

ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DOKTORA TEZİ

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE YAPISAL VERİLERİN VEKTÖR
GÖMÜLÜMLERİ

Sarmad Nihad Mohammed SALIH

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2024

Her hakkı saklıdır

ÖZET

Doktora Tezi

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE YAPISAL VERİLERİN VEKTÖR GÖMÜLÜMLERİ

Sarmad Nihad Mohammed SALIH

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Semra GÜNDÜÇ

Günümüzde artan veri miktarı ve karmaşıklığı, verilerdeki ilişkileri ve gizli kalıpları ortaya çıkarabilecek güçlü yöntemlere olan ihtiyacı artırmaktadır. Ağlar, bu karmaşık ilişkilerin yakalanmasında ve analizinde etkili bir araç sunmaktadır. Ancak ağların düzensiz yapısı, geleneksel makine öğrenimi algoritmalarının gerektirdiği sabit boyutlu vektör formatına uygun değildir. Bu nedenle, ağların yapısal bilgilerini vektör temsillerine dönüştüren yöntemlerin geliştirilmesine ihtiyaç duyulmaktadır. Bu tez çalışmada, ağ verilerinin işlenmesi ve düğüm yerleştirme yöntemleri incelenerek, ağların yapısal özelliklerini koruyan düşük boyutlu vektör temsillerinin nasıl elde edileceği araştırılmıştır. Düğüm sınıflandırma, bağlantı tahmini ve topluluk tespiti gibi çeşitli ağ madenciliği görevlerinde makine öğrenimi ile ağ temsili öğreniminin entegrasyonunun performansı artırdığı gösterilmiştir. Bu kapsamda, ağların makine öğrenimi ile etkin bir şekilde analiz edilmesi için üç yeni yöntem önerilmiştir. İlk olarak, derece tabanlı rasgele yürüyüş yöntemi geliştirilmiştir. Bu yöntem, düğüm derecelerine orantılı rasgele yürüyüşler yaparak, daha az sayıda yürüyüşle yüksek hassasiyette vektör temsilleri elde edilmesini sağlamıştır. İkinci olarak, Geçiş Olasılık Matrisi (TPM) yöntemi önerilmiş ve düğümlerin yerel yapısal özellikleri, anonim rasgele yürüyüşler aracılığıyla analiz edilmiştir. TPM, düğüm kimliklerinden ziyade bağlantı yapılarını vurgulayarak ağın derinliklerine inmeden daha verimli bilgi toplamayı sağlamıştır. Son olarak, metin verilerinin ağ yapısı ile incelenmesi amacıyla Kılavuzlu Geçiş Olasılık Matrisi (GTPM) yöntemi önerilmiştir. Bu yöntem, metinlerin gizli içeriklerini ve sözdizimsel unsurlarını dikkate alarak, kelimeler arasındaki ilişkileri daha iyi temsil eden yerleştirme vektörleri oluşturmuştur. Önerilen yöntemler, ağ analizi ve metin sınıflandırma gibi çeşitli deneysel çalışmalara dayanarak değerlendirilmiş ve hesaplama verimliliği ile doğruluk açısından mevcut yöntemlere üstünlük sağlamıştır.

Ekim 2024, 109 sayfa

Anahtar Kelimeler: Ağ veri madenciliği, öznitelik öğrenimi, düğüm yerleştirme, rasgele yürüyüş, ağ temsili öğrenimi

ABSTRACT

PhD Thesis

VECTOR EMBEDDINGS OF STRUCTURAL DATA BY MACHINE LEARNING TECHNIQUES

Sarmad Nihad Mohammed SALIH

Ankara University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Prof. Dr. Semra GÜNDÜÇ

In today's world, the increasing volume and complexity of data have amplified the need for powerful methods capable of uncovering relationships and hidden patterns within the data. Networks provide an effective tool for analyzing such complex relationships, but their irregular structure is not compatible with the fixed-dimensional vector format required by traditional machine learning algorithms. Therefore, methods are needed to convert the structural information of networks into vector representations. This thesis explores how to obtain low-dimensional vector representations that preserve network structures through the examination of node embedding techniques. It demonstrates that integrating machine learning with network representation learning improves performance in tasks like node classification, link prediction, and community detection. In this context, three novel methods have been proposed for the effective analysis of networks using machine learning. First, a degree-based random walk method is developed, producing high-precision vector representations with fewer walks. Second, the Transition Probability Matrix (TPM) method analyzes local node structures using anonymous random walks, efficiently collecting information without deep exploration of the network. Lastly, the Guided Transition Probability Matrix (GTPM) method has been proposed for the analysis of text data within a network structure. This method generates embedding vectors that more accurately represent the relationships between words by taking into account the latent content and syntactic elements of the texts. The proposed methods have been evaluated through various experimental studies, including network analysis and text classification, demonstrating superiority over existing methods in terms of computational efficiency and accuracy.

October 2024, 109 pages

Keywords: Network data mining, feature learning, node embedding, random walk, network representation learning

ÖNSÖZ VE TEŞEKKÜR

Çalışmalarımı yönlendiren, araştırmalarımın her aşamasında bilgi, öneri ve yardımlarını esirgemeyen danışman hocam Prof. Dr. Semra GÜNDÜÇ (Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı),

Tez çalışmalarım süresince yapılan tez izleme komitesi toplantılarına katılarak fikir ve önerileriyle çalışmamın gelişimine katkıda bulunan Prof. Dr. Şahin EMRAH (Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı) ve Prof. Dr. Nizami GASİLOV'a (Başkent Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı),

Bugünlere gelmemde en büyük emekleri olan anneme ve babama; sağladığı koşulsuz ve sınırsız manevi destek, gösterdiği hoşgörü ve fedakârlıklar için sevgili eşim Sara PERİ ve enerjileriyle bana her daim güç veren oğullarım Eren SALİH ve Oğuz SALİH'a en içten teşekkürlerimi sunarım.

Sarmad Nihad Mohammed SALİH
Ankara, Ekim 2024

İÇİNDEKİLER

TEZ ONAY SAYFASI

ETİK.....	i
ÖZET.....	ii
ABSTRACT	iii
ÖNSÖZ VE TEŞEKKÜR	iv
SİMGELER VE KISALTMALAR DİZİNİ	vii
ŞEKİLLER DİZİNİ	ix
ÇİZELGELER DİZİNİ	xi
1. GİRİŞ.....	1
1.1 Motivasyon ve Tez Kapsamı	6
1.2 Tezin Özgün Değeri	8
1.3 Tezin Yapısı	10
2. LİTERATÜR ÖZETİ.....	11
2.1 Matris Çarpanlara Ayırma Modelleri	13
2.2 Rasgele Yürüyüşe Dayalı Modeller	15
2.3 Derin Sinir Ağı Modelleri.....	22
3. AĞLAR.....	29
3.1 Ağlara Genel Bakış	29
3.2 Ağların Matris Temsili	30
3.3 Ağ Yapısının Çıkarımı.....	31
3.3.1 Derece ve derece dağılımı.....	31
3.3.2 Yürüyüşler ve rasgele yürüyüşler	32
3.4 Ölçekten Bağımsız Ağlar	33
3.5 Ağların Analizi	34
3.6 Düğüm Yerleştirme Yöntemlerinin Uygulamaları	35
3.6.1 Düğüm sınıflandırma	35
3.6.2 Bağlantı tahmini	36
3.6.3 Düğüm kümelemesi	37
3.6.4 Görselleştirme	37
3.7 Metin Temsili Öğrenimi	38
3.8 Ağ Temsili Öğrenimi.....	40
3.9 Node2vec	43
4. AĞ TEMSİLİ ÖĞRENİMİ YÖNTEMLERİNİN GELİŞTİRİLMESİ	47
4.1 Derece Tabanlı Rasgele Yürüyüş Modeli.....	47
4.2 Geçiş Olasılık Matrisi Modeli	52
4.2.1 Anonim rasgele yürüyüşler	54
4.2.2 Geçiş olasılıkları aracılığıyla yerel topoloji çıkarımı.....	55
4.2.3 Geçiş matrisi hesaplanmasına yönelik kavramsal yaklaşım.....	56
4.2.4 Geçiş olasılık matrisi hesaplanması – simülasyon	60
4.2.5 TPM modelinin mimarisi	62
4.3 Kılavuzlu Geçiş Olasılık Matrisi Modeli.....	64
4.3.1 Evrensel ağırlıklı kelime ağı	66
4.3.2 Evrensel kelime ağı aracılığıyla kelime yerleştirmesi.....	68
5. BULGULAR ve TARTIŞMA	71
5.1 Değerlendirme Metrikleri	71

5.2 Veri Setler	72
5.3 Derece Tabanlı Rasgele Yürüyüş Modelinin Sonuçları	75
5.3.1 Boyutsal indirgeme ile topluluk yapılarının incelenmesi	75
5.3.2 CORA ve CiteSeer veri setleri ile elde edilen sonuçlar	76
5.4 Geçiş Olasılık Matrisi Modelinin Sonuçları	80
5.4.1 Düğüm sınıflandırma sonuçları	81
5.4.2 Bağlantı tahmini sonuçları	82
5.4.3 Görselleştirme sonuçları	83
5.4.4 Ağlar arası genelleme sonuçları	84
5.4.5 Parametre hassasiyeti sonuçları	86
5.5 Kılavuzlu Geçiş Olasılık Matrisi Modelinin Sonuçları	87
5.5.1 Veri ön işleme	87
5.5.2 Metin sınıflandırma süreci	88
5.5.3 Metin sınıflandırma sonuçları	88
5.5.4 Sınırlı eğitim verileriyle yerleştirme algoritmalarının performans analizi	90
5.5.5 Görselleştirme sonuçları	92
5.5.6 Parametre hassasiyeti sonuçları	92
6. SONUÇ	94
6.1 Gelecek Çalışmalar	96
KAYNAKLAR	97
ÖZGEÇMİŞ	109

SİMGELER VE KISALTMALAR DİZİNİ

A	Komşuluk matrisi
At	Öznitelikler kümesi
AW	Anonim rasgele yürüyüş
$\alpha_{pq}(t, x)$	p ve q tarafından tanımlanan yanlı arama
$\delta_{i,j}$	v_i, v_j düğümleri arasındaki bağlantıyı belirten Kronecker deltası
$\mathbb{d}$	Düğüm yerleştirme vektör boyutu
d_i	i . düğümün derecesi
d_{tx}	t ve x düğümleri arasındaki en kısa yol
E	Bağlantılar kümesi
e_{ij}	i ve j düğümlerini birbirine bağlayan kenar
$ E $	Toplam bağlantı sayısı
f	Haritalama fonksiyonu
G	Ağ / Çizge
$g_i(u, v)$	Bağlantı temsili $g(u, v)$ 'nin i -inci bileşeni
L	kayıp fonksiyonu
M	Kelime dizisindeki kelime sayısı
μ	LFR ağı için gürültü parametresi
N_i	i . düğümün komşu kümesi
nw	Rasgele yürüyüş sayısı
nw_i	i . düğümden başlatılan yürüyüş sayısı
p	Geri dönme ağırlığı parametresi
$P_{i,j}^t$	t zaman adımında i düğümünden j düğümüne geçiş olasılığı
π_{vx}	Düğüm v 'den x 'e geçiş için normalleştirilmemiş olasılık
q	İçeri-dışarı ağırlığı parametresi
RW	Rasgele yürüyüş
S	Cümle benzeri düğüm dizileri
v_i	i . düğüm
v_w	Hedef düğümün vektör temsili
$\tilde{v}_w$	Bağlam düğümlerin vektör temsili
V	Düğüm kümesi
$ V $	Toplam düğüm sayısı
W	Kelime daracığı (sözlük)
$\mathcal{W}$	Kenar ağırlıkları kümesi
ws	Pencere boyutu (window size)
wpd	Derece başına düşen yürüyüş sayısı
wl	Rasgele yürüyüş uzunluğu
ω_{ij}	i ve j düğümlerini birbirine bağlayan kenarın ağırlığı
$x(t)$	t anındaki yürüyüşçünün konumu

Kısaltmalar

BFS	Genişlik Öncelikli Arama (Breadth First Search)
CNN	Evrişimli Sinir Ağları (Convolutional Neural Networks)
DDRW	Ayırt Edici Derin Rasgele Yürüyüş (Discriminative Deep Random Walk)
DFS	Derinlik Öncelikli Arama (Depth First Search)
GATs	Çizge Dikkat Ağları (Graph Attention Networks)
GCN	Çizge Evrişimli Ağ (Graph Convolutional Network)
GNN	Çizge Sinir Ağları (Graph Neural Networks)
GTPM	Kılavuzlu Geçiş Olasılık Matrisi (Guided Transition Probability Matrix)
IDRW	Ters Derece Rasgele Yürüyüş (Inverse Degree Random Walk)
LDA	Doğrusal Ayırma Analizi (Linear Discriminant Analysis)
LFR	Lancichinetti-Fortunato-Radicchi
LLE	Yerel Doğrusal Yerleştirme (Locally Linear Embedding)
LSTM	Uzun Kısa Süreli Bellek (Long Short-Term Memory)
MDS	Çok Boyutlu Ölçekleme (Multidimensional Scaling)
MLP	Çok Katmanlı Algılayıcı (Multi-Layer Perceptron)
NLP	Doğal Dil İşleme (Natural Language Programming)
PCA	Temel Bileşenler Analizi (Principal Component Analysis)
R-GCNs	İlişkisel Çizge Evrişimli Ağlar (Relational Graph Convolutional Networks)
RNN	Tekrarlayan Sinir Ağı (Recurrent Neural Network)
SVM	Destek Vektör Makinesi (Support Vector Machine)
TPM	Geçiş Olasılık Matrisi (Transition Probability Matrix)
t-SNE	t-Dağıtımli Olasılıksal Komşu Yerleştirme (t-Distributed Stochastic Neighbor Embedding)

ŞEKİLLER DİZİNİ

Şekil 1.1 Beş makale arasındaki atıf ilişkilerini gösteren bir ağ, yönlü kenarlar (bağlantılar) atıfları temsil eder	2
Şekil 1.2 Düğüm temsili öğrenimi	4
Şekil 2.1 2010'dan Ağustos 2024'e kadar ağ yerleştirme üzerine odaklanan makalelerin istatistikleri.....	11
Şekil 2.2 (a) Bir görüntü ızgara grafiğine dönüştürülebilir (b) Metin veya konuşma sinyali çizgi grafiğine dönüştürülebilir (c) Farklı düğümlerin farklı derecelere sahip olduğu düzensiz bir ağ örneği	23
Şekil 3.1 Farklı ağ türlerinin temsilleri	29
Şekil 3.2 Beş düğüm ve dokuz kenardan oluşan yönsüz örnek bir ağın komşuluk matrisi	31
Şekil 3.3 Zachary Karate Club ağındaki düğüm derecelerinin dağılımı.....	32
Şekil 3.4 Düğüm sınıflandırmasının şematik gösterimi.....	36
Şekil 3.5 Eksik bağlantılar için bağlantı tahmini problemi.....	36
Şekil 3.6 Beş topluluğun yer aldığı örnek bir ağ.....	37
Şekil 3.7 Zachary Karate Club ağına ait 2 boyutlu temsillerin görselleştirilmesi (Perozzi vd. 2014)	38
Şekil 3.8 Word2vec modelinin CBOW ve Skip-gram mimarileri (Mikolov vd. 2013).....	39
Şekil 3.9 Ağ yerleştirme işleminin gösterimi (Baptista vd. 2023).....	42
Şekil 3.10 Node2vec algoritmasının yanlış rasgele yürüyüş prosedürü (Grover ve Leskovec 2016).....	44
Şekil 3.11 Düğüm u 'dan başlayan genişlik öncelikli arama ve derinlik öncelikli arama stratejileri ($wl = 3$) (Grover ve Leskovec 2016).....	45
Şekil 4.1 Örnek bir ağda v_3 düğümünden başlatılan 2 rasgele yürüyüşün temsili.....	49
Şekil 4.2 Dereceye dayalı rasgele yürüyüşler ile sabit sayıda rasgele yürüyüşler arasındaki sayısal farklılıkların şematik temsili	50
Şekil 4.3 Dereceye dayalı rasgele yürüyüş algoritması	50
Şekil 4.4 Homofili ve yapısal rol benzerliğinin açıklayıcı bir örneği	52

Şekil 4.5 Anonim yürüyüş kavramını gösteren bir örnek	55
Şekil 4.6 Örnek bir ağ	57
Şekil 4.7 Örnek bir ağ, vi düğümünün derece sayısı di olarak ifade edilmektedir	58
Şekil 4.8 Üç adımlı anonim yürüyüşün olası dizilimleri.....	58
Şekil 4.9 Üç adımlık bir komşulukta düğümleri ziyaret etme olasılıkları	59
Şekil 4.10 Her zaman adımı için geçiş olasılığı matris elemanı	59
Şekil 4.11 A düğümü için rasgele ve anonim yürüyüşler kullanılarak geçiş olasılığı matrisi oluşturma süreci.....	61
Şekil 4.12 Önerilen TPM modelinin ardışık düzeni.....	63
Şekil 4.13 Geçiş olasılık matrisi oluşturma yönteminin algoritması	64
Şekil 4.14 Dört döküman için kelime ağının oluşturulması.....	67
Şekil 4.15 Önerilen GTPM modelinin ardışık düzeni.....	70
Şekil 5.1 Zachary Karate Club ağı kullanılarak iki boyutlu düğüm temsillerinin görselleştirilmesi.....	76
Şekil 5.2 CORA veri setinde, rasgele yürüyüş sayısı ve yürüyüş uzunluğunun düğüm sınıflandırması ve bağlantı tahmini üzerindeki doğruluğa etkileri	78
Şekil 5.3 Üç topluluk, 600 düğüm ve 1.334 kenardan oluşan yapay LFR ağı.....	83
Şekil 5.4 LFR ağı için 2 boyutlu temsillerin görselleştirilmesi	84
Şekil 5.5 Parametre hassasiyeti deneyinin sonuçları. (a) Üç alıntı ağı üzerindeki yürüyüş sayısının etkisi. (b) Üç alıntı ağı üzerindeki yürüyüş uzunluğunun etkisi.....	86
Şekil 5.6 Farklı eğitim boyutlarıyla Ohsumed ve 20NG veri setlerinde test performansı (Micro F1).....	91
Şekil 5.7 Reuters veri seti için 2 boyutlu temsillerin görselleştirilmesi.....	92
Şekil 5.8 Parametre hassasiyeti deneyinin sonuçları. (a) Dört veri seti üzerindeki yürüyüş sayısının etkisi. (b) Dört veri seti üzerindeki yürüyüş uzunluğunun etkisi.....	93

ÇİZELGELER DİZİNİ

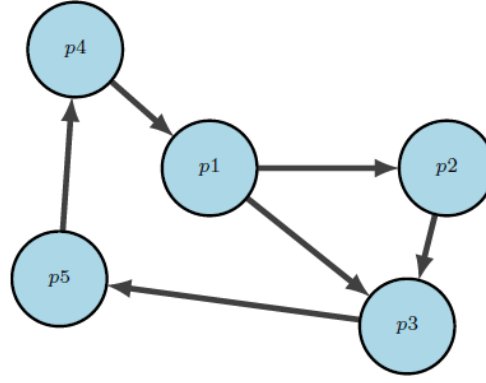
Çizelge 4.1 Üç adımlı anonim yürüyüşler kümesi için anonim rasgele yürüyüş olasılıkları	60
Çizelge 4.2 Farklı sınıflara ait örnek dökümanlar.....	66
Çizelge 5.1 Ağ tabanlı veri setlerinin istatistikleri.....	72
Çizelge 5.2 Metin tabanlı veri setlerinin istatistikleri	72
Çizelge 5.3 Derece tabanlı node2vec ve orijinal node2vec yöntemlerinin Cora veri seti üzerindeki düğüm sınıflandırma performanslarının özeti.....	77
Çizelge 5.4 Derece tabanlı node2vec ve orijinal node2vec yöntemlerinin CiteSeer veri seti üzerindeki düğüm sınıflandırma performanslarının özeti	77
Çizelge 5.5 CiteSeer veri setinde derece tabanlı node2vec ve orijinal node2vec için düğüm sınıflandırması açısından doğruluk değerleri (%)......	79
Çizelge 5.6 CiteSeer veri setinde derece tabanlı node2vec ve orijinal node2vec için bağlantı tahmini açısından doğruluk değerleri (%)......	80
Çizelge 5.7 Düğüm sınıflandırma sonuçlarının özeti. Kalın yazılmış değerler en iyi sonuçları temsil etmektedir	81
Çizelge 5.8 Bağlantı tahmini sonuçlarının özeti. Kalın yazılmış değerler en iyi sonuçları temsil etmektedir.....	83
Çizelge 5.9 İki ölçekten bağımsız ağ için ağlar arası genelleme açısından sonuçların özeti. Kalın yazılmış değerler en iyi sonuçları temsil etmektedir	85
Çizelge 5.10 Önerilen GTPM yönteminin MR, SST-2 ve CoLA veri setlerinde ikili metin sınıflandırma modelleri arasındaki karşılaştırmalı performansı	89
Çizelge 5.11 Önerilen GTPM yönteminin Reuters, 20NG ve Ohsumed veri setlerinde çok sınıflı metin sınıflandırma modelleri arasındaki karşılaştırmalı performansı...	90
Çizelge 5.12 Tümevarımsal ayarda çok sınıflı metin sınıflandırma veri setleri için Micro F1 ve Macro F1 skorları. Çizelge 5.11'deki performans düşüşleri parantez içinde belirtilmiştir.....	91

1. GİRİŞ

Son yıllarda, günümüz toplumun her yönü tarafından üretilen veri miktarı ve çeşitliliği hızla artmaktadır (Labrinidis ve Jagadish 2012). Bu veri artışı, sosyal medya ve çevrimiçi ticaret gibi günlük alanların yanı sıra sağlık hizmetleri (Raghupathi ve Raghupathi 2014) ve astronomi (Zhang ve Zhao 2015) gibi çeşitli bilimsel disiplinleri de kapsamaktadır. Büyük ölçekli veri kümelerinden elde edilen örüntü ve eğilimlerin belirlenmesi, son dönemde gerçekleştirilen çeşitli önemli bilimsel keşiflerde kritik bir öneme sahip olmuştur (Reed ve Dongarra 2015). Bu bağlamda, büyük ve karmaşık veri kümelerinin en iyi şekilde nasıl depolanacağı, işleneceği ve içerindeki desenlerin nasıl belirleneceği konularında büyük bir araştırma ilgisi oluşmuştur.

Belirli bir veri kümesinden yeni bilgiler çıkarma girişiminde bulunurken göz önünde bulundurulması gereken iki temel hedef mevcuttur: verilerin içerebileceği önemli ilişkileri daha belirgin kılmak için en iyi şekilde nasıl temsil edileceği ve bu verilerden örüntülerin en etkili biçimde nasıl öğrenileceğidir. Bu hedefler her zaman birbiriyle uyumlu olmayabilir ve bazen biri diğerinden daha önemli olabilir. Bununla birlikte, her iki amacı da gerçekleştirme potansiyeline sahip bir veri temsili, bu tezde sunulan çalışmanın odak noktasını oluşturacak olan ağ (çizge) yapısıdır.

Verilerdeki karmaşık ilişkileri veya karşılıklı bağımlılıkları bir çizge veya ağ biçiminde temsil etme düşüncesi uzun yıllardır mevcuttur (Euler 1741). Çizgeler / Ağlar, birçok alanda kilit rol oynayan yaygın veri yapılarıdır. Bunların geniş çaplı kullanımı, sosyal ağ analizinden moleküler ağ yapılarına, biyolojik protein-protein etkileşimlerinden öneri sistemlerinin karmaşıklıklarına kadar uzanmaktadır (Hamilton vd. 2017b). En basit haliyle, bir ağ yalnızca iki temel bileşenden oluşur: düğümler ve kenarlar (bağlantılar). Verileri bir ağ olarak temsil ederken, düğümler genellikle varlıkları (bir kişi veya bilimsel makale) temsil ederken, kenarlar bunlar arasındaki ilişkileri (arkadaşlık veya atıf) yansıtır. Örneğin, farklı araştırma makaleleri arasındaki atıfları bir ağ olarak temsil etmek yaygın bir uygulamadır; burada düğümler makaleleri temsil ederken, iki makale arasındaki bir kenarın varlığı, bir atfın mevcudiyetini gösterir.



Şekil 1.1 Beş makale arasındaki atıf ilişkilerini gösteren bir ağ, yönlü kenarlar (bağlantılar) atıfları temsil eder

Şekil 1.1'de, beş makalenin (p1, p2, p3, p4, p5) ve bunların karşılıklı atıflarını gösteren örnek bir ağ sunulmuştur. Yönlü kenarlar, hangi makalenin diğerine atıfta bulunduğunu belirtmektedir. Bu basit örnekle bile, ağların diğer veri biçimleriyle kıyaslandığında, verilerdeki doğal ilişkilerin anlamsal olarak anlaşılabilir ve yorumlanabilir bir şekilde temsil edilmesine olanak tanımaktadır.

Tüm veri kaynaklarında olduğu gibi, depolanan ve işlenen ağların boyutu ve sayısı hızla artmaktadır (Khan ve Elnikety 2014). Bu artış, geleneksel ağ madenciliği yaklaşımlarının bu büyük ölçekli verilerle başa çıkmakta zorlanmasına veya kabul edilemez derecede uzun çalışma süreleri gerektirmesine yol açmıştır (Aridhi ve Nguifo 2016). Bu sorunu aşmak için, son yıllarda dağıtık bir hesaplama ortamında paralel işlem yeteneği sunan yazılım çerçeveleri geliştirilmiştir (Kang vd. 2009, Malewicz vd. 2010, Xin vd. 2013, Zhang vd. 2016). Ayrıca, ağlara özgü işlemciler de geliştirilmiştir (Jia vd. 2019). Bununla birlikte, ağların giderek daha karmaşık görevler için kullanılması, verilen görevi çözmek için en uygun topolojik yapının belirlenmesini zorunlu kılmaktadır. Bu süreç, özellik mühendisliği olarak da adlandırılabilir ve genellikle bir alan uzmanının belirli bir görev için doğru metriği seçmesini veya yeni bir metrik oluşturmasını gerektirir (Lam vd. 2017). Bu süreç hem zaman hem de maliyet açısından önemli bir yük oluşturabilmektedir.

Son yıllarda, bilgisayar programları aracılığıyla veri setlerindeki örüntülerin otomatik olarak tanımlanması ihtiyacı önemli ölçüde artmıştır. Makine Öğrenimi alanından doğan

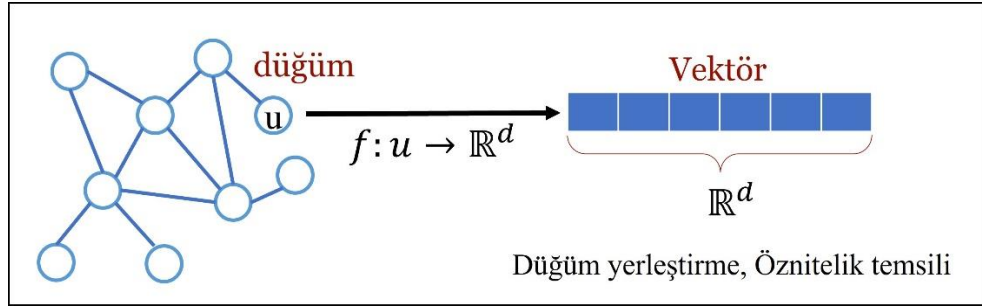
bu teknikler, hem karmaşıklık hem de yetenek açısından önemli ölçüde gelişmiş ve geniş bir uygulama yelpazesi bulmuştur (Goodfellow vd. 2016). Makine öğrenmesi algoritmalarının, otonom araç kullanımı (Grigorescu vd. 2020), otomatik dil çevirisi (Vaswani vd. 2017) ve hatta tıbbi teşhis (De Fauw vd. 2018) gibi birbirinden oldukça farklı görevlere uygulanması, bu algoritmaların geleneksel yaklaşımlara kıyasla daha üstün performans sergilediğini göstermiştir. Bununla birlikte, bir makine öğrenmesi modelinin belirli bir görevi yerine getirme kabiliyeti, nihayetinde eğitildiği veri setinin kalitesiyle sınırlıdır (Bishop ve Nasrabadi 2006).

Ağ tabanlı verilerin makine öğrenimi modelleriyle birleştirilmesi üzerine yapılan çalışmalar, diğer alanlarla karşılaştırıldığında oldukça sınırlıdır (Hamilton vd. 2017a). Ancak, makine öğrenimi kullanılarak ağ madenciliği görevlerine benzer faydalar sağlanabilecek geniş bir alan bulunmaktadır. Makine öğrenimi ile gerçekleştirilebilecek önemli bir görev, global ve düğüm/kenar düzeylerinde incelenebilecek ağ sınıflandırma görevidir. Örneğin, ağlar halinde temsil edilen kimyasal moleküllerin doğru bir şekilde sınıflandırılması, yeni ilaçların keşfine (Kearnes vd. 2016) yardımcı olabilir. Makine öğrenimi ve ağ veriler kullanılarak gerçekleştirilebilecek bir diğer önemli ağ madenciliği görevi ise eksik kenar (bağlantı) tahminidir (Arrar vd. 2024). Sosyal ağlarında yeni bir kenarın oluşacağını tahmin etmek, iki kullanıcının arkadaş olmasını önermek için kullanılabilir (Adamic ve Adar 2003). Alternatif olarak, protein-protein etkileşim ağında yeni bir bağlantının tahmin edilmesi, iki proteinin bir tür etkileşimde bulunma olasılığının yüksek olduğunu gösterebilir (Kovács vd. 2019).

Ağlar ve makine öğrenmesinin bütünleştirilmesinin sunduğu olanaklara rağmen, ağların diğer veri biçimlerine kıyasla kendine özgü özellikleri nedeniyle birtakım engeller ortaya çıkmaktadır. Mevcut makine öğrenmesi algoritmalarının çoğu, girdileri vektör veya küçük sayısal değer matrisleri şeklinde kabul edecek biçimde tasarlanmış olup, ağ verilerini doğrudan işleme kabiliyetinden yoksundur. Ağ tabanlı makine öğreniminde temel bir diğer zorluk, ağların özünde bulunan yapısal bilgiyi etkili bir şekilde makine öğrenimi modellerine entegre etmektir. Bu entegrasyon, model performansını çeşitli görevlerde artırmak için kritik öneme sahiptir. Örneğin, sosyal ağlarda bağlantı tahmini yaparken, düğümler arasındaki ikili öznitelikleri, ilişki yoğunluğunu veya ortak arkadaş

sayısı gibi özellikleri yakalamak, tahmin doğruluğunu önemli ölçüde artırabilir. Benzer şekilde, düğüm sınıflandırmasında, bir düğümün ağdaki konumu, lokal ve global çevresi hakkında bilgiyi kullanmak, örneğin merkeziyeti veya yakın komşularının özellikleri, daha sağlam ve doğru sınıflandırmalara katkıda bulunabilir. Makine öğrenimi perspektifinden bakıldığında, ağın karmaşık ve öklid olmayan yapısal bilgisini geleneksel bir özellik vektörüne dönüştürmek için doğrudan bir yöntemin eksikliği önemli bir engel teşkil etmektedir. Bu durum, çoğu makine öğrenimi algoritmasının öklid uzayda temsil edilen yapılandırılmış veriler üzerinde çalışmak üzere tasarlanmış olması nedeniyle bir zorluk yaratmaktadır.

Yaygın olarak kullanılan bir yöntem, ağların yapısal bilgilerini özet ağ istatistikleri ile çıkarmaktır (örneğin, çekirdek fonksiyonlar (Gärtner vd. 2003)). Alternatif bir strateji ise, yerel komşuluk yapılarını analiz etmek için özenle tasarlanmış el işçiliğiyle özellikler geliştirmektir (Liben-Nowell ve Kleinberg 2003). Ancak bu yöntemlerin iki temel sınırlaması vardır. Birincisi, etkili özelliklerin tasarlanması uzmanlık gerektirir; bu da alan uzmanlarına duyulan ihtiyacı artırır ve süreci genellikle zaman alıcı hale getirir. İkincisi, el yapımı özellikler esneklikten yoksundur ve dinamik olarak değişen ağlara kolayca uyarlanamaz.



Şekil 1.2 Düğüm temsili öğrenimi

Son araştırmalar, ağların doğasında var olan yapısal incelikleri kapsayan temsillerin elde edilmesine yönelik metodolojilerde önemli bir artışa yol açmıştır. Ağ temsil öğrenimi, semantik arama (Xiong vd. 2017), soru cevaplama (Hao vd. 2017), trafik tahmini (Cui vd. 2019) ve kullanıcılara reklam ve ürün önerme (Wang vd. 2018) gibi çok sayıda başarılı uygulamaya olanak sağlayan umut vadeden yeni bir öğrenme paradigması olarak

belirmiştir. Bu temsili öğrenme tekniklerdeki temel fikir, düğümleri veya tüm ağı daha düşük boyutlu bir vektör uzayına (**Şekil 1.2**) yerleştiren algoritmik bir eşleme fonksiyonu geliştirmeye odaklanmaktadır. Bu eşleme fonksiyonu, yerleştirme uzayındaki geometrik ilişkilerin orijinal ağı özünde var olan yapısını hassasiyetle yansıtmayı sağlamak amacıyla titizlikle optimize edilmektedir. Yerleştirme uzayında optimal bir duruma erişildikten sonra, bu optimize edilmiş vektör yerleştirmeler, sonraki makine öğrenimi görevleri için kritik önem taşıyan öznitelik girdileri olarak işlev görmektedir. Yerleştirmelerin yararları üç yönlüdür:

- Çoğunlukla bilgi içeren sayısal vektörler veya özellikler gerektiren hazır makine öğrenimi algoritmalarına doğrudan girdi olarak kullanılabilirler.
- Öğrenildikten sonra, düğüm sınıflandırma, bağlantı tahmini ve topluluk tespiti gibi farklı ağ analiz görevlerinde uygulanabilirler.
- Genellikle ağ analiziyle ilgili görevlerde daha yüksek performans gösterme eğilimindedirler.

Temel zorluk, öğrenilen yerleştirme fonksiyonunun ağı yapısal bilgisini etkili bir şekilde yakalaması ve koruması, aynı zamanda bu bilgiyi yerleştirme uzayında geometrik olarak temsil edebilmesini sağlamaktır. Böylece, orijinal yapısal uzayda birbirine benzer veya yakın olan düğümler, öğrenilen vektör uzayında da benzer şekilde temsil edilmelidir.

Bahsi geçen ağ temsili öğrenimi sorunlarını ele almak amacıyla, bu tezde, ağ temsili öğrenimi alanında gerçekleştirilen önemli çalışmaların kapsamlı bir incelemesi sunulmaktadır. Literatürde yaygın olarak kullanılan rasgele yürüyüşe dayalı yöntemlerden esinlenerek, düğüm öznitelik vektörlerinin elde edilmesi süreci ayrıntılı bir şekilde ele alınmıştır. Her aşamada, kontrol edilebilen tüm parametreler dikkate alınarak, ağların zengin yapısal bilgileri ile sonraki ağ analiz görev özelliklerinin etkileşimini hesaba katarak öznitelik vektörlerinin kalitesi optimize edilmiş ve daha yenilikçi ve gelişmiş düğüm öznitelik vektörlerinin elde edilmesi sağlanmıştır. Elde edilen bu vektörler, düğüm sınıflandırma, bağlantı tahmini, görselleştirme ve topluluk tespiti gibi çeşitli ağ analiz işlemlerinde kullanılmıştır.

1.1 Motivasyon ve Tez Kapsamı

Birçok alanda, ağlar verilerdeki karmaşık ilişkileri yakalamak için etkili bir temsil aracı olduğunu kanıtlamıştır (Newman 2010). Ağların ve makine öğreniminin birleştirilmesi, ağ yapılarında doğal olarak bulunan zengin ilişkisel bilgileri kullanarak çeşitli alanlarda sayısız olasılığın önünü açmaktadır. Ağ temelli makine öğrenimi yöntemleri, geleneksel yaklaşımların ötesine geçerek, veri arasındaki karmaşık bağlantıları keşfetme ve bu bağlantılardan anlamlı içgörüler türetme kabiliyetini önemli ölçüde artırmaktadır. Örneğin, sosyal ağların ağ yapısını analiz ederek, makine öğrenimi modelleri, daha isabetli arkadaşlık önerileri, içerik tavsiyeleri ve hedefe yönelik reklamlar sunma kabiliyetini önemli ölçüde artırabilmektedir. Makine öğrenimi yöntemlerinin ağlara uygulanması, ağ verilerinin kendine özgü yapısı ve özellikleri nedeniyle birtakım zorluklara yol açmaktadır. Bu bağlamda karşılaşılan başlıca güçlükler şu şekilde sıralanabilir:

- **Veri Yapısının Karmaşıklığı ve Düzensizliği:** Ağlar, geleneksel makine öğrenimi yöntemlerinin genellikle varsaydığı düzenli ve sabit boyutlu veri yapılarından farklıdır. Bu düzensizlik, standart öğrenme algoritmalarının doğrudan uygulanmasını engeller.
- **Ölçeklenebilirlik Sorunları:** Büyük ölçekli ağlar üzerinde makine öğrenimi uygulamak, hesaplama ve bellek açısından önemli zorluklar ortaya çıkarır. Çoğu standart ağ algoritması en az karmaşıklığına sahiptir, burada düğüm sayısını ve kenar sayısını temsil eder. Milyonlarca ve hatta milyarlarca düğüm ve kenar içeren büyük ölçekli ağ verileriyle çalışmayı zorlaştırır.
- **Özellik Mühendisliği:** Ağlar için uygun özelliklerin seçimi ve çıkarılması, model performansını doğrudan etkilemektedir. Geleneksel özellik mühendisliği teknikleri, ağların karmaşık yapısı nedeniyle yetersiz kalabilir. Ağların yapısına göre yeni özellik çıkarma teknikleri geliştirmek, önemli bir araştırma alanı olmuştur.
- **Düğüm ve Kenar Özelliklerinin Temsil Edilmesi:** Ağ verilerinde düğümler ve kenarlar, genellikle farklı özelliklerle karakterize edilir. Bu özelliklerin makine öğrenimi modellerine uygun bir şekilde temsil edilmesi ve işlenmesi önemlidir.

Ancak, düğüm ve kenar özelliklerinin heterojen yapısı, bu temsili zorlaştırmaktadır. Özellik mühendisliği süreci, bu tür veriler için daha karmaşık ve zahmetli olabilir.

