	Node2Vec ve TOPSIS
1	290	288	290	290	290	290	603	603
2	288	247	289	288	289	289	1709	2261
3	289	290	288	247	288	288	288	2325
4	247	717	1	473	1	1	1	2265
5	1	720	291	497	291	291	2674	2288
6	2629	289	2631	514	2631	247	335	2263
7	2231	291	292	530	292	2631	2570	2308
8	291	1	2532	588	2532	292	1524	2247
9	292	292	2630	598	2630	2629	2678	2291
10	2631	473	2629	627	2629	2231	2697	2306

5. SONUÇLAR ve ÖNERİLER

Epidemiyolojide bireysel niteliklerin dikkate alındığı uygulamalar yakın zamana kadar hesaplama açısından fazla maliyetli ve gereksiz olarak görülmekteydi (Bansal ve ark., 2007). Fakat günümüz gelişmeleri sayesinde heterojen popülasyonlara yayılmış hastalığı modellemek için analitik yaklaşımların geliştirilmesi artmaktadır. Bu varsayım nedeniyle ortaya çıkan sınırlılığın sonuçlara olan olumsuz etkisini ortadan kaldırmak amacıyla literatürde geniş bir uygulama alanı bulunan Louvain, TOPSIS ve Node2Vec yöntemleri kullanılarak örnek hastalık yayılımı simülasyonları gerçekleştirilmiştir.

Elde edilen sonuçlara göre, Veri seti-1 ve Veri seti-2 için topluluk ağının Louvain yöntemi ile alt ağlarının tespit edilmesi ve elde edilen alt topluluklara ait düğümlerin ağdaki konumlarına göre BC, CC, DC, EC ve PR değerlerinin hesaplanması ve bu değerlerin TOPSIS yöntemine girdi olarak verilmesi sonucu elde edilen TOPSIS sıralamasının, hastalığın yayılmasını durdurma amaçlı yapılacak faaliyetlerde (örneğin aşılama) dikkate alınmasının kullanılan diğer yöntemlere göre başarılı sonuçlar vereceği görülmüştür. Veri seti-3 için ise bu durum az fark ile derece merkeziliği yöntemi olarak karşımıza çıkmaktadır. Çalışmada özellikle TOPSIS algoritmasının alternatif çözüm yöntemleri ile bütünleştirilmektedir. Bu sayede yöntemlerde dikkate alınmak istenilen öznelilik sayıları kolay şekilde arttırılabilir ya da azaltılabilir hale getirildiği için düğümlere ait farklı öznelilik değerleri de rahatlıkla kullanılabilir.

Yapılan çalışmada karşılaşılan zorluklar ve gelecekte yapılacak çalışmalara yardımcı olması bakımından, TOPSIS yöntemine girdi olarak kullanılan kriterlerin ağırlıklandırılması ve Node2Vec yönteminde kullanılan parametrelere ait değer seçimleri üzerine yapılacak iyileştirme çalışmaları ile mevcut sonuçların daha başarılı olacağı düşünülmektedir.

KAYNAKLAR

- AHMAD, A., TANVIR, A., MUSHEER, A., CHANCHAL, K., FAYADH, A., and MAJID, N., 2023. A Complex Network-Based Approach for Security and Governance in the Smart Green City. *Expert Systems with Applications*, 214(Mart):1-12.
- ALDABOBI, A., AHMAD, S., and RIAD, J., 2022. An Improved Louvain Algorithm Based on Node Importance for Conformity Detection. 100:23-39.
- ANNAS, SUWARDI, MUH ISBAR PRATAMA, MUH RIFANDI, WAHIDAH SANUSI and SYAFRUDDIN SIDE. 2020. Stability Analysis and Numerical Simulation of SEIR Model for Pandemic COVID-19 Spread in Indonesia. *Chaos, Solitons & Fractals*, 139: 1100:1172.
- AZIZI, A., CESAR, M., BALTAZAR, E., YUN, Y., and CARLOS. C., C., 2020. Epidemics on Networks: Reducing Disease Transmission Using Health Emergency Declarations and Peer Communication. *Infectious Disease Modelling*, 5(Ocak):12–22.
