

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

KARMAŞIK AĞLARDA BASAMAKLI ÇÖKME TAHMİNİ

Ahmet TÜRK

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2019**

Her hakkı saklıdır

TEZ ONAYI

Ahmet TÜRK tarafından hazırlanan “**Karmaşık Ağlarda Basamaklı Çökme Tahmini**” adlı tez çalışması 24/07/2019 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Doç. Dr. Recep ERYİĞİT

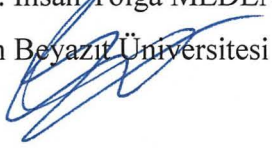


Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Jüri Üyeleri :

Başkan : Doç. Dr. İhsan Tolga MEDENİ

Yıldırım Beyazıt Üniversitesi Yönetim Bilişim Sistemleri Anabilim Dalı



Üye : Doç. Dr. Semra GÜNDÜÇ

Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı



Üye : Doç. Dr. Recep ERYİĞİT

Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı



Yukarıdaki sonucu onaylarım.

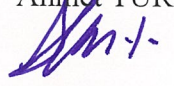
Prof. Dr. Özlem YILDIRIM
Fen Bilimleri Enstitü Müdürü

ETİK

Ankara Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

24.07.2019

Ahmet TÜRK



ÖZET

Yüksek Lisans Tezi

KARMAŞIK AĞLARDA BASAMAKLI ÇÖKME TAHMİNİ

Ahmet TÜRK

Ankara Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Recep ERYİĞİT

İletişim altyapıları, ulaşım sistemleri, sosyal ağlar ve bilgisayar ağları gibi gerçek dünyadaki birçok karmaşık sistem, karmaşık ağlar olarak ifade edilebilir. Çoğu karmaşık ağ, hatalar ve saldırılar nedeniyle basamaklı çökme durumlarıyla karşı karşıya kalabilir. Bu çökmeler, ağ bakımı ve ağ ölçeklenebilirliği için göz önünde bulundurulması gereken temel unsurlardan biridir. Basamaklı çökmelerin tahmini, bu tarz kritik ağlar için hayati öneme sahiptir. Bu çalışmada, karmaşık ağlar için düğüm tabanlı bir basamaklı başarısızlık tahmin modeli önerilmiştir. Normal dağılımlı çizgelerde, hatalar ve saldırılar simüle edilerek bir veri seti oluşturulmuştur. Düğüm önemini belirlemek için global verimlilik ve düğüm verimliliği ölçütleri kullanılmıştır., Düğümlerin başarısızlığa karşı ne derece savunmasız olduklarını tahmin etmek ve düğümleri başarısız olma mesafelerine göre sınıflandırmak için makine öğrenmesi yöntemleri kullanılmıştır. Sonuçlar, Extreme Learning Machines, Random Forest ve Long Short Term Memory modellerinin isabetli bir şekilde düğüm tabanlı basamaklı başarısızlık tahminleri için kullanılabileceğini göstermektedir.

Temmuz 2019, 60 sayfa

Anahtar Kelimeler: Karmaşık Ağlar, Basamaklı Çökme, Ağ Kararlılığı, Makine Öğrenmesi.

ABSTRACT

M.Sc. Thesis

CASCADING FAILURE PREDICTION on COMPLEX NETWORKS

Ahmet TÜRK

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

Supervisor: Assoc. Prof. Dr. Recep ERYİĞİT

There are many real-world complex systems that can be stated as complex networks, such as communication infrastructures, transportation systems, social networks and computer networks. Most complex networks are facing cascading overload failures due to errors and attacks. These failures are one of the main concerns for network maintenance and network scalability. Prediction of the cascading overload failures is vital for such critical networks. In this study, a node-based cascading failure prediction model is proposed for complex networks. A data set has been generated by simulating errors and attacks on normal distribution graphs. Global efficiency and nodal efficiency metrics were used to determine the node importance. Machine learning methods have been used to predict how nodes are vulnerable to failures and to classify the nodes based on their distance to fail. The results show that Extreme Learning Machines, Random Forest and Long Short Term Memory models can be used for accurate node-based cascading failure predictions.

July 2019, 60 pages

Keywords: Complex Networks, Cascading Failure, Network Robustness, Machine Learning.

TEŐEKKÜR

Çalıőmalarımda bana yol göstererek bilgisini ve deneyimini esirgemeyen danıőman hocam sayın Doç. Dr. Recep ERYİĐİT'e teőekkürlerimi sunarım.

Ayrıca eđitim hayatım boyunca bilgilerini bizlerden esirgemeyen bütün hocalarıma ve bana sürekli destek veren aileme saygı ve sevgilerimi sunarım.

Ahmet TÜRK

Ankara, Temmuz 2019



İÇİNDEKİLER

TEZ ONAY SAYFASI

ETİK	i
ÖZET.....	ii
ABSTRACT.....	iii
TEŞEKKÜR	iv
KISALTMALAR DİZİNİ	vii
ŞEKİLLER DİZİNİ	viii
ÇİZELGELER DİZİNİ.....	ix
1. GİRİŞ	1
2. KARMAŞIK AĞLAR.....	5
2.1 Çizge Kuramında Temel Tanımlar	7
2.2 Çizge Çeşitleri.....	8
2.3 Çizge ve Düğümler İçin Bazı Metrikler	10
2.4 Ağların Dağılımı.....	13
2.4.1 Normal dağılımlı ağlar	13
2.4.2 Scale-Free ağlar.....	14
2.4.3 Small-World ağları	15
2.5 Basamaklı Başarısızlık.....	16
2.5.1 Başarısızlık sebepleri	17
2.5.2 Basamaklı çökme simülasyonu	18
3. YAPAY ZEKA	20
3.1 Makine Öğrenmesi.....	21
3.2 Makine Öğrenmesi Süreci.....	21
3.3 Başarım Metrikleri	23
3.4 Yapay Sinir Ağları	25
3.5 Makine Öğrenmesi Yöntemleri.....	26
3.5.1 Random Forest.....	26
3.5.2 ELM	27
3.5.3 LSTM	29
4. MATERYAL ve YÖNTEM.....	31
4.1 Veri Setinin Oluşturulması	31
4.1.1 Deneysel ağların dağılımı.....	33

4.1.2 Deney ađlarına saldırılar	35
4.2 Gerçekleřtirilen Deneyler	38
4.2.1 Çökme tahmini	39
4.2.1.1 Random Forest uygulaması.....	40
4.2.1.2 ELM uygulaması	41
4.2.1.3 LSTM uygulaması.....	43
4.2.2 Çökme tahmini yöntemlerinin karşılařtırılması	45
4.2.3 Çökme uzaklıđı tahmini.....	47
4.2.3.1 Random Forest uygulaması.....	48
4.2.3.2 ELM uygulaması.....	48
4.2.3.3 LSTM uygulaması.....	49
4.2.4 Tařma sınırı deđerinin etkisinin test edilmesi	50
4.2.5 Parametre etkisinin test edilmesi.....	52
5. SONUÇLAR	54
5.1 Öneriler.....	55
KAYNAKLAR	57
ÖZGEÇMİŐ.....	60

KISALTMALAR DİZİNİ

ELM	Extreme Learning Machine
LSTM	Long Short Term Memory
RAM	Random Access Memory
CPU	Central Processing Unit
GB	Giga Byte



ŞEKİLLER DİZİNİ

Şekil 2.1 Altı düğüm ve dokuz kenardan oluşan basit çizge (G1).....	5
Şekil 2.2 Altı düğüm ve beş kenardan oluşan yönsüz çizge (G2).....	6
Şekil 2.3 Sırasıyla 1, 2, 3, 4, 5, 6, 7, 8 ve 9 numaralı çizge örnekleri.....	10
Şekil 2.4 Normal dağılımlı ağ örneği dereceler dağılımı.....	14
Şekil 2.5 Scale-Free ağ örneği dereceler dağılımı	15
Şekil 2.6 Small-World ağ örneği dereceler dağılımı.....	16
Şekil 2.7 500 elemanlı bir ağda saldırı etkisi	19
Şekil 2.8 1000 elemanlı bir ağda saldırı etkisi	19
Şekil 3.1 Eğitimde karmaşıklık hata ilişkisi.....	22
Şekil 3.2 İnsan sinir hücresi bileşenleri	25
Şekil 3.3 İnsan sinir hücresi - yapay sinir ağı sinir hücresi.....	25
Şekil 3.4 Yapay sinir ağı genel modeli	26
Şekil 3.5 Random Forest modeli.....	27
Şekil 3.6 ELM modelinin genel yapısı.....	28
Şekil 3.7 LSTM ünitesi	29
Şekil 4.1 433 numaralı testte kullanılan ağın dereceler dağılımı grafiği	34
Şekil 4.2 Basamaklı başarısızlık adım sayıları.....	37
Şekil 4.3 Random Forest ağaç sayısı - R Kare doğruluk ilişkisi.....	40
Şekil 4.4 ELM modeli karmaşıklık - doğruluk ilişkisi.....	42
Şekil 4.5 LSTM modeli karmaşıklık - doğruluk ilişkisi	44

ÇİZELGELER DİZİNİ

Çizelge 2.1 G1 çizgesi komşuluk matrisi.....	6
Çizelge 2.2 G2 çizgesi komşuluk matrisi.....	6
Çizelge 2.3 Çizge sınıfları.....	10
Çizelge 4.1 Örnek basamaklı başarısızlık durumları	32
Çizelge 4.2 Deney ağ grupları - 1	33
Çizelge 4.3 Deney ağ grupları - 2	33
Çizelge 4.4 433 numaralı testte kullanılan ağın dereceler dağılımı	34
Çizelge 4.5 Başarısızlığa uzaklık etiketleri	38
Çizelge 4.6 Başarısızlık durumları	38
Çizelge 4.7 Random Forest ikili sınıflandırma karışıklık matrisi	41
Çizelge 4.8 ELM ikili sınıflandırma karışıklık matrisi	43
Çizelge 4.9 LSTM ikili sınıflandırma karışıklık matrisi	44
Çizelge 4.10 Çökme tahmini yöntemleri doğruluk karşılaştırması.....	46
Çizelge 4.11 Random Forest çökme uzaklığı sınıflandırma karışıklık matrisi	48
Çizelge 4.12 Random Forest yöntemi çökme uzaklığı sınıflandırma başarımı	48
Çizelge 4.13 ELM çökme uzaklığı sınıflandırma karışıklık matrisi	49
Çizelge 4.14 ELM yöntemi çökme uzaklığı sınıflandırma başarımı	49
Çizelge 4.15 LSTM çökme uzaklığı sınıflandırma karışıklık matrisi.....	49
Çizelge 4.16 LSTM yöntemi çökme uzaklığı sınıflandırma başarımı	50
Çizelge 4.17 2 numaralı veri seti başarısızlık tahmini sonucu.....	50
Çizelge 4.18 2 numaralı veri seti başarısızlık tahmini karışıklık matrisi.....	50
Çizelge 4.19 2 numaralı veri seti başarısızlığa uzaklık tahmini sonucu	51
Çizelge 4.20 Birleşik veri seti başarısızlık tahmini sonucu	51
Çizelge 4.21 Birleşik veri seti başarısızlık tahmini karışıklık matrisi	51
Çizelge 4.22 Birleşik veri seti başarısızlığa uzaklık tahmini sonucu.....	52
Çizelge 4.23 Düğüm parametreleri ikili sınıflandırma karışıklık matrisi	53
Çizelge 4.24 Düğüm parametreleri etkisi testi sonuçları	53

1. GİRİŞ

Gerçek dünyada, karmaşık ağ olarak adlandırılabilen birçok sistem bulunmaktadır. Enerji hatları, sosyal medya ortamları, bilgisayar ağları, web servisleri, karayolu ulaşım hatları gibi birçok ağ örneği için saldırı riski bulunmaktadır. Özellikle iletişim ve ulaşım sistemleri saldırılara sıklıkla konu olmaktadır. Bu saldırılarda amaç, ağı kullanılamaz hale getirmektir. Ağ için kritik öneme sahip noktalara yapılan saldırılar, önü alınamaz, birbirini tetikleyen çökmelere sebebiyet vererek ağın işlevini yitirmesine sebep olabilir. Bu gibi durumlara basamaklı başarısızlık veya basamaklı çökme (cascading failures) adı verilmektedir (Crucitti vd. 2004). Bir sosyal medya ortamında çok sayıda takipçisi bulunan birkaç hesap ele geçirilerek yapılan kötücül bir yayın kısa sürede geniş kitlelere gösterilebilir. Takip eden süreçte takipçiler içindeki belirli bir oranda kullanıcı bu kötücül yayını açarak saldırganların yeni kurbanı olur. Bu şekilde onlardan yapılan yayınlarla daha da geniş kitlelere yayılan kötücül yayın kendi kendini besleyerek sosyal medya ağında basamaklı bir çökmeye neden olur. Yine bir karmaşık ağ olarak düşünülebilen ülkeler arası ticaret ağında, kritik ticaret anlaşmaları bozularak veya değiştirilerek ilgili tedariklerin farklı kaynaklardan yapılması sağlanabilir. Ülke ihtiyaçları büyük oranda sabit olduğundan bazı ticaret yönlerinin değişmesi, farklı ticaret yönlerinin de değişmesine sebep olarak belli kalemler için ekonomi ve tedarik anlamında basamaklı çökme durumu gerçekleşebilir.

Saldırı dışında, doğal hata durumlarıyla da basamaklı çökme gerçekleşebilir. Doğal hatalar bütün sistemler için hayatın bir gerçeğidir ve tamamen önlenemez bir durumdur. Rastlantısal olarak gerçekleşen hatalar sonucu, ağ kendi kendini tetikleyen hatalara sebebiyet verebilir. Gerek doğal hatalar sonucu gerekse saldırılarla, gerçek dünyadaki birçok karmaşık ağ örneğinde basamaklı çökme durumu söz konusu olabilir.

Gerçek hayat örnekleri düşünüldüğünde basamaklı çökme durumunun önceden kestirilmesi hayati öneme sahiptir. Hem sistemlerin korunması ve bakımından sorumlu ekipler için hem de sistemlere saldırıda bulunacak gruplar için bu sistemde basamaklı çökmeye sebep olacak senaryoların iyi bilinmesi gerekir. Bu çalışmada, saldırı sonucu oluşabilecek bir basamaklı çökme durumunda, ağdaki her eleman için bireysel çökme tahmini yapabilen bir model önerilmektedir.

Yapılan çalışmalarda basamaklı çökme tahminine yönelik farklı yöntemler denenmiş ve farklı sonuçlar ortaya konulmuştur. Bu çalışmalar temel olarak 3 ana grupta toplanabilir. Basamaklı çökme modelleri (Motter ve Lai 2002, Albert vd. 2000), ağlara saldırı stratejileri (Holme vd. 2002, Zhao vd. 2004) ile basamaklı çökmenin kontrol edilmesi ve savunma teknikleri (Motter 2004, Ash ve Newth 2007, Majdandzic vd. 2014) bu üç grubun odak noktaları olmuştur.

Shunkun vd. (2016) tarafından yapılan çalışmada büyük ölçekli ağlarda basamaklı çökmenin tahmin edilmesi adına sanal olarak ağlar oluşturularak veri toplanmış ve çeşitli makine öğrenmesi yöntemleriyle deneyler gerçekleştirilmiştir. 50x50 boyutunda, kare şeklinde yerleşmiş 2500 düğümlü ağlar üretilmiş ve komşu düğümler arasındaki bağlar Gaus dağılımına göre rastgele oluşturulmuştur. Merkezdeki 16 düğüm çizgeden çıkartılarak basamaklı çökmelerin meydana gelmesi sağlanmıştır. Düğüm başarısızlığı için düğümün arasındalık merkeziliğinin ilk duruma göre belirli bir yüzdeden fazla artmış olması etken olmuştur. Üretilen veri seti kullanılarak merkeze uzaklıklarına göre (1 ile 23 arası) hangi mesafede kaç düğümün başarısız olacağı bilgisi çıkartılmaya çalışılmıştır. Farklı makine öğrenmesi yöntemleri kullanılarak 0.04052 hata oranından 0.02688 hata oranına kadar çözümler geliştirilmiştir.

Havlin ve Kenett (2015) tarafından ekonomik ağlar, birçok alt ağdan oluşan kompleks bir ağ olarak incelenmiş ve ekonomik anlamda olası gelişmeler sonucunda meydana gelebilecek değişimlerin modellenmesi için bir ihtiyacı ortaya koymuştur. Ağların ağı olarak ifade edilen ekonomik ağdaki bir düğümde meydana gelebilecek bir başarısızlığın, basamaklı çökmeye sebebiyet verebileceğini kurdukları modelle iddia etmişlerdir.

Yi vd. (2014) tarafından yapılan çalışmada ise basamaklı çökme durumu sosyal ağlara yapılan saldırılar üzerinden incelenmiştir. Twitter ve Sina Weibo sosyal medya platformlarına ait gerçek veri seti üzerinde 4 farklı strateji ile sanal saldırılar yapılarak basamaklı çökmeler oluşturulmuş ve senaryolar sonucu ortaya çıkan durumlar incelenmiştir. Düğüm başarısızlığı için arasındalık merkeziliği metriği kullanılmıştır. Bunların yanı sıra son yıllarda farklı alanlarda basamaklı başarısızlıkların etkisinin incelendiği çeşitli çalışmalar yapılmıştır. Enerji hatları (Soltan vd. 2014, Hines vd.

2009), iletişim hatları (Zhu vd. 2016) ve karmaşık altyapı sistemleri (Duenas-Osorio ve Vemuru 2009) bu alanlardan bazılarıdır.

Pi vd. (2018) tarafından yapılan çalışmada enerji sistemlerinde oluşan basamaklı çökmeler incelenmiş ve enerji hatlarındaki basamaklı çökmenin yayılımının makine öğrenmesi yöntemleriyle tahminine dair ITEPV adında bir model sunulmuştur. Tecrübe adı verilen deneyler sonucunda bir Bayes ağı kurulmuş ve basamaklı çökme yayılımı tahmini gerçekleştirilmiştir. Çalışmada düğümlerin çökme yayılımıyla ilişkilendirilip ilişkilendirilmeyeceğine karar verilmiş ve başarı oranı verilmeden model belirli senaryolar üzerinden açıklanmış ve değerlendirilmiştir.

Schäfer vd. (2018) tarafından yapılan çalışmada da benzer şekilde altyapı ağlarında büyük ölçekli ağ kesintileri incelenmiştir. Elektrik iletim hatlarına odaklanılarak, basamaklı çökmelerin etkinlik tabanlı doğası ve ağ dinamik özellikleri göz önüne alınarak bir model geliştirilmiştir. Bu şekilde basamaklı çökme tahmin yöntemleri uygulanarak farklı Avrupa ülkelerinin ulusal elektrik altyapılarındaki basamaklı çökme ilişkileri vurgulanmıştır. Yine çalışma kapsamında, ağ için kritik yolların tespit edilmesi sağlanmış ve basamaklı çökmenin yayılımı incelenmiştir.

İncelenen çalışmalarda basamaklı başarısızlıkların nasıl oluştuğu araştırılmış, bazen de bu başarısızlıkların tahmini yapılmıştır. Son yıllarda yapılan, sınıflandırma ve değer tahmini içeren çalışmaların çoğunda (Shunkun vd. 2016, Bhatia vd. 2015, Zarrabian vd. 2016, Pi vd. 2018) makine öğrenmesi yöntemlerinin kullanıldığı görülmüştür. Çalışmalar çoğunlukla ağın geneliyle alakalı tahminlerde bulunmaktadır.

Bu tez kapsamında yapılan çalışmada ise ağın geneli yerine düğüm bazlı başarısızlık tahminine yönelik yöntemler geliştirilmiştir. Basamaklı başarısızlıklar, doğal sebepler veya saldırılar sonucunda gelişebilmektedir. Bu çalışmada saldırılar sonucu oluşabilecek basamaklı başarısızlıklara odaklanılarak; her düğümün, basamaklı çökmenin kendisini nasıl ve ne zaman etkileyeceğini hesaplayabileceği şekilde makine öğrenmesi yaklaşımları uygulanmıştır. Extreme Learning Machine (ELM), Random Forest (RF) ve Long Short Term Memory (LSTM) olmak üzere 3 farklı makine öğrenmesi yöntemi ile sınıflandırmalar yapılmış ve bu yöntemlerin karşılaştırılmasına olanak sağlanmıştır. Son yıllarda bu alanda yapılan çalışmalar incelendiğinde ELM ve

LSTM yöntemlerinin uygulandıđı ciddi bir alıřma bulunamadıđı iin bu iki yntem karřılařtırma yapılmak zere eklenmiř ve bu yntemlerle birlikte Random Forest yntemi bařarım karřılařtırması iin alıřmada kullanılmıřtır.