- **Dinamik ve Evrimleşen Ağ Yapıları:** Zaman içinde değişen ağ yapıları, statik ağlar için geliştirilen yöntemlerin ötesinde ek zorluklar getirir. Ağın zaman içindeki değişimlerini tutarlı bir şekilde modellemek, özellikle düğüm ve kenarların eklenip çıkarıldığı durumlarda zorlu bir işlemdir. Sürekli güncellenen ağlarda, modelin yeni bilgileri hızlı ve verimli bir şekilde öğrenmesi gerekir, bu da geleneksel makine öğrenme yaklaşımlarının ötesinde teknikler gerektirir.
- **Geleneksel Makine Öğrenmesi Yöntemlerinin Uygulanabilirliği:** Ağlar yapısal verilerdir, ancak ağların sınırlamalarından biri öznitelik vektörlerinin olmamasıdır. Öznitelik vektörleri, düğüm ve kenarların belirli özelliklerini temsil eden sayısal değerler kümesidir. Çoğu klasik makine öğrenmesi algoritması (örneğin, regresyon modelleri, karar ağaçları, destek vektör makineleri), girdi olarak sabit boyutlu öznitelik vektörleri bekler. Ağlarda bu tür vektörlerin doğrudan mevcut olmaması, bu algoritmaların doğrudan uygulanmasını zorlaştırır.

Ağ temsili öğrenimi, ağ yapılarının karmaşıklığını ele alarak, etkili özellik çıkarımı sağlayarak ve hesaplama karmaşıklığını azaltarak ağ üzerinde makine öğrenimi uygulamanın zorluklarına çözümler sunmaktadır. Ağ temsili öğrenimi, ağın yapısal ve topolojik özelliklerini koruyarak, yüksek boyutlu ve seyrek ağ yapısını, düşük boyutlu, yoğun ve anlamlı vektörler olarak temsil etmeyi amaçlamaktadır. Literatürde mevcut yerleştirme algoritmaları arasında, rasgele yürüyüş temelli algoritmalar oldukça başarılı olduğunu kanıtlamıştır. Bu yöntemler, önceden belirlenmiş adım sayısına sahip çok sayıda rasgele yürüyüş oluşturarak düğüm çevresinden bilgi toplamaktadır. Söz konusu algoritmalar, karmaşık ağ yapılarını analiz etmek ve temsil etmek için etkili bir araç olarak öne çıkmaktadır.

Bu doğrultuda, tez kapsamında ağ yapılı veriler için ağ temsili öğrenimi alanında kapsamlı bir araştırma gerçekleştirilmiştir. Ayrıca, yapısal ağ verileri için rasgele yürüyüş prensibine dayanan iki farklı ağ temsil öğrenimi (düğüm yerleştirme) yöntemi

önerilmiştir. Bu yenilikçi yaklaşımlar, ağın topolojik özelliklerini dikkate alarak düğüm bağlantısalılığının örnekleme sürecine entegre edilmesini sağlamıştır. Bu sayede, ağın yapısal karakteristiklerini koruyan ve aynı zamanda hesaplama karmaşıklığını önemli ölçüde azaltan örnekleme stratejileri geliştirilmiştir.

1.2 Tezin Özgün Değeri

Bu tez çerçevesinde yapılan katkılar şu şekilde sıralanabilir:

- Literatürde öncü ağ temsil öğrenimi yöntemlerinden olan ve rasgele yürüyüş prensibini kullanan node2vec yaklaşımı (Grover ve Leskovec 2016) modife edilerek derece tabanlı node2vec yöntemi önerilmiştir (Mohammed ve Gündüç 2022). Node2vec yöntemi her bir düğümden sabit sayıda yanlı rasgele yürüyüş başlatarak düğüm çevresinden yapısal bilgi toplamayı amaçlar. Ancak gerçek dünya ağları, düşük dereceli çok sayıda düğüm ve az sayıda yüksek dereceli düğüm içeren ölçeksiz ağlara sahiptir. Sabit yürüyüş sayısı kullanıldığında düşük dereceli düğümlerin yüksek dereceli düğümlere göre daha sık ziyaret edilmesine yol açar. Bu nedenle, yüksek dereceli düğümler için daha fazla rasgele yürüyüş gerekli olsa da, bilgisayar kapasitesi sınırlamaları ve düşük dereceli düğümler için gereksiz yürüyüşler sorun oluşturmaktadır. Önerilen yöntemde, rasgele yürüyüşlerin sayısı, düğümün derecesiyle orantılı olarak oluşturulmuştur. Sonuç olarak, önerilen yöntem yaklaşık %50 daha az yürüyüş yaparak aynı hassasiyette düğüm yerleştirme vektörler elde edilmesini sağlamıştır.
- Homofili (benzerler arasında bağlantı eğilimi) ve yapısal rol benzerliği (farklı bağlamlarda benzer bağlantı kalıplarına sahip düğümler) birbirlerini tamamlayan ve gerçek dünya ağlarının doğasında var olan iki önemli kavramdır. Mevcut ağ temsili öğrenimi yöntemlerin çoğu düğümlerin yerel yapısal kalıplara duyarsızken temel olarak homofili üzerine odaklanmaktadır. Bu nedenle, anonim yürüyüş kavramına dayanarak düğüm temsili vektörlerini öğrenmek için komşu bilgisini, yapısal bilgiyi ve düğümlerin yerel yapısal kalıplarını birleştiren Geçiş Olasılık Matrisi (Transition Probability Matrix - TPM) adını verdiğimiz esnek bir yaklaşım önerilmiştir (Mohammed ve Gündüç 2023). Önerilen yaklaşım, düğümün kimliği

yerine komşu yapısal desenlerin yakalanmasına odaklanır. Fikir, birkaç yakın komşuyu ziyaret ederek bir düğümün yerel bağlantı yapısını yakalamaktır. Dolayısıyla, hesaplama süresinden tasarruf sağlar, performansı artırır ve bir ağın derinliklerine inmeden bilgi toplamayı mümkün kılar. Önerilen yaklaşım, literatürde güncel ve iyi performans sergileyen (state-of-the-art) algoritmalara kıyasla çeşitli ağ analiz görevlerinde üstün performans sergilemiştir.

- Önerilen TPM modeli, yalnızca aynı ağ üzerinde eğitim ve tahmin yapmakla kalmayıp, benzer bağlantı yapılarına sahip ağlardan bilgi öğrenerek yeni ağlar üzerinde tahminler yapma görevini literature kazandırmıştır. Buradaki temel fikir, düğümlerin ayrıntılarını dikkate almaksızın bağlantıların benzerliğinden faydalanmaktır. Geleneksel sınıflandırma problemlerinde, eğitim verisi aynı ağ üzerinde tahminler yapmak için kullanılır. Ancak, önerilen yaklaşım, yerleştirme işleminin performansını aynı topolojik yapıya sahip farklı bir ağda test eden literatürde ilk çalışma olmuştur. TPM modeli, dört farklı öznitelik çıkarma algoritmalarına kıyasla bağlantı tahmini görevinde önemli ölçüde üstün performans sergilemiştir. Bu sonuçlar, TPM modelimizin, benzer topolojilere sahip farklı ağlarda dahi düğümlerin yerel yapısal örüntülerini başarıyla yakaladığını kanıtlamıştır.
- Metin sınıflandırma, Doğal Dil İşleme'deki (Natural Language Programming – NLP) temel görevlerden biridir. Metin temsili öğrenimi, metin sınıflandırma probleminin ilk ve temel adımıdır. Geleneksel doğal dil işleme modelleri karmaşık ilişkileri ve bağlamsal bilgileri yakalama konusunda sınırlı yetenek gösterirken, çok anlamlılığı (bir kelimenin birden fazla anlamı) ele almanın zorluğu ve daha derin anlamsal anlayış gerektiren görevler için pek uygun değildir. Metin verileri ve ağ verileri görünüşte farklı olsa da, kelimeler arasındaki doğal ilişkiler bir metindeki kelimelerin ağ yapısının düzenlenmesine yardımcı olur. Bu doğrultuda, tez çalışmasında, metin verilerinin incelenmesinde TPM yöntemi benimsenmiştir. Anlamlı cümlelerin ağ yapısını esas alan Kılavuzlu Geçiş Olasılık Matrisi (Guided Transition Probability Matrix - GTPM) adını verdiğimiz yeni bir metin yerleştirme yöntemi önerilmiştir (Mohammed ve Gündüç 2024). Algoritmanın tasarımı, metin verilerinin gizli içeriğinin yanı sıra sözdizimsel ve anlamsal unsurları da oluşturan ve hem yerel bağlamı hem de kelimeler arasındaki küresel

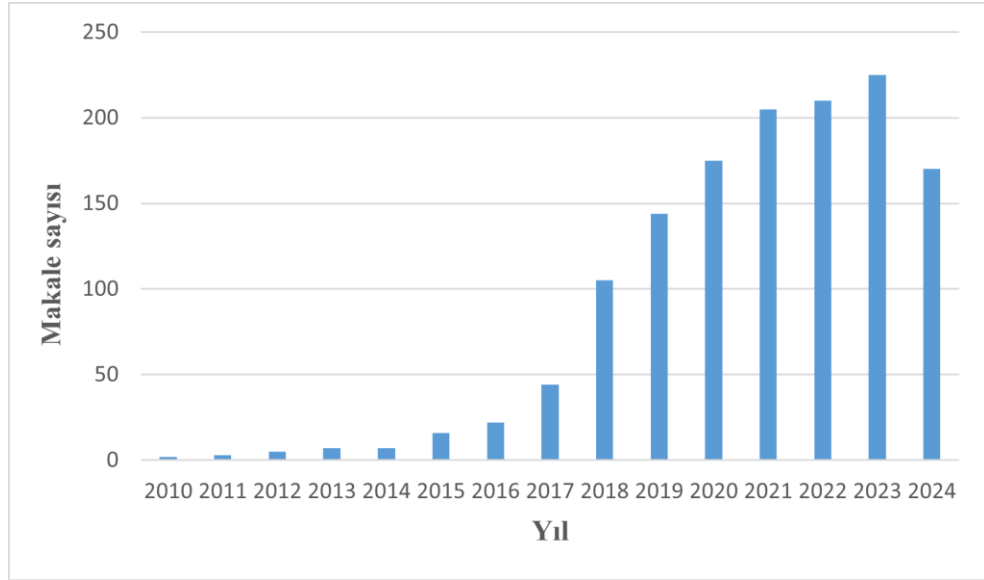
ilişkileri dikkate alan bir yerleştirme vektörü oluşturmayı amaçlamıştır. Bu yöntemin güvenilirliği ve başarısı, yaygın olarak kullanılan 6 farklı gerçek veri kümelerini kullanarak literatürde iyi bilinen yöntemler ile kıyaslayarak takip edilmiştir. Deneysel sonuçlar, önerilen modelin metin sınıflandıra görevinde diğer yöntemlere göre daha iyi performans gösterdiğini göstermiştir.

1.3 Tezin Yapısı

2. bölümde, literatürdeki temel kavramlar ve yaklaşımlar ele alınmış, ağ yapılarının ve makine öğrenmesi yöntemlerinin birleşiminden elde edilen avantajlar incelenmiştir. 3. bölümde, ağ yapıları, temel ağ kavramları ve ağ analizi teknikleri ele alınmıştır. 4. bölümde, tez kapsamında geliştirilen üç yenilikçi algoritma detaylı bir şekilde anlatılmıştır. Her bir algoritmanın, ağ ve metin verilerinin temsilinde karşılaşılan zorluklara çözüm ürettiği ve bu yöntemlerin çeşitli uygulamalardaki potansiyelini ortaya koyduğu gösterilmiştir. 5. bölümde, önerilen algoritmaların performansı değerlendirilmiş, deneysel sonuçlar sunulmuş ve tartışılmıştır. Son olarak, 6. bölümde, sonuçlar ve öneriler sunulmuş olup, araştırmanın genel değerlendirmesi yapılmıştır.

2. LİTERATÜR ÖZETİ

Ağlar, karmaşık ilişkisel verileri temsil etmekte yaygın olarak kullanılan bir yapı sunmaktadır. Ağ verileri, çeşitli uygulama alanlarında yaygın olarak karşımıza çıkmaktadır. Bu alanlar arasında ilaç keşfinde moleküler ağlar (Liu vd. 2019a), genetikte protein-protein etkileşim ağları (Kashyap vd. 2018), sosyal medyada sosyal ağlar (Wang vd. 2019) ve bilimsel iş birliklerinde yayın ağları (West vd. 2016) bulunmaktadır. Çoğu ağ verisinin karmaşık ve yüksek boyutlu olması sebebiyle, araştırmacıların bu verilerden değerli bilgiler çıkarması oldukça güç bir süreç haline gelmiştir. Bu nedenle, ağ verilerinin işlenmesi ve araştırmacıların çeşitli ağ analiz görevlerde kullanabileceği bir forma (sabit boyutlu vektörlere) dönüştürülmesi, son derece önemli bir aşamayı teşkil etmektedir (Hamilton 2020). Ağ temsili öğreniminin temel amacı, ağdaki düğüm ve kenar gibi varlıkların vektör temsillerini çıkararak, düğüm sınıflandırma, bağlantı tahmini ve topluluk tespiti gibi farklı ağ analiz görevlerini daha etkin hale getirmektir. Bu bağlamda, ağ temsili öğrenimi, ağ analizinde performansı önemli ölçüde artırabilme potansiyeline sahip olduğu için kritik bir öneme sahiptir.



Şekil 2.1 2010'dan Ağustos 2024'e kadar ağ yerleştirme üzerine odaklanan makalelerin istatistikleri

Son yıllarda ağ temsil öğrenimi, veri madenciliği, makine öğrenimi ve ağ analiz alanlarında vazgeçilmez ve öne çıkan bir konu haline gelmiştir. Springer Link, IEEE Xplore, ACM ve Elsevier Science Direct gibi önde gelen akademik kütüphanelerde, ağ yerleştirme, çizge yerleştirme ve ağ temsil öğrenimi ile ilgili anahtar kelimeler kullanarak kapsamlı bir akademik literatür taraması gerçekleştirilmiştir. Araştırmamız neticesinde, son beş yıl içerisinde bu alanda yayımlanmış 900'ün üzerinde ilgili makale tespit edilmiştir. **Şekil 2.1**, 2010'dan Ağustos 2024'e kadar bu konuda yayımlanan çalışmaların sayısal dağılımını göstermektedir. Ağ temsili öğrenimi çalışmalarının sayısı, on dört yıllık süreç zarfında kayda değer bir artış göstermiştir.

Ağ verisinin temsili, görüntü ve metinsel verilerden farklı olup, kendine özgü zorluklar barındırmaktadır (Baptista vd. 2023). Metinsel verilerde, kelimeler bir cümle içerisinde birbirine bağlı olup, o cümlede sabit bir konuma sahiptir. Görüntü verilerde ise, pikseller düzenli bir ızgara uzayında sıralanmış olup, bir ızgara matrisi ile temsil edilebilmektedir. Ancak, ağlardaki düğümler ve kenarlar düzensiz bir yapıya sahip olup (sıralı değildir), kendilerine özgü özellikler taşımaktadır. Ancak, ağ yapısını ve yakınlık ilişkilerini koruyarak ağ unsurlarını gizli uzaya haritalama sürecini oldukça zorlu bir hale getirmektedir. Sosyal ağlar bağlamında, kullanıcılar genellikle çok sayıda bağlantıya (komşuya) sahip olmakla birlikte, memleket, eğitim düzeyi ve ilgi alanları gibi çeşitli kişisel bilgileri de barındırmaktadır. Bu çok boyutlu yapı, ağ topolojisinin ve özelliklerinin bütünlüğünü koruma noktasında önemli zorluklar teşkil etmektedir. Ayrıca, birçok gerçek dünya ağı, ağ yapılarının ve düğüm özelliklerinin zamanla değişebildiği dinamik davranışlar sergilemektedir (Yang vd. 2024). Bu değişkenlik, ağ yapısını doğru bir şekilde yakalamada ve ağ varlıklarını vektör uzayına haritalamada önemli zorluklar oluşturmaktadır.

Tarihsel açıdan değerlendirildiğinde, ağ temsili öğrenimi modellerinin ilk modelleri ağ çekirdekleri olarak karşımıza çıkmaktadır. Ağ çekirdeği yöntemlerinin kavramsal temeli, 1968 yılında ortaya konulan ve alanın temel taşlarından biri olarak kabul edilen Weisfeiler-Lehman izomorfizm testine dayanmaktadır (Weisfeiler ve Lehman 1968). Ağ çekirdekleri, ağlar ve ağların varlıkları arasındaki benzerliği ölçmeyi amaçlayan çekirdek fonksiyonlardır (Nikolentzos vd. 2021). Ağ çekirdeklerinin temel ilkesi, orijinal ağları alt

yapılara ayrıştırmak ve bu alt yapıların özelliklerine dayanarak vektörel yerleştirme temsilleri inşa etmektir. Ağ çekirdekleri, temelde iki ana kategoriye ayrılmaktadır: ağlar için çekirdekler ve ağlar üzerinde çekirdekler. İlki, iki ağ arasındaki benzerliği ölçmeyi amaçlarken, ikincisi ise ağ düğümleri arasındaki benzerliği tahmin etmeye odaklanmaktadır.

Weisfeiler-Lehman izomorfizm testini temel alan bir dizi ağ çekirdeği, ağ varlıklarının benzerliğini ölçmek amacıyla oluşturulmuştur. Bunlar arasında Weisfeiler-Lehman çekirdekleri (Weisfeiler ve Leman 1968), Weisfeiler-Lehman alt ağaç çekirdekleri (Shervashidze vd. 2011, Morris vd. 2017, Nguyen vd. 2021) ve rasgele yürüyüş tabanlı yöntemler (Orsini vd. 2015, Ye vd. 2022) bulunmaktadır. Çeşitli çekirdek tabanlı modeller, kullanışlı olmalarına rağmen, özellikle büyük ölçekli ağlarda kullanıldıklarında hesaplama karmaşıklığı gibi önemli bir kısıtlamaya sahiptir. Bu modeller, alt yapıların dağılımını dikkate alarak hesaplama süresini azaltmayı hedeflese de, bu yaklaşımlar karmaşıklığı artırmaktadır ve küresel yapıyı anlama yeteneğini zayıflatmaktadır. Bu nedenle, ağ çekirdeklerinin hesaplanması NP-zor (NP-hard) bir problem olarak nitelendirilmektedir (Borgwardt ve Kriegel 2005).

Geçtiğimiz yıllar içerisinde, ağları ve ağ nesnelerini (düğümleri ve bağlantıları) sabit uzunlukta vektörlere eşlemek amacıyla çok sayıda ağ temsil öğrenimi modeli geliştirilmiştir. Bu modeller, temel olarak üç ana kategoriye ayrılmaktadır: matris çarpanlara ayırma modelleri, rasgele yürüyüşe dayalı modeller, derin sinir ağı modelleri.

2.1 Matris Çarpanlara Ayırma Modelleri

Matris çarpanlara ayırma temelli ağ yerleştirme yöntemleri, doğrudan Temel Bileşenler Analizi (Principal Component Analysis - PCA) (Wold vd. 1987), Doğrusal Ayırma Analizi (Linear Discriminant Analysis - LDA) (Ye vd. 2004) veya Çok Boyutlu Ölçekleme (Multidimensional Scaling - MDS) (Cox ve Cox 2000) gibi doğrusal boyut indirgeme yöntemlerinden esinlenmiştir. Diğer yöntemler ise, MDS'in bir uzantısı olan Isomap (Samko vd. 2006), Yerel Doğrusal Yerleştirme (Locally Linear Embedding - LLE) (Roweis ve Saul 2000) veya t-dağıtımlı olasılıksal komşu yerleştirme (t-Distributed

Stochastic Neighbor Embedding- t-SNE) (Van der Maaten ve Hinton 2008) gibi doğrusal olmayan boyut indirgeme yöntemlerinden esinlenmiştir.

Bu yöntemlerin temel hedefi, ağları tanımlayan yüksek boyutlu matrisleri (örneğin, komşuluk matrisi, Laplace matrisi) daha küçük boyutlu matrislere ayırtmak ve akabinde benzerliği ve yakınlığı hassasiyetle temsil eden yerleştirme yapıları öğrenmektir.

Ağın topolojik özelliklerini muhafaza etmeyi hedefleyen matris ayırtma modelleri, iki temel kategoriye sınıflandırılır: Laplace özharitaları vasıtasıyla ağ yerleştirme ve düğüm yakınlık matrisi ayırtması. Laplace özharitaları yöntemleri, her bir düğümü ilk (k) özdeğere tekabül eden Laplace özvektörleri vasıtasıyla temsil etmeyi hedeflemektedir. Buna karşılık, düğüm yakınlık matrisi ayırtma yöntemleri ise, düğümler arasındaki ilişkileri doğrudan matris operasyonları yoluyla yakalamaya çalışmaktadır.

IsoMap, en kısa yol matrisini kullanarak çok boyutlu ölçekleme (MDS) yöntemini uygular ve böylece düğüm yerleştirme vektörlerini öğrenir (Samko vd. 2006). LLE yöntemi, komşu düğümlerin birbirini etkilediği ağırlık matrisini yeniden oluşturarak düğüm benzerliğini öğrenir ve elde edilen matrisi ile çok yönlü (manifold) matrisini öğrenmek için bu prosedürü tekrarlar (Roweis ve Saul 2000). LPP yöntemi ise, ağırlık matrisini ısı çekirdeği olarak tahmin eder ve Laplace Özharitalar tekniği ile 'nin indirgenmesi yoluyla çok yönlü matrisini öğrenir (He ve Niyogi 2003).

Matris ayırtma, ağ yerleştirme problemlerinde yaygın olarak kullanılmasına rağmen, hala birtakım sınırlamalara sahiptir. Hesaplama karmaşıklığı açısından, milyonlarca düğüm içeren büyük ölçekli ağlarda matris ayırtma, matrisi küçük boyutlu matrisler çarpımına ayırtma işleminin gerektirdiği süre nedeniyle önemli zorluklar yaşamaktadır (Cao vd. 2015). Ayrıca, gözlemlenmeyen ve eksik değerler içeren tamamlanmamış ağları işlemekte yetersiz kalmakta ve ağ verilerinin yetersiz olduğu durumlarda genelleştirilmiş vektör yerleştirmelerini öğrenememektedir (Safavi ve Koutra 2020, Baptista vd. 2023).

2.2 Rasgele Yürüyüşe Dayalı Modeller

Matris çarpanlara ayırma yöntemlerinin sınırlılıklarından yola çıkarak, düğümlerin yerel komşuluklarını ve özelliklerini rasgele yürüyüşler aracılığıyla korumaya odaklanan alternatif bir yaklaşım geliştirilmiştir (Fouss vd. 2007). Bu yaklaşım, temel olarak, Mikolov ve çalışma arkadaşlarının (Mikolov vd. 2013) doğal dil işleme alanında geliştirdikleri Skip-gram modelinden ilham alarak, bir düğümün yerleştirme temsiline dayalı olarak komşuluk ilişkilerinin gözlemleme olasılığını maksimize etmeyi amaçlamaktadır. Bu tür bir amaç fonksiyonu, tek katmanlı bir sinir ağı üzerinde olasılıksal dereceli alçalma (stochastic gradient descent) yöntemi kullanılarak etkin bir şekilde optimize edilmiştir; bu da hesaplama karmaşıklığının azalmasını sağlamıştır.

Rasgele yürüyüş tabanlı yöntemler, ağ temsillerindeki etkinlikleri nedeniyle önemli ölçüde dikkat çekmiştir. Bu yöntemler, ağdaki her düğüm için rasgele yürüyüşler oluşturarak ağı yapısını yakalamayı ve aynı rasgele yürüyüşlerde görünen düğümler için benzer düğüm yerleştirme vektörleri üretmeyi amaçlamaktadır. Rasgele yürüyüşlerde düğümlerin birlikte bulunma durumunun düğüm benzerliği ölçütü olarak kullanılması, önceki araştırmalarda benimsenen sabit yakınlık ölçütlerine nazaran daha uyarlanabilir olduğu tespit edilmiştir. Bu yaklaşım, çizge içindeki düğümler arasındaki dinamik ilişkileri yakalamada daha yüksek bir esneklik sağlamıştır. Neticede, çeşitli uygulama alanlarında ümit vadeden bir performans sergileyerek, ağ temsil görevlerini iyileştirmedeki etkinliğini ortaya koymuştur.

DeepWalk (Perozzi vd. 2014) ve node2vec (Grover ve Leskovec 2016) modelleri, düğüm yerleştirme alanında önemli bir ilerleme sağlayan öncü rasgele yürüyüş tabanlı modeller olarak değerlendirilmektedir. Bu modeller, ağ yapılarındaki düğümleri vektör uzayına yerleştirmeyi başarıyla gerçekleştirerek, ağlardaki düğümlerin yapısal ve semantik benzerliklerini koruyacak şekilde temsil etme yeteneğine sahiptirler. DeepWalk ve node2vec, doğal dil işleme alanındaki word2vec yerleştirme yöntemini (Mikolov vd. 2013) temel almaktadır. Word2vec, aynı cümlede birçok kez birlikte geçen kelimelerin benzer anlamlara sahip olduğu gözlemine dayanmaktadır. DeepWalk ve node2vec, bu varsayımı ağlara uyarlayarak, rasgele yürüyüşlerde birlikte ortaya çıkan düğümlerin

benzer olduğunu öne sürmektedir. Dolayısıyla, bu yöntemler komşu düğümler için benzer düğüm yerleştirme vektörleri üretmektedir.

DeepWalk, düğüm temsillerini öğrenmek için kesik rasgele yürüyüşlerden elde edilen yerel bilgileri kullanmaktadır. Başlangıçta, DeepWalk ağ üzerinde kesik rasgele yürüyüşler aracılığıyla cümle benzeri düğüm dizileri oluşturur. Daha sonra, optimize aşaması, düğüm yerleştirmesine bağlı olarak bir düğüm komşusunun gözlemlenme olasılığını maksimize etmek amacıyla, düğüm dizilerine Skip-gram modeli (Mikolov vd. 2013) uygulamaktadır. DeepWalk yönteminin geleneksel sınıflandırma yöntemlerinden daha üstün performans sergilediğini göstermek amacıyla BlogCatalog, Flickr ve YouTube gibi çeşitli sosyal ağlar üzerinde deneyler gerçekleştirmiştir. Özellikle sınırlı etiketli veri senaryolarında, bu üstünlük belirgin bir şekilde ortaya çıkmıştır. Elde edilen bulgular, DeepWalk'ın ağ içerisindeki topluluk yapılarını ve ilişkileri etkin bir biçimde yakaladığını göstermiş olup, bu yöntemin ölçeklenebilirliğini ve çevrimiçi öğrenme görevlerine uygulanabilirliğini sergilemiştir.

DeepWalk kesik rasgele yürüyüşleri kullanarak yalnızca yerel yapısal bilgiyi yakalayabilmekte, küresel yapısal bilgiyi ise göz ardı etmektedir. Bu sınırlamanın üstesinden gelmek için Grover ve Leskovec (2016), orijinal ağdan hem yerel hem de küresel yapısal bilgiyi dahil ederek düğüm bağlamlarını örnekleyen geliştirilmiş bir yanlı rasgele yürüyüş örnekleme stratejisi ile node2vec yöntemini önermişlerdir. Bir düğümün komşuluğunun keşfini zenginleştirmek amacıyla, node2vec, genişlik öncelikli (Breadth First Search - BFS) ve derinlik öncelikli (Depth First Search - DFS) ağ aramalarını dengeleyen yanlı rasgele yürüyüşler kullanmaktadır. Bu sayede hem yerel hem de küresel bilgiler elde edilmektedir. Söz konusu denge, p ve q olmak üzere iki parametre aracılığıyla sağlanmaktadır. Geri dönüş parametresi olarak bilinen p parametresi, mevcut düğüme geçildikten sonra bir önceki düğüme geri dönme olasılığını düzenlemektedir. Buna karşılık, içeri-dışarı parametresi olarak adlandırılan q parametresi ise, yürüyüşün doğrudan komşuluk içerisindeki düğümleri yeniden ziyaret etme eğilimini etkilemektedir. Gerçekleştirilen deneyler, node2vec'in çok etiketli sınıflandırma ve bağlantı tahmini görevlerinde, gerçek dünya veri kümelerinde literatürde güncel ve iyi performans sergileyen yöntemleri sırasıyla %26,7 ve %12,6'ya ulaşan bir performans

artışıyla geride bıraktığını ortaya koymuştur. Öğrenilen düğüm yerleştirmeleri, kenar özniteliklerine de genişletilerek bağlantı tahmini performansını daha da ileriye taşımıştır. Algoritma, gürültülü veya eksik kenarlara karşı sağlamlık göstermiş ve büyük ölçekli ağlara verimli bir şekilde uyarlanabilirliğini kanıtlamıştır.

Perozzi vd. (2017), DeepWalk modelinin geliştirilmiş bir versiyonu olan WalkLets modelini literatüre kazandırmışlardır. Bu modelde, ağın yapısal bilgisini daha etkin bir şekilde yakalamak amacıyla rasgele yürüyüş örnekleme yaklaşımını modifiye etmişlerdir. Yapılan değişiklik, yürüyüşün her adımda birden fazla düğümü atlayabilmesine ve üzerinden geçebilmesine olanak tanımaktadır. Bu modifikasyon, örnekleme stratejilerinin geçiş matrisinin gücünden faydalanmasını sağlayarak, daha küresel bir ağ yapısının yakalanmasına imkân vermektedir. Çalışma, çeşitli sosyal ağlar üzerinde gerçekleştirilen çok etiketli sınıflandırma görevlerinde DeepWalk ve LINE (Tang vd. 2015b) yöntemlerini geride bırakarak ve büyük ölçekli ağlara uygulanabilirliğini kanıtlayarak önemli katkılar sağlamıştır. Elde edilen sonuçlar, Micro-F1 skorlarında DeepWalk ve LINE'a göre sırasıyla %10 ve %58'e varan oranlarda iyileştirmeler kaydedildiğini ortaya koymuştur (Perozzi vd. 2017).

Jeong vd. (2020), DeepWalk'ın modifiye edilmiş bir versiyonu olan div2vec modelini önermişlerdir. İki model arasındaki temel fark, div2vec'in rasgele yürüyüş yolunda bir sonraki düğümü seçme yönteminde, komşu düğümlerin derecesine göre belirlenmektedir. Komşu düğümlerin derecesine vurgu yapılması, modellerin sosyal ağlarda öne çıkan düğümlerin önemini öğrenmelerine yardımcı olmayı amaçlamıştır. Önerilen modelin etkinliği, MovieLens, WATCHA ve LINE Wallet Kupon hizmetleri veri kümeleri kullanılarak deneysel olarak değerlendirilmiştir. Elde edilen sonuçlar, div2vec'in hem öneri çeşitliliğini hem de doğruluğunu artırdığını göstermiş, böylece gerçek dünya uygulamalarındaki etkinliğini kanıtlamıştır (Jeong vd. 2020).

Zhang vd. (2019) mevcut rasgele yürüyüş temelli ağ yerleştirme tekniklerinin ölçekten bağımsız ağları işlemedeki verimsizliklerini çözmek için DiaRW yöntemini önermiştir. Araştırmacılar, topolojik bilgiyi daha iyi yakalamak ve gereksiz tekrarları azaltmak amacıyla değişken uzunluklu yürüyüşlere sahip derece yanlı bir rasgele yürüyüş

algoritması geliřtirmiřlerdir. Metodoloji, yksek dereceli dğmleri tercih eden olasılıksal bir geri izleme mekanizması ve dğm merkezietine dayalı yryř uzunluęu ayarlamasını iermektedir. Dğm sınıflandırması ve baęlantı tahmini zerine yapılan deneyler, DiaRW'nin etkililięini ve verimlilięini gstermiř olup, mevcut yntemlerden on kat daha hızlı olduęunu ve tek bir makine zerinde saatler iinde milyonlarca dğm ve kenardan oluřan aęları iřleyebildięini ortaya koymuřtur (Zhang vd. 2019).

Qi (2022) geleneksel rasgele yryř rneklemeye yaklařımlarında gzlemlenen yksek dereceli dğmlere ynelik nyargıyı hafifletmek amacıyla Ters Derece Rasgele Yryř (Inverse Degree Random Walk - IDRW) olarak adlandırılan zgn bir rasgele yryř rneklemeye yntemi nermiřtir. IDRW, aday dğmn derecesinin tersini kullanarak dğmleri seici bir řekilde filtrelemekte, bylelikle dřk dereceli dğmleri n plana ıkarmakta ve birrnek daęılımdan sapmaları en aza indirmektedir. Arařtırmacılar, bu yaklařımı daha geniř bir ereve ierisinde derece aęırlıklandırması zerinde ayarlanabilir bir kontrol saęlayan α parametresini tanıtarak geniřletmiřlerdir. Gerek dnya veri kmeleri zerinde kapsamlı deneyler gerekleřtirilmiř ve IDRW'nin, tahmin doęruluęu aısından mevcut yntemleri ařtıęı gsterilmiřtir (Qi 2022).

Dong vd. (2017) dğmlerin ve baęlantıların oklu trlere sahip olduęu ayrıřık aęlarda temsil ęrenme sorununu ele almıřtır. Yazarlar, dğmler iin ayrıřık komřuluklar oluřturmak amacıyla meta-yol tabanlı rasgele yryřlerden yararlanan iki leklenebilir model olan metapath2vec ve metapath2vec++ modellerini nermiřlerdir. Metapath2vec modeli, dğm yerleřtirmelerini ęrenmek iin ayrıřık skip-gram modelini kullanırken, metapath2vec++ modeli bir dğmn ayrıřık komřuluęunun tahminini iyileřtirmek amacıyla ayrıřık negatif rneklemeyi dahil etmiřtir. Gerek dnya veri kmeleri zerinde yrtlen kapsamlı deneyler, her iki modelin de dğm sınıflandırması, kmeleme ve benzerlik araması gibi eřitli grevlerde mevcut yntemleri nemli lde ařtıęını gstermiřtir. alıřma ayrıca, bu modellerin farklı dğm trleri arasındaki anlamsal iliřkileri otomatik olarak keřfetme yeteneęini de ortaya koymuřtur (Dong vd. 2017).

Li vd. (2016) znitelikli aę sınıflandırması iin zgn bir yntem olan Ayırt Edici Derin Rasgele Yryř (Discriminative Deep Random Walk - DDRW) modelini nermiřtir.

DDRW, DeepWalk (Perozzi vd. 2014) yöntemini genişleterek, sınıflandırma hedefini ve düğümleri ağıın topolojik yapısını koruyan gizli bir uzaya yerleştirme hedeflerini bütünleştirmiştir. Araştırmacılar, ağıın topolojik yapısını yakalamak için kesik rasgele yürüyüşleri ve gizli temsilleri öğrenmek için word2vec (Mikolov vd. 2013) dil modelini kullanmışlardır. Ardından, bu yerleştirme hedefini bir sınıflandırıcı hedefiyle birleştirerek, ortak hedefi olasılıksal dereceli alçalma yöntemiyle optimize etmişlerdir. Çeşitli gerçek sosyal ağlar üzerinde yürütülen deneysel sonuçlar, DDRW 'nin çok etiketli ağ sınıflandırma görevlerinde DeepWalk'u önemli ölçüde aştığını göstermiştir. Bununla birlikte çeşitli etiketlenmiş veri yüzdelerinde temel yöntemleri tutarlı bir şekilde aştığı kanıtlanmıştır (Li vd. 2016).

Pan vd. (2016) ağ yapısı, düğüm içeriği ve düğüm etiketlerinden elde edilen bilgileri eş zamanlı olarak kullanarak optimal düğüm temsillerini öğrenmeyi amaçlayan üç taraflı derin ağ temsili modeli TriDNR'yi önermiştir. TriDNR, çeşitli boyutlarda işleyen bağlantılı bir derin doğal dil modülünü kullanmaktadır: ağ yapısı anlayışını geliştirmek için rasgele yürüyüşlerde komşu düğümleri gözlemleme olasılığını en üst düzeye çıkarma, düğümlerle ilişkili kelime dizilerinin eş-oluşumunu maksimize ederek düğüm-kelime korelasyonlarını yakalama ve sınıf etiketleri verildiğinde kelime dizilerinin olasılığını en üst düzeye çıkararak etiket-kelime ilişkilerini modellemektedir. İki atıf ağı (DBLP ve CiteSeer-M10) üzerinde gerçekleştirilen deneysel değerlendirmeler, TriDNR'nin literatürde güncel ve iyi performans sergileyen yöntemlere kıyasla önemli iyileştirmeler sağladığını ve sınıflandırma doğruluğunda %79'a varan bir artış elde ettiğini göstermiştir (Pan vd. 2016).

Liu vd. (2023) ağırlıklı ağlarda yanlı rasgele yürüyüşlerde kenar ağırlıklarını daha etkin bir şekilde kullanmak üzere popüler node2vec yönteminin bir uzantısı olan node2vec+'yı önermiştir. Çalışma node2vec'in yoğun ve ağırlıklı çizgiler için daha az etkili olduğunu göstermiştir. Node2vec+, yanlı rasgele yürüyüş sürecine kenar ağırlıklarını dahil ederek bu sınırlamayı ele almakta ve böylece seyrek bağlantılı düğümleri daha iyi ayırt edebilmektedir. Çeşitli veri kümeleri ve gerçek dünya gen etkileşim ağları üzerinde yapılan deneyler, çeşitli gen işlevi ve hastalık tahmin görevlerinde node2vec'e kıyasla üstün bir performans gösterdiğini ortaya koymuşlardır (Liu vd. 2023).

Salamat vd. (2020) dengeli rasgele yürüyüşleri kullanarak ağlardaki düğümlerin özellik temsillerini öğrenmek için yenilikçi bir örnekleme algoritması olan BalNode2Vec'i önermiştir. Çalışma, genellikle yüksek giriş dereceli düğümlere yönelik yanlılık gösteren ve daha derin ağ bağlantılarını yakalamakta yetersiz kalan geleneksel rasgele yürüyüşlerin sınırlamalarını ele almıştır. Öncelikle rasgele yürüyüşlerdeki düğümlerin sıklığını ölçer ve bunu düğümlerin derece dağılımıyla karşılaştırır. Ardından, genellikle yetersiz temsil edilen düşük dereceli düğümlerin temsillerini dengelemek için yürüyüş olasılıklarını yeniden ayarlar. Dengelenmiş rasgele yürüyüşler, akabinde nihai düğüm yerleştirme işlemini öğrenmek üzere skip-gram modeline beslenir. Çeşitli gerçek dünya ağları üzerinde yapılan deneyler, BalNode2Vec'in bağlantı tahmin görevlerinde en gelişmiş yerleştirme yöntemlerinden daha üstün performans sergilediğini ortaya koymuştur (Salamat vd. 2020).

Heidari ve Papagelis (2020) rasgele yürüyüşlere dayalı evrilen ağların temsillerini öğrenmek için özgün bir yöntem olan EVONRL'yi önermiştir. Yazarlar, yalnızca statik ağlara uygulanabilen mevcut ağ temsil öğrenme yöntemlerinin sınırlamalarını ele almışlardır. EVONRL, ağ topolojisi değiştikçe rasgele yürüyüşleri dinamik olarak güncellemek için verimli bir indeksleme ve arama kütüphanesi kullanarak temsillerin doğruluğunu korumaktadır. Gerçek veri kümeleri üzerinde yapılan deneyler, EVONRL'nin evrilen ağların temsillerini etkili bir şekilde öğrendiğini ve önemli ağ değişikliklerinde bile yüksek doğruluğu koruduğunu göstermiştir (Heidari ve Papagelis 2020).