- BABAY, A., E., MICHAEL, D., ARAVIND, S., LEONIDAS, T., and ANIL, V., 2022. Controlling Epidemic Spread Using Probabilistic Diffusion Models on Networks. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, 11:641–654.
- BANSAL, S., BRYAN, T., G., and LAUREN, A., M., 2007. When Individual Behaviour Matters: Homogeneous and Network Models in Epidemiology. *Journal of The Royal Society Interface*, 4:879-891.
- BARABÁSI, A., L., and RÉKA, A., 1999. Emergence of Scaling in Random Networks. *Science*, 286(5438):509–512.
- BARTHÉLEMY, M., 2004. Betweenness Centrality in Large Complex Networks. *The European Physical Journal B*, 38(2):163–168.
- BARTHÉLEMY, M., ALAIN, B., ROMUALDO, P., S., and ALESSANDRO, V., 2005. Dynamical Patterns of Epidemic Outbreaks in Complex Heterogeneous Networks. *Journal of Theoretical Biology*, 235(2): 275–288.
- BEAUCHAMP, M., A., 1965. An Improved Index of Centrality. *Behavioral Science*, 10(2):161–163.
- BEN EL KOUNI, I., WAFA, K., and LOTFI, B., R., 2020. Node Importance Based Label Propagation Algorithm for Overlapping Community Detection in Networks. *Expert Systems with Applications*, 162(Aralık):113-120.
- BIGGS, N., E., KEITH, L., and ROBIN, J., W., 1986. *Graph Theory*. Oxford University Press, 1:1736-1936.
- BLONDEL, V., D., JEAN-LOUP, G., RENAUD, L., and ETIENN, L., 2008. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment*, 10(3): 100-108.
- BOLLAND, J., M., 1988. Sorting out Centrality: An Analysis of the Performance of Four Centrality Models in Real and Simulated Networks. *Social Networks*, 10: 233–253.
- BOLLOBAS, B., 1998. *Modern Graph Theory*. Springer Science & Business Media, 1:1-93.
- BONACICH, P., and PAULETT, L., 2001. Eigenvector-like Measures of Centrality for Asymmetric Relations. *Social Networks*, 23(3): 191–201.
- BORGATTI, S., P., and MARTIN, G., E., 2006. A Graph-Theoretic Perspective on Centrality. *Social Networks*, 28(4): 466–84.
- BOUYER, A., HAMID A., B., BAHMAN, A., ZAHRA, A., and REZA, G., 2023. FIP: A Fast-Overlapping Community-Based Influence Maximization Algorithm Using Probability Coefficient of Global Diffusion in Social Networks. *Expert Systems with Applications*, 213(Mart): 180-201.
- BRIN, S., and LAWRENCE, P., 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems, Proceedings of the Seventh International World Wide Web Conference*, 30(1): 107–117.

- CARROLL, L., N., ALAN, P., A., LANDON, T., D., TSUNG-CHIEH, F., IAN, S., P., and NEIL, F. A., 2014. Visualization and Analytics Tools for Infectious Disease Epidemiology: A Systematic Review. *Journal of Biomedical Informatics*, 51(Ekim): 287–98.
- CHAHARBORJ, S., S., KHONDOKER, N., N., KOO, L., F., SHAHRIAR, S., C., and PEI S., P., 2022. Controlling COVID-19 Transmission with Isolation of Influential Nodes. *Chaos, Solitons & Fractals* 159(Haziran): 112:135.
- CHANG, L., WANG, J., and XIA, R., 2023. Node importance evaluation in multi-platform avionics architecture based on TOPSIS and PageRank. *EURASIP Journal on Advances in Signal Processing*, 27:143-160.
- CHEN, J., STEFAN, H., ACHLA, M., HENNING, M., BRYAN, L., SRINIVASAN V., and ARASH, H., 2022. Effective Social Network-Based Allocation of COVID-19 Vaccines. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4675–83.
- CHEN, X., AIYIN, Z., HUI, W., ADAM, G., and XIAOLIN, Z., 2021. Compliance and Containment in Social Distancing: Mathematical Modeling of COVID-19 across Townships. *International Journal of Geographical Information Science*, 35(3): 446–465.