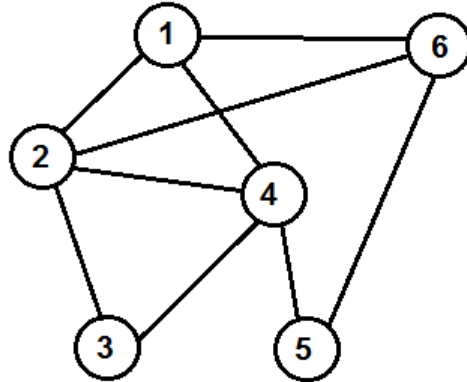
Bu iřlem iin sanal olarak oluřturulan farklı yođunluklardaki ađlara saldırılar dzenlenmiř ve oluřan durum kaydedilerek veri setleri oluřturulmuřtur. Kaydedilen veri seti, makine đrenmesi yntemlerine girdi olarak kullanılarak ađlardaki dđmlerin bařarısız olup olmayacađı ve eđer bařarısız olunacađı tahmin ediliyorsa ka adım sonra bařarısız olunacađı ıkarımları yapılmıřtır.



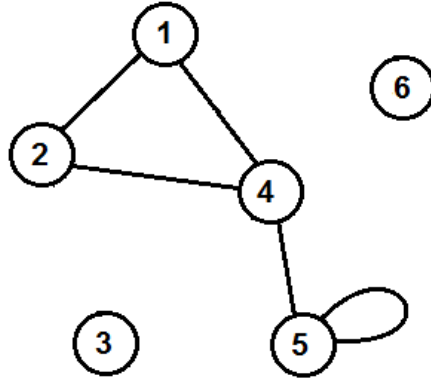
2. KARMAŞIK AĞLAR

Karmaşık ağlar, gerçek dünyadan çeşitli sistemleri ifade etmek için kullanılan çizgelerdir (graph). Basit çizgeler, düğümlerin komşuluk sayıları bakımından bir düzene sahipken, karmaşık ağları basit çizgelerden ayıran en önemli özellikler bu ağların önemsiz olmayan topolojik özellikler içermesi ve istatistiksel olarak düzenli olmamalarıdır. Karmaşık ağ kavramı, 1990'lı yıllardan itibaren bilgisayar ağları gibi gerçek dünya sistemlerini modellemek için kullanılabilir oluşu ile popüler olmaya başlamıştır. Belirli sayıda bileşenden ve bu bileşenler arasındaki ilişkilerden oluşan bu ağlar; fiziksel, kimyasal, sosyal ağları ifade etmek için kullanılır (Albert ve Barabasi 2002). Örneğin bir karayolu sistemi, bilgisayar ağı veya elementlerin birbirleriyle olan bağları birer karmaşık ağ örneğidir. Karmaşık ağlarda bileşen ve bağlantı sayıları genellikle ne tamamen sabittir ne de tamamen rastgeledir, yani koşullu değişkendir.

Karmaşık ağ tanımı, gerçek dünya olgularını tanımlamak için kullanılan bir yöntemi ifade eder. Sosyal ilişkiler, elektrik, su, doğalgaz altyapıları, bilgisayar ağları, karayolu, denizyolu, demiryolu, havayolu gibi ulaşım sistemleri, sosyal medya etkileşimleri gibi gerçek dünya ilişkileri karmaşık ağlar olarak ifade edilebilir. Karmaşık ağları matematiksel olarak ifade etmek için Çizge Kuramı öğretisi kullanılır. Bu şekilde karmaşık ağlar matematiksel olarak modellenebilir. Bu yapı, düğüm(köşe) adı verilen bileşenlerden ve bu bileşenler arasındaki ilişkiyi ifade eden kenarlardan(kirişlerden) oluşur (Diestel 2000). Şekil 2.1'de 6 düğüm ve 9 kenardan oluşan basit bir çizge bulunmaktadır. Şekil 2.2'de ise 6 düğüm ve 5 kenardan oluşan yönsüz bir çizge bulunmaktadır.



Şekil 2.1 Altı düğüm ve dokuz kenardan oluşan basit çizge (G1)



Şekil 2.2 Altı düğüm ve beş kenardan oluşan yönsüz çizge (G2)

Çizgeler grafik olarak bu şekilde gösterilebildiği gibi komşuluk matrisi şeklinde de ifade edilebilir. Şekil 2.1 ve şekil 2.2'deki G1 ve G2 çizgeleri için komşuluk matrisleri çizelge 2.1 ve çizelge 2.2'de verilmiştir.

Çizelge 2.1 G1 çizgesi komşuluk matrisi

	1	2	3	4	5	6
1	0	1	0	1	0	1
2	1	0	1	1	0	1
3	0	1	0	1	0	0
4	1	1	1	0	1	0
5	0	0	0	1	0	1
6	1	1	0	0	1	0

Çizelge 2.2 G2 çizgesi komşuluk matrisi

	1	2	3	4	5	6
1	0	1	0	1	0	0
2	1	0	0	1	0	0
3	0	0	0	0	0	0
4	1	1	0	0	1	0
5	0	0	0	1	1	0
6	0	0	0	0	0	0

Sosyal ilişkiler örneğinde; düğümler insanları, kenarlarsa bunlar arasındaki arkadaşlık, akrabalık gibi ilişkileri temsil edebilir. Bilgisayar ağları örneğinde; düğümler bilgisayarlar, yönlendiriciler, sunucular gibi elektronik bileşenleri, kenarlarsa bu elektronik cihazlar arasındaki fiziksel bağlantıları ifade eder. Havayolu ulaşımı örneğinde ise; düğümler havalimanlarını, kenarlarsa bu havalimanları arasındaki

uçuşları ifade eder. Gerçek dünya ilişkileri, bu örneklere benzer şekilde, Çizge Kuramı yardımıyla modellenilebilir.

Çizge kuramındaki temel bazı kavramlar ve çizge çeşitleri Diesel'e (2000) göre şu şekilde ifade edilmiştir.

2.1 Çizge Kuramında Temel Tanımlar

Çizge: Ağ ifade etmek için kullanılan yapıdır. $G = (V, E)$ olarak ifade edilir. Boş ve sonsuz olmayan bir düğümler ve kenarlar kümesidir. Gerçek dünyadan bir ağ örneğini ifade edebilir. Örneğin Türkiye havayolları ağı bir çizge olarak ifade edilebilir. Şekil 2.1 bir çizgeyi göstermektedir.

Düğüm (Köşe): Çizgenin temel elemanıdır. Düğümler u, v gibi harflerle, düğüm kümesi $V(G)$ olarak ifade edilir. Havayolu ağı için, havalimanları birer düğüm olarak gösterilir. Şekil 2.1'de 1, 2, 3, 4, 5 ve 6 sayılı gösterimler birer düğümdür.

Kenar (Kiriş): İki ucu düğümde sonlanan bağıntıdır. İki farklı düğümü birbirine veya tek bir düğümü kendisine bağlar. Kenarlar $\{u,v\}$ veya uv şeklinde, kenar kümesi $E(G)$ olarak ifade edilir. Havayolu ağı için uçuş tarifeleri birer kenar olarak gösterilir. Şekil 2.1'de $\{1,2\}$, $\{1,4\}$, $\{1,6\}$, $\{2,3\}$, $\{2,4\}$, $\{2,6\}$, $\{3,4\}$, $\{4,5\}$ ve $\{5,6\}$ gösterimleri birer kenardır.

Komşuluk: Eğer bir çizgedeki iki düğüm(u ve v), bu çizgedeki bir $e=\{u, v\}$ kenarının uçlarıysa, u ve v birbirine komşudur. Şekil 2.1'de 1 ve 2 düğümleri komşu iken 4 ve 6 düğümleri komşu değildir.

Bağlılık: Bir çizgedeki $e=\{u, v\}$ kenarı, bu kenarın uçları olan u ve v düğümleri ile bağlıdır. Bu kenar için; e, u ile bağlıdır ve e, v ile bağlıdır denilir. Şekil 2.1'de $\{1,2\}$ kenarı 1 düğümü ile bağlıdır ancak 3 düğümü ile bağlı değildir.

Mertebe: Bir çizgedeki düğüm sayısı, o çizgenin mertebesi olarak ifade edilir. $|V(G)|$ şeklinde gösterilir. Şekil 2.1'deki çizgenin mertebesi 6 olarak görülmektedir.

Boyut: Bir çizgedeki kenar sayısı, o çizgenin boyutu olarak ifade edilir. $|E(G)|$ şeklinde gösterilir. Şekil 2.1'deki çizgenin boyutu 9 olarak görülmektedir.

Düğüm derecesi: Bir çizgedeki v düğümünün komşu sayısı o düğümün derecesi olarak ifade edilir. v düğümünün derecesi $\deg(v)$ şeklinde gösterilir. Şekil 2.1'deki çizgede 2 düğümünün derecesi 4, 3 düğümünün derecesi 2 olarak görülmektedir.

Tecrit edilmiş düğüm: Bir çizgedeki v düğümünün hiçbir komşusu yoksa yani $\deg(v)=0$ olarak gösteriliyorsa o düğüme tecrit edilmiş düğüm denir. Şekil 2.2'de 3 ve 6 düğümleri tecrit edilmiş düğümlerdir.

Parkur: Bir çizgede düğümlerin kenarlar ile birer değişmeli dizisine parkur denir. Parkurlar, geçilen düğümlerin sırasıyla $\{u,v,w\}$ şeklinde gösterilir. Şekil 2.1'deki çizge için $\{1,6,2,1,6,5\}$ bir parkur oluşturur.

Yol: Bir çizge üzerinde bir ya da daha fazla düğümden ve kenardan geçen, düğümleri tekrar etmeyen parkura yol adı verilir.. Şekil 2.1'deki çizgede $\{1,6,5,4\}$ bir yolu ifade eder.

Döngü: Bir çizgede, başladığı düğümden biten yollar için döngü ifadesi kullanılır. Şekil 2.1'deki çizgede $\{2,3,4,5,6,2\}$ yolu aynı zamanda bir döngüdür.

Çevrim: Çizgedeki bir $e=\{u, v\}$ kenarı için $u=v$ ifadesi doğruysa, yani kenarın başlangıç ve bitiş düğümleri aynıysa bu kenara çevrim denir. Şekil 2.2'de $\{5,5\}$ kenarı bir çevrimdir.

2.2 Çizge Çeşitleri

Yönsüz Çizge: Bir çizgede, u ve v şeklindeki her düğüm çifti için u 'dan v 'ye ve v 'den u 'ya doğrudan erişim varsa bu çizgeye yönsüz çizge denir. Bu çizgelerin bütün kenarları yönsüzdür yani her iki ucunun birbirine komşuluğu vardır. Gerçek dünyadaki arkadaşlık ilişkileri bu tip çizgelere örnektir. Bir A kişisi, B kişisine arkadaşsa, B kişisi de A kişisine arkadaştır.

Yönlü Çizge: Bir çizgede, u ve v şeklindeki her düğüm çifti için u 'dan v 'ye ve v 'den u 'ya doğrudan erişim olmayabiliyorsa bu çizgeye yönlü çizge denir. Bu çizgelerin $\{uv\}$ şeklindeki kenarları yalnızca u düğümünü v düğümüne bağlarken, v düğümünü u düğümüne bağlamaz. Çizgede bu kenarlar bağlantı yönünde okla gösterilir. Gerçek dünyadaki karayolu ağları bu tip çizgelere örnektir. Bir A kavşağından B kavşağına yol

varken, B kavşağından A kavşağına yol olmayabilir. Tek yönlü yollar bu ağların yönsüz çizge ile ifade edilmesini engeller.

Yönlü çizgelerde derece kavramı, yönsüz çizgelerden biraz daha farklıdır. Herhangi bir v düğümünden çıkarak diğer düğümlere veya kendisine olan kenarların sayısı dış derece olarak ifade edilir ve $\deg^+(v)$ şeklinde gösterilirken, diğer düğümlerden veya kendisinden bu düğüme olan kenarların sayısı iç derece olarak ifade edilir ve $\deg^-(v)$ olarak gösterilir.

Basit Çizge: Aynı düğümler arasında birden fazla kenar olmayan, çevrim içermeyen yönsüz çizgelere basit çizgeler denir.

Çok Katlı Çizge: Herhangi iki düğümü arasında birden fazla kenar olan ve çevrim içermeyen çizgelere çok katlı çizge denir.

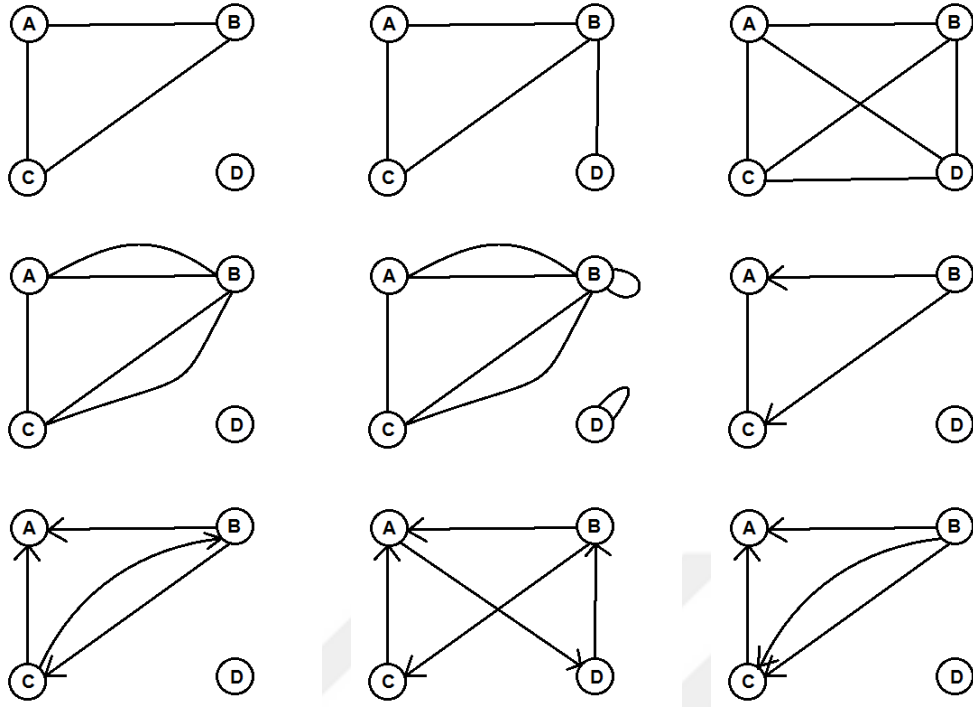
Sözde Çizge: Çevrim içeren çok katlı çizgelere sözde çizge denir.

Çok Katlı Yönlü Çizge: Herhangi iki u ve v düğümü arasında birden fazla $\{u,v\}$ kenarı olan yönlü çizgelere çok katlı yönlü çizge denir.

Tam Çizge: Bütün düğümlerin birbirleri ile komşu olduğu çizgelere tam çizge denir.

Bağlı Çizge: Bir çizgede herhangi bir düğümden diğer bütün düğümlere en az bir yol varsa bu çizgeye bağlı çizge denir.

Şekil 2.3'deki örnek çizgeler, çizge çeşitlerini görsel olarak ifade etmek için kullanılabilir. Bu çizgelerin sınıfları çizelge 2.3'de verilmiştir.



Şekil 2.3 Sırasıyla 1, 2, 3, 4, 5, 6, 7, 8 ve 9 numaralı çizge örnekleri

Çizelge 2.3 Çizge sınıfları

Çizge No	Yönsüz Çizge	Yönlü Çizge	Basit Çizge	Çok Kath Çizge	Sözde Çizge	Tam Çizge	Bağlı Çizge
1	+		+				
2	+		+				+
3	+		+			+	+
4	+			+			
5	+				+		
6		+					
7		+					
8		+					+
9		+		+			

2.3 Çizge ve Dügümler İçin Bazı Metrikler

Yoğunluk: Ağdaki elemanların ne kadar sık yerleştiğinin yani ağın ne kadar yoğun olduğunun göstergesi yoğunluk (density) olarak adlandırılır. Ağdaki elemanlar birbirleriyle çok sayıda bağlantı kurmuşsa bu ağ yoğun bir ağ olarak düşünülür ve

elemanlar arasındaki mesafeler kısadır. Ağdaki elemanlar arasında az sayıda bağlantı olduğundaysa bu ağ için seyrek ağıdır diyebiliriz. Çizgedeki bütün düğümler arasında kenar varsa bu ağın yoğunluğu 1 olur. Ağın yoğunluğu şu şekilde hesaplanır:

$$D(G) = \frac{2 E(G)}{V(G)(V(G)-1)} \text{ Yönlü çizgeler için}$$

$$D(G) = \frac{E(G)}{V(G)(V(G)-1)} \text{ Yönsüz çizgeler için}$$

Burada $D(G)$, G ağının yoğunluğunu ifade ederken, $E(G)$ G ağının kenar sayısı ve $V(G)$ G ağının düğüm sayısıdır.

Verim: Ağ üzerinde herhangi iki eleman arasındaki iletişimin etkinliği verim (efficiency) olarak ifade edilir (Latora ve Marchiori 2001). Verim değeri yüksek olan ağlarda, elemanlar arasındaki aktarım daha kısa yoldan yapılabilmektedir. Gerçek dünyadaki karayolu ağını düşünecek olursak, bütün şehirlerden diğer şehirlere ulaşımın kısa olduğu karayolu ağının verimi yüksek, şehirler arasındaki ulaşımın daha uzun sürdüğü ağ içinse verimi düşüktür diyebiliriz. Çizge üzerinde bir v düğümünden u düğümüne gidilebilecek yolların en kısa olanına u ile v arasındaki en kısa yol denir ve d_{uv} olarak gösterilir. Global verim, ağın geneli için önemli bir ölçüttür ve çizgedeki en kısa yollar kullanılarak şu şekilde hesaplanır:

$$Verim_{global} (G) = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{d_{ij}}$$

Denklemden N çizgedeki düğüm sayısını, d_{ij} ise çizgedeki bütün i,j düğüm ikilileri için bu düğümler arasındaki en kısa yol değerini ifade eder. Düğümler arasındaki en kısa yollar azaldıkça formüldeki ters orantı sayesinde ağın verimi artacaktır.

Ağdaki her eleman için düğüm verimi de benzer şekilde hesaplanabilir. Çizgedeki herhangi bir düğüm için düğüm verimi, bu düğümle çizgedeki diğer bütün düğümler arasındaki en kısa yolun ters ortalamasıdır. Formülde i düğümü için kendisi hariç bütün diğer düğümlerle (j) olan en kısa yol değerleri d_{ij} ile ifade edilmiştir.

$$Verim_{düğüm}(G_i) = \frac{1}{N-1} \sum_j \frac{1}{d_{ij}}$$

Çizgedeki herhangi bir düğümün komşularından oluşan alt çizge için, çizge içi en kısa yollar kullanılarak lokal verim hesaplanabilir. G_i , i düğümünün bütün komşularından oluşan ve i düğümünü içermeyen bir alt çizgedir. Formülde NG_i , G_i alt çizgesinin eleman sayısını, d_{jk} ise bu alt çizgedeki farklı her iki düğüm için en kısa yolu ifade eder.

$$Verim_{lokal}(G_i) = \frac{1}{NG_i(NG_i-1)} \sum_{j \neq k} \frac{1}{d_{jk}}$$

Lokal verim, lokal verimin hesaplandığı düğümde bir başarısızlık durumunda ağın ne kadar dirençli olduğunun göstergesidir. Çünkü bu düğüm çizgeden çıkarıldığında, üzerinden geçen yollar, hesaplama yapılan alt çizgeden alternatifleriyle karşılanacaktır. Herhangi bir düğüm çizgeden çıkarıldığında, diğer düğümlerin iletişimlerine etkin şekilde devam edebilmeleri lokal verim değerinin yüksek olması anlamında gelir. Alt ağın hata toleransı, lokal verimle doğru orantılıdır. Bu yüzden çizgede hangi düğümün çizge için ne kadar önemli olduğunu hesaplamak için lokal verim ve düğüm verimi kullanılabilir.