Tang vd. (2015) rasgele yürüyüşlere dayanmamakla birlikte, hesaplamasal olarak DeepWalk R27 ve Node2vec R28 ile ilişkili olduğundan, sonuçları literatürde genellikle bu yöntemlerle karşılaştırılmaktadır. LINE, iki bölümden oluşan bir kayıp fonksiyonu kullanarak, ağdaki birinci dereceden ve ikinci dereceden yakınlıkları koruyan düğüm yerleştirmeleri oluşturur. İlk bölüm olan $L1$, bağlantılı düğümler arasındaki nokta çarpımının tersini minimize eder. İkinci dereceden yakınlığı korumak için tasarlanan $L2$ bölümünde ise, çok sayıda ortak bağlantıya sahip düğümlerin benzer olduğunu varsayar. LINE, $L1$ ve $L2$ 'yi ayrı ayrı minimize eden iki model eğitilir ve bir düğümün yerleştirme

vektörü, bu iki modelden elde edilen yerleřtirmelerin birleřtirilmesiyle oluřturulur(Tang vd. 2015b).

Rasgele yürüyüş temelli tekniğin aksine, bazı arařtırmalar (Ribeiro vd. 2017, Ahmed vd. 2020) düğüm rollerini muhafaza etmek amacıyla "rol tabanlı" terimini kullanmıřtır; zira rasgele yürüyüş temelli örnekleme stratejileri bu rolleri sabit bir uzunluk dahilinde yakalayamamaktadır. Düğüm rollerini koruyarak, rol tabanlı modeller yapısal eřdeğerliğı yakalama kabiliyetine sahip olmuřtur. Örneğin, sosyal ağırlar bağlamında, bu yaklaşım, toplulukların çeřitli faaliyetlerini etkileme potansiyeline sahip etkileyicilerin tespit edilmesini kolaylařtırmıřtır. Ribeiro vd. (2017) düğüm rollerine dayalı olarak ağ yapısını yakalamak için struc2vec modelini önermiřtir. Bu modele göre, aynı dereceye sahip düğümler, vektör uzayında birbirine yakın kodlanmalıdır. Önerilen yöntem, düğümler arasındaki yapısal benzerliğı ölçmek için hiyerarřık bir yaklaşım benimsemiř, düğümlerin k-atlama komřuluklarını ve derece dizilerini dikkate almıřtır. Bu yapısal benzerlikleri kodlamak amacıyla çok katmanlı bir ağ oluřturularak her düğüm için bağlamlar üretmek üzere bu ağ üzerinde yönlendirilmiř rasgele yürüyüş gerçekleřtirilmiřtir. Elde edilen düğüm dizileri, ardından Skip-gram kullanılarak gizli temsilleri öğrenmek için kullanılmıřtır. Gerçek ağırlar üzerinde yürütölen deneyler, struc2vec'in yapısal kimliğı etkili bir řekilde yakaladığıını göstermiř ve DeepWalk ile node2vec gibi mevcut yöntemleri performans açısından geride bıraktığıını ortaya koymuřtur. Ahmed vd. (2020) rasgele yürüyüş yollarını yönlendirmek için bir düğüm-tipi parametresi ekleyerek düğüm benzerliğini ve yapısını yakalayan Role2Vec modelini geliřtirmiřtir. Role2Vec'in temel prensibi, aynı küme içindeki düğümlerin rasgele yürüyüş yolunda birlikte örneklenmesi gerektiğidir. Yalnızca aynı kümeler içindeki düğümleri örnekleyerek, Role2Vec daha az hesaplama karmařıklığı ile doğıru örüntöleri öğrenebilmiřtir. Akabinde model, düğüm yerleřtirmelerini öğrenmek için Skip-gram yaklaşımını kullanmıřtır.

Yukarıda bahsedilen modellerin çoğı, yan bilgiler olmaksızın düğüm yerleřtirmelerini öğrenmeyi amaçlamaktadır; halbuki bu yan bilgiler ağ yapısının öğrenilmesinde bilgilendirici olabilmektedir. Bununla birlikte, gerçeık dünyadaki ağırlar, düğümlerin ve kenarların öznitelikleri gibi yan bilgilerden oluřabilmektedir. Örneğin, sosyal ağırlardaki

kullanıcı profilleri ve etkileşimleri gibi ek bilgiler, düğüm yerleştirmelerinin daha anlamlı olmasına katkıda bulunabilir. Bazı çalışmalar, öznitelikli ağlar olarak sunulan düğüm özelliklerini ekleyerek, öznitelikli ağlarda düğüm yerleştirmelerini öğrenmeye çalışmıştır (Sheikh vd. 2019, Dou vd. 2020).

Sheikh vd. (2019) düğümlerin bağlamsal özniteliklerini yakalamak için gat2vec modelini önermişlerdir. Model, her iki ağ (yönlü ve yönsüz) türünde de ağ yapısını yakalamak amacıyla rasgele yürüyüş örnekleme stratejisinden faydalanmıştır. Yapısal bağlamları oluşturmak için yapısal bilgileri ve öznitelik bağlamlarını oluşturmak için nitelikleri kullanmış, ardından bu bağlamları birlikte ele alarak sık bir sinir ağı modeli aracılığıyla temsil öğrenmeyi gerçekleştirmiştir. Gat2vec gerçek dünya veri kümeleri üzerinde düğüm sınıflandırma, bağlantı tahmini ve görselleştirme görevlerinde değerlendirilmiştir. Deneyler, gat2vec'in çoklu bilgi kaynaklarını etkili bir şekilde kullandığını ve tüm görevlerde en güncel temel yöntemleri performans açısından geride bıraktığını göstermiştir .

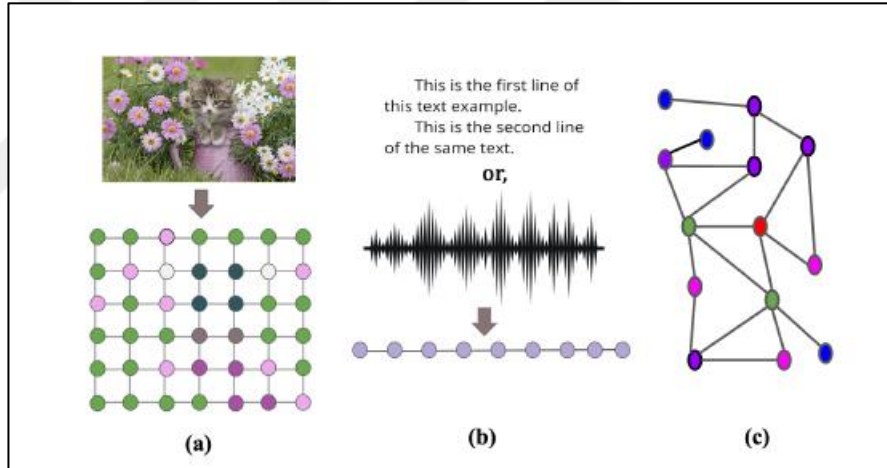
Dou vd. (2020) düğümlerin hem yapısal hem de öznitelik bilgilerini birleştiren bir ağ temsil öğrenme yöntemi olan ANRLBRW'yi önermiştir. Önerilen yöntem, ağ temsili geliştirmek amacıyla yanlı rasgele yürüyüşleri öznitelik bilgileriyle birleştirmektedir. ANRLBRW, ağ yapısının yerel ve küresel özelliklerini yakalamak için yanlı rasgele yürüyüş ve düğüm öznitelik bilgilerini hesaba katmak için öznitelik rasgele yürüyüşü kullanmaktadır. Üç gerçek öznitelikli ağ üzerinde yapılan deneyler, ANRLBRW'nin hem yerel hem de küresel düğüm özelliklerini etkili bir şekilde yakaladığını ve düğüm sınıflandırma görevlerinde en gelişmiş yöntemlere kıyasla üstün doğruluk elde ettiğini göstermiştir.

2.3 Derin Sinir Ağı Modelleri

Derin öğrenme çeşitli araştırma alanlarında başarılı uygulamalarıyla ortaya çıkan son derece etkili bir çerçeve haline gelmiştir. Bu derin öğrenme yöntemleri arasında, evrişimli sinir ağları (Convolutional Neural Networks - CNN) (LeCun vd. 1998), görüntü sınıflandırma (Krizhevsky vd. 2012), anlamsal segmentasyon (Chen 2014) ve nesne

tespiti (Ren vd. 2016) gibi birçok görüntü ile ilgili uygulamada umut verici performans sergilemiştir. Görüntülerin yanı sıra, evrişimli sinir ağları, sinirsel makine çevirisi gibi doğal dil işleme görevlerinde de başarıyla uygulanmıştır (Cho vd. 2014, Amodei vd. 2016, Gehring vd. 2017). Evrişimsel işleç konsepti, derin öğrenmeyi daha esnek hale getirerek, yapıli girdilerden özellikleri otomatik olarak çıkarabilmesini sağlar, zahmetli elle tasarlanmış özellik çıkarımını azaltır ve sınıflandırma sürecini otomatikleştirir.

Derin evrişimli sinir ağlarında, genellikle bir filtre (görüntünün boyutundan daha küçük iki boyutlu bir matris) her evrişim katmanında görüntü üzerinde kayarak matris çarpımı yapar ve bilgi havuzlama yoluyla özellikleri çıkarır. Bu evrişimsel işleç, matris çarpımı ve havuzlama yoluyla mekansal özellikleri çıkarabildiği için düzenli yapıli verilerde etkili olmuştur.



Şekil 2.2 (a) Bir görüntü ızgara grafiğine dönüştürülebilir (b) Metin veya konuşma sinyali çizgi grafiğine dönüştürülebilir (c) Farklı düğümlerin farklı derecelere sahip olduğu düzensiz bir ağ örneği

Örneğin, **Şekil 2.2.a**'da gösterilen bir görüntü, bir ızgara grafiği olarak temsil edilebilir; daha önce belirtilen bir filtre bu görüntü üzerinde soldan sağa ve yukarıdan aşağıya kayarak mekansal bilgileri yakalar. Benzer şekilde, metinler veya konuşma sinyalleri, **Şekil 2.2.b**'de gösterildiği gibi, bir çizgi grafiği olarak temsil edilebilir ve bu tür veriler için bir evrişimsel işleç tasarlamak oldukça basittir. Ancak, ağlar için evrişimsel bir işleç tasarlamak, son derece düzensiz yapıları nedeniyle önemli bir zorluk teşkil etmektedir.

Ağlar üzerinde düzenli evrimsel işlemler kullanmak iki ana zorlukla karşılaşmaktadır. Bu zorluklar, düzenli evrimlerin her düğüm için komşu düğümlerin sayısının aynı kalmasını ve bu komşu düğümlerin sıralı olmasını gerektirmesinden kaynaklanmaktadır. Ancak bir ağ içindeki farklı düğümler için komşu düğümlerin sayısı genellikle farklılık gösterir ve bir düğümün komşu düğümleri arasında, çıktının belirleyici olmasını sağlamak için sıralayabileceğimiz bir derecelendirme bilgisi bulunmamaktadır. **Şekil 2.2.c'**de, farklı düğümlerin farklı derecelere sahip olduğu bir ağ gösterilmektedir.

Derin öğrenmedeki ilerlemeler, evrimsel işlemleri ve derin sinir ağlarının ağ verilerine uygulanmasına yönelik yeni bir çalışma alanı doğurmuştur. Ağ yapıları üzerinde uygulanan evrimli ağlar, günümüzde yaygın olarak Çizge Sinir Ağları (Graph Neural Networks - GNN) olarak bilinmektedir (Henaff vd. 2015). Ağ evrim işlemlerinin tanımlanmasına dair özgün fikir, ağ sinyal işleme alanından doğmuştur. Bu kavram, sonradan ağ Fourier dönüşümünde ağ Laplasyanı formunda ortaya çıkan filtre parametrelerinin öğrenilmesi problemi olarak tanımlanmıştır (Bruna vd. 2013).

Önceki rasgele yürüyüş ve matris çarpanlara ayırma modellerden farklı olarak, derin sinir ağı tabanlı modellerin çoğu, düğüm yerleştirme işlemlerini öğrenmek amacıyla ağ yapısını ve düğüm özniteliklerini/özelliklerini benimsemiştir. Örneğin, bir sosyal ağdaki kullanıcılar, profil bilgileri gibi metin verilerine sahip olabilir. Düğüm özniteliklerinin bulunmadığı ağlarda, düğüm derecesi veya konumu gibi ağ istatistikleri, düğüm öznitelikleri olarak kullanılabilir.

SDNE (Wang vd. 2016) ve DNGR (Cao vd. 2016) adlı yöntemler, ağ yerleştirme işlemlerini oluştururken ağlardaki doğrusal olmayan yapıyı yakalamak ve eş zamanlı olarak boyut indirgeme uygulamak amacıyla derin otokodlayıcıları kullanmışlardır. SDNE, birinci dereceden yakınlığı koruyan bir otokodlayıcı kullanmış ve benzer düğümlerin yerleştirme vektörleri arasındaki uzun mesafeleri cezalandırmak için Laplace Özhartılarından yararlanmıştır (Wang vd. 2016). Ancak, SDNE modeli başlangıçta türdeş ağlarda düğüm yerleştirmelerini öğrenmek için önerilmiştir. SDNE modelinin türdeş olmayan ağlara uyarlanması, çeşitli ağ otokodlayıcı modelleri tarafından önerilmiştir (Cao vd. 2016, Tu vd. 2018). DNGR ise, rasgele gezinmeye dayalı benzerlik

bilgisinden türetilen pozitif noktasal karşılıklı bilgi matrisi (positive pointwise mutual information matrix) üzerinde yığılmış gürültü giderici otokodlayıcılar kullanmıştır (Cao vd. 2016). Her iki yöntem de küresel bilgilerden faydalanmış olup, bu özellik onları büyük ölçekli ağlar için uygunsuz kılmaktadır.

Kipf ve Welling (2016), Çizge Evrişimli Ağ (Graph Convolutional Network - GCN) kavramını ilk kez ortaya atan araştırmacılar olarak literatürde önemli bir yer edinmiştir. Bu öncü çalışma, ağ yapılarını etkili bir şekilde işleyebilen derin öğrenme modellerinin geliştirilmesinde dönüm noktası niteliğindedir ve sonraki birçok araştırmaya ilham kaynağı olmuştur. Önerilen yaklaşımda, her düğüm için tüm komşu düğümlerden özellikleri toplamak amacıyla evrişim benzeri bir işlem kullanmayı ve ardından verilen düğüm için yeni bir özellik temsili oluşturmak üzere doğrusal bir dönüşüm uygulamayı önermişlerdir. Spesifik olarak, merkezi düğümün kendisi de dahil olmak üzere, komşuluktaki tüm özellik vektörleri toplanır ve komşu sayısına bağlı olarak eğitilemeyen ağırlıklarla çarpılır. Bu işlem evrişim benzeri olarak kabul edilebilir, ancak iki temel açıdan düzenli evrişimsel işlemlerden farklılık göstermektedir. İlk olarak, her düğümü taramak için aynı yerel filtreyi kullanmaz; farklı sayıda komşu düğüme sahip olan düğümler, değişken boyutlarda ve ağırlıklarda filtrelere sahiptir. İkinci olarak, filtrelerdeki ağırlıklar, alıcı alandaki tüm komşu düğümler için özdeştir ve komşu sayısı tarafından belirlenir (Kipf ve Welling 2016).

GCN modelinin öne sürülmesini takiben, bu çerçeve üzerine inşa edilen çok sayıda GNN modeli geliştirilmiş olup, çeşitli iyileştirmeler bünyelerine dahil edilmiştir. Bu geliştirmeler arasında örnekleme stratejilerinin entegrasyonu (Hamilton vd. 2017a, Chen vd. 2018a, Huang vd. 2018), dikkat mekanizmalarının ilavesi (Velickovic vd. 2017, Thekumparampil vd. 2018) ve filtre çekirdeğine yönelik optimizasyonlar (Nt ve Maehara 2019, Wu vd. 2019) gibi yenilikçi yaklaşımlar yer almaktadır.

GraphSAGE'in yazarları (Hamilton vd. 2017a), tümevarımsal gözetimsiz temsil öğrenimi için GCN'yi geliştirerek, komşuluk analizi için GCN'de kullanılan basit evrişimler yerine eğitilebilir toplama fonksiyonlarını dahil etmişlerdir. GraphSAGE yönteminde üç farklı toplama fonksiyonu kullanılmıştır: (1) Bir düğümün tüm komşularından gelen katkıların

aritmetik ortalamasını hesaplayan Ortalama toplayıcı (Mean aggregator), (2) komşulardan gelen maksimum katkıyı seçen Maksimum havuzlama toplayıcısı (Max-pool aggregator) ve (3) genellikle sıralı veya zaman serisi veriler için kullanılan klasik Tekrarlayan Sinir Ağı'nın (Recurrent Neural Network - RNN) bir varyantı olan Uzun Kısa Süreli Bellek (Long Short-Term Memory - LSTM). GraphSAGE'de bu yöntemler, hem dönüştürücü (transductive) hem de tümevarımsal (inductive) ortamlarda önceki tekniklere kıyasla üstün performans sergilemiştir.

PinSage (Ying vd. 2018), rasgele yürüyüşlere dayalı önem örneklemesini dahil ederek GraphSAGE algoritmasını genişletmiştir. Önem puanı, ziyaret sayıları olarak hesaplanmış olup, bu durum hem ölçeklenebilirliği hem de düğüm yerleştirme kalitesini artırmıştır. Liu vd. (2019b) çalışmalarında, BFS ve DFS örneklemesi üzerinden yerel komşuluk bilgisinin toplanmasını düzenleyen belirli katmanları tespit ederek node2vec (Grover ve Leskovec 2016) modelini modifiye etmişlerdir. Bu sayede node2vec yöntemini çizge sinir ağlarına genişletmişlerdir (Liu vd. 2019b).

Velickovic vd. (2017), düğüm sınıflandırması için tasarlanmış dikkat tabanlı bir mimari olan Çizge Dikkat Ağlarını (Graph Attention Networks - GATs) önermişlerdir. Temel konsept, öz-dikkat stratejisi kullanarak her düğümün komşularına odaklanarak gizli temsillerini hesaplamayı içermektedir. GAT, diğer GCN'lerin sınırlamalarını ele almak için dikkat kavramını kullanan bir GCN türü olmuştur. Bu yaklaşım, düğümlerin komşuluk özelliklerini birleştirirken her bir komşuya farklı ağırlıklar atamalarına olanak tanımaktadır (Velickovic vd. 2017).

GAT (Velickovic vd. 2017) modelini geliştirerek dikkat mekanizması aracılığıyla daha kapsamlı yapısal bilgileri yakalamayı amaçlayan Haonan vd. (2019), GraphStar modelini ortaya koymuşlardır. Bu model, her gizli katmanda küresel bilgileri muhafaza etmek amacıyla sanal bir düğüm (sanal bir başlangıç noktası) kullanmaktadır. GAT'lardan farklı olarak GraphStar modeli, üç ayrı ilişki türünü bünyesinde barındırmaktadır: düğümden düğüme (öz-dikkat), düğümden başlangıca (küresel dikkat) ve düğümden komşulara (yerel dikkat). Bu farklı ilişki türlerinden yararlanarak GraphStar, ilave sinir ağı

katmanları eklenirken karşılaşılan aşırı düzleştirme problemine çözüm getirmektedir (Haonan vd. 2019).

GAT (Velickovic vd. 2017) modelinin karşılaştığı sorunlardan biri, ağırlıklı olarak birkaç komşu düğüme yüksek ağırlıklı dikkat yoğunlaştıran statik dikkat mekanizması sunmasıdır. Bunun sonucunda GAT, bir ağdaki tüm düğümlere uygulanabilir evrensel bir dikkat mekanizması öğrenmekte zorlanmaktadır. Bu sınırlamaları aşmak amacıyla Brody vd. (2021), GATv2 modelini önermişlerdir. Bu model, hedef düğümden komşu düğümlere olan ağ yapısını daha etkin bir şekilde öğrenmek için dinamik dikkat mekanizmasını kullanmaktadır (Brody vd. 2021).

Yukarıda Bahsi geçen GNN yöntemleri, oldukça küçük ağlar üzerinde dahi yavaş çalışmakta ve kayda değer miktarda bellek tüketmektedir. Bu sebeple, araştırmacılar büyük ölçekli ağlar üzerinde etkin bir şekilde işlem yapabilmek adına eğitim süresini iyileştirmeye yönelik çaba sarf etmişlerdir. Fast-GCN (Chen vd. 2018a), doğruluk bakımından performanstan ödün vermesine rağmen, önceki yöntemlere kıyasla kayda değer ölçüde daha hızlı çalışan bir yöntem olmuştur. Fast-GCN, GCN'nin tüm komşu düğümleri işleme yerine, işlenmesi gereken komşu düğümleri belirlemek için örnekleme yöntemini kullanmaktadır. Bu yaklaşım, daha büyük ve yoğun ağlar üzerindeki hesaplama hızını önemli ölçüde artırmıştır.

GNN yöntemlerinin sınırlılıklarından biri, gizli katmanda modelin tüm komşu düğümlerin durumunu güncellemesidir. Bu durum, etkin olmayan düğümler sebebiyle yavaş eğitim ve güncellemeye yol açabilmektedir. Bazı modeller, örnekleme stratejisini geliştirerek GNN modellerini iyileştirmeyi hedeflemişlerdir (Chang vd. 2018, Chen vd. 2018a, Chen vd. 2018b). Örneğin, Chen vd. (2018b), düğümlerin durum güncellemesini iyileştirmek amacıyla yenilikçi bir model geliştirmişlerdir. Bu model, her bir düğümün komşularından tüm bilgileri toplamak yerine, alıcı alanını sınırlandırmak maksadıyla düğümlerin aktivasyon geçmişi durumlarını takip etme seçeneğini öne sürmektedir (Chen vd. 2018b).

GNN yöntemlerinin türdeş ağlara uygulanmasına ek olarak, çeşitli araştırmalar GNNs'lerin türdeş olmayan ve bilgi ağları üzerindeki kullanımına odaklanmıştır (Schlichtkrull vd. 2018, Park vd. 2020). Türdeş olmayan farklı türde kenar ve düğümler içerdiğinden, bu çeşitli kenar türlerine dayalı mesajların toplanması önemli bir zorluk teşkil etmektedir. Schlichtkrull vd. (2018), bilgi ağlarındaki ilişkisel varlıkları modellemek amacıyla İlişkisel Çizge Evrişimli Ağlar (Relational Graph Convolutional Networks - R-GCNs) modelini ortaya koymuşlardır. R-GCNs, bağlantı tahmini ve düğüm sınıflandırması gibi çeşitli ağ analiz görevleri için türdeş olmayan ağlarda düğüm yerleştirme öğrenmeye yönelik tasarlanan ilk model olmuştur. Buna ilaveten, model düğüm yerleştirme öğreniminin verimliliğini artırmak için parametre paylaşımından yararlanmaktadır.

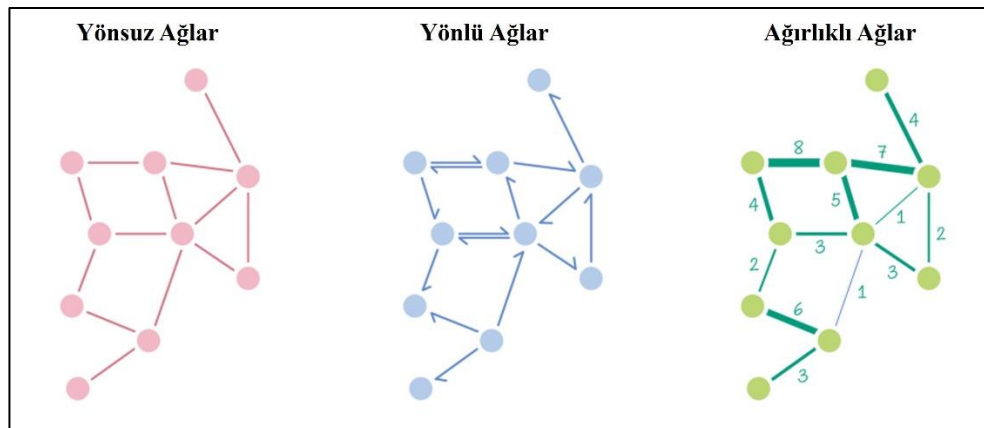
Derin sinir ağı modelleri, ağ yapısındaki verileri işleme ve analiz etme kabiliyetleriyle yaygın bir kullanım alanı bulmuştur. Ancak, bu modeller birtakım sınırlamalar ve dezavantajlar barındırmaktadır (Khoshraftar ve An 2024). Ölçeklenebilirlik açısından, GNNs'lar genellikle büyük ölçekli ağlarla başa çıkmada zorlanmakta, bu durum hesaplama ve bellek gereksinimleri bakımından aşırı maliyetli olabilmektedir (Hoang vd. 2023). Hiper parametre hassasiyeti bağlamında, GNNs'lar öğrenme oranı, katman sayısı ve düğüm yerleştirme boyutları gibi faktörlere karşı duyarlılık göstermekte, optimal hiper parametre setinin belirlenmesi zorlu ve hesaplama açısından yoğun bir süreç gerektirmektedir. Düğüm özniteliklerinin yokluğu, GNN modellerinin performansını ve etkinliğini önemli ölçüde etkileyebilmektedir. Özellikle komşuluk bilgilerini toplama veya dikkat mekanizmaları kullanan modellerde, düğüm öznitelikleri olmadığında, toplama süreci daha az bilgiyle çalışmak durumunda kalmakta, bu da potansiyel olarak daha az bilgilendirici düğüm yerleştirmeleriyle sonuçlanabilmektedir. Ayrıca, GAT gibi dikkat mekanizmalarını kullanan GNN'larda, düğüm özniteliklerinin yokluğu, dikkat mekanizmasının düğümleri bireysel özelliklerine göre ayırt etme yeteneğini kaybetmesine ve dolayısıyla etkinliğinin azalmasına neden olabilmektedir (Hoang vd. 2023, Khoshraftar ve An 2024).

3. AĞLAR

3.1 Ağlara Genel Bakış

Ağlar, sosyal bilimler, biyoloji ve mühendislik gibi çeşitli alanlardaki karmaşık ilişkileri ve etkileşimleri temsil etmek için temel yapılar olarak hizmet eder. Temelinde, bir ağ, farklı varlıklar arasındaki bağlantıları kapsayan düğümler (veya köşeler) ve kenarlardan (veya bağlantılardan) oluşur. Bu organizasyon, verilerin geleneksel yöntemlerle gerçekleştirilemeyen şekillerde görselleştirilmesine ve analiz edilmesine olanak tanır (Nettleton 2013). Ağlar, ham veri kümelerinde gizlenmiş olabilecek örüntülerin, davranışların ve eğilimlerin keşfedilmesine olanak tanır. Düğümler ve kenarları kullanarak ağlar, varlıklar arasındaki etkileşimleri tasvir eder; böylece problem çözme ve karar alma süreçleri için hayati önem taşıyan görsel ve analitik bir çerçeve sunar. Ağların önemi, gerçek dünya olgularını etkili bir şekilde modelleme yeteneklerinde yatar; örneğin, bağlantı tahmini, düğüm sınıflandırma ve topluluk tespiti görevleri için makine öğreniminde yaygın olarak kullanılmaktadırlar (Sundaram 2023).

Bir ağ G , (V, E) ikilisi olarak tanımlanır; burada V düğümler kümesini, E ise düğüm çiftlerinden oluşan bağlantılar kümesini temsil eder. $V = \{v_1, v_2, v_3, \dots, v_n\}$, $E = \{(v_1, v_2), (v_1, v_3)\}$.



Şekil 3.1 Farklı ağ türlerinin temsilleri

Ağların çeşitli biçimlerini kavramak, farklı alanlardaki çeşitli uygulamalarını tanımak için hayati öneme sahiptir. Ağlar, yapısal özelliklerine ve kenar kümelerine göre sınıflandırılabilir; örneğin, yönlü, yönsüz, ağırlıklı, ağırlıksız ve öznitelikli ağlar (**Şekil 3.1**). Eğer (E) sıralı çiftlerden $(e_{i,j} \neq e_{j,i})$ oluşuyorsa, ağ yönlüdür; (E) sırasız çiftlerden oluşuyorsa, ağ yönsüzdür. Örneğin, Twitter yönlü bir ağı temsil ederken, Facebook yönsüz bir ağı örneklendirir. Ayrıca, her bir $(e_{i,j})$ kenarına bir $(\omega_{i,j})$ ağırlığı atanmışsa, ağ ağırlıklıdır; kenarlar ağırlık taşımıyorsa, ağ ağırlıksızdır. Ağırlıklı ağlar $G = (V, E, \mathcal{W})$ şeklinde gösterilir; burada $(\mathcal{W})$ kenar ağırlıkları kümesidir. Öznitelikli ağlar ise, düğümler ve/veya kenarlar üzerinde ek öznitelikler veya etiketler barındırır; örneğin, düğümlerin proteinleri temsil ettiği ve özniteliklerin türleri veya işlevleri belirttiği biyolojik ağlar. Bu ağlar $G = (V, E, At)$ biçiminde ifade edilir; burada (At) öznitelikler kümesini temsil eder.

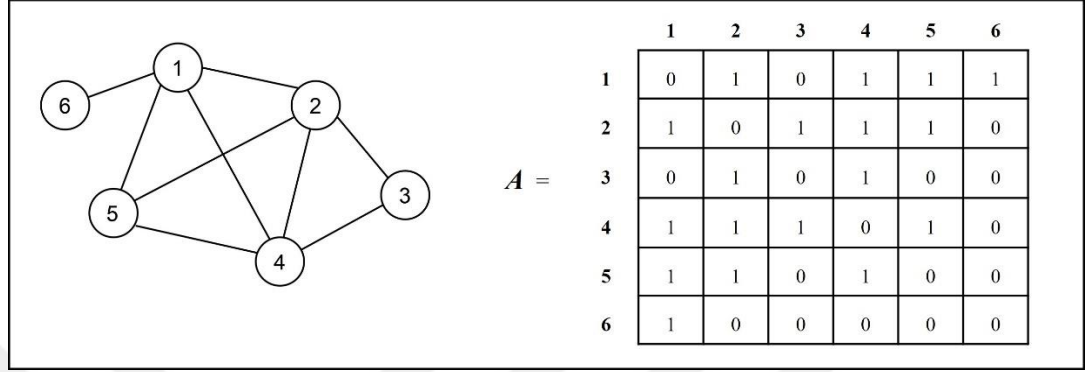
3.2 Ağların Matris Temsili

Bir ağın matris temsili, ağın yapısını ve bağlantılarını bir matris kullanarak göstermenin bir yöntemidir. Yaygın olarak kullanılan bir matris temsili, boyutu $|V| \times |V|$ olan komşuluk matrisi A 'dır; burada $|V|$, ağdaki düğüm sayısını belirtir. Bu matriste, hem satırlar hem de sütunlar düğümlere karşılık gelir ve girişler düğümler arasındaki kenarın varlığını veya yokluğunu ifade eder. Özellikle, düğümler v_i ve v_j arasında bir kenar varsa A_{ij} 1 değerini alır; aksi takdirde A_{ij} 0'dır.

$$A_{ij} = \begin{cases} 1, & \text{eğer } e_{ij} \in E \\ 0, & \text{aksi takdirde} \end{cases} \quad (3.1)$$

Komşuluk matrisi, ağ verilerini saklamak ve yönetmek için özlü ve etkili bir yöntem sunarak, gelişmiş matris algoritmalarının ve tekniklerinin ağ analizi ve işlenmesi için uygulanmasını kolaylaştırır. Ayrıca, ağ temsiliinde çok yönlü faydalar sağlamaktadır. Açık yapısı, ağ bağlantısının hem basit bir şekilde uygulanmasına hem de net bir biçimde görselleştirilmesine olanak tanımaktadır. Çok sayıda kenara sahip yoğun ağlar için avantajlı olan bu yöntem, sabit zamanlı $O(1)$ kenar varlığı sorgularını desteklemektedir.

Yönsüz ağlar söz konusu olduğunda, matrisin simetrik özelliği, analitik ve algoritmik kullanışlılığını daha da artırmaktadır. Beş düğüm ve dokuz kenardan oluşan yönsüz örnek bir ağın komşuluk matrisi **Şekil 3.2**'de gösterilmiştir.



Şekil 3.2 Beş düğüm ve dokuz kenardan oluşan yönsüz örnek bir ağın komşuluk matrisi

3.3 Ağ Yapısının Çıkarımı

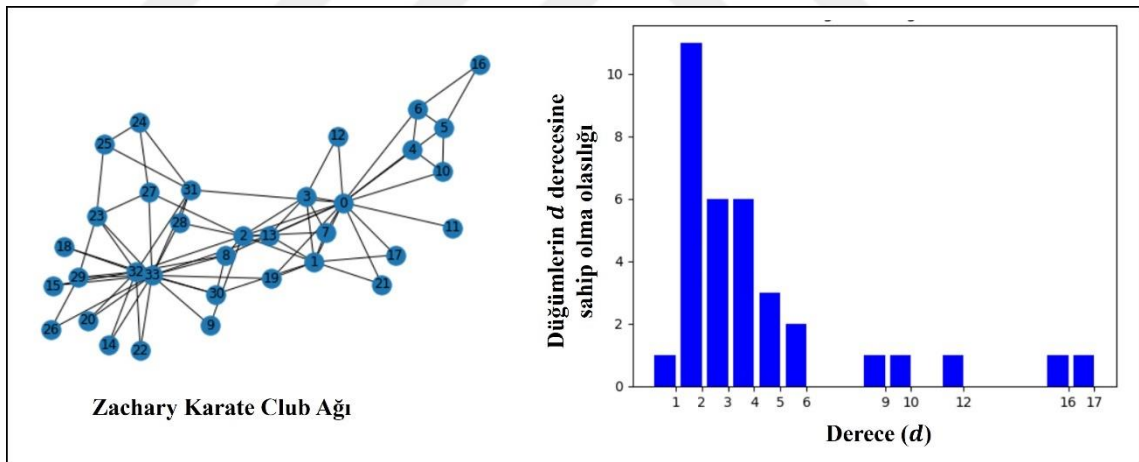
Verilerin ağ biçiminde temsil edilmesinin en belirgin avantajlarından biri, istatistiksel açıdan anlamlı yapısal bilgileri çıkarmaya yönelik çok sayıda metriğin bulunmasıdır. Bu metrikler, ağ içerisindeki çeşitli bağlantı örüntülerini ortaya çıkararak, veriler hakkında benzersiz içgörüler sunma potansiyeline sahiptir (Newman 2003). Çizge teorisinin zengin matematiksel çerçevesinden yararlanarak, sosyal ağlardan biyolojik sistemlere, iletişim ağlarından daha geniş alanlara kadar uzanan çok yönlü veri kümelerini analiz etmek ve yorumlamak mümkün kılmaktadır. Bu çok yönlülük, karmaşık veri kümelerinden anlamlı bilgiler çıkarmada ağ temsillerinin değerini vurgulamakta, hem teorik araştırmalarda hem de pratik uygulamalarda ilerlemeleri kolaylaştırmaktadır (Nettleton 2013).

3.3.1 Derece ve derece dağılımı

Bir ağda bir düğümün derecesi, ona bağlı kenarların sayısını temsil eder ve düğümün bağlantı yapısının temel bir ölçüsü olarak hizmet eder (Newman 2003). Genellikle i düğümünün derecesi d_i ile gösterilir ve $|V|$ boyutundaki basit bir ağ için derece, komşuluk matrisi $A_{i,j}$ cinsinden şu şekilde hesaplanır:

$$d_i = \sum_{j=1}^{|V|} A_{i,j} \quad (3.2)$$

Derece dağılımı ise rasgele seçilen bir düğümün belirli bir derece değerine sahip olma olasılığını hesaplamak için kullanılır. Bu dağılım, ağdaki bağlantıların doğal bir genel görünümünü sunar ve genellikle bir birimlik kutu boyutuna sahip bir histogram olarak gösterilir (**Şekil 3.3**). Derece dağılımı, ağ analizinin temel bir unsuru olup, ağın yapısal özellikleri ve bağlantı desenleri hakkında değerli içgörüler sunar. Örneğin, sosyal ağlarda, oldukça eğimli bir derece dağılımı, az sayıdaki yüksek bağlantılı düğümlerin (merkezlerin) yanı sıra, daha az bağlantıya sahip birçok düğümün varlığını ortaya koyar. Derece dağılımını analiz etmek, ağın rasgele arızalara ve hedefli saldırılara karşı dayanıklılığını anlamak ve bilgi yayılımı veya hastalık bulaşması gibi olguları modellemek için kritik öneme sahiptir (Meidanis 2021). Bu nedenle, derece ve derece dağılımının doğru analizi, hem teorik hem de pratik çizge teorisi ve ağ bilimi araştırmaları için vazgeçilmez bir gerekliliktir.



Şekil 3.3 Zachary Karate Club ağındaki düğüm derecelerinin dağılımı

3.3.2 Yürüyüşler ve rasgele yürüyüşler

Yürüyüş, her düğümün bir öncekine bir kenarla bağlandığı sonlu bir düğüm dizisi olarak tanımlanmıştır (Newman 2003). Çizge teorisinde yürüyüşler, düğümleri birden fazla kez ziyaret edebilen basit yürüyüşler ve başlangıç ile bitiş düğümlerinin aynı olduğu kapalı yürüyüşler gibi çeşitli kategorilere ayrılmaktadır (Backstrom ve Leskovec 2011). Bu

yürüyüşler, çizge teorisi içerisindeki çeşitli analitik görevlerde büyük önem taşımaktadır. Bunlar arasında, iki düğüm arasındaki bir güzergâhı belirlemeyi içeren yol bulma; düğümlerin erişilebilirliğini ve farklı yürüyüşler içerisindeki konumlarına dayalı olarak önemli düğümleri tespit eden merkezîyet ölçütleri gibi uygulamalar yer almaktadır (Umbrey ve Rahman 2023).

Rasgele yürüyüşler ise, özel bir yürüyüş türü olup ağ dolaşım sürecine olasılıksal bir bileşen katmıştır (Umbrey ve Rahman 2023). Belirli bir düğümden rasgele yürüyüş gerçekleştirmek için, başlangıç düğümlerine bir kenarla bağlı olan komşu düğümlerden biri rasgele seçilir. Bu yeni düğümden, komşu bir düğüm yine rasgele seçilir ve bu işlem, istenilen yürüyüş uzunluğuna ulaşılan kadar tekrarlanır. Standart rasgele yürüyüş sürecinde, geçiş olasılığı doğrudan kenar ağırlığı $A_{i,j}$ ile orantılıdır.

$$p(x(t+1) = j | x(t) = i) = \frac{A_{i,j}}{d_i} \quad (3.3)$$

Burada $x(t) \in V$, t anındaki yürüyüşçünün konumunu temsil etmektedir. Ağırlıksız ağlar söz konusu olduğunda ise $A_{i,j} = 1$ olarak kabul edilmektedir. Standart rasgele yürüyüşler, yönsüz bağlantılı ağlarda düğüm komşuluğunu doğal bir şekilde yakalamak için elverişli bir yöntem sunmaktadır. Ayrıca, komşuluk kavramının farklı boyutlarını keşfetmek amacıyla yanlı rasgele yürüyüşler de tasarlanabilmektedir (Grover ve Leskovec 2016).