- CLAUSET, A., NEWMAN, M., E., J., and MOORE, C., 2004. Finding Community Structure in Very Large Networks. *Physical Review E*, 70(6):66-111.
- CORDASCO, G., and GARGANO, L., 2010. Community Detection via Semi-Synchronous Label Propagation Algorithms. *Business Applications of Social Network Analysis (BASNA)*, 10:1–8.
- CORUM, J., GRADY, D., WEE S., and ZIMMER, C., 2020. Coronavirus Vaccine Tracker. *The New York Times*, 5:1-8.
- DERÉNYI, I., PALLA, G., and VICSEK, T., 2005. Clique Percolation in Random Networks. *Physical Review Letters* 94(16): 160-202.
- DIEKMANN, O., DE JONG, M., C., M., and METZ, J., A., J., 1998. A Deterministic Epidemic Model Taking Account of Repeated Contacts between the Same Individuals. *Journal of Applied Probability*, 35(2): 448–462.
- ERDÖS, P., and RÉNYI, A., 1960. On the Evolution of Random Graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1): 17–60.
- EULER, L., 1741. *Solutio Problematis Ad Geometriam Situs Pertinentis. Commentarii Academiae Scientiarum Petropolitanae*, 17:128–140.
- EVANS, T., S., and CHEN, B., 2022. Linking the network centrality measures closeness and degree. *Communications Physics*, 5:172-185.
- FREEMAN, L., C., 1977. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1):35–41.
- FREEMAN, L., C., ROEDER, D., and MULHOLLAND, R., R., 1979. Centrality in Social Networks Experimental Results. *Social Networks*, 2(2): 119–141.
- GAO, S., MA, J., CHEN, Z., WANG, G., and XING, C., 2014. Ranking the Spreading Ability of Nodes in Complex Networks Based on Local Structure. *Physica A: Statistical Mechanics and Its Applications*, 403(Haziran): 130–147.
- GROVER, A., and LESKOVEC, J., 2016. Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 22:855–864.
- GÜRSAKAL, N., TÜZÜNTÜRK, S., ve SERT, F., 2014. Sosyal Ağ Verilerinin Kuvvet Yasası Olasılık Dağılımına Uygunluk Analizi: Twitter Örneği, 15. Uluslararası Ekonometri, Yöneylem ve İstatistik Sempozyumu Bildiriler Kitabı, 23-28 Ekim, Cilt 1, Isparta, s.464–482.
- GÜZEY, N., 2021. Türkiye’deki COVID-19 yayılımının SIR temelli modellenmesinde RSS yöntemi ile parametre kestirimi. *Gümüşhane Üniversitesi Fen Bilimleri Dergisi* 11(3): 956–963.

- HAGBERG, A., and CONWAY, D., 2020. Networkx: Network Analysis with Python. URL: <https://networkx.github.io/>. Erişim: May 29, 2023.
- HE, S., PENG, Y., and SUN, K., 2020. SEIR Modeling of the COVID-19 and Its Dynamics. *Nonlinear Dynamics*, 101(3): 1667–1680.
- HONG, J., Y., 2015. Spatio-Temporal Models Incorporating Network Topology Measures for Pedestrian Exposure Estimation, 26:31-36.
- HONG, J., Y., TAMAKLOE, R., LEE, S., and PARK, D., 2019. Exploring the Topological Characteristics of Complex Public Transportation Networks: Focus on Variations in Both Single and Integrated Systems in the Seoul Metropolitan Area. *Sustainability*, 11(19):54-94.
- HOSSEINI, R., and REZVANIAN, A., 2020. AntLP: Ant-Based Label Propagation Algorithm for Community Detection in Social Networks. *CAAI Transactions on Intelligence Technology*, 5(1): 34–41.
- HUANG, X., and LAKSHMANAN, L., V., S., 2017. Attribute-Driven Community Search. *Proceedings of the VLDB Endowment*, 10(9): 949–960.
- HUO, J., and ZHAO, H., 2016. Dynamical Analysis of a Fractional SIR Model with Birth and Death on Heterogeneous Complex Networks. *Physica A: Statistical Mechanics and Its Applications*, 448(Nisan): 41–56.