Arasındalık Merkeziliği: Ağdaki elemanların ağ için önemini belirtmek için kullanılan bir diğer metrik arasındalık merkeziliğidir (betweenness centrality) (Freeman 1979). Bu metrik, çizgedeki herhangi bir düğümün, diğer düğümlerin iletişimindeki önemini ortaya çıkartır. Bu şekilde çizgeden bu düğümün çıkarılması sonucu çizgenin ne ölçüde etkileneceği tahmin edilebilir. Formülde i düğümü için arasındalık değerinin nasıl hesaplandığı görülmektedir. σ_{st} çizgedeki bütün s ve t ikili düğümleri arasındaki en kısa yolların sayısı, $\sigma_{st}(i)$ ise bu en kısa yolların kaç tanesinin i düğümünü içerdiği bilgisidir. Arasındalık değeri hesaplanan düğüm bu işlemlerin dışında tutulur.

$$Arasındalık(i) = \sum_{s \neq t \neq i} \frac{\sigma_{st}(i)}{\sigma_{st}}$$

Yakınlık Merkeziliği: Ağdaki elemanların ağ için önemini belirlemek için kullanılan metriklerden bir diğeri de yakınlık merkeziliğidir (closeness centrality) (Freeman 1979). Çizgedeki herhangi bir düğümün diğeri düğümlerle olan en kısa yollarına bakılarak düğümün yakınlık merkeziliğine karar verilir. Bir düğüm diğeri düğümlere ne kadar kısa yoldan ulaşabiliyorsa yakınlık değeri o kadar büyük olur. Formülde i düğümü için diğeri bütün düğümlerle olan en kısa yollarının toplamının tersinin yakınlık değerine karşılık geldiği görülmektedir.

$$Yakınlık(i) = \frac{1}{\sum_j d_{ij}}$$

Aynı ağ içinde düğümler arasında kıyaslama yapmak için bu formül doğrudan uygulanabilir ancak farklı düğüm sayısına sahip ağlar arasında düğüm kıyaslaması yapabilmek için çıkan yakınlık merkeziliği değeri ağdaki düğüm sayısıyla çarpılarak kullanılabilir.

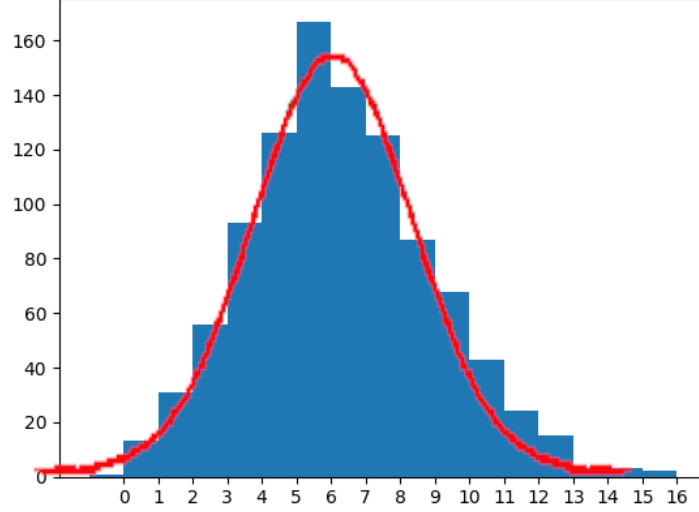
2.4 Ağların Dağılımı

Ağlar, yapılarına göre birkaç dağılım tipinde değerlendirilir. Ağdaki elemanların komşuluk sayıları, ağın hangi dağılım tipine sahip olduğunu belirler. Sıkça kullanılan dağılım tiplerinden bazıları şunlardır.

2.4.1 Normal dağılımlı ağlar

Aynı zamanda Gaus Dağılımı ve Çan Eğrisi olarak da bilinir (Sanner ve Slotine 1992). Ağdaki elemanların komşuluk sayılarının, bu sayıların ortalama değerinin etrafında simetriğe yakın şekilde ayrıştığı durumlarda, ağ için Normal Dağılıma sahiptir denir. Matematikteki olasılıksal dağılım modelinin, ağ için düğümlerin komşuluk sayılarına uyması durumu olarak da ifade edilebilir. Düğümlerin dereceleri, ortalama değer etrafında yoğun şekildedir ve bu değerden her iki tarafa doğru uzaklaştıkça yoğunluk azalır. Şekil olarak ifade edildiğinde çan benzeri bir görüntü oluşturur. Gerçek dünya ilişkileri dağılım konusunda genellikle simetrikliği sağlamaz ancak bu dağılıma benzerlik gösterir. Şekil 2.4'de normal dağılım özelliklerindeki bir ağın dereceler dağılımı bilgisi görülmektedir. Bu şekil için ortalama komşuluk sayısı 6.78 olarak

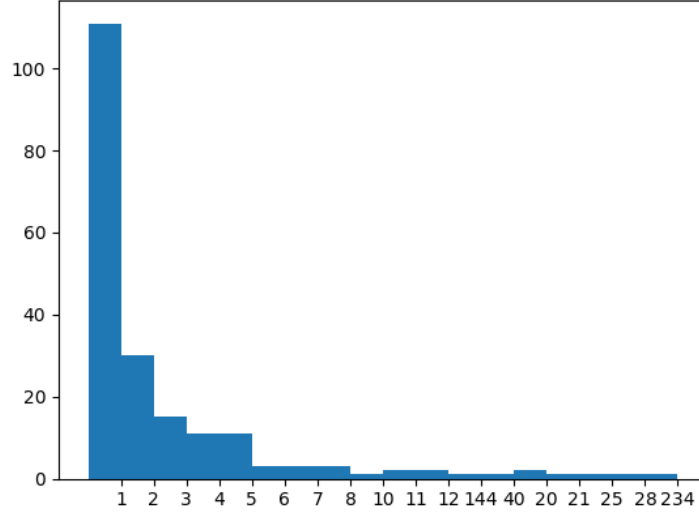
gerçekleşmiştir ve yoğunluk bu değerin etrafındadır. Bu değerden azalma ve artma yönünde uzaklaştıkça, o değerde komşuluğa sahip düğümlerin azaldığı görülmektedir.



Şekil 2.4 Normal dağılımlı ağ örneği dereceler dağılımı

2.4.2 Scale-Free ağlar

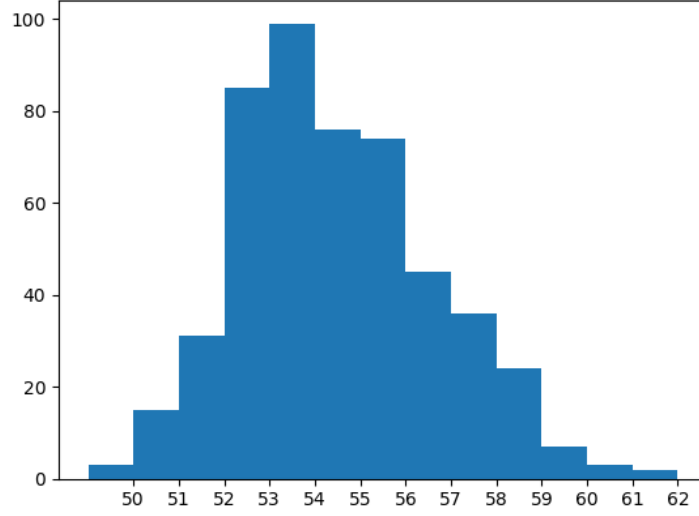
Ağdaki elemanların komşuluk sayılarının Güç Yasasına uygun şekilde dağıldığı ağ tipidir (Barabasi ve Bonabeau 2003). Yani az sayıdaki düğümün derecesi yüksek, çok sayıdaki düğümün derecesi düşüktür. Aradaki değerler ise üstel bir fonksiyonun ara değerleri şeklinde oluşur. Özellikle gerçek dünyadaki sosyal medya ağları ve havayolu ağları bu dağılım tipine yakındır. Örneğin birçok şehirden sadece büyük havalimanlarına uçuş bulunurken, az sayıdaki büyük havalimanından birçok noktaya uçuş gerçekleştirilir. Bu durum hem ülke için havayolu hem de ülkeler arası havayolu için geçerli olduğundan ortaya scale-free bir ağ çıkmış olur. Şekil 2.5'de dereceler dağılımı gösterilen örnek ağ için ortalama komşuluk sayısı 2.41'dir. Ancak düğümlerin büyük çoğunluğunun(111 düğüm) sadece 1 adet komşusu olduğu, 30 adet düğümünse 2 komşusu olduğu görülmektedir. Bununla birlikte bir düğümün 234, bir düğümün 144 komşusu olması bu ağın scale-free bir ağ olduğunu göstermektedir.



Şekil 2.5 Scale-Free ağ örneği dereceler dağılımı

2.4.3 Small-World ağları

Herhangi iki eleman arasındaki en kısa yolun genellikle kısa olduğu bir komşuluk dağılımına sahip ağlardır (Watts ve Strogatz 1998). Gerçek dünyadaki arkadaşlık ilişkileri ve sosyal ağlar bu dağılım tipine yakındır. Dünya üzerindeki herhangi biri, bir başka insana ortak tanıdıklar yardımıyla genellikle 6 adım içinde ulaşabilir. Ayrımın 6 derecesi olarak adlandırılan bu fikir Small-World ağ dağılım tipinin ilham kaynağı olmuştur. Ağ içerisinde belirli sayıdaki alt gruplar kendi aralarında yüksek ihtimalle bağlıdır ve ağ içinde birbirine bağlı küçük ağlar gibi gözükürler. Şekil 2.6'daki dağılım grafiğine sahip çizge 500 düğümden oluşmaktadır ve 50 düğümlük alt çizgelere sahiptir. Bu alt çizgelerin kendi aralarındaki bağ olasılığı %10 olduğunda bu şekilde bir dağılım elde edilebilmektedir. En az derece olarak alt çizge nüfusu olan 50 görülmektedir ve bu düğümler alt çizge dışından herhangi bir bağ kurmamışlardır. 62'ye kadar olan derecelere sahip düğümler ise dış alt çizgelerle 1 ila 12 arasında bağ kurmuştur.



Şekil 2.6 Small-World ağ örneği dereceler dağılımı

2.5 Basamaklı Başarısızlık

Ağlarda çeşitli sebeplerden dolayı işlevsizlik meydana gelebilir. Bu durum başarısızlık (failure) olarak ifade edilir. Ağda belirli bir miktar başarısızlık oluşması durumunda, bu başarısızlık ağın daha geniş bir alanına yayılmaya başlayabilir ve her başarısızlık yeni başarısızlıkları tetikleyebilir. Bu duruma ise basamaklı başarısızlık (cascading failure) denir (Crucitti vd. 2004). Başarısız olarak görevini yerine getiremeyen birimlerin görevleri, diğer birimler tarafından yapılmak zorunda olduğu için bu birimlere fazladan bir yük oluşmakta ve bu yükün belirli bir değeri aşmasıyla bu birim de başarısız olarak etkisiz hale gelmektedir. Bir bilgisayar ağı için bozukluk, ağdaki cihazlardan biri veya birkaçının çalışmaması ve hizmet vermemesi anlamına gelir. Bir hizmet sağlayıcısı olarak çalışan bir ağ üzerinde başarısızlıklar meydana geldiğinde, hizmet veremeyen cihazların yerine diğer cihazlar hizmet vermeye başlar ve bu durum daha fazla kullanıcıya fazladan hizmet vermek zorunda kalan cihazların başarısız olmasına sebep olabilir. Bir enerji altyapısı hattı için, enerji sağlayan santrallerden bazılarının başarısız olması durumunda, gerekli enerji diğer santrallerden sağlanmaya başlayacak ve bu santraller de başarısız olarak işlevlerini kaybedebileceklerdir. Karayolu örneğinde ise bazı kavşakların başarısız olması yani tıkanması, diğer kavşaklara yönelen araçlarla birlikte diğer kavşakların da zaman içinde başarısız olmasına yani tıkanmasına sebebiyet verecektir.

Her elemanın kendi yükünün üzerinde taşıyabileceği bir sınır vardır. Bu çalışmada bu sınır değerine taşma sınırı adı verilmiştir. Karayolu örneğinde 2 katına çıkan kavşakların başarısız olması durumunda taşma sınırı %100 olarak belirtilir. Yükü %100 arttığında başarısız olduğu için taşma sınırı %100'dür veya 1'dir denir. Yükü 1.5 katına çıktığında etkisiz hale gelen elemanların taşma sınırları %50 veya 0.5'tir.

2.5.1 Başarısızlık sebepleri

Ağ üzerinde bir elemanın başarısız olması iki tür sebepten ötürü gerçekleşebilir. Bunlardan ilki, düğümün doğal sebeplerle başarısızlık durumunun ortaya çıkmasıdır, bu çalışmada bu durumu başarısızlık olarak adlandıracağız. Bir enerji hattında, santralin mekanik arıza, sel, heyelan gibi doğal sebeplerle devre dışı kalması; bilgisayar ağındaki bir donanımın voltaj düşüklüğü veya bileşen eskimesi sebebiyle kapanması gibi durumlar başarısızlığa örnek olarak gösterilebilir. Bu durum tamamen rastlantısal olarak meydana gelir ve ağın hangi bölgesinde gerçekleşeceği genellikle bilinemez.

Başarısızlığa sebep olan bir diğer durumsa saldırıdır. Planlı olarak, sistemi işlevsizleştirmek amacıyla yapılır. Genellikle sistemin tutarlılığı bozulana veya sistem tamamen çökene dek saldırılara devam edilerek ağdaki elemanlara teker teker saldırı uygulanır. Saldırıları, bir noktadan sonra basamaklı hatalara sebebiyet vererek sistemin saldırıya devam edilmeden de çökmesine sebep olabilir. Bir enerji hattında sabotaj veya uzaktan erişimle sistemin çökertilmesi sonucu santraller bozulabilir ve enerji hattı işlevsiz hale gelebilir. Karayolu sisteminde ise yoğun kavşaklar bilinçli olarak trafiğe kapatılarak karayolu sistemi çökertilebilir. Bu gibi durumlar ağlarda saldırıya örnek olarak değerlendirilebilir.

Doğal başarısızlık sonucu olarak basamaklı hata oluşumu, saldırı durumuna göre daha zordur ve daha uzun zaman alır (Albert vd. 2000). Çünkü bu tip başarısızlıklar rastlantısal olduğundan, başarısız olan elemanlar ağ için kritik elemanlar olmayabilir ve bu elemanlar ağın farklı bölgelerinde bulunabilir. Saldırı ise genellikle ağ için önemli olan elemanlara gerçekleştirilir ve saldırı öncesinde bir analiz yapılmasını gerektirir. Bu şekilde az sayıda saldırıyla ağa büyük zayıf verilmek istenir.

Saldırıları birkaç şekilde yapılabilir. Ağ hakkında ön bir analiz yapıldıktan sonra saldırıya başlanır. Ön analiz sonucu düğümler, önemlerine göre sıralanır. Hangi

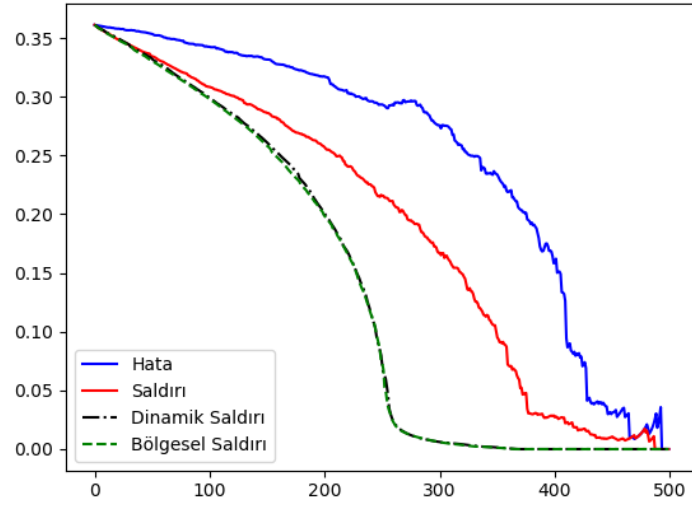
düğümün ağ için daha önemli olduğu bilgisi için verimlilik, arasındalık merkeziliği, yakınlık merkeziliği, düğüm derecesi gibi metrikler kullanılabilir. Bu çalışmada düğüm önemi metriği olarak düğüm verimliliği kullanılmıştır. Ağın etkinliğinin ölçümü ise global verimlilik metriği ile yapılmıştır. Saldırı, en önemli düğümü hedef alarak başlar. İlk yapılan önem sıralamasına göre, kalan düğümler içinden en önemlisine saldırılarak devam eden saldırma işlemlerine statik saldırı denir. Her saldırı sonrası tekrar önem hesaplaması ve sıralaması yapıldıktan sonra en önemli düğüme yapılan saldırılara dinamik saldırı adı verilir. Genellikle bu iki saldırı tipi kullanılarak basamaklı başarısızlık oluşturulmaya çalışır. Dinamik saldırı tipi her adımda yeniden hesaplama gerektirdiği için statik saldırıya göre daha yavaş uygulanır. Ancak dinamik saldırı güncel önem sıralaması kullandığı için daha etkilidir ve ağı daha hızlı şekilde etkisizleştirir.

Bu çalışmada dinamik saldırı tipi biraz daha değiştirilerek bölgesel saldırı adı verilen bir saldırı yöntemi kullanılmıştır. Bölgesel saldırıda, saldırı daha önceden etkisiz hale getirilmiş düğümlerin komşularının içinde en önemli düğüme yapılır. Her saldırı sonrasında komşusu etkisizleştirilmiş düğümler yeniden önem sırasına göre sıralanır. Bu şekilde bütün ağ için hesaplama yapılmadan yalnızca daha önce komşusu etkisizleştirilmiş düğümler için hesaplama yapılır. Dinamik saldırıya göre daha hızlı uygulanabilir. Ağın etkinliğini azaltma konusunda dinamik saldırıyla benzer bir performans gösterir.

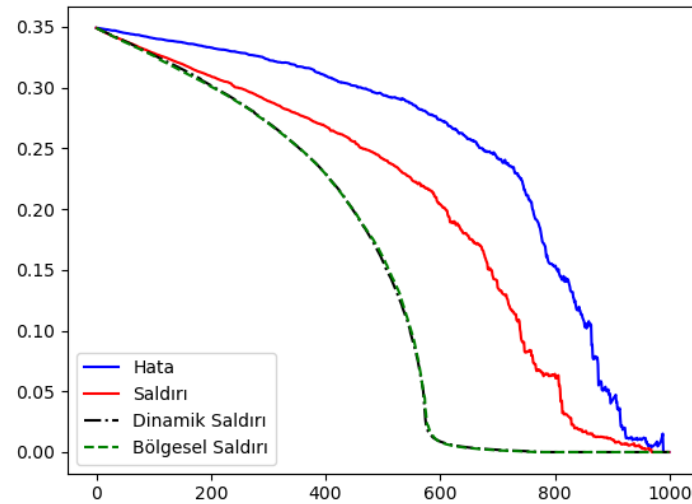
2.5.2 Basamaklı çökme simülasyonu

Basamaklı çökme durumlarının karşılaştırılabilmesi için farklı düğüm sayısına sahip ağlar oluşturulmuş ve bu ağlar 4 farklı stratejiyle çökertilerek oluşan durumlar kaydedilmiştir. Çökertme stratejisi olarak doğal hata durumları, statik saldırı, dinamik saldırı ve bölgesel saldırı yöntemleri kullanılmıştır. Şekil 2.7'de yoğunluğu 0.02 olan 500 elemanlı bir ağa uygulanan saldırıların, ağa etkisi görülebilir. Benzer şekilde şekil 2.8'de yoğunluğu 0.01 olan 1000 elemanlı bir ağa uygulanan saldırıların, ağa etkisi gözlenebilir. Farklı eleman sayısındaki ağlarda saldırıların etkilerinin benzer olduğu şekil üzerinden anlaşılabilir. İlk verimi benzer olan ağlara hata simülasyonu uygulandığında, ağın veriminin çok yavaş şekilde azaldığı görülmektedir. Burada hata, doğal başarısızlık anlamında kullanılmıştır. Tamamen rastgele olarak başarısızlık

gerçekleştğinde ağ verimi buna benzer şekilde azalır. Statik saldırı durumunda hata durumuna göre çok daha hızlı şekilde ağ veriminin azaldığı, dinamik saldırı durumundaysa statik saldırıdan daha hızlı ağ veriminin azaldığı görülebilir. Bölgesel saldırının ağ verimliğine etkisi dinamik saldırıyla çok benzerdir.