3.4 Ölçekten Bağımsız Ağlar

Ağların oluşumunu incelemek için en yaygın kullanılan modellerden biri Barabási-Albert (BA) modelidir (Barabási ve Albert 1999). Ölçekten bağımsız ağlar, derece dağılımının güç yasasına uyduğu, yani az sayıda düğümün çok yüksek derecelere sahip olduğu, çoğunluğun ise nispeten düşük derecelere sahip olduğu bir ağ sınıfıdır (Wang ve Lu 2023). Bu ağlar, doğal bir ölçekten yoksun olmaları nedeniyle "ölçeksiz" ağlar olarak adlandırılmıştır (Dorogovtsev ve Mendes 2003). Ölçekten bağımsız ağlar, yüksek dereceye sahip düğümlerin yeni bağlantıları daha fazla çekme eğiliminde olduğu tercihli

bağlanma süreçlerinden ortaya çıkmıştır (Jeong vd. 2003). Bu tür ağlar, sosyal ağlar, biyolojik ağlar ve internet gibi çeşitli gerçek dünya sistemlerinde yaygındır ve ağ büyümesi ve dayanıklılığı dinamiklerine dair değerli içgörüler sunmuştur (Dorogovtsev ve Mendes 2003). Bu ağların incelenmesi, bilginin yayılmasını, ağların arızalara karşı dayanıklılığını ve karmaşık sistemlerin altında yatan mekanizmaları anlamak açısından önemli sonuçlar taşımaktadır.

BA modeli, derece dağılımı güç yasasına yaklaşık olan ağlar üretmek üzere tasarlanmıştır ve böylece gerçek dünya verilerini daha yakından taklit etmektedir. Bu model, V adet izole düğümden bir ağ oluşturur ve her yeni düğüm ağa eklendiğinde, mevcut bir düğüm i ile bir kenar oluşturma olasılığı p olasılığına sahiptir:

$$p(i) = \frac{d_i}{\sum_{i \in V} d_j} \quad (3.4)$$

Yeni bağlantıların oluşma olasılığı, düğüm derecesi değeriyle doğru orantılı olduğundan, yoğun bağlantılı düğümler (merkezler) ortaya çıkacaktır.

3.5 Ağların Analizi

Ağ analizi, çeşitli ağ tabanlı verilerde mevcut olan karmaşık ilişkileri ve örüntüleri anlamak ve bunlardan yararlanmak için güçlü bir araçtır. Bu disiplin, sosyal ağlar, biyolojik ağlar ve ulaşım ağları gibi çeşitli alanlara uygulanabilen geniş bir yelpazede teknik ve metodolojileri kapsamaktadır (Tabassum vd. 2018). Gerçek dünya ağlarının analizine yönelik tarihsel süreci incelediğimizde, İsviçreli ünlü matematikçi Leonhard Euler'in 1736 yılında çizge teorisi kullanarak çözdüğü Doğu Prusya'nın Königsberg şehrindeki köprü problemi, bu alandaki bilinen ilk çalışma olarak öne çıkmaktadır (Euler 1956). Ağ analizindeki temel tekniklerden biri, ağ yapısının ve topolojisinin analizidir. Bu analiz, ağ içerisindeki önemli düğümleri, kenarları ve toplulukları tespit etmeyi, ayrıca ağın zaman içerisindeki dinamiklerini ve evrimini kavramayı kapsar. Örneğin, Nettleton (2013) çalışmasında, sosyal ağlardaki bireylerin genellikle birden fazla ağ ortağı aradığını ve bunun neticesinde karmaşık ağ yapılarının oluşabileceğini ortaya koymuştur.

Ağ analizinin bir diğer önemli yönü, ağ davranışını öngörmek ve analiz etmek için modeller ve algoritmalar geliştirmektir. Bu, etkili düğümleri belirleme, bilgi veya hastalık yayılımını tahmin etme ve ağ içinde yeni bağlantılar veya ilişkiler önerme tekniklerini içermektedir (Tabassum vd. 2018). Ayrıca, makine öğrenimi tekniklerinin ağ analizi ile entegrasyonu, bu analizlerin öngörü gücünü ve değişen verilere uyum sağlama yeteneğini daha da artıran hayati bir unsur olarak ortaya çıkmıştır.

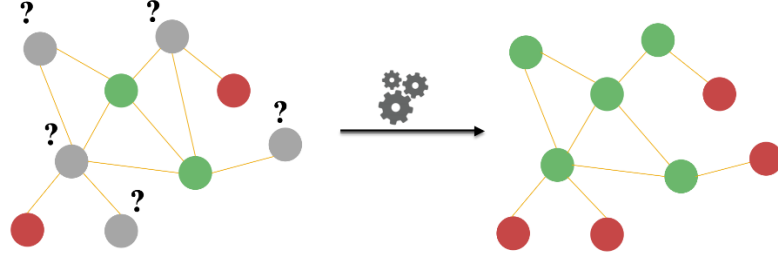
3.6 Düğüm Yerleştirme Yöntemlerinin Uygulamaları

Düğüm yerleştirme yöntemleri, düğümleri sürekli bir vektör uzayında kodlayarak ağ yapısındaki verilere makine öğrenmesi modellerinin uygulanmasını kolaylaştırmaktadır. Vektörel temsillerin hem zaman hem de alan açısından verimli bir şekilde işlenebilmesi sayesinde, geniş bir yelpazedeki ağ analizi uygulamalarına önemli katkılar sağlamaktadır. Bu sayede, geleneksel olarak tablo verilerine uygulanan kümeleme, sınıflandırma ve regresyon gibi algoritmaların kullanımına imkân sağlamaktadır. Düğüm yerleştirme yöntemlerinin temel uygulamaları arasında düğüm sınıflandırma, bağlantı tahmini, topluluk tespiti ve görselleştirme yer almaktadır. Bu uygulamaların her biri, gerçek dünya ağlarına uzanan çeşitli uygulama alanlarında büyük önem taşımaktadır.

3.6.1 Düğüm sınıflandırma

Genellikle ağlardaki düğümler, varlıkların özelliklerini tanımlayan anlamsal etiketlerle ilişkilendirilmektedir. Sosyal ağlarda bu etiketler, ilgi alanlarını, inançları veya demografik özellikleri temsil edebilir. Dil ağlarında bir belge, konular veya anahtar kelimeler ile etiketlenebilirken, biyolojik ağlarda etiketler işlevsel özellikleri yansıtabilir. Birçok ağda, etiketleme maliyetinin yüksek olması, düğümlerin yalnızca kısmen veya seyrek olarak etiketlenmesine yol açmakta ve bu da önemli sayıda düğümün etiketlerinin bilinmemesiyle sonuçlanmaktadır. Düğüm sınıflandırması, kısmen etiketlenmiş ağdan elde edilen bilgiler doğrultusunda, ağdaki etiketlenmemiş düğümlerin etiketlerini tahmin ederek bu sorunu çözmeyi hedeflemektedir (**Şekil 3.4**). Kırmızı ve yeşil renkler, etiketleri önceden bilinen düğümleri temsil ederken, gri renk etiketlenmemiş düğümlere karşılık

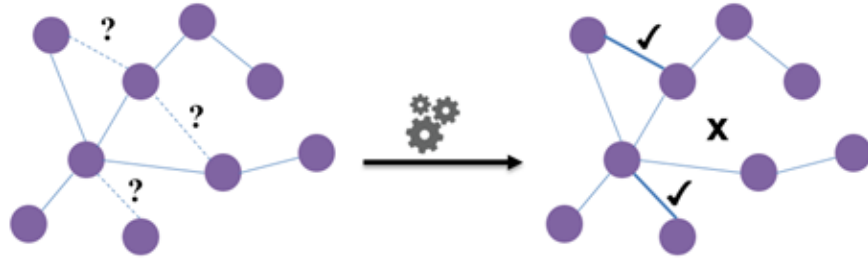
gelmektedir. Temel amaç, renkli düğümlerin taşıdığı bilgilerden yararlanarak her bir gri düğüme uygun bir etiket atamaktır.



Şekil 3.4 Düğüm sınıflandırmasının şematik gösterimi

3.6.2 Bağlantı tahmini

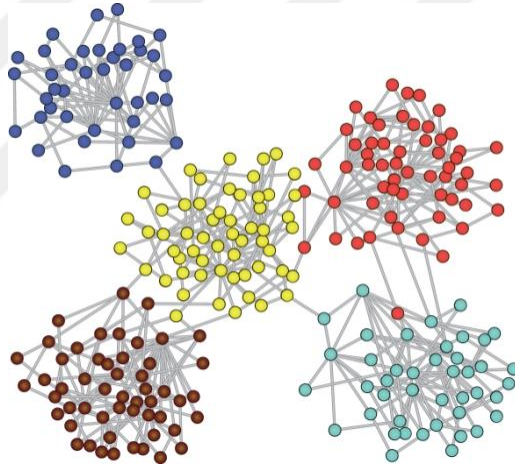
Bağlantı tahmini, ağdaki mevcut bağlantıları ve bunların niteliklerini analiz ederek, eksik veya var olabilecek yeni ilişkilerin (örneğin, dinamik ağlar için gelecekte ortaya çıkabilecek bağlantıların) potansiyel varlığını tahmin etmeyi amaçlar (Backstrom ve Leskovec 2011) (Şekil 3.5). Bu yöntem, sosyal ağlarda bireyler arasındaki bilinmeyen bağlantıları öngörmede geniş çapta kullanılmakta olup, arkadaşlık önerilerinde bulunma veya tavsiye sistemlerinde kullanıcı ürün etkileşimlerinin öngörülmesi uygulamaları kolaylaştırmaktadır.



Şekil 3.5 Eksik bağlantılar için bağlantı tahmini problemi

3.6.3 D ğ m k melemesi

D ğ m k meleme, sosyal ađ analizinde topluluk tespiti olarak da adlandırılan bu y ntem, aynı k me i erisindeki d ğ mlerin, k me dıřındaki d ğ mlere nazaran birbirleriyle daha sıkı bir iliřki i erisinde olduđu bir d ğ m b l mlendirmesini tespit etmeyi ama lar (Malliaros ve Vazirgiannis 2013) (**řekil 3.6**). Bu bađlamda, biyolojik ađlarda benzer niteliklere sahip protein grupları veya sosyal ađlarda ortak ilgi alanlarına sahip bireylerin oluřturduđu belirgin topluluklar gibi  eřitli alanlarda k melerin varlıđı g zlemlenebilmektedir. Bu yaklařım, d ğ mlerin vekt rel temsilleri oluřturulduktan sonra K-ortalamalar veya DBSCAN gibi standart k meleme algoritmalarının uygulanmasını i erir (Cao vd. 2016).

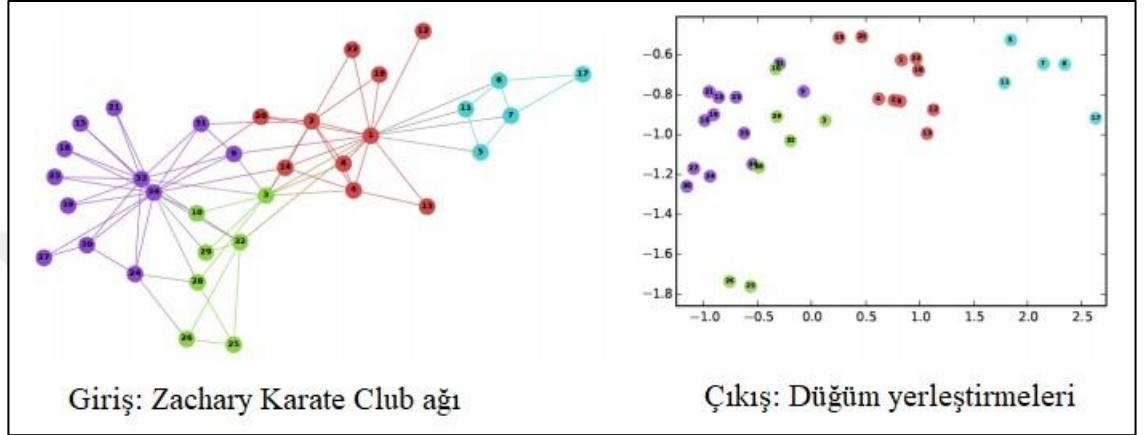


řekil 3.6 Beř topluluđun yer aldıđı  rnek bir ađ

3.6.4 G rselleřtirme

Ađ g rselleřtirme tekniđinin kullanımı,  nl  matematik i Leonhard Euler'in 1736 yılında Dođu Prusya'nın K nigsberg řehrindeki "K nigsberg K pr leri Problemi"ni   zmek i in kullandıđı y nteme dayanmaktadır (Euler 1956). Ađ g rselleřtirme s reci,  ncelikle d ğ mlerin vekt rel temsilleri oluřturularak bařlar. Ardından, her d ğ m temsilini iki boyutlu bir uzaya haritalamak i in t-SNE veya PCA gibi y ntemler kullanılır. Eđer d ğ mlerin  znitelikleri varsa, bu  znitelikler 2 boyutlu g rselleřtirme grafiklerinde d ğ mleri renklendirmek i in kullanılabilir. Bu yaklařım, karmařık ađ yapılarının daha

anlaşılır ve yorumlanabilir hale getirilmesinde önemli bir rol oynamaktadır. Etkili yerleştirme yöntemleri, benzer özelliklere sahip düğümleri yerleştirme alanında birbirine yakın konumlandırılmalıdır. DeepWalk yazarları (Perozzi vd. 2014), yerleştirme tekniklerinin etkinliğini Zachary Karate Club ağına ait bir görselleştirme aracılığıyla en iyi şekilde göstermiştir (Şekil 3.7).



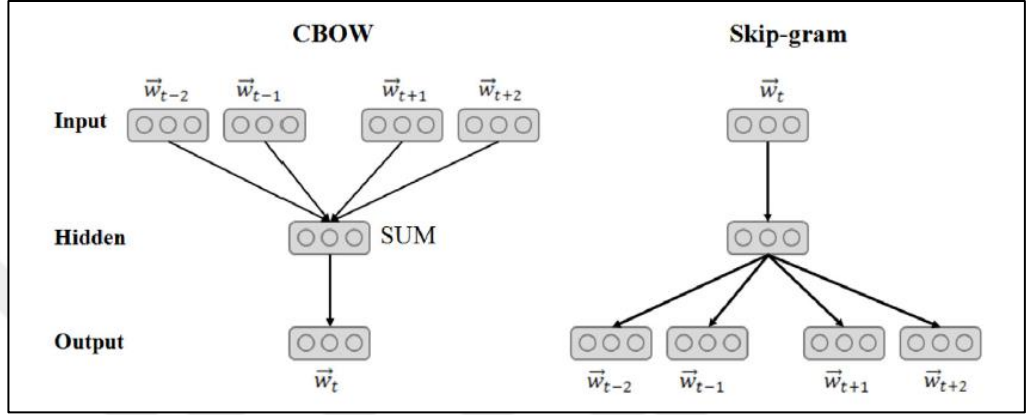
Şekil 3.7 Zachary Karate Club ağına ait 2 boyutlu temsillerin görselleştirilmesi (Perozzi vd. 2014)

3.7 Metin Temsili Öğrenimi

Kelime yerleştirmeleri, ilişkili veya benzer kelimelerin nispeten yakın konumlandırılacağı şekilde, kelimeleri sürekli uzayda düşük boyutlu vektörlere etkin bir şekilde eşlemeyi sağlamak üzere geliştirilen dil teknikleridir (Mikolov 2013, Mikolov vd. 2013, Pennington vd. 2014). Bu modeller, bir kelime sembolünü girdi olarak alır ve her kelimeye benzersiz gerçek değerli bir "öznitelik vektörü" ilişkilendirmeyi öğrenerek onu öklid uzayına yerleştirir. Öğrenilen metin temsilleri, metin sınıflandırma, duygu analizi, konu modelleri, makine çevirisi ve sözcük türü etiketleme gibi çeşitli doğal dil işleme görevlerinin iyileştirilmesini hedeflemiştir (Jiao ve Zhang 2021).

Word2vec (Mikolov vd. 2013), etkili metin temsili için yaygın olarak kullanılan ve en tanınmış kelime yerleştirme modellerinden biri olmuştur. Word2vec modeli, bir girdi katmanı, bir çıktı katmanı ve bir gizli katmandan oluşan bir sinir ağı dil modelidir

(Mikolov vd. 2013). Kelimelerin kullanıldıkları bağlamlardan anlamsal özellikleri yakalayan etkili bir mimari üzerine kurulmuştur. İki ilişkili ancak farklı mimarisi mevcuttur. **Şekil 3.8**'de gösterildiği gibi, word2vec CBOW, hedef kelimeyi bağlam kelimelerinden tahmin etmeyi amaçlarken; word2vec Skip-gram ise hedef kelimeyi kullanarak bağlam kelimelerini tahmin eder.



Şekil 3.8 Word2vec modelinin CBOW ve Skip-gram mimarileri (Mikolov vd. 2013)

W kelime sözlüğündeki w kelime türünün "hedef" ve "bağlam" vektör temsilleri sırasıyla v_w ve $\tilde{v}_w$ olarak gösterilir. $w_1, w_2, \dots, w_M$ kelime dizisi verildiğinde, Skip-gram aşağıdaki kayıp işlevini en aza indirmeyi amaçlar:

$$L_{\text{Skip-gram}} = -\frac{1}{M} \sum_{t=1}^M \sum_{-k \leq j \leq k, j \neq 0} \log P(w_{t+j} | w_t) \quad (3.5)$$

Burada k , hedef kelime w_t 'nin pencere boyutudur ve w_{t+j} , w_t 'nin bağlam kelimelerinden biridir. Genel olarak, Denklem 3.5'teki hedef kelime w için bağlam kelimesi c 'nin olasılığı $P(c|w)$, softmax fonksiyonu kullanılarak hesaplanır.

$$P(c | w) = \frac{\exp(\tilde{v}_c^T v_w)}{\sum_{x \in W} \exp(\tilde{v}_x^T v_w)} \quad (3.6)$$

Diğer yandan, word2vec CBOW modeli ise, bağlam kelimelerinin vektör temsilleri ortalamasını $\tilde{v}_{avg}$ şeklinde alarak, hedef kelimeyi tahmin etmeyi amaçlar. Word2vec CBOW modelinin amacı, aşağıdaki kayıp işlevini en aza indirmektir:

$$L_{CBOW} = -\frac{1}{M} \sum_{t=1}^M \log P(w_t | w_{t-k}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+k}) \quad (3.7)$$

$$= -\frac{1}{M} \sum_{t=1}^M \log P(w_t | \text{avg})$$

$$= -\frac{1}{M} \sum_{t=1}^M \log \frac{\exp(v_{w_t}^T \tilde{v}_{avg})}{\sum_{x \in V} \exp(v_x^T \tilde{v}_{avg})}$$

Word2vec modelinden elde edilen kelime vektörleri, "güçlü" kelimesinin "kudretli" kelimesine yakın olması gibi, semantik olarak benzer kelimeleri temsil uzayında yakın konumlandırma özelliğini sergiler (Jiao ve Zhang 2021). Word2vec algoritmasının dikkat çekici bir diğer özelliği, belirli kelime vektörlerinin doğrusal kombinasyonu yoluyla anlamsal olarak anlamlı sonuçlar üretebilme yeteneğidir. Örneğin, "en büyük" - "büyük" + "küçük" = "en küçük" (Mikolov vd. 2013) gibi vektör aritmetiği kullanarak kelime benzetmeleri belirlemek mümkün hale gelmektedir.

3.8 Ağ Temsili Öğrenimi

Ağ analizinde, çok sayıda önemli görev, düğümler veya bunların ilişkileri (kenarlar) hakkında tahminler gerektirmektedir. Makine öğrenmesi alanında, çeşitli yöntemler girdi olarak bilgilendirici, ayırt edici ve bağımsız özelliklerden yararlanmaktadır. Bu durum, ağların makine öğrenmesi görevlerinde kullanılabilmesi için düğümler için bir özellik vektörü temsilinin oluşturulmasının elzem olduğunu göstermektedir. Geleneksel olarak, bu işlem, alan uzmanlarının sezgi ve uzmanlığına dayanarak alan özelinde özelliklerin manuel olarak tasarlanmasıyla gerçekleştirilir. Ancak, bu özellikler genellikle belirli görevlere özgüdür ve farklı tahmin görevlerine etkili bir şekilde genellenememektedir.

Metin temsilleri üzerine yapılan arařtırmalar, word2vec (Mikolov vd. 2013) gibi gncel modellerin olađanst dođruluđu ve verimliliđi sayesinde byk ilgi toplamıřtır. Bu kavramlar, diđer arařtırma alanlarını da etkilemiř olup, zellikle ađ bilimi alanında, bu ilerlemelerden esinlenen modeller aracılıđıyla kayda deđer bir etki yaratmıřtır.

Word2vec (Mikolov vd. 2013) modeli, metin verileri iin yođun temsiller đrenmek amacıyla nerilmiřtir. Bu yerleřtirme vektrleri, kelimelerin anlamlarını yakalayarak, anlamsal olarak benzer kelimeleri vektrel uzayda birbirine yakın konumlandırmaktadır.

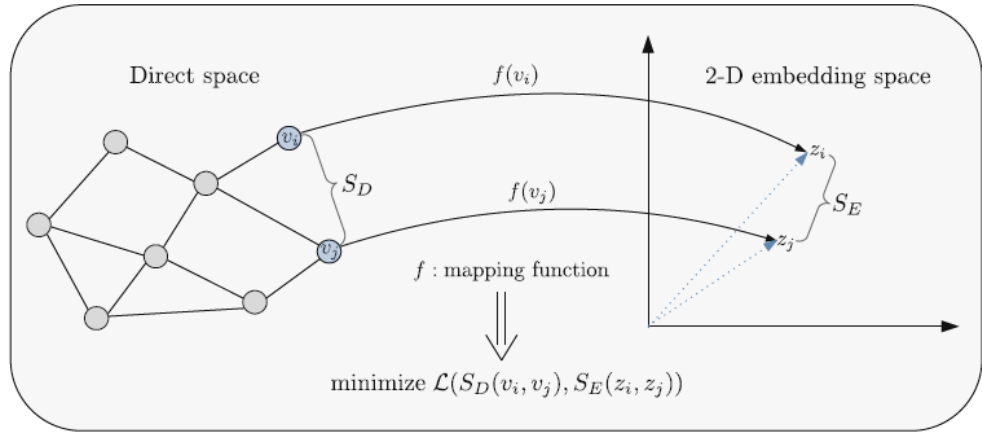
Komřuluk matrisleri gibi ađ temsillerinin yksek boyutlu ve ođunlukla seyrek yapısı gz nne alındıđında, dđm yerleřtirme vektrlerinin đrenilmesi, ađ verileri zerindeki makine đrenmesi uygulamaları iin kritik nem tařımaktadır. Bununla birlikte, genellikle zamansal dizileri girdi olarak gerektiren word2vec benzeri modellerin, ađ temsilleri đrenmek zere uyarlanması, ađ verilerinin dođası geređi dođrusal olmaması nedeniyle zorlu bir grev teřkil etmektedir.

Ađlar zerinde bir dil modeli đrenme kavramı ilk kez DeepWalk (Perozzi vd. 2014) ve node2vec (Grover ve Leskovec 2016) modelleri tarafından ortaya atılmıřtır. Her iki alıřmanın arařtırmacıları, word2vec Skip-gram kelime yerleřtirme modelinden yararlanarak, dođal dil iřleme ile ađ analizi arasındaki dikkat ekici benzerlikleri vurgulamıřlardır. Metin verisin dođrusal yapısı nedeniyle, ardıřık kelimeler zerinde kayan bir pencere kullanarak komřuluđun kolayca tanımlanmasına olanak tanır. Ancak, ađlar dođrusallıktan yoksun olduđundan, uygun bir komřuluk kavramının oluřturulması gerekmektedir. Bir dđmn komřuluđunu tanımlamak iin pratik ve verimli bir yntem, rasgele yryřlerden elde edilen dđm dizilerini kullanmaktır. Temel fikir, kelime yerleřtirme algoritmalarına girdi olarak kelimeler yerine dđm dizileri sunmaktır. Sezgisel olarak, aynı Skip-gram penceresindeki dđmler ađda birbirine yakın olduđundan, đrenilen temsiller ođunlukla ađda birbirine yakın olan dđmleri yakalamaktadır.

Her iki yntemin algoritmaları iki temel bileřenden oluřmaktadır. İlk ařama, bir dizi rasgele yryřn oluřturulmasını ierirken, ikinci ařamada ise bu rasgele yryřler,

yerleştirme vektörlerini oluşturmak üzere Skip-gram modelinin eğitiminde kullanılmaktadır. DeepWalk ve node2vec arasındaki temel fark, rasgele yürüyüşleri oluşturma yöntemlerinde yatmaktadır. DeepWalk, rasgele yürüyüşte bir sonraki düğümü, mevcut düğümün komşuları arasından birörnek bir şekilde seçmektedir. Buna karşılık, node2vec rasgele yürüyüşleri oluşturmada daha karmaşık bir yaklaşım benimseyerek, ağ yapısını daha etkili bir biçimde yakalamayı hedeflemektedir.

Ağ yerleştirme, giriş ağını $G = (V, E)$, düğüm sayısı $|V|$ 'den çok daha küçük olan d boyutlu bir uzaya eşlemeyi içerir. Amaç, ağın özelliklerini bu indirgenmiş boyutlu uzayda mümkün olduğunca doğru bir şekilde koruyan bir temsil belirlemektir. Bu sürecin çıktısı, ya tüm ağı temsil eden tek bir d boyutlu vektör, ya da ağın belirli bir parçasını (düğüm, kenar veya alt yapı gibi) temsil eden d boyutlu vektörler kümesi olabilir. **Şekil 3.9**'da gösterildiği gibi, bir ağ, f haritalama fonksiyonu kullanılarak, iki boyutlu yerleştirme uzayı gibi düşük boyutlu bir vektör uzayına izdüşürülür. f haritalama fonksiyonu, bir L kayıp fonksiyonunun optimize edilmesiyle belirlenir. Bu kayıp fonksiyonu, düğümlerin orijinal uzaydaki benzerlik ölçüleri (S_D) ile yerleştirme uzaydaki karşılık gelen benzerlikler (S_E) arasındaki farkı en aza indirmeyi amaçlar.



Şekil 3.9 Ağ yerleştirme işleminin gösterimi (Baptista vd. 2023)

3.9 Node2vec

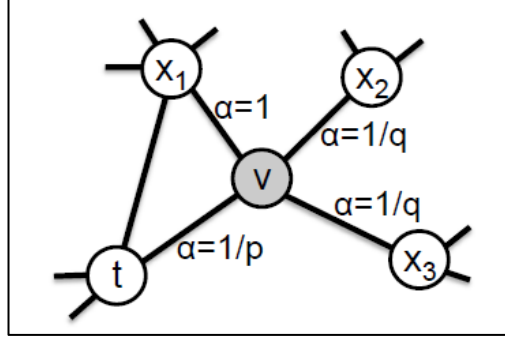
Node2vec (Grover ve Leskovec 2016), bir düğümün ağ komşuluğu kavramını daha esnek bir biçimde ele alarak DeepWalk (Perozzi vd. 2014) modelini genişletir. DeepWalk yalnızca rasgele yürüyüşlere dayalı komşuluklar inşa ederken, node2vec bu yaklaşımı genelleştirerek sadece aynı topluluktaki düğümleri birbirine yakın konumlandıran temsiller öğrenmenin yanı sıra, aynı zamanda benzer yapısal rollere sahip düğümleri de yakalar. Özellikle, ağlardaki düğümler ya ait oldukları topluluklara göre (homofiliye dayalı) ya da yapısal rollerine göre (yapısal eşdeğerlik) organize edilebilir. Node2vec, bu amaca ulaşmak için düğümlerin ağ komşuluklarını örneklemek amacıyla yanlı ikinci dereceden rasgele yürüyüş yaklaşımını kullanır.

Node2vec modelinin yazarlarına göre, bir düğümünden başlayarak bir ağı taramak için iki temel strateji mevcuttur: genişlik öncelikli arama ve derinlik öncelikli arama. Genişlik öncelikli arama ile örneklenen komşuluklar, genellikle yapısal eşdeğerlikle yakından örtüşen temsiller üretme eğilimindedir. Yapısal eşdeğerliği belirlemek için yerel komşulukların doğru bir şekilde karakterize edilmesi çoğunlukla yeterlidir. Örneğin, yapısal eşdeğerliğin göstergesi olan köprüler ve merkezler gibi ağ rolleri, genellikle her düğümün doğrudan komşuluklarını gözlemleyerek çıkarılabilir. Derinlik öncelikli aramada ise örneklenen düğümler, homofili temelli toplulukları koruduğu görülmektedir; bu da aynı topluluktaki düğümlerin birbirine yakın yerleştirme olasılığının daha yüksek olmasını sağlar.

Yukarıdaki gözlemlerden yola çıkarak, node2vec, yanlı bir rasgele yürüyüş prosedürü kullanarak esnek bir komşuluk örneklem stratejisi uygular. Bu yaklaşım, genişlik öncelikli arama ve derinlik öncelikli arama arasında düzgün bir geçişe olanak tanır, böylece node2vec'in ağ içerisindeki hem yerel yapısal eşdeğerliliği hem de küresel topluluk yapılarını yakalamasına yardımcı olur.

Node2vec, DeepWalk ile aynı parametreleri paylaşır; bunlar arasında her düğümünden başlatılan w_l uzunluğundaki rasgele yürüyüşlerin sayısı (n_w) ve Skip-gram modeli tarafından bağlam olarak kullanılan pencere boyutu (w_s) bulunmaktadır. Bununla

birlikte, node2vec, rasgele yürüyüşün adımlarını yönlendirmek amacıyla kenarlara ağırlık atamak için p ve q olmak üzere iki ek parametre kullanmaktadır.



Şekil 3.10 Node2vec algoritmasının yanlı rasgele yürüyüş prosedürü (Grover ve Leskovec 2016)

Şekil 3.10’da yürüyüş, t düğümünden v düğümüne geçiş yapmış ve şu anda v düğümünden bir sonraki adımını değerlendirmektedir. Kenar etiketleri, α ile temsil edilen arama yanlılıklarını işaret etmektedir. (t, v) kenarını kat etmiş ve şu anda v düğümünde bulunan bir rasgele yürüyüş ele alındığında (Şekil 3.10). Rasgele yürüyüş, bir sonraki adımına karar vermek için v düğümünden çıkan (v, x) kenarlarındaki geçiş olasılıklarını değerlendirir. Düğüm v ’den x ’e geçiş için normalleştirilmemiş olasılık aşağıdaki gibi belirlenir:

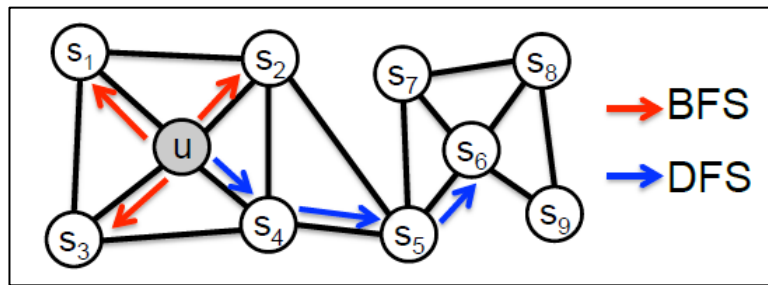
$$\pi_{vx} = \alpha_{pq}(t, x) \cdot w_{vx} \quad (3.8)$$

Burada v mevcut düğümü, t önceki düğümü ve x sonraki düğümü temsil etmektedir. Burada, $\alpha_{pq}(t, x)$ Denklem (3.9)’da gösterildiği üzere p ve q tarafından tanımlanan yanlı aramayı ifade ederken, w_{vx} , (v, x) düğüm çifti arasındaki bağlantı ağırlığını belirtir; ağırlıksız ağlar için w_{vx} değeri 1’e eşittir.

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p} & \text{eğer } d_{tx} = 0 \\ 1 & \text{eğer } d_{tx} = 1 \\ \frac{1}{q} & \text{eğer } d_{tx} = 2 \end{cases} \quad (3.9)$$

Burada d_{tx} , t ve x düğümleri arasındaki en kısa yol mesafesini ifade eder ve 0, 1 veya 2 değerlerinden birini almalıdır. p ve q parametreleri, rasgele yürüyüşün komşuluğu nasıl keşfedeceğini kontrol eder. Geri dönüş parametresi (p), rasgele yürüyüş sırasında bir düğümün hemen tekrar ziyaret edilme olasılığını düzenler. (p) yüksek bir değere ($> \max(q, 1)$) ayarlandığında, yürüyüşün sonraki iki adımda geldiği düğüme geri dönme olasılığı azalır. Bu ayar, ağıın ölçülü keşfedilmesini teşvik eder ve yürüyüşün iki adım içinde aynı düğümleri tekrar ziyaret etmesini engelleyerek gereksiz tekrarları önler. Aksine, (p) düşük bir değere ($< \min(q, 1)$) ayarlandığında, yürüyüşün önceki ziyaret edilen düğüme geri dönme olasılığı artar, böylece yürüyüş başlangıç düğüme daha yakın kalır.

İçeri-dışarı parametresi (q), rasgele yürüyüşün "içe dönük" ve "dışa dönük" düğümler arasında ayrım yapmasına olanak tanır. ($q > 1$) olduğunda, yürüyüş daha önce ziyaret edilen (t) düğüme yakın olan düğümlere yönelim gösterir (Şekil 3.10). Bu durum, yürüyüşün başlangıç düğüme göre ağıın yerel bir görünümünü oluşturur ve örneklenen düğümlerin çoğunlukla küçük bir yerel alan içinde bulunduğu genişlik öncelikli arama davranışına yaklaşır (Şekil 3.11). Buna karşılık, ($q < 1$) olduğunda, yürüyüş (t) düğümünden daha uzak düğümleri ziyaret etme eğilimi gösterir, dışa doğru keşfi teşvik eder ve derinlik öncelikli arama davranışına benzerlik gösterir (Şekil 3.11).



Şekil 3.11 Düğüm u 'dan başlayan genişlik öncelikli arama ve derinlik öncelikli arama stratejileri ($wl = 3$) (Grover ve Leskovec 2016)

İlk aşamada oluşturulan rasgele yürüyüşler, düğümlerin görüldüğü bağlamı oluşturur; bu düğüm dizileri, kelimelerin metin verilerindeki bağlam penceresinde görünmesine benzer. İkinci aşamada ise, node2vec, oluşturulan düğüm dizilere word2vec Skip-gram

(Mikolov vd. 2013) modelini uygulayarak düğümler için vektör temsillerini oluşturur. Sonuç olarak, ağdaki hem yerel hem de küresel yapıları yakalayıarak zengin ve bilgilendirici düğüm yerleřtirmeleri saęlar.



4. AĞ TEMSİLİ ÖĞRENİMİ YÖNTEMLERİNİN GELİŞTİRİLMESİ

Bu bölümde tez çalışmasında önerilmiş olan ağ temsil öğrenimi yöntemler hakkında detaylı bilgi verilmiştir.

4.1 Derece Tabanlı Rasgele Yürüyüş Modeli

Ağ temsil öğrenimi, hem yerel hem de küresel komşuluk bilgisini sayısal vektörlere kodlayarak, çeşitli gerçek dünya sistemlerinin matematiksel modellemesinde kritik bir rol oynamaktadır. Farklı yerleştirme teknikleri arasında, rasgele yürüyüş temelli algoritmalar oldukça başarılı olduklarını kanıtlamışlardır. Rasgele yürüyüş istatistiklerinden faydalanmanın temelindeki kavram, düğümlerin eş zamanlı oluşma durumlarını tespit etmektir. Düğümlerin aynı rasgele yürüyüşte eş zamanlı oluşumu, düğüm grubunun bağlantısallığını vurgulamaktadır. Bu rasgele yürüyüşler sırasında toplanan bilgiler, doğal dil işleme algoritmalarında kullanılan verilere benzer nitelik taşımaktadır.

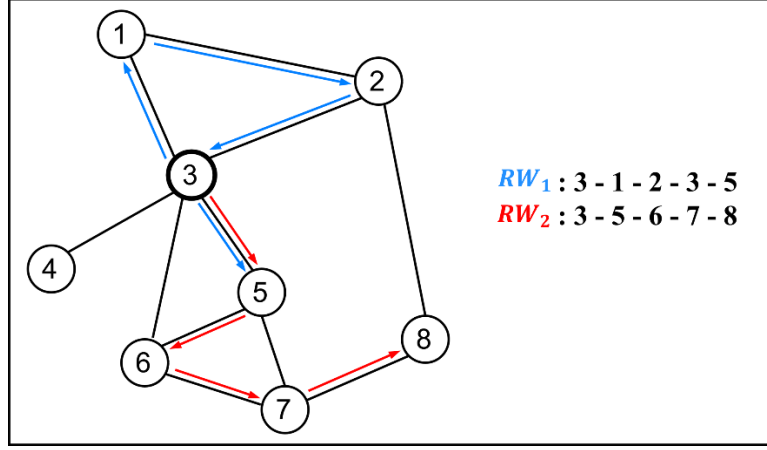
Literatürde yaygın olarak bilinen rasgele yürüyüş temelli yerleştirme yöntemlerinden biri node2vec (Grover ve Leskovec 2016) yöntemidir. Node2vec, düğümlerin eş zamanlı oluşum istatistiklerinden faydalanan bir optimizasyon stratejisi izleyen sık bir yerleştirme tekniği olarak karşımıza çıkmaktadır. Node2vec yöntemi her bir düğümden sabit sayıda yanlı rasgele yürüyüş başlatarak düğüm çevresinden yapısal bilgi toplamayı amaçlar. Node2vec ve diğer rasgele yürüyüş temelli yöntemlerde, rasgele yürüyüşlerin üretimi, yerleştirme sürecinin hesaplama açısından en temel kısmını teşkil etmiş olup, hesaplama gereksinimleri ağı boyutuyla doğru orantılı olarak artış göstermektedir.