- HWANG, C., and YOON, K., 1981. Methods for Multiple Attribute Decision Making. In *Multiple Attribute Decision Making: Methods and Applications A State-of-the-Art Survey*, 81:158–191.
- ISELLA, L., STEHLÉ, J., BARRAT, A., CATTUTO, C., PINTON, J., and VAN DEN BROECK, W., 2011. What's in a Crowd? Analysis of Face-to-Face Behavioral Networks. *Journal of Theoretical Biology*, 271(1): 166–180.
- ISHFAQ, U., KHAN, H., U., and IQBAL, S., 2022. Identifying the Influential Nodes in Complex Social Networks Using Centrality-Based Approach. *Journal of King Saud University - Computer and Information Sciences*, 34(10): 9376–9392.
- İŞSEVER, H., İŞSEVER, T., and ÖZTAN, G. 2020. COVID-19 epidemiyolojisi. *Sağlık Bilimlerinde İleri Araştırmalar Dergisi, Özel Sayı*, 3:1-13.
- JAIN, S., and KUMAR, S., 2021. Dynamical Analysis of SEIS Model with Nonlinear Innate Immunity and Saturated Treatment. *The European Physical Journal Plus*, 136(9): 1-17.
- KARADENİZ, E., 2016. Sosyal Ağlarda Örtüşen Toplulukların Tespit Edilmesi. İnönü Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 63s.
- KARLSSON, D., EKBERG, J., SPRECO, A., ERIKSSON, H., and TIMPKA, T., 2013. Visualization of Infectious Disease Outbreaks in Routine Practice. In *MEDINFO*, 192:697–701.
- KASORO, N., KASEREKA, S., MAYOGHA, E., VINH, H., T., and KINGANGA, J., 2019. PercoMCV: A Hybrid Approach of Community Detection in Social Networks. *Procedia Computer Science, Ambient Systems, Networks*, 19(Ocak):45–52.
- KATHERINE, F., 1997. Centrality in Affiliation Networks. *Social Networks*, 19(2): 1157–1191.
- KATZ, L., 1953. A New Status Index Derived from Sociometric Analysis. *Psychometrika*, 18(1): 39–43.
- KATZ, N., LAZER, D., ARROW, H., and CONTRACTOR, N., 2004. Network Theory and Small Groups. *Small Group Research*, 35(3): 307–332.
- KEELING, M., J., and EAMES, K., T., 2005. Networks and Epidemic Models. *Journal of the Royal Society Interface*, 2(4): 295–307.
- KEMPINSKA, K., LONGLEY, P., and SHAW-TAYLOR, J., 2018. Interactional Regions in Cities: Making Sense of Flows across Networked Systems. *International Journal of Geographical Information Science*, 32(7): 1348–1367.
- KERMACK, W., O., and MCKENDRICK, A., G., 1927. A Contribution to the Mathematical Theory of Epidemics. *Series A, Containing Papers of a Mathematical and Physical Character* 115(772): 700–721.

- KOCH, T., and DENIKE, K., 2009. Crediting His Critics' Concerns: Remaking John Snow's Map of Broad Street Cholera, 1854. *Social Science & Medicine*, 69(8): 1246–1251.
- LI, J., ZHANG, H., HAN, Z., RONG, Y., CHENG, H., and HUANG, J., 2020. Adversarial Attack on Community Detection by Hiding Individuals. In *Proceedings of The Web Conference*, 20-24 Nisan, Tayvan, Cilt 1, s917–927.
- LI, M., ZHOU, S., WANG, D., and CHEN, G., 2023. Identifying Influential Nodes Based on Resistance Distance. *Journal of Computational Science* 67(Mart): 901-972.
- LIN, J., and BAN, Y., 2013. Complex Network Topology of Transportation Systems. *Transport Reviews* 33(6): 658–685.
- LIU, C., ZHOU, J., GAN, J., WU, Y., HUANG, Y., SHAO, J., and OUYANG, L., 2022. Optimizing the Ground Intra-City Express Delivery Network: An Integrated Multiple Centrality Assessment, Multi-Criteria Decision-Making, and Multi-Objective Integer Programming Model. *Journal of Intelligent Transportation Systems*, 11:1–19.