Şekil 2.7 500 elemanlı bir ağda saldırı etkisi



Şekil 2.8 1000 elemanlı bir ağda saldırı etkisi

3. YAPAY ZEKA

Binlerce yıllık bir insan ideali olan Yapay Zeka, insanın yetenekleri, zekası ve tecrübesiyle yapabildiği bir işin, makineler tarafından yapılabilmesi anlamına gelir (Russell ve Norvig 2016). Özellikle bilgisayar biliminin gelişmesiyle, uzun yıllardır insanlık için bir ideal olan bu kavramı gerçekleştirme çalışmaları da hız kazandı. Bilgisayar alanında gerçekleşen her yenilikle birlikte, yapay zeka çalışmaları da evrildi ve insanın düşünerek yapabildiği işleri başarılı şekilde yapabilen sistemler oluşmaya başladı.

Günümüzde sağlıktan savunmaya, eğlenceden ekonomiye kadar insan faktörünün bulunduğu bütün alanlarda insan gibi karar veren yazılımlar ve bu kararları uygulayan yazılım ve donanımlar bulunmaktadır. Bir bilgisayar oyununda karşımızda insan gibi hamleler yapan rakip oyuncu, sağlık testinden elde edilen veriye göre hastanın olası hastalıklarını çıkararak doktora sunan yardımcı yazılım sistemi, havayolu trafiğini yolcuları en kısa süre bekletecek ve ekonomik olarak en verimli olacak şekilde düzenleyen kontrol yazılımları yapay zeka uygulamalarına birer örnek olarak verilebilir. Bunların yanı sıra insan gibi düşünerek insan gibi hareket edebilen yapay zeka destekli sistemler de geliştirilmektedir. Bir insandan daha başarılı şekilde masa tenisi oynayan robotlar, futbol oynayan robotlar, cerrahi müdahale yapan robotlar, trafikte seyir eden sürücüsüz otomobiller birer bilim kurgu filmi olmaktan çıkıp hayatımıza girmeye başladı. Yapay zeka uygulamaları artık insan kararlarına benzer şekilde karar vermekten öteye gidip insanın yaratıcılık özelliklerini de taklit etmeye başladı. Şarkı besteleme, film senaryosu yazma, resim yapma, espri yapma gibi becerilere sahip yazılımlar ortaya çıkmaya başladı.

Yapay zekaya ulaşabilmek için zaman içinde farklı teknikler kullanıldı. Önceleri kural temelli veya karar ağaçlarından oluşan sistemler geliştirildi. Bu sistemler eğer bu gerçekleştiyse şöyle yap, gerçekleşmediyse böyle yap şeklinde işleyen kurallar bütünü olarak düşünülebilir. Bu yapay zeka sistemlerini geliştirmek uzun zamanlar alabiliyordu ve başarımlar olarak karmaşık problemlerin çözümünde yetersiz kalınıyordu. Bu sorunlardan dolayı istatistiksel yöntemler üzerinde çalışmalar yoğunlaştı ve makine öğrenmesi kavramı doğdu.

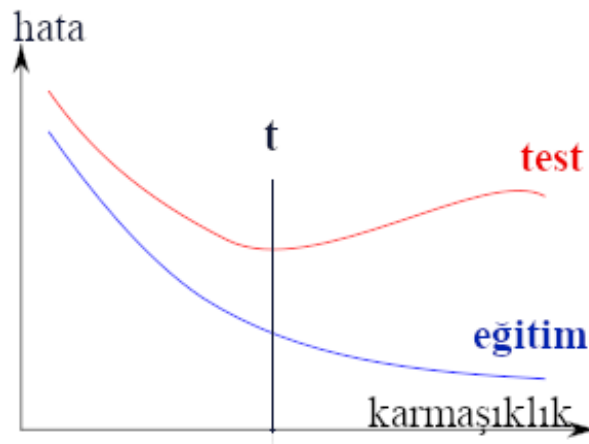
3.1 Makine Öğrenmesi

Makine öğrenmesi, yapay zekanın alt alanı olan istatistiksel bir yöntemdir. Kural temelli değil veri temellidir (Russell ve Norvig 2016). Tanımlanmış sabit kurallar yoktur, eğitim için kullanılan veri vardır. Eğitim verisi olarak adlandırılan bu veri, öğrenilmek istenen ilişkiyi ortaya koyan örnekler içerir. Fotoğraf üzerinde erikleri tespit etmek istediğimizi düşünelim. Bu tespiti kural temelli olarak yapabilmek için, rengi yeşil veya sarıysa, şekli yuvarlağa yakınsa, yaprağı yoksa veya yeşil yaprakları varsa şeklinde oluşan çok sayıda kural tanımlamamız gerekecektir. Ayrıca genel kurallara uymayan birçok erik de mevcuttur ve bunların bu kural sisteminde ayırt edilmesi mümkün gözükmemektedir. Bu ayırt etme işini makine öğrenmesi yöntemiyle yapabilmek için, içlerinde erik olan ve olmayan çok sayıda fotoğraf toplamak yeterli olacaktır. Kullanılan makine öğrenmesi yöntemine bu eğitim fotoğraflarını verdiğimizde, eğitim sonucunda sistem artık bir fotoğrafta erik olup olmadığını anlıyor hale gelecektir. Zaman içerisinde çeşitli makine öğrenmesi yöntemleri geliştirilmiştir. Bütün makine öğrenmesi yöntemleri, insanın yeni bir şey öğrenirken kullandığı temel stratejiyi kullanır. Basketbol topunu potaya sokmaya çalışırken topu attığımızda top potaya girmezse bu atış olumsuz örnek olarak beynimizde işlenir. Yaptığımız atışların isabetli olması durumu da benzer şekilde tecrübe edilir ve beynimizde işlenir. Bu şekilde beyin, yapılan atışla isabet arasında eldeki veriyi kullanarak bir bağlantı kurar. Makine öğrenmesi yöntemleri de eldeki olumlu ve olumsuz örneklerden çıkarım yaparak karar mekanizmasını gerçekleştirir.

3.2 Makine Öğrenmesi Süreci

Makine öğrenmesi yöntemlerinde öğrenilen bilgi, belirli sayıdaki değişkenden sayısal olarak tutulur. Bu değişkenler, eğitim veri seti kullanılarak ayarlanır. Bu sayısal değerlerin ayarlanması sürecine eğitim adı verilir. Bu değişkenlerin sayısı ve değerleri kullanılan makine öğrenmesi yöntemine göre değişiklik gösterir. Eğitim tamamlandıktan sonra yeni bir bilgi üretmek için, eğitimle oluşturulmuş değişkenler kullanılır. Eğitim süreci genellikle uzun zaman alır ancak eğitimden sonra yeni bir bilgi oluşturma işlemi hızlıca gerçekleştirilebilir.

Seçilen makine öğrenmesi yöntemi uygulanmadan önce, elde sadece eğitim için kullanılacak veri seti bulunur. Bu veri seti, eğitim seti ve test seti olarak ikiye ayrılır. Eğitim seti, modelin iç değişkenlerini belirlemek için kullanılırken, test seti modelin doğru şekilde istenen bilgiyi öğrenip öğrenemediğini tespit etmek için kullanılır. Eğitim sürecinde, modelin iç değişkenleri eğitim setindeki her girdiyle birlikte güncellenir. Değişkenler uygun şekilde oluşturulamazsa başarısız bir eğitim yapılmış olur. Model eğitim seti için bile doğru sonuçlar üretemez hale gelebilir. Öğrenilmek istenen ilişkiyi çözemez ve istenen oranda doğru sonuç üretemez. Bu duruma yetersiz uyma (underfitting) denir (Russell ve Norvig 2016). Uygun olmayan bir veri seti kullanılması veya eğitimin yetersiz miktarda gerçekleşmesinden kaynaklanabilir. Değişkenlerin güncellenmesi süreci gereğinden fazla uzatıldığında ise model, eğitim setine aşırı derecede bağımlı hale gelebilir. Bu durumda eğitim setindeki veriyi ezberleyerek, ancak girdinin tamamen aynı olması durumunda eğitim setindeki sonucu oluşturur. Bu duruma aşırı uyma (overfitting) adı verilir (Russell ve Norvig 2016). Test seti bu durumların tespiti için kullanılır. Eğitim setindeki başarı oranları düşükse yetersiz uyma tespit edilebilir. Ancak aşırı uyma durumu, eğitim seti içinde çok iyi sonuçlar verecektir çünkü gördüğü veriye aşırı bir uyum yani ezberleme söz konusudur. Test setindeki veri, model tarafından daha önce görülmemiş örnekler içerdiği için aşırı uyma durumunda test seti için model yanlış çıkarımlar yapacaktır. Eğitim başarımının artmasına rağmen test başarımının artmaması aşırı uymaya işaret eder.



Şekil 3.1 Eğitimde karmaşıklık hata ilişkisi

Şekil 3.1'de modelin karmaşıklığı arttıkça eğitim hatasının düştüğü görülmektedir. Ancak t karmaşıklığına kadar azalan test hatasının bu noktadan sonra artmaya başladığı

yani daha önceden görmediği veri üzerinde daha kötü çalışmaya başladığı gözlemlenebilir. Bu adımdan sonra aşırı uyuma başladığı yani modelin ezberlemeye başladığı söylenebilir. Karmaşıklığın t değerinin altında olduğu bölüm için ise model yetersiz uyuma sahiptir denilebilir. Bu problem için bu yöntem ve bu veri setiyle üretilebilecek en başarılı çözüm eğitimin t karmaşıklığında sonlandırıldığı çözümdür diyebiliriz.

Eldeki veri probleme göre %60-75 oranında eğitim ve %25-40 oranında test için kullanılır. Ancak bazı durumlar için eldeki veri, doğrulama seti ile birlikte üçe de ayrılabilir. Bazı yöntemlerde dışarıdan modele müdahale edilebilirken bazı yöntemlerde model, bütün dinamikleri kendi içinde belirler.

Bazı durumlarda elde yeteri kadar nitelikli veri bulunmayabilir. Bu durumda bazı yöntemlerle veri seti çoğaltılabilir. Eldeki erik fotoğrafları veri seti içinde yeterli miktarda fotoğraf yoksa bu fotoğrafların açıları değiştirilerek yeni erik fotoğrafları üretilebilir. Çünkü hangi açısı değişmiş erik fotoğrafı gerçek dünyada yine eriği ifade etmektedir. Görüntü yatay veya dikey olarak ters çevrilerek, kırılarak veya görüntünün renk tonu değiştirilerek yeni erik görüntüleri oluşturulabilir.

3.3 Başarım Metrikleri

Bir modelin başarısı, model ile üretilen sonuçların hata miktarının az olması ile belirlenir. Bazı hata ölçüm metrikleri şu şekildedir (Botchkarev 2018).

Doğruluk: Sınıflandırma problemlerinde kullanılan basit bir hata hesaplama yöntemidir. Doğru sonucu veren örneklerin bütün örneklere oranıdır. Her sınıf için, o sınıfa ait veri girişi model çıktısının o sınıf çıkması durumu doğru bir tahmin olarak değerlendirilir. Yapılan test sonucu doğru sonuçların, bütün örneklerin sayısına oranı doğruluktur. 0 ile 1 arasında bir değer oluşturur. 1 kesin doğruluk 0 ise tamamen yanlışlık anlamına gelir. Doğruluk, hatanın tersidir yani doğruluğun artması hatanın azalması anlamına gelir.

R Kare: Kararlılık katsayısı olarak da bilinir. Model sonucu ortaya çıkan sonuçların, olması gereken sonuçlarla ne denli ilişkili olduğunu ölçmek için kullanılan bir hata hesaplama yöntemidir. Test sonucu ile olması gereken sonuç arasındaki korelasyonu

ifade eder. 1 veya 1'den küçük bir değer üretir. 1 mutlak korelasyon, 0 ise hiç bir ilişki bulunamaması anlamına gelirken 0'dan küçük değerler rastgele doğruluğun bile altında bir sonuç oluşturulduğunu gösterir. Bu hata miktarının 1'e yakın olmasını hedeflenir.

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}$$

$$\bar{y} = \frac{1}{n} \sum_i y_i$$

Bu denklemde y_1, y_2 şeklinde n adet değerden oluşan y gerçek sonuçlar kümesi ve f_1, f_2 şeklinde n adet tahmin sonucundan oluşan f tahmin kümesi için R^2 değeri hesaplanmaktadır. \bar{y} değeri gerçek sonuçların ortalamasıdır.

RMSE: Kök kare ortalama hatası (Root Mean Square Error), model sonucu ortaya çıkan sonuç değerlerinin, olması gereken değerlere göre ne kadar yanlış olduğunu ifade eder. 0 veya 0'dan büyük bir değer üretir. 0 hiç yanlışlık olmaması anlamında gelirken bu değer ne kadar büyüksse o kadar isabetli sonuçlardan uzaklaşıldığı anlamına gelir. Bu hata miktarının 0'a yakın olması hedeflenir.

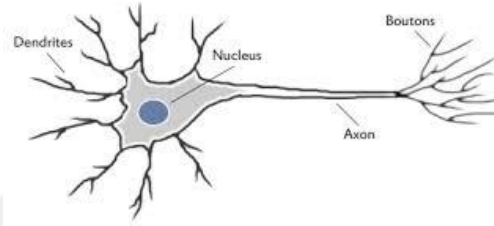
$$RMSE = \sqrt{\frac{\sum_i (y_i - f_i)^2}{n}}$$

Bu denklemde y_1, y_2 şeklinde n adet değerden oluşan y gerçek sonuçlar kümesi ve f_1, f_2 şeklinde n adet tahmin sonucundan oluşan f tahmin kümesi için RMSE değeri hesaplanmaktadır.

Karışıklık Matrisi: Karışıklık matrisi(Confusion Matrix), sınıflandırma problemlerinde hangi sınıfa ait veriye karşı hangi sınıf sonucunun oluştuğunun gözlemlenebilmesi için oluşturulan tablo şeklinde bir yapıdır. Kullanılan sınıf kadar satır ve sütuna sahiptir. Karışıklık matrisinin asal köşegeni doğru tahminleri içerir. Bu yapı sayesinde hangi sınıflar arasında karışıklık ve geçişkenlik olduğu gözlemlenebilir.

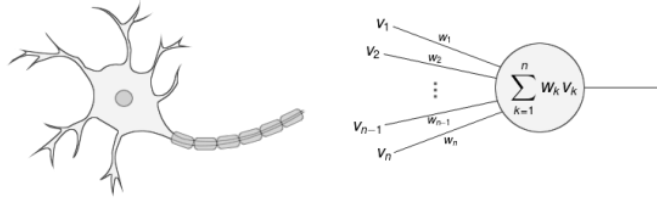
3.4 Yapay Sinir Ağları

Yapay sinir ağları, en sık kullanılan makine öğrenmesi yöntemlerinden biridir. Temel olarak insan beynine benzeyen bir model sunar (Russell ve Norvig 2016). Beyin, şekil 3.2'deki gibi nöron adı verilen birbirleriyle bağlantılı hücrelerden oluşur. Bu hücrelerin şekil 3.2'de görüldüğü gibi birbirlerine bağlandığı ve sinyal ilettiği giriş ve çıkış kısımları vardır. Yapay sinir ağları modeli de beynin çalışma yapısına benzer şekilde, giriş ve çıkışları olan ve bu giriş çıkışlarıyla diğer nöronlarla bağlantı kuran nöron adı verilen bileşenlerden oluşur.



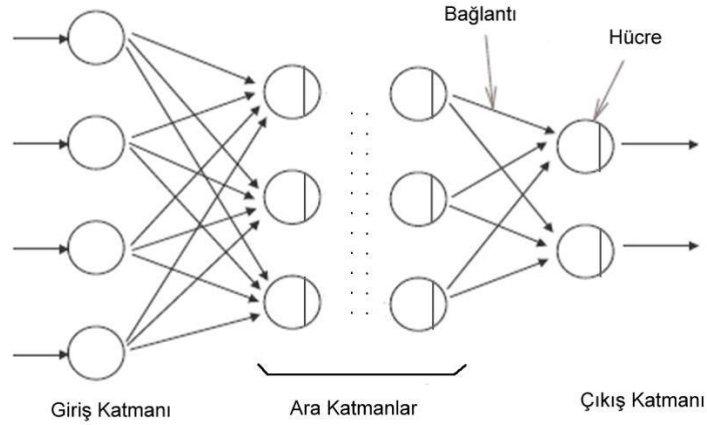
Şekil 3.2 İnsan sinir hücresi bileşenleri

Sinir hücrelerinde dendrit denilen giriş ve akson denilen çıkış kısımları bulunur. Hücre bunlar vasıtasıyla diğer sinir hücreleriyle bağlantı kurar.



Şekil 3.3 İnsan sinir hücresi - yapay sinir ağı sinir hücresi

Yapay sinir ağı, şekil 3.3'deki gibi sinir hücrelerine benzeyen ve nöron denilen bileşenlerin birbirlerine bağlanması ile oluşturulur. Her sinir ağında bir giriş katmanı, bir çıkış katmanı ve bu iki katman arasında kalan gizli katmanlar bulunur. En az bir adet gizli katman olması gerekir. Girdi verisi giriş katmanından geçerek ulaştığı her nöronun ağırlığıyla çarpılır. Her nöron kendisine ulaşan sinyal ve ağırlık çarpımlarını toplar. Bu toplam değeri bir aktivasyon fonksiyonunun çıktısı olarak nöronun çıkışına iletir. Çıktı katmanı hariç bütün nöronların çıktıları başka nöronların girdileridir. Bu şekilde ağ ileriye doğru beslenerek çıktı katmanına ulaşılır.



Şekil 3.4 Yapay sinir ağı genel modeli

Şekil 3.4'de yapay sinir ağlarının genel yapısı görülmektedir. Yapay sinir ağları, sınıflandırma, değer tahmini, kümeleme, özellik çıkarımı gibi çeşitli problemlere uygulanabilir. Probleme ve kullanılan veriye göre çeşitli yapay sinir ağı modelleri mevcuttur. Günümüzde revaçta olan yöntemlerden derin sinir ağları da bir yapay sinir ağıdır. Derin sinir ağları çok sayıda gizli katmandan oluşur.

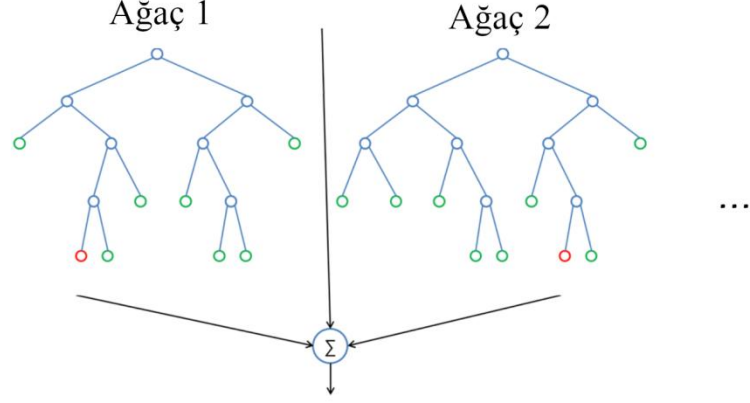
3.5 Makine Öğrenmesi Yöntemleri

Bu çalışmada 3 farklı makine öğrenmesi yöntemi kullanılmıştır. Bu yöntemler Random Forest, Extreme Learning Machine ve Long Short Term Memory'dir.