Gerçek dünya ağları arasında, ölçekten bağımsız ağlar özel bir konuma sahiptir; zira bu ağlar, gerçek dünya ağlarının çoğunluğunu teşkil etmektedir (Barabási ve Bonabeau 2003). Ölçekten bağımsız ağların karakteristik güç yasası derece dağılımı, çok sayıda düğümün düşük dereceye sahip olduğunu, az sayıda düğümün ise yüksek dereceye sahip olduğunu göstermektedir. Bunun sonucunda, node2vec yönteminde sabit sayıda rasgele yürüyüş durumu göz önüne alındığında, düşük dereceli düğümleri aşırı örnekleme eğilimi

gösterirken, yüksek dereceli düğümler yeterince örneklenememektedir. Tüm düğümlere eşit muamele edildiğinde, yüksek dereceli düğümlerin örnekleme oranını artırmak, her düğümden başlatılan yürüyüş sayısının artırılmasını gerektirmektedir. Ancak bu yaklaşım, ağın boyutuyla orantılı olarak hesaplama süresinde bir artışa yol açmakta ve düşük dereceli düğümler için aşırı sayıda yürüyüşle sonuçlanmaktadır. Neticede, yerleştirme aşamasında dengesiz veri oluşumuna sebebiyet vermektedir.

Önerilen yöntem, node2vec özellik çıkarma algoritmasında bir değişiklik önermeyi amaçlamaktadır. Aşırı örnekleme ortadan kaldırmak amacıyla, rasgele yürüyüşlerin sayısı düğümün derecesiyle orantılı olarak seçilmektedir. Düşük dereceli düğümler, yüksek dereceli düğümlere kıyasla nispeten daha az örneklenmektedir. Ölçekten bağımsız ağlardaki düşük ve yüksek dereceli düğümlerin dağılımı göz önüne alındığında, bu yaklaşım toplam yürüyüş sayısını önemli ölçüde azaltırken yüksek dereceli düğümlerin örnekleme oranını artırmaktadır. Bu değişiklik, artan ağ boyutuyla birlikte ortaya çıkan aşırı hesaplama süresi sorununa çözümler sunmakta ve dengesiz veri problemlerinin giderilmesine katkıda bulunmaktadır. Önerilen modelde, her düğümden başlayan yürüyüşlerin sayısı, sabit bir sayı yerine ilgili düğümün derecesiyle orantılı olacak şekilde ayarlanırken, yerleştirme işleminin geri kalan kısmı için node2vec stratejisi kullanılmaktadır. Sonuç olarak, önerilen yöntem yaklaşık %50 daha az yürüyüş yaparak aynı hassasiyette düğüm yerleştirme vektörler elde edilmesini sağlamıştır.

Ağıdaki her düğüm (v_i) için bir komşu kümesi (N_i) bulunmaktadır. Bu komşular, belirli bir stratejiyi takip eden bir dizi rasgele yürüyüş aracılığıyla belirlenmektedir. Bu strateji, hem yerel hem de küresel komşuluk bilgisinin yerleştirmeye dahil edilmesini sağlamalıdır. **Şekil 4.1**'de gösterildiği üzere, 3. düğümün 5 komşusu vardır, $N_3 = \{v_1, v_2, v_4, v_5, v_6\}$. Yürüyüş uzunluğu 4 olmak üzere v_3 düğümünden iki rasgele yürüyüş başlatıldığında, ilk yürüyüş $RW_1 (v_3 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_5)$ şeklinde, ikinci yürüyüş RW_2 ise $(v_3 \rightarrow v_5 \rightarrow v_6 \rightarrow v_7 \rightarrow v_8)$ şeklinde gerçekleşebilir.



Şekil 4.1 Örnek bir ağda v_3 düğümünden başlatılan 2 rasgele yürüyüşün temsili

Bu yürüyüşler, cümle benzeri diziler oluşturmaktadır:

$$S_1: v_3 v_1 v_2 v_3 v_5$$

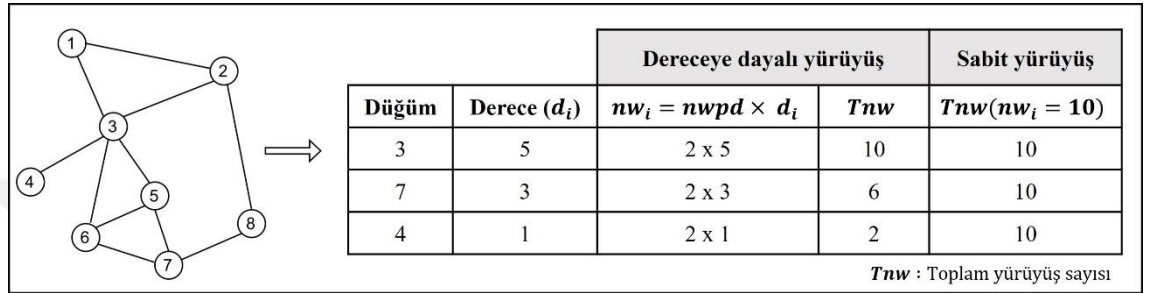
$$S_2: v_3 v_5 v_6 v_7 v_8$$

Cümle benzeri düğüm dizileri oluşturma fikri, düğümler arasındaki benzerlikleri ilgili yerleştirme vektörleri aracılığıyla yakalamayı hedeflemektedir. Her rasgele yürüyüş, düğümler arasında paylaşılan gizli özellikleri kapsayan bir dizi üretmektedir. Yürüyüş uzunluğu, ağın daha derinlemesine keşfedilmesine olanak tanırken, yürüyüş sayısı çeşitli bağlantı örüntülerini ortaya çıkarmaktadır. Netice itibarıyla, daha hassas yerleştirme vektörleri elde edilebilmektedir. Bu sebeple, her düğümden kaynaklanan farklı yolları doğru bir şekilde haritalandırmak için yeterli uzunlukta ve yeterli sayıda yürüyüş gerçekleştirilmelidir.

Sabit sayıda yürüyüşün kullanıldığı senaryolarda, her bir düğümden eşit sayıda yürüyüş başlatılmaktadır. Buna karşın, yürüyüş sayısının düğümün derecesiyle orantılı olduğu durumda, bir düğümden başlatılan yürüyüş sayısı şu şekilde ifade edilmektedir:

$$nw_i = nw_{pd} \times d_i \quad (4.1)$$

nw_i ve d_i sırasıyla i 'nci düğümden başlatılan yürüyüş sayısını ve bu düğümün derecesini temsil etmektedir. $nwpd$ ise derece başına düşen yürüyüş sayısını ifade etmektedir. **Şekil 4.2**, örnek bir ağda dereceye dayalı node2vec ile standart node2vec arasındaki rasgele yürüyüş stratejilerindeki farklılığı görsel olarak sunmaktadır. **Şekil 4.3**, derece tabanlı ağırlıklı örnekleme yaklaşımını kullanarak düğüm başına rasgele yürüyüşler oluşturma sözde kodunu göstermektedir.



Şekil 4.2 Dereceye dayalı rasgele yürüyüşler ile sabit sayıda rasgele yürüyüşler arasındaki sayısal farklılıkların şematik temsili

Algoritma : Dereceye Dayalı Örnekleme

Girdi: Ağ $G = (V, E)$

Başlangıç:

- Düğüm sayısı N ;
- Düğüm derecelerinin dizisi D ;
- Yürüyüş uzunluğu WL ;
- Derece başına yürüyüş sayısı WD ;
- Geri dönme ağırlığı parametresi p ;
- Komşu ağırlık parametresi q ;

Dereceye dayalı rasgele yürüyüşler oluşturma bölümü:

```

for node in N do
    StartNode = node
    epochs = WD * D[node]
    Walk[node] = Random_Walk(G, StartNode, WL, epochs, p, q)
endfor

```

- Düğümler üzerinde döngü
- Başlangıç düğümü
- Düğümden kaç kez yürüyüş başlatılacağını gösterir

Şekil 4.3 Dereceye dayalı rasgele yürüyüş algoritması

Sabit sayıda rasgele yürüyüş örneği için, ağ büyüdükçe düğüm başına yürüyüş sayısının artması gerekmektedir. Bu gereklilik, ağ boyutu ile maksimum derece arasındaki korelasyondan kaynaklanmaktadır. Bu ayarlama yapılmadığı takdirde, merkez düğümlerin doğru bir şekilde örneklenmesi mümkün olmayacaktır. Eşdeğer hassasiyeti

korumak adına, yürüyüş sayısının ağ boyutuyla orantılı olarak ölçeklenmesi elzemdir. Bunun sonucunda, hesaplama yükü boyutla doğrusal bir artış göstermemektedir. Ayrıca, her düğümden eşit sayıda rasgele yürüyüş başlatmak, düşük dereceli düğümler için oldukça benzer dizilerin oluşmasına sebebiyet verebilmektedir. Bu aşırı örnekleme, gereksiz bilgi tekrarına yol açarak, yerleştirme aşamasında veri dengesizliğine neden olmaktadır.

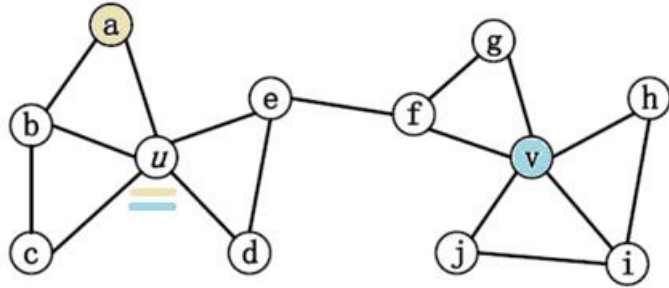
Doğal dil işlemede, cümlelerin doğal akışı kelimelerin bağlantı yapısını belirler. Bazı kelimeler sıkça kullanılırken, daha az yaygın kelimeler cümlelerde ender olarak yer almaktadır. Bu nedenle, anlamlı cümleleri işleyen bir yerleştirme algoritması, yaygın kelimeleri (yüksek dereceli merkez düğümler) daha az yaygın kelimelerden (düşük dereceli düğümler) doğal bir şekilde ayırt edebilmektedir. Buna karşın, ağ temsil algoritmalarında böyle doğal bir ayırım yalnızca ağın bağlantı yapısı aracılığıyla elde edilebilmektedir. En basit ve en sezgisel seçenek, düğüm derecesidir. Bir düğümden başlatılan yürüyüş sayısı, o düğümün derecesiyle orantılı olduğunda, bu düğümlerin görünme sıklığı, bağlantı yapısıyla orantılı olacaktır.

Önerilen model, komşu düğümleri ziyaret etmek için node2vec (Grover ve Leskovec 2016) stratejisini kullanmaktadır; ancak buradaki fark, her düğümden başlayan yürüyüş sayısının sabit bir sayı yerine düğümün derecesiyle orantılı olmasıdır. Bu yaklaşımın üç temel avantajı bulunmaktadır:

- Yüksek dereceli düğümler, cümle benzeri yapılarda daha fazla önem kazanarak, ağırlıklı yerleştirme vektörlerin oluşmasına yol açmaktadır. Bu derece tabanlı yöntem, rasgele yürüyüşlerin olasılıksal doğasıyla birleştiğinde, önem örnekleme işlevi görerek verimliliği artırmaktadır.
- Düşük dereceli düğümlerin baskınlığı, önemli katkılarının bazı ağ özelliklerini gölgeleyebileceği dengesiz bir veri problemi yaratabilmektedir. Rasgele yürüyüş sayısında derece orantısallığının uygulanmasıyla bu dengesizlik etkili bir şekilde azaltılmaktadır.
- Derece tabanlı rasgele yürüyüşlerin benimsenmesi, hesaplama süresini önemli ölçüde azaltarak büyük ölçekli ağların işlenmesini mümkün kılmaktadır.

4.2 Geçiş Olasılık Matrisi Modeli

Çeşitli ağ analiz görevlerine uyarlanabilen esnek bir ağ temsili öğrenme modeli, iki temel ilkeye dayanmalıdır: Birincisi, benzer yerel bağlamları paylaşan düğümler için benzer vektör temsilleri oluşturulmalıdır. İkincisi, benzer yerel yapıları paylaşan düğümler birbirine yakın bir şekilde yerleştirilmelidir. Mevcut yaklaşımların çoğu, ağlara özgü yerel yapısal kalıplara duyarsızdır ve yalnızca düğümlerin yerel bağlamına (homofili) üzerine yoğunlaşır; yani, bir topluluk içindeki sıkı bağlı düğümler benzer kabul edilir. Bazı algoritmalar, bir düğümü çevreleyen ağ bölgesini genişleterek ve daha fazla yapısal bilgi toplayarak bu sorunu ele almaktadır. Ancak bu yaklaşım, başka bir önemli zorluğu beraberinde getirmektedir: söz konusu algoritmalar hesaplama açısından yoğun olup, genellikle yalnızca daha küçük ağlara uygulanabilmektedir. Bu nedenle, ağdan toplanan bilgi hacmi, algoritmanın karmaşıklığı ve bilgi toplama teknikleri arasında bir denge kurulması gereklidir.



Şekil 4.4 Homofili ve yapısal rol benzerliğinin açıklayıcı bir örneği

Gerçek dünya ağları, düğüm rollerini belirleyen çok sayıda yerel yapısal kalıpları içermektedir. Örneğin, Şekil 4.4'te, u düğümü ve a düğümünün aynı sıkı bağlı topluluğa ait olduğunu gözlemlenebilir, dolayısıyla homofili açısından benzerdirler. Buna karşılık, u ve v düğümleri aynı toplulukta değildir, ancak aynı bir yapısal merkez rolünü paylaşıyorlar, bu nedenle yapısal eşdeğerliğe dayalı olarak her iki (u ve v) düğümü de benzer olarak kabul edebilir.

Bu iki eşdeğerlik, gerçek dünyada var olan herhangi bir ağ için geçerlidir ve bu nedenle birbirleriyle rekabet halinde değiller, aksine birbirlerini tamamlayıcı niteliktedirler. Bu durum, esnek bir ağ temsili öğrenme modelinin, yalnızca aynı topluluktaki komşu düğümlerin vektör uzayında yakın temsillere sahip olmasını sağlamakla kalmayıp, aynı yapısal rolü paylaşan düğümlerin de benzer temsil vektörlerine sahip olmasını gerektirir. Bu sayede, düğümlerin vektör temsilleri, gelecekte daha çeşitli makine öğrenimi uygulamalarında kullanılabilir.

Düğüm özelliği çıkarma algoritmaları sıklıkla rasgele yürüyüşleri temel bir teknik olarak kullanmaktadır. Özellikle, literatürde çok iyi bilinen ve yüksek atıf alan DeepWalk (Perozzi vd. 2014) ve node2vec (Grover ve Leskovec 2016) yöntemleri, doğal dil işlemede cümlelerin oluşturulma biçimine benzer şekilde, kapsamlı rasgele yürüyüş dizilerinin üretilmesine dayanmaktadır. Bu bağlamda, düğümler kelimelere, rasgele yürüyüşler ise cümlelere tekabül etmektedir. Hedef düğüm ile komşu düğümleri arasındaki ilişkisel bilgi, bu rasgele yürüyüş dizilerinden türetilmektedir. Bu iki algoritmaların ötesinde, Ivanov ve Burnaev (2018) çalışmalarında ağ temsillerini öğrenmek için doğal bir ağ nesnesi olan anonim yürüyüşlerden yararlanmanın mümkün olduğunu kanıtlanmışlar. Bu yeni yöntem, rasgele yürüyüşleri, düğümleri yürüyüş içindeki oluşumlarına göre yeniden etiketleyerek anonim yürüyüş dizilerine dönüştürmeyi içermektedir (Ivanov ve Burnaev 2018).

Bu çalışma, komşu düğümler arasındaki geçiş olasılıklarını hesaplamak üzere tasarlanmış yenilikçi, ölçeklenebilir ve denetimsiz bir düğüm yerleştirme algoritmasını sunmakta ve aynı zamanda bu algoritmanın temelini oluşturan geçiş olasılık matrisi (TPM) adını verdiğimiz esnek bir yaklaşımı önermektedir. Bu olasılıklar, hedef düğümü çevreleyen bölgenin özgün bağlantı yapısını özetlemektedir. Temel fikir, bir düğümün yerel bağlantı örüntülerini, sınırlı sayıdaki doğrudan komşusunu tarayarak yakalamaktır. Bu yaklaşım, hesaplama karmaşıklığını azaltmanın yanı sıra, ağın derinlemesine keşfini gerektirmeden verimli bilgi toplanmasını sağlayarak performansı da artırmaktadır. Bu, her düğümden önceden belirlenmiş adım sayısı kadar rasgele yürüyüş başlatılarak gerçekleştirilir. Akabinde, bu rasgele yürüyüşler, düğümlerin etrafındaki bağlantıların çeşitliliğini ve zenginliğini yakalamak için anonim yürüyüşlere dönüştürülür. Düğümlerin kimliğine

odaklanmak yerine, bu yöntem yerel yapısal örüntüleri vurgulamaktadır. Bu anonim yürüyüşler, kritik yerel yapısal bilgiyi muhafaza eder ve geçiş olasılığı matrisinin elemanlarının hesaplanmasında belirleyici rol oynar. TPM, her zaman adımında komşu düğümlere geçiş olasılığını tanımlar ve önceki, yeni veya daha önce ziyaret edilmiş düğümlere ulaşma olasılıklarını içerir. Bu veriler, yerel bağlantıların yoğunluğunu veya sıklığını karakterize eden ve özgün geçiş olasılıkları desenini ifade etmektedir.

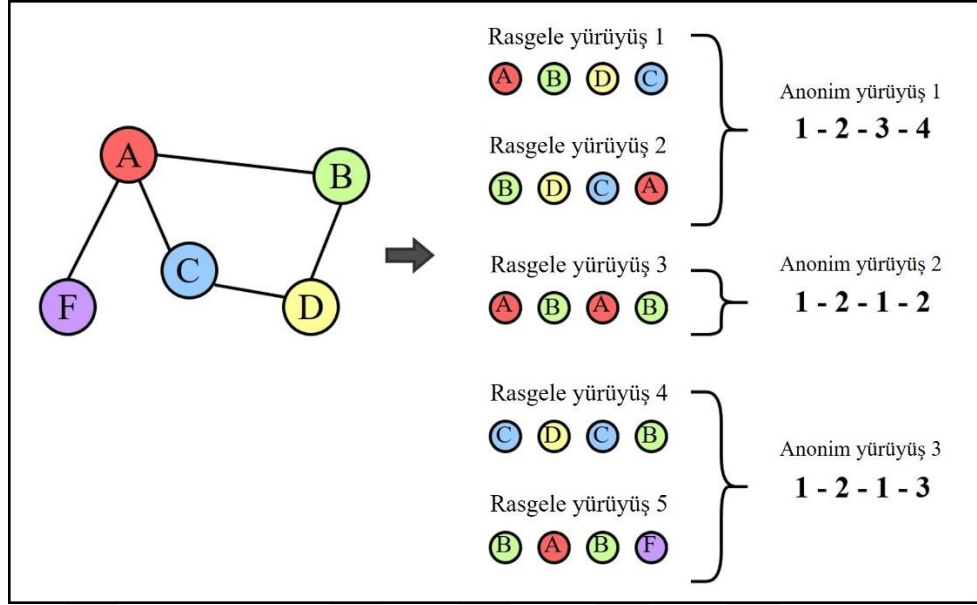
4.2.1 Anonim rasgele yürüyüşler

Anonim rasgele yürüyüş, bir ağdaki düğümlerin yeniden etiketlenilmesi sürecini ifade etmektedir. Rasgele yürüyüş dizisinde, düğümler ortaya çıkış sıralarına göre etiketlenilerek anonim bir yürüyüş dizisi oluşturulur. v_i düğümünden başlayan wl uzunluğundaki bir rasgele yürüyüş, $RW = (v_1, v_2, \dots, v_{wl})$ şeklinde bir düğüm etiketleri dizisi olarak temsil edilir; burada her v_i , V kümesinin bir elemanıdır. Anonimleştirme süreci, özgün düğüm etiketlerini dizideki karşılık gelen konumlarına eşler ve $AW = (f(v_1), f(v_2), \dots, f(v_{wl}))$ biçiminde gösterilir.

Verilen $RW = (v_0, v_1, \dots, v_{wl})$ dizisi, uzunluğu wl olan bir rasgele yürüyüşü temsil etmektedir. Bu durumda, $pos(RW, v_i)$ ifadesi tanımlanır; burada $pos(RW, v_i)$, RW dizisindeki v_i düğümünün tüm konumlarını içeren bir listeyi belirtir ve bu liste $p = (p_1, p_2, \dots, p_k)$ şeklinde gösterilir. Yürüyüşün herhangi bir adımında daha önce ziyaret edilmiş bir düğüm tekrar ziyaret edilirse, ilk kez görünen konum indeksi değeri kullanılır. Burada p_i , v_i düğümünün RW dizisinde bulunduğu konumu ifade eder. Bir rasgele yürüyüş RW için anonim yürüyüş, aşağıdaki şekilde tanımlanır:

$$AW = (f(v_1), f(v_2), \dots, f(v_{wl})) \quad (4.2)$$

$$f(v_i) = \min_{p_j \in pos(RW, v_i)} pos(RW, v_i) \quad (4.3)$$



Şekil 4.5 Anonim yürüyüş kavramını gösteren bir örnek

Şekil 4.5'te belirtildiği üzere, ağdaki çeşitli rasgele yürüyüşler (1 ve 2) ve (4 ve 5), aynı anonim yürüyüşler 1 ve 3'ü temsil etmektedir. Buna karşın, rasgele yürüyüş 3 ise başka bir anonim yürüyüş olan 2'ye karşılık gelmektedir. Bu anonimleştirme işlemi, küresel düğüm etiketlerine ihtiyaç duymaksızın ağın yerel yapısını yeniden oluşturmak için gerekli temel bilgileri muhafaza eder. Micali ve Zhu (2016), v_i düğümünden başlatılan w_l uzunluğundaki anonim rasgele yürüyüşlerin, v_i düğümünün yakın çevresini yeniden yapılandırmada kullanılabileceğini göstermiştir (Micali ve Zhu 2016). Sonuç olarak, anonim rasgele yürüyüşler düğümlerin yerel topolojik özelliklerini yakalar ve bu özellikler, ağın küresel yapısının yeniden inşasına yönelik tutarlı bilgiler sağlar.

4.2.2 Geçiş olasılıkları aracılığıyla yerel topoloji çıkarımı

Rasgele yürüyüşlerin davranışı, herhangi bir v_i düğümünün yerel bağlantı yapısını temsil eder. Seyrek bağlantılı bölgelerde, rasgele yürüyüş dizileri neredeyse her adımda yeni düğümler içerir. Buna karşın, yoğun bağlantılı komşuluklarda, önceden karşılaşılan düğümlere yeniden uğrama olasılığı artış gösterir. Bu bilgi, v_i düğümüne yakın bağlantı örüntüleri hakkında özgün bir içgörü sağlar. Dolayısıyla, geçiş olasılığı, düğümlerin topolojik özelliklerini çıkarmak için ideal bir aday olarak ortaya çıkar.

Önerilen TPM yöntemi, geçiş olasılığı aracılığıyla belirli bir düğümün yerel topolojik yapısını haritalandırmayı amaçlar. Bu yöntemde, düğüm yerleştirme süreci, v_i düğümünün belirli bir adım sayısı içinde herhangi bir düğüme ulaşma olasılıklarının hesaplanmasına dayanır. Komşu bir düğüme ulaşma olasılığı, ilgili düğümün derecesiyle orantılıdır. Böylece, v_i düğümünden başlayan bir dizi anonim rasgele yürüyüşten, belirli bir zaman adımında yeni bir düğüme ulaşma veya daha önce ziyaret edilen bir düğüme geri dönme dağılımı, geçiş matrisinin bileşenlerini oluşturur. Sonuç olarak, v_i düğümünün özellik yerleştirme vektörü, bu geçiş matrisinden elde edilen geçiş olasılığı matrisinden türetilir.

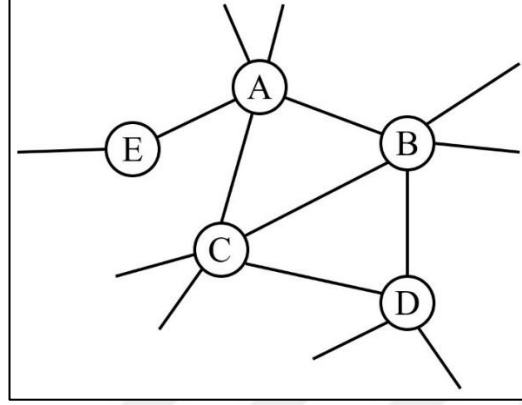
4.2.3 Geçiş matrisi hesaplanmasına yönelik kavramsal yaklaşım

Rasgele yürüyüşlerin kullanımı, bir ağ içerisinde düğüm dizilerinin oluşturulmasını kolaylaştırır. Bu dizideki düğümlerin düzeni, ağın bağlantısallık yapısını yansıtır ve yerel komşuluk hakkında zengin bilgiler içerir. Ağ üzerindeki gezinme genişledikçe, daha küresel bilgiler kademeli olarak elde edilir.

Bu bağlamda, her düğümünden başlayan ve sabit uzunlukta wl olan rasgele yürüyüşler gerçekleştirilerek, bu yürüyüşler wl adımlı anonim yürüyüşlere dönüştürülür. Başlangıç düğümü v_i , anonim yürüyüşteki sıfırıncı düğüm a_0 olarak kabul edilir. Dizideki bir sonraki düğüm, rasgele yürüyüşün ilk adımında ulaşılan v_i düğümünün komşularından biridir. v_i düğümünün her bir komşusunun ziyaret edilme olasılığı eşittir ve bu olasılık v_i düğümünün derecesi d_i ile orantılıdır. Bu işlem, rasgele yürüyüşlerin sonraki adımları için de tekrarlanır; böylece ziyaret edilen düğümlerin komşuluklarının altta yatan topolojik yapısı, bir sonraki adımın ayrıntılarına gömülür.

Teorik olarak, daha önce ziyaret edilmiş bir düğümün bağlantı yapısı göz önüne alındığında, bir sonraki adımda ya yeni, ziyaret edilmemiş bir düğüme ya da daha önce ziyaret edilmiş bir düğüme ulaşım gerçekleşebilir. Önceden ziyaret edilmiş bir düğüme yeniden uğramanın tek ön koşulu, söz konusu düğümle doğrudan bir bağlantının mevcut olmasıdır. Bu sebeple, halihazırda keşfedilmiş bir düğümü yeniden ziyaret etme

olasılığının yüksek olması, yoğun bir şekilde bağlantılı yerel bölgenin bir göstergesi niteliğindedir.



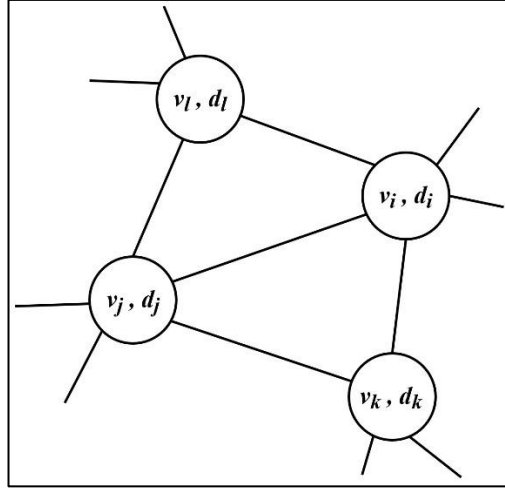
Şekil 4.6 Örnek bir ağ

Şekil 4.6’da gösterilen örnek ağda, A düğümünden başlayan ve uzunluğu 4 olan bir dizi rasgele yürüyüş başlatıldığında ve bu yürüyüşler anonim yürüyüşlere dönüştürüldüğünde aşağıdaki gibi gösterilir:

$$\begin{aligned}
 RW_1: A \rightarrow B \rightarrow C \rightarrow A \rightarrow E & \quad AW_1: a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_0 \rightarrow a_3 \\
 RW_2: A \rightarrow C \rightarrow D \rightarrow C \rightarrow A & \quad AW_2: a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_1 \rightarrow a_0 \\
 RW_3: A \rightarrow B \rightarrow D \rightarrow B \rightarrow A & \quad AW_3: a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_1 \rightarrow a_0
 \end{aligned}$$

Bu anonim yürüyüşler, komşu çeşitliliği ve topolojik yapılar hakkında bilgi taşımaktadır. Bu nedenle, iki farklı rasgele yürüyüşün (RW_2 ve RW_3) aynı anonim yürüyüşleri üretmesi mümkündür. Mevcut düğümden bir sonraki adım için rasgele bir yürüyüşte, temel olarak üç farklı olasılık mevcuttur:

P_{new}	Yeni bir düğüm ziyaret edilir
$P_{previous}$	Önceki düğüm ziyaret edilir (geriye dönüş)
$P_{already}$	Önceden ziyaret edilmiş bir düğüm ziyaret edilir



Şekil 4.7 Örnek bir ağ, v_i düğümünün derece sayısı d_i olarak ifade edilmektedir

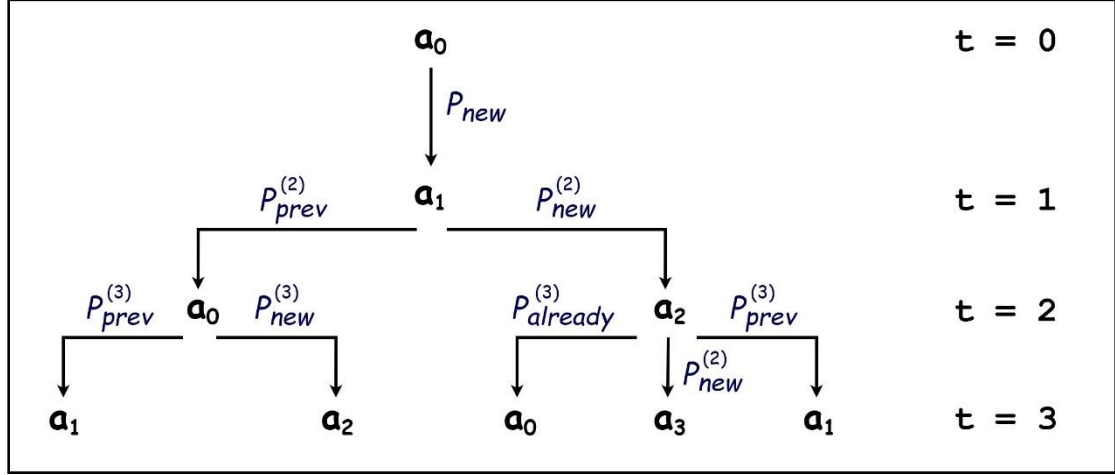
Şekil 4.7’de gösterilen rasgele yürüyüşün v_l düğümünden başladığını varsayalım. İlk adımın v_i düğümüne, ikinci adımın ise v_j düğümüne ($v_l \rightarrow v_i \rightarrow v_j$) atıldığını kabul edelim. Bir sonraki adım için olası seçenekler şu şekilde sıralanabilir:

$P_{j,k}$	Yeni bir düğümü ziyaret
$P_{j,i}$	Önceki düğüme geriye dönüş
$P_{j,l}$	Önceden ziyaret edilmiş bir düğüme ziyaret

At step 0	with	$P = 1$	be in node a_0	a_0
At step 1	with	$P_{new}^{(1)}$	be in node a_1	$a_0 \rightarrow a_1$
At step 2	with	$P_{prev}^{(2)}$	be in node a_0	$a_0 \rightarrow a_1 \rightarrow a_0$
	with	$P_{new}^{(2)}$	be in node a_2	$a_0 \rightarrow a_1 \rightarrow a_2$
At step 3	with	$P_{prev}^{(3)}$	be in node a_0	$a_0 \rightarrow a_1 \rightarrow a_0 \rightarrow a_1$
if $P_{prev}^{(2)} == 1$	with	$P_{new}^{(3)}$	be in node a_2	$a_0 \rightarrow a_1 \rightarrow a_0 \rightarrow a_2$
if $P_{new}^{(2)} == 1$	with	$P_{prev}^{(3)}$	be in node a_1	$a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_1$
	with	$P_{new}^{(3)}$	be in node a_3	$a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_3$
	with	$P_{already}^{(3)}$	be in node a_0	$a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_0$

Şekil 4.8 Üç adımlı anonim yürüyüşün olası dizilimleri

$P_{i,j}^t$, t zaman adımında i düğümünden j düğümüne geçiş olasılığını temsil etsin. Üç adımlı bir rasgele yürüyüşü ele aldığımızda, anonim yürüyüşün olası dizilimi **Şekil 4.8**'deki gibi formüle edilebilir.



Şekil 4.9 Üç adımlık bir komşulukta düğümleri ziyaret etme olasılıkları

Şekil 4.9'da ağaç benzeri gösterim, 3 adımlık komşuluk içerisindeki düğümlerin ziyaret edilme olasılıklarını göstermektedir. Her bir zaman adımında düğümlere ulaşma olasılıkları, her düğüm için geçiş olasılığı matrisinin bileşenlerini hesaplamak üzere toplanmaktadır. Üç adımlı anonim rasgele yürüyüş için hesaplanan olasılıklar, kavramı örneklendirmek amacıyla **Çizelge 4.1**'de sunulmuştur. Üç adımlı anonim yürüyüş için elde edilen geçiş olasılığı matrisi, **Şekil 4.10**'da gösterildiği şekilde temsil edilmektedir.

$$TPM = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ 1 & 0 & 0 & 0 \\ 0 & P_{0,1}^{(1)} & 0 & 0 \\ P_{1,0}^{(2)} & 0 & P_{1,2}^{(2)} & 0 \\ P_{2,0}^{(3)} & P_{0,1}^{(3)} + P_{2,1}^{(3)} & P_{0,2}^{(3)} & P_{2,3}^{(3)} \end{bmatrix} \begin{matrix} t = 0 \\ t = 1 \\ t = 2 \\ t = 3 \end{matrix}$$

Şekil 4.10 Her zaman adımı için geçiş olasılığı matris elemanı

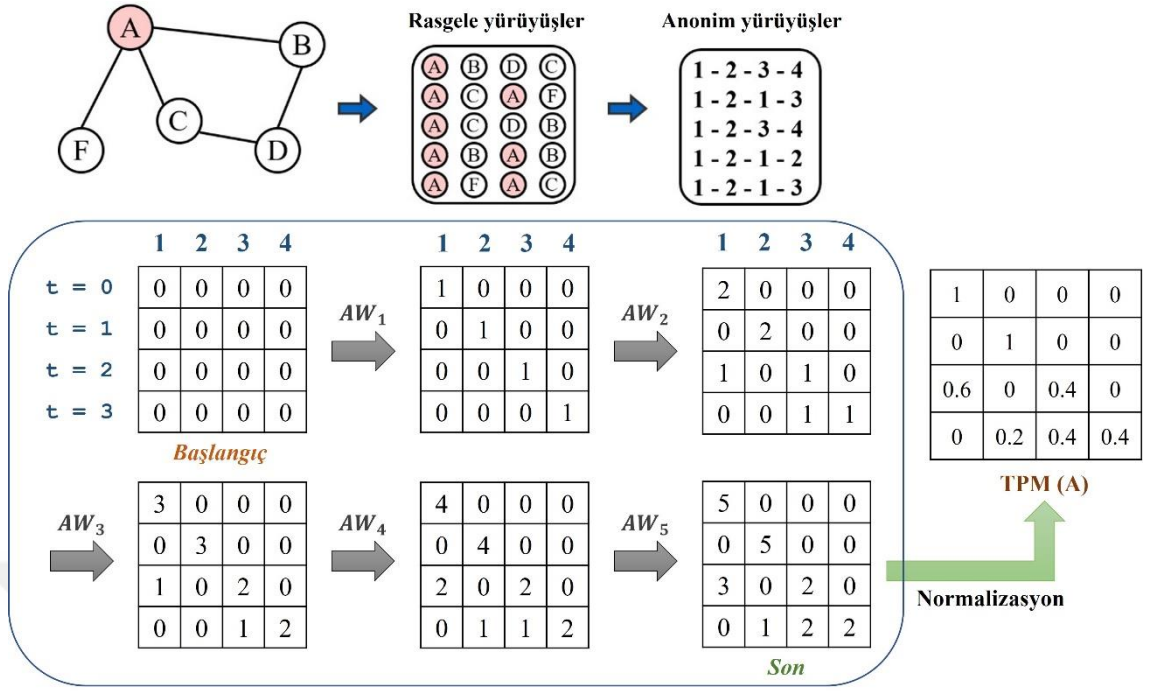
Çizelge 4.1 Üç adımlı anonim yürüyüşler kümesi için anonim rasgele yürüyüş olasılıkları

Adım		Olasılık	Düğüm
1	$P_{0,1}^{(1)}$	$\frac{1}{d_0}$	$0 \rightarrow 1$
2	$P_{1,0}^{(2)}$	$P_{0,1}^{(1)} \times \frac{1}{d_1}$	$1 \rightarrow 0$
	$P_{1,2}^{(2)}$	$P_{0,1}^{(1)} \times \frac{d_1 - 1}{d_1}$	$1 \rightarrow 2$
3	$P_{2,0}^{(3)}$	$P_{1,2}^{(2)} \times \delta_{2,0} \times \frac{1}{d_2}$	$2 \rightarrow 0$
	$P_{2,1}^{(3)}$	$P_{1,2}^{(2)} \times \frac{1}{d_2}$	$2 \rightarrow 1$
	$P_{2,3}^{(3)}$	$P_{1,2}^{(2)} \times \left[\frac{d_2 - 2}{d_2} + [1 - \delta_{2,0}] \times \frac{1}{d_2} \right]$	$2 \rightarrow 3$
	$P_{0,1}^{(3)}$	$P_{1,0}^{(2)} \times \frac{1}{d_0}$	$0 \rightarrow 1$
	$P_{0,2}^{(3)}$	$P_{1,0}^{(2)} \times \delta_{0,2} \times \frac{1}{d_0}$	$0 \rightarrow 2$

Çizelge 4.1'de üç adımlı anonim yürüyüşler kümesi için anonim rasgele yürüyüş olasılıkları hesaplanmaktadır. Teorik olasılık değerleri, her zaman adımında komşu düğümleri ziyaret etme ihtimalini temsil eder. Burada, d_i her düğümün derecesini gösterirken, $\delta_{i,j}$ ise v_i düğümü ile v_j düğümü arasındaki bağlantıyı belirten Kronecker deltasını ifade eder.

4.2.4 Geçiş olasılık matrisi hesaplanması – simülasyon

Belirli bir düğüm için geçiş olasılığı matrisi'nin oluşturulması süreci, rasgele yürüyüş üretimi, bu yürüyüşlerin anonim yürüyüşlere dönüştürülmesi, düğüm ziyaret sayılarının toplanması ve normalizasyon aşamalarını içeren sistematik bir yaklaşıma kapsamaktadır. Bu süreç, yalnızca düğümün yerel yapısal özelliklerini yakalamakla kalmaz, aynı zamanda daha geniş ağ içindeki bağlantısallığını analiz etmek için olasılıksal bir çerçeve sunar.