- LIU, X., YE, S., FIUMARA, G., and DE MEO, P., 2023. Influential Spreaders Identification in Complex Networks with TOPSIS and K-Shell Decomposition. *IEEE Transactions on Computational Social Systems*, 10(1): 347–361.
- LIU, X., and TAO, C., 2022. A Dead Code Evaluation Method Based on Complex Network. *Software Quality, Reliability, and Security Companion (QRS-C)*, 22:431–436.
- LIU, Z., YANG, L., and LI, Y., 2023. Influence Evaluation of Academic Groups Based on Multi-Attribute Decision Method. *Computer Application and Information Security*, 12(1):85–91.
- LU, M., ZHANG, Z., QU, Z., and KANG, Y., 2019. LPANNI: Overlapping Community Detection Using Label Propagation in Large-Scale Complex Networks. *IEEE Transactions on Knowledge and Data Engineering* 31(9): 1736–1749.
- LU, P., CHEN, Y., and LIAO, Y., 2023. Novel Scheme for Essential Proteins Identification Based on Improved Multicriteria Decision Making. *Journal of Shanghai Jiaotong University (Science)*, 12(Mart):88–95.
- MA, T., LIU, Q., CAO, J., TIAN, Y., AL-DHELAAN, A., and AL-RODHAAN, M., 2020. LGIEM: Global and Local Node Influence Based Community Detection. *Future Generation Computer Systems* 105(Nisan):533–546.
- MA, T., ZHAO, Y., ZHOU, H., TIAN, Y., AL-DHELAAN, A., and AL-RODHAAN, M., 2019. Natural Disaster Topic Extraction in Sina Microblogging Based on Graph Analysis. *Expert Systems with Applications*, 115(Ocak): 346–355.
- MAHADEVI, S., KAMATH, S., S., and SHETTY, D., P., 2022. Graph Energy Centrality: A New Centrality Measurement Based on Graph Energy to Analyse Social Networks. *International Journal of Web Engineering and Technology*, 17(2):144–169.
- MCKINNEY, W. 2011. pandas: a foundational Python library for data analysis and statistics. *Python for high performance and scientific computing*, 14:1-9.
- MEGHANATHAN, N., 2016. A Greedy Algorithm for Neighborhood Overlap-Based Community Detection. *Algorithms* 9(1): 8-17.
- MENON, A., RAJENDRAN, N., K., CHANDRACHUD, A., and SETLUR, G., 2020. Modelling and Simulation of COVID-19 Propagation in a Large Population with Specific Reference to India. *MedRxiv*, 20(2):20-28.
- MEUNIER, D., LAMBIOTTE, R., FORNITO, A., ERSCHKE, K., and BULLMORE, E., 2009. Hierarchical Modularity in Human Brain Functional Networks. *Frontiers in Neuroinformatics*, 18:3-19.
- MIKOLOV, T., CHEN, K., CORRADO, C., and DEAN, J., 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv*, 13(1):37-81.
- MILGRAM, S., 1967. The Small World Problem. *Psychology Today*, 2(1): 60–67.
- MOHAMMADI, M., FAZLALI, M., and HOSSEINZADEH, M., 2021. Accelerating Louvain Community Detection Algorithm on Graphic Processing Unit. *The Journal of Supercomputing*, 77(6): 6056–6077.

- NADINI, M., RIZZO, A., and PORFIRI, M., 2020. Epidemic Spreading in Temporal and Adaptive Networks with Static Backbone. *IEEE Transactions on Network Science and Engineering*, 7(1):549–561.
- NGUYEN, P., N., and WOO, S., 2022. Port Connectivity and Competition among Container Ports in Southeast Asia Based on Social Network Analysis and TOPSIS. *Maritime Policy & Management*, 49(6):779–796.
- OKUDA, M., SATOH, S., SATO Y., and KIDAWARA, Y., 2021. Community Detection Using Restrained Random-Walk Similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):89–103.
- OLINKY, R., and STONE, L., 2004. Unexpected Epidemic Thresholds in Heterogeneous Networks: The Role of Disease Transmission. *Physical Review E*, 70(3):367–382.