3.5.1 Random Forest

Sınıflandırma ve değer tahmini için kullanılan bir makine öğrenmesi yöntemidir (Breiman 2001). Türkçe karşılığı Rastgele Orman olsa da genel kabul görmüş bir karşılığı olmadığı için İngilizce şekliyle kullanılmıştır. Gerçekleştirilmesi son derece basit olduğu için birçok problemde kullanılır. İsminden anlaşılacağı üzere rastgele olarak oluşturulan karar ağaçlarından oluşur. Şekil 3.5'de Random Forest modelinin genel yapısı görülmektedir. Dışarıdan parametre olarak oluşturulmak istenen karar ağacı sayısını alır. Bu karar ağaçları, veri setindeki örneklerle rastgele olarak oluşturulur. Girdi olarak kullanılan özellikler, farklı ağaçlarda farklı eşik değerler üretir. Tahmin kısmında, üretilen bütün ağaçlar kullanılarak tek bir sonuç ortaya çıkartılır. Problem bir sınıflandırma problemiye, ağaçlar tarafından en çok çıkartılan sonuç döndürülürken, problemin bir değer üretme (regression) problemi olması durumunda ise ağaçlardan

ıkan sonuların ortalaması alınarak sonu retilir. Modelin karmaşıklığı ađa sayısıyla belirlenir.



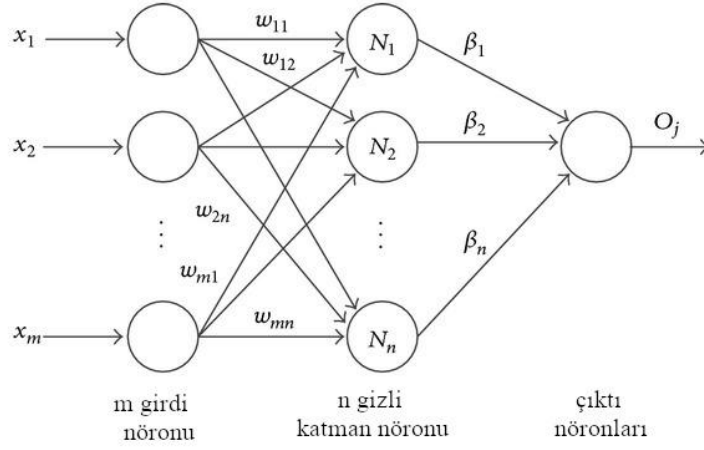
Őekil 3.5 Random Forest modeli

$$x = \frac{1}{n} \sum_t T(x)$$

Formlde, bulunmak istenen deđer x'in, n adet ađatan oluŐan bir Random Forest modeline $T(x)$ Őeklinde girdi olarak verilmesi ve bu ađaın ıkardığı sonuların ortalaması alınarak deđerin elde edilmesi ifade edilmektedir.

3.5.2 ELM

Makine đrenmesi yntemlerinden biri olan Extreme Learning Machine (kısaca ELM) geleneksel makine đrenmesi yntemlerine gre ok daha hızlı bir đrenme performansı sunar. Temel olarak deđiŐkenlerin ađırlıklarını matris arpımı yardımıyla hesaplar. ELM tek katmanlı ve yalnızca ileri beslemeli bir modeldir. Bir giriŐ katmanı, bir ıktı katmanı ve yalnızca bir adet gizli katmandan oluŐur. Tekrarlayan ađırlık gncellemesi iŐlemlerine ihtiya duymaz. Yalnızca tek tekrarlamayla btn nron ađırlıkları belirlenir. Modelin karmaşıklığı gizli katmandaki nron sayısıyla belirlenir (Huang vd. 2006). Tek bir matris iŐlemi yapılması ve bu Őekilde tekrarlayan ađırlık gncellemelerine ihtiya duymaması sayesinde diđer makine đrenmesi yntemlerinden ok daha hızlı bir đrenme performansı gerekleŐtirir.



Şekil 3.6 ELM modelinin genel yapısı

ELM modelinin genel yapısı Şekil 3.6'da gösterilmektedir. m adet girdi değeri ile n adet gizli katman nöronu beslenerek bir çıktı değeri elde edilir. Eğitim için kullanılacak veri seti eğitim ve test seti olarak ayrıldıktan sonra eğitim seti(N) girdi(x_i) ve çıktı(t_i) değerleri olarak ayrılır. Gizli katmandaki nöron sayısı(\tilde{N}) belirlendikten sonra her bir nöron için rastgele ağırlıklar atanır. Eğitim setindeki bütün girdi verileri bu aşda $g(x)$ aktivasyon fonksiyonu kullanılarak ileri besleme ile işleme sokulur ve bir çıktı matrisi(H) hesaplanır.

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(\mathbf{x}_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{o}_j, \\ j = 1, \dots, N,$$

Bu aşamada bilinmeyen tek şey β değerleridir. $\mathbf{H}\beta = \mathbf{T}$ olduğundan $\beta = \mathbf{H}^{-1}\mathbf{T}$ matris işlemi ile de β değerleri elde edilebilir.

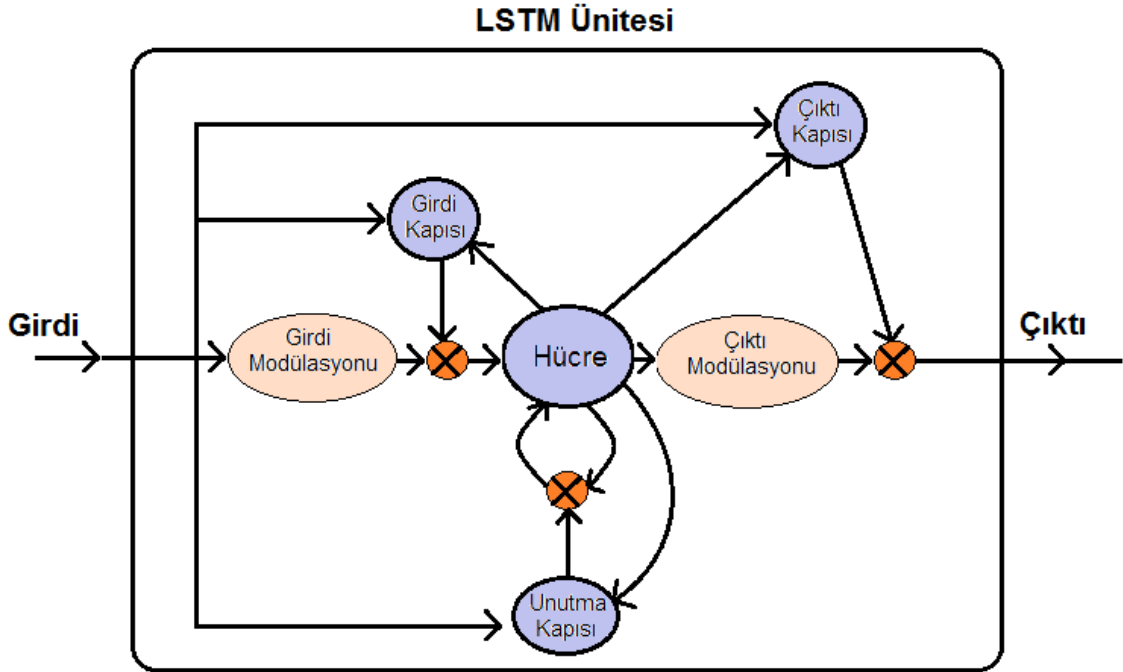
$$\mathbf{H} = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_1 + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_N + b_{\tilde{N}}) \end{bmatrix}_{\tilde{N} \times N} \quad \beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N \times m}$$

H matrisinin tersinin alınması için bir işlem daha yapılmalıdır çünkü H matrisi kare bir matris olmayabilir. Bu noktada Moore-Penrose Genelleştirilmiş Tersleme işlemi kullanılarak matris tersi alınabilir (Huang vd. 2006).

3.5.3 LSTM

Son yıllarda popüler olan makine öğrenmesi yöntemlerinden biri olan LSTM (Long Short Term Memory), tekrarlamalı bir yapay sinir ağı modelidir (Gers vd. 2000). Klasik Yapay sinir ağı modelleri, her örneği kendi özelinde değerlendirerek tahmin yapar. Bazı durumlarda örneklerin sıralaması önemli olabilir. Bu gibi durumlarda örnekler zaman bilgileriyle birlikte daha çok anlam ifade etmektedir. Metin tamamlama, görüntü sahne çıkarımı (Byeon vd. 2015), konuşma tanıma (Graves vd. 2013), gerçek zamanlı çeviri gibi konularda yakın zamanlarda LSTM tabanlı çalışmalar yapılmıştır.

LSTM modelinde her örneğin kendisi girdi olarak eğitime katıldığı gibi eğitim çıktıları da tekrar geri besleme olarak eğitime girdi olmaya devam eder. Bu şekilde verinin yalnızca kendisine bakmaz, veriyi sekans olarak işleme tabi tutar. LSTM modeli ünitelerden oluşur. Bu üniteler şekil 3.7'deki gibi bir hücre ve bu hücreye bağlı girdi kapısı, çıktı kapısı ve unutmaya kapısından oluşur.



Şekil 3.7 LSTM ünitesi

Her bir LSTM ünitesinde merkezde hücre yer alır ve 3 kapı ile etkileşimdedir. Unutmaya kapısı, belirli şartlarda hangi bilgilerin üniteden atılacağına karar verir. Girdi kapısı,

belirli şartlarda girdideki hangi bilginin hafızayı güncelleyeceğine karar verir. Çıktı kapısı ise belirli şartlarda girdi ile hafızadaki bilgiden hangi çıktı verisinin çıkartılacağına karar verir. Üniteye girdi olarak sekans halinde veri girer ve bu veriye göre kapılar aktif olup olmayacaklarına karar verir. Her kapının kendi ağırlığı vardır ve bu ağırlıklar eğitim süresince güncellenir. Bu 3 kapı sayesinde LSTM modeli klasik yapay sinir ağlarından daha akıllı bir hale gelebilir. Bu kapılarla model, yalnızca o andaki veriyi kullanmak yerine, veri sekansındaki eski bilgileri de kullanabilir hale gelir. Üniteye, hücre bir hafıza birimi gibi çalışır. Hangi bilginin tutulacağı hangi bilginin ne derece gereksiz olduğu bu yapı sayesinde sürekli olarak güncellenerek belirlenir. Girdi değeri ve çıktı değeri modülasyona tabi tutulur.

LSTM modeli, veride zamana bağlı bir ilişki olduğunda diğer makine öğrenmesi yöntemlerine göre daha iyi çalışmaktadır. Modelin karmaşıklığı ünite sayısı ile belirlenir. Ünite sayısının artırılmasıyla daha karmaşık ilişkilerin tespit edilmesi sağlanabilir.

Verinin modele girdi olarak sunulması yani beslenmesi kısmında bazı metrikler dikkate değerdir. Eğitim veri setinin tamamının modele girdi olarak 1 kez sunulması 1 epoch anlamına gelir. Bu tanıma göre 5 epoch'luk bir eğitim, eğitim veri setinin tamamının 5 kez model tarafından eğitim sürecine sokulduğu anlamına gelir. LSTM gibi yapay sinir ağlarında veri setinin bir kez beslenmesi genellikle yeterli olmaz ve model defalarca bu veri ile beslenerek iç ağırlıklar güncellenir ve daha iyi bir eğitim sağlanır. Eğitim setinin tamamı, tek seferde besleme için çok büyük kalabilir. Bu yüzden veri genellikle batch'lere bölünerek model beslemesi yapılır. Batch boyutu kavramı tek seferde modele girdi olarak sunulan örnek sayısını ifade eder.

4. MATERYAL ve YÖNTEM

4.1 Veri Setinin Oluşturulması

Karmaşık ağlar konusunda gerçek dünyadan, ağların yapısıyla ilgili bilgiler içeren veri bulmak oldukça zordur. Bulunan az sayıda veri ile bir eğitim gerçekleştirmek mümkün olmadığı için eğitim ve test aşamalarında kullanılan örnek veri, sanal olarak üretilmiştir. Üretilen bu sanal ağlar gerçek dünyadaki ağlarla benzer özellikler taşımaktadır. Normal dağılıma sahip, farklı yoğunluklarda 2913 adet ağ üretilmiş ve bu ağlara yine sanal olarak saldırılar gerçekleştirilerek ortaya çıkan veri kaydedilmiştir. 400, 500 ve 1000 düğümlü ağlar oluşturularak bu ağlar üzerinde çalışılmıştır. İşlem zamanını azaltmak adına 500 düğümlü ağlar üzerinde yoğunlaşmıştır ancak diğer ağ gruplarının da benzer sonuçlar oluşturduğu gözlemlenmiştir.

Çizgedeki her düğüm, kendi düğüm verimliliğinin taşma sınırı kadar altına inebilirken bu değer altına inilmesi durumunda düğüm başarısız oldu kabul edilir. Örneğin ilk düğüm verimliliği 0.2 olan ve taşma sınır 1 yani %100 olan bir deney grubunda, bu düğümün düğüm verimi 0.1'in altına indiğinde, düğüm başarısız oldu kabul edilmektedir. Düğüm verimi, düğümün çizge için ne kadar önemli olduğunu ifade ettiği gibi, düğümün o anki durumu hakkında da bilgi vermektedir. Çizgeden çıkartılan bir düğümün yükünün komşularına dağıtılması durumu, komşularının düğüm verimlerinin azalması olarak ifade edilebilir. Çünkü bir düğümü çizgeden çıkarttığımızda

$\frac{1}{N-1} \sum_j \frac{1}{d_{ij}}$ formülü uyarınca $\frac{1}{N-1}$ ifadesi düğüm verimliliklerinde küçük bir değer

artırımına sebep olsa da, bu düğümün komşularının en kısa yollarında kuvvetle muhtemel bir artışa sebebiyet verecek ve $\frac{1}{d_{ij}}$ ifadesi ile büyük bir değer kaybına uğratacaktır. Çizgeden her düğüm çıkarılması, bu düğümün yakın çevresinden başlayarak azalacak şekilde bir düğüm verimi düşüşü oluşturur. Zhou ve Wang (2018) tarafından yapılan çalışmada hata ve saldırı durumlarında bağlantı ile verimlilik arasındaki ilişki incelenmiştir ve ağda saldırıların veya hataların etkisinin bu metrikle açıklanabileceği dile getirilmiştir.

Deney ağlarına bölgesel saldırılar uygulanarak basamaklı başarısızlıklar oluşturulmuştur. Düğüm önemi için düğüm verimliliği, ağın etkinliği içinse global

verimlilik metrikleri kullanılarak saldırılar gerçekleştirilmiştir. Çizge içindeki en yüksek düğüm verimliliğine sahip düğümle başlanarak gittikçe büyüyen bir alt çizgede saldırılar dinamik hesaplamalarla gerçekleştirilmiştir. Örneğin deney ağlarından birkaçı için basamaklı başarısızlık durumları çizelge 4.1'deki gibi oluşmuştur:

Çizelge 4.1 Örnek basamaklı başarısızlık durumları

	Deney Grubu	Sırasıyla Her Adımdaki Başarısızlık Sayıları	Kalan Düğüm
Deney 1	#1	8 - 10 - 6 - 4 - 3 - 5 - 10 - 18 - 180 - 230	0
Deney 2	#6	5 - 10 - 19 - 2	416
Deney3	#6	3 - 7 - 21 - 24 - 33 - 31 - 20 - 14 - 2 - 1 - 1	296

Örnek saldırılarda görüldüğü gibi basamaklı başarısızlığın etkisi, basamak sayısı, saldırılardan kurtulabilen düğüm sayısı rastgele olarak ağın yapısına göre değişmektedir. Örneğin 1 numaralı deney için, saldırılar gerçekleştirildikten sonra kalan düğümlerde 10 adımlı bir basamaklı başarısızlık oluştuğu görülmektedir. Saldırı sonrası ilk adımda 8 düğümün başarısız olduğu, daha sonra tekrar yük dağılımı yapıldıktan sonra ikinci adımda 10 düğümün, üçüncü adımda 6 düğümün başarısız olduğu görülebilir. Bu şekilde devam eden basamaklı başarısızlıklar sonucu ağda hiçbir düğüm kalmamıştır. Benzer şekilde 2 numaralı deneyin 4 adım sürdüğü ve 416 düğümün basamaklı başarısızlıktan kurtulduğu, 3 numaralı deneyin 11 adım sürdüğü ve 296 düğümün bu basamaklı başarısızlıktan kurtulduğu görülmektedir.

Deney ağlarının oluşturulması ve saldırıların gerçekleştirilerek kayıt altına alınması için Python programlama dili ile birlikte NetworkX, Numpy ve Pandas kütüphaneleri kullanılmıştır. Kullanım kolaylığı ve hızlı geliştirmeye imkan vermesi dolayısıyla Python dili, iyi dokümente edilmiş olması ve kullanım kolaylığı nedeniyle NetworkX kütüphanesi tercih edilmiştir. Deney ağları NetworkX kütüphanesi yardımıyla oluşturulmuş, global verim ve düğüm verimi metrikleri paket kullanmaksızın kodla hesaplanarak saldırılar gerçekleştirilmiştir. Numpy ve Pandas kütüphaneleri yardımıyla ağlar üzerinde çeşitli parametreler hesaplanarak dosya yapısında kayıt altına alınmıştır.

4.1.1 Deney ağları dağılımı

Üretilen ağlar, düğüm bağlantıları açısından rastgele olmakla birlikte, belirli kriterlere göre gruplanabilir.

Çizelge 4.2 Deney ağ grupları - 1

Grup	D.S.	K.S.	Yoğ.	Ort. Km.	Taş. Sn.	Ü.A.S.
#1	500	1240	0.01	4.96	0.5	609
#2	500	1655	0.0133	6.62	0.5	497
#3	500	1905	0.0153	7.62	0.5	432
#4	500	2767	0.0222	11.07	0.5	245
#5	500	2255	0.0181	9.02	0.5	292
#6	500	2437	0.0195	9.75	0.5	319
TOPLAM	500	1890	0.0151	7.543	0.5	2394

Çizelge 4.3 Deney ağ grupları - 2

Grup	D.S.	K.S.	Yoğ.	Ort. Km.	Taş. Sn.	Ü.A.S.
#6	500	2437	0.0195	9.75	0.5	319
#7	500	2437	0.0195	9.75	0.33	286
#8	500	2437	0.0195	9.75	0.75	233
TOPLAM	500	2437	0.0195	9.75	0.526	838

D.S. : Ağdaki düğüm sayısı

K.S. : Ağdaki ortalama toplam kenar sayısı

Yoğ.: Ağın ortalama yoğunluğu

Ort. Km. : Düğümler için ortalama komşu sayısı (derecesi)

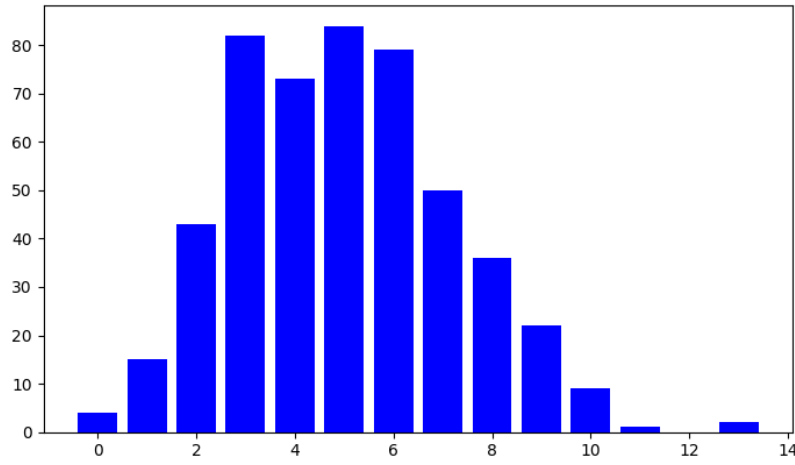
Ü.A.S. : Üretilen ağ sayısı

Taş. Sn. Ağdaki düğümlerin taşma sınırı

Belirli kısıtlarla rastgele olarak oluşturulan ağlar çizelge 4.2 ve çizelge 4.3'teki gibidir. Örneğin 1 numaralı deney grubu için 609 adet ağ üretilmiştir. Bu gruptaki bütün ağlar 500 düğümden oluşmaktadır ve ortalama olarak 1240 kenar içermektedir. Bu deney seti için ağlar, en az 1180 en fazla ise 1300 kenar içerecek şekilde oluşturulmuştur ve ağların ortalama yoğunlukları 0.01 olmuştur. Ağlardaki düğümler, normal dağılım özelliği göstermektedir. Ortalama komşuluk sayıları 4.96 olarak ortaya çıkmıştır. Örnek olarak bu setteki bir ağ için komşuluk sayıları yani dereceler dağılımı çizelge 4.4'teki gibi oluşmuştur. Dereceler dağılımı grafik olarak şekil 4.1'deki gibidir.