Şekil 4.11 A düğümü için rasgele ve anonim yürüyüşler kullanılarak geçiş olasılığı matrisi oluşturma süreci

Şekil 4.11’de A düğümü için geçiş olasılığı matrisin oluşturulmasının basit bir örneği verilmektedir. İlk adımda A düğümünden, ağın yerel komşuluğunu keşfetmek amacıyla w_l uzunluğunda rasgele yürüyüşler başlatılır. Bu üretilen rasgele yürüyüşler daha sonra anonim yürüyüşlere dönüştürülür. Düğüm kimliklerini koruyan geleneksel rasgele yürüyüşlerin aksine, anonim yürüyüşler yalnızca yürüyüş içindeki konum ve geçiş dizisine odaklanarak, gerçek düğüm kimliklerini göz ardı eder. Akabinde, bu anonim yürüyüşler, çeşitli zaman adımlarında düğüm ziyaretlerini kaydeden bir dizi matrisi güncellemek için kullanılır.

Başlangıçta, matris henüz hiçbir yürüyüş gerçekleşmediğinden tüm sıfır girişlerini içerir. Matrisin satırları ($t = 0$)’dan ($t = 3$)’e kadar olan zaman adımlarına, sütunlar ise anonim yürüyüşteki pozisyonlara (1’den 4’e kadar) karşılık gelir. Her anonim yürüyüş (AW_1 ’den AW_5 ’e kadar) matrisi kademeli olarak güncellenir. Örneğin, AW_1 matrisi, göreceli pozisyonlar arasındaki geçişlerin sayımlarını ekleyerek modifiye eder. İlave anonim yürüyüşler işlendikçe, matris sayımları biriktirmeye devam eder ve farklı zaman adımları boyunca yürüyüşteki pozisyonlar arasındaki geçişlerin sıklığını yansıtır.

Bu iteratif süreç, tüm anonim yürüyüşler dikkate alınana kadar devam ederek nihai bir matris oluşturulur. Son matris bir normalizasyon adımından geçer ve sayımlar olasılıklara dönüştürülür. Bu normalizasyon, A düğümü ile ilişkili geçiş olasılığı matrisi'ni oluşturur. Bu matris, A düğümünden başlatılan tüm rasgele yürüyüşlerin katkılarını toplayarak, A düğümünün komşuluğu içindeki genel yapısal rolünü yakalar.

Nihai $TPM(A)$ matrisi, A düğümünün yerel yapısının olasılıksal bir görünümünü sunar. Bu matrisi analiz ederek, A düğümünün komşularıyla nasıl etkileşime girdiği ve A 'dan başlayarak ağ üzerinden farklı yolları kat etme olasılıklarının neler olduğu çıkarılabilir. Bu olasılıksal temsil, hem doğrudan hem de dolaylı etkileşimleri kapsadığı için daha karmaşık ağlarda düğüm merkeziliğini, etkisini ve bağlantısallığını anlamak açısından kritik öneme sahiptir.

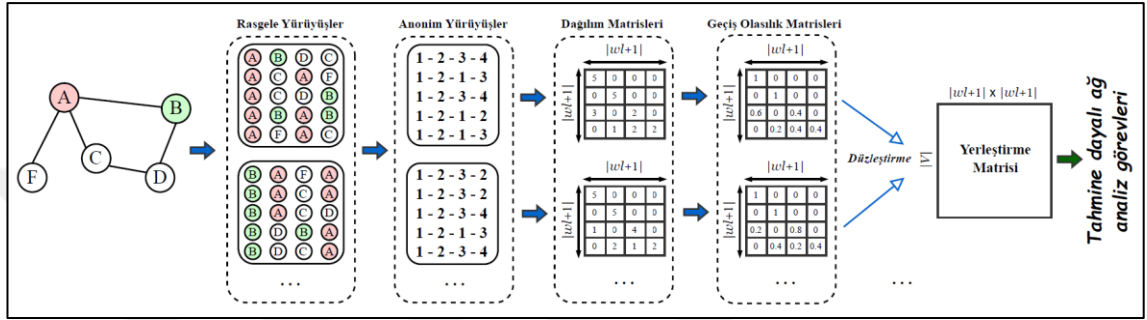
4.2.5 TPM modelinin mimarisi

Geçiş olasılığı her düğüm için elde edildikten sonra, ortaya çıkan geçiş olasılığı matrisi, ağ özelliklerini hesaplamak için çeşitli şekillerde kullanılabilir. Bu çalışmada, amaç, belirli bir düğümün yerel özelliklerinin geçiş olasılığı matrisinin tanımında ne kadar iyi korunduğunu göstermektir. Bu doğrultuda, düğüm sınıflandırma, bağlantı tahmini ve görselleştirme görevlerinde düğüm özellikleri olarak kullanılmıştır. Yerleştirme sürecinin uygulaması beş aşamadan oluşmaktadır (**Şekil 4.12**):

1. Her düğümünden başlayarak, wl uzunluğunda, nw olarak gösterilen bir rasgele yürüyüşler kümesi oluşturulur.
2. Her rasgele yürüyüş, belirli düğüm kimliklerini göz ardı ederek ve adım dizisine odaklanarak bir anonim yürüyüşe dönüştürülür.
3. Her düğüm için, $(|wl + 1| \times |wl + 1|)$ boyutunda bir dağılım matrisi hesaplanır.
4. Dağılım matrisinin normalleştirilmesinden sonra, ilk 4 adım tekrarlanarak her düğüm için $(|wl + 1| \times |wl + 1|)$ boyutunda geçiş olasılığı matrisi türetilir.
5. Her bir düğüm için ayrı ayrı oluşturulan geçiş olasılığı matrisleri, tek boyutlu vektörler haline getirilecek şekilde düzleştirilir. Bu işlem sonrasında elde edilen tüm

vektörler birleştirilerek, düğümleri temsil eden tek bir yerleştirme matrisi oluşturulur. Bu matrisin her bir satırı, bir düğüme ait yerleştirme vektörünü ifade etmektedir.

Burada wl , bir rasgele yürüyüşteki adım sayısını ifade eder. Amaç bir düğüm hakkında yerel bilgi elde etmek olduğundan, çok fazla adım seçmek yürüyüşün ağı çok derinlerine inmesine neden olabilir ve böylece yerel özelliklere odaklanmasını azaltabilir.



Şekil 4.12 Önerilen TPM modelinin ardışık düzeni

Şekil 4.13, anonim rasgele yürüyüşler kümesi oluşturma ve geçiş olasılığı matrisi elde etme sürecinin sözde kodunu göstermektedir.

Algoritma : Geçiş Olasılık Matrisi Oluşturma Yöntemi

Girdi: Ağ $G = (V, E)$

Başlangıç:

Düğüm sayısı V ;

Düğüm başına yürütiş sayısı nw ;

Yürütiş uzunluğu wl ;

Çıktı: Her düğüm $v \in V$ için düğüm temsili X_v

for v **in** V **do**

$walks_v$ 'yi boş olarak başlat

$anons_v$ 'yi boş olarak başlat

tpm_v 'yi $([t]_{|wl+1| \times |wl+1|}, t = 0)$ olarak başlat

for $i = 0$ **to** nw **do**

$w = \text{UniformlyRandomWalk}(G, v, wl)$

w yürütişünü $walks_v$ 'ye ekle

$anons_v = \text{AnonymousRandomWalk}(walks_v, v, wl)$

$k = 0$

for $step$ **in** $anons_v$ **do**

 // Anonim yürütişün her adımında ziyaret edilen düğüm sayısını hesapla

$tpm_v[k, step] += 1$

$k += 1$

$X_v = \text{normalize}(tpm_v)$

X 'i geri döndür

► Düğümler üzerinde döngü

► Rasgele yürütişler kümesi

► Anonim Rasgele yürütişler kümesi

► Geçiş olasılık matrisi

Şekil 4.13 Geçiş olasılık matrisi oluşturma yönteminin algoritması

4.3 Kılavuzlu Geçiş Olasılık Matrisi Modeli

Büyük ölçekli verilerden verimli bilgi çıkarımı, bilgisayar bilimi ve makine öğrenmesinin temel sorunlarından biri haline gelmiştir. Metin verisi, sözdizimi, anlambilim ve hatta gizli bilgilere sahip olması nedeniyle, bu veri türleri arasında ayrıcalıklı bir konuma sahiptir (Hemmatian ve Sohrabi 2019). Günümüzde, büyük miktarda metinsel verinin varlığı, metnin etkili bir şekilde temsil edilmesini, özellikle metin sınıflandırması gibi çeşitli doğal dil işleme görevlerinde kritik bir faktör haline getirmiştir. Metin sınıflandırma, doğal dil işlemenin temel görevlerinden biri olup sosyal medya analizi, duygu madenciliği ve sağlık gibi geniş bir uygulama yelpazesi sunmaktadır. Metin verilerinin işlenmesi, metin içeriğinin sayısal vektörlere dönüştürülmesi anlamına gelen metin temsili öğrenimini gerektirmektedir. Metin temsili öğrenimi, metin sınıflandırma probleminin ilk ve en temel aşamasını oluşturur. Etkili bir metin temsili algoritması, metin verilerinin tam bilgi içeriğini elde etmek için kritik bir başlangıç noktasıdır.

Literatürde çok iyi bilinen metin temsili öğrenimi modelleri olan Bag-of-Words, TF-IDF, word2vec ve GloVe, birçok NLP görevinde yaygın olarak kullanılmış ve etkinlikleri kanıtlanmıştır (Lewis vd. 2020). Bu modeller, genellikle kelimeleri bağımsız semboller olarak ele almakta ve onları ortak oluşum istatistiklerine veya sürekli vektör uzayında öğrenilen yerleştirmelere dayalı olarak temsil etmektedir. Ancak, bu yöntemler, karmaşık ilişkileri ve bağlamsal bilgileri yakalama konusunda sınırlı yeteneklere sahiptir. Özellikle, çok anlamlılık (bir kelimenin birden fazla anlam taşıması) gibi zorlukları ele almakta yetersiz kalmakta ve daha derin anlamsal anlayış gerektiren görevler için uygunlukları sınırlı kalmaktadır. Bu durum, önemli bilgilerin kaybına yol açarak, daha karmaşık görevlerde bu modellerin kullanımını sınırlandırabilmektedir. Ayrıca, dil zamanla evrim geçirip değişim gösterdiğinden, sınıflandırma gereksinimleri de dinamik bir yapıya sahiptir (He vd. 2018). Bu sebeple, metnin gizli özelliklerini derinlemesine anlayabilmek için dinamik bir öğrenme algoritmasına ihtiyaç duyulmaktadır.

Metin verileri ve ağ verileri görünüşte farklı olsa da, kelimeler arasındaki doğal ilişkiler bir metindeki kelimelerin ağ yapısının düzenlenmesini kolaylaştırmaktadır. Kelimelerin anlambilimsel özelliklerinin komşu kelimelerle olan sıkı ilişkisi göz önünde bulundurulduğunda, ağ yapıları bu ilişkileri ortaya çıkarmak için ideal bir çerçeve sunmaktadır. Bu nedenle, metnin ağ verisi olarak temsil edilmesi sürecinde etkili bir stratejinin belirlenmesi büyük önem taşımaktadır.

Önceki bölümde önerilen TPM yöntemi, bir ağın düğümleri etrafındaki yerel topolojik yapının, komşu alanlar arasındaki geçiş olasılıkları üzerinden elde edilebileceğini ortaya koymuştur. Belirli bir düğümden başlatılan anonim yürüyüşlerden elde edilen geçiş olasılıkları, bu düğümün özellik vektörü olarak kullanılan bir geçiş olasılığı matrisi oluşturmak için kullanılmıştır. Yöntem, etkili bir öznitellik yerleştirme yöntemi olarak başarısını kanıtlamıştır. Bu bölümde, metin verilerinin incelenmesinde TPM yöntemi benimsenmiştir. Anlamlı cümlelerin ağ yapısını esas alan kılavuzlu geçiş olasılık matrisi (GTPM) adını verdiğimiz yeni bir metin yerleştirme yöntemi önerilmiştir. Algoritmanın tasarımı, metin verilerinin gizli içeriğini, sözdizimsel ve anlamsal unsurları bir araya getirerek hem yerel bağlamı hem de kelimeler arasındaki küresel ilişkileri dikkate alan bir yerleştirme vektörü oluşturmayı amaçlamaktadır.

Önerilen yöntemin temelinde, metin dokümanlarını oluşturan verilerden dilin temel kurallarını çıkarmak yatmaktadır. Bu amaçla geliştirilen rasgele yürüyüş tabanlı yerleştirme modeli, kelimelerin anlamsal özelliklerini evrensel bir kelime ağı (kelime uzayı) oluşturarak ortaya çıkarmayı hedeflemektedir. Çalışmada, iki adımlı bir öznitellik çıkarma algoritması önerilmiştir. Bu adımlar şunlardır:

- Evrensel ağırlıklı bir kelime ağının oluşturulması.
- Evrensel kelime ağı kullanılarak her bir kelime için yerleştirme vektörlerinin hesaplanması.

4.3.1 Evrensel ağırlıklı kelime ağı

Farklı sınıflara ait dokümanlara sahip olduğumuzu varsayalım (**Çizelge 4.2**). Burada, w_i cümlelerdeki kelimeleri ve d_i ise dokümanları temsil etmektedir. Bu bağlamda, kelimeler başlangıçta aralarında herhangi bir bağlantısı bulunmayan ağ düğümlerini temsil eder. Ancak, belgeler işlendikçe, cümlelerdeki ifadeler temel alınarak kelimeler arasında bağlantılar oluşmaya başlar ve bu süreç sonunda, belgelerin içerdiği kelime yapısına dayalı olarak bağlantılı bir ağ meydana gelir. Örneğin, dört dokümanın ifadelerinden oluşan altı kelimelik bir dil düşünüldüğünde, bu altı kelime arasında, dokümanlar işlenmeden önce herhangi bir bağlantı bulunmamaktadır. Dolayısıyla bağlantı matrisi A_{ij} şu şekilde tanımlanır: $a_{ij} = 0$, $i = 1, 2, \dots, 6$, $j = 1, 2, \dots, 6$.

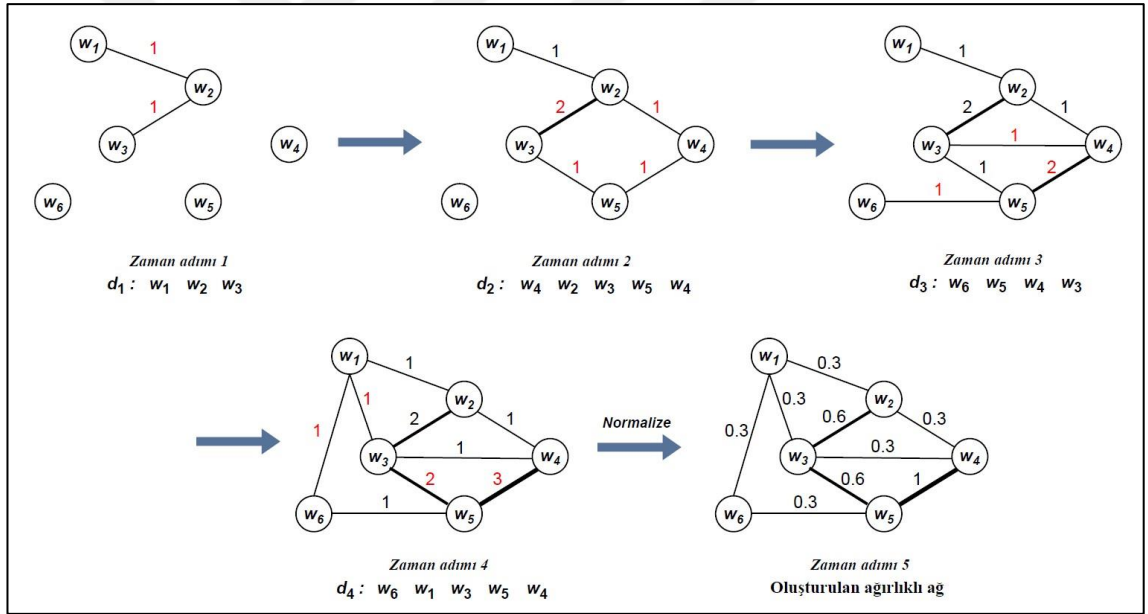
Çizelge 4.2 Farklı sınıflara ait örnek dokümanlar

Döküman	Döküman Türü
$w_1 w_2 w_3$	d_1
$w_4 w_2 w_3 w_5 w_4$	d_2
$w_6 w_5 w_4 w_3$	d_3
$w_6 w_1 w_3 w_5 w_4$	d_4

$$A_{ij} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j} \\ a_{21} & a_{22} & \dots & a_{2j} \\ \dots & \dots & \dots & \dots \\ a_{i1} & a_{i2} & \dots & a_{ij} \end{bmatrix}$$

$$a_{ij} = \begin{cases} a_{ij} = a_{ij} + 1 & i \text{ ve } j \text{ arasında bağlantı varsa, kenar değerini artır} \\ a_{ij} = 1 & \text{bağlantı yoksa, kenarı ağıza ekle} \end{cases}$$

Metinde iki kelime (i ve j) ardışık olarak yer aldığında, matrisin ilgili elemanı a_{ij} bir artırılır. Sürecin sonunda, a_{ij} değeri, her cümlede i kelimesinin ardışık komşusu j ile birlikte kaç kez geçtiğini gösterir. Bu şekilde, bağlantı matrisi, dokümanlardaki kelimelerin birbirleriyle olan ardışık ilişkilerini yansıtan bir yapıya dönüşür.



Şekil 4.14 Dört doküman için kelime ağıının oluşturulması

Şekil 4.14, dört farklı kategoriye ait dört küçük doküman kullanılarak ağırlıklı bir ağ oluşturma süreci göstermektedir. Her adımda, düğümler arasındaki kenarlar aynı zaman adımıdaki eş zamanlılıklarına bağlı olarak kurulur veya güçlendirilir ve kenar ağırlıkları artırılır. 1. adımda, ilk doküman tanıtılarak üç düğümden oluşan bir ağ oluşturulur. 2. adımda, ağı iki yeni düğüm (w_4 ve w_5) eklenir ve w_2 ile w_3 ardışık geldikleri için kenar değerleri artırılır. Son adımda ise, tüm kenarlar maksimum kenar değeri ile normalize

edilir. Kenar ağırlıkları, süreç ilerledikçe biriken eş zamanlılık sıklıklarını temsil etmektedir. Tüm kenarları içeren normalleştirme işlemi, konuya özgü karakteristik kelimelerin merkez kelimeler olduğu ağırlıklı bir ağ oluşturulur. Bu aşamada, daha fazla verinin çıkarılması için önceden belirlenmiş dil sınırları dahilindeki tüm bağlantı bilgilerini içeren bir "evrensel ağ" elde edilir. Bu süreç, düğümler arasındaki ilişkilerin gücünü, bunların zamansal etkileşimlerine dayanarak etkili bir şekilde yakalar. Ağ oluşturma stratejisi (topolojik bilgiler) dilin kuralları ve kelimeler arasındaki ilişkiler bağlantıyı sağlar. Ayrıca, bu süreç dinamiktir: düğüman kümesine yeni düğüman eklemek, ağı bağlantılarını genişletir veya geliştirir.

4.3.2 Evrensel kelime ağı aracılığıyla kelime yerleştirilmesi

Kelime yerleştirme süreci, evrensel kelime ağını temsil eden bir komşuluk matrisinin oluşturulmasıyla başlar. Ardından, kelime ağı üzerinde rasgele yürüyüşler gerçekleştirilerek, her kelimenin çevresindeki komşularıyla olan bağlantılarına dair bilgi toplanır ve bu şekilde cümle benzeri diziler oluşturulur. Her yürüyüş önceden belirlenmiş bir uzunlukta gerçekleştirilir ve her düğüm için n_w adet yürüyüşün sonunda elde edilen ziyaret sıklıkları, o kelimenin belge içindeki ilişkili önemini ve diğer kelimelerle olan anlamsal ilişkilerini yansıtır.

Ağırlıksız bir ağda, bir gezginin i bitişik komşularından herhangi birine j uğrama olasılığı, tüm komşular için eşdeğerdir. Bu eşit olasılık dağılımı, bağlantıların farklı önem derecelerini göz önünde bulundurmaz. Bunun yerine, bağlantıların gerçek önem dereceleri ağırlıklar tarafından yansıtılır ve bu ağırlıklar normalize edilerek ağ içindeki rasgele yürüyüşlerin yönlendirilmesinde kullanılır. Önerilen yöntemde, rasgele yürüyüşlerin olasılıkları şu şekilde tanımlanmaktadır:

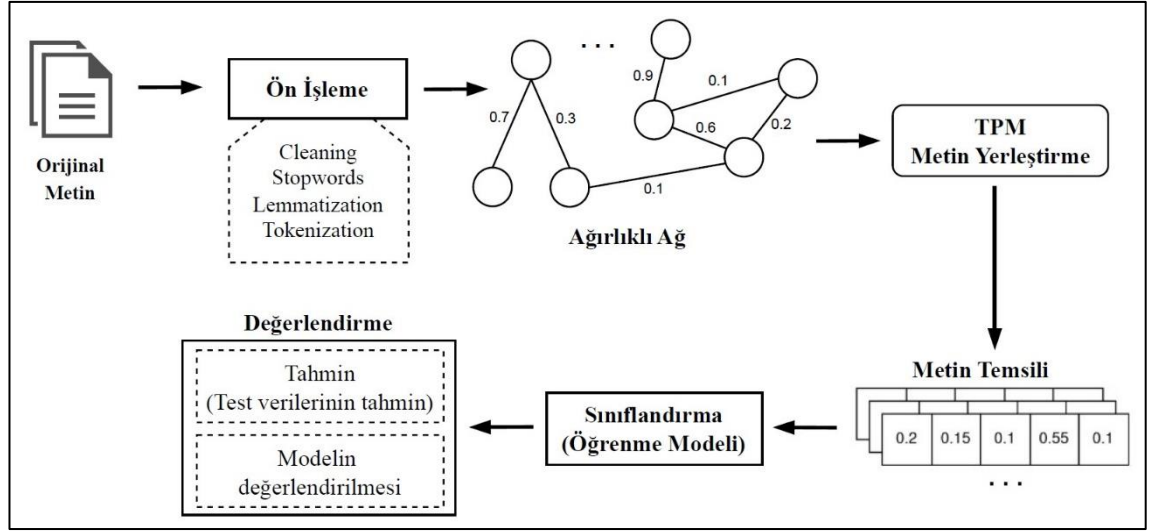
$$p_{i,j} = \frac{\omega_{ij}}{\sum_{j \in N(i)} \omega_{ij}} \quad (4.4)$$

Burada ω_{ij} , i ve j düğümlerini birbirine bağlayan kenarın ağırlığını ifade eder. Paydada yer alan $\sum_{j \in N(i)} \omega_{ij}$ ise normalizasyon faktörünü temsil eder.

Rasgele bir yürüyüş esnasında bir komşuyu ziyaret etme olasılığı, o komşuyla ilişkili kenar ağırlığından etkilenir. Daha güçlü bir bağlantı, seçilme olasılığını artırır. Sonuç olarak, birlikte görülme sıklığı, seçim sürecini daha yakın ilişkiye sahip komşulara doğru yönlendirir. Bu nedenle, yerleştirme vektörlerinin sadece kelimeler arasındaki ilişkileri değil, aynı zamanda topolojik yapıdan elde edilen komşuluk bilgisini de içermesi öngörülmektedir.

Bölüm 4.2’de, önerdiğimiz TPM modelinin, anonim rasgele yürüyüşler kullanarak yenilikçi bir yerleştirme tekniği sunduğunu ortaya koymuştur. Bu teknik, yerel bağlantısallık ilişkilerinin bir geçiş olasılığı matrisi olarak temsil edilebileceğini göstermektedir. Söz konusu matris, düğümlerin kimliklerine dayanılmaksızın, komşuluk topolojisinin yapısını yansıtmaktadır. Bu bağlamda, önceki aşamada oluşturulan rasgele yürüyüşler anonim yürüyüşlere dönüştürülmüş olup, anonim yürüyüşlerde kodlanan bilgi, bitişik düğümler arasındaki ilişkileri yansıtmaktadır. Sonrasında, TPM yönteminde olduğu gibi anonim yürüyüşler bir geçiş olasılığı matrisine dönüştürülmekte ve elde edilen bu olasılık matrisi, düğümlerin komşuları ile olan ilişkilerinin topolojik yapısını temsil etmektedir.

Bu süreçte, yürüyüşlerin miktarı nw ve uzunluğu wl , ağ içerisindeki hem yerel hem de küresel bilgilerin entegrasyonunu kolaylaştırmaktadır. Ağ oluşturma yöntemi, topolojik ayrıntıları dilbilimsel kurallar ve sözcük ilişkileriyle etkili bir şekilde bağdaştırmaktadır. Yürüyüşlerin sayısını veya uzunluklarını artırarak, daha hassas komşuluk bilgisi toplanabilmekte, böylelikle bu parametrelerin optimizasyonu yoluyla yüksek kaliteli yerleştirme vektörlerinin üretilmesi mümkün kılınmaktadır. **Şekil 4.15**, önerilen GTPM yönteminin ardışık düzenini göstermektedir.



Şekil 4.15 Önerilen GTPM modelinin ardışık düzeni

5. BULGULAR ve TARTIŞMA

Bu bölümde, tez kapsamında geliştirilen düğüm yerleştirme modelleriyle elde edilen bulgular ve bu bulguların analizi sunulmaktadır. Önerilen modellerin değerlendirilmesinde kullanılan metrikler, veri setleri ve temel modeller hakkında ayrıntılı bilgi verilmiş olup, elde edilen sonuçlar literatürde mevcut yöntemlerin sonuçlarıyla karşılaştırılarak tartışılmıştır.

5.1 Değerlendirme Metrikleri

Tez kapsamında değerlendirme metrikleri olarak, Doğruluk, Macro-F1 ve Micro-F1 skorları kullanılmıştır. Bu metriklerin yüksek değerler alması, ilgili ağ yerleştirme tekniğinin üstün bir performans gösterdiğini ortaya koymaktadır.

Doğruluk metriği: Bir sınıflandırma modelinin genel olarak ne kadar doğru tahmin yaptığını ölçer. Doğruluk, doğru pozitif ve doğru negatif tahminlerin, tüm tahminler içindeki oranını temsil eder.

$$\text{Doğruluk} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5.1)$$

Burada:

- Doğru Pozitif (True Positive - TP): Modelin, doğru şekilde pozitif olarak tahmin ettiği örnekler.
- Doğru Negatif (True Negative - TN): Modelin, doğru şekilde negatif olarak tahmin ettiği örnekler.
- Yanlış Pozitif (False Positive - FP): Modelin, yanlış şekilde pozitif olarak tahmin ettiği örnekler.
- Yanlış Negatif (False Negative - FN): Modelin, yanlış şekilde negatif olarak tahmin ettiği örnekler.

Macro-F1 skoru: Her sınıf için ayrı ayrı F1 skorları hesaplanarak, bu skorların ortalamasının alınmasıyla elde edilir. Yani, her sınıfın performansı eşit ağırlıkla değerlendirilir. Bu hesaplama, sınıflar arasındaki dengesizlikleri göz ardı eder ve her sınıfın katkısını eşit tutar (Zaki ve Meira 2014).

Micro-F1 skoru: Tüm sınıflardaki TP, FP ve FN sayılarının toplamlarına dayalı olarak hesaplanır. Bu yöntem, örnek sayısının fazla olduğu sınıfların daha fazla ağırlık taşıdığı bir ölçüt sağlar ve dolayısıyla sınıf dengesizliklerini dikkate alır (Zaki ve Meira 2014).

5.2 Veri Setler

Tez kapsamında önerilen modellerin değerlendirilmesinde erişime açık çeşitli ağ tabanlı ve metin tabanlı veri setleri kullanılmıştır. Derece tabanlı rasgele yürüyüş modelinin performansı, Zachary Karate Club, CORA ve CiteSeer veri setleri ile test edilmiştir. Geçiş olasılık matrisi modeli ise CORA, CiteSeer ve PubMed veri setleri kullanılarak değerlendirilmiştir. Kılavuzlu geçiş olasılık matrisi modeli için ise SST-2, MR, CoLA, Ohsumed, Reuters ve 20NG veri setlerinden yararlanılmıştır. Derece tabanlı rasgele yürüyüş ve geçiş olasılık matrisi modellerinde kullanılan ağ tabanlı veri setlerinin istatistikleri **Çizelge 5.1**'de sunulmuştur. Kılavuzlu geçiş olasılık matrisi modelinde kullanılan metin tabanlı veri setlerinin istatistikleri ise **Çizelge 5.2**'de verilmiştir. **Çizelge 5.2**, belgelerin sayısı, eğitim seti sayısı, doğrulama test sayısı, test seti sayısı, sınıf sayısı ve ortalama kelime uzunluğu içeren bilgileri göstermektedir.

Çizelge 5.1 Ağ tabanlı veri setlerinin istatistikleri

	Zachary Karate Club	CORA	CiteSeer	PubMed
Düğüm sayısı	34	2.708	3.327	19.717
Kenar sayısı	78	5.429	4.732	44.338
Sınıf sayısı	2	7	6	3

Çizelge 5.2 Metin tabanlı veri setlerinin istatistikleri

	SST-2	MR	CoLA	Ohsumed	Reuters	20NG
Belge sayısı	9.613	10.662	10.657	7.400	11.228	18.846
Eğitim sayısı	7.792	7.108	9.594	3.357	8.982	11.314
Test sayısı	1.821	3.554	1.063	4.043	2.246	7.532
Sınıf sayısı	2	2	2	23	46	20
Ortalama kelime uzunluğu	19,3	21	7,7	79,49	60,28	221,26

Zachary Karate Club veri seti, bir üniversite karate kulübü üyelerinin oluşturduğu sosyal ağı modellemektedir. Bu veri seti, yönsüz ve ağırlıksız bir ağ olarak ifade edilebilmektedir. Ağ, 34 düğümden (karate kulübü üyeleri) oluşmaktadır; bunlardan biri eğitmen, diğeri ise yönetici pozisyonundadır. Ayrıca ağ, kulüp üyelerinin kulüp dışındaki etkileşimlerini temsil eden 78 kenardan oluşmaktadır (Girvan ve Newman 2002).

CORA veri seti, akademik yayınları düğümler ve bu yayınlar arasındaki atıfları bağlantılar olarak ele almaktadır. Eğer A yayını B yayınına atıfta bulunuyorsa, ağ üzerinde A ve B arasında bir kenar bulunur. Cora veri seti, yedi farklı sınıftan birine ait toplam 2.708 bilimsel yayını içermektedir. Atıf ağı, bu yayınlar arasındaki 5.429 bağlantıyı temsil eder. Veri setindeki her bir yayın, bir sözlükteki belirli kelimelerin varlığını veya yokluğunu gösteren 0/1 değerli bir kelime vektörüyle tanımlanmaktadır. Sözlük, 1.433 farklı kelimedenden oluşmaktadır (McCallum vd. 2000).

CiteSeer veri seti, altı farklı sınıftan birine ait 3.327 bilimsel yayını içermektedir. Atıf ağı, bu yayınlar arasındaki 4.732 bağlantıyı kapsar. Veri setindeki her yayın, bir sözlükteki belirli kelimelerin varlığını veya yokluğunu gösteren 0/1 değerli bir kelime vektörüyle tanımlanmaktadır. Sözlük, 3.703 farklı kelimedenden oluşmaktadır (Giles vd. 1998).

PubMed veri seti, PubMed veritabanından alınan ve diyabetle ilgili üç farklı sınıfa ayrılmış toplam 19.717 bilimsel yayını içermektedir. Atıf ağı, bu yayınlar arasındaki 44.338 bağlantıyı kapsar. Veri setindeki her yayın, 500 farklı kelimedenden oluşan bir sözlük kullanılarak, TF-IDF ağırlıklı kelime vektörü ile temsil edilmektedir (Sen vd. 2008).

SST-2, sinema eleştirilerinden alınan cümleler ve bu cümlelere ilişkin insan tarafından yapılan duygu durumu etiketlemelerinden oluşan iki sınıflı bir metin veri setidir. Çalışmamızda, 7.792 eğitim ve 1.821 test örneği içeren herkese açık versiyonu kullanılmıştır. Veri setinde toplam 4.963 olumlu ve 4.650 olumsuz eleştiri yer almakta olup, eleştirilerin ortalama uzunluğu 19,3 kelimedir (Socher vd. 2013).

MR veri seti, ikili duygu sınıflandırması amacıyla tasarlanmış bir sinema eleştirisi metin veri setidir. Bu veri setindeki her bir değerlendirme, yalnızca tek bir cümleden oluşur ve her cümle olumlu veya olumsuz olarak etiketlenmiştir. Veri seti, toplamda 5.331 olumlu ve 5.331 olumsuz sinema eleştirisi içermektedir (Tang vd. 2015a).

CoLA veri seti, New York Üniversitesi tarafından yayımlanan bir dilbilgisi veri setidir ve esas olarak verilen bir cümlelerin dilbilgisi açısından doğru olup olmadığını değerlendirmeyi amaçlamaktadır. Bu veri seti, tekil cümlelerin ikili sınıflandırması kategorisine girmektedir. Tez kapsamında, 9.594 eğitim ve 1.063 test örneğinden oluşan, toplamda 10.657 örneği içeren açık sürüm kullanılmıştır. Bu örneklerin 6744'ü dilbilgisi açısından doğru (olumlu), 2.850'si ise yanlış (olumsuz) olarak etiketlenmiştir. Cümlelerin ortalama uzunluğu 7,7 kelimedir (Warstadt 2019).

Ohsumed veri seti, Amerikan Ulusal Tıp Kütüphanesi tarafından yönetilen MEDLINE veritabanının bir parçası olup, tıbbi literatürün önemli eserlerini içermektedir. Bu veri seti, her bir özeti, ilgili hastalık kategorilerine göre sınıflandırılmış metinlerden oluşmaktadır. Toplamda 7.400 metin içeren veri seti, 23 farklı hastalık kategorisini kapsamaktadır (Yao vd. 2019).

20NG veri seti, çeşitli haber gruplarından derlenmiş metinlerden oluşan bir metin veri setidir ve 20 farklı kategoriye ayrılmış uzun yazıları içermektedir. Toplamda 18.846 doküman barındıran bu veri seti, 11.314 eğitim dokümanı ve 7.532 test dokümanından oluşmaktadır (Lang 1995).

Reuters (Keras) veri seti, Keras kütüphanesi tarafından sağlanan bir metin sınıflandırma verisidir. Bu veri seti, Reuters haber ajansından alınan ve 46 farklı konu başlığı altında etiketlenmiş 11.228 haberden oluşmaktadır. Koleksiyon, klasik Reuters-21578 (Lewis ve Ringuette 1994) veri setinin ayrıştırılması ve ön işlemden geçirilmesi sonucunda elde edilmiştir (Chollet 2021).

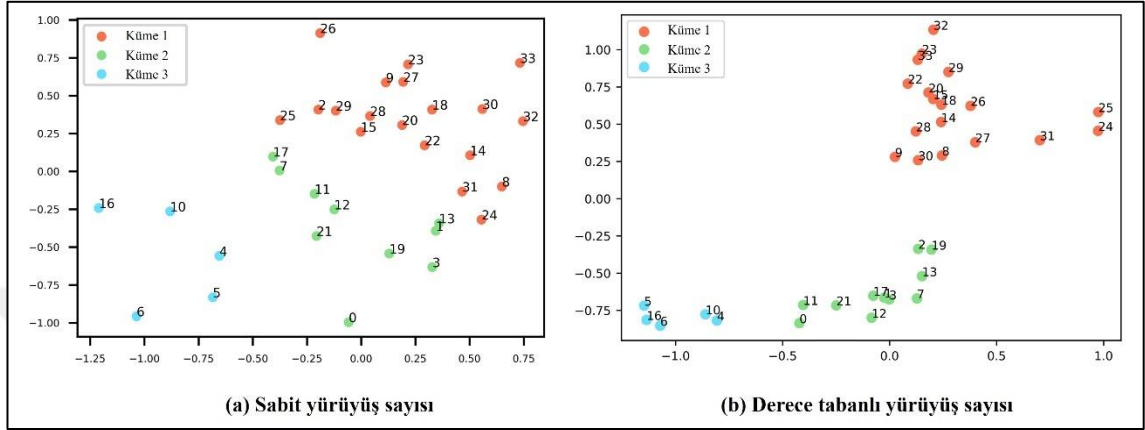
5.3 Derece Tabanlı Rasgele Yürüyüş Modelinin Sonuçları

5.3.1 Boyutsal indirgeme ile topluluk yapılarının incelenmesi

Zachary Karate Club ağı kullanılarak, sabit ve derece tabanlı modeller için yerleştirme vektörleri hesaplanmıştır. Bu hesaplamada, 10 sabit yürüyüş uzunluğu, 5 pencere boyutu ve 32 boyutlu yerleştirme vektörleri kullanılmıştır. Boyut indirgeme ve kümeleme işlemleri için Python'un NetworkX paketindeki çok boyutlu ölçekleme (MDS) ve k-ortalamlar (K-Means) algoritmaları kullanılmıştır. Yerleştirme vektörleri, topluluk tespiti amacıyla k-ortalamlar algoritmasına uygulanmak üzere iki boyutlu vektörlere dönüştürülmüştür. Yerleştirme kalitesi, yüksek ve düşük boyutlu vektör uzayları arasındaki benzerliğin korunmasıyla doğrulanmıştır. **Şekil 5.1**, toplulukları ve bunlara ait düğümleri göstermektedir. Şekildeki mesafeler ve renkler, kümelenmeyi göstermektedir. **Şekil 5.1.a**'da her düğüm için sabit 40 rasgele yürüyüş kullanılmış, **Şekil 5.1.b**'de ise her düğüm için $5 \times$ derece sayısı kadar rasgele yürüyüş başlatılmıştır

Derece tabanlı senaryoda, topluluk ayrımı için her derece başına 5 yürüyüş (toplam 780 yürüyüş) yeterli olurken, sabit rasgele yürüyüş modelinde aynı sonuca ulaşmak için her düğüm başına 40 yürüyüş (toplam 1.360 yürüyüş) gerekmektedir. **Şekil 5.1**, küçük ağlarda bile derece tabanlı rasgele yürüyüş yaklaşımının daha az hesaplama gerektirdiğini

göstermektedir. Ağ büyüdükçe, ölçekten bağımsız ağların karakteristik özelliği olan güç yasası dağılımı, düşük dereceli düğüm sayısında artışa neden olmakta ve bu da derece tabanlı yaklaşımın büyük ölçekli ağlar üzerindeki avantajlarını daha da belirgin hale getirmektedir.