- ONOFRIO, A., 2008. A Note on the Global Behaviour of the Network-Based SIS Epidemic Model. *Nonlinear Analysis: Real World Applications*, 9(4):67–72.
- OPRICOVIC, S., and TZENG, G., 2004. Compromise Solution by MCDM Methods: A Comparative Analysis of VIKOR and TOPSIS. *European Journal of Operational Research*, 156(2): 445–455.
- ORMAN, G., K., and LABATUT, V., 2009. A Comparison of Community Detection Algorithms on Artificial Networks. In *Discovery Science*, 20:242–256.
- ORMAN, G., K., LABATUT, V., and CHERIFI, H., 2011. Qualitative Comparison of Community Detection Algorithms. In *Digital Information and Communication Technology and Its Applications*, 12:265–279.
- PARAND, F., RAHIMI, H., and GORZIN, M., 2016. Combining Fuzzy Logic and Eigenvector Centrality Measure in Social Network Analysis. *Physica A: Statistical Mechanics and Its Applications*, 459(Eylül): 24–31.
- PEROZZI, B., AL-RFOU, R., and SKIENA, S., 2014. DeepWalk: Online Learning of Social Representations. *Knowledge Discovery and Data Mining*, 24-27 Ağustos, Newyork, Cilt 1,s701–710.
- PONS, P., and LATAPY, M., 2005. Computing Communities in Large Networks Using Random Walks. In *Computer and Information Sciences*, 18:284–293.
- PUJOL, J., M., ERRAMILI, V., and RODRIGUEZ, P., 2009. Divide and Conquer: Partitioning Online Social Networks. *arXiv*, 379:189-203.
- RAEDER, T., and CHAWLA, N., V., 2011. Market Basket Analysis with Networks. *Social Network Analysis and Mining*, 1(2): 97–113.
- RAHMAN, B., SADRADDIN, E., and PORRECA, A., 2020. The Basic Reproduction Number of SARS-CoV-2 in Wuhan Is about to Die out, how about the Rest of the World. *Reviews in Medical Virology*, 30(4): 211-221.
- RAI, K., A., MACHKOUR, M., and ANTARI, J., 2023. Unsupervised Learning-Based New Seed-Expanding Approach Using Influential Nodes for Community Detection in Social Networks. *International Journal of Advanced Computer Science and Applications*, 14(1):263-277.
- RANI, S., and KUMAR, M., 2022a. Influential Node Detection and Ranking with Fusion of Heterogeneous Social Media Information. *IEEE Transactions on Computational Social Systems*, 18:1–23.
- RANI, S., and KUMAR, M., 2022b. Ranking Community Detection Algorithms for Complex Social Networks Using Multilayer Network Design Approach. *International Journal of Web Information Systems*, 18(5/6): 310–341.
- REICHARDT, J., and BORNHOLDT, S., 2006. Statistical Mechanics of Community Detection. *Physical Review E*, 74(1): 110-116.
- ROCCO, C., M., BARKER, K., and MORONT, J., 2022. Determining the Best Algorithm to Detect Community Structures in Networks: Application to Power Systems. *Environment Systems and Decisions*, 42(2): 251–564.
- ROCHA, L., E., C., and BLONDEL, V., D., 2013. Bursts of Vertex Activation and Epidemics in Evolving Networks. *PLOS Computational Biology* 9(3): 954-974.

- RONG, H., MA, T., CAO, J., TIAN, Y., AL-DHELAAN, A., and AL-RODHAAN, M., 2019. Deep Rolling: A Novel Emotion Prediction Model for a Multi-Participant Communication Context. *Information Sciences*, 488(Temmuz): 158–180.
- SABAH, L., and ŞİMŞEK, M., 2023. A New Fast Entropy-Based Method to Generate Composite Centrality Measures in Complex Networks. *Concurrency and Computation: Practice and Experience*, 35(10): 57-76.
- SABIDUSSI, G., 1966. The Centrality Index of a Graph. *Psychometrika*, 31(4): 581–603.
- SAEEDIAN, M., KHALIGHI, M., AZIMI-TAFRESHI, N., JAFARI, G., R., and AUSLOOS, M., 2017. Memory Effects on Epidemic Evolution: The Susceptible-Infected-Recovered Epidemic Model. *Physical Review E* 95(2): 22-40.