Çizelge 4.4 433 numaralı testte kullanılan ağın dereceler dağılımı

Test #433			Toplam kenar sayısı: 1248									
0	1	2	3	4	5	6	7	8	9	10	11	13
4	15	43	82	73	84	79	50	36	22	9	1	2



Şekil 4.1 433 numaralı testte kullanılan ağın dereceler dağılımı grafiği

Bu ağda hiç komşusu olmayan 4 düğüm, 13 komşusu olan 2 düğüm vardır ve bu ağ için ortalama komşuluk sayısı 4.992 olarak oluşmuştur. Bunun gibi oluşturulan diğer bütün ağlar normal dağılım örneği göstermektedir.

Benzer şekilde diğer deney grupları da normal dağılıma sahiptir ve farklı yoğunluklardadır. Bu gruplardan 1, 2, 3, 4, 5 ve 6 deney grupları başarısızlık ve çökme

tahmini için kullanılmıştır. Her grup farklı ortalama yoğunluğa sahiptir. Bu gruplarda en düşük ortalama yoğunluk 0.01 olurken en yüksek ortalama yoğunluk 0.0222 olmuştur. Bütün ilk aşama deney grupları içinse ortalama yoğunluk 0.0151 olmuştur. Bu 6 grup için de taşma sınırı 0.5 olarak belirlenmiştir. Yani herhangi bir düğüm, mevcut kapasitesinin %50 sinin üzerine çıktığında işlevini kaybederek başarısız olacaktır.

Taşma sınırının, bu tahmin yönteminde ne kadar etkin olduğunu test edebilmek için 7 ve 8 numaralı deney grupları oluşturulmuştur. 6 numaralı deney seti ile aynı ortalama yoğunluğa sahip olan bu gruplar için taşma sınırı değeri sırasıyla 0.33 ve 0.75'tir. Yani 0.33 taşma sınırı olan 7. grup için ağdaki düğümlerin her biri ilk durum kapasitesinin %33 üzerine çıkması durumunda başarısız olacaktır. Taşma sınırı 0.75 olan 8. deney grubu ağlarında ise, ilk kapasitesinin %75 üzerine çıkan düğümler benzer şekilde başarısız olacaktır.

4.1.2 Deney ağlarına saldırılar

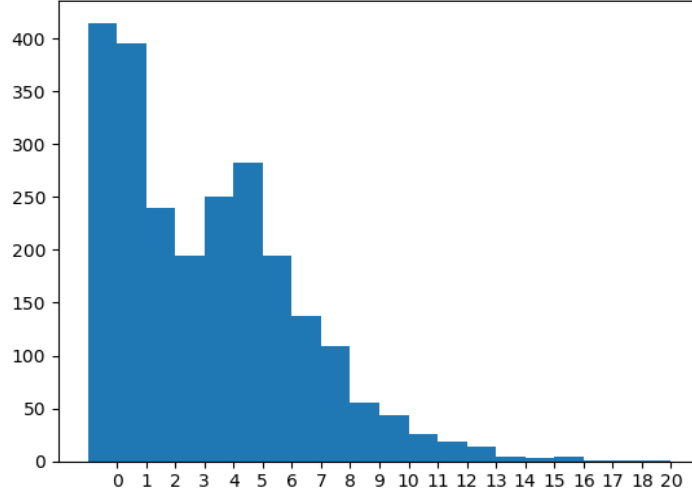
İlk aşama deney grupları için, üretilen ağlara kontrollü olarak saldırılar düzenlenmiştir. Saldırı yöntemi olarak bölgesel saldırı seçilmiş ve ağlar üretildikten sonra, ağın yoğunluğuna bağlı olarak belirli sayıda saldırı düzenlenmiştir. Yapılan denemelerde bu şekilde normal dağılıma sahip ağlarda, ilk yoğunluk değerini yaklaşık olarak %25 oranda azaltana dek yapılan bölgesel saldırılar, ağda basamaklı başarısızlığa neden olabilmektedir. Gerçek dünya problemlerinde olduğu gibi saldırılar büyük oranda basamaklı hata oluşturmaktadır. Bazı durumlarda ise saldırılar bittiğinde herhangi bir basamaklı başarısızlık oluşmamıştır. Bu durumun da tespit edilebilmesi açısından bu yoğunluk azaltma yöntemi önem arz etmektedir. Örneğin 6. deney grubunda ortalama ağ yoğunluğu 0.0195'tir. Bu gruptaki bir ağ için yoğunluk yaklaşık olarak 0.0146 değerine düşene dek saldırı yapılır. Saldırı miktarı her zaman %25'lik bir yoğunluk kaybına neden olmayabilir çünkü bu değer de rastlantısal olarak belirlenir ancak saldırılar ortalama olarak yaklaşık %25'lik bir kayba neden olmaktadır. Yoğunluğu daha az olan ağlarda, düğümler arasında daha az sayıda yol olduğundan dolayı saldırıların etkisi daha büyük olacaktır. Düşük yoğunluklu ağlarda, saldırılarla başarısız hale getirilen düğümden geçen yolların alternatifleri, yüksek yoğunluklu ağlardaki düğümlere göre daha azdır. Bu durum da düşük yoğunluklu ağların saldırılar sonucu daha fazla global verimliliğini kaybetmesine neden olur. Bu yüzden deneyler sırasında

sabit bir saldırı sayısı belirlenmemiş, ağın yoğunluğuyla orantılı olarak saldırı yapılmıştır.

Saldırıları gerçekleştirildikten sonra, saldırı sonucu başarısız olan düğümler komşularından başlayarak diğer düğümlerin düğüm verimliliklerini azaltır. Bu verim azalması gerçek dünya problemlerinde yükün artması anlamına gelmektedir. Bir düğüme ait düğüm verimliliği değeri azaldıkça o düğümün yükü artmış olur. Saldırıları sonrası yeniden verim hesaplamaları yapılır ve ağdaki sağlıklı düğümlerde herhangi bir başarısızlık olup olmadığına bakılır. Eğer başarısızlık oluştuysa, başarısız olan düğümler de ağdan çıkartılarak bütün ağ için yeniden düğüm verimlilikleri hesaplaması yapılır. Saldırı sonrasında yapılan her hesaplama aşaması bir adım olarak kaydedilir. Bu şekilde ağda düğüm kalmayana ya da ağda başarısızlık oluşturan yeni bir düğüm kalmayana dek basamaklı başarısızlık oluşturmaya devam edilir. Her deney adımı için ağda kalan düğümlerin bazı değerleri kaydedilir. Deney sonucunda düğümlerin başarısız olup olmadığı ve eğer başarısız olunduysa hangi adımda başarısız olduğu bilgisi çıkartılarak kaydedilir. Her deney için bütün düğümlere ait bilgiler adım bilgisiyle oluşturulur.

Bazı deneylerde, saldırı sonucu hiçbir basamaklı başarısızlık adımı oluşmazken bazı deneylerde 20 adıma kadar basamaklı başarısızlık oluştuğu gözlenmiştir. İlk aşamadaki bütün deney ağları için 2394 deney sonucu, basamaklı başarısızlık adımları şekil 4.2'de grafik olarak gösterildiği gibi şu şekilde oluşmuştur:

0 adım: 415 adet, 1 adım: 396 adet, 2 adım: 240 adet, 3 adım: 195 adet, 4 adım: 251 adet, 5 adım: 283 adet, 6 adım: 195 adet, 7 adım: 138 adet, 8 adım: 109 adet, 9 adım: 56 adet, 10 adım: 44 adet, 11 adım: 26 adet, 12 adım: 18 adet, 13 adım 14 adet, 14 adım: 4 adet, 15 adım: 3 adet, 16 adım: 4 adet, 17 adım: 1 adet, 18 adım: 1 adet ve 20 adım: 1 adet.



Şekil 4.2 Basamaklı başarısızlık adım sayıları

Örneğin 2 numaralı deney grubu için 497 deney sonucunda, basamaklı başarısızlık adımları şu şekildedir:

0 adım: 60 adet, 1 adım: 83 adet, 2 adım: 49 adet, 3 adım: 38 adet, 4 adım: 32 adet, 5 adım: 57 adet, 6 adım: 31 adet, 7 adım: 29 adet, 8 adım: 29 adet, 9 adım: 6 adet, 10 adım: 8 adet, 11 adım: 3 adet, 12 adım: 5 adet, 13 adım: 1 adet, ve 15 adım: 1 adet

Düğümlere ait tekil veri miktarını artırmak için her basamaklı başarısızlık adımı kendi içinde bağımsız olarak da değerlendirilebilir. Yani 1. adım sonrası, 2. adım sonrası durumları yeni bir ağ olarak da değerlendirilebilir. Bu şekilde düğümler için tekil veri miktarını artırmış oluruz. Örneğin bir v düğümü 10. adım sonrasında başarısız olmuş olsun. Bu düğüm, saldırısı sonrasında ait veriyle 10 adım sonra başarısız olacak şekilde etiketlenebilir. Bu v düğümü için 1. adım verisiyle 9 adım sonra başarısız olacak şekilde de etiketlendirme yapabiliriz.

Saldırı sonucu 9 adımı geçen çok adımlı basamaklı başarısızlık oluşması durumu çok sık görülmemektedir. Bununla birlikte düğüm bazlı veriyi artırmak için gerçekleştirilen ara adımlara ait verilerin eklenmesiyle bu değerler oransal olarak daha küçük bir alanı ifade etmektedir. Bunun için 9 adımı geçen bütün değerler 10+ olarak etiketlenmiştir ve bu şekilde başarısızlığa uzaklık verisi toplam 11 sınıftan oluşmuştur.

Gerçekleştirilen başarısızlık testleri sonucu, düğüm bazlı olarak çizelge 4.5'teki gibi başarısızlığa uzaklık etiketleri ve çizelge 4.6'daki gibi başarısızlık durumu değerleri elde edilmiştir.

Çizelge 4.5 Başarısızlığa uzaklık etiketleri

0 Adım	1 Adım	2 Adım	3 Adım	4 Adım	5 Adım
814265	462180	450292	427844	366983	277470
6 Adım	7 Adım	8 Adım	9 Adım	10+ Adım	Toplam
192666	130060	82126	53202	91776	3348864

Çizelge 4.6 Başarısızlık durumları

Başarısız Olacak	Başarısız Olmayacak
2534599	814265

4.2 Gerçekleştirilen Deneyler

Veri seti kullanılarak bir takım testler gerçekleştirilmiştir. Bütün deneyler için veri seti her seferinde rastgele olarak %70-%30 oranında ayrılmıştır. %70'lik kısımlar gerekli eğitimlerin gerçekleştirilmesi için kullanılmış, %30'luk kısımlar ise eğitim sonuçlarının test edilmesi için ayrılmıştır.

Çalışmada, programlama dili olarak Python kullanılmıştır. Yardımcı olarak TensorFlow, Scikit-Learn, Matplotlib Numpy ve Pandas kütüphanelerinden faydalanılmıştır. Hızlı geliştirme yapabilme olanağı ve örnek kodların mevcudiyeti bu dil ve ilgili kütüphanelerin seçiminde en önemli etkenler olmuştur. Tensorflow kütüphanesi; LSTM uygulamaları için kullanılmışken Scikit-Learn kütüphanesi; veri ön işleme, normalizasyon, veri seti ayırma, Random Forest uygulamaları ve ELM uygulamaları (extentions paketinde) için kullanılmıştır. Numpy ve Pandas kütüphaneleri verinin matris olarak okunması, kaydedilmesi ve işlenmesi için kullanılmıştır. Başarım metrikleri paket kullanmaksızın kodla oluşturulmuş ve grafik olarak çizimler Matplotlib kütüphanesiyle üretilmiştir.

Çalışmada, LSTM uygulamaları ekran kartı üzerinde çalıştırılmıştır. Windows 10 işletim sistemi ve 6 GB hafızalı NVIDIA GTX 1060 GPU kullanılarak LSTM çalışmaları gerçekleştirilmiştir. Diğer bütün uygulamalar Windows 7 işletim sistemi üzerinde, Intel i5-4200M işlemci ve 8 GB RAM içeren bir bilgisayar üzerinde gerçekleştirilmiştir.

4.2.1 Çökme tahmini

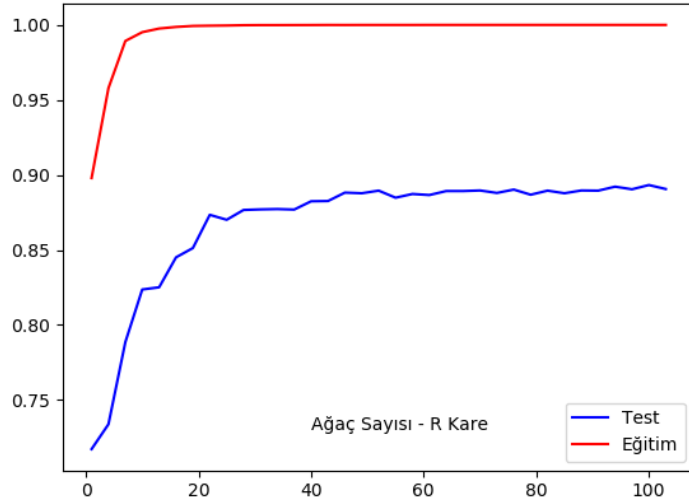
Düğümlere ait bireysel veri kullanılarak, düğümlerin saldırılar sonucu meydana gelecek basamaklı çökme sonucunda başarısız olup olmayacağını tahmin edilmesi için farklı makine öğrenmesi yöntemleriyle denemeler yapılmıştır. Bu çalışmada, 1 numaralı deney ağ gruplarına ait düğüm verileri kullanılmıştır. Söz konusu veri seti 3348865 örnekten oluşmaktadır. Veri setindeki düğümlere ait bireysel metriklerden 9 tanesi girdi olarak kullanılmış ve başarısız olup olunmayacağı ikili bilgisi çıktı olarak beklenmiştir. Şu bilgiler girdi olarak üretilerek kullanılmıştır:

- 1) Saldırı öncesi ilk durumdaki düğüm veriminin ağıın genel verimine oranı (0'dan büyük bir değer)
- 2) Şu anki düğüm veriminin saldırılar öncesi ilk durumdaki düğüm verimine oranı (0-1 arası bir değer)
- 3) Başarısız olan düğüm sayısı
- 4) Düğümün derecesi
- 5) Komşularının komşularından oluşan kümenin eleman sayısı (en kısa yolu 2 olan düğümlerin sayısı)
- 6) Komşularının düğüm verimlerinin en düşüğü (0-1 arası bir değer)
- 7) Komşularının düğüm verimlerinin ortalaması (0-1 arası bir değer)
- 8) Ağıın saldırılar öncesi ilk durumdaki yoğunluğu (0-1 arası bir değer)
- 9) Ağıın şu anki yoğunluğu (0-1 arası bir değer)

Bu girdi değerlerinden 2 ve 4 numaralı veri, düğümün kendi içinde hesaplayabileceği şekildedir. 1 ve 8 numaralı değerler en başta hesaplanır ve bütün süreç boyunca sabit kalır. Düğüm yalnızca komşularıyla haberleşerek 5, 6, ve 7 numaralı değerleri hesaplayabilir. Ağ üzerinde genel haberleşme gerektiren yalnızca 3 ve 9 numaralı veri bulunmaktadır.

4.2.1.1 Random Forest uygulaması

Random Forest makine öğrenmesi yöntemiyle farklı ağaç sayıları kullanılarak çökme tahmini yapılmıştır. Ağaç sayısının az olması durumu yetersiz uyma sorununu, fazla olması ise aşırı uyma sorununu beraberinde getirmektedir. Ağaç sayısı ile çıktılarının R Kare sonucu arasındaki ilişki Şekil 4.3'teki grafikte belirtildiği gibidir.



Şekil 4.3 Random Forest ağaç sayısı - R Kare doğruluk ilişkisi

Veri setinin %10'luk bir kısmı rastgele seçilerek, ağaç sayısı ile başarı oranı arasındaki ilişkiyi belirlemek için bu şekilde bir test yapılmıştır. Ağaç sayısı artırıldıkça eğitim R Kare başarımının da arttığı ve 60 ağaçtan sonra artık eğitim setine tamamen uyulduğu yani R Kare başarımının 1 olduğu görülmektedir. Test R Kare başarımı ise 52 ağaca kadar yükselmiş daha sonrasında ise azalıp artmasına rağmen bu değer üzerine çıkamamıştır. Bu ikili sınıflandırma işlemi için 52 ağaçlık bir Random Forest modeli kullanmanın başarımı artırmak adına mantıklı olduğu görülmüştür.

52 ağaçlık bir Random Forest modeli ile eğitim veri seti kullanılarak eğitim yapılmış ve test veri seti ile eğitim sonucu değerlendirilmiştir. Eğitim sonucu test veri seti için çizelge 4.7'deki gibi bir karışıklık matrisi ortaya çıkmıştır.

Çizelge 4.7 Random Forest ikili sınıflandırma karışıklık matrisi

	Tahmin Başarısız Değil	Tahmin Başarısız
Gerçek Başarısız Değil	241889	2048
Gerçek Başarısız	1754	758969

R Kare başarımları (test veri seti): 0.9794

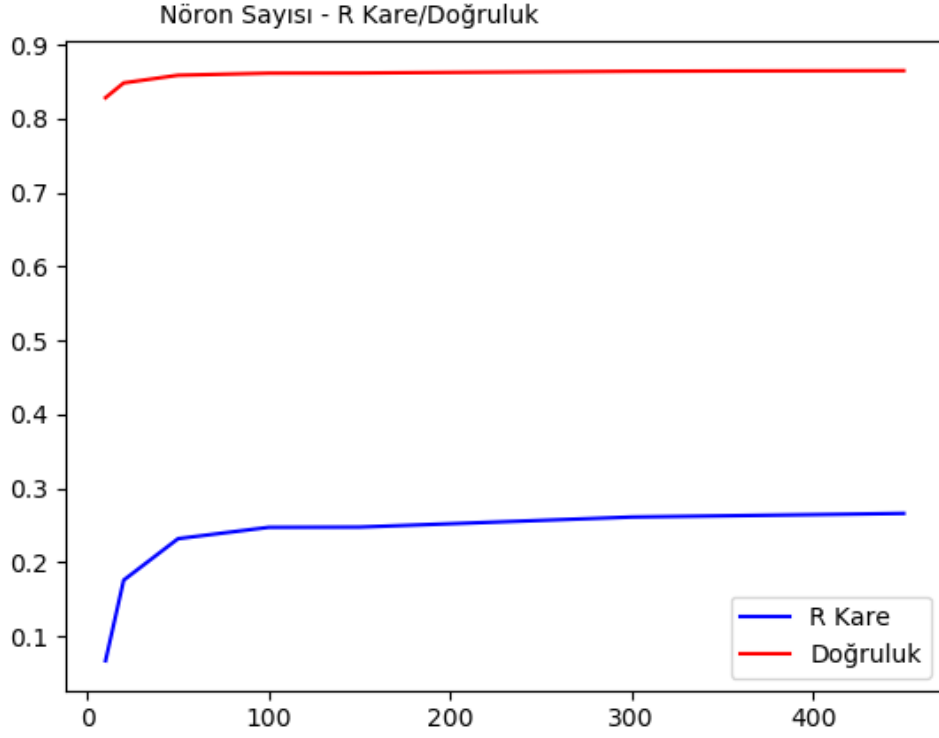
R Kare başarımları (eğitim veri seti): 0.9999

Doğruluk oranı: 0.9962

Düğüm bazlı olarak, düğümün çökmeye maruz kalıp başarısız olacağı veya çökmeye maruz kalmayıp başarısız olmayacağı bilgisi %99.62'lik bir doğrulukla tahmin edilebilmektedir.

4.2.1.2 ELM uygulaması

ELM, tek seferde işlem yaparak sonuca ulaşan bir yöntem olduğu için çok miktarda hafıza kullanımına ihtiyaç duyar. ELM modelinin karmaşıklığı gizli katmandaki nöron sayısı ile belirlenir. Bu sayı hafıza kısıtları nedeniyle yeteri kadar artırılamamıştır. Çalışılan bilgisayarda nöron sayısı 450'ye kadar çıkartılabildiği için eğitim sonucu modelin öğrendiği, ancak karmaşık yeteri kadar artırılamadığı için eksikleri bulunduğu gözlenmiştir. Yapılan testler sonucunda, ELM modelinin gizli katman nöron sayısı ile test doğruluğu arasındaki ilişki şekil 4.4'de verilmiştir.