Şekil 5.1 Zachary Karate Club ağı kullanılarak iki boyutlu düğüm temsillerinin görselleştirilmesi

5.3.2 CORA ve CiteSeer veri setleri ile elde edilen sonuçlar

CORA veri seti 2.708 düğüm ve 5.429 kenardan, CiteSeer veri seti ise 3.327 düğüm ve 4.732 kenardan oluşmaktadır (**Çizelge 5.1**). Bu veri kümeleri kullanılarak, hem derece tabanlı hem de sabit sayıda rasgele yürüyüş yöntemleriyle yerleştirme vektörleri oluşturulmuştur. Bu yerleştirme vektörleri düğüm sınıflandırması ve bağlantı tahmini görevleri için kullanılmıştır. Tahminler için scikit-learn Python paketinin LogisticRegressionCV fonksiyonunu (Pedregosa 2011) kullanılmıştır. Bağlantı yerleştirme vektörleri, düğüm yerleştirme vektörlerinden Hadamard, L1, L2 ve Ortalama algoritmaları (Grover ve Leskovec 2016) kullanılarak oluşturulmuştur. Test çalışmaları sırasında, Hadamard tekniğinin en iyi bağlantı tahmin puanını verdiği gözlemlenmiştir. Bu çalışmada sunulan tüm bağlantı tahmini hesaplamalarında Hadamard operatörü kullanılmıştır. Bu deneysel çalışmada, temel odak noktası doğruluk ile rasgele yürüyüşlerin sayısı ve uzunlukları arasındaki ilişkileri gözlemlemek olmuştur.

Çizelge 5.3 Derece tabanlı node2vec ve orijinal node2vec yöntemlerinin Cora veri seti üzerindeki düğüm sınıflandırma performanslarının özeti

Yürüyüş sayısı	Toplam yürüyüş sayısı	Yürüyüş sayısında azalma (%)	Doğruluk (%)	Doğruluk (azalma / kazanç)
$1 \times d_i$	10.138	79,6	80,3	-4,9
$2 \times d_i$	20.279	59	83,0	-1,9
$3 \times d_i$	30.414	38,8	85,7	+0,8
$4 \times d_i$	40.552	18,4	85,0	+0,1
Sabit ($20 \times v_i$)	49.700		84,9	

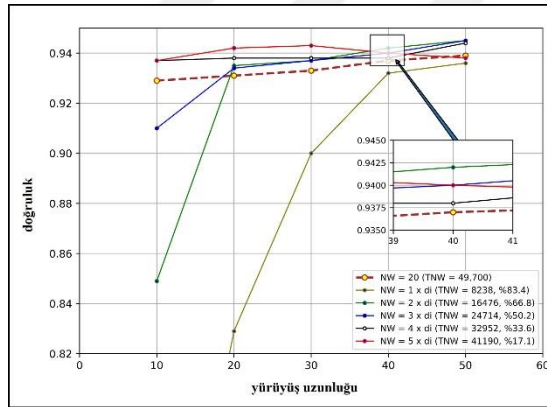
Çizelge 5.4 Derece tabanlı node2vec ve orijinal node2vec yöntemlerinin CiteSeer veri seti üzerindeki düğüm sınıflandırma performanslarının özeti

Yürüyüş sayısı	Toplam yürüyüş sayısı	Yürüyüş sayısında azalma (%)	Doğruluk oranı (%)	Doğruluk (azalma / kazanç)
$1 \times d_i$	7.388	82,49	72,6	-3,4
$2 \times d_i$	14.776	64,98	75,5	-0,5
$3 \times d_i$	22.164	47,47	76,5	+0,5
$4 \times d_i$	29.552	29,97	76,9	+0,9
$5 \times d_i$	36.940	12,46	76,7	+0,7
Sabit ($20 \times v_i$)	42.200		76,0	

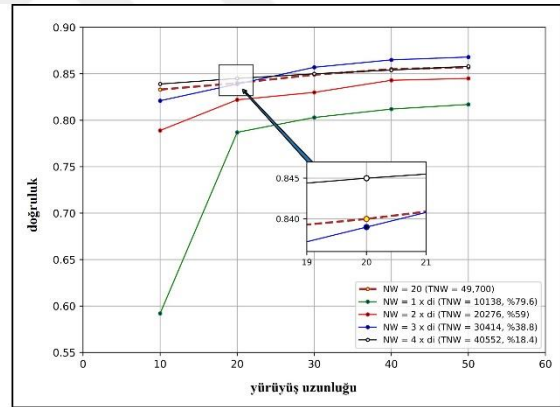
İki parametre (yürüyüşlerin sayısı ve uzunluğu) ağın topolojik yapısının yerleştirme üzerine haritalanmasında belirleyici rol oynamaktadır. Başlangıçta, 30 adım uzunluğundaki sabit yürüyüşler kullanılarak, toplam yürüyüş sayısındaki değişikliklere bağlı olarak doğruluk değerlerindeki değişim, **Çizelge 5.3** ve **Çizelge 5.4**'te ayrıntılı olarak incelenmiştir. Bu çizelgeler, sırasıyla CORA ve CiteSeer veri setleri için düğüm sınıflandırma doğruluk değerlerini, toplam yürüyüş sayısının göreceli yüzdesini ve düğüm başına (v_i) sabit 20 yürüyüş senaryosuyla karşılaştırıldığında elde edilen doğruluk kazanımlarını göstermektedir. Sonuçlar, yürüyüş sayısının artmasıyla doğruluk değerlerinde bir iyileşme olduğunu göstermektedir. Önerilen derece tabanlı yaklaşımda, doğruluk değerleri, derece başına bir yürüyüşten itibaren artmaya başlamakta ve derece

başına üç yürüyüşte bir platoya ulaşmaktadır. Bu plato değeri, düğüm başına sabit 20 yürüyüş kullanılarak elde edilen doğruluk değeriyle de örtüşmektedir.

Çizelge 5.3 ve **Çizelge 5.4** incelendiğinde, önerilen derece tabanlı yöntemin, doğruluk oranı orijinal node2vec algoritmayla kıyaslanabilir bir değere ulaştıktan sonra dahi artmaya devam ettiği gözlemlenmektedir. Örneğin, CORA veri seti için **Çizelge 5.3**, orijinal node2vec algoritma (düğüm başına sabit 20 yürüyüş ile) %84,9'luk bir doğruluk oranına ulaşmaktadır. Derece tabanlı yaklaşım ise aynı doğruluk oranına, her düğüm için derece başına yalnızca 2 ila 3 yürüyüşle ulaşmakta, bu da yürüyüş sayısında %59 ila %38,8'lik bir azalmayı temsil etmektedir. Benzer şekilde, CiteSeer veri setinde (**Çizelge 5.4**), düğüm başına 20 sabit yürüyüş yapan orijinal node2vec yöntem %76'lık bir doğruluk oranı sağlamaktadır. Derece tabanlı yaklaşım bu doğruluk oranına her düğüm için derece başına 2 ila 3 yürüyüşle erişmekte, bu da hesaplama verimliliğinde yaklaşık %64,98 ila %47,74'lük bir artışa yol açmaktadır.



(a) Düğüm sınıflandırması



(b) Bağlantı tahmini

Şekil 5.2 CORA veri setinde, rasgele yürüyüş sayısı ve yürüyüş uzunluğunun düğüm sınıflandırması ve bağlantı tahmini üzerindeki doğruluğa etkileri

Rasgele yürüyüşler kullanılarak cümle benzeri yapılar oluşturulurken, iki temel parametre olan yürüyüş sayısı ve yürüyüş uzunluğu (yani, her yürüyüşteki adım sayısı) büyük önem taşımaktadır. Bu parametrelerin doğru bir şekilde optimize edilmesi, yöntemin doğruluğunu artırırken hesaplama maliyetlerini düşürmeye yardımcı olmaktadır. **Şekil 5.2** düğüm sınıflandırması (**Şekil 5.2.a**) ve bağlantı tahmini (**Şekil 5.2.b**) örneklerinde

önerilen derece tabanlı yaklaşım ile orijinal node2vec algoritmasını karşılaştırarak bu durumu göstermektedir. Burada CORA veri seti kullanılarak yürüyüş sayısı, yürüyüş uzunluğu ve doğruluk arasındaki ilişki gösterilmektedir. Deneyde, 10 ile 50 adım arasında değişen yürüyüş uzunlukları incelenmiştir. Her alt şekil, düğüm başına artan rasgele yürüyüş sayısını ve sabit 20 yürüyüş sayısını temsil eden grafikler içermektedir. Grafikteki çizgiler, düğüm başına yürüyüş sayısındaki artışa göre sıralanmıştır. Belirli bir noktada, derece tabanlı yöntem ile sabit yürüyüş sayısının doğruluğu birbirine yaklaşmaktadır. Genel olarak, yürüyüş uzunluğu arttıkça doğruluk da artmakta, ancak belirli bir uzunluktan sonra bu artış durmaktadır. Ayrıca, yürüyüş sayısındaki artış, kısa yürüyüşlerde bile doğruluğun hızla plato seviyelerine ulaşmasını sağlamaktadır. Benzer eğilimler CiteSeer veri setinde de gözlemlenmiştir.

Çizelge 5.5 ve Çizelge 5.6, CiteSeer veri kümesi için düğüm sınıflandırması ve bağlantı tahmini verilerini sunmaktadır. Yürüyüş uzunluğuna 20 adımda başladığında, derece tabanlı yaklaşım 3 ila 4 yürüyüş başına yüksek doğruluk değerlerine ulaşmaktadır. Derece tabanlı ve sabit yürüyüş sayısı gerektiren yaklaşım (orijinal node2vec) için yapılan toplam yürüyüş sayısı arasındaki oran, derece tabanlı rasgele yürüyüş yaklaşımının verimliliğini göstermektedir. **Çizelge 5.5 ve Çizelge 5.6**'daki verilere göre, derece tabanlı yaklaşım, düğüm başına sabit sayıda yürüyüşle elde edilen doğruluklara benzer doğrulukları sağlamak için yaklaşık %50 daha az yürüyüş gerektirdiğini göstermektedir.

Çizelge 5.5 CiteSeer veri setinde derece tabanlı node2vec ve orijinal node2vec için düğüm sınıflandırması açısından doğruluk değerleri (%)

Yürüyüş sayısı	Yürüyüş uzunluğu				
	10	20	30	40	50
$1 \times d_i$	52,3	67,0	72,6	73,3	73,7
$2 \times d$	69,9	74,5	75,5	75,5	76,8
$3 \times d_i$	72,8	76,0	76,5	76,9	77,1
$4 \times d_i$	73,2	76,3	76,9	77,5	78,6
$5 \times d_i$	74,6	76,2	76,7	76,7	77,2
Sabit ($20 \times v_i$)	75,1	75,8	76	76,7	77,2

Çizelge 5.6 CiteSeer veri setinde derece tabanlı node2vec ve orijinal node2vec için bağlantı tahmini açısından doğruluk değerleri (%)

Yürüyüş sayısı	Yürüyüş uzunluğu				
	10	20	30	40	50
$1 \times d_i$	76,1	81,7	88,3	92,0	94,0
$2 \times d_i$	84,9	93,6	95,1	95,1	95,5
$3 \times d_i$	93,5	94,9	95,6	96,4	96,6
$4 \times d_i$	95,4	96,4	96,3	96,5	96,3
$5 \times d_i$	96,2	96,4	96,5	96,2	96,1
$6 \times d_i$	96,2	96,2	96,2	96,2	95,9
Sabit ($20 \times v_i$)	94,7	95,9	96,0	96,0	96,2

5.4 Geçiş Olasılık Matrisi Modelinin Sonuçları

Önerdiğimiz TPM modelinin performansı, literatürde güncel ve iyi performans sergileyen 6 düğüm yerleştirme yöntemi (DeepWalk (Perozzi vd. 2014), node2vec (Grover ve Leskovec 2016), LINE (Tang vd. 2015b), GCN (Kipf ve Welling 2016), GraphSAGE (Hamilton vd. 2017a) ve GAT (Velickovic vd. 2017)) ile karşılaştırılmıştır. Bu karşılaştırmalar, düğüm sınıflandırması, bağlantı tahmini ve görselleştirme gibi ağ analiz görevleri üzerinde gerçekleştirilmiştir. TPM modelinin, özellikle doğruluk ve verimlilik açısından bu temel yöntemlerle rekabet ettiği ve birçok durumda üstün performans gösterdiği gözlemlenmiştir. Literatürdeki çalışmalar, bu temel modellerin testlerimizde kullanılan veri setlerinde en güncel yerleştirme performansını sergilemiştir.

Parametre ayarları açısından, DeepWalk ve node2vec için düğüm başına yürüyüş sayısı, yürüyüş uzunluğu ve pencere boyutu sırasıyla 10, 80 ve 30 olarak belirlenmiştir. Ayrıca, node2vec'te $p = 2$ ve $q = 1$ ayarlarının tüm veri setlerinde en iyi sonuçları verdiği deneysel olarak tespit edilmiştir. Diğer temel yöntemler için orijinal makalelerde önerilen hiperparametreler ve varsayılan mimariler kullanılmıştır. Önerdiğimiz TPM modelinde düğüm başına yürüyüş sayısı $nw = 20$, yürüyüş uzunluğu $wl = 10$ olarak belirlenmiştir. Adil bir karşılaştırma yapılabilmesi için tüm yöntemlerde düğüm temsil boyutu (d) 128 olarak sabitlenmiştir.

5.4.1 Dügüm sınıflandırma sonuçları

Cora Citeseer ve Pubmed veri setleri kullanılarak, önerilen geçiş olasılık matrisi yöntemiyle yerleştirme vektörleri oluşturulmuştur. Dügüm özelliklerinin yerleştirme verileri, üç ağ için %80 eğitim ve %20 test verisi kullanılarak ayrılmıştır. Sınıf dengesizliğini ele almak amacıyla 10 katlı çapraz (cross-validation) doğrulama uygulanmış ve sonuçlar 10 denemenin ortalaması olarak sunulmuştur. Sınıflandırma, scikit-learn Python paketinin varsayılan ayarlarıyla Logistic Regression, Destek Vektör Makinesi (Support Vector Machine - SVM), Karar Ağaçları ve Çok Katmanlı Algılayıcı (Multi-Layer Perceptron - MLP) algoritmaları kullanılarak gerçekleştirilmiştir. Sonuçlar kabul edilebilir hata sınırları içinde kaldığı için yalnızca MLP'nin sonuçları burada ele alınmıştır.

Çizelge 5.7 Dügüm sınıflandırma sonuçlarının özeti. Kalın yazılmış değerler en iyi sonuçları temsil etmektedir

	CORA		CiteSeer		Pubmed	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
DeepWalk	71,5	69,2	56,0	52,1	65,9	65,2
node2vec	84,6	83,8	75,1	70,3	72,2	71,7
LINE	79,7	79,3	58,2	53,6	72,8	72,3
GCN	81,9	80,7	71,4	68,0	78,8	78,1
GSAGE	82,9	81,3	68,4	66,1	76,0	75,4
GAT	83,4	83,0	72,9	68,8	79,5	78,9
TPM	85,8	84,9	78,1	74,3	82,1	81,8

Çizelge 5.7, önerilen yöntemin üç katmanlı bir sinir ağı kullanılarak test edilen ve literatürde iyi performans sergileyen 6 güncel yerleştirme algoritması ile karşılaştırıldığı düğüm sınıflandırma sonuçlarını sunmaktadır. MLP modeli, 128 ve 64 nöronlu üç yoğun katmandan oluşmakta olup, ReLU aktivasyon fonksiyonu kullanılmaktadır. Son katmandaki nöron sayısı sınıf sayısına göre belirlenmiş ve softmax aktivasyon fonksiyonu kullanılmıştır. Doğruluk, Micro-F1 ve Macro-F1 skorlarıyla ölçülmüştür. **Çizelge 5.7**, önerilen TPM tabanlı özellik vektörlerinin tüm üç ağda düğüm sınıflandırmasında diğer

yöntemlerden üstün olduğunu göstermektedir. Bu üstünlük, hem Micro-F1 hem de Macro-F1 skorlarıyla belirtilmiştir.

5.4.2 Bağlantı tahmini sonuçları

Bağlantı temsili $g(u, v)$ oluşturmak için, iki düğüm u ve v arasında ikili vektör işlemleri uygulanmaktadır. Bu çalışmada, düğüm yerleştirme vektörleri, literatürde daha önce tanıtilan dört birleşim operatörü ile birleştirilmiştir (Grover ve Leskovec 2016). Bu birleşim yöntemleri, bağlantı vektörünün boyutunu düğüm vektörleriyle eşit tutarak hesaplama verimliliğini artırmaktadır. Ortalama, Hadamard, L1 ve L2 operatörleri, en iyi performans için değerlendirilmiş ve Hadamard operatörü en doğru tahminleri vermiştir. Hadamard işleminin formülü şu şekildedir:

$$g_i(u, v) = \{f(u) \cdot f(v)\}_i = f_i(u) * f_i(v) \quad (5.2)$$

Burada, $f(u)$ ve $f(v)$ (u, v) çiftine karşılık gelen vektörler olup, $f_i(u)$, $f(u)$ 'nin i -inci bileşenini, $g_i(u, v)$ ise bağlantı temsili $g(u, v)$ 'nin i -inci bileşenini temsil eder.

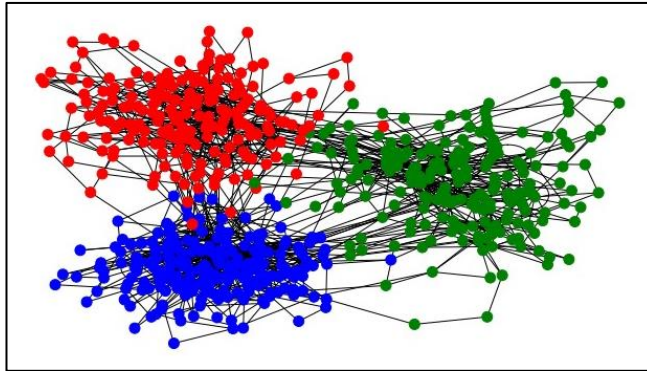
Bu bölümde, Cora, Citeseer ve Pubmed olmak üzere üç alıntı ağı üzerinde ikili sınıflandırma gerçekleştirilmiştir. Tahmin yöntemi olarak MLP algoritması test ve karşılaştırma amacıyla kullanılmıştır. MLP modeli, ReLU aktivasyonuna sahip 128 ve 64 nöronlu üç yoğun katmandan oluşmakta olup, son katmanda sigmoid aktivasyon fonksiyonu uygulanmaktadır. Tüm deneyler 10 kez gerçekleştirilmiş ve ortalama doğruluk değeri rapor edilmiştir. Başlangıç aşamasında, mevcut pozitif bağlantıların %10'u rasgele bir test seti olarak seçilir ve geri kalan ağı bağlantılılığını koruyarak bu kenarlar ağdan çıkarılır. Ayrıca, mevcut olmayan bağlantıların %10'u seçilerek düğüm çiftleri oluşturulur ve bu çiftler negatif örnekler olarak tanımlanır. Bu süreçten sonra, temel algoritmalar kalıntı ağ üzerinde eğitilir. Son olarak, bu algoritmalarından elde edilen yerleştirme vektörleri, test setindeki kenarların tahmin edilmesinde kullanılır. **Çizelge 5.8**, bağlantı tahmini sonuçlarının karşılaştırmasını sunmakta olup, TPM tüm veri setlerinde diğer yöntemlere kıyasla üstün performans sergilemektedir.

Çizelge 5.8 Bağlantı tahmini sonuçlarının özeti. Kalın yazılmış değerler en iyi sonuçları temsil etmektedir

	CORA	CiteSeer	Pubmed
DeepWalk	85,1	82,4	86,4
node2vec	92,7	94,1	91,2
LINE	85,6	80,3	83,3
GCN	91,8	88,9	95,7
GSAGE	91,0	87,7	93,1
GAT	90,8	91,4	92,4
TPM	95,9	96,2	96,9

5.4.3 Görselleştirme sonuçları

Görselleştirme, yerleştirme algoritmalarının performansını değerlendirmek için etkili bir yöntem olarak kabul edilmektedir (Hoang vd. 2023). Bu amaçla genellikle Zachary Karate Club (Girvan ve Newman 2002) veya Cora (McCallum vd. 2000) gibi yaygın olarak bilinen ağlar tercih edilmektedir. Ancak, gerçek dünya ağları genellikle topluluk büyüklükleri ve düğüm dereceleri gibi yönlerden türdeş olmayan (ayrışık) dağılımlar sergilemektedir.

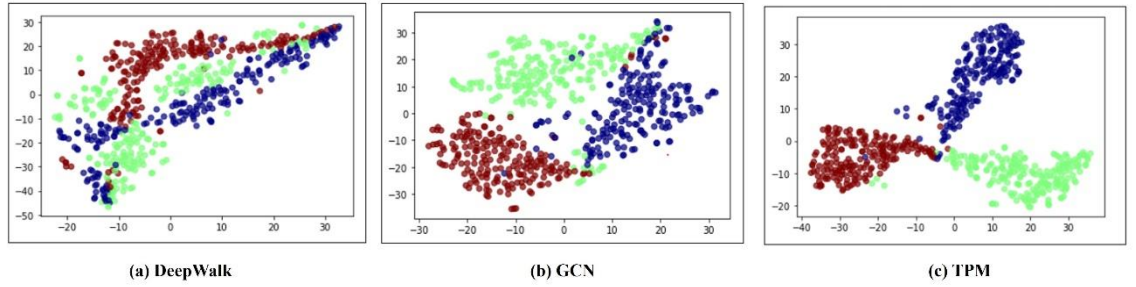


Şekil 5.3 Üç topluluk, 600 düğüm ve 1.334 kenardan oluşan yapay LFR ağı

Bu nedenle, yaygın olarak kullanılan ağlara başvurmak yerine, yerleştirme algoritmalarını test etmek için özel olarak tasarlanmış Lancichinetti-Fortunato-Radicchi (LFR) algoritması (Lancichinetti vd. 2008) bu deneyde tercih edilmiştir. Bu bağlamda,

TPM algoritması, 3 topluluk, 600 düğüm ve 1.334 kenardan oluşan yapay bir ağa uygulanmıştır. LFR algoritması, ağın türdeş olmayan özelliğini kontrol eden ek bir μ parametresine sahiptir. 0 ile 1 arasında değişen bu "gürültü" parametresi, ağın türdeş olmayan özelliğini yansıtmaktadır. Bu çalışmada, türdeş olmayan özellik parametresi $\mu = 0,2$ olarak belirlenmiştir. Bu deney oluşturulan yapay LFR ağı **Şekil 5.3**'te gösterilmektedir.

Literatürde güncel ve iyi performans sergileyen iki yerleştirme algoritması olan DeepWalk (Perozzi vd. 2014) ve GCN (Kipf ve Welling 2016), önerilen TPM algoritması ile görsel değerlendirme amacıyla karşılaştırılmıştır. Boyut indirgeme işlemi için t-SNE aracı kullanılmıştır. **Şekil 5.4**, önerilen modelin diğer iki yönteme kıyasla daha kompakt ve belirgin kümeler oluşturduğunu ortaya koymaktadır. **Şekil 5.4**'te, DeepWalk yöntemi tarafından üretilen temsillerin, örtüşen birden fazla kümeye sahip olduğu gözlemlenmektedir. GCN modeli, düğüm özelliklerini yaymak ve ek küresel yapısal bilgileri yakalamak için bağlantı yapısını kullanarak daha tutarlı temsiller sağlamaktadır. **Şekil 5.4.c**'nin görsel incelemesi, önerilen yerleştirme algoritması TPM'nin ağ içindeki üç sınıf arasında belirgin bir ayrım sağladığını göstermektedir.



Şekil 5.4 LFR ağı için 2 boyutlu temsillerin görselleştirilmesi

5.4.4 Ağlar arası genelleme sonuçları

Ölçekten bağımsız ağlar, çoğu gerçek dünya ağını oluşturduğundan dolayı özel bir öneme sahiptir. Gerçek dünya ağlarının çoğu, benzer davranışlar sergileyerek ölçeklenebilir derece dağılımına sahip olma eğilimindedir (Barabási ve Bonabeau 2003). Bu bağlamda, önerilen TPM algoritmasını ölçeklenebilir ağlar üzerinde çalıştırmak ve test etmek kritik

bir önem taşımaktadır. Bu bölümde, benzer bağlantı yapılarına sahip ağlardan elde edilen bilgileri kullanarak yeni ağlarda tahminler yapmayı amaçlayan ağlar arası genelleme işlemi gerçekleştirilmektedir. Deney sırasında, eğitim sürecinde karşılaşılmayan ağlar üzerinde bağlantı tahminleri yapılmaktadır. Başlangıçta, 12 Barabási-Albert ağı, $n = \{1.000, 10.000\}$ (düğüm sayısı) ve $\alpha = 6$ (mevcut düğümlerin yeni bir düğüme bağlanabileceği sayı) parametreleriyle üretilmiştir. Sonrasında, her yerleştirme yöntemi bu ağların on tanesinde eğitilmiş ve ardından iki test ağı üzerindeki ortalama doğruluk puanı hesaplanmıştır. Önerilen TPM modelimizin performansı, literatürde güncel ve iyi performans sergileyen dört özellik çıkarım algoritması (DeepWalk (Perozzi vd. 2014), node2vec (Grover ve Leskovec 2016), GCN (Kipf ve Welling 2016) ve GraphSAGE (Hamilton vd. 2017a)) ile karşılaştırılmıştır. Deneysel sonuçlar **Çizelge 5.9**'da sunulmuştur.

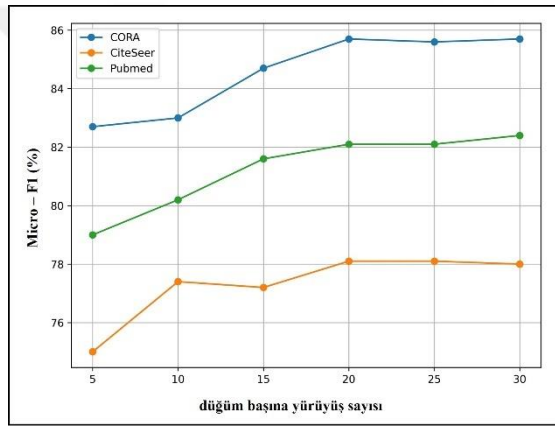
Çizelge 5.9 İki ölçekten bağımsız ağ için ağlar arası genelleme açısından sonuçların özeti. Kalın yazılmış değerler en iyi sonuçları temsil etmektedir

	DeepWalk	node2vec	GCN	GSAGE	TPM
Ölçekten bağımsız ağ $n = (1.000)$	61,1	66,4	70,2	71,3	74,6
Ölçekten bağımsız ağ $n = (10.000)$	62,0	65,9	71,9	71,7	75,1

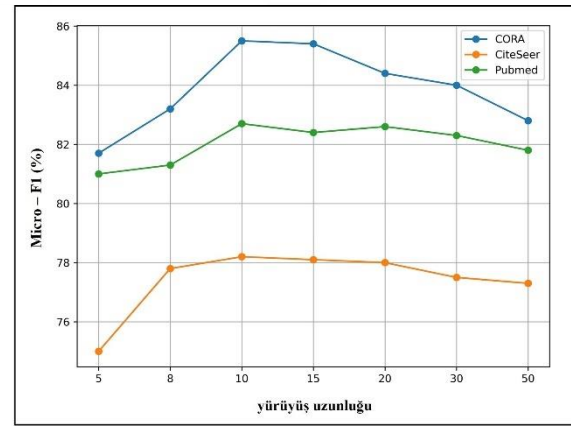
Tüm rasgele yürüyüş tabanlı modeller (TPM, DeepWalk ve node2vec) için, her düğümden toplamda 50 rasgele yürüyüş gerçekleştirilmiştir (her üretilen eğitim Barabási-Albert ağı için 5 rasgele yürüyüş). Çizelge evrişimli ağ tabanlı modeller (GCN ve GraphSAGE) için ise, her düğümden bilgi toplama işlemi, on üretilen eğitim Barabási-Albert ağı üzerinden eşit olarak gerçekleştirilmiştir. Üretilen Barabási-Albert ağlarında düğümler özelliklerden yoksundur; özellikler olarak düğüm dereceleri ve eğitim sırasında her düğüm için güncellenen yerleştirme ağırlıkları kullanılmıştır. **Çizelge 5.9**, TPM'nin dört özellik çıkarım algoritmasını önemli ölçüde geride bıraktığını göstermektedir. Bu sonuçlar, TPM modelimizin, benzer topolojilere sahip farklı ağlarda bile düğümlerin yerel yapısal kalıplarını etkili bir şekilde yakaladığını ortaya koymaktadır.

5.4.5 Parametre hassasiyeti sonuçları

Önerilen yerleştirme algoritması TPM, iki temel hiperparametre içerir: nw (düğüm başına yürüyüş sayısı) ve wl (yürüyüş uzunluğu). Farklı nw ve wl değerleri ile Cora, Citeseer ve Pubmed veri setleri üzerinde düğüm sınıflandırmasının Micro-F1 skoru hesaplanmıştır. Bu hesaplamalar, TPM'deki hiperparametrelerin etkisini ortaya koymaktadır. Düğüm başına yürüyüş sayısı nw ve yürüyüş uzunluğu wl , sırasıyla $\{5, 10, 15, 20, 25, 30\}$ ve $\{5, 8, 10, 15, 20, 30, 50\}$ olarak belirlenmiştir. Eğitim oranı %80 olarak ayarlanmış ve elde edilen sonuçlar **Şekil 5.5**'te sunulmuştur.



(a) Yürüyüş sayısının etkisi



(b) Yürüyüş uzunluğunun etkisi

Şekil 5.5 Parametre hassasiyeti deneyinin sonuçları. (a) Üç alıntı ağı üzerindeki yürüyüş sayısının etkisi. (b) Üç alıntı ağı üzerindeki yürüyüş uzunluğunun etkisi

Şekil 5.5.a'da gösterildiği üzere, önerilen modelin performansı, nw 'nin 20'den 30'a değiştiği üç alıntı ağı üzerinde nispeten sabit kalmaktadır. Bu durum, yürüyüş sayısının artmasının modelin performansını artırdığını, çünkü verilen düğüm etrafındaki bölgenin karakteristik bağlantı yapısının daha ayrıntılı bir şekilde yakalandığını göstermektedir. **Şekil 5.5.b**'de ise, önerilen yerleştirme algoritması TPM'nin, üç alıntı ağı üzerinde en iyi Micro-F1 skorunu wl değerinin 10 olduğu durumda elde ettiği gözlemlenmiştir. Yürüyüş uzunluğu arttıkça, skor kademeli olarak azalmaktadır; çünkü wl çok büyük olduğunda yürüyüşü düğümün komşuluk bölgesinden uzaklaşabilir ve bu nedenle yerel yapısal kalıpları yürüyüş istatistiklerine düzgün bir şekilde dahil edemez. Bu nedenle, nw ve wl parametrelerinin çeşitli uygulamalar için uygun şekilde ayarlanması gerekmektedir.

5.5 Kılavuzlu Geçiş Olasılık Matrisi Modelinin Sonuçları

Bu bölümde, önerilen GTPM yerleştirme algoritmasının metin sınıflandırma performansı incelenmektedir. Algoritmanın sınıflandırma başarımını değerlendirmek amacıyla SST-2, MR, CoLA, Ohsumed, Reuters (Keras) ve 20NG olmak üzere altı farklı metin tabanlı veri seti kullanılmıştır. İlk üç veri seti ikili sınıflandırma problemleri için tercih edilirken, geri kalan veri setleri daha karmaşık bir görev olan çok sınıflı sınıflandırma problemlerinde uygulanmıştır. Bu sayede, algoritmanın hem basit hem de karmaşık sınıflandırma görevlerindeki performansı kapsamlı bir şekilde test edilmiştir.

Karşılaştırmalar, metin tabanlı ve ağ tabanlı olmak üzere sekiz iyi bilinen temel model ve başarılı yerleştirme sınıflandırma algoritması kullanılarak gerçekleştirilmiştir. Tüm yerleştirme algoritmaları için, orijinal makalelerde veya uygulamalarda belirtilen varsayılan parametre ayarları tercih edilmiştir. Ayrıca, önceden eğitilmiş kelime yerleştirmeleri kullanan modellerde 300 boyutlu BERT kelime yerleştirme modeli kullanılmıştır. Bu çalışmada sunulan deneysel bulgular, aynı ön işleme adımları izlenerek gerçekleştirilen beş bağımsız çalıştırmanın ortalama sonuçlarını yansıtmaktadır.

5.5.1 Veri ön işleme

Yerleştirme algoritmalarını uygulamadan önce, tüm veri setleri dikkatlice bir veri temizleme sürecine tabi tutulmuştur. Bu süreç, NLTK kütüphanesi kullanılarak gerçekleştirilen birkaç aşamadan oluşmuştur. İlk olarak, tüm metinler küçük harfe dönüştürülerek (lowercasing), büyük-küçük harf farklılıklarının yol açabileceği tutarsızlıklar ortadan kaldırılmıştır. Ardından, metinlerden noktalama işaretleri (punctuations) temizlenmiş, böylece dil bilgisel işaretlerin veri analizi üzerinde olumsuz bir etkisi engellenmiştir. Sonrasında, tokenizasyon işlemi uygulanarak, metinler kelime ve diğer anlamlı birimlere ayrılmıştır. Bu aşamadan sonra, sık kullanılan ve analiz açısından anlam taşımayan durdurma kelimeleri (stopwords) metinlerden çıkarılmıştır. Son olarak, kökleme (stemming) işlemi gerçekleştirilerek, kelimeler kök formuna dönüştürülmüş ve böylece kelime varyasyonlarının analizde yarattığı karmaşıklık azaltılmıştır. Bu kapsamlı veri temizleme adımları, veri setlerinin daha tutarlı ve analiz

edilebilir bir biçime getirilmesini sağlamış ve dolayısıyla yerleştirme algoritmalarının etkin bir şekilde uygulanmasına zemin hazırlamıştır.

Düğüm öznitelik vektörlerini üretmek amacıyla önerilen GTPM algoritması, her veri seti için aşağıdaki prosedürü izler: Ön işleme aşamasının tamamlanmasının ardından, özgün kelimeler ağıın düğümleri olarak tanımlanır. Bağlantılar ve bu bağlantılara ait ağırlıklar, orijinal metinde bulunan bilgiler doğrultusunda belirlenir. Ağırlıklı ağ üzerinde rasgele yürüyüşler gerçekleştirerek geçiş olasılığı matrisini (TPM) hesaplamak için bir süreç yürütülür (Mohammed ve Gündüç 2023). Elde edilen geçiş olasılığı matrisi, her bir düğümün yerel topolojisini ve bağlantısallığını yansıtan belirgin öznitelik vektörlerine dönüştürülür.

5.5.2 Metin sınıflandırma süreci

Önerilen yerleştirme algoritması kullanılarak üretilen düğüm özellik vektörleri, metin sınıflandırması amacıyla tasarlanmış çok katmanlı bir sinir ağına giriş olarak sunulmaktadır. TensorFlow'un Keras API'si aracılığıyla inşa edilen sinir ağı, sırasıyla (64, 128, 256, 512, çıkış katmanı) olmak üzere beş katmandan oluşmaktadır. Gizli katmanlarda ReLU aktivasyon fonksiyonu kullanılırken, çıkış katmanında sınıflandırma görevine bağlı olarak sigmoid veya softmax aktivasyon fonksiyonu uygulanmaktadır. Öğrenme oranları ve dropout oranları için sırasıyla (0,1, 0,001, 0,0001, 0,02, 0,002, 0,003) ve (0,1, 0,2, 0,5) kümelerinden seçilmiştir. Ayrıca, epoch sayısını düzenlemek amacıyla erken durdurma stratejisi uygulanmaktadır. Veri seti, metin sınıflandırma amaçları doğrultusunda eğitim ve test setlerine ayrılmış olup, eğitim verilerinin %10'u rasgele olarak doğrulama seti olarak belirlenmiştir.

5.5.3 Metin sınıflandırma sonuçları

Önerilen yerleştirme algoritmasının etkinliği ve dayanıklılığı, literatürde güncel ve iyi performans sergileyen sekiz yerleştirme algoritması ile performanslarının karşılaştırılması yoluyla değerlendirilmiştir. Testler, iki ana kategori altında

gerçekleştirilmiştir: ikili sınıflandırma ve çok sınıflı sınıflandırma. Sınıflandırma doğruluğu ölçütleri (Micro-F1 ve Macro-F1) temel alınarak değerlendirilmiştir. En yüksek değerler kalın harflerle, ikinci en yüksek değerler ise altı çizili olarak belirtilmiştir. Elde edilen bulgular, yerleştirme algoritmasının karmaşıklığından bağımsız olarak, verimliliğin belgelerin içerdiği sınıf sayısına büyük ölçüde bağlı olduğunu ortaya koymuştur. İkili sınıflandırma performansını değerlendirmek amacıyla SST-2, MR ve CoLA veri setleri kullanılmıştır. Çok sınıflı sınıflandırma performansını test etmek için ise Reuters (Keras), 20NG ve Ohsumed veri setleri tercih edilmiştir.

Çizelge 5.10 Önerilen GTPM yönteminin MR, SST-2 ve CoLA veri setlerinde ikili metin sınıflandırma modelleri arasındaki karşılaştırmalı performansı

Modeller	MR		SST-2		CoLA	
	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1
TF-IDF + LR	77,14	76,21	82	81,31	65,39	63,2
CNN-pretrain	76,93	76,01	79,45	78,98	63,27	60,07
Bi-LSTM	76,81	76,21	79,6	79,4	63,21	60,1
SWEM	77,29	76,33	82,14	81,6	66,34	64,6
TextGCN	77,43	77,02	82,45	82,4	68,1	66,71
TextING	79,69	79,1	83,11	83	69,4	67,11
BERT	86,6	85,31	91,53	91,02	82,17	77,08
VGCN-BERT	<u>86,73</u>	<u>86,23</u>	<u>91,94</u>	<u>91,55</u>	<u>83,91</u>	<u>80,5</u>
GTPM	87,41	87,05	92,45	92,13	85,14	83,67

Çizelge 5.10'da, ikili metin sınıflandırma sonuçlarını üç veri seti bazında sunmakta iken, çok sınıflı metin sınıflandırma sonuçları **Çizelge 5.11**'de detaylandırılmıştır. Deneysel sonuçlar, **Çizelge 5.10** ve **Çizelge 5.11**'de yer alan verilere göre, önerilen modelin ikili ve çok sınıflı metin sınıflandırma görevlerinde üstün performans sergilediğini ve tüm temel modelleri geride bıraktığını göstermektedir. Bu bulgular, önerilen GTPM yerleştirme algoritmasının her iki sınıflandırma türü için de yüksek performanslı ve dayanıklı bir çözüm sunduğunu desteklemektedir.