- SALHA-GALVAN, G., LUTZEYER, J., F., DASOULAS, G., HENNEQUIN, R., and VAZIRGIANNIS, M., 2022. Modularity-Aware Graph Autoencoders for Joint Community Detection and Link Prediction. *Neural Networks*, 153(Ekim): 474–495.
- SHAHRIVARI JOGHAN, H., and AZAD, M., 2019. Weighted Label Propagation Based on Local Edge Betweenness. *The Journal of Supercomputing*, 75(Aralık):15-25.
- SHAW, M., E., 1954. Group Structure and the Behavior of Individuals in Small Groups. *The Journal of Psychology*, 38(1):139–149.
- SINGH, P., and GUPTA, A., 2022. Generalized SIR (GSIR) Epidemic Model: An Improved Framework for the Predictive Monitoring of COVID-19 Pandemic. *ISA Transactions*, 124(Mayıs):31–40.
- TANG, W., ALLMAN, A., POURKARGAR, D., B., and DAOUIDIS, P., 2018. Optimal Decomposition for Distributed Optimization in Nonlinear Model Predictive Control through Community Detection. *Computers & Chemical Engineering*, 111(Mart): 43–54.
- TIOBE., 2023. <https://www.tiobe.com/tiobe-index/>. Transmission Dynamics of the Etiological Agent of SARS in Hong Kong: Impact of Public Health Interventions | Science. Accessed May 24, 2023.
- ULLAH, A., SHAO, J., YANG, Q., KHAN, N., BERNARD, C., M., and KUMAR, R., 2023. LSS: A Locality-Based Structure System to Evaluate the Spreader's Importance in Social Complex Networks. *Expert Systems with Applications*, 228(Ekim):120-126.
- WANG, M., WANG, C., YU, J., X., and ZHANG, J., 2015. Community Detection in Social Networks: An In-Depth Benchmarking Study with a Procedure-Oriented Framework. *Proceedings of the VLDB Endowment*, 8(10):998–1009.
- WANG, S., DU, Y., and DENG, Y., 2017. A New Measure of Identifying Influential Nodes: Efficiency Centrality. *Communications in Nonlinear Science and Numerical Simulation*, 47(Haziran):151–163.
- WANG, S., YANG, J., DING, X., ZHANG J., and ZHAO, M., 2023. A Local Community Detection Algorithm Based on Potential Community Exploration. *Frontiers in Physics* 11:278-296.
- WATABE, M., 1998. <https://www.sscnet.ucla.edu/soc/faculty/mcfarland/soc112/cent-ans.htm>. Accessed 29 May,2023.
- WATTS, D., J., and STROGATZ, S., H., 1998. Collective Dynamics of ‘Small-World’ Networks. *Nature*, 393(6684): 440–442.
- WEN, X., CHEN, W., LIN, Y., GU, T., ZHANG, H., LI, Y., YIN, Y., and ZHANG, J., 2017. A Maximal Clique Based Multiobjective Evolutionary Algorithm for Overlapping Community Detection. *IEEE Transactions on Evolutionary Computation* 21(3): 363–377.
- XING, F., MA, T., TANG, M., and GUAN, D., 2019. Friend Circle Identification in Ego Network Based on Hybrid Method. *International Journal of Ad Hoc and Ubiquitous Computing* 30(4): 224–234.
- XU, J., and LUO, J., 2023. “Improved TOPSIS Key Intersection Recognition Based on Fusion Improved CRITIC Weight Method and Independent Weight Method.” In *Sixth International Conference on Traffic Engineering and Transportation*, 23-25 Ekim, Cilt 1, Çin, s807–816.

- ZHANG, J., and LUO, Y., 2017. Degree Centrality, Betweenness Centrality, and Closeness Centrality in Social Network. Atlantis Press, 189:301-313.
- ZHAO, Y., LI, S., and JIN, F., 2016. Identification of Influential Nodes in Social Networks with Community Structure Based on Label Propagation. Neurocomputing, SI:Behavior Analysis In SN, 210(Ekim): 34–44.