Şekil 4.4 ELM modeli karmaşıklık - doğruluk ilişkisi

Modeldeki gizli katman nöron sayısı 10'a çıktığında test seti için R Kare değerinin pozitif değerlere yükseldiği görülmektedir. R Kare değerinin pozitif olması, modelin bir korelasyon bulduğunu göstermektedir yani ELM modeli 10 ve daha fazla sayıda nöron kullanarak, girdi değerleriyle düğümün başarısız olup olmayacağı arasında bir ilişki tespit edebilmektedir. 10 nöron ile 0.0668 R Kare ve 0.8284 doğruluk oranı elde edilirken, 50 nöron ile 0.23196 R Kare ve 0.8588 doğruluk oranı, 150 nöron ile 0.2476 R Kare ve 0.8618 doğruluk oranı elde edilmektedir. Karmaşıklık arttıkça R Kare değerinin ve doğruluğun arttığı görülmektedir. Yapılan denemelerde hafıza kısıtları nedeniyle en fazla 450 nöron kullanılabilmiştir ve bu eğitimin sonucunda oluşan karışıklık matrisi çizelge 4.8'deki gibidir.

Çizelge 4.8 ELM ikili sınıflandırma karışıklık matrisi

	Tahmin Başarısız Değil	Tahmin Başarısız
Gerçek Başarısız Değil	149625	94312
Gerçek Başarısız	41246	719477

R Kare başarımları (test veri seti): 0.2661

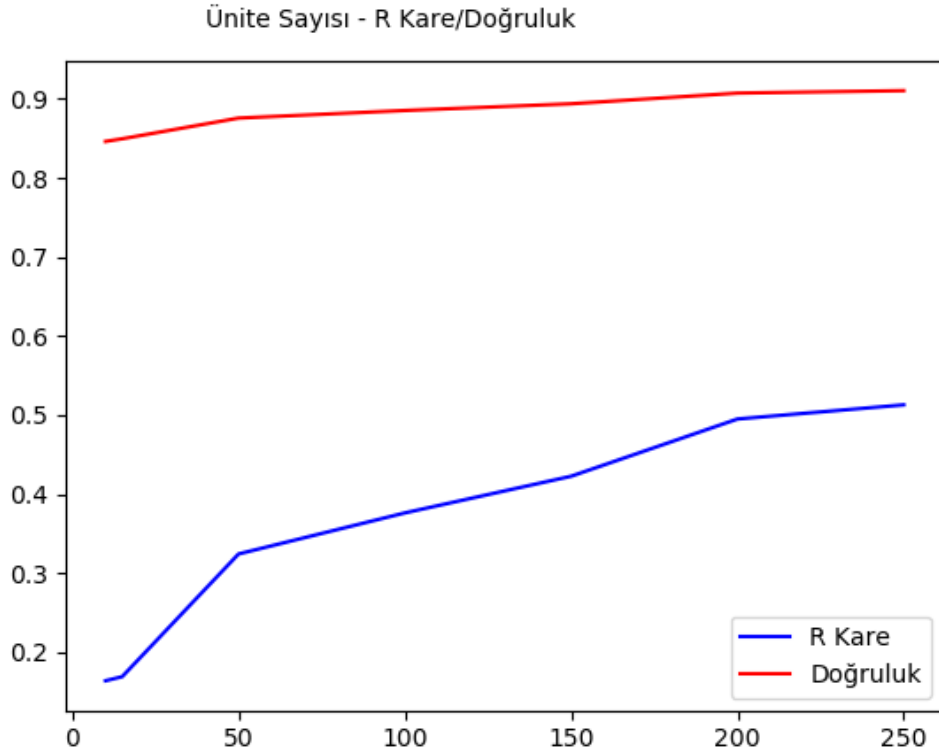
R Kare başarımları (eğitim veri seti): 0.2702

Doğruluk oranı: 0.86505

Nöron sayısı ile doğruluk oranı arasındaki ilişkinin daha büyük nöron sayılarında da korunduğunu denemek için veri setinin %10'luk bir kısmı olan 334886 adet rastgele seçilmiş örnek kullanılarak, 2500 nöronluk bir ELM modeli ile eğitim gerçekleştirilmiştir. Verinin %10'unun kullanılmasının başarımları azaltması beklenir ancak nöron sayısının artırılmasıyla daha iyi bir eğitim gerçekleşmiş ve eğitim sonucunda test veri seti için 0.2963 R Kare değeri ve 0.8705 başarımları oranı elde edilmiştir.

4.2.1.3 LSTM uygulaması

LSTM, zamana bağlı ilişkilerin çıkarımında başarılı olan ve son yıllarda oldukça popüler olmuş bir derin sinir ağı modelidir. Modelin karmaşıklığı, ünite sayısı ile alakalıdır. Veri seti defalarca modele girdi olarak sunulabilir ve bu şekilde eğitim süresi uzatılarak başarımları artırılabilir. Modeldeki ünite sayısı ile başarımları arasındaki ilişkinin tespit edilmesi için farklı ünite sayıları ile model eğitilmiş ve sonuçlar kayıt altına alınmıştır. Bu sonuçlar şekil 4.5'te görülmektedir. 15 ünite ile yapılan eğitimde 0.8495 doğruluk ve 0.169 R Kare değeri elde edilirken, 100 ünite ile yapılan eğitimde 0.88531 doğruluk oranı ve 0.3762 R Kare değeri elde edilmiştir.



Şekil 4.5 LSTM modeli karmaşıklık - doğruluk ilişkisi

LSTM modeli için RAM miktarı sorunu, test seti batch boyutu değiştirilerek aşılabılır. Çalışmanın gerçekleştirildiği bilgisayarda en fazla 400 ünite ile eğitim yapılmasının temel sebebi eğitim zamanını makul seviyeye çekmektir.

400 ünite ile 512'lik batch boyutuyla ve 50 epoch kadar yapılan eğitimin sonucu olarak çizelge 4.9'daki gibi bir karışıklık matrisi ve şu başarımlar oluşmuştur.

Çizelge 4.9 LSTM ikili sınıflandırma karışıklık matrisi

	Tahmin Başarısız Değil	Tahmin Başarısız
Gerçek Başarısız Değil	213115	30822
Gerçek Başarısız	32670	728053

R Kare başarıml (test veri seti): 0.6563

R Kare başarıml (eđitim veri seti): 0.8202

Dođruluk oranı: 0.9368

LSTM uygulamasının sonucu olarak, modelin problemi öğrenebildiđi, ancak zaman kısıtı nedeniyle belirli bir başarımlın üzerine çıkılamadıđı görölmüştür.

4.2.2 Çökme tahmini yöntemlerinin karşılaştırılması

Bu çalışmada kullanılan Random Forest, ELM ve LSTM yöntemlerinin üçünün de düđüm başarısızlık tahmininde kullanılabileceđi görölmüştür. Bütün yöntemler, girdi deđerleriyle başarısızlık arasında pozitif bir korelasyon tespit edebilmiştir. Çalışma kapsamındaki başarısızlık tahmini için üç makine öğrenmesi yöntemi arasında Random Forest yönteminin en kullanışlı yöntem olduđu görölmüştür.

ELM yönteminde eğitim son derece hızlı gerçekleşmektedir ancak yüksek miktarda RAM (rastgele erişimli hafıza) ihtiyacı bulunduđu için model karmaşıklığı belirli bir noktaya kadar artırılabilir. Veri miktarı çok olduđunda veya eğitim yüksek karmaşıklık istediđinde ELM modeli yetersiz kalabilmektedir. Çökme tahmini için 3 milyonun üzerinde örnekle çalışıldıđı için ELM modelinin gizli katman nöron sayısı en fazla 450'ye çıkarılabilmektedir. Veri miktarı azaltılarak başarıml bir miktar artırılabilir ancak yine hafıza ihtiyacı nedeniyle karmaşıklık istenen seviyeye getirilememiştir.

LSTM yönteminde eğitim, ünite sayısı artırılarak karmaşılaştırılır. Ancak bu sayının artması aynı zamanda eğitim zamanını da artırır. Modelin karmaşıklığı arttıkça kullanılması gereken hafıza miktarı da artar. Ancak test seti batch boyutu ayarlanarak bu sorun aşılabilir. Verinin tamamı ile yapılan eğitimin belirli bir başarımla ulaşması için modele veri setinin defalarca beslenmesi gerekmektedir. Zaman kısıtları nedeniyle veri setini tekrar besleme işlemi (epoch sayısı) belli bir miktara kadar çıkarılabilmektedir. Veri miktarının azaltılması ile eğitim zamanı düşürölmüş ve daha fazla epoch sayısı boyunca eğitim gerçekleştirilebilmiştir. LSTM modeli, zaman tabanlı veri üzerinde daha iyi çalışan bir model olmasına rağmen, çalışma zamanını artırması nedeniyle ünite sayısı ve epoch sayısı sınırlandırılmak zorunda kalındığı için istenen başarıml oranına ulaşamamıştır.

Random Forest yöntemi ise ELM modelinden çok daha az hafıza ihtiyacı ve LSTM modelinden çok daha kısa eğitim süresiyle diğer iki yöntemden ayrılmaktadır. Gerçekleştirilen testler sonucunda en iyi başarı oranına sahip yöntem olarak karşımıza Random Forest çıkmaktadır. Modelin basitliği, işlem kolaylığı sağlamak ve kısıtlı kaynaklarla (işlemci ve hafıza) yüksek başarı oluşturmaktadır.

Sonuç olarak bu problem için Random Forest yöntemi, diğer iki yöntem olan ELM ve LSTM yöntemlerinden daha kullanılabilir ve daha isabetlidir.

ELM ve LSTM yöntemlerinin daha iyi çalışabildiği bir konfigürasyonda daha isabetli karşılaştırma yapabilmek için daha küçük bir veri seti üzerinde çalışılmıştır. Yöntemlerin doğruluk oranlarının karşılaştırılması için çökme tahmininde kullanılan veri seti rastgele seçim yöntemiyle %10 oranına küçültülmüştür. Bu şekilde hafıza ve eğitim zamanı olarak kısıtlı kalan yöntemlerin daha iyi bir başarı elde etmesi sağlanmak istenmiştir. Sonuç olarak yöntemler, 234420 adet örnek içeren aynı eğitim veri seti ile eğitilerek, 100466 adet örnek içeren aynı test veri seti ile test edilmiştir. Test sonuçları çizelge 4.9'daki gibi oluşturmuştur. Karışıklık matrislerinde sol üst alan 0 beklenip 0 elde edilen, sağ üst alan 0 beklenip 1 elde edilen, sol alt alan 1 beklenip 0 elde edilen ve sağ alt alan ise 1 beklenip 1 elde edilen örneklerin sayısını gösterir.

Çizelge 4.10 Çökme tahmini yöntemleri doğruluk karşılaştırması

Yöntem	R Kare	Doğruluk	Karışıklık Matrisi	
Random Forest	0.8907	0.9799	23193	1230
			789	75254
ELM	0.2962	0.8705	16125	8298
			4712	71331
LSTM	0.4561	0.9	19655	4768
			5287	70756

Yöntemlerin zaman olarak karşılaştırılması için bütün yöntemler aynı CPU (Merkezi işlemci birimi) üzerinde çalıştırılarak, çalışma zamanları kayıt altına alınmıştır. Buna göre ELM yöntemi 10 saniyede, Random Forest yöntemi 125 saniyede işlemini tamamlarken LSTM yönteminin çalışması 205 dakika almıştır.

Sonuç olarak ELM ve LSTM yöntemlerinin büyük veri miktarına göre daha iyi performans gösterdikleri ancak yine de Random Forest yönteminin başarımına ulaşamadıkları görülmüştür.

4.2.3 Çökme uzaklığı tahmini

Düğümlere ait anlık veriyle, düğümlerin kaç adım sonra çökeceği bilgisinin tahmin edilmesi için Random Forest, ELM ve LSTM yöntemlerinden yararlanılmıştır. Bu çalışmada, 1 numaralı deney ağ gruplarına ait düğüm verileri kullanılmıştır. Modellere girdi olarak çökme tahmininde kullanılan 9 girdi verilmiş ve çıktı olarak başarısızlığa uzaklık etiketi (0 ile 10 arasında tamsayı bir değer) beklenmiştir. Bu etiketin 0 olması, düğümün başarısız olmayacağı anlamına gelirken, 1 ile 9 arasındaki değerler düğümün kaç adım sonra başarısız olacağı anlamına gelmektedir. Bu etiketin 10 olması ise düğümün 10 veya daha fazla adım sonra başarısız olacağını ifade etmektedir.

Kullanılan yöntemlerin karşılaştırması çökme tahmini çalışmasındakiyle son derece benzer olduğu için bu karşılaştırmaya ayrıca değinilmemiştir.

Çökme uzaklığı tahmininde, klasik sınıflandırma problemlerinin aksine, çıktı olarak elde edilen değerlerin birbirleriyle yakınlıkları da bir anlam ifade etmektedir. Örneğin tahmin edilmek istenen örneğin çıktısı 4 olması gerekirken 10 veya 0 olduysa bu durum istenen sonucun uzağında bulunduğu anlamına gelmektedir. Bu örnek için çıktı değeri olarak 3 veya 5 elde edilmesi ise tahminin bir miktar başarılı olduğunu gösterir. Bu nedenle çökme uzaklığı tahmini yapılırken başarımlar için R Kare doğruluğunu baz almak daha isabetli olur. Uygulanan yöntemler için R Kare ve doğruluk değerlerinin yanı sıra yakın doğruluk adında üçüncü bir başarımlar metriği kullanılmıştır. Yakın doğruluk, çıktı olarak elde edilen değerlerin gereken değerle aynı veya komşu olması olasılığıdır. Gerçek değeri N olan bir örnek için çıktı olarak $N-1$, N veya $N+1$ elde edilmesi doğru olarak kabul edilerek doğruluk değerinin hesaplanmasıyla yakın doğruluk değeri elde edilir.

Çökme uzaklığı tahmini, çökme tahmininden daha karmaşık bir modele ihtiyaç duymaktadır. İkili sınıflandırma yerine 11 sınıflı bir sınıflandırma yapılması gerekmektedir.

4.2.3.1 Random Forest uygulaması

Çökme tahmininde kullanılan model, 150 ağaca çıkartılarak karmaşıklık artırılmıştır. Sonuç olarak çizelge 4.11'deki gibi bir karışıklık matrisi oluşmuştur. Modelin başarımları sonucunda çizelge 4.12'de verilmiştir.

Çizelge 4.11 Random Forest çökme uzaklığı sınıflandırma karışıklık matrisi

	0	1	2	3	4	5	6	7	8	9	10
0	242232	338	397	204	114	162	85	123	89	23	170
1	270	129849	8597	199	30	9	1	2	1	0	0
2	830	7122	118044	8645	248	33	7	4	4	4	0
3	1284	493	8638	109842	7808	230	68	20	14	31	0
4	1373	152	949	7071	93993	6092	229	80	54	31	25
5	1644	45	367	545	5469	70472	4432	134	45	22	89
6	1424	17	160	254	321	3487	48712	3009	112	32	79
7	1187	5	70	190	144	125	2339	33036	1930	81	20
8	773	4	34	90	149	83	41	1504	20689	1305	55
9	461	2	11	77	67	54	27	68	922	13489	933
10	737	0	1	7	34	89	105	7	2	538	25991

Çizelge 4.12 Random Forest yöntemi çökme uzaklığı sınıflandırma başarımları

Yöntem	R Kare (test)	Doğruluk	Yakın Doğruluk
Random Forest	0.9238	0.9021	0.9822

4.2.3.2 ELM uygulaması

Çökme tahmininde kullanılan modelin daha karmaşık bir hale getirilmesi gerektiği için gizli katman nöron sayısı artırılmıştır. Ancak bütün sınıfları kapsayacak bir eğitim gerçekleştirilememiştir. Bu nedenle yalnızca 4 sınıftaki çökme uzaklığı verisi kullanılarak bir eğitim gerçekleştirilmiştir. Daha fazla sınıftaki verinin çıkarımı için karmaşıklığın artırılması gerekmektedir. Çalışma yapılan bilgisayarda en fazla 400 nöronla bir eğitim gerçekleştirilebilmiştir. Sonuç olarak ortaya çıkan 4 sınıflı çökme

uzaklığı tahmini karışıklık matrisi çizelge 4.13'teki gibi oluşmuştur. Bu uygulamaya ait başarımlar sonuçları çizelge 4.14'te verilmiştir.

Çizelge 4.13 ELM çökme uzaklığı sınıflandırma karışıklık matrisi

	0	1	2	3
0	232242	1114	1918	8663
1	775	125747	12274	162
2	1801	18991	106273	7876
3	7785	3239	62035	55369

Çizelge 4.14 ELM yöntemi çökme uzaklığı sınıflandırma başarımları

Yöntem	R Kare (test)	Doğruluk	Yakın Doğruluk
ELM	0.6744	0.804	0.9635

4.2.3.3 LSTM uygulaması

LSTM modelinin karmaşıklığı zaman sınırı nedeniyle belirli bir noktaya kadar artırılabilmiştir. 400 ünite ve 512 batch boyutu ile 50 epoch boyunca yapılan eğitim sonucunda 11 etiketli bir sınıflandırma için yeteri kadar başarımlar gösteremediği görülerek, ELM yöntemi gibi bu yöntem de 4 sınıflı bir sınıflandırma için kullanılmıştır. Eğitim gerçekleştirildikten sonra, test veri seti ile yapılan sınıflandırmanın sonucu çizelge 4.15'teki karışıklık matrisinde ve çizelge 4.16'daki başarımlar sonuçlarında görülmektedir.

Çizelge 4.15 LSTM çökme uzaklığı sınıflandırma karışıklık matrisi

	0	1	2	3
0	207080	2590	8111	27686
1	1088	113565	19943	4362
2	3595	31975	65211	34160
3	23052	7700	45429	52247

Çizelge 4.16 LSTM yöntemi çökme uzaklığı sınıflandırma başarımı

Yöntem	R Kare (test)	Doğruluk	Yakın Doğruluk
ELM	0.2023	0.6763	0.885

4.2.4 Taşma sınırı değerinin etkisinin test edilmesi

Bu aşamaya kadar yapılan çalışmalar, düğümler için tek bir taşma sınırıyla yapılan simülasyon verisiyle gerçekleştirilmiştir. Farklı taşma sınırı değerleriyle oluşan başarısızlıkların ne derece tahmin edilebildiğini tespit etmek için 2 numaralı deney ağ grupları verisi kullanılmıştır. Bu veri setinde 1066449 adet düğüm verisi bulunmaktadır. Bu verinin %70'i eğitim, %30'u test için kullanılarak, Random Forest yöntemiyle bir eğitim gerçekleştirilmiştir. Eğitim sonrası test veri seti ile yapılan ikili sınıflandırma sonuçları çizelge 4.17'de verilmektedir. Test sonucu oluşan karışıklık matrisi çizelge 4.18'de gösterilmektedir.

Çizelge 4.17 2 numaralı veri seti başarısızlık tahmini sonucu

R Kare (test)	R Kare (eğitim)	Doğruluk
0.9914	1	0.9985

Çizelge 4.18 2 numaralı veri seti başarısızlık tahmini karışıklık matrisi

	Tahmin Başarısız Değil	Tahmin Başarısız
Gerçek Başarısız Değil	69825	182
Gerçek Başarısız	284	249644

2 numaralı veri seti ile yapılan çalışmada, başarısızlığa uzaklık sınıflandırması sonucunda ortaya çıkan başarımlar çizelge 4.19'da verilmiştir.