Çizelge 5.11 Önerilen GTPM yönteminin Reuters, 20NG ve Ohsumed veri setlerinde çok sınıflı metin sınıflandırma modelleri arasındaki karşılaştırmalı performansı

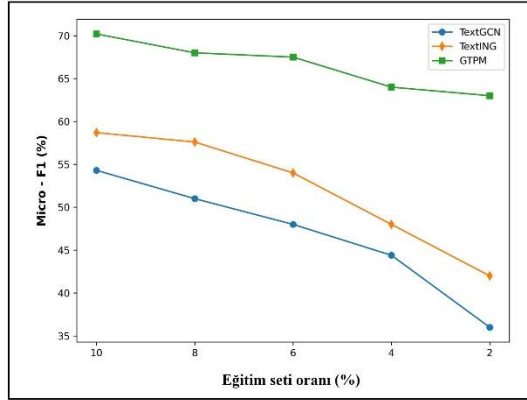
Modeller	Reuters (Keras)		20NG		Ohsumed	
	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1
TF-IDF + LR	84,75	84,52	79,75	79	60,42	52,2
CNN-pretrain	86,66	86,41	81,09	80,18	62,45	57,13
Bi-LSTM	86,92	86,2	80,86	80,8	62,48	59,45
SWEM	87,13	86,86	81,77	81,69	68,03	63,2
TextGCN	87,24	87,15	<u>85,88</u>	<u>85,34</u>	68,11	60,61
TextING	87,59	87,58	82,76	82,6	71,92	68,8
BERT	89,68	89,6	84,03	84,01	69,38	65,1
VGCN-BERT	<u>90,34</u>	<u>90,3</u>	84,22	84,16	<u>72,04</u>	<u>68</u>
GTPM	92,91	92,8	86,43	86,09	76,19	71,11

5.5.4 Sınırlı eğitim verileriyle yerleştirme algoritmalarının performans analizi

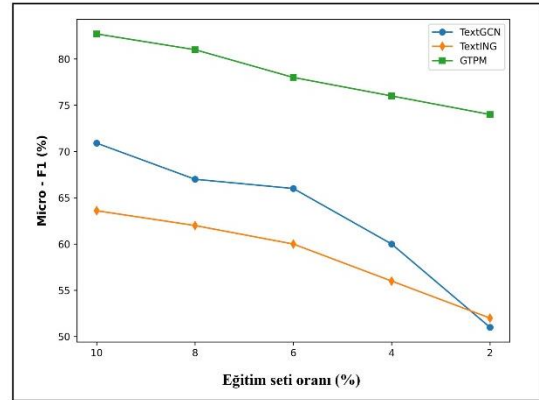
Yerleştirme algoritmalarının performansı, özellikle sınırlı büyüklükteki eğitim setleriyle çalışıldığında, sistemin genel doğruluğu ve verimliliği üzerinde kayda değer bir etki yaratmaktadır. Küçük veri setleriyle çalışan modellerde, algoritmanın genelleme kapasitesinin doğru bir şekilde değerlendirilmesi, sonuçların güvenilirliği açısından kritik bir önem taşımaktadır. Bu bağlamda, sınırlı veri setleriyle elde edilen sonuçların hem modelleme sürecinde hem de performans analizinde dikkate alınması gerekmektedir. Bu kapsamda, önerilen modelin dayanıklılığı, iki ağ tabanlı (TextGCN ve TextING) yerleştirme algoritmalarına karşı incelenmiştir. Yapılan deneyler, Ohsumed ve 20NG veri setleri kullanılarak ve sınırlı eğitim verisiyle gerçekleştirilmiştir. Eğitim verisinin yalnızca %10'u kullanılmış olup, test seti ise orijinal haliyle korunmuştur. Bu durum, test setindeki kelimelerin büyük bir bölümünün eğitim sırasında görülmemesine yol açmıştır. Sonuçlar **Çizelge 5.12**'de sunulmuştur. Modellerin performansı Micro F1 ve Macro F1 skorlarına dayalı olarak değerlendirilmiş ve her iki veri setinden elde edilen karşılaştırmalı sonuçlar, GTPM yönteminin belirgin bir üstünlük sağladığını göstermiştir.

Çizelge 5.12 Tümevarımsal ayarda çok sınıflı metin sınıflandırma veri setleri için Micro F1 ve Macro F1 skorları. Çizelge 5.11'deki performans düşüşleri parantez içinde belirtilmiştir

	Ohsumed (10%)		20NG (10%)	
	Micro F1	Macro F1	Micro F1	Macro F1
TextGCN	54,31 (-14)	44,07 (-17)	70,89 (-15)	69,39 (-16)
TextING	58,63 (-13)	54,11 (-15)	63,52 (-19)	63,27 (-19)
GTPM	70,26 (-6)	60,97 (-10)	82,67 (-4)	80,21 (-6)
Eğitim setindeki örnek/kelime sayısı	335/5.409		1.131/7.568	
Test setindeki örnek sayısı	4.043		7.532	



(a) Ohsumed



(b) 20NG

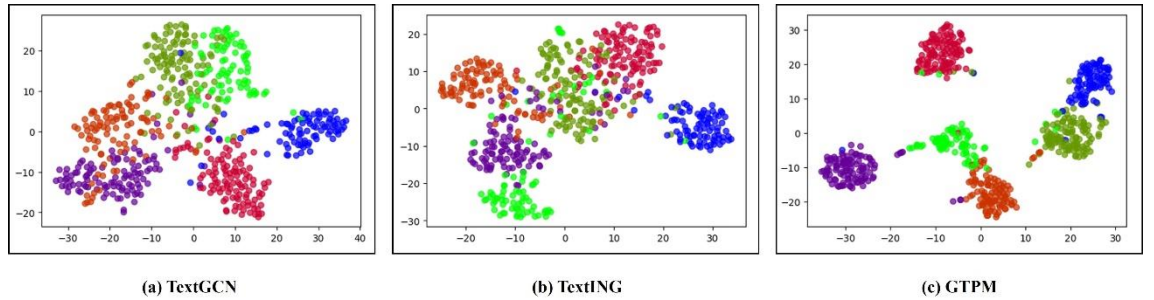
Şekil 5.6 Farklı eğitim boyutlarıyla Ohsumed ve 20NG veri setlerinde test performansı (Micro F1)

Şekil 5.6, Ohsumed ve 20NG veri setlerinde etiketli eğitim verisinin değişen oranları (%10'dan %2'ye kadar) kullanılarak TextGCN, TextING ve GTPM yöntemlerinin performanslarını karşılaştırmaktadır; bu süreçte orijinal test seti boyutu sabit tutulmuştur. Önerilen GTPM yöntemi, eğitim sırasında görünmeyen kelime sayısındaki artışa rağmen tutarlı bir performans sergilemektedir. Her iki veri setinde de GTPM'nin performans düşüşü yaklaşık %8 ile sınırlı kalırken, diğer yöntemlerde %15 ile %20 arasında daha belirgin düşüşler gözlemlenmiştir. Bu bulgular, GTPM yönteminin sınırlı etiketli verilerle

geniş test setlerine güçlü bir genelleme kapasitesine sahip olduğunu ve bu bağlamda indüktif öğrenme prensipleriyle etkili bir uyum sağladığını göstermektedir.

5.5.5 Görselleştirme sonuçları

Görsel inceleme, sınıflandırma algoritmalarının değerlendirilmesinde oldukça etkili bir yöntem olarak kabul edilmektedir. Çok boyutlu vektörlerin iki veya üç boyutlu temsillere dönüştürülmesi süreci, görselleştirme açısından kritik bir öneme sahiptir. Bu bağlamda, boyut indirgeme, çok boyutlu verilerin daha düşük boyutlu ve grafiksel olarak daha anlamlı temsillere dönüştürülmesi amacıyla kullanılan standart bir tekniktir. Reuters veri seti için, özellik vektörleri üç farklı ağ tabanlı yerleştirme yöntemi kullanılarak türetilmiştir: TextGCN, TextING ve önerilen model (GTPM). Bu özellik vektörleri, t-SNE algoritması aracılığıyla iki boyuta indirgenmiştir. **Şekil 5.7**, GTPM'nin farklı sınıfları diğer yerleştirme algoritmalarına kıyasla daha iyi ayırt edebildiğini göstermektedir. Bu bulgu, GTPM yaklaşımının performans beklentilerini artırmakta ve tüm metin işleme amaçları için etkili bir yerleştirme temeli sunduğunu ortaya koymaktadır.

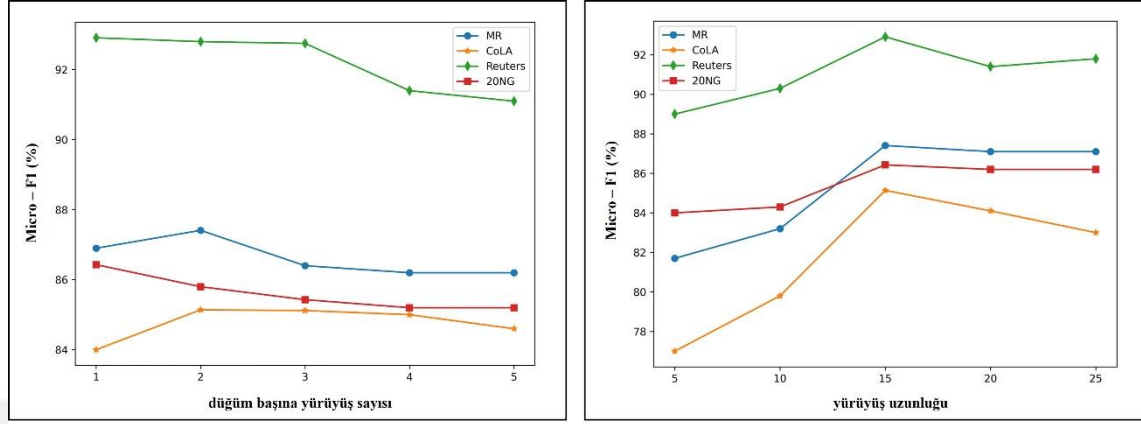


Şekil 5.7 Reuters veri seti için 2 boyutlu temsillerin görselleştirilmesi

5.5.6 Parametre hassasiyeti sonuçları

Önerilen yerleştirme algoritması GTPM, iki temel hiperparametreye sahiptir: nw (düğüm başına yürüyüş sayısı) ve wl (yürüyüş uzunluğu). Bu iki parametrenin optimize edilmiş değerlerini belirlemek amacıyla yürüyüş uzunlukları ve düğüm başına yürüyüş sayıları

sistematik olarak değiştirilmiş ve performansın bu parametrelerle olan bağımlılığı incelenmiştir. **Şekil 5.8**, gerçekleştirilen testlerin sonuçlarını göstermektedir.



(a) Yürüyüş sayısının etkisi

(b) Yürüyüş uzunluğunun etkisi

Şekil 5.8 Parametre hassasiyeti deneyinin sonuçları. (a) Dört veri seti üzerindeki yürüyüş sayısının etkisi. (b) Dört veri seti üzerindeki yürüyüş uzunluğunun etkisi

Gözlemler, GTPM modelin etkinliğinin yürüyüş uzunluğunun belirli bir noktaya kadar artış göstermesiyle iyileştiğini ve bu noktayı aştığında performansın azaldığını ortaya koymaktadır. En yüksek performans, yürüyüş uzunluğunun yaklaşık 15 adım olduğu durumlarda elde edilmektedir. Her düğümden (veya belgelerdeki kelimelerden) başlatılan rasgele yürüyüşlerin ideal sayısı, veri setinin ortalama cümle uzunluğuna bağlı olarak değişkenlik göstermektedir. Özellikle Reuters ve 20NG gibi uzun ortalama cümle uzunluklarına sahip veri setlerinde, her düğümden tek bir rasgele yürüyüş başlatmak en yüksek performansı sağlamaktadır. Bununla birlikte, MR ve CoLA gibi kısa ortalama cümle uzunluklarına sahip veri setlerinde, her kelime için birden fazla rasgele yürüyüş başlatmak daha etkili sonuçlar vermektedir. Bu bağlamda, tez kapsamında yerleştirme performansının karşılaştırmalı analizlerinde, GTPM yöntemi için düğüm başına yürüyüş sayısı ($nw = 1$) ve yürüyüş uzunluğu ($wl = 15$) olarak sabitlenmiştir. Aynı şekilde, karşılaştırma amacıyla kullanılan diğer yerleştirme algoritmalarının performansı, literatürde bildirilen optimal parametre değerleri temel alınarak değerlendirilmiştir.

6. SONUÇ

Tez kapsamında, ağ tabanlı yerleştirme yöntemleri incelenmiş ve bu alanda önemli katkılar sağlanarak çeşitli ağ analiz işlemlerinin makine öğrenimi çözümleri ile etkili bir şekilde gerçekleştirilebilmesi hedeflenmiştir. Özellikle, bu yöntemlerin ağ analizi ve metin sınıflandırma alanlarındaki uygulamaları üzerinde durulmuştur. Yenilikçi yaklaşımlar ve kapsamlı deneyler aracılığıyla, bu yöntemlerin karmaşık yapısal ve anlamsal bilgileri yakalama konusundaki etkinliği ve potansiyeli ortaya konulmuştur. Bu tezin temel katkıları, üç yenilikçi algoritmanın geliştirilmesine dayanmaktadır: derece tabanlı rasgele yürüyüş, geçiş olasılık matrisi (TPM) ve kılavuzlu geçiş olasılık matrisi (GTPM). Her bir algoritma, ağ ve metinsel bilgilerin temsilde karşılaşılan belirli zorluklara çözüm sağlamakta olup, ağ tabanlı yerleştirme tekniklerinin çeşitli uygulamalardaki potansiyelini ortaya koymaktadır.

Araştırma, mevcut ağ temsil tekniklerinin genellikle değerli ağ bilgilerini ileri işlemeye uygun sayısal vektörlere dönüştürme konusundaki sınırlamalarını ele alarak başlamıştır. Bu bağlamda, dereceye dayalı rasgele yürüyüş yaklaşımı önerilmiştir; bu yaklaşım, DeepWalk ve node2vec gibi sabit sayılı yürüyüş yöntemlerine kıyasla birkaç avantaj sunmaktadır. Yürüyüş sayısının düğüm derecesine orantılı olarak seçilmesi sayesinde, bu yöntem yüksek dereceli düğümlere vurgu yapmakta, düşük dereceli düğümleri veri dengesizliğini azaltmak amacıyla bastırmakta ve özellikle ölçekten bağımsız ağlar için hesaplama süresini önemli ölçüde kısaltmaktadır. Bu yaklaşım, CORA ve CiteSeer veri setlerinde yaklaşık %50 daha az hesaplama gücü gerektirirken, düğüm sınıflandırması ve bağlantı tahmini görevlerinde mevcut yöntemlerle karşılaştırılabilir doğruluk sağlamaktadır.

Bu temelin üzerine, yerel bağlantı yapılarını ve düğüm benzerliklerini yakalayan denetimsiz bir düğüm yerleştirme algoritması olan geçiş olasılık matrisi (TPM) yöntemi önerilmiştir. Belirlenen bir komşuluk yarıçapı içinde anonim yürüyüşler kullanılarak, TPM her düğüm için benzersiz geçiş olasılığı matrisleri oluşturmakta ve düğüm kimliğinden bağımsız olarak yerel yapısal desenleri etkili bir biçimde temsil etmektedir. TPM yöntemi, sınırlı komşuluklarda kısa rasgele yürüyüşlere odaklanarak hesaplama

süresini azaltmaktadır. Bu yöntem, ağın farklı bölümlerinde benzer bağlantı yapılarını tanımlamada üstün bir performans sergilemiş ve yalnızca aynı ağ üzerinde eğitim ve tahmin yapmakla kalmayıp, benzer bağlantı yapılarına sahip ağlardan bilgi öğrenerek yeni ağlar üzerinde tahminlerde bulunma görevini literatüre kazandırmıştır. TPM modeli, benzer topolojilere sahip farklı ağlarda dahi düğümlerin yerel yapısal örüntülerini başarıyla yakaladığını kanıtlamıştır.

Sonuç olarak, cümlelerin doğal ağ yapısını kullanarak yerleştirme vektörleri oluşturan bir metin yerleştirme yöntemi olan kılavuzlu geçiş olasılık matrisi (GTPM) modeli geliştirilmiştir. Bu model, metinden bir kelime ağı oluşturarak ve rasgele yürüyüşler uygulayarak, metinsel verilerdeki sözdizimsel, anlamsal ve gizli içerik özelliklerini kapsayan yerleştirme vektörleri üretmektedir. Bu süreç, dil kurallarının metin belgelerinden çıkarılmasını kolaylaştırmaktadır. GTPM ile gerçekleştirilen metin sınıflandırma çalışması, yöntemimizin hem ikili hem de çok sınıflı sınıflandırma görevlerinde temel modellere kıyasla üstün etkinlik sağladığını ortaya koymuştur. Yöntemin dayanıklılığı, özellikle sınırlı eğitim verileri içeren bağlamlarda belirgin bir şekilde gözlemlenmiş ve etiketli verilerin sınırlı olabileceği pratik uygulamalar için potansiyeli vurgulanmıştır.

Yerleştirme vektörleri, TSNE gibi boyut indirgeme teknikleri kullanılarak görsel olarak incelenmiş ve önerilen yöntemlerin etkinliğini doğrulayan belirgin kümelenme desenleri ortaya konulmuştur. Bu kümelenme desenlerinin, sınıflandırma doğruluğunu artırdığı tespit edilmiştir. Ayrıca, parametre seçimi deneylerinde, çeşitli görevlerde en iyi performansın elde edilmesi için düğüm başına rasgele yürüyüş sayısı ve rasgele yürüyüş uzunluğunun optimize edilmesinin önemli olduğu gösterilmiştir.

Sonuç olarak, bu çalışmada, ağ ve metin yerleştirme yöntemlerindeki önemli eksiklikler, ölçeklenebilir, verimli ve son derece hassas temsili öğrenme teknikleri sunularak ele alınmıştır. Önerilen algoritmaların, yalnızca hesaplama maliyetlerini düşürmekle kalmayıp, aynı zamanda önemli yapısal ve anlamsal bilgileri yakalama kapasitesini artırdığı gösterilmiştir. Bu sayede, ağ analizi ve doğal dil işleme alanlarında literatürde mevcut ve yüksek performanslı yerleştirme modellerine önemli katkılar sağlanmıştır.

6.1 Gelecek Çalışmalar

Ağ topolojisinin yanı sıra, gerçek dünya ağlarının düğümleri, zengin öznitelik bilgilerine sahiptir. Örneğin, Wikipedia makaleleri birbirine bağlanarak bir ağ oluşturur ve her makale bir düğüm olarak önemli metin bilgisi içerir. Benzer şekilde, atıf ağlarındaki makaleler ve sosyal ağlardaki kullanıcı profilleri, düğüm öznitelikleri olarak zengin içerikler sunar. Düğüm öznitelikleri, ağların yapısının oluşumuna katkıda bulunmakla kalmaz, aynı zamanda düğümler arasındaki öznitelik düzeyindeki benzerliği değerlendirmek için de doğrudan kanıt sağlar. Bu nedenle, öznitelik bilgisi uygun şekilde kullanıldığında, ağ yapısıyla birleştirilerek daha anlamlı düğüm temsilleri oluşturulabilir.

Gelecek çalışmalar, düğüm temsilinin kalitesini artırmak için düğümlere ait karakteristik özelliklerin topoloji ile birleştirilebilir. Önerilen yöntemlerin daha karmaşık makine öğrenimi süreçlerine entegrasyonunu, dinamik, öznitelikli ağlara ve farklı alanlara transfer edilebilirliğinin daha derinlemesine araştırılması planlanmaktadır. Ayrıca, bu yerleştirme tekniklerinin çizge sinir ağları ve transformatör tabanlı gibi ileri düzey makine öğrenimi modelleriyle entegrasyonu, ağ analizi ve doğal dil işleme alanlarındaki en son gelişmeleri daha da ileriye taşımak için umut verici fırsatlar sunmaktadır.

KAYNAKLAR

- Adamic, L. A. ve Adar, E. 2003. Friends and neighbors on the web. *Social networks*, 25(3), 211-230.
- Ahmed, N. K., Rossi, R. A., Lee, J. B., Willke, T. L., Zhou, R., Kong, X. ve Eldardiry, H. 2020. Role-based graph embeddings. *IEEE Transactions on Knowledge and Data Engineering*, 34(5), 2401-2415.
- Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q. ve Chen, G. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, PMLR, 173-182.
- Aridhi, S. ve Nguifo, E. M. 2016. Big graph mining: Frameworks and techniques. *Big Data Research*, 6, 1-10.
- Arrar, D., Kamel, N. ve Lakhfif, A. 2024. A comprehensive survey of link prediction methods. *The journal of supercomputing*, 80(3), 3902-3942.
- Backstrom, L. ve Leskovec, J. 2011. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, 635-644.
- Baptista, A., Sánchez-García, R. J., Baudot, A. ve Bianconi, G. 2023. Zoo guide to network embedding. *Journal of Physics: Complexity*, 4(4), 042001.
- Barabási, A.-L. ve Albert, R. 1999. Emergence of scaling in random networks. *science*, 286(5439), 509-512.
- Barabási, A.-L. ve Bonabeau, E. 2003. Scale-free networks. *Scientific american*, 288(5), 50-59.
- Bishop, C. M. ve Nasrabadi, N. M. 2006. *Pattern recognition and machine learning*. Springer, 4.
- Borgwardt, K. M. ve Kriegel, H.-P. 2005. Shortest-path kernels on graphs. In *Fifth IEEE international conference on data mining (ICDM'05)*, IEEE, 8 pp.
- Brody, S., Alon, U. ve Yahav, E. 2021. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*.

- Bruna, J., Zaremba, W., Szlam, A. ve Lecun, Y. 2013. Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203.
- Cao, S., Lu, W. ve Xu, Q. 2015. Grarep: Learning graph representations with global structural information. In Proceedings of the 24th ACM international on conference on information and knowledge management, 891-900.
- Cao, S., Lu, W. ve Xu, Q. 2016. Deep neural networks for learning graph representations. In Proceedings of the AAAI conference on artificial intelligence.
- Chang, J., Gu, J., Wang, L., Meng, G., Xiang, S. ve Pan, C. 2018. Structure-aware convolutional neural networks. Advances in neural information processing systems, 31.
- Chen, J., Ma, T. ve Xiao, C. 2018a. Fastgcn: fast learning with graph convolutional networks via importance sampling. arXiv preprint arXiv:1801.10247.
- Chen, J., Zhu, J. ve Song, L. 2018b. Stochastic Training of Graph Convolutional Networks with Variance Reduction. In International Conference on Machine Learning, PMLR, 942-950.
- Chen, L.-C. 2014. Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv preprint arXiv:1412.7062.
- Cho, K., Van Merriënboer, B., Bahdanau, D. ve Bengio, Y. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In, October, Association for Computational Linguistics, 103-111, Doha, Qatar.
- Chollet, F. 2021. Deep learning with Python. Simon and Schuster.
- Cox, T. F. ve Cox, M. A. 2000. Multidimensional scaling. CRC press.
- Cui, Z., Henrickson, K., Ke, R. ve Wang, Y. 2019. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. IEEE Transactions on Intelligent Transportation Systems, 21(11), 4883-4894.
- De Fauw, J., Ledsam, J. R., Romera-Paredes, B., Nikolov, S., Tomasev, N., Blackwell, S., Askham, H., Glorot, X., O'donoghue, B. ve Visentin, D. 2018. Clinically applicable deep learning for diagnosis and referral in retinal disease. Nature medicine, 24(9), 1342-1350.
- Dong, Y., Chawla, N. V. ve Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 135-144.

- Dorogovtsev, S. N. ve Mendes, J. F. 2003. Evolution of networks: From biological nets to the Internet and WWW. Oxford university press.
- Dou, W., Zhang, W. ve Weng, Z. 2020. An attributed network representation learning method based on biased random walk. *Procedia Computer Science*, 174, 291-298.
- Euler, L. 1741. *Solutio problematis ad geometriam situs pertinentis*. *Commentarii academiae scientiarum Petropolitanae*, 128-140.
- Euler, L. 1956. The seven bridges of Konigsberg. Wm. Benton, 1.
- Fouss, F., Pirotte, A., Renders, J.-M. ve Saerens, M. 2007. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on knowledge and data engineering*, 19(3), 355-369.
- Gärtner, T., Flach, P. ve Wrobel, S. 2003. On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003*. *Proceedings, Springer*, 129-143.
- Gehring, J., Auli, M., Grangier, D. ve Dauphin, Y. 2017. A Convolutional Encoder Model for Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 123-135.
- Giles, C. L., Bollacker, K. D. ve Lawrence, S. 1998. CiteSeer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, 89-98.
- Girvan, M. ve Newman, M. E. 2002. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12), 7821-7826.
- Goodfellow, I., Bengio, Y. ve Courville, A. 2016. *Deep Learning*. MIT Press.
- Grigorescu, S., Trasnea, B., Cocias, T. ve Macesanu, G. 2020. A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, 37(3), 362-386.
- Grover, A. ve Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855-864.
- Hamilton, W., Ying, Z. ve Leskovec, J. 2017a. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

- Hamilton, W. L. 2020. Graph Representation Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3), 1-159.
- Hamilton, W. L., Ying, R. ve Leskovec, J. 2017b. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*.
- Hao, Y., Zhang, Y., Liu, K., He, S., Liu, Z., Wu, H. ve Zhao, J. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 221-231.
- Haonan, L., Huang, S. H., Ye, T. ve Xiuyan, G. 2019. Graph star net for generalized multi-task learning. *arXiv preprint arXiv:1906.12330*.
- He, X. ve Niyogi, P. 2003. Locality preserving projections. *Advances in neural information processing systems*, 16.
- He, Y., Li, J., Song, Y., He, M. ve Peng, H. 2018. Time-evolving Text Classification with Deep Neural Networks. In *IJCAI*, 2241-2247.
- Heidari, F. ve Papagelis, M. 2020. Evolving network representation learning based on random walks. *Applied network science*, 5(1), 18.
- Hemmatian, F. ve Sohrabi, M. K. 2019. A survey on classification techniques for opinion mining and sentiment analysis. *Artificial intelligence review*, 52(3), 1495-1545.
- Henaff, M., Bruna, J. ve Lecun, Y. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*.
- Hoang, V. T., Jeon, H.-J., You, E.-S., Yoon, Y., Jung, S. ve Lee, O.-J. 2023. Graph representation learning and its applications: a survey. *Sensors*, 23(8), 4168.
- Huang, W., Zhang, T., Rong, Y. ve Huang, J. 2018. Adaptive sampling towards fast graph representation learning. *Advances in neural information processing systems*, 31.
- Ivanov, S. ve Burnaev, E. 2018. Anonymous walk embeddings. In *International conference on machine learning*, PMLR, 2186-2195.
- Jeong, H., Néda, Z. ve Barabási, A.-L. 2003. Measuring preferential attachment in evolving networks. *Europhysics letters*, 61(4), 567.
- Jeong, J., Yun, J.-M., Keam, H., Park, Y.-J., Park, Z. ve Cho, J. 2020. div2vec: diversity-emphasized node embedding. *arXiv preprint arXiv:2009.09588*.

- Jia, Z., Tillman, B., Maggioni, M. ve Scarpazza, D. P. 2019. Dissecting the graphcore ipu architecture via microbenchmarking. arXiv preprint arXiv:1912.03413.
- Jiao, Q. ve Zhang, S. 2021. A Brief Survey of Word Embedding and Its Recent Development. 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). Published online March, 12.
- Kang, U., Tsourakakis, C. E. ve Faloutsos, C. 2009. Pegasus: A peta-scale graph mining system implementation and observations. In 2009 Ninth IEEE international conference on data mining, IEEE, 229-238.
- Kashyap, S., Kumar, S., Agarwal, V., Misra, D. P., Phadke, S. R. ve Kapoor, A. 2018. Protein protein interaction network analysis of differentially expressed genes to understand involved biological processes in coronary artery disease and its different severity. Gene Reports, 12, 50-60.
- Kearnes, S., Mccloskey, K., Berndl, M., Pande, V. ve Riley, P. 2016. Molecular graph convolutions: moving beyond fingerprints. Journal of computer-aided molecular design, 30, 595-608.
- Khan, A. ve Elnikety, S. 2014. Systems for big-graphs. Proceedings of the VLDB Endowment, 7(13), 1709-1710.
- Khoshraftar, S. ve An, A. 2024. A survey on graph representation learning methods. ACM Transactions on Intelligent Systems and Technology, 15(1), 1-55.
- Kipf, T. N. ve Welling, M. 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.
- Kovács, I. A., Luck, K., Spirohn, K., Wang, Y., Pollis, C., Schlabach, S., Bian, W., Kim, D.-K., Kishore, N. ve Hao, T. 2019. Network-based prediction of protein interactions. Nature communications, 10(1), 1240.
- Krizhevsky, A., Sutskever, I. ve Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.
- Labrinidis, A. ve Jagadish, H. V. 2012. Challenges and opportunities with big data. Proceedings of the VLDB Endowment, 5(12), 2032-2033.
- Lam, H. T., Thiebaut, J.-M., Sinn, M., Chen, B., Mai, T. ve Alkan, O. 2017. One button machine for automating feature engineering in relational databases. arXiv preprint arXiv:1706.00327.

- Lancichinetti, A., Fortunato, S. ve Radicchi, F. 2008. Benchmark graphs for testing community detection algorithms. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 78(4), 046110.
- Lang, K. (1995). Newsweeder: Learning to filter netnews, Machine learning proceedings 1995, Elsevier, 331-339.
- Lecun, Y., Bottou, L., Bengio, Y. ve Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Lewis, D. D. ve Ringuette, M. 1994. A comparison of two learning algorithms for text categorization. In *Third annual symposium on document analysis and information retrieval*, 81-93.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. ve Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In, July, *Association for Computational Linguistics*, 7871-7880, Online.
- Li, J., Zhu, J. ve Zhang, B. 2016. Discriminative deep random walk for network classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1004-1013.
- Liben-Nowell, D. ve Kleinberg, J. 2003. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, 556-559.
- Liu, K., Sun, X., Jia, L., Ma, J., Xing, H., Wu, J., Gao, H., Sun, Y., Boulnois, F. ve Fan, J. 2019a. Chemi-net: a molecular graph convolutional network for accurate drug property prediction. *International journal of molecular sciences*, 20(14), 3389.
- Liu, R., Hirn, M. ve Krishnan, A. 2023. Accurately modeling biased random walks on weighted networks using node2vec+. *Bioinformatics*, 39(1), btad047.
- Liu, Z., Chen, C., Li, L., Zhou, J., Li, X., Song, L. ve Qi, Y. 2019b. Geniepath: Graph neural networks with adaptive receptive paths. In *Proceedings of the AAAI conference on artificial intelligence*, 4424-4431.
- Malewicz, G., Austern, M. H., Bik, A. J., Dehnert, J. C., Horn, I., Leiser, N. ve Czajkowski, G. 2010. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 135-146.
- Malliaros, F. D. ve Vazirgiannis, M. 2013. Clustering and community detection in directed networks: A survey. *Physics reports*, 533(4), 95-142.

- Mccallum, A. K., Nigam, K., Rennie, J. ve Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3, 127-163.
- Meidanis, J. 2021. Network Science: a primer. *Matemática Contemporânea*, 48, 167-180.
- Micali, S. ve Zhu, Z. A. 2016. Reconstructing markov processes from independent and anonymous experiments. *Discrete Applied Mathematics*, 200, 108-122.
- Mikolov, T. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. ve Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26.
- Mohammed, S. N. ve Gündüç, S. 2022. Degree-based random walk approach for graph embedding. *Turkish Journal of Electrical Engineering and Computer Sciences*, 30(5), 1868-1881.
- Mohammed, S. N. ve Gündüç, S. 2023. TPM: Transition Probability Matrix - Graph Structural Feature based Embedding. *Kybernetika*, 59, 234-253.
- Mohammed, S. N. ve Gündüç, S. 2024. GuideWalk--Heterogeneous Data Fusion for Enhanced Learning--A Multiclass Document Classification Case. *arXiv preprint arXiv:2404.18942*.
- Morris, C., Kersting, K. ve Mutzel, P. 2017. Glocalized weisfeiler-lehman graph kernels: Global-local feature maps of graphs. In *2017 IEEE International Conference on Data Mining (ICDM)*, IEEE, 327-336.
- Nettleton, D. F. 2013. Data mining of social networks represented as graphs. *Computer Science Review*, 7, 1-34.
- Newman, M. 2010. *Networks: An Introduction*. Oxford university press, 784.
- Newman, M. E. 2003. The structure and function of complex networks. *SIAM review*, 45(2), 167-256.
- Nguyen, D. H., Nguyen, C. H. ve Mamitsuka, H. 2021. Learning subtree pattern importance for Weisfeiler-Lehman based graph kernels. *Machine Learning*, 110(7), 1585-1607.

- Nikolentzos, G., Siglidis, G. ve Vazirgiannis, M. 2021. Graph kernels: A survey. *Journal of Artificial Intelligence Research*, 72, 943-1027.
- Nt, H. ve Maehara, T. 2019. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*.
- Orsini, F., Frasconi, P. ve De Raedt, L. 2015. Graph invariant kernels. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
- Pan, S., Wu, J., Zhu, X., Zhang, C. ve Wang, Y. 2016. Tri-party deep network representation. In *International Joint Conference on Artificial Intelligence 2016, Association for the Advancement of Artificial Intelligence (AAAI)*, 1895-1901.
- Park, C., Kim, D., Han, J. ve Yu, H. 2020. Unsupervised attributed multiplex network embedding. In *Proceedings of the AAAI conference on artificial intelligence*, 5371-5378.
- Pedregosa, F. 2011. Scikit-learn: Machine learning in python Fabian. *Journal of machine learning research*, 12, 2825.
- Pennington, J., Socher, R. ve Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532-1543.
- Perozzi, B., Al-Rfou, R. ve Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 701-710.
- Perozzi, B., Kulkarni, V., Chen, H. ve Skiena, S. 2017. Don't walk, skip! online learning of multi-scale network embeddings. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, 258-265.
- Qi, X. 2022. Efficient Random Walk based Sampling with Inverse Degree. *arXiv preprint arXiv:2209.12804*.
- Raghupathi, W. ve Raghupathi, V. 2014. Big data analytics in healthcare: promise and potential. *Health information science and systems*, 2, 1-10.
- Reed, D. A. ve Dongarra, J. 2015. Exascale computing and big data. *Communications of the ACM*, 58(7), 56-68.
- Ren, S., He, K., Girshick, R. ve Sun, J. 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6), 1137-1149.

- Ribeiro, L. F., Saverese, P. H. ve Figueiredo, D. R. 2017. struc2vec: Learning node representations from structural identity. In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 385-394.
- Roweis, S. T. ve Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. science, 290(5500), 2323-2326.
- Safavi, T. ve Koutra, D. 2020. CoDEX: A Comprehensive Knowledge Graph Completion Benchmark. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 8328-8350.
- Salamat, A., Luo, X. ve Jafari, A. 2020. Balnode2vec: Balanced random walk based versatile feature learning for networks. In 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 1-8.
- Samko, O., Marshall, A. D. ve Rosin, P. L. 2006. Selection of the optimal parameter value for the Isomap algorithm. Pattern Recognition Letters, 27(9), 968-979.
- Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I. ve Welling, M. 2018. Modeling relational data with graph convolutional networks. In The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15, Springer, 593-607.
- Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B. ve Eliassi-Rad, T. 2008. Collective classification in network data. AI magazine, 29(3), 93-93.
- Sheikh, N., Kefato, Z. ve Montresor, A. 2019. gat2vec: representation learning for attributed graphs. Computing, 101(3), 187-209.
- Shervashidze, N., Schweitzer, P., Van Leeuwen, E. J., Mehlhorn, K. ve Borgwardt, K. M. 2011. Weisfeiler-lehman graph kernels. Journal of Machine Learning Research, 12(9).
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y. ve Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing, 1631-1642.
- Sundaram, K. 2023. An Analysis of the Use of Topological Graphs in Network Connectivity. Journal of Advances in Mathematics and Computer Science, 38, 87-100.

- Tabassum, S., Pereira, F. S., Fernandes, S. ve Gama, J. 2018. Social network analysis: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(5), e1256.
- Tang, J., Qu, M. ve Mei, Q. 2015a. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1165-1174.
- Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J. ve Mei, Q. 2015b. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, 1067-1077.
- Thekumparampil, K. K., Wang, C., Oh, S. ve Li, L.-J. 2018. Attention-based graph neural network for semi-supervised learning. *arXiv preprint arXiv:1803.03735*.
- Tu, K., Cui, P., Wang, X., Wang, F. ve Zhu, W. 2018. Structural deep embedding for hyper-networks. In *Proceedings of the AAAI conference on artificial intelligence*.
- Umbrey, G. ve Rahman, S. 2023. Determining Affinity of Social Network using Graph Semirings. *Qeios*.
- Van Der Maaten, L. ve Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. ve Polosukhin, I. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Curran Associates Inc., 6000–6010, California, USA.
- Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. ve Bengio, Y. 2017. Graph attention networks. *stat*, 1050(20), 10-48550.
- Wang, D., Cui, P. ve Zhu, W. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 1225-1234.
- Wang, J., Huang, P., Zhao, H., Zhang, Z., Zhao, B. ve Lee, D. L. 2018. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 839-848.
- Wang, Y., Feng, C., Chen, L., Yin, H., Guo, C. ve Chu, Y. 2019. User identity linkage across social networks via linked heterogeneous network embedding. *World Wide Web*, 22(6), 2611-2632.

- Wang, Z. ve Lu, D. 2023. The edge rings of compact graphs. arXiv preprint arXiv:2309.07587.
- Warstadt, A. 2019. Neural Network Acceptability Judgments. arXiv preprint arXiv:1805.12471.
- Weisfeiler, B. ve Leman, A. 1968. The reduction of a graph to canonical form and the algebra which appears therein. *nti, Series*, 2(9), 12-16.
- West, J. D., Wesley-Smith, I. ve Bergstrom, C. T. 2016. A recommendation system based on hierarchical clustering of an article-level citation network. *IEEE Transactions on Big Data*, 2(2), 113-123.
- Wold, S., Esbensen, K. ve Geladi, P. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3), 37-52.
- Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T. ve Weinberger, K. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, PMLR, 6861-6871.
- Xin, R. S., Gonzalez, J. E., Franklin, M. J. ve Stoica, I. 2013. Graphx: A resilient distributed graph system on spark. In *First international workshop on graph data management experiences and systems*, 1-6.
- Xiong, C., Power, R. ve Callan, J. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*, 1271-1279.
- Yang, L., Chatelain, C. ve Adam, S. 2024. Dynamic graph representation learning with neural networks: A survey. *IEEE Access*, 12, 43460-43484.
- Yao, L., Mao, C. ve Luo, Y. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, 7370-7377.
- Ye, J., Janardan, R. ve Li, Q. 2004. Two-dimensional linear discriminant analysis. *Advances in neural information processing systems*, 17.
- Ye, W., Tian, H. ve Chen, Q. 2022. Graph Kernels Based on Multi-scale Graph Embeddings. arXiv preprint arXiv:2206.00979.
- Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L. ve Leskovec, J. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 974-983.

- Zaki, M. J. ve Meira, W. 2014. Data mining and analysis: fundamental concepts and algorithms. Cambridge University Press.
- Zhang, Q., Yan, D. ve Cheng, J. 2016. Quegel: A general-purpose system for querying big graphs. In Proceedings of the 2016 International Conference on Management of Data, 2189-2192.
- Zhang, Y., Shi, Z., Feng, D. ve Zhan, X.-X. 2019. Degree-biased random walk for large-scale network embedding. Future Generation Computer Systems, 100, 198-209.
- Zhang, Y. ve Zhao, Y. 2015. Astronomy in the big data era. Data Science Journal, 14, 11-11.