Çizelge 4.19 2 numaralı veri seti başarısızlığa uzaklık tahmini sonucu

R Kare (test)	Doğruluk	Yakın Doğruluk
0.9668	0.9218	0.9939

Sınıflandırma sonuçlarının bu kadar doğru olmasının temel sebebi kullanılan veri setidir. 2 numaralı veri seti, tamamı aynı yoğunlukta olan ağlara yapılan saldırılar sonucu oluşan düğüm metriklerinin kaydedildiği örnekleri içermektedir. Taşma sınırı değerlerinin sabit tutulduğu denemelerde kullanılan 1 numaralı veri seti ise farklı yoğunluklardaki ağlara ait kayıtlardan oluşmaktadır. Karşılaştırmanın daha isabetli olması için 2 numaralı veri setine, 1 numaralı veri setinde bulunup 2 numaralı veri setinde bulunmayan örneklerden eklenerek birleşik bir veri seti oluşturulmuştur. 2119084 adet örnek içeren birleşik veri seti kullanılarak eğitim tekrarlanmıştır. Başarısızlık tahmini için yapılan ikili sınıflandırma sonucu oluşan başarımların değerleri çizelge 4.20'de ve karışıklık matrisi çizelge 4.21'de verilmektedir. Aynı değerlerle gerçekleştirilen başarısızlığa uzaklık sınıflandırması sonucu oluşan başarımların, çizelge 4.22'de gösterilmektedir.

Çizelge 4.20 Birleşik veri seti başarısızlık tahmini sonucu

R Kare (test)	R Kare (eğitim)	Doğruluk
0.9874	0.9999	0.9979

Çizelge 4.21 Birleşik veri seti başarısızlık tahmini karışıklık matrisi

	Tahmin Başarısız Değil	Tahmin Başarısız
Gerçek Başarısız Değil	131343	519
Gerçek Başarısız	788	503075

Çizelge 4.22 Birleşik veri seti başarısızlığa uzaklık tahmini sonucu

R Kare (test)	Doğruluk	Yakın Doğruluk
0.9467	0.9083	0.9898

Testin sonucunda, düğümler için taşma sınırının farklı değerler olduğu durumlarda da başarısızlık tahminin başarılı olduğu görülmüştür. Birleşik veri seti için ikili sınıflandırma sonuçları, 1 numaralı veri seti ile yapılan ikili sınıflandırma sonuçlarıyla son derece benzer niteliktedir. Aradaki küçük farkın sebebi olarak, birleşik veri setindeki örneklerin içerisinde aynı yoğunluktaki ağa ait örneklerin çokluğu gösterilebilir. Birleşik veri seti için çok etiketli sınıflandırma sonuçları ile 1 numaralı veri seti için çok etiketli sınıflandırma sonuçları arasında da benzer bir ilişkinin söz konusu olduğu görülmüştür. Çok etiketli sınıflandırma problemi için, veri setinin aynı yoğunlukta ağa ait veriden çok sayıda örnek içermesinin pozitif etkisinin ikili sınıflandırma problemine göre daha büyük olduğu gözlemlenmiştir.

4.2.5 Parametre etkisinin test edilmesi

Uygulamalarda girdi olarak kullanılan parametrelerle ilgili açıklama, çökme tahmini bölümünde yapılmıştır. Buna göre girdi olarak kullanılan parametrelerden 2 ve 4 numaralı veri, haberleşme ihtiyacı olmadan; 5, 6 ve 7 numaralı veri ise yalnızca komşularıyla haberleşerek düğüm tarafından hesaplanabilecek şekildedir. 1 ve 8 numaralı girdi verisi ise ağın ilk değerleridir ve en baştan bilinebilir. Girdi değerleri içinde, ağ üzerinde merkezi hesaplama gerektiren yalnızca 3 ve 9 numaralı veri bulunur.

Düğüm bazlı çökme tahmini için ağ geneli haberleşme gerektirmeyen girdi değerleriyle nasıl bir başarımlar elde edileceğinin tespit edilmesi için girdi parametreleri değiştirilerek sonuçlar kayıt altına alınmıştır. Buna göre 1, 2, 4, 5, 6, 7 ve 8 numaralı 7 adet girdi verisiyle Random Forest yöntemi kullanılarak bir eğitim gerçekleştirilmiştir. Bu eğitimin sonucunda oluşan test verisi karışıklık matrisi çizelge 4.23'teki gibidir. 9 girdi parametresiyle yapılan eğitim sonuçları çizelge 4.7'de gösterilmektedir. Girdi

parametrelerinin sonuca etkisini gösteren eğitim sonuçları karşılaştırmalı olarak çizelge 4.24'te görülmektedir.

Çizelge 4.23 Düğüm parametreleri ikili sınıflandırma karışıklık matrisi

	Tahmin Başarısız Değil	Tahmin Başarısız
Gerçek Başarısız Değil	155025	88912
Gerçek Başarısız	40100	720623

Çizelge 4.24 Düğüm parametreleri etkisi testi sonuçları

Kullanılan girdiler	R Kare (test)	Doğruluk (test)
1,2,3,4,5,6,7,8,9	0.9794	0.9962
1,2,3,4,5,6,7,8	0.5261	0.9129
1,2,4,5,6,7,8,9	0.76538	0.9568
1,2,4,5,6,7,8	0.3015	0.8716

Düğümler için başarısızlığa uzaklık etiketlerinin tahmini uygulamasında bütün girdi parametreleriyle, 0.9238 R Kare değeri ve 0.9021 doğruluk değeri elde edilmiştir. Bu uygulama sonucunda 0.9822 yakın doğruluk değeri elde edilmiştir. Ağ geneli haberleşme gerektiren 3 ve 9 numaralı girdi parametreleri çıkartılarak, Random Forest yöntemi kullanılarak yapılan eğitim sonucunda ise 0.2561 R Kare değeri, 0.503 doğruluk ve 0.692 yakın doğruluk başarımları elde edilmiştir. Söz konusu parametreler kullanılmadan, sonuçların yarısından fazlası doğru tahmin edilebilmiştir.

Ağ geneli haberleşme gerektiren girdi parametrelerinin sonuca büyük bir etkisi olduğu görülmüştür. Ancak bu parametreler kullanılmadığında da belli bir başarımla düğüm başarısızlık tahmini yapılabilmektedir.

5. SONUÇLAR

Bu çalışmanın en önemli sonuçlarından birisi, karmaşık ağlarda düğüm bazlı olarak başarısızlık tahmini yapılabileceğinin gösterilmesidir. Farklı makine öğrenmesi yöntemlerinin bu tahmin için kullanılabileceği görülmüştür. Bir düğümün başarısız olup olmayacağı ikili bilgisi ve başarısız olması beklenen düğümün kaç adım sonra başarısız olacağı bilgisi, düğümüne ait metrikler girdi olarak kullanılarak büyük bir başarıyla elde edilebilmiştir.

Ağdaki herhangi bir düğümün başarısız olup olmayacağı bilgisi için %99.62 oranında doğru tahmin edilmiştir. Düğümün başarısızlığa uzaklık etiketleri 11 farklı sınıftan oluşmaktadır. Bu uzaklığın tahmininde ise %90.21 oranında bir doğruluk elde edilmiştir. Uzaklığın tahmininde 1 uzaklık hata ile yapılan tahminleri başarılı olarak kabul ettiğimizde ortaya %98.22'lik bir doğruluk yüzdesi çıkmaktadır. Bu gibi sınıflandırma problemlerinde doğruluk oranı yerine R Kare değerinin kullanılması daha doğru bir yaklaşımdır. R Kare değeri ortaya çıkan başarımın, sınıf sayısından ve sınıflardaki örnek sayısından bağımsız olarak değerlendirilebilmesini sağlar. Bu açıdan bakıldığında düğümün başarısız olup olmayacağı ikili bilgisi 0.9794, düğümün başarısızlığa uzaklık etiketi 0.9238 R Kare değerleriyle tespit edilmiştir. Kullanılan üç yöntem için de R Kare değerleri pozitif çıkmış ve her üç yöntemin de problemi öğrendiği görülmüştür.

Kullanılan yöntemler "Çökme tahmini yöntemlerinin karşılaştırılması" bölümünde performans açısından karşılaştırılmıştır. Bu çalışmadaki ikili sınıflandırma probleminde Random Forest yönteminin en iyi sonucu verdiği görülmüştür. ELM yönteminin temel problemi olarak yüksek miktardaki RAM ihtiyacı karşımıza çıkmıştır. ELM modelinin karmaşıklığı hafıza sınırı nedeniyle yeteri kadar artırılamamıştır. ELM yönteminin çalışma süresi olarak diğer yöntemlerden daha kısa bir zaman aldığı görülmüştür. LSTM yönteminin temel sorunu olarak yüksek miktardaki işlem sayısı ve buna bağlı olarak uzun eğitim zamanı karşımıza çıkmıştır. Çalışma süresi sınırlandırıldığında tahmin başarımının istenen seviyeye gelemediği, daha iyi başarımlar için modelin çok daha uzun zaman çalıştırılması gerektiği görülmüştür. Random Forest yöntemi hem çalışma süresi ve hafıza kullanımı açısından uygulanabilir olmasıyla hem de bu problem

için yüksek tahmin başarımıyla, düğüm başarısızlık tahmininde kullanılacak uygun bir yöntem olarak görülmüştür.

Düğümler için taşma sınırı değerinin, düğüm başarısızlıklarının tahminindeki etkisine dair deneyler yapılmış ve sonuç olarak geliştirilen yöntemin taşma sınırı değerinin farklılığından fazla etkilenmediği görülmüştür. Sabit taşma sınırı değeriyle yapılan ikili sınıflandırma sonucu 0.9794 R Kare değeri ve 0.9962 doğruluk oranı oluşurken, farklı taşma sınırı değerleriyle yapılan ikili sınıflandırma sonucunda 0.9874 R Kare değeri ve 0.9979 doğruluk yüzdesi oluşmuştur. Çok etiketli sınıflandırma karşılaştırması da benzer şekildedir. Sabit taşma sınırı değeriyle gerçekleşen çökmelere ait veri ile yapılan eğitim sonucu ve farklı taşma sınırı değerleriyle gerçekleşen çökmelere ait veri ile yapılan eğitim sonucu birbirine oldukça yakındır. Bu durum, geliştirilen çözümün, taşma sınırı farklı olan ağ örnekleri için de başarılı bir şekilde çalıştığını göstermektedir.

Düğümlerin yalnızca komşularıyla iletişim kurarak elde edebildikleri 7 adet veri kullanılarak yapılan eğitimlerde pozitif bir korelasyon elde edilmiştir. Yani düğümün, kolay yoldan hesaplayabileceği veri ile başarısız olup olmayacağı arasında bir ilişki kurabileceği görülmüştür. Ancak bu girdilerle yapılan eğitim, bütün 9 girdi kullanılarak yapılan eğitime göre daha düşük bir başarımla göstermiştir.

5.1 Öneriler

Bu çalışmada düğüm ve çizge metriklerinden 21 metrik, tahmin çalışmalarında denenmiş ancak 9 tanesi haricinde diğer metriklerin başarıma önemli katkılarda bulunmadıkları görülmüştür. Bazı metrikler birbirlerinin alternatifi olduğundan içlerinden yalnızca bir tanesinin kullanılması yeterli olmuştur. Farklı düğüm ve çizge metrikleri deneyerek bu başarımlar artırılabilir. Özellikle düğümün kendi başına hesaplayabileceği, tahmin için faydalı metrikler, ağ için genel haberleşme gerektirmeyen bir çözümü daha başarılı kılabilir.

En sık kullanılan makine öğrenmesi yöntemlerinden Random Forest, son yıllarda oldukça popüler olan LSTM ve diğer yöntemlerden daha farklı eğitim tarzıyla ELM yöntemi bu çalışma kapsamında tahminler için kullanılmış ve makine öğrenmesi yöntemlerinin düğüm başarısızlık tahmini için kullanılabilir olduğu görülmüştür. Ancak daha farklı makine öğrenmesi yöntemleri deneyerek bu başarımlar artırılabilir.

LSTM yöntemi zaman bilgisinin önemli olduğu problemler için daha iyi sonuçlar verebilmektedir. Düğüm başarısızlığı tahmininde kullanılan veri de zaman olarak sıralıdır. LSTM yönteminde, yalnızca çalışma süresi artırılarak daha fazla epoch boyunca eğitim sürdürülerek başarımlar artırılabilir.

Bu çalışmada, yöntemlerin hafıza ve zaman açısından daha uygulanabilir olması için 500 elemanlı ağlarla çalışılmıştır. Ancak daha büyük ağlar üretilerek veri seti çeşitliliği sağlanıp, daha kapsayıcı bir başarısızlık tahmin modeli geliştirilebilir. Bilgisayar donanımları geliştikçe bu tarz kapsayıcı çalışmalar daha uygulanabilir hale gelecektir.

Bu çalışma, bölgesel saldırı sonrası oluşan basamaklı başarısızlıkları incelemektedir. Ancak basamaklı başarısızlık durumu doğal hatalar sebebiyle de oluşabilmektedir. Hata sonucu basamaklı başarısızlık oluşturacak şekilde bir veri seti oluşturularak mevcut veri setine eklenebilir. Bu şekilde daha kapsayıcı bir eğitim gerçekleştirilerek daha isabetli tahminler yapılabilir. Benzer şekilde, statik saldırı gibi farklı saldırı tipleriyle saldırılar gerçekleştirilerek oluşturulan veri setleri de kullanılabilir.

Basamaklı başarısızlık için önemli değerlerden birisi, düğümler için taşma sınırıdır. Bu değerlerin etkisi, çalışma kapsamında test edilmiş ve sonuçları gösterilmiştir. Ancak 3 farklı taşma sınırı ile yapılan bu testler genişletilerek daha değişken taşma sınırlarının kullanıldığı bir veri seti oluşturulabilir ve bu şekilde öğrenme daha kapsayıcı bir hale getirilebilir. Taşma sınırı değerinin başarıma etkisinin testi için, ağ yoğunluğu açısından daha düzgün dağıtılmış bir veri seti oluşturularak çalışma daha isabetli hale getirilebilir.

KAYNAKLAR

- Albert, R. and Barabasi, A.L. 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics*, Vol 74, 47-98.
- Albert, R., Jeong, H. and Barabasi, A.L. 2000. Error and attack tolerance of complex networks, *Nature* 406, 378-382.
- Ash, J. and Newth, D. 2007. Optimizing Complex Networks for Resilience Against Cascading Failure. *Physica A*, 380, 673–683.
- Barabasi, A. L. and Bonabeau, E. 2003. Scale-Free Networks. *Scientific American*, Vol 288, Num 5, 60-69.
- Bhatia, G., Khemani, R., Nebhani, N. and Bhatia, G. 2015. Forecasting Cascading Failures in Power Grids Using Cascading Method. *International Journal of Emerging Technology and Advanced Engineering*, Vol 5, Issue 2, 357-362.
- Botchkarev, A. 2018. Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology. *Interdisciplinary Journal of Information, Knowledge, and Management*, 2019, 14, 45-79.
- Breiman, L. 2001. Random Forests. *Machine Learning*, Vol 45, 5-32.
- Byeon, W., Breuel, T. M., Raue, F. and Liwicki, M. 2015. Scene Labeling with LSTM Recurrent Neural Networks. *IEEE CVPR 2015*, 7298977.
- Crucitti, P., Latora, V. and Marchiori, M. 2004. Model for Cascading Failures in Complex Networks. *Physical Review E*, 69(4), 045104.
- Diestel, R. 2000. *Graph Theory*. Springer, New York.
- Duenas-Osorio, L. and Vemuru, S. M. 2009. Cascading Failures in Complex Infrastructure Systems. *Structural Safety*, 31, 157-167.
- Freeman, L. C. 1979. Centrality in Social Networks Conceptual Clarification *Social Networks*, 1 (1978/1979), 215-239.
- Gers, F. A., Schmidhuber, J. and Cummins, F. 2000. Learning to Forget: Continual Prediction with LSTM. *Neural Computation* 12, 2451-2471.
- Graves, A., Jaitly, N. and Mohamed, A. 2013. Hybrid speech recognition with Deep Bidirectional LSTM. *IEEE ASRU 2013*, 6707742.
- Havlin, S. and Kenett, D. Y. 2015. Cascading Failures in Interdependent Economic Networks. *Proceedings of the International Conference on Social Modeling and Simulation, Econophysics Colloquium 2014*, Springer, 87–97.
- Hines, P., Balasubramaniam, K. and Sanchez, E. C. 2009. Cascading Failures in Power Grids. *IEEE Potentials*, Vol 28, 24-30.

- Holme, P., Kim, B. J., Yoon, C. N. and Han, S. K. 2002. Attack Vulnerability of Complex Network. *Physical Review E*, 65 (2002), 056109.
- Huang, G., Zhu, Q. and Siew, C. 2006. Extreme Learning Machine: Theory and Applications. *Neurocomputing*, 70 (2006), 489–501.
- Latora, V. and Marchiori, M. 2001. Efficient Behavior of Small-World Networks. *Physical Review Letters*, Vol 87, Num 19, 198701.
- Majdandzic, A., Podobnik, B., Buldyrev, S.V. and Kenett, D. Y. 2014. Spontaneous Recovery in Dynamical Networks. *Nature Physics*, 10, 34–38.
- Motter, A. E. and Lai, Y. C. 2002. Cascade Based Attacks on Complex Networks. *Physical Review E*, 66 (2), 065102.
- Motter, A.E. 2004. Cascade Control and Defense in Complex Network. *Physical Review, Letter* 93, 098701.
- Pi, R., Cai, Y., Li, Y. and Cao, Y. 2018. Machine Learning Based on Bayes Networks to Predict the Cascading Failure Propagation. *IEEE Access*, Vol 6, 44815-44823.
- Russell, S. J. and Norvig, P. 2016. *Artificial Intelligence: A Modern Approach*. Pearson Education Inc., Upper Saddle River, New Jersey, 2016.
- Sanner, R. M. and Slotine, J. E. 1992. Gaussian Networks for Direct Adaptive Control. *IEEE Transactions on Neural Networks*, Vol 3, Issue 6, 837-863.
- Schäfer, B., Witthaut, D., Timme, M. and Latora, V. 2018. Dynamically Induced Cascading Failures in Power Grids. *Nature Communications* 9, Art. 1975.
- Shunkun, Y., Jiaquan, Z. and Dan, L. 2016. Prediction of Cascading Failures in Spatial Networks. *PLoS ONE*, 11(4), e0153904.
- Soltan, S., Mazauric, D. and Zussman, G. 2014. Cascading Failures in Power Grids: Analysis and Algorithms. *Proceedings of the 5th International Conference on Future Energy Systems*. 11-13 Haziran 2014. Cambridge/Birleşik Krallık
- Watts, D. J. and Strogatz, S. H. 1998. Collective Dynamics of Small-World Networks. *Nature*, 393 (6684), 440–442.
- Yi, C., Bao, Y., Jiang, J., Xue Y. and Dong, Y. 2014. Cascading Failures of Social Networks Under Attacks. *IEEE/ACM ASONAM 2014*, 679-686.
- Zarrabian, S., Belkacemi, R. and Babalola, A. A. 2016. Reinforcement Learning Approach for Congestion Management and Cascading Failure Prevention with Experimental Application. *Electric Power Systems Research*, Vol 141, 179-190.
- Zhao, L., Park, K. and Lai, Y. C. 2004. Attack Vulnerability of Scale Free Networks Due to Cascading Breakdown. *Physical Review, E* 70 (2004), 035101.
- Zhou, Y. and Wang, J. 2018. Efficiency of Complex Networks Under Failures and Attacks: A Percolation Approach. *Physica A*, 512(2018), 658-664.

Zhu, L., Liu, X., Yu, L. and Wu, X. 2016. Model of Cascading Failures for Communication Networks. International Journal of Computer and Communication Engineering, Vol 5, Num 5, 302-310.



ÖZGEÇMİŞ

Adı Soyadı : Ahmet TÜRK
Doğum Yeri : Bakırköy
Doğum Tarihi : 21/10/1988
Medeni Hali : Bekar
Yabancı Dili : İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise : Fahrettin Kerim Gökay Anadolu Lisesi (2006)
Lisans : Gebze Yüksek Teknoloji Enstitüsü Mühendislik Fakültesi
Bilgisayar Mühendisliği (2015)
Yüksek Lisans : Ankara Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği A.B.D. (09/2016 - 08/2019)

Çalıştığı Kurumlar ve Yıl

Katmerciler Araç Üstü Ekipman San. Tic. A.Ş. Ankara Ar-Ge Departmanı (2016-2018)
Simsoft Bilgi Teknolojileri İstanbul Ofisi (2016)
TÜBİTAK - BİLGEM B750 - Sınır ve İç Güvenlik Özel Destek Sistemleri (2012-2016)